HEWLETT
PACKARD

HEWLETT-PACKARD COMPANY
LOGIC SYSTEMS DIVISION

HP 64000
Logic Development
System

SYSTEM STATUS BULLETIN

Part Number: **5958-6018**
E1088

Printed: **OCTOBER 1988**

HP STARS II

64000 SOFTWARE STATUS BULLETIN

Logic Systems Division

Issue 88.10A * ALL PRODUCTS

OCTOBER, 1988

This document supersedes all previously dated SSBs.

HEWLETT
PACKARD

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

READER COMMENT SHEET

STARS II SSB (STARS B)

Issue _____._____ DATE _____/_____/_____

We welcome your evaluation of this bulletin. Your comments and suggestions help us to improve our publications. Please use additional pages if necessary.

Is this bulletin technically accurate?	Yes [ ] No [ ]	(If no, explain under Comments, below.)
Are the concepts and wording easy to understand?	Yes [ ] No [ ]	(If no, explain under Comments, below.)
Is the format of this bulletin convenient in size, arrangement and readability?	Yes [ ] No [ ]	(If no, explain or suggest improvements under Comments, below.)

Comments:

****

Date: ____________

FROM:

Name ______________________________

Company ______________________________

Address ______________________________

____

____

STARS B
Printed in U.S.A.

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, CO.

POSTAGE WILL BE PAID BY ADDRESSEE

STARS Administration
Hewlett-Packard Company
Logic Systems Division (LSD)
P.O. Box 617
Colorado Springs, Colorado. 80901-0617

FOLD FOLD

PREFACE

This Software Status Bulletin (SSB) documents all known problems in the software product line designated on the cover page. The SSB is derived from Known Problem Reports (KPR) which result from Service Requests (SR) submitted by users of these products. The SSB is provided as a benefit of Hewlett-Packard's Account Management Support, Response Center Support, Software Materials Subscription, and Software Notification Service.

Not all SR's submitted to HP are listed in the SSB. Ones which involve problems that cannot be duplicated, requests for enhancements and misunderstandings about an application or a feature are not listed in the SSB. SR's which refer to a previously documented problem are cross referenced within the report which originally identified the problem. Every SR verified by an HP Systems Engineer and sent to the factory is assigned a unique identifier and acknowledged by letter to the submitter. When the SR is classified as a documentation problem or a software design fault, a KPR is written for it. The KPR then appears in the next issue of the SSB. After the problem has been corrected and signed off by Product Assurance, the fact is noted in the KPR with the following statement: "Date fix signed off : mm/dd/yy Rel: uu.ff.". When a new software release is made for the product line, all problems that were corrected in that release are reported in the Software Release Bulletin for that release and the KPR's are removed from the SSB.

The SSB is distributed in complete form once every calendar quarter. Between quarterly issues, monthly issues containing only those problems documented since the last SSB issue are distributed. This means, that to have a complete list of all outstanding problems you must have the last quarterly issue and all monthly issues since that quarterly.

Of the five sections contained in the SSB, only the last (known problem reports) has page numbers. The product, KPR number and keyword indices all reference these page numbers to direct the user to a particular area or individual detailed report. The five sections are described below:

SOFTWARE RELEASE CONTENTS

This section gives you the release ID of the current software release and the previous software release for the product line. Additionally, the current and previous update/fix levels are given for each product contained in the product line.

PRODUCT INDEX

The monthly issues have one product index. The quarterly issue has two; the first referencing the problems that were documented since the last monthly issue, and the second referencing problems that were reported in a previous issue of the SSB. Each unique product name/number has an entry listing the page number where the KPR's against that product begin.

KPR NUMBER INDEX

A sequential list of KPR numbers with the corresponding page number where the KPR can be found.

KEYWORD INDEX

This index is sorted by product name, keyword, product number (including update/fix level) and by KPR number in that order. Along with the sort items, each entry has a brief (72 character) description and the page number where the KPR can be found. In the quarterly issue, entries that are new since the last update are denoted by an asterisk (*) immediately following the KPR number.

KNOWN PROBLEM REPORTS

Each report contains all the available information relevant to the problem.

Product name		Product number	uu.ff	prev
1000L ASSEMB		64852	01.00	00.00
1802 ASSEMB		64848	01.00	00.00
2A03 EMUL		64209	01.00	00.00
6301V EMULATION	300	64206S004	01.20	01.10
6301V/03R EMUL		64206	01.01	00.00
6301X EMULATION	300	64207S004	01.10	01.00
6301X/03X EMUL		64207	01.00	00.00
6301Y EMULATION	300	64208S004	01.10	01.00
6301Y/03Y EMUL		64208	01.00	00.00
64000 UX GENERIC		64003S001	01.00	00.00
64000 UX GENERIC		64003S004	01.00	00.00
64000-UX OP-ENV	300	64801S004	01.80	01.60
64180 ASSEMB		64864	01.00	00.00
64180 ASSEMB	300	64864S004	01.01	01.01
64180 EMUL		64180	01.10	01.00
64180 EMUL		64180S004	01.00	00.00
64340 P.V.		64340	01.00	00.00
650X ASSEMB		64843	01.80	00.00
650X ASSEMB	300	64843S004	01.80	00.00
650X ASSEMB	500	64843S001	01.80	01.20
650X ASSEMB	DOS	64843S006	01.80	00.00
650X ASSEMB	VAX	64843S003	01.80	01.20
6800 C		64821	02.20	02.10
6800 C	300	64821S004	02.20	02.10
6800 C	500	64821S001	02.20	02.10
6800 C	VAX	64821S003	02.20	02.10
6800 EMULATION		64212	01.05	00.56
6800 INTERFACE	300	64672S004	01.00	00.00
6800 PASCAL		64811	02.00	01.90
6800 PASCAL	300	64811S004	02.00	01.90
6800 PASCAL	500	64811S001	02.00	01.90
6800 PASCAL	VAX	64811S003	02.00	01.90
6800 PRE-PROCESSOR		64672	00.53	00.00
6800/2 ASSEMB		64841	01.80	01.14
6800/2 ASSEMB	300	64841S004	01.80	01.00
6800/2 ASSEMB	500	64841S001	01.80	01.30
6800/2 ASSEMB	DOS	64841S006	01.80	00.00
6800/2 ASSEMB	VAX	64841S003	01.80	01.40
68000 12MHZ EMUL	FW	64742	00.05	00.04
68000 12MHZ EMUL	300	64742S004	01.00	00.00
68000 12MHZ EMUL	DOS	64742S006	01.01	01.00
68000 12MHZ EMUL	VAX	64742S003	00.01	00.00
68000 16MHZ EMUL	FW	64743	00.01	00.00
68000 16MHZ EMUL	300	64743S004	00.01	00.00
68000 16MHZ EMUL	DOS	64743S006	00.01	00.00
68000 16MHZ EMUL	VAX	64743S003	00.01	00.00
68000 ASSEMB		64845	02.10	01.13
68000 ASSEMB	300	64845S004	02.10	01.30
68000 ASSEMB	500	64845S001	02.10	01.70
68000 ASSEMB	DOS	64845S006	02.11	02.10
68000 ASSEMB	VAX	64845S003	02.10	02.00
68000 BBA	300	64380S004	01.00	00.00
68000 BBA	300	64380S005	01.10	00.00
68000 C		64819	02.20	02.10
68000 C	300	64819S004	02.20	02.10

Product name		Product number	uu.ff	prev
68000 C	500	64819S001	02.20	02.10
68000 C	VAX	64819S003	02.20	02.10
68000 DQ EMUL	300	64243S004	01.30	01.20
68000 DQ SW ANAL		64331B	01.02	01.01
68000 DQ SW ANALYZER		64341G	01.03	01.02
68000 EMUL 12.5 MHZ		64243	01.01	01.01
68000 EMULATION		64242	01.07	01.06
68000 EMULATION	300	64242S004	01.00	00.00
68000 INTERFACE	300	64674S004	01.00	00.00
68000 MONITOR		64742-11001	00.04	00.00
68000 PASCAL		64815	02.00	01.90
68000 PASCAL	300	64815S004	02.00	01.90
68000 PASCAL	500	64815S001	02.00	01.90
68000 PASCAL	VAX	64815S003	02.00	01.90
68000 PRE-PROCESSOR		64670	01.00	00.56
68000 PRE-PROCESSOR		64674	01.00	00.00
68000 SW ANAL		64331	02.03	02.02
68000 SW ANALYZER		64341B	02.02	02.01
68000C AXLS COMP	300	64902S004	01.00	00.00
68000C AXLS COMP	800	64902S005	02.00	00.00
68008 EMULATION		64244	01.01	01.01
68008 EMULATION	300	64244S004	01.30	01.10
68008 INTERFACE	300	64673S004	01.00	00.00
68008 PRE-PROCESSOR		64673	00.65	00.00
68008 SW ANAL		64337	01.02	01.01
6801/3 EMULATION		64256	01.04	00.70
6801/3 EMULATION	300	64256S004	01.00	00.00
68010 16MHZ EMUL	FW	64745	00.01	00.00
68010 16MHZ EMUL	300	64745S004	00.01	00.00
68010 16MHZ EMUL	DOS	64745S006	01.01	01.00
68010 16MHZ EMUL	VAX	64745S003	00.01	00.00
68010 DQ SW ANAL		64334B	01.02	01.01
68010 DQ SW ANALYZER		64341I	01.02	01.01
68010 EMUL 12.5 MHZ		64245	01.01	01.01
68010 EMUL 12.5M	300	64245S004	01.30	01.20
68010 G.P. EMUL	300	64249S004	01.00	00.00
68010 G.P. EMULATOR		64249	01.02	01.01
68010 SW ANAL		64334	02.03	02.02
68010 SW ANALYZER		64341D	02.02	02.01
6802 EMULATION		64213	01.05	00.56
68020 ASSEMB	300	64870S004	01.00	00.00
68020 BBA	300	64381S004	01.10	01.00
68020 BBA	800	64381S005	01.20	01.00
68020 EMUL	300	64410S004	02.10	02.00
68020 EMUL	300	64416S004	02.00	01.00
68020 INV ASSEMB		64675	01.00	00.00
68020 INV ASSEMB	300	64675S004	01.00	00.00
68020C AXLS COMP	300	64903S004	01.10	01.00
68020C AXLS COMP	800	64903S005	02.00	01.00
6805 U&R EMUL	300	64192S004	01.00	00.00
6805 E EMUL	300	64195S004	01.00	00.00
6805 G EMUL	300	64194S004	01.00	00.00
6805 P EMUL	300	64193S004	01.00	00.00
6805/9 ASSEMB		64844	01.90	01.10
6805/9 ASSEMB	300	64844S004	01.90	01.00

Product name	Product number	uu.ff	prev
6805/9 ASSEMB 500	64844S001	01.90	01.30
6805/9 ASSEMB DOS	64844S006	01.90	00.00
6805/9 ASSEMB VAX	64844S003	01.90	01.40
6805U/R&P EMULATION	64192	01.07	01.06
6805U/R&P EMULATION	64193	01.07	01.05
6805U/R&P EMULATION	64194	01.07	01.05
6809 C	64822	01.90	01.80
6809 C 300	64822S004	01.90	01.80
6809 C 500	64822S001	01.90	01.80
6809 C VAX	64822S003	01.90	01.80
6809 EMULATION	64215	01.08	00.56
6809 EMULATION 300	64215S004	01.00	00.00
6809 PASCAL	64813	01.70	01.60
6809 PASCAL 300	64813S004	01.70	01.60
6809 PASCAL 500	64813S001	01.70	01.60
6809 PASCAL VAX	64813S003	01.70	01.60
6809/E INTERFACE 300	64671S004	01.00	00.00
6809/E PRE-PROCESSOR	64671	00.49	00.00
6809E EMULATION	64216	01.08	00.56
6809E EMULATION 300	64216S004	01.00	00.00
68HC11 EMUL	64265	01.01	01.00
68HC11 EMUL 300	64265S004	01.10	01.00
68HCII ASSEMB	64865	01.40	01.30
68HCII ASSEMB 300	64865S004	01.40	01.30
68HCII ASSEMB 500	64865S001	01.40	01.30
68HCII ASSEMB DOS	64865S006	01.40	01.30
68HCII ASSEMB VAX	64865S003	01.40	01.30
70016 EMUL (JLO) 300	64294S004	01.10	01.00
70108 EMUL	64295	01.00	00.00
70108 EMUL (JLO) 300	64295S004	01.10	01.00
70108 SW ANAL	64339	01.00	00.00
70108 SW ANALYZER	64342B	01.00	00.00
70116 EMUL	64294	01.00	00.00
70116 SW ANAL	64338	01.00	00.00
70116 SW ANALYZER	64342A	01.00	00.00
70208 EMUL	64297	01.00	00.00
70208 EMUL	64297S004	01.00	00.00
70216 EMUL	64296	01.00	00.00
70216 EMUL	64296S004	01.00	00.00
78310/12 ASSEMB	64866	01.02	01.00
78310/12 ASSEMB 300	64866S004	01.00	01.00
80186 EMUL FW	64764	00.01	00.00
80186 EMUL 300	64764S004	00.01	00.00
80186 EMUL DOS	64764S006	01.02	01.01
80186 EMUL VAX	64764S003	00.01	00.00
80186 EMULATION	64224	01.05	01.04
80186 EMULATION 300	64224S004	01.20	01.10
80186 INTERFACE 300	64658S004	01.00	00.00
80186 MONITOR	64764-11001	00.01	00.00
80186 PRE-PROCESSOR	64658	00.57	00.00
80186 SW ANAL	64335	02.03	02.02
80186 SW ANALYZER	64341E	02.02	02.01
80188 EMUL FW	64765	00.01	00.00
80188 EMUL 300	64765S004	00.01	00.00
80188 EMUL DOS	64765S006	01.02	01.00

Product name	Product number	uu.ff	prev
80188 EMUL VAX	64765S003	00.01	00.00
80188 EMULATION	64225	01.03	01.02
80188 EMULATION 300	64225S004	01.20	01.10
80188 MONITOR	64765-11001	00.01	00.00
80188 SW ANAL	64336	02.04	02.03
80188 SW ANALYZER	64341F	01.02	01.01
80196 EMUL FW	64771	00.01	00.00
80286 EMULATION	64228	01.02	01.01
80286 INTERFACE 300	64657S004	01.00	00.00
80286 PRE-PROCESSOR	64657	00.67	00.00
80286 UDE	64227	01.00	00.00
80286B ASSEMB	64859	01.50	01.40
80286B ASSEMB 300	64859S004	01.50	01.40
80286B ASSEMB 500	64859S001	01.50	01.40
80286B ASSEMB DOS	64859S006	01.50	01.40
80286B ASSEMB VAX	64859S003	01.50	01.40
8048 ASSEMB	64846	01.80	00.00
8048 ASSEMB 300	64846S004	01.80	00.00
8048 ASSEMB 500	64846S001	01.80	01.20
8048 ASSEMB DOS	64846S006	01.80	00.00
8048 ASSEMB VAX	64846S003	01.80	01.20
8048 EMULATION	64262	01.07	01.06
8051 ASSEMB	64855	01.80	01.07
8051 ASSEMB 300	64855S004	01.80	01.10
8051 ASSEMB 500	64855S001	01.80	01.40
8051 ASSEMB DOS	64855S006	01.80	00.00
8051 ASSEMB VAX	64855S003	01.80	01.50
8051 EMULATION	64264	01.04	01.02
8051 EMULATION 300	64264S004	01.00	00.00
8080 EMULATION	64202	01.07	01.06
8080/5 ASSEMB	64840	01.80	00.00
8080/5 ASSEMB 300	64840S004	01.80	00.00
8080/5 ASSEMB 500	64840S001	01.80	01.20
8080/5 ASSEMB DOS	64840S006	01.80	00.00
8080/5 ASSEMB VAX	64840S003	01.80	01.20
8080/5 INTERFACE 300	64655S004	01.00	00.00
8080/5 PRE-PROCESSOR	64655	00.56	00.00
8085 B PASCAL	64825	02.00	01.90
8085 B PASCAL 300	64825S004	02.00	01.90
8085 B PASCAL 500	64825S001	02.00	01.90
8085 B PASCAL VAX	64825S003	02.00	01.90
8085 C	64826	02.20	02.10
8085 C 300	64826S004	02.20	02.10
8085 C 500	64826S001	02.20	02.10
8085 C VAX	64826S003	02.20	02.10
8085 EMULATION	64203	01.08	01.07
8085 EMULATION 300	64203S004	01.40	01.30
8085 PASCAL	64810	00.70	00.00
8086 EMUL FW	64762	00.01	00.00
8086 EMUL 300	64762S004	00.01	00.00
8086 EMUL DOS	64762S006	00.01	00.00
8086 EMUL VAX	64762S003	00.01	00.00
8086 MONITOR	64762-11001	00.01	00.00
8086 DQ EMUL 300	64220S004	01.30	01.20
8086 DQ EMULATION	64220	01.01	00.00

Product name	Product number	uu.ff	prev
8086 DQ SW ANAL	64332B	01.03	01.01
8086 EMUL 300	64222S004	01.00	00.00
8086 EMULATION	64222	01.07	01.06
8086 PLM A,LLL 500	64891S001	01.00	00.00
8086 PLM ASM LLL	64891S003	01.10	00.00
8086 PLM C,A,LLL 500	64890S001	01.00	00.00
8086 PLM CMP ASM LLL	64890S003	01.10	00.00
8086 SW ANAL	64332	02.03	02.02
8086 SW ANALYZER	64341A	01.02	01.01
8086 SYMBOL CONV	64892S003	01.10	00.00
8086 SYMBOL CONV 500	64892S001	01.00	00.00
8086/8 ASSEMB	64853	02.80	02.70
8086/8 ASSEMB 300	64853S004	02.80	02.70
8086/8 ASSEMB 500	64853S001	02.80	02.70
8086/8 ASSEMB DOS	64853S006	02.80	02.70
8086/8 ASSEMB VAX	64853S003	02.80	02.70
8086/8 C	64818	03.80	03.70
8086/8 C 300	64818S004	03.80	03.70
8086/8 C 500	64818S001	03.80	03.70
8086/8 C VAX	64818S003	03.80	03.70
8086/8 INTERFACE 300	64653S004	01.00	00.00
8086/8 PASCAL	64814	03.60	03.50
8086/8 PASCAL 300	64814S004	03.60	03.50
8086/8 PASCAL 500	64814S001	03.60	03.50
8086/8 PASCAL VAX	64814S003	03.60	03.50
8086/8 PRE-PROCESSOR	64653	00.10	00.00
8088 DQ SW ANALYZER	64341C	01.02	01.02
8088 SW ANAL	64333	02.03	02.02
8088 DQ EMUL 300	64221S004	01.20	01.10
8088 DQ EMULATION	64221	01.01	00.00
8088 DQ SW ANAL	64333B	01.03	01.01
8088 EMULATION	64226	01.08	01.07
8088 EMULATION 300	64226S004	01.00	00.00
8096 ASSEMB	64860	01.80	01.70
8096 ASSEMB 300	64860S004	01.80	01.70
8096 ASSEMB 500	64860S001	01.80	01.70
8096 ASSEMB DOS	64860S006	01.80	01.70
8096 ASSEMB VAX	64860S003	01.80	01.70
9900/0 ASSEMB	64847	01.80	00.46
9900/0 ASSEMB 300	64847S004	01.80	01.00
9900/0 ASSEMB 500	64847S001	01.80	01.30
9900/0 ASSEMB DOS	64847S006	01.80	00.00
9900/0 ASSEMB VAX	64847S003	01.80	01.30
DIAG/CS 80 EXER/XFER	64934	01.04	01.03
EBPP	64304	01.03	00.70
F8/3870 ASSEMB	64849	00.01	00.00
F9450 EMUL 300	64286S004	01.00	00.00
F9450 EMULATION	64286	01.05	01.04
FILE XFER UTIL 2,3	64893S003	01.10	00.00
FILE-XFER UT 2&3 500	64893S001	01.00	00.00
GENERIC ANALYSIS FW	64740	00.03	00.02
GENERIC EMULATION FW	64700	00.05	00.04
HI SPD RS422 INTF	64037	00.02	00.01
HOST PASCAL	64817	01.04	00.46
HOST SOFTWARE / VAX	64882	02.40	02.30

Product name	Product number	uu.ff	prev
HOST SOFTWARE / 300	64883	01.10	01.00
HOST SOFTWARE / 500	64880	01.90	01.80
HP TEAMWORK 300	64711S004	02.30	02.20
HP-IB INTERFACE 300	64695S004	01.00	00.00
HP-IB PRE-PROCESSOR	64695	01.00	00.00
IMB EXTENDER	64303	01.01	00.55
INVERSE ASSEMB	64856	01.01	00.00
MICRO ASSEMB	64861	01.01	01.00
MS1750A ASSEMB	64857	01.90	01.04
MS1750A ASSEMB 300	64857S004	01.90	01.00
MS1750A ASSEMB 300	64857S006	01.00	00.00
MS1750A ASSEMB 500	64857S001	01.90	01.30
MS1750A ASSEMB VAX	64857S003	01.90	01.40
NETWORK TRANSFER 300	64887S004	01.00	00.00
NETWORK TRANSFER 500	64887S001	01.00	00.00
NETWORK TRANSFER 500	64888S001	01.20	01.00
NETWORK TRANSFER VAX	64887S003	01.10	00.00
NSC800 EMULATION	64292	01.03	01.02
NSC800 INTERFACE 300	64690S004	01.00	00.00
NSC800 PRE-PROCESSOR	64690	00.48	00.00
OPERATING SYSTEM	64100	02.11	02.10
P1750 EMUL	64288	01.00	00.00
P1750 EMUL 300	64288S004	01.10	01.00
PROM PROGRAMMER	64501	01.10	01.09
PROM PROGRAMMER 300	64501S004	01.30	01.10
ROM EMULATION	64272	01.04	01.03
RS-232 TRANSFER 300	64885	01.30	01.20
RS-232 TRANSFER 500	64884	01.40	01.30
RS-232 TRANSFER VAX	64886	01.60	01.50
SOFTKEY EDITOR 300	64790S004	02.10	01.00
SOFTKEY EDITOR 500	64790S001	02.10	01.10
STATE 80386	64659	01.00	00.00
STATE ANALYZER	64620	00.71	00.00
STATE ANALYZER	64621	01.07	00.71
STATE ANALYZER 300	64620S004	01.30	01.10
STATE_25MHZ	64320	01.01	01.00
STATE_25MHZ	64321	01.01	01.00
SW PERF ANALYZER	64310	01.11	01.10
SW PERF ANALYZER 300	64310S004	01.20	01.10
TIMING	64601A	01.04	01.03
TIMING	64601B	01.04	00.00
TIMING ANALYZER	64600	00.26	00.00
TIMING ANALYZER 300	64610S004	01.60	01.00
TIMING/STATE	64610	01.00	00.00
TMS 320 ASSEMB	64858	01.80	00.00
TMS 320 ASSEMB 300	64858S004	01.80	00.00
TMS 320 ASSEMB 500	64858S001	01.80	01.20
TMS 320 ASSEMB DOS	64858S006	01.80	00.00
TMS 320 ASSEMB VAX	64858S003	01.80	01.20
TMS 32010 MODULES	64285	01.02	01.01
TMS 32020/25 ASMB300	64867S004	01.00	00.00
TMS 32020/25 ASMB300	64867S006	01.00	00.00
TMS32020 EMUL FW	64786	00.01	00.00
TMS320C25 EMUL FW	64787	00.01	00.00
UPROG	64276	02.00	01.01

Product name		Product number	uu.ff	prev
USER DEF ASSEMB		64851	00.70	00.00
USER DEF ASSEMB	300	64851S004	02.20	02.10
USER DEF ASSEMB	300	64861S004	02.10	00.00
USER DEF ASSEMB	500	64851S001	02.20	02.10
USER DEF ASSEMB	500	64861S001	02.10	00.00
USER DEF ASSEMB	DOS	64851S006	02.20	02.11
USER DEF ASSEMB	VAX	64851S003	02.20	02.10
USER DEF ASSEMB	VAX	64861S003	02.10	00.00
USER DEF EMUL	300	64274S004	01.20	01.10
USER DEF EMULATION		64274	01.06	01.05
USER DEF INV ASM	300	64856S004	01.00	00.00
USER INTERFACE	300	64808S004	02.10	01.20
USER INTERFACE	500	64808S001	02.10	01.40
UTILITIES PKG	300	64888S003	01.50	01.10
UTILITIES PKG	300	64888S004	01.20	01.10
Z8 ASSEMB		64850	00.01	00.00
Z80 EMUL	FW	64753	00.01	00.00
Z80 EMUL	300	64753S004	00.01	00.00
Z80 EMUL	DOS	64753S006	01.01	01.00
Z80 EMUL	VAX	64753S003	00.01	00.00
Z80 MONITOR		64753-11001	00.01	00.00
Z80 ASSEMB		64842	01.90	01.11
Z80 ASSEMB	300	64842S004	01.90	01.00
Z80 ASSEMB	500	64842S001	01.90	01.30
Z80 ASSEMB	DOS	64842S006	01.91	01.90
Z80 ASSEMB	VAX	64842S003	01.90	01.40
Z80 EMULATION		64252	01.05	00.56
Z80 EMULATION	300	64252S004	01.00	00.00
Z80 INTERFACE	300	64683S004	01.00	00.00
Z80 PASCAL		64812	00.70	00.00
Z80 PRE-PROCESSOR		64683	00.56	00.00
Z80/NSC800 C		64824	02.20	02.10
Z80/NSC800 C	300	64824S004	02.20	02.10
Z80/NSC800 C	500	64824S001	02.20	02.10
Z80/NSC800 C	VAX	64824S003	02.20	02.10
Z80/NSC800PASCAL		64823	02.00	01.90
Z80/NSC800PASCAL	300	64823S004	02.00	01.90
Z80/NSC800PASCAL	500	64823S001	02.00	01.90
Z80/NSC800PASCAL	VAX	64823S003	02.00	01.90
Z8000 C		64820	02.20	02.10
Z8000 C	300	64820S004	02.20	02.10
Z8000 C	500	64820S001	02.20	02.10
Z8000 C	VAX	64820S003	02.20	02.10
Z8000 PASCAL		64816	02.00	01.90
Z8000 PASCAL	300	64816S004	02.00	01.90
Z8000 PASCAL	500	64816S001	02.00	01.90
Z8000 PASCAL	VAX	64816S003	02.00	01.90
Z8001 EMUL	300	64232S004	01.00	00.00
Z8001 EMULATION		64232	02.00	01.07
Z8001 INTERFACE	300	64680S004	01.00	00.00
Z8001 PRE-PROCESS		64680	00.56	00.00
Z8001/2 ASSEMB		64854	01.80	00.00
Z8001/2 ASSEMB	300	64854S004	01.80	00.00
Z8001/2 ASSEMB	500	64854S001	01.80	01.20
Z8001/2 ASSEMB	DOS	64854S006	01.80	00.00

Product name		Product number	uu.ff	prev
Z8001/2 ASSEMB	VAX	64854S003	01.80	01.20
Z8002 EMUL	300	64233S004	01.00	00.00
Z8002 EMULATION		64233	02.01	02.00
Z8002 INTERFACE	300	64681S004	01.00	00.00
Z8002 PRE-PROCESS		64681	00.56	00.00
Z80H EMULATION		64253	01.02	01.01
Z80H EMULATION	300	64253S004	01.00	00.00

Product name	Product number	SSB ISSUE DATE	Page
6301V EMULATION 300	64206S004	09/01/88	87
6301V/03R EMUL	64206	09/01/88	1
		09/01/88	89
6301X EMULATION 300	64207S004	09/01/88	2
6301Y EMULATION 300	64208S004	09/01/88	4
6301Y/03Y EMUL	64208	09/01/88	90
64 HP-UX VMS 8096 AM	64860-90901	09/01/88	91
64000 UX GENERIC	64003S004	09/01/88	92
64000-UX OP-ENV 300	64801S004	09/01/88	93
64HP-UXVMS8086/8 A M	64853-90908	09/01/88	94
650X ASSEMB	64843	09/01/88	7
6800 C	64821	09/01/88	8
		09/01/88	95
6800 C M	64821-90901	09/01/88	106
6800 PASCAL	64811	09/01/88	9
		09/01/88	108
6800-03 ASSM M	64841-90905	09/01/88	111
6800/2 ASSEMB	64841	09/01/88	15
		09/01/88	112
6800/2 ASSEMB 500	64841S001	09/01/88	115
68000 12MHZ EMUL FW	64742	09/01/88	16
		09/01/88	117
68000 12MHZ EMUL DOS	64742S006	09/01/88	17
		09/01/88	121
68000 ASSEMB	64845	09/01/88	123
68000 BBA 300	64380S004	09/01/88	128
68000 C	64819	09/01/88	18
		09/01/88	130
68000 C M	64819-90902	09/01/88	154
68000 DQ EMUL 300	64243S004	09/01/88	19
		09/01/88	156
68000 DQ SW ANALYZER	64341G	09/01/88	161
68000 EMUL 12.5 MHZ	64243	09/01/88	162
68000 EMULATION	64242	09/01/88	163
68000 EMULATION 300	64242S004	09/01/88	164
68000 HL SOFT ANAL M	64331-90902	09/01/88	169
68000 PASCAL	64815	09/01/88	20
		09/01/88	170
68000 SW ANALYZER	64341B	09/01/88	189
68000/08 EMUL M	64242-90906	09/01/88	190
68000/08 EMULHP-UX M	64243-90903	09/01/88	191
68000/08/10 ASM M	64845-90904	09/01/88	193
68000/10 EMUL M	64245-90903	09/01/88	196
68000/10 RT S-ANAL M	64341-90903	09/01/88	197
68000C AXLS COMP M	64902-90901	09/01/88	22
		09/01/88	198
68000C AXLS COMP 300	64902S004	09/01/88	23
		09/01/88	199
68008 EMULATION 300	64244S004	09/01/88	25
		09/01/88	204
6801/3 EMULATION 300	64256S004	09/01/88	208
68010 16MHZ EMUL DOS	64745S006	09/01/88	26
68010 EMUL 12.5M 300	64245S004	09/01/88	27
		09/01/88	213
68010 G.P. EMUL 300	64249S004	09/01/88	217

Product name	Product number	SSB ISSUE DATE	Page
8051 ASSEMB	64855	09/01/88	347
8051 ASSM M	64855-90902	09/01/88	351
8051 ASSM + AL REF M	64855-90905	09/01/88	352
8051 EMUL M	64264-90901	09/01/88	353
8051 EMULATION	64264	09/01/88	354
8051 EMULATION 300	64264S004	09/01/88	355
8080/5 ASSEMB	64840	09/01/88	46
		09/01/88	359
8085 B PASCAL	64825	09/01/88	47
		09/01/88	360
8085 C	64826	09/01/88	48
		09/01/88	363
8085 C M	64826-90901	09/01/88	366
8085 C 300	64826S004	09/01/88	368
8085 EMULATION	64203	09/01/88	49
8085 EMULATION 300	64203S004	09/01/88	50
		09/01/88	369
8085 PASCAL	64810	09/01/88	372
8086 DQ EMUL 300	64220S004	09/01/88	51
		09/01/88	374
8086 DQ SW ANAL	64332B	09/01/88	378
8086 EMUL 300	64222S004	09/01/88	379
8086 SW ANAL	64332	09/01/88	384
8086 SW ANALYZER	64341A	09/01/88	385
8086-89 ASSM M	64853-90907	09/01/88	386
8086/8 ASSEMB	64853	09/01/88	52
		09/01/88	387
8086/8 C	64818	09/01/88	53
		09/01/88	396
8086/8 PASCAL	64814	09/01/88	59
		09/01/88	417
8086/88 C M	64818-90905	09/01/88	439
8086/88 PASCAL M	64814-90903	09/01/88	441
8086/88/186/188HLSAM	64332-90902	09/01/88	442
8088 DQ SW ANALYZER	64341C	09/01/88	443
8088 SW ANAL	64333	09/01/88	444
8088 DQ EMUL 300	64221S004	09/01/88	63
		09/01/88	445
8088 DQ SW ANAL	64333B	09/01/88	448
8088 EMULATION 300	64226S004	09/01/88	449
8096 ASSEMB	64860	09/01/88	64
		09/01/88	455
C COMPILER REF M	64800-90907	09/01/88	458
EBPP	64304	09/01/88	459
F9450 EMUL 300	64286S004	09/01/88	460
F9450 EMULATION	64286	09/01/88	461
GENERIC ANALYSIS M	64740-90909	09/01/88	462
GENERIC ANALYSIS FW	64740	09/01/88	463
GENERIC EMULATION FW	64700	09/01/88	467
HI SPD RS422 INTF	64037	09/01/88	65
HOST PASCAL	64817	09/01/88	476
HOST SOFTWARE / VAX	64882	09/01/88	66
		09/01/88	478
HOST SOFTWARE / 300	64883	09/01/88	67
		09/01/88	482

Product name	Product number	SSB ISSUE DATE	Page
USER DEF ASSEMB 300	64851S004	09/01/88	76
		09/01/88	560
USER DEF ASSEMB 500	64851S001	09/01/88	565
USER DEF ASSEMB DOS	64851S006	09/01/88	567
USER DEF ASSEMB VAX	64851S003	09/01/88	77
		09/01/88	570
USER DEF EMUL 300	64274S004	09/01/88	78
		09/01/88	572
USER DEF EMULATION	64274	09/01/88	79
USER DEFIN ASM M	64851-90904	09/01/88	578
USER INTERFACE M	64808-90901	09/01/88	580
USER INTERFACE 300	64808S004	09/01/88	80
		09/01/88	581
USER INTERFACE 500	64808S001	09/01/88	582
UTILITIES PKG 300	64888S003	09/01/88	583
UTILITIES PKG 300	64888S004	09/01/88	584
VMS 6500 ASSM M	64843-90904	09/01/88	585
VMS 6800 ASSM M	64841-90907	09/01/88	586
VMS 6800-03 C M	64821-90903	09/01/88	587
VMS 68000/08/10 C M	64819-90904	09/01/88	589
VMS 68000/8/10 ASM M	64845-90906	09/01/88	591
VMS 68000/8/10 P M	64815-90908	09/01/88	594
VMS 6805/9/9E ASM M	64844-90906	09/01/88	595
VMS 6809 PASCAL M	64813-90905	09/01/88	596
VMS 6809/09E C M	64822-90903	09/01/88	597
VMS 8051 ASSM M	64855-90904	09/01/88	601
VMS 8080/85 ASSM M	64840-90903	09/01/88	603
VMS 8085 C M	64826-90903	09/01/88	604
VMS 8085 PASCAL M	64825-90903	09/01/88	606
VMS 8086/88 C M	64818-90904	09/01/88	607
VMS 8086/88 PASCAL M	64814-90905	09/01/88	609
VMS 9900 ASSM M	64847-90905	09/01/88	610
VMS FILE FORMATS M	64882-90903	09/01/88	611
VMS M-STD1750A ASM M	64857-90903	09/01/88	612
VMS SYSTEM INSTAL M	64882-90904	09/01/88	613
VMS TMS 320 ASSM M	64858-90903	09/01/88	614
VMS USER DEFIN ASM M	64851-90907	09/01/88	615
VMS USERS GUIDE M	64882-90902	09/01/88	617
VMS Z80/NSC800 ASM M	64842-90905	09/01/88	618
VMS Z80/NSC800 C M	64824-90903	09/01/88	619
VMS Z80/NSC800 P M	64823-90903	09/01/88	621
VMS Z8001/02 C M	64820-90903	09/01/88	622
VMS Z8001/2 ASSM M	64854-90904	09/01/88	624
VMS Z8001/2 PASCAL M	64816-90905	09/01/88	625
VMS6800/01/02/03 P M	64811-90904	09/01/88	626
Z8 ASSEMB	64850	09/01/88	627
Z80 EMUL FW	64753	09/01/88	628
Z80 EMUL DOS	64753S006	09/01/88	629
Z80 ASSEMB	64842	09/01/88	81
		09/01/88	630
Z80 EMULATION 300	64252S004	09/01/88	632
Z80 PASCAL	64812	09/01/88	636
Z80 PASCAL M	64812-90903	09/01/88	637
Z80/NSC800 C	64824	09/01/88	82
		09/01/88	638

Product name		Product number	SSB ISSUE DATE	Page
Z80/NSC800 C	M	64824-90901	09/01/88	643
Z80/NSC800 P	M	64823-90901	09/01/88	645
Z80/NSC800PASCAL		64823	09/01/88	83
			09/01/88	646
Z8000 C		64820	09/01/88	84
			09/01/88	651
Z8000 PASCAL		64816	09/01/88	85
			09/01/88	661
Z8001 EMUL	300	64232S004	09/01/88	667
Z8001/02 C	M	64820-90901	09/01/88	671
Z8001/2 ASSEMB		64854	09/01/88	86
Z8001/2 EMUL	M	64980-90923	09/01/88	673
Z8002 EMUL	300	64233S004	09/01/88	674
Z80H EMULATION	300	64253S004	09/01/88	678

KPR number	page
1650004499	495
1650006544	651
1650006700	170
1650008128	507
1650016618	478
1650020396	264
1650028860	538
1650032698	488
1650033720	488
1650034066	379
1650034082	379
1650037556	369
1650038240	379
1650041624	636
1650042606	327
1650042630	327
1650042655	354
1650044016	325
1650047167	632
1650047340	632
1650048355	309
1650049163	646
1650051649	297
1650053660	369
1650054486	488
1650055251	199
1650055806	396
1650056838	268
1650057893	253
1650058123	638
1650058925	527
1650059162	489
1650061572	396
1650061580	544
1650063636	50
2700005280	325
2700005561	170
2700005769	527
2700005918	627
5000089359	554
5000089367	554
5000093708	304
5000094615	170
5000110098	417
5000117002	112
5000117507	524
5000119933	171
5000122374	549
5000123497	661
5000125229	193
5000128751	397
5000129023	360
5000131029	442
5000131573	673
5000132662	346
5000134619	455
5000134916	661
5000135285	398
5000135855	352
5000135905	387
5000136135	547
5000138941	417
5000139535	630
5000141150	442
5000141747	344
5000143628	264
5000146779	398
5000146829	418
5000149724	478
5000150151	662
5000150292	265
5000151050	111
5000151241	458
5000151290	478
5000152439	284
5000152819	630
5000152892	578
5000153981	578
5000160143	591
5000160671	651
5000161182	171
5000162651	344
5000163048	130
5000163295	621
5000163303	476
5000163808	197
5000164012	265
5000166983	112
5000167601	613
5000169037	191
5000169250	172
5000169698	484
5000169995	347
5000170118	71
5000170175	418
5000170191	645
5000171470	347
5000172221	387
5000173716	191
5000173815	130
5000174805	513
5000176065	131
5000176891	399
5000178301	581
5000180000	455
5000180323	479
5000181131	344
5000181545	652
5000182006	190
5000182246	513
5000182519	191
5000182824	514
5000183475	353
5000183913	173
5000184317	297
5000184374	154
5000187617	538
5000187674	164
5000188813	512
5000191361	399
5000191544	484
5000191767	455
5000192054	132
5000194951	542
5000195701	613
5000196428	174
5000198952	233
5000199059	223
5000203620	527
5000205054	544
5000206458	351
5000209007	528
5000209742	133
5000211359	510
5000211375	543
5000211557	325
5000211599	234
5000213983	234
5000214106	528
5000214148	196
5000214189	71
5000214452	234
5000214841	235
5000214858	400
5000215558	235
5000215913	387
5000216051	123
5000216267	123
5000219220	359
5000219865	95
5000220418	134
5000220764	591
5000220772	591
5000221200	111
5000222307	134
5000222489	617
5000223107	135
5000225078	455
5000225748	326
5000226530	135
5000226563	113
5000227991	388
5000228023	400
5000229237	136
5000229245	401
5000231076	524
5000231571	539
5000232744	419
5000232991	572
5000234849	236
5000235143	492
5000236828	137
5000236844	236
5000238337	441
5000238543	611
5000239012	193
5000239905	237
5000239921	479
5000239939	630
5000240259	342
5000240580	94
5000240929	347
5000240937	505
5000240952	538
5000242032	193
5000242818	319
5000243048	123
5000243907	268
5000244343	342
5000244368	175
5000244392	420
5000245704	638

KPR number	page	KPR number	page	KPR number	page	KPR number	page
5000246157	421	5000283184	489	D200031773	448	D200047126	523
5000246983	653	5000283630	521	D200031781	221	D200047217	480
5000247437	124	5000285742	223	D200031799	332	D200047423	178
5000247536	401	5000286591	40	D200031807	340	D200047613	291
5000247783	388	5000291427	580	D200031815	385	D200047779	425
5000250274	389	5000291914	198	D200031823	189	D200048199	115
5000250456	390	5000291930	403	D200031831	443	D200048421	561
5000251322	557	5000293779	95	D200031849	222	D200049833	348
5000251348	560	5000294181	561	D200031856	333	D200049882	178
5000251363	319	5000294199	404	D200031864	341	D200049908	405
5000252825	529	5000294207	31	D200031872	161	D200050997	178
5000252833	560	5000296855	79	D200032045	141	D200051276	291
5000254730	386	5000296921	80	D200033399	459	D200052316	368
5000255752	113	5000296947	53	D200033647	326	D200052563	179
5000256867	312	5000297754	53	D200035261	556	D200053132	611
5000258590	124	5000297879	16	D200035287	556	D200053157	179
5000258616	678	5000398396	49	D200035873	288	D200053165	180
5000259150	639	5000401349	72	D200036608	484	D200053298	116
5000259176	421	5000401372	72	D200036749	177	D200053512	562
5000259598	422	5000402214	54	D200036889	177	D200053785	505
5000260869	390	5000402701	77	D200036913	499	D200053793	601
5000263111	489	D200004929	140	D200036921	594	D200053801	506
5000264481	137	D200006080	423	D200037135	289	D200055202	613
5000264986	631	D200007237	269	D200037267	265	D200055541	562
5000266684	78	D200008342	404	D200037275	485	D200055558	269
5000267005	583	D200010280	424	D200038836	405	D200055590	643
5000267054	540	D200010363	637	D200040600	372	D200055657	439
5000267468	580	D200012104	175	D200040766	290	D200055665	510
5000269381	514	D200013110	169	D200041178	529	D200055673	607
5000269407	138	D200013334	372	D200041335	290	D200055699	154
5000269415	139	D200013359	175	D200042036	530	D200055707	497
5000269779	7	D200014332	176	D200043794	554	D200055715	589
5000270637	124	D200014357	476	D200043877	485	D200055731	671
5000271957	139	D200014399	141	D200045096	479	D200055749	519
5000272021	423	D200014944	424	D200045492	617	D200055756	622
5000273250	342	D200015123	326	D200045864	194	D200055772	106
5000273268	342	D200015297	529	D200045880	495	D200055780	493
5000273458	113	D200015305	476	D200045971	291	D200055798	587
5000273474	114	D200015636	288	D200046268	592	D200055814	286
5000273730	140	D200018747	125	D200046714	344	D200055822	501
5000275305	456	D200027516	425	D200046813	586	D200055830	597
5000275727	343	D200027995	404	D200046839	516	D200055855	517
5000278127	402	D200029702	288	D200046847	618	D200055863	619
5000278606	561	D200030585	177	D200046896	500	D200055889	366
5000279117	156	D200031088	115	D200046904	595	D200055897	507
5000280750	355	D200031740	384	D200047019	515	D200055905	604
5000280958	654	D200031757	378	D200047027	615	D200055947	194
5000283077	391	D200031765	444	D200047118	601	D200059030	291

KPR number	page	KPR number	page	KPR number	page	KPR number	page
D200080457	243	D200081414	381	D200082164	668	D200084913	357
D200080465	260	D200081422	451	D200082172	675	D200084921	382
D200080473	257	D200081430	249	D200082180	165	D200084939	337
D200080481	253	D200081448	260	D200082222	218	D200084947	452
D200080507	293	D200081455	254	D200082230	634	D200084954	337
D200080515	305	D200081489	633	D200082248	679	D200085019	426
D200080549	380	D200081497	277	D200082255	209	D200085035	393
D200080572	449	D200081505	150	D200082263	356	D200085043	534
D200080580	667	D200081513	412	D200082271	309	D200085076	482
D200080598	674	D200081521	659	D200082289	574	D200085084	463
D200080606	164	D200081539	100	D200082305	314	D200085092	463
D200080648	217	D200081547	278	D200082313	316	D200085241	251
D200080655	632	D200081554	640	D200082321	320	D200085258	467
D200080663	678	D200081562	363	D200082339	318	D200085274	467
D200080671	208	D200081570	349	D200082347	550	D200085282	665
D200080689	355	D200081596	90	D200082354	245	D200085290	628
D200080697	573	D200081646	564	D200082370	553	D200085308	393
D200080762	249	D200081679	165	D200082446	302	D200085316	345
D200080770	260	D200081794	126	D200082453	357	D200085332	634
D200080788	257	D200081836	495	D200082503	251	D200085357	117
D200080796	253	D200081844	214	D200082594	166	D200085365	468
D200080804	369	D200081851	306	D200082610	166	D200085373	150
D200080812	294	D200081869	667	D200082628	413	D200085381	660
D200080820	305	D200081877	675	D200082727	210	D200085399	152
D200080879	450	D200081885	165	D200082776	219	D200085423	199
D200080887	667	D200081893	218	D200083055	261	D200085431	200
D200080895	674	D200081901	633	D200083063	258	D200085449	201
D200080903	164	D200081919	679	D200083071	254	D200085456	247
D200080945	218	D200081927	209	D200083097	294	D200085472	641
D200080952	633	D200081935	356	D200083105	306	D200085530	468
D200080960	678	D200081943	574	D200083113	376	D200085563	468
D200080978	209	D200081968	245	D200083121	446	D200085571	117
D200080986	356	D200081976	250	D200083139	382	D200085597	468
D200080994	573	D200082008	250	D200083147	328	D200085605	463
D200081026	550	D200082016	261	D200083154	337	D200085613	463
D200081059	243	D200082024	257	D200083162	452	D200085621	463
D200081067	375	D200082032	254	D200083170	668	D200085639	469
D200081125	244	D200082057	87	D200083188	675	D200085647	469
D200081166	327	D200082065	2	D200083196	166	D200085662	117
D200081190	380	D200082073	4	D200083238	219	D200085696	118
D200081208	327	D200082081	294	D200083246	634	D200085712	427
D200081224	450	D200082099	306	D200083253	680	D200085720	428
D200081232	445	D200082107	375	D200083261	210	D200085738	414
D200081240	380	D200082115	445	D200083279	357	D200085746	428
D200081265	336	D200082123	381	D200083287	309	D200085753	429
D200081273	451	D200082131	328	D200083295	574	D200085779	101
D200081372	162	D200082149	336	D200083584	199	D200085787	102
D200081406	370	D200082156	451	D200084897	533	D200085795	251

KPR number	page	KPR number	page	KPR number	page	KPR number	page
D200085803	103	D200086413	310	D200087387	464	D200088310	158
D200085811	431	D200086421	246	D200087395	462	D200088328	206
D200085829	432	D200086439	351	D200087445	92	D200088336	215
D200085837	261	D200086512	471	D200087452	472	D200088344	681
D200085845	258	D200086520	471	D200087502	247	D200088351	211
D200085852	255	D200086579	433	D200087544	566	D200088369	358
D200085878	87	D200086595	471	D200087551	571	D200088377	310
D200085886	2	D200086603	278	D200087569	564	D200088385	314
D200085894	4	D200086611	280	D200087577	567	D200088393	316
D200085902	295	D200086629	281	D200087601	321	D200088401	320
D200085910	307	D200086637	464	D200087619	465	D200088419	318
D200085928	376	D200086645	321	D200087676	321	D200088427	246
D200085936	446	D200086652	471	D200087684	321	D200088435	393
D200085944	382	D200086660	471	D200087718	187	D200088443	669
D200085951	329	D200086678	127	D200087726	567	D200088450	677
D200085969	452	D200086686	631	D200087742	472	D200088492	567
D200085977	668	D200086694	534	D200087759	157	D200089276	223
D200085985	676	D200086702	565	D200087882	433	D200089342	489
D200085993	167	D200086710	570	D200087890	435	D200089409	559
D200086025	219	D200086728	535	D200087916	321	D200089417	202
D200086033	634	D200086736	566	D200087940	152	D200089433	559
D200086041	680	D200086744	571	D200087957	323	D200089458	559
D200086058	210	D200086801	232	D200087973	334	D200089490	536
D200086066	309	D200086868	472	D200088013	465	D200089516	203
D200086074	575	D200086876	118	D200088021	465	D200089524	248
D200086090	314	D200086884	118	D200088039	158	D200089631	119
D200086108	316	D200086900	93	D200088047	628	D200089649	121
D200086116	446	D200086918	201	D200088054	473	D200089714	224
D200086132	337	D200086926	202	D200088062	158	D200089722	225
D200086173	470	D200086934	252	D200088070	474	D200089730	225
D200086199	470	D200086942	415	D200088088	89	D200089748	226
D200086207	470	D200087015	119	D200088104	252	D200089763	226
D200086215	470	D200087023	464	D200088112	322	D200089771	227
D200086223	262	D200087031	472	D200088120	334	D200089789	228
D200086231	258	D200087049	535	D200088138	466	D200089805	121
D200086249	255	D200087114	321	D200088153	466	D200089813	323
D200086264	87	D200087148	482	D200088187	262	D200089839	330
D200086272	2	D200087189	108	D200088195	259	D200089847	324
D200086280	4	D200087197	157	D200088203	255	D200089854	330
D200086298	295	D200087213	206	D200088229	88	D200089888	228
D200086306	307	D200087270	575	D200088237	3	D200089896	543
D200086314	329	D200087288	220	D200088245	5	D200089920	453
D200086322	338	D200087304	109	D200088252	295	D200090118	69
D200086330	669	D200087312	303	D200088260	307	D200090134	121
D200086348	676	D200087320	186	D200088278	329	D200090167	323
D200086389	680	D200087338	665	D200088286	338	D200090175	641
D200086397	211	D200087346	648	D200088294	669	D200090183	568
D200086405	357	D200087353	361	D200088302	676	D200090191	568

- 6301V EMULATION -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64206S004	00.00	Processes sometimes left running after parent has stopped.	D200082057	87
	64206S004	00.00	Tracelist symbols dissappear.	D200085878	87
	64206S004	00.00	Using simio, then continuing , may not be possible	D200086264	87
	64206S004	00.00	"end" softkey after HP-IB error does not clear command line	D200088229	88
	64206S004	00.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090688	88

- 6301V/03R EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64206	01.01	6301V/03R module cannot be accessed with HP-UX 6.01	D200088088	89
	64206	01.01	Illegal opcode error occur when displaying memory repetetively	D200092122*	1

- 6301X EMULATION -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64207S004	00.00	Processes sometimes left running after parent has stopped.	D200082065*	2
	64207S004	00.00	Tracelist symbols dissappear.	D200085886*	2
	64207S004	00.00	Using simio, then continuing , may not be possible	D200086272*	2
	64207S004	00.00	"end" softkey after HP-IB error does not clear command line	D200088237*	3
	64207S004	00.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090696*	3

- 6301Y EMULATION -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64208S004	00.00	Processes sometimes left running after parent has stopped.	D200082073*	4
	64208S004	00.00	Tracelist symbols dissappear.	D200085894*	4
	64208S004	00.00	Using simio, then continuing , may not be possible	D200086280*	4
	64208S004	00.00	"end" softkey after HP-IB error does not clear command line	D200088245*	5
	64208S004	00.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090704*	5

- 6301Y/03Y EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64208	01.00	Emulator can't work when external clock is selected and E clock = 160khz	D200081596	90

- 64 HP-UX VMS 8096 AM - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64860-90901	01.00	.LIS file should be put in same directory as .A and .R files.	D200067322	91

- 64000 UX GENERIC - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64003S004	01.00	Option_test does not support set (environmental variable) command	D200087445	92

- 64000-UX OP-ENV -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64801S004	01.80	You can not enter ICC for IMB stimulus when in "option_test"	D200086900	93

- 64000-UX OP-ENV -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64801S004	01.80	EDB problems with scoping of locals from new com/asm/linker	D200090472	93
	64801S004	01.80	"makecdf(1)" error may occur in msinit on a new discless cluster	D200091280	93

- 64HP-UXVMS8086/8 A M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64853-90908	02.00	.LIS file should be put in same directory as .A and .R files.	D200067223	94
MANUAL	64853-90908	02.03	Need Manual change to explain AC and PH phase errors.	5000240580	94

- 650X ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
LINKER	64843	01.00	LNK does load NOLOAD files.	5000269779*	7
PROBLEM ON 9000/S300	64843	01.00	LNK does load NOLOAD files.	5000269779*	7
PROBLEM ON VAX	64843	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093328*	7

- 6800 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64821	02.10	Switch statement causes infinite loop.	5000293779	95
	64821	02.10	Posedecrement operator used on structure pointer may fail	D200085779	101
	64821	02.10	Type cast of constant to (char *) in pointer expression error	D200085787	102
	64821	02.10	SHORT_ARITH OFF for some short experssions used as conditional branch	D200085803	103
	64821	01.07	USE OF MANY FUNCTION CALLS WITH CONSTANT PARAMATERS MAY CAUSE ERR #1007	D200074989	98
	64821	01.07	Real variable used as a test condition cause error.	D200081539	100
	64821	01.06	Illegal initialization causes error 1113.	D200068197	97
NO PROBLEM/ PISCES I	64821	02.10	Compilers do not list complete information about source file path name.	D200092908*	8
PASS 3	64821	01.06	Conditional compile fails if it suceeds a fixed parm function call.	D200069823	97
PROBLEM ON 9000/S300	64821	02.10	Compilers do not list complete information about source file path name.	D200092908*	8
	64821	01.20	Libraries cause write to ROM	5000219865	95
	64821	01.07	If condition is tested with a CMP D1,D1	D200079624	99
PROBLEM ON 9000/S500	64821	02.10	Compilers do not list complete information about source file path name.	D200092908*	8
PROBLEM ON VAX	64821	02.10	Compilers do not list complete information about source file path name.	D200092908*	8

- 6800 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64821-90901	01.05	Declaring a function which returns a ptr to a function causes error.	D200055772	106

- 6800 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64811	02.00	Type casting the ADDR function to SET for masking may cause error	D200093534*	9
	64811	02.00	Large Sets may produce invalid results for elements outside set range	D200093542*	11
	64811	01.90	Bad code generated fot ADDR of first record element used as a parameter.	D200087189	108
	64811	01.90	Compare using var pointer to first record item fails.	D200093682*	12
	64811	01.90	Asignment of string to double dereference string pointer causes error	D200093708*	12
	64811	01.90	Pointer dereference of VAR pointer to structure as a parameter fails.	D200093716*	13
	64811	01.09	Compiler $FAR ON$, creates incorrect data offsets in listing	D200059980	108

- 6800 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
CODE GENERATOR	64811	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087304	109
NO PROBLEM/ PISCES I	64811	01.90	Compilers do not list complete information about source file path name.	D200092817*	9
NOT ON 64100 SYSTEM	64811	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087304	109
PROBLEM ON 9000/S300	64811	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087304	109
	64811	01.90	Compilers do not list complete information about source file path name.	D200092817*	9
PROBLEM ON 9000/S500	64811	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087304	109
	64811	01.90	Compilers do not list complete information about source file path name.	D200092817*	9
PROBLEM ON VAX	64811	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087304	109
	64811	01.90	Compilers do not list complete information about source file path name.	D200092817*	9
	64811	01.90	VAX Pascal xref prints many garbage characters of first line xref list	D200092940*	9

- 6800-03 ASSM - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64841-90905	01.15	Mask pseudo works incorrectly in certain cases.	5000151050	111
MANUAL	64841-90905	01.15	Support OIM, AIM, EIM, TIM	5000221200	111

- 6800/2 ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64841	01.15	External MASKS are not handled properly by the assembler.	5000166983	112
	64841	01.13	Comments are listed in the xref table when not delimited by a ;	5000117002	112
	64841	01.10	Assembler allows the inst. "LDA A". "LDA A" isn't a valid instruction.	5000273474	114
PROBLEM ON 9000/S300	64841	01.10	6301 AIM instruction with ".NT." operator causes LR error.	5000273458	113
PROBLEM ON 9000/S500	64841	01.40	Xref table is not listing all symbol references.	5000226563	113
	64841	00.00	Very long file causes problems with xref listing on a 2563B	5000255752	113
PROBLEM ON VAX	64841	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093302*	15

- 6800/2 ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64841S001	01.30	Macro def. including .IF, within a IF causes assembler to stop code gen.	D200053298	116
	64841S001	01.20	Assembler flagging out of range error when it should not.	D200031088	115
MACRO	64841S001	01.30	Conditional instr. .IF with rational oper. in Macro creates bad code	D200048199	115

- 68000 12MHZ EMUL -F SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64742	00.05	Measurements between the external/internal analyzers aren't synchronized	5000297879*	16
	64742	00.05	All states requested from emtrdata should be valid	D200091587	119
	64742	00.04	Slow Clock interferes with configuring monitor... Poor error messages.	D200089631	119
	64742	00.00	Stepping says "PC=123456@sp"; reg says "PC=12345678"	D200085357	117
	64742	00.00	Trace list mne heading doesn't indicate base (hex)	D200085571	117
	64742	00.00	Overlapping IAL output if 6800 cycle and GRD or ROM	D200085662	117
	64742	00.00	In RESET state, try to b(reak), end up running and unable to break	D200085696	118
	64742	00.00	Stepping in user space does not work with foreground monitor	D200086876	118
	64742	00.00	Emulator stays in monitor after the run command without giving a message	D200086884	118
	64742	00.00	Can do a "load/display target memory" with no target system	D200087015	119

- 68000 12MHZ EMUL DOS - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64742S006	01.01	Invalid COM port in 64700tab file	D200093435*	17

- 68000 12MHZ EMUL DOS - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu:ff	Description	KPR number	page
********none********	64742S006	01.00	Remove stack checking programa in production build code	D200091546	122
	64742S006	01.00	gregnumarray[0] not allocated in regdisp() or regmod ()	D200091561	122
	64742S006	00.01	Performance problem in the PC interface	D200089649	121
	64742S006	00.01	Invalid expressions can corrupt PC memory	D200089805	121
	64742S006	00.01	The "stty" command doesn't work correctly for baud rate <= 1200.	D200090134	121
	64742S006	00.00	OR'ing more than 4 label values in analyzer specification doesn't work	D200090605	121

- 68000 ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64845	01.13	Math operators not working on 64100.	5000258590	124
	64845	01.12	Incorrect code generated for Bit family of instructions when .L specif.	5000216267	123
	64845	01.12	Size qualifiers in cross reference.	5000247437	124
	64845	01.12	Assembler produces incorrect code for several instructions	D200079319	126
	64845	01.10	ILLEGAL OPCODE IS BEING GENERATED FOR LEA INSTRUCTION.	5000216051	123
	64845	01.10	RORG may cause generation of invalid errors and warnings.	D200064998	125
ASSEMBLER	64845	00.70	The legal range for data in the MOVEQ instruction is incorrect	D200018747	125
PROBLEM ON 9000/S300	64845	02.10	No A5 prompt when non-existient .R file specified.	5000270637	124
	64845	02.10	TITLE directive inserting garbage control characters.	D200086678	127
	64845	01.30	Bcc causes linker error if incorrect syntax is used.	D200081794	126
	64845	01.10	Missing whitespace is not flagged.	5000243048	123

- 68000 BBA - SSB ISSUE DATE: 09/01/88

********none********	64380S004	01.00	Complex conditional assignment delcarations cause bbacpp to core dump	D200090498	128
	64380S004	01.00	Switch statement followed immediately by a label cases bbacpp to fail	D200090506	128
	64380S004	01.00	A switch statement with no statement causes bbacpp to fail	D200090514	129

- 68000 C - SSB ISSUE DATE: 09/01/88

********none********	64819	02.10	Bad code is generated when a char var is compared to a negative number	5000269415	139
	64819	02.10	Arrays of 64k Multiples of size caus 210 error unnecessarily	D200087940	152
	64819	01.10	Terenary expression causing incorrect code to be generated.	5000176065	131
	64819	01.10	Logical operators '&&' and '||' causing bad code to be generated.	5000209742	133
	64819	01.10	Address is not incremented past 0xFFFF for data areas > 32k.	5000220418	134
	64819	01.10	float += float(unsigned -unsigned) hangs compiler.	5000223107	135
	64819	01.10	Real variable used as a test condition cause error.	5000226530	135
	64819	01.10	USE OF FUNCTION POINTER TYPE CAST ON MULTIPLE EXTERNAL ARRAYS CAN ERROR	D200074997	146
	64819	01.10	Terenary expression causing error 1113 or "Too many errors. "	D200076455	147
	64819	01.10	Address comparisons for variables located on negative base-page may fail	D200076513	148
	64819	01.10	Libraries generate incorrect code 68010 processor.	D200081505	150
	64819	01.10	SHORT_ARITH OFF use of mixed short int in conditionals may not work	D200085373	150
	64819	01.10	SHORT_ARITH OFF with unsigned short int in conditional branch error	D200085399	152
	64819	01.09	Illegal initialization causes error 1113.	5000173815	130
	64819	01.09	Fields of a structure are dereferenced incorrectly (if fields are big).	5000192054	132
	64819	01.09	Shift of wrong sized value in register.	D200063115	142
	64819	01.09	An "if" statement may cause the compiler to go astray.	D200065193	143

- 68000 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64819	01.09	A0 register not initialized properly when using ||.	D200067587	144
	64819	01.09	Libraries load constants into the data area	D200071829	145
	64819	01.08	Pass 3 error 1113 flagged.	5000163048	130
	64819	01.07	Compiler uses MSB of word containing char value rather than LSB.	D200032045	141
CODE GENERATOR	64819	01.07	Bad code using $OPTIMIZE$ and successive uses of the same pointer.	D200014399	141
	64819	00.21	Multiple assignments may cause compiler to reuse an overwritten reg.	D200004929	140
NO PROBLEM/ PISCES I	64819	02.10	Compilers do not list complete information about source file path name.	D200092882*	18
PASS 3	64819	01.09	Conditional compile fails if it suceeds a fixed parm function call.	D200069674	144
PREPROCESSOR	64819	01.10	Station reset during preprocessor pass.	5000273730	140
PROBLEM ON 9000/S300	64819	02.10	The EXT.L command does not work properly.	5000269407	138
	64819	02.10	Problem with EXT.L command.	5000271957	139
	64819	02.10	Compilers do not list complete information about source file path name.	D200092882*	18
	64819	01.20	Optimize directive causing bad code to be generated.	5000222307	134
	64819	01.20	Inconsistient error messages for too large of data area.	5000229237	136
	64819	01.20	Calling a function w/o assigning result causes stack to get messed up.	5000236828	137
	64819	01.20	Problem with Type Name cast - causes Pass 1 error.	5000264481	137
	64819	01.10	If condition is tested with a CMP D1,D1	D200079590	149
PROBLEM ON 9000/S500	64819	02.10	Compilers do not list complete information about source file path name.	D200092882*	18
PROBLEM ON VAX	64819	02.10	Compilers do not list complete information about source file path name.	D200092882*	18

- 68000 C - SSB ISSUE DATE: 09/01/88

********none********	64819-90902	01.09	List library link range in manuals.	5000184374	154
	64819-90902	01.08	Declaring a function which returns a ptr to a function causes error.	D200055699	154

- 68000 DQ EMUL - SSB ISSUE DATE: 09/01/88

********none********	64243S004	01.20	Mnemonic scroll and step display may not read memory	D200087197	157
	64243S004	01.20	Analyzer break fails when stepping over software breakpoint	D200087759	157
	64243S004	01.20	Multiple commands on the line after "set" will not be executed	D200088062	158
	64243S004	01.20	"end" softkey after HP-IB error does not clear command line	D200088310	158
	64243S004	01.20	Display memory w/part of the line not readable may hang the system	D200090357	158
	64243S004	01.20	bbaunload causes memory growth problems in emulators	D200090548	159
	64243S004	01.20	_display_message causes 'access guarded memory'	D200090555	159
	64243S004	01.20	Code disp. with trace not right if code changed w/o ending emul. session	D200090829	160
	64243S004	01.20	"tlist" can oferwrite memory it did not allocate, causing core dumps	D200092106*	19
	64243S004	01.10	Enabling DMA to emulation memory halts emulator.	5000279117	156
	64243S004	01.00	Questions not asked when switch from real-time to non-real.	D200078188	156
	64243S004	01.00	Wrong breakpoint behavior on continuing emulation	D200078196	156
	64243S004	01.00	EBPP as analyzer fails intermittently	D200078972	156
	64243S004	01.00	Doing a wait while tracing MAY cause subsuquent traces to never complete	D200088039	158

- 68000 DQ SW ANALYZER - SSB ISSUE DATE: 09/01/88

********none********	64341G	01.00	Using local static variables in C causes a lockup in the analyzer	D200031872	161

- 68000 EMUL 12.5 MHZ - SSB ISSUE DATE: 09/01/88

********none********	64243	01.01	State IA generates wrong instruction for Adr Reg. Indirect w/Indexing	D200081372	162

- 68000 EMULATION - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64242	01.07	Load of more than 1 abs. to targ. mem. not allowed when restricted to RT	D200067637	163

- 68000 EMULATION -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64242S004	01.00	Ehalt occurring too often while running user code.	5000187674	164
	64242S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069484	164
	64242S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080606	164
	64242S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080903	164
	64242S004	01.00	Tracing on status int_ack does not work.	D200081679	165
	64242S004	01.00	The Inter-Module-Bus Trigger signal latches when set to drive & receive	D200081885	165
	64242S004	01.00	Processes sometimes left running after parent has stopped.	D200082180	165
	64242S004	01.00	Memory breaks during stepping are not detected	D200082594	166
	64242S004	01.00	Loading a trace file from a different processor may cause core dump	D200083196	166
	64242S004	01.00	Tracelist symbols dissappear.	D200085993	167
	64242S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090811	167
BREAKPOINT	64242S004	01.00	Software breakpoint in target memory will hang system.	D200082610	166

- 68000 HL SOFT ANAL M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64331-90902	01.00	Tracing a variable declared as a pointer to a function doesn't work in C	D200013110	169

- 68000 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64815	02.00	Type casting the ADDR function to SET for masking may cause an error.	D200093450*	20
	64815	01.12	Bad code when taking ADDR of record element when using WITH.	5000161182	171
	64815	01.12	Subrange parameter not passed properly when function returning integer	D200076562	185
	64815	01.11	Declaring a boolean array may cause an out of bounds error.	5000169250	172
	64815	01.11	Casting address to int and adding a signed_16 var generates bad code.	5000183913	173
	64815	01.11	Bytes sign extened in a case statement.	5000196428	174
	64815	01.11	The WARN option cannot be turned off.	D200065045	183
	64815	01.11	Libraries load constants into the data area	D200071696	183
	64815	01.11	Problems with routine STRWRITE & $BASE_PAGE$ mode with ASPIOLIB	D200073007	184
	64815	01.10	Program causes compiler to hang up.	D200050997	178
	64815	01.10	Missing semicolon causes compiler to hang in Pass 1.	D200052563	179
	64815	01.10	Real number library routine "ROUND" not working in some cases.	D200053157	179
	64815	01.10	Library routine TRUNC will sometimes return wrong value.	D200053165	180
	64815	01.10	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060103	181
	64815	01.10	Compiler generates a LEA instruction with an illegal source operand.	D200060343	181
	64815	01.09	Compiler allows non-standard funct. with EXTENSIONS OFF or ANSI ON	5000119933	171
	64815	01.09	TOO MANY ERRORS IN PASS 3 IF >127 PROCEDURES	D200047423	178
	64815	01.05	Specific file causes pass three error when compiled. (Too many errors)	5000244368	175
	64815	00.00	Immediate operand's value is altered when doing a logical and.	1650006700	170
BOOLEAN	64815	01.09	NOT(function) as boolean expression in "IF" statement doesn't work.	D200030585	177
CODE GENERATOR	64815	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087320	186
	64815	01.09	Bad code using $RANGE$ or $DEBUG$ with $CALL_PC_LONG$ or LIB_PC_LONG	D200014332	176
	64815	01.08	$DEBUG$ may cause undesired TRAPV.	D200012104	175

- 68000 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
CODE GENERATOR	64815	01.08	Compiler generates incorrect code for set inclusion check.	D200013359	175
	64815	01.02	B := ABS(B) fails to write to the data area.	5000094615	170
DIV	64815	01.09	UNSIGNED_32 division with dividend or divisor > 8000,0000H may not work.	2700005561	170
INCLUDE	64815	01.09	Nested INCLUDE files 3 or more deep cause 64000 to "hang" in pass 3.	D200036749	177
NO PROBLEM/ PISCES I	64815	01.90	Compilers do not list complete information about source file path name.	D200092841*	20
NOT ON 64100 SYSTEM	64815	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087320	186
PASS 2	64815	01.09	K := K + K + K; causes too many pass 2 errors to continue.	D200036889	177
PASS 3	64815	01.10	Compiler $FAR ON$, creates incorrect data offsets in listing	D200049882	178
PROBLEM ON 9000/S300	64815	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087320	186
	64815	01.90	Array offset is incorrect if BOOLEAN is the data type for the indices.	D200087718	187
	64815	01.90	Compilers do not list complete information about source file path name.	D200092841*	20
PROBLEM ON 9000/S500	64815	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087320	186
	64815	01.90	Compilers do not list complete information about source file path name.	D200092841*	20
PROBLEM ON VAX	64815	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087320	186
	64815	01.90	Compilers do not list complete information about source file path name.	D200092841*	20
	64815	01.90	VAX Pascal xref prints many garbage characters of first line xref list	D200092973*	20

- 68000 SW ANALYZER - SSB ISSUE DATE: 09/01/88

********none********	64341B	02.00	Using local static variables in C causes a lockup in the analyzer	D200031823	189

- 68000/08 EMUL - SSB ISSUE DATE: 09/01/88

********none********	64242-90906	01.07	Monitor is not reentrant. Please document consequences and limitations.	5000182006	190

- 68000/08 EMULHP-UX M - SSB ISSUE DATE: 09/01/88

********none********	64243-90903	01.00	Explain useage/definition in manual of INTR character to terminate wait	5000169037	191
	64243-90903	01.00	Page 4-9 Fig 4-3 The first ORG statement should be UNCOMMENTED	5000173716	191
	64243-90903	01.00	Single Stepping slow when in "display trace source only mode".	5000182519	191

- 68000/08/10 ASM - SSB ISSUE DATE: 09/01/88

********none********	64845-90904	01.09	RORG ONLY WORKS IF THE CODE IS NOT PC INDEPENDENT.	5000125229	193
	64845-90904	01.08	Incorrect opcode is generated for Move from CCR instruction.	D200055947	194
	64845-90904	01.00	Wrong offset calculated when using PC+index reg+ offset mode of addr.	D200045864	194
	64845-90904	01.00	Include support for BHS and BLO.	D200065565	195
	64845-90904	01.00	Clear up confusion on correct symbols.	D200065607	195
MANUAL	64845-90904	01.10	Alter all assembler manuals to reflect new syntax.	5000239012	193
	64845-90904	01.10	Manual indicates EXT is a legal psuedo for an external declaration.	5000242032	193
	64845-90904	01.00	The MOVE example for program counter with index address mode is wrong.	D200062521	195

- 68000/10 EMUL - SSB ISSUE DATE: 09/01/88

********none********	64245-90903	01.00	Manual has error on fig 3-1, should be "cp" and "sample.*".	5000214148	196

- 68000/10 RT S-ANAL M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64341-90903	02.00	Non-adjacent symbols not traceable in some conditions.	5000163808	197

- 68000C AXLS COMP - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64902-90901	01.00	Fig. 4.1 does not compile as the manual indicates.	5000291914	198
MANUAL	64902-90901	01.00	Cport68k documentation has incorrect path name for executable	D200092148*	22
	64902-90901	01.00	Recursive includes not allowed with lister.	D200092155*	22
	64902-90901	01.00	lister does not allow recursive includes	D200092171*	22

- 68000C AXLS COMP 300 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64902S004	01.00	Option for inverse C code only works if it is specified first.	1650055251	199
	64902S004	01.00	When cpp macros expansion includes new-line, debug line numbers get off.	D200083584	199
	64902S004	01.00	All variables following an ORG are put at same address.	D200085423	199
	64902S004	01.00	Operations on bitfields > 17 bits wide may fail.	D200085431	200
	64902S004	01.00	Compiler won't take address of a function pointer.	D200085449	201
	64902S004	01.00	Compiler erroneously constant folds pointer offsets	D200086918	201
	64902S004	01.00	Function entry code over writes stack when widened param is not passed.	D200086926	202
	64902S004	01.00	#progma BBA_IGNORE still produces warrnings	D200089417	202
	64902S004	01.00	Runtime and support libraries contain loadtime initializers.	D200089516	203
	64902S004	01.00	Cpp looks in the wrong directory for local include files.	D200093013*	23
	64902S004	01.00	Cpp requires white space after #define macro name.	D200093039*	23
CODE GENERATOR	64902S004	01.00	'Cannot open file' message coming from lister, if open many includes.	D200092163*	23

- 68008 EMULATION -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64244S004	01.20	Emulator does not work reliably with 64155b Memory	D200087213	206
	64244S004	01.20	"end" softkey after HP-IB error does not clear command line	D200088328	206
	64244S004	01.20	Code disp. with trace not right if code changed w/o ending emul. session	D200090837	206
	64244S004	01.20	Multiple commands on the line after "set" will not be executed	D200091116	207
	64244S004	01.20	Enabling DMA to emulation memory halts emulator.	D200091132	207
	64244S004	01.20	"bba unload" causes memory growth problems in emmulators	D200092353*	25
	64244S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069518	204
	64244S004	01.00	Incorrect breakpoint behaviour on continuing emulation.	D200072470	204
	64244S004	01.00	Questions not asked when switch from real-time to non-real.	D200078170	204
	64244S004	01.00	EBPP as analyzer fails intermittently	D200078964	206
	64244S004	01.00	Doing a wait while tracing MAY cause subsuquent traces to never complete	D200092346*	25

- 6801/3 EMULATION 300 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64256S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069567	208
	64256S004	01.00	State inverse assembler for 6801 does not work	D200077545	208
	64256S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080671	208
	64256S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080978	209
	64256S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081927	209

Keyword	Product number	uu.ff	Description	KPR number	page
			- 6801/3 EMULATION 300 - SSB ISSUE DATE: 09/01/88		
********none********	64256S004	01.00	Processes sometimes left running after parent has stopped.	D200082255	209
	64256S004	01.00	Under certain conditions the 6801 may not work correctly with SPA	D200082727	210
	64256S004	01.00	Loading a trace file from a different processor may cause core dump	D200083261	210
	64256S004	01.00	Tracelist symbols dissappear.	D200086058	210
	64256S004	01.00	Using simio, then continuing , may not be possible	D200086397	211
	64256S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088351	211
	64256S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090886	211
			- 68010 16MHZ EMUL DOS - SSB ISSUE DATE: 09/01/88		
********none********	64745S006	01.01	Invalid COM port in 64700tab file	D200093443*	26
			- 68010 EMUL 12.5M 300 - SSB ISSUE DATE: 09/01/88		
********none********	64245S004	01.20	"end" softkey after HP-IB error does not clear command line	D200088336	215
	64245S004	01.20	Code disp. with trace not right if code changed w/o ending emul. session	D200090845	215
	64245S004	01.20	Multiple commands on the line after "set" will not be executed	D200091124	216
	64245S004	01.20	Enabling DMA to emulation memory halts emulator.	D200091140	216
	64245S004	01.20	"bba unload" causes memory growth problems in emmulators.	D200092361*	27
	64245S004	01.10	Function Code not correct for user mem. access during display registers	D200081844	214
	64245S004	01.00	simulated I/O, Ints can't be modified when changing from rtrt to nrtrt	D200068874	213
	64245S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069526	213
	64245S004	01.00	Incorrect breakpoint behaviour on continuing emulation.	D200072488	213
	64245S004	01.00	Ehalt occurring too often while running user code.	D200078055	214
	64245S004	01.00	SPA emulator title exceeds 9 character limit and will be truncated	D200078220	214
	64245S004	01.00	EBPP as analyzer fails intermittently	D200078956	214
	64245S004	01.00	Doing a wait while tracing MAY cause subsuquent traces to never complete	D200092338*	27
			- 68010 G.P. EMUL -3 SSB ISSUE DATE: 09/01/88		
********none********	64249S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069534	217
	64249S004	01.00	Incorrect breakpoint behaviour on continuing emulation.	D200072496	217
	64249S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080648	217
	64249S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080945	218
	64249S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081893	218
	64249S004	01.00	Processes sometimes left running after parent has stopped.	D200082222	218
	64249S004	01.00	Memory breaks during stepping are not detected	D200082776	219
	64249S004	01.00	Loading a trace file from a different processor may cause core dump	D200083238	219
	64249S004	01.00	Tracelist symbols dissappear.	D200086025	219
	64249S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090852	220
BREAKPOINT	64249S004	01.00	Software breakpoint in target memory will hang system.	D200087288	220
			- 68010 SW ANAL - SSB ISSUE DATE: 09/01/88		
********none********	64334	02.00	Using local static variables in C causes a lockup in the analyzer	D200031781	221

Keyword	Product number	uu.ff	Description	KPR number	page
			- 68010 SW ANALYZER - SSB ISSUE DATE: 09/01/88		
********none********	64341D	02.00	Using local static variables in C causes a lockup in the analyzer	D200031849	222
			- 68020 ASSEMB - SSB ISSUE DATE: 09/01/88		
********none********	64870S004	01.00	Packed BCD constants cause incorrect code to be generated.	5000199059	223
	64870S004	01.00	Using asm psued END with numeric expression causes linker error.	D200089276	223
	64870S004	01.00	Ar68k can not handle long list in command line options.	D200089714	224
	64870S004	01.00	PLEN directive does not work properly	D200089722	225
	64870S004	01.00	LLEN directive does not work properly with tab characters	D200089730	225
	64870S004	01.00	Temporary files should be created in /tmp directory	D200089748	226
	64870S004	01.00	Incremental link and strip results in corrupted relocatable	D200089763	226
	64870S004	01.00	Reference to label in empty section causes ld68k error	D200089771	227
	64870S004	01.00	Section mismatch causes bad info in HP link_sym file	D200089789	228
	64870S004	01.00	Embedded assembly code will not substitute defined variables correctly.	D200089888	228
MACROS	64870S004	01.00	>37 parameters in a MACRO heading and it silently does not expand.	D200092312*	28
PROBLEM ON 9000/S300	64870S004	01.00	NOPAGE option does not work for the 68000 assembler.	5000285742	223
			- 68020 BBA - SSB ISSUE DATE: 09/01/88		
********none********	64381S004	01.10	Complex conditional assignment delcarations cause bbacpp to core dump	D200090449	230
	64381S004	01.10	Switch statement followed immediately by a label cases bbacpp to fail	D200090456	230
	64381S004	01.10	A switch statement with no statement causes bbacpp to fail	D200090464	231
			- 68020 EMUL - SSB ISSUE DATE: 09/01/88		
********none********	64410S004	02.00	"at_execution run" may fail to run upon execution,	D200086801	232
	64410S004	02.00	Leading comma in some addtess indirect assembly is not needed	D200091306	232
	64416S004	02.00	"end" softkey after HP-IB error does not clear command line	D200088427	246
	64416S004	01.10	Failure occurs when executing a software breakpoint in user state	5000236844	236
	64416S004	01.10	OR instruction is not disassembled properly	5000239905	237
	64416S004	01.00	FMOVE instructions are disassembled incorrectly.	5000198952	233
	64416S004	01.00	Error may result when executing "run from a until b" for target address.	5000211599	234
	64416S004	01.00	Enhancement to the dissassemble feature of the trace display.	5000213983	234
	64416S004	01.00	Failures when running the 68020 and any other emulator in same cardcage.	5000214452	234
	64416S004	01.00	Failure occurs with "modify memory <address> to <procedure name>".	5000214841	235
	64416S004	01.00	Improve comments in the emulation monitor regarding the "TRACE" vector.	5000215558	235
	64416S004	01.00	Sporadic HP64120 I/O failures.	5000234849	236
	64416S004	01.00	Memory Mapper deletes the map_overlay definition when address bit 31=1.	D200077024	237
	64416S004	01.00	Unable to access transfer address of .X file if .L file does not exist.	D200078071	238
	64416S004	01.00	Use of disassemble instructions_only may show no data with new trace.	D200078105	239
	64416S004	01.00	Modify memory map attributes does not release mapped memory.	D200078113	239
	64416S004	01.00	A DISPLAY MEMORY MNEUMONIC DISASSEMBLY ERROR.	D200078907	239
	64416S004	01.00	Overlapped memory entries may not be resolved properly.	D200078915	240
	64416S004	01.00	Modify memory using symbolic data incorrectly loads "value -1".	D200078931	240
	64416S004	01.00	Can not map_overlay memory if range includes 0ffffff00h-0ffffffffh.	D200078949	241
	64416S004	01.00	68020 PV incorrectly defaults co-cards for 2nd emulator in a cage.	D200079061	241

- 68020 EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64416S004	01.00	Load/copy/store using files over net may fail with 68020 emulator.	D200079707	241
	64416S004	01.00	Some interaction problems with "at_execution run/trace".	D200080408	242
	64416S004	01.00	Analyzer GLOBAL_CONTEXT specification core dumps when running HP-UX 5.5.	D200080416	242
	64416S004	01.00	Trace data not updated if trace command follows at_execution run.	D200080424	242
	64416S004	01.00	Some command file sequences may not work.	D200080440	243
	64416S004	01.00	Turning off BNC usage does not update the trace display	D200080457	243
	64416S004	01.00	PV for coordinated emulation start may miss failure.	D200081059	243
	64416S004	01.00	MON_XFR_BUF area in monitor not properly defined.	D200081125	244
	64416S004	01.00	Tracelist symbols disappear from the trace display in certain conditions	D200081968	245
	64416S004	01.00	Processes sometimes left running after parent has stopped.	D200082354	245
	64416S004	01.00	Using simio, then continuing , may not be possible	D200086421	246

- 68020C AXLS COMP 300 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64903S004	01.10	#include files are not searched in order of -I options	D200087502	247
	64903S004	01.10	Runtime and support libraries contain loadtime initializers.	D200089524	248
	64903S004	01.10	Cpp looks in the wrong directory for local include files.	D200093021*	29
	64903S004	01.10	Cpp requires white space after #define macro name.	D200093047*	30
	64903S004	01.00	Compiler won't take address of a function pointer.	D200085456	247
INCLUDE	64903S004	01.10	Nested #INCLUDE's cause too many files to be open.	D200092189*	29

- 6805 -U SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64192S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080150	249
	64192S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080762	249
	64192S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081430	249
	64192S004	01.00	Tracelist symbols dissappear.	D200081976	250
	64192S004	01.00	Processes sometimes left running after parent has stopped.	D200082008	250
	64192S004	01.00	Loading a trace file from a different processor may cause core dump	D200082503	251
	64192S004	01.00	Load <file> noupdate without database should give error message.	D200085241	251
	64192S004	01.00	Using simio, then continuing , may not be possible	D200085795	251
	64192S004	01.00	Core dump can happen when displaying guarded memory mnemonic	D200086934	252
	64192S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088104	252
	64192S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090639	252

- 6805 E - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64195S004	01.00	Can't load a program into target memory or emul. mem. with slow clock.	1650057893	253
	64195S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080481	253
	64195S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080796	253
	64195S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081455	254
	64195S004	01.00	Processes sometimes left running after parent has stopped.	D200082032	254
	64195S004	01.00	Loading a trace file from a different processor may cause core dump	D200083071	254
	64195S004	01.00	Tracelist symbols dissappear.	D200085852	255
	64195S004	01.00	Using simio, then continuing , may not be possible	D200086249	255
	64195S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088203	255

- 6805 E - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64195S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090662	256

- 6805 G - SSB ISSUE DATE: 09/01/88

********none********	64194S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080473	257
	64194S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080788	257
	64194S004	01.00	Processes sometimes left running after parent has stopped.	D200082024	257
	64194S004	01.00	Loading a trace file from a different processor may cause core dump	D200083063	258
	64194S004	01.00	Tracelist symbols dissappear.	D200085845	258
	64194S004	01.00	Using simio, then continuing , may not be possible	D200086231	258
	64194S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088195	259
	64194S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090654	259

- 6805 P - SSB ISSUE DATE: 09/01/88

********none********	64193S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080465	260
	64193S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080770	260
	64193S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081448	260
	64193S004	01.00	Processes sometimes left running after parent has stopped.	D200082016	261
	64193S004	01.00	Loading a trace file from a different processor may cause core dump	D200083055	261
	64193S004	01.00	Tracelist symbols dissappear.	D200085837	261
	64193S004	01.00	Using simio, then continuing , may not be possible	D200086223	262
	64193S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088187	262
	64193S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090647	262

- 6805/9 ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64844	01.11	LR error flagged for legal expression of the form 'label-value'.	1650020396	264
	64844	01.11	HEX pseudo causes byte counter to quit incrementing in certain cases.	5000150292	265
	64844	01.11	Arithmetic expression is not being evaluated correctly.	5000164012	265
	64844	01.11	NT operator not operating consistiently.	D200063164	265
	64844	01.11	BEXT address is not calculated correclty.	D200076950	266
	64844	01.10	Label in IF stmnt. does not appear in XREF	5000143628	264
	64844	00.15	No error generated when overflow occurs.	D200037267	265
CODE GENERATOR	64844	01.40	BRSET range not checked.	5000294207*	31
PROBLEM ON 9000/S300	64844	01.40	BRSET range not checked.	5000294207*	31
PROBLEM ON VAX	64844	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093336*	31

- 6809 C - SSB ISSUE DATE: 09/01/88

********none********	64822	01.80	Use of "+=" accessing first element of structure using pointer error	D200086603	278
	64822	01.80	Compare error using address of local variable on right of expression	D200086611	280
	64822	01.80	SHORT_ARITH OFF expressions in branches may not work as K&R	D200086629	281
	64822	01.80	Switch statement using unsigned int values 0 and 0xFFFF creates error	D200093575*	32
	64822	01.08	Some C programs using pointer & structure dereferences cause error #1006	D200075036	273

- 6809 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64822	01.08	Programs with duplicate goto labels may fail in Pass 3 on VAX&HPUX C	D200075663	274
	64822	01.08	Real variable used as a test condition cause error.	D200081547	278
	64822	01.07	Illegal initialization causes error 1113.	D200068239	270
	64822	01.07	Conditional compile fails if it suceeds a fixed parm function call.	D200069864	271
	64822	01.07	Use of address (&) stack vars on right side of conditional expression	D200073171	271
NO PROBLEM/ PISCES I	64822	01.80	Compilers do not list complete information about source file path name.	D200092916*	32
PASS 1	64822	00.00	COMP_SYM file not purged when COMP_SYM option not selected.	D200007237	269
PASS 3	64822	01.06	Illegal opcode generated when assigning value to a char. array pointer.	D200055558	269
PROBLEM ON 9000/S300	64822	01.80	There is a problem with incrementing pointer structures using '++'.	1650056838	268
	64822	01.80	Compilers do not list complete information about source file path name.	D200092916*	32
	64822	01.20	Incorrect code is generaaated for while statment.	D200081497	277
	64822	01.08	If condition is tested with a CMP D1,D1	D200079632	276
PROBLEM ON 9000/S500	64822	01.80	Compilers do not list complete information about source file path name.	D200092916*	32
	64822	01.40	Compiler generating bad code which may cause run-time crash.	5000243907	268
PROBLEM ON VAX	64822	01.80	Compilers do not list complete information about source file path name.	D200092916*	32
			- 6809 C - SSB ISSUE DATE: 09/01/88		
********none********	64822-90901	01.06	Clarification of interface for USER_DEFINED and real number routines.	5000152439	284
	64822-90901	01.06	Declaring a function which returns a ptr to a function causes error.	D200055814	286
			- 6809 C - SSB ISSUE DATE: 09/01/88		
********none********	64822S001	01.20	++ and -- operators evaluated with improper precedence.	D200051276	291
	64822S001	01.20	Host compilers do not put absolute pats specifications in relocatables	D200059030	291
	64822S001	01.00	File fails to compile. Error 1113 is generated.	D200029702	288
	64822S001	00.00	Problem with integer pointer in conditional statement.	D200041335	290
	64822S001	00.00	Title description is incorrect.	D200045971	291
	64822S001	00.00	TOO MANY ERRORS IN PASS 3 IF >127 PROCEDURES	D200047613	291
CODE GENERATOR	64822S001	00.00	16 bit comparison on a 8 bit unsigned short field.	D200035873	288
PASS 1	64822S001	01.00	Incorrect code is generated when complementing a parm. in a return stmt.	D200015636	288
PASS 3	64822S001	00.00	Compiler option $LIST_OBJ ON$ generates wrong output information.	D200037135	289
	64822S001	00.00	Pass 3 fails to detect relative jump address out-of-range.	D200040766	290
			- 6809 EMULATION - SSB ISSUE DATE: 09/01/88		
********none********	64215S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069401	293
	64215S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200070557	293
	64215S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080507	293
	64215S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080812	294
	64215S004	01.00	Processes sometimes left running after parent has stopped.	D200082081	294
	64215S004	01.00	Loading a trace file from a different processor may cause core dump	D200083097	294
	64215S004	01.00	Tracelist symbols dissappear.	D200085902	295
	64215S004	01.00	Using simio, then continuing , may not be possible	D200086298	295
	64215S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088252	295
	64215S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090712	296

- 6809 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64813	01.70	Type casting the ADDR function to SET for masking may cause an error.	D200093468*	34
	64813	01.70	Large Sets may produce invalid results for elements outside set range	D200093526*	35
	64813	01.30	If >39 functions declared; following funcs may include bad code.	1650051649	297
	64813	01.11	With statements used in FOR loops on records may cause error #1006	D200075010	301
	64813	01.11	Compiler incorrectly assumes the value of a var is in the D register.	D200082446	302
	64813	01.10	Records of pointers to text not handled correctly.	5000184317	297
	64813	01.10	ADDR function for stack relative variables in right side conditionals	D200073155	299
CODE GENERATOR	64813	01.60	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087312	303
NO PROBLEM/ PISCES I	64813	01.60	Compilers do not list complete information about source file path name.	D200092825*	34
NOT ON 64100 SYSTEM	64813	01.60	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087312	303
PASS 3	64813	01.09	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060020	298
PROBLEM ON 9000/S300	64813	01.60	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087312	303
	64813	01.60	Compilers do not list complete information about source file path name.	D200092825*	34
PROBLEM ON 9000/S500	64813	01.60	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087312	303
	64813	01.60	Compilers do not list complete information about source file path name.	D200092825*	34
PROBLEM ON VAX	64813	01.60	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087312	303
	64813	01.60	Compilers do not list complete information about source file path name.	D200092825*	34
	64813	01.60	VAX Pascal xref prints many garbage characters of first line xref list	D200092957*	34

- 6809 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64813-90903	00.02	Parameter passing thru the registers has changed.	5000093708	304

- 6809E EMULATION -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64216S004	01.00	Measurement System end released when terminal cannot be initialized	D200069419	305
	64216S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080515	305
	64216S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080820	305
	64216S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081851	306
	64216S004	01.00	Processes sometimes left running after parent has stopped.	D200082099	306
	64216S004	01.00	Loading a trace file from a different processor may cause core dump	D200083105	306
	64216S004	01.00	Tracelist symbols dissappear.	D200085910	307
	64216S004	01.00	Using simio, then continuing , may not be possible	D200086306	307
	64216S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088260	307
	64216S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090720	308

- 68HC11 EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64265S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088377	310
	64265S004	01.10	Code disp. with trace not right if code changed w/o ending emul. session	D200090902	310
	64265S004	01.00	68HC11 will work alone as a measurement system.	1650048355	309
	64265S004	01.00	Processes sometimes left running after parent has stopped.	D200082271	309
	64265S004	01.00	Loading a trace file from a different processor may cause core dump	D200083287	309
	64265S004	01.00	Tracelist symbols dissappear.	D200086066	309
	64265S004	01.00	Using simio, then continuing , may not be possible	D200086413	310

- 68HCII ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64865	01.00	Incorrect object code generated for BSET external_sym,mask	5000256867	312

- 68HCII ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64865	01.00	Illegal and incorrect object code for STAA, STD operators.	D200072397	312
	64865	01.00	BAD CODE GENERATED FOR "JSR" INSTRUCTION.	D200091835	313

- 70016 EMUL (JLO) 300 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64294S004	01.10	Processes sometimes left running after parent has stopped.	D200082305	314
	64294S004	01.10	Tracelist symbols dissappear.	D200086090	314
	64294S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088385	314

- 70108 EMUL (JLO) 300 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64295S004	01.10	Processes sometimes left running after parent has stopped.	D200082313	316
	64295S004	01.10	Tracelist symbols dissappear.	D200086108	316
	64295S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088393	316

- 70208 EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64297S004	01.00	Processes sometimes left running after parent has stopped.	D200082339	318
	64297S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088419	318

- 70216 EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64296	01.00	V50 Disassembler generates "illegal" opcode for "POP PS" instruction	5000242818	319
	64296	01.00	Can not specify needed trigger specification.	5000251363	319
	64296S004	01.00	Processes sometimes left running after parent has stopped.	D200082321	320
	64296S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088401	320

- 80186 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64764	00.01	Guarded memory might not cause gaurded-mem break	D200087114	321
	64764	00.01	Invalid "cf mon" setting in firmware gives PC intfc. problems	D200087676	321
	64764	00.01	Incorrect report of bp when breakpoint feature is disabled	D200087916	321
	64764	00.01	Bad Background monitor	D200088112	322
	64764	00.01	Regnumarray[0] is not being allocated	D200092445*	38
	64764	00.00	Each "init" command eats memory - crash after 20 "init"'s	D200086645	321
	64764	00.00	Help message for the "loc config" item is incomplete	D200087601	321
	64764	00.00	Invalid "cf mon" setting in firmware gives PC intfc. problems	D200087684	321

- 80186 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64764S006	01.01	Invalid COM port in 64700tab file	D200093419*	39
	64764S006	00.01	Modify memory with an invalid string 1..100 could fail	D200087957	323
	64764S006	00.01	Invalid expressions can corrupt PC memory	D200089813	323

- 80186 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64764S006	00.01	The "stty" command doesn't work correctly for baud rate <= 1200.	D200090167	323

- 80186 EMUL FW - SSB ISSUE DATE: 09/01/88

********none********	64764-90901	01.00	The Manual says that step is not allowed in real time mode.	D200089847	324

- 80186 EMULATION - SSB ISSUE DATE: 09/01/88

********none********	64224	01.05	LODS instructions with segment override not properly disassembled.	5000225748	326
	64224	01.04	"disp. memory mnemonic" shows incorrect inv. assembly for JMP NEAR inst	5000211557	325
	64224	01.03	PROBLEMS APPEAR WHEN LISTING MEMORY INCLUDING ADDRESSES OFFFEH & OFFFFH.	D200015123	326
	64224	00.00	"run from <addr>", "modify reg <reloc>" generates 16 extra I/O writes.	1650044016	325
DISASSEMBLER	64224	01.03	Disassembler displays segment override though it is not coded.	D200033647	326
EBPP	64224	00.00	80186 Emulator with EBPP and State calls the wrong disassembler.	2700005280	325

- 80186 EMULATION -3 SSB ISSUE DATE: 09/01/88

********none********	64224S004	01.10	"modify memory" command results in an "end release".	1650042606	327
	64224S004	01.10	"trace only status INTACK" always displays interrupt type 0.	1650042630	327
	64224S004	01.10	80186 DISFUNCTION WHEN MONITOR NOT LOADED ON 64000-UX	5000286591*	40
	64224S004	01.10	Processes sometimes left running after parent has stopped.	D200082131	328
	64224S004	01.10	Loading a trace file from a different processor may cause core dump	D200083147	328
	64224S004	01.10	Tracelist symbols dissappear.	D200085951	329
	64224S004	01.10	Using simio, then continuing , may not be possible	D200086314	329
	64224S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088278	329
	64224S004	01.10	Support baunload feature in the emulator.	D200089839	330
	64224S004	01.10	SOURCE LINES are missing from "absolute trace display" with "SOURCE ON"	D200089854	330
	64224S004	01.10	Code disp. with trace not right if code changed w/o ending emul. session	D200090761	330
	64224S004	01.00	Loading/modifing configuration after continue may cause reset.	D200081166	327
	64224S004	01.00	Modify/Store memory abort at physical addr 0 for seg/offset procs	D200081208	327

- 80186 SW ANAL - SSB ISSUE DATE: 09/01/88

********none********	64335	02.00	Using local static variables in C causes a lockup in the analyzer	D200031799	332

- 80186 SW ANALYZER - SSB ISSUE DATE: 09/01/88

********none********	64341E	02.00	Using local static variables in C causes a lockup in the analyzer	D200031856	333

- 80188 - SSB ISSUE DATE: 09/01/88

********none********	64765	00.01	Incorrect report of bp when breakpoint feature is disabled	D200087973	334
	64765	00.01	Bad Background monitor	D200088120	334

- 80188 - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64765S006	01.00	Invalid COM port in 64700tab file	D200093427*	41

- 80188 EMULATION - SSB ISSUE DATE: 09/01/88

USER MEMORY	64225	01.03	Emulator would not recover from errors during display memory repetitive.	D200065805	335

- 80188 EMULATION -3 SSB ISSUE DATE: 09/01/88

********none********	64225S004	01.10	Processes sometimes left running after parent has stopped.	D200082149	336
	64225S004	01.10	Loading a trace file from a different processor may cause core dump	D200083154	337
	64225S004	01.10	"modify memory" command results in an "end release".	D200084939	337
	64225S004	01.10	"trace only status INTACK" always displays interrupt type 0.	D200084954	337
	64225S004	01.10	Software Breakpoints don't work in target memory.	D200086132	337
	64225S004	01.10	Using simio, then continuing , may not be possible	D200086322	338
	64225S004	01.10	"end" softkey after HP-IB error does not clear command line	D200088286	338
	64225S004	01.10	Code disp. with trace not right if code changed w/o ending emul. session	D200090779	338
	64225S004	01.10	NO warning message if parts of the monitor are in target memory	D200093252*	42
	64225S004	01.00	Display memory line crossing segment boundary will be wrong	D200081265	336

- 80188 SW ANAL - SSB ISSUE DATE: 09/01/88

********none********	64336	02.01	Using local static variables in C causes a lockup in the analyzer	D200031807	340

- 80188 SW ANALYZER - SSB ISSUE DATE: 09/01/88

********none********	64341F	01.00	Using local static variables in C causes a lockup in the analyzer	D200031864	341

- 80286 EMULATION - SSB ISSUE DATE: 09/01/88

********none********	64228	01.02	trace only <Odd Address> data 0: analyzer doesn't qualify properly.	5000240259	342
	64228	01.02	80286 emul. fails to run programs mapped as user memory at the target.	5000244343	342
	64228	01.02	80286 Emulator may not display proper Interrupt Type number.	5000273250	342
	64228	01.02	trace abt addr 0:0E0H status rd mem triggers on addresses 0E0h, 0C0H.	5000273268	342
	64228	01.02	"list printer memory" command gives wrong addresses using seg:offset.	5000275727	343
	64228	01.02	First PV cycle shows failure with some 64155B cards, if PV'd 1st on 228.	D200080127	343

- 80286 UDE - SSB ISSUE DATE: 09/01/88

DISASSEMBLER	64227	01.00	Incorrect data is returned on a trace about an I/O port.	5000181131	344
DISPLAY MEMORY	64227	01.00	Inverse assembler does not work properly during display memory mnemonic.	5000141747	344
	64227	01.00	The IDIV instruction is not correct during a display memory mnemonic.	5000162651	344
INSTRUCT. EXECUTION	64227	01.00	Single step function does not work after a software breakpoint.	D200046714	344

- 80286B ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64859	01.40	MODULE pseudo generates random relocation type	D200092734*	43

- 80286B ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64859	01.02	Aliases not allowed in the linker to specify library paths.	D200068775	345
	64859	01.02	Address in 8086 family assemblers lost segment information.	D200085316	345
PROBLEM ON VAX	64859	01.40	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093732*	43

- 8048 ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64846	01.00	Error message LR generated on valid JMP instruction	5000132662	346
PROBLEM ON VAX	64846	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093351*	44

- 8051 ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64855	01.08	Assembler inconsistant in permitting forward referencing	5000169995	347
	64855	01.08	Defining a transfer address causes an ET error	5000171470	347
	64855	01.08	HIGH operator does not function correctly	D200068379	349
	64855	01.08	CONT in linker will overwrite addresses of variables in different module	D200091710	349
	64855	01.08	Cross reference goes into endless loop on macro reference.	D200092098*	45
	64855	00.00	Link maps produced on VAX are different than on 64000 and are wrong.	D200049833	348
CODE GENERATOR	64855	01.20	Special operator "HIGH" does not work with DS pesudo opcode	5000240929	347
	64855	01.08	HIGH does not work	D200081570	349
PROBLEM ON 9000/S300	64855	01.20	Special operator "HIGH" does not work with DS pesudo opcode	5000240929	347
PROBLEM ON VAX	64855	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093385*	45

- 8051 ASSM - SSB ISSUE DATE: 09/01/88

CODE GENERATOR	64855-90902	01.05	In the manula pg 8-2 states the BIT instruc. shows operand is address.	5000206458	351
MANUAL	64855-90902	01.07	The assmblr manual needs to be updated w/ information in reference manul	D200086439	351

- 8051 ASSM + AL REF M - SSB ISSUE DATE: 09/01/88

********none********	64855-90905	01.05	The $ operand does not work as defined.	5000135855	352

- 8051 EMUL - SSB ISSUE DATE: 09/01/88

********none********	64264-90901	01.01	Manual enhancement to reflect Port display info in more detail.	5000183475	353

- 8051 EMULATION - SSB ISSUE DATE: 09/01/88

********none********	64264	00.00	Cannot load absolute file using remote file access.	1650042655	354

- 8051 EMULATION - SSB ISSUE DATE: 09/01/88

********none********	64264S004	01.00	"Core dumps" when displaying user memory<odd address> blocked word.	5000280750	355

- 8051 EMULATION - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64264S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069575	355
	64264S004	01.00	Msg "Monitor must reside in emul pgm mem" is flaky	D200075788	355
	64264S004	01.00	Monitor is not recognized when overwritten,re-entered after "end-lock".	D200077438	355
	64264S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080689	355
	64264S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080986	356
	64264S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081935	356
	64264S004	01.00	Processes sometimes left running after parent has stopped.	D200082263	356
	64264S004	01.00	Emulator end_releases when displaying int. data mem. repet. at odd addr.	D200082453	357
	64264S004	01.00	Loading a trace file from a different processor may cause core dump	D200083279	357
	64264S004	01.00	"modify memory" command results in an "end release".	D200084913	357
	64264S004	01.00	Using simio, then continuing , may not be possible	D200086405	357
	64264S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088369	358
	64264S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090894	358

- 8080/5 ASSEMB - SSB ISSUE DATE: 09/01/88

CODE GENERATOR	64840	01.00	xref incorrect with conditional assmbly IF when code generated for false	5000219220	359
PROBLEM ON VAX	64840	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093278*	46

- 8085 B PASCAL - SSB ISSUE DATE: 09/01/88

********none********	64825	01.90	Type casting the ADDR function to SET causes error #1006 on the VAX	D200093641*	47
	64825	01.02	Incorrect data offsets in listing file.	D200060228	360
CODE GENERATOR	64825	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087353	361
NOT ON 64100 SYSTEM	64825	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087353	361
PASS 1	64825	01.01	$Range ON$ causes incorrect code to be generated for a test operation.	5000129023	360
PROBLEM ON 9000/S300	64825	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087353	361
PROBLEM ON 9000/S500	64825	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087353	361
PROBLEM ON VAX	64825	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087353	361
	64825	01.90	VAX Pascal xref prints many garbage characters of first line xref list	D200093005*	47

- 8085 C - SSB ISSUE DATE: 09/01/88

********none********	64826	02.10	Indirect func calls fail following calls with temporary results	D200090258	364
	64826	01.04	Real variable used as a test condition cause error.	D200081562	363
NO PROBLEM/ PISCES I	64826	02.10	Compilers do not list complete information about source file path name.	D200092932*	48
PASS 3	64826	01.03	Conditional compile fails if it suceeds a fixed parm function call.	D200069948	363
PROBLEM ON 9000/S300	64826	02.10	Compilers do not list complete information about source file path name.	D200092932*	48
PROBLEM ON 9000/S500	64826	02.10	Compilers do not list complete information about source file path name.	D200092932*	48
PROBLEM ON VAX	64826	02.10	Compilers do not list complete information about source file path name.	D200092932*	48

- 8085 C - SSB ISSUE DATE: 09/01/88

********none********	64826-90901	01.02	Declaring a function which returns a ptr to a function causes error.	D200055889	366

- 8085 C - SSB ISSUE DATE: 09/01/88

CODE GENERATOR	64826S004	00.00	Incorrect opcode "MOV A,ACC" allowed by our assembler	D200052316	368

```
                                           - 8085 EMULATION -        SSB ISSUE DATE:  09/01/88

Keyword              Product number   uu.ff Description                                                   KPR number page

********none******** 64203            01.07 64203A (8085) MEMORY MAPPING PROBLEMS                         5000398396*  49
                     64203            01.00 Config memory map is corrupted if examined during modify config D200093104*  49
                                           - 8085 EMULATION -        SSB ISSUE DATE:  09/01/88


********none******** 64203S004        01.30 Code disp. with trace not right if code changed w/o ending emul. session D200090670  370
                     64203S004        01.30 Display target mem shows incorrect data when crossing 256 byte boundary. D200093112*  50
                     64203S004        01.10 Display user memory causes a time-out, requiring end-release to recover. 1650037556  369
                     64203S004        01.10 Using Emulation across RFA can give incomplete symbol information       D200080804  369
                     64203S004        01.03 64000-UX 8085 EMULATION SW - BLOCK BOUNDARY PROBLEM                     1650063636*  50
                     64203S004        01.00 I/O Failure message when "run from <> until <>"; HPUX 6.0.              1650053660  369
                     64203S004        01.00 Absolute code part user,part emul, will be overwritten at boundary.     D200080226  369
                     64203S004        01.00 Relative path names (e.g. ./cmd) should not search PATH                 D200081406  370
                                           - 8085 PASCAL -           SSB ISSUE DATE:  09/01/88


********none******** 64810            00.70 Run time UNDERFLOW error using ZDSBSUB library if result has even parity D200040600  372
CODE GENERATOR       64810            00.70 Compiler generates incorrect code for BOOLEAN assignment statement.      D200013334  372
                                           - 8086 DQ EMUL -          SSB ISSUE DATE:  09/01/88


********none******** 64220S004        01.20 Processes sometimes left running after parent has stopped.              D200082107  375
                     64220S004        01.20 Loading a trace file from a different processor may cause core dump     D200083113  376
                     64220S004        01.20 Tracelist symbols dissappear.                                           D200085928  376
                     64220S004        01.20 Code disp. with trace not right if code changed w/o ending emul. session D200090738  376
                     64220S004        01.20 NO warning message if parts of the monitor are in target memory         D200093237*  51
                     64220S004        01.10 Measurement System end_released when terminal cannot be initialized     D200069427  374
                     64220S004        01.10 Display memory line crossing segment boundary will be wrong             D200080432  374
                     64220S004        01.10 Loading/modifing configuration after continue may cause reset.          D200081067  375
DISASSEMBLER         64220S004        01.10 8087 FSUB & FDIV instructs are not disassembled properly.               D200075838  374
                                           - 8086 DQ SW ANAL -       SSB ISSUE DATE:  09/01/88


********none******** 64332B           01.00 Using local static variables in C causes a lockup in the analyzer       D200031757  378
                                           - 8086 EMUL -             SSB ISSUE DATE:  09/01/88


********none******** 64222S004        01.00 Software Breakpoints lock up terminal, when mapped to target.           1650034066  379
                     64222S004        01.00 Paging at a segment end produces a confusing CS:IP.                     1650034082  379
                     64222S004        01.00 Using Emulation across RFA can give incomplete symbol information       1650038240  379
                     64222S004        01.00 Measurement System end_released when terminal cannot be initialized     D200069443  379
                     64222S004        01.00 pwd truncates the /net7system portion of the path when RFA'ed to system. D200080549  380
                     64222S004        01.00 Modify/Store memory abort at physical addr 0 for seg/offset procs       D200081190  380
                     64222S004        01.00 Display memory line crossing segment boundary will be wrong             D200081240  380
```

- 8086 EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64222S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081414	381
	64222S004	01.00	Processes sometimes left running after parent has stopped.	D200082123	381
	64222S004	01.00	Loading a trace file from a different processor may cause core dump	D200083139	382
	64222S004	01.00	"modify memory" command results in an "end release".	D200084921	382
	64222S004	01.00	Tracelist symbols dissappear.	D200085944	382
	64222S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090753	383

- 8086 SW ANAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64332	02.00	Using local static variables in C causes a lockup in the analyzer	D200031740	384

- 8086 SW ANALYZER - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64341A	01.00	Using local static variables in C causes a lockup in the analyzer	D200031815	385

- 8086-89 ASSM - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
MANUAL	64853-90907	02.01	8086 Asm/linker manual doesn't doc. valid DQ and DT directives.	5000254730	386

- 8086/8 ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64853	02.70	OUTM does not allow override prefix.	D200090340	394
	64853	02.70	JMP immediate instructions do not work on the 8086 Assembler	D200092114*	52
	64853	02.70	MODULE pseudo generates random relocation type	D200092726*	52
	64853	02.03	Labels used in the operand field of a DBS instr causes ET error	5000172221	387
	64853	02.03	Address in 8086 family assemblers lost segment information.	D200085308	393
	64853	02.00	Assembler does not allow [SI] as operand for OUTS	5000135905	387
CODE GENERATOR	64853	02.30	Assembler does not handle all string comparisons correctly.	5000247783	388
	64853	02.30	Cannot use DS for a var that is EQU'd to another var that used "SET".	5000250274	389
	64853	02.30	External EQU'ed veriables may not be resolved properly.	5000250456	390
	64853	02.03	Incorrect code generated when EQU offset used in MOV REG,REG/MEM	5000215913	387
	64853	02.03	intra segment indirect calls	D200079582	392
	64853	02.03	LXI E,addr and LXI C,addr are not flagged as errors in 8080 mode	D200085035	393
	64853	02.02	Using arithmetic to calc address of mem loc of destin. of MOV causes err	5000227991	388
	64853	00.00	Using 'WORD PTR' to a EQU'd constant can result in bad code	5000260869	390
LINKER	64853	02.03	will not link if segment address not equal 0	D200079566	391
PROBLEM ON 9000/S300	64853	02.70	CMP statement is producing wrong label address.	5000283077	391
	64853	02.70	INSTALLATION PROBLEM	D200088435	393
	64853	02.70	Linker locks for no apparent reason.	D200091918	394
	64853	02.30	Assembler does not handle all string comparisons correctly.	5000247783	388
	64853	02.30	Cannot use DS for a var that is EQU'd to another var that used "SET".	5000250274	389
	64853	02.30	External EQU'ed veriables may not be resolved properly.	5000250456	390
	64853	00.00	Using 'WORD PTR' to a EQU'd constant can result in bad code	5000260869	390

- 8086/8 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64818	03.70	Constant divided by short in function call generates wrong code.	1650061572	396

- 8086/8 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64818	03.70	Bad code generated on 64000 with "80286" directive.	5000278127	402
	64818	03.70	Use of structure causing fatal 1006 error.	5000291930	403
	64818	03.70	IF statement loads wrong segment for compare statement.	D200090332	415
	64818	03.70	ES reg used instead of SS when assign string to structure	D200093054*	56
	64818	03.07	ES register used but never defined.	1650055806	396
	64818	03.02	Fields of a structure are dereferenced incorrectly (if fields are big).	D200076695	409
	64818	03.02	Real variable used as a test condition cause error.	D200081513	412
	64818	03.01	Conditional expressions with unsigned mixed operands may fail	D200063057	406
	64818	03.01	Illegal initialization causes error 1113.	D200068080	407
	64818	03.01	Libraries load constants into the data area	D200071787	409
	64818	03.00	1006 message generated when referenced to unspecified array element	5000135285	398
	64818	03.00	Illegal instruction generated by ASM_FILE	D200049908	405
	64818	02.00	One's complement operator ~ causes incorrect code when used in if stmt.	D200027995	404
	64818	02.00	Compiler using unacceptable amount of stack space for procedure returns.	D200038836	405
CODE GENERATOR	64818	03.70	PASS 2 error when pntr type used to invoke code stored in array.	D200085738	414
	64818	03.70	ADDR routine causes "Access to guarded mem" msg, due to prob w/ FINDMARK	D200086942	415
	64818	03.20	Err 1006 generated if passing address of array into array of pointers.	5000247536	401
	64818	03.02	Bad code generated when casting a real constant into an integer	5000176891	399
	64818	03.02	When $POINTER_SIZE 32$ generates 32 bit arithmetic for 16 bit variables	5000191361	399
	64818	03.02	Error 1113 generated in PASS 3, when using "case 0xffff" in switch stmt	5000228023	400
	64818	03.02	Compiler generates MOV SP, BP and LEAVE. This is redundant.	5000229245	401
	64818	03.02	Casting ptr. to int as short & incrementing it generates bad code	D200068700	407
	64818	03.02	~, &, | and ^ may not correctly expand shorts in condidionals	D200079343	410
	64818	03.02	Case stment nested in With stment w/ variant records generates bad code	D200082628	413
	64818	03.01	Bad code generated when left shift short variable & AND w/ unsigned int	5000214858	400
	64818	03.00	Divide operation byte divisor & quotient > a byte which causes 0 Int.	5000146779	398
	64818	02.00	Assignment operator used with mult. arrays of double/float type - bad cd	5000128751	397
	64818	00.56	Vars ORGed in seg. 0 in SHORT env. access current DS seg with no warning	D200008342	404
NO PROBLEM/ PISCES I	64818	03.70	Compilers do not list complete information about source file path name.	D200092874*	56
PASS 3	64818	03.01	Conditional compile fails if it suceeds a fixed parm function call.	D200069716	408
PROBLEM ON 9000/S300	64818	03.70	& address operator generates PUSH DS1 when DS1 not defined.	5000294199	404
	64818	03.70	Wrong code generated for structure in while loop.	500402214*	54
	64818	03.70	Long arithmetic expression generates incorrect code.	D200092080*	55
	64818	03.70	Compilers do not list complete information about source file path name.	D200092874*	56
	64818	03.20	Unsigned Short with bit field aligned on word boundary.	5000296947*	53
	64818	01.10	If condition is tested with a CMP D1,D1	D200079608	411
PROBLEM ON 9000/S500	64818	03.70	Call to function using LONGS uses wrong segment.	5000297754*	53
	64818	03.70	Compilers do not list complete information about source file path name.	D200092874*	56
PROBLEM ON VAX	64818	03.70	Compilers do not list complete information about source file path name.	D200092874*	56

- 8086/8 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64814	03.50	Boolean Index into array generates bad code	D200093476*	59
	64814	03.50	Test for set inclusion checks beyond the set boundary.	D200093484*	60
	64814	03.02	CASE statement produces bad code for complicated expression	5000272021	423
	64814	03.02	Nested IFs inside a WITH may generate incorrect code.	D200027516	425
	64814	03.00	Out of expression storage error generated on code that ran on old ver.	5000138941	417
	64814	03.00	Libraries load constants into the data area	5000146829	418

- 8086/8 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64814	02.01	80186Generates wrong offset within CONST_data area	D200047779	425
	64814	01.90	Error 1006 for complex statement using MOD operator	D200093518*	61
CODE GENERATOR	64814	03.50	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	5000246157	421
	64814	03.50	Err 1006 generated in compex equation w/ in 2 FOR stmts in an IF statmnt	5000259598	422
	64814	03.50	Compiler produces bad code for accessing parameters in nested procedures	D200085720	428
	64814	03.50	Bad code created when assgn ext real valu to real variable in a procedur	D200085746	428
	64814	03.50	Addr Function for ORG'ed integer generating bad code	D200085753	429
	64814	03.50	IMUL instruction will overwrite a value in DX register.	D200085811	431
	64814	03.50	Bad cd gen if proc declared EXT in another proc is called w/ FAR PROC ON	D200085829	432
	64814	03.50	Procedure Environ_init is missing from the simlib.R library	D200086579	433
	64814	03.50	WITH statement generates bad code.	D200087882	433
	64814	03.50	WITH statement generates bad code.	D200087890	435
	64814	03.20	MOD operative in complex equation generates bade code.	5000259176	421
	64814	03.02	Array reference overrides DX register	5000232744	419
	64814	03.02	ERROR 117 generated, but does not indicate variable in error	5000244392	420
	64814	03.02	Complex data structure produces bad code	D200085019	426
	64814	03.01	DIV of array of signed_16 by signed_16 in FOR loop produces bad code	5000170175	418
	64814	02.01	WITH statement generating bad code	D200085712	427
	64814	01.10	Incorrect code generated when CASE stmnt. uses an arrayed record field.	5000110098	417
	64814	00.60	Byte values may be converted to 16-bit before comparison with byte var.	D200010280	424
	64814	00.46	Data structures larger than 64K are not flagged as an error.	D200006080	423
NO PROBLEM/ PISCES I	64814	03.50	Compilers do not list complete information about source file path name.	D200092833*	59
NOT ON 64100 SYSTEM	64814	03.50	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	5000246157	421
PASS 3	64814	03.00	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060061	426
PROBLEM ON 9000/S300	64814	03.50	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	5000246157	421
	64814	03.50	Assignment of constant into array of 3 elements does not work.	D200090597	437
	64814	03.50	Compilers do not list complete information about source file path name.	D200092833*	59
PROBLEM ON 9000/S500	64814	03.50	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	5000246157	421
	64814	03.50	Compilers do not list complete information about source file path name.	D200092833*	59
PROBLEM ON VAX	64814	03.50	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	5000246157	421
	64814	03.50	Compilers do not list complete information about source file path name.	D200092833*	59
	64814	03.50	VAX Pascal xref prints many garbage characters of first line xref list	D200092965*	59
RUN-TIME LIBRARY	64814	01.10	Failed to detect out-of-bounds case.	D200014944	424

- 8086/88 C - SSB ISSUE DATE: 09/01/88

********none********	64818-90905	03.00	Declaring a function which returns a ptr to a function causes error.	D200055657	439

- 8086/88 PASCAL - SSB ISSUE DATE: 09/01/88

MANUAL	64814-90903	03.00	Change manual to say that libraries need to be in same segment	5000238337	441

- 8086/88/186/188HLSAM - SSB ISSUE DATE: 09/01/88

********none********	64332-90902	02.00	Display variable may result in "ERROR:E64".	5000131029	442
	64332-90902	02.00	Data structures too large to display in "display variable" command.	5000141150	442

- 8088 -D SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64341C	01.00	Using local static variables in C causes a lockup in the analyzer	D200031831	443

- 8088 -S SSB ISSUE DATE: 09/01/88

********none********	64333	02.00	Using local static variables in C causes a lockup in the analyzer	D200031765	444

- 8088 DQ EMUL - SSB ISSUE DATE: 09/01/88

********none********	64221S004	01.10	Display memory line crossing segment boundary will be wrong	D200081232	445
	64221S004	01.10	Processes sometimes left running after parent has stopped.	D200082115	445
	64221S004	01.10	Loading a trace file from a different processor may cause core dump	D200083121	446
	64221S004	01.10	Software Breakpoints don't work in target memory.	D200086116	446
	64221S004	01.10	Code disp. with trace not right if code changed w/o ending emul. session	D200090746	447
	64221S004	01.10	NO warning message if parts of the monitor are in target memory	D200093245*	63
	64221S004	01.00	Tracelist symbols dissappear.	D200085936	446

- 8088 DQ SW ANAL - SSB ISSUE DATE: 09/01/88

********none********	64333B	01.00	Using local static variables in C causes a lockup in the analyzer	D200031773	448

- 8088 EMULATION - SSB ISSUE DATE: 09/01/88

********none********	64226S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069476	449
	64226S004	01.00	Absolute code part user,part emul, will be overwritten at boundary.	D200080218	449
	64226S004	01.00	Apparent error during disassembly of the offset at an intrasegment jump.	D200080275	449
	64226S004	01.00	Paging at a segment end produces a confusing CS:IP.	D200080325	449
	64226S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080572	449
	64226S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080879	450
	64226S004	01.00	Modify/Store memory abort at physical addr 0 for seg/offset procs	D200081224	450
	64226S004	01.00	Display memory line crossing segment boundary will be wrong	D200081273	451
	64226S004	01.00	Relative path names (e.g. ./cmd) should not search PATH	D200081422	451
	64226S004	01.00	Processes sometimes left running after parent has stopped.	D200082156	451
	64226S004	01.00	Loading a trace file from a different processor may cause core dump	D200083162	452
	64226S004	01.00	"modify memory" command results in an "end release".	D200084947	452
	64226S004	01.00	Tracelist symbols dissappear.	D200085969	452
	64226S004	01.00	Emulator does not work reliably with 64155B memory controller	D200089920	453
	64226S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090787	453

- 8096 ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64860	01.04	Pseudo instruction DCB treats absolute variable as relocatable.	5000275305	456
	64860	01.03	Rom emulator does not display local/global symbols correctly w/ 8096 cod	5000180000	455
	64860	01.00	display GLOBAL/LOCAL does not work when using the ROM emulator	5000134619	455
CODE GENERATOR	64860	01.03	Linker does not allocate the file at even addresses	5000191767	455

- 8096 ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
CODE GENERATOR	64860	01.03	Using ORG statemnts can generate ERR_LR errors	5000225078	455
PROBLEM ON VAX	64860	01.70	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093740*	64
			- C COMPILER REF - SSB ISSUE DATE: 09/01/88		
MANUAL	64800-90907	00.07	Add note to compiler supplements regarding 3000 symbol limit.	5000151241	458
			- EBPP - SSB ISSUE DATE: 09/01/88		
********none********	64304	01.03	Configuration file not automatically loaded when using EBPP	D200033399	459
			- F9450 EMUL - SSB ISSUE DATE: 09/01/88		
********none********	64286S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090928	460
			- F9450 EMULATION - SSB ISSUE DATE: 09/01/88		
********none********	64286	01.04	RS232 Simulated IO will overrun the user's read buffer sometimes.	D200075150	461
			- GENERIC ANALYSIS - SSB ISSUE DATE: 09/01/88		
********none********	64740-90909	01.00	Errors in xtt help screen.	D200087395	462
			- GENERIC ANALYSIS -F SSB ISSUE DATE: 09/01/88		
********none********	64740	00.02	"ts" after "init -c" shows incorrect "trigger in memory"	D200087619	465
	64740	00.02	Changing the trace configuration causes error with the fast clock speed	D200088013	465
	64740	00.02	Arm to trigger time can be incorrect if: clock is set to the fast mode	D200088021	465
	64740	00.02	HELP xteq scrolls off screen	D200088138	466
	64740	00.02	short help for trc says telif is "seq glb restart"	D200088153	466
	64740	00.02	Incorrect absolute time count when trigger is not found	D200090290	466
	64740	00.00	Analyzer ROM PV fails about 1 time every 6 hours.	D200085084	463
	64740	00.00	"tck" Command does not give errors for invalid options.	D200085092	463
	64740	00.00	Help tsq in easy mode still showes -t option	D200085605	463
	64740	00.00	The use of the xttq command can cause the storage of incorrect trans	D200085613	463
	64740	00.00	CMB-exec trace started message should be flagged ASYNC-STAT	D200085621	463
	64740	00.00	When "tg arm and addr=4" command then "Label not defined:any"	D200086637	464
	64740	00.00	Arm to trigger time count is off by 120ns	D200087023	464
	64740	00.00	"xtarm always" generates an error message	D200087387	464
			- GENERIC EMULATION FW - SSB ISSUE DATE: 09/01/88		
********none********	64700	00.05	Odd byte format records may cause an extra byte written to memory	D200091264	474

- GENERIC EMULATION FW - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64700	00.04	Some problems with the "step" command	D200087742	472
	64700	00.04	Break, Breakpoint, Mem Access cause confusion if they occur simultaneous	D200088054	473
	64700	00.04	Error message priority of break messages needs to be changed.	D200088070	474
	64700	00.04	Restricted load fails when file loads to guarded memory	D200090308	474
	64700	00.03	"Cntl C" after a power-on can crash the emulator	D200086520	471
	64700	00.00	Baud rate setting of 38400 changes to 19200	D200085258	467
	64700	00.00	The command "bc -d bp" can result in strange behaviour.	D200085274	467
	64700	00.00	Main Help screen has emul listed twice.	D200085365	468
	64700	00.00	Ending value of data stream does not report proper error.	D200085530	468
	64700	00.00	Failure to run from R state after rx rst, x,x,rst,r	D200085563	468
	64700	00.00	Can't ser up CMB to run-from-power-up	D200085597	468
	64700	00.00	Too many Commands in command line causes command truncation	D200085639	469
	64700	00.00	Improper coverage calculation of overlapping ranges	D200085647	469
	64700	00.00	Incorrect documentation in help screen for grammer	D200086173	470
	64700	00.00	stty command changes on 1200 and 300 will not return a prompt.	D200086199	470
	64700	00.00	Extended hex format symbol records cause download problems	D200086207	470
	64700	00.00	"Stepping aborted" status message may or may not appear.	D200086215	470
	64700	00.00	The help screen says load-load emulation memory	D200086512	471
	64700	00.00	"ser" gives incorrect pattern match address for TMS32020	D200086595	471
	64700	00.00	Command "map -d 0" hangs the system	D200086652	471
	64700	00.00	A cntl-C break after "init" may not inititialize properly	D200086660	471
	64700	00.00	The "run" and "step" commands do not check for ambigous addr obj	D200086868	472
	64700	00.00	Reading into garded memory can't return less than 16 bytes	D200087031	472
	64700	00.00	Incorrect info on "help io", can't display IO in long words	D200087452	472

- HI SPD RS422 INTF - SSB ISSUE DATE: 09/01/88

********none********	64037	00.01	The product does not generate a proper XON/XOFF, in "handshake" mode.	D200093765*	65

- HOST PASCAL - SSB ISSUE DATE: 09/01/88

********none********	64817	01.04	IOERROR not generated.	5000163303	476
	64817	01.04	Spurious run-time error doing WRITE(REAL_VAL) after previous I/O error	D200014357	476
	64817	01.04	STRWRITE function may produce run time error in specific case.	D200015305	476

- HOST SOFTWARE -/ SSB ISSUE DATE: 09/01/88

********none********	64882	02.40	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093591*	66
	64882	02.40	Break or ^C may not abort a foreground transfer with a file list	D200093625*	66
	64882	01.60	Transfer fails when downloading relocatable libraries	1650016618	478
	64882	01.60	RCMAIN corrupts RCDEVICE.dat file when aborted with Cntl C or Y	5000151290	478
	64882	01.20	Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.	D200045096	479
	64882	01.20	LONG COMMANDS GREATER THAN 1024 CHAR. MALFUNCTION WITH DMF-32 I/O CARD	D200047217	480
HIGH SPEED LINK	64882	01.60	HSL transfer from within RCMAIN does not return control to RCMAIN.	5000149724	478
RCMAIN	64882	01.70	RCDEVICE.DAT is not properly maintained.	5000180323	479
	64882	01.70	/DEVICES= does not work with a list of stations.	D200064055	480

- HOST SOFTWARE -/ SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
RCMAIN	64882	01.60	HSL transfer from within RCMAIN does not return control to RCMAIN.	5000149724	478
	64882	01.60	Vax rcdevice file not updated correctly	D200059428	480
	64882	01.60	VAX remote control dumps when a very long command is entered	D200059444	480
TRANSFER	64882	02.00	Transfer of files over DECnet causes program to crash	5000239921	479

- HOST SOFTWARE / -3 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64883	01.10	Transfer may abort on >32K files.	D200087148	482
	64883	01.10	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093609*	67
	64883	01.10	Break or ^C may not abort a foreground transfer with a file list	D200093633*	67
	64883	01.00	Cluster - Cluster Transfer does not work with filelist	D200085076	482
TRANSFER	64883	01.00	Transfer does not handle extra line-feeds in file.	D200079483	482
	64883	01.00	Incorrect syntax/usage may not result in warning or error message.	D200079681	482

- HOST SOFTWARE / -5 SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64880	01.90	Cluster to cluster transfers have a strange err.msg if >47 files in list	D200093583*	68
	64880	01.90	Break or ^C may not abort a foreground transfer with a file list	D200093617*	68
	64880	01.20	Transfer to blank userid does not translate file names correctly.	D200036608	484
	64880	01.20	xx.L TO xx:link_sym translation wrong for 0 length records (types 3 & 4)	D200037275	485
RCMAIN	64880	01.20	A session command is req'd before entering the menu in batch jobs.	D200043877	485
TRANSFER	64880	01.60	Transfer may not move library files.	5000191544	484
	64880	01.06	Transfer does not correctly parse "FILE:USERID:@HSL".	5000169698	484
TRANSLATE	64880	01.50	C.K.1 and C.K.2 both translate to C_K on the 64000.	D200062539	485

- HP 64020A UPGRADE -M SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64020-90902	00.00	Retrofit kit does not include fans.	D200065938	486

- HP 64120A CARDCAGE M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64120-90902	00.00	Communications PCA is 64120-66508 (new) 64120-69508 (exchange)	D200066241	487

- HP TEAMWORK - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64711S004	02.30	"background" colour change when text is selected.	1650054486	488
	64711S004	02.30	Simultaneous socket connections cause a hang.	D200089342	489
	64711S004	02.30	DOMAIN -SQRT ERROR generated when Data Flows become tangential to bubble	D200090118*	69
	64711S004	02.30	When a bubble is moved in a data flow diag., the old num. isn't removed.	D200090480	490
	64711S004	02.20	Data base error caused by NOTE manipulation. -- fixed in 2.3	5000263111	489
	64711S004	02.20	When moving a large group with "group move", some boxes are left hanging	5000283184	489
	64711S004	02.03	The funct. of the mid. and rt. button of 46060b needs to be exchanged.	1650059162	489
	64711S004	02.00	Removing models from the index does not delete all its files.	1650033720	488
	64711S004	01.00	The laser printer 'loses' a few columns across page breaks in a DFD.	1650032698	488

- HP TEAMWORK SA - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64711-90903	01.00	Manual should include guidelines for swap space config when installed.	5000235143	492
	64711-90903	01.00	PRINT OBJECTS from the PI doesn't work correctly.	D200077636	492
	64711-90903	01.00	Spline is too large for binder.	D200077891	492
	64711-90903	01.00	twk_image -dd misbehaves.	D200090274	492
MANUAL	64710-90901	00.01	Need clarification in Default Printer section of manual.	D200066654	491
			- HP-UX 6800-03 C - SSB ISSUE DATE: 09/01/88		
********none********	64821-90902	01.40	Declaring a function which returns a ptr to a function causes error.	D200055780	493
			- HP-UX 68000/8/10 A M - SSB ISSUE DATE: 09/01/88		
********none********	64845-90905	01.30	Assembler flagging LR error for correct offset when using PC+IND+OFFSET.	1650004499	495
	64845-90905	01.30	Wrong offset calculated when using PC+index reg+ offset mode of addr.	D200045880	495
MANUAL	64845-90905	01.04	Cannot substitute Macro parameter at beginning of variable.	D200081836	495
			- HP-UX 68000/8/10 C M - SSB ISSUE DATE: 09/01/88		
********none********	64819-90903	01.40	Declaring a function which returns a ptr to a function causes error.	D200055707	497
	64819-90903	01.40	Byte parameters are pushed onto the stack incorrectly.	D200064386	498
			- HP-UX 68000/8/10 P M - SSB ISSUE DATE: 09/01/88		
TYPE CONVERSION	64815-90907	01.20	Signed_8 to Unsigned_16 is incorrect.	D200036913	499
			- HP-UX 6805/9/9E A -M SSB ISSUE DATE: 09/01/88		
********none********	64844-90905	01.00	Assembler should denote an error on non-absolute .SET expressions.	D200046896	500
			- HP-UX 6809/09E C - SSB ISSUE DATE: 09/01/88		
********none********	64822-90902	01.20	Declaring a function which returns a ptr to a function causes error.	D200055822	501
	64822-90902	01.00	Clarification of interface for USER_DEFINED and real number routines.	D200063651	502
			- HP-UX 8051 ASSM - SSB ISSUE DATE: 09/01/88		
********none********	64855-90903	01.30	The $ operand does not work as defined.	D200053785	505
	64855-90903	01.00	The $ operand does not work as defined.	D200053801	506
MANUAL	64855-90903	01.40	Change 8051 manual page 8-4	5000240937	505
			- HP-UX 8085 C - SSB ISSUE DATE: 09/01/88		
********none********	64826-90902	01.50	New and dispose have inconsistient parameters	1650008128	507

- HP-UX 8085 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64826-90902	01.40	Declaring a function which returns a ptr to a function causes error.	D200055897	507

- HP-UX 8086/88 ASSM M - SSB ISSUE DATE: 09/01/88

CODE GENERATOR	64853-90905	02.20	.	D200079574	509
MANUAL	64853-90905	02.20	.	D200079574	509

- HP-UX 8086/88 C - SSB ISSUE DATE: 09/01/88

********none********	64818-90903	03.10	Declaring a function which returns a ptr to a function causes error.	D200055665	510
	64818-90903	03.02	Additional info about the $SEPARATE_CONST$ directive works, pg. 2-3.	5000211359	510

- HP-UX 8086/88 PAS -M SSB ISSUE DATE: 09/01/88

********none********	64814-90904	01.01	DOC. FOR THE PASCAL LIB. ERROR HANDLING ROUTINES NEEDS IMPROVEMENT.	5000188813	512

- HP-UX OP SYS - SSB ISSUE DATE: 09/01/88

********none********	64801-90903	01.00	Method of entering this CONTROL-M should be explained to the reader.	5000182246	513
	64801-90903	01.00	Meas system unuseable if WINDEX exited without ending measurement.	D200079517	513
	64801-90903	01.00	ftio command for hp-ux 6.01 does not function as documented.	D200090431	513
	64801-90903	00.09	Setting the TERM variable to vt101a will allow use of pmon	5000174805	513

- HP-UX SYSTEM INST -M SSB ISSUE DATE: 09/01/88

********none********	64880-90901	01.02	DOC SHOULD INCLUDE LIST OF SUPPORTED CARDS FOR RS232 XFER.	5000182824	514
	64880-90901	01.00	Manual needs to be more explicit about /dev/ttyXX where XX is numeric	5000269381	514
HIGH SPEED LINK	64880-90901	01.02	Fails to transfer first passworded file, but doesn't notify the user.	D200068429	514

- HP-UX USER DEFIN A M - SSB ISSUE DATE: 09/01/88

********none********	64851-90906	01.20	Assembler should denote an error on non-absolute .SET expressions.	D200047019	515
MANUAL	64851-90906	01.30	'&' is comment field of a macro causes a parameter error.	D200064030	515

- HP-UX Z80/NSC800 A M - SSB ISSUE DATE: 09/01/88

********none********	64842-90904	01.20	Assembler should denote an error on non-absolute .SET expressions.	D200046839	516

- HP-UX Z80/NSC800 C M - SSB ISSUE DATE: 09/01/88

********none********	64824-90902	01.40	Declaring a function which returns a ptr to a function causes error.	D200055855	517

- HP-UX Z8001/02 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64820-90902	01.30	Declaring a function which returns a ptr to a function causes error.	D200055749	519
			- INT SNSL BD 8-16 UPM - SSB ISSUE DATE: 09/01/88		
********none********	64404-90901	01.00	The 68020 emul. may not show all source lines assoc. with exec. code.	5000283630	521
			- INVERSE ASSEMB - SSB ISSUE DATE: 09/01/88		
********none********	64856	01.01	Can loop forever when a source file contains macros.	D200077933	522
			- M-STD 1750A ASM - SSB ISSUE DATE: 09/01/88		
********none********	64857-90901	00.54	Assembler should denote an error on non-absolute .SET expressions.	D200047126	523
			- MS1750A ASSEMB - SSB ISSUE DATE: 09/01/88		
********none********	64857	01.04	Incorrect code generated for immediate negative data.	5000231076	524
	64857	01.04	Invalid instruction assembles without error message	D200060491	525
	64857	00.00	'DEFF' generating incorrect code for NAN's.	5000117507	524
			- NETWORK TRANSFER 300 - SSB ISSUE DATE: 09/01/88		
********none********	64887S004	01.00	The transferII utility does not work using nft as the transport	D200093088*	70
			- OP NOTE 68000C AXLSM - SSB ISSUE DATE: 09/01/88		
********none********	5959-2191	01.00	5959-2191 op note say to use "what" to determine revision code. No!	D200091637	526
			- OPERATING SYSTEM - SSB ISSUE DATE: 09/01/88		
********none********	64100	02.10	Problem with Macro code generation.	1650058925	527
	64100	02.10	Xref cannot be generate as an independent listing	5000170118*	71
	64100	02.10	The '.' character is causing problems for rev 2.06 of asm.exe.	5000203620	527
	64100	02.10	ef directive does not work with mneumonics with "."	5000214106	528
	64100	02.10	ASSEMBLER ISSUES DUPLICATE SYMBOL ERRORS FOR THE 'SEGMENT' PSEUDO	D200042036	530
	64100	02.10	Illegal logical expressions are not flagged.	D200066308	532
	64100	02.10	Comment field can not be delimited by whitespace.	D200069658	532
	64100	02.10	Macro use of a label is missing from xref.	D200086694	534
	64100	02.10	Undefined label not flagged when passed as a parameter to a macro.	D200086728	535
	64100	02.10	Erroneaous phase errors generated by Z8000 assembler	D200089490	536
	64100	02.07	Recover cmd on 64000(PISCES I) will recover all types on disc's > 150Mb	D200084897	533
	64100	02.02	Nested macro calls cause incorrect macro expansion.	D200041178	529

Keyword	Product number	uu.ff	Description	KPR number	page
			- OPERATING SYSTEM - SSB ISSUE DATE: 09/01/88		
********none********	64100	02.00	CDC FLOPPY DRIVE DOESN'T FORMAT CORRECTLY IN A COMMAND FILE.	D200015297	529
	64100	00.70	Phase error incorrectly reported on 64000 and hosted assemblers.	D200085043	534
CODE GENERATOR	64100	02.10	Intractive link gives invalid lnk/command (.K) file if full path too big	5000209007	528
	64100	02.10	Condtl code not assembled if condtl statmnt is false and missing ENDIF	5000214189*	71
	64100	02.10	Assembler output listing is missing part of line number at EQU statement	5000252825	529
	64100	02.10	Hosted version of GET_ASCII_BYTE strips high order bits of input.	D200079368	533
	64100	02.10	XREF option does not work for instructions that contain a period.	D200087049	535
DC600	64100	02.06	store to DC600 causes 64000 to reboot.	D200069989	532
	64100	02.06	May cause inadvertent overwrite of user's disc.	D200074450	532
	64100	01.39	DC600 backup hangs up when it encounters a defective tape.	2700005769	527
			- P1750 -E SSB ISSUE DATE: 09/01/88		
********none********	64288S004	01.00	Need different monitor names for the F9450 and P1750	D200090282	537
	64288S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090936	537
			- PROM PROGRAMMER -3 SSB ISSUE DATE: 09/01/88		
********none********	64501S004	01.30	PROM programmer has problems in UX envr programming 32 bit system.	5000240952	538
	64501S004	00.00	prommer taking too long to program prom.	1650028860	538
SFNONE	64501S004	01.00	8751 does not program in 64000-ux environment.	5000187617	538
			- ROM EMULATION - SSB ISSUE DATE: 09/01/88		
********none********	64272	01.04	store command generates 16-bit width absolute file only	5000231571	539
			- RS-232 TRANSFER -3 SSB ISSUE DATE: 09/01/88		
********none********	64885	01.30	Cannot install software on AXE environment machine.	5000267054	540
			- RS-232 TRANSFER -5 SSB ISSUE DATE: 09/01/88		
TRANSFER	64884	01.10	Transfer hangs after bad options message is displayed.	D200065219	541
			- RS-232 TRANSFER -V SSB ISSUE DATE: 09/01/88		
TRANSFER	64886	01.10	Inaccurate specification in HELP for TRANSFER command	5000194951	542
			- SOFTKEY EDITOR - SSB ISSUE DATE: 09/01/88		
********none********	64790-90901	01.00	The find command does not work correctly, cannot find string includ '$'.	5000211375	543
	64790-90901	01.00	AND '$' NEEDS TO BE ESCAPED ON COMMAND LINE TO PREVENT SHELL EXPANSION	D200089896	543

Keyword	Product number	uu.ff	Description	KPR number	page
			- SOFTKEY EDITOR - SSB ISSUE DATE: 09/01/88		
********none********	64790S004	02.10	Status line does not change after file is written for the save command.	5000401349*	72
	64790S004	02.10	sk editor replace command does not work properly with anystring (*).	5000401372*	72
	64790S004	02.10	When retrieving enough lines to get file exactly 1024 in size; core dump	D200090241	544
	64790S004	02.00	When 4 retrieves are done, the sk editor jumps to shell.	1650061580	544
	64790S004	01.10	Sk may not work when called from pmon.	5000205054	544
			- STATE ANALYZER - SSB ISSUE DATE: 09/01/88		
********none********	64620	00.71	Source referencing will not work with non-zero segments (8086, etc)	D200075028	546
	64621	01.07	trace file may end up in random userid.	5000136135	547
			- STATE ANALYZER - SSB ISSUE DATE: 09/01/88		
********none********	64620S004	01.10	File names <8 chars in link_sym will cause translate problems	D200091538	548
			- SW PERF ANALYZER - SSB ISSUE DATE: 09/01/88		
********none********	64310	01.11	"show curr_meas" after measurement change crashes station.	5000122374	549
			- SW PERF ANALYZER 300 - SSB ISSUE DATE: 09/01/88		
********none********	64310S004	01.20	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080176	550
	64310S004	01.20	Using Emulation across RFA can give incomplete symbol information	D200081026	550
	64310S004	01.20	Processes sometimes left running after parent has stopped.	D200082347	550
			- SYSTEM SOFTWARE - SSB ISSUE DATE: 09/01/88		
MANUAL	64980-90934	02.00	Passing parameters to command files is inconsistient.	D200061515	552
			- TIMING ANALYZER -3 SSB ISSUE DATE: 09/01/88		
********none********	64610S004	01.60	Processes sometimes left running after parent has stopped.	D200082370	553
			- TIMING/STATE - SSB ISSUE DATE: 09/01/88		
********none********	64610	01.00	label cannot be deleted in trigger specification	5000089359	554
	64610	01.00	Hitting CLR LINE causes softkeys to return to first level	5000089367	554
	64610	01.00	TIMING PV MODIFICATION TO ALLOW GREATER TIME TOLLERANCE	D200043794	554
			- TMS 320 ASSEMB - SSB ISSUE DATE: 09/01/88		
PROBLEM ON VAX	64858	00.00	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093401*	73

```
                                                  - TMS320C25 -          SSB ISSUE DATE:  09/01/88

Keyword               Product number   uu.ff Description                                                 KPR number page

********none******** 64787            00.01 Data words at address 6 & 7 can apparantly be displayed and modified     D200090522  555
                                                  - UPROG -              SSB ISSUE DATE:  09/01/88

********none******** 64276            01.00 IN UP_CNTL,"LIST TRACEDATA" SHOWS "AND" EVEN IF NO "ABSOLUTE_IS"         D200035261  556
                     64276            01.00 IN UP_CNTL, NO ERRMSG ISSUED IF "RUN UNTIL W/JAM ATTEMPTED W/O JAM LABEL D200035287  556
                                                  - USER DEF ASSEMB -    SSB ISSUE DATE:  09/01/88

********none******** 64851            00.70 Expand Directive not working on 64000.                                   5000251322  557
                     64851            00.70 Duplicate Symbols in Symbols Declarations not flagged as an error.       D200068924  557
                     64851            00.70 Duplicate SYMBOLS Definitions are not flagged as an error                D200068932  557
                     64851            00.70 Bad table code generated when more than 25 SYMBOLS definitions           D200068940  558
                     64851            00.70 REPT will only take arguments range 1 thru 32767                         D200089409  559
                     64851            00.70 Page size is different between PI and Hosted assemblers                  D200089433  559
                     64851            00.70 line number only 16-bits in size...  This is too small for long files.   D200089458  559
                     64851            00.70 COPY :asmb_sym to display behaves like disc_image on.                    D200092619*  74
CODE GENERATOR       64851            00.70 High order bits stripped from source characters in Pisces I              D200079376  558
                     64851            00.70 Problem with parameter passing in macros                                 D200093781*  74
                                                  - USER DEF ASSEMB -3   SSB ISSUE DATE:  09/01/88

********none******** 64851S004        02.10 Undefined label not flagged when passed as a parameter to a macro.       5000251348  560
                     64851S004        02.10 Macro use of a label is missing from xref.                               5000252833  560
                     64851S004        02.10 Undefined Error placed on all macro usage, if just one label undefined   5000294181  561
                     64851S004        02.10 DE errors anr not declared in all cases for forward references.          D200087569  564
                     64851S004        02.00 Line number for EQU is not completely dislayed after line 1000           5000278606  561
                     64851S004        01.20 expressions of the form 123456.78 cause errors                           D200081646  564
                     64851S004        01.10 Conditional assembly for INCLUDE files causes error.                     D200065417  563
                     64851S004        01.00 Macro def. including .IF, within a IF causes assembler to stop code gen. D200053512  562
                     64851S004        01.00 Comments not delimited by semi-colons appear in the assembler xref.      D200055541  562
                     64851S004        01.00 Host compilers do not put absolute pats specifications in relocatables   D200059311  563
                     64851S004        01.00 QUOTING CHARACTERS WITHIN STRINGS ARE ALL TRANSLATED TO "."              D200059964  563
                     64851S004        00.70 Jump to MACRO label causing expression type error.                       D200092288*  76
MACRO                64851S004        01.00 Conditional instr. .IF with rational oper. in Macro creates bad code     D200048421  561
                                                  - USER DEF ASSEMB -5   SSB ISSUE DATE:  09/01/88

********none******** 64851S001        02.10 Macro use of a label is missing from xref.                               D200086702  565
                     64851S001        02.10 Undefined label not flagged when passed as a parameter to a macro.       D200086736  566
                     64851S001        02.10 DE errors anr not declared in all cases for forward references.          D200087544  566
                     64851S001        01.50 Conditional assembly for INCLUDE files causes error.                     D200065391  565
                                                  - USER DEF ASSEMB -D   SSB ISSUE DATE:  09/01/88

********none******** 64851S006        02.11 Linker xref has bad defs & refs - may crash                              D200090183  568
```

- USER DEF ASSEMB -D SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851S006	02.11	MS-DOS linker will not accept .k file as a linker command file	D200090191	568
	64851S006	02.11	REAL number pseudos & SCAN_REAL function no good on MS-DOS	D200090266	568
	64851S006	02.11	Can not assemble a file on a different disk...(ie: 'A:' from C:)	D200091272	569
	64851S006	02.11	Assembler crashes when directory path name is too long	D200091314	569
	64851S006	02.10	The "INSTALL.BAT" file has some problems, preventing installation	D200088492	567
	64851S006	00.00	DE errors anr not declared in all cases for forward references.	D200087577	567
CODE GENERATOR	64851S006	00.00	Do not refer to the serial port config as "emulator config"	D200087726	567
			- USER DEF ASSEMB -V SSB ISSUE DATE: 09/01/88		
********none********	64851S003	02.10	NO LOAD files are not handled correctly.	5000402701*	77
	64851S003	02.10	Macro use of a label is missing from xref.	D200086710	570
	64851S003	02.10	Undefined label not flagged when passed as a parameter to a macro.	D200086744	571
	64851S003	02.10	DE errors anr not declared in all cases for forward references.	D200087551	571
	64851S003	01.50	Conditional assembly for INCLUDE files causes error.	D200065409	570
			- USER DEF EMUL - SSB ISSUE DATE: 09/01/88		
********none********	64274S004	01.10	Displaying memory takes much longer in 64000-UX environment than 64000.	5000232991	572
	64274S004	01.10	UDE CLK SOURCE DIFFERENT BETWEEN 64000/64000-UX CONFIGURATION FILES	5000266684*	78
	64274S004	01.10	64000-UX UDE mnemonic mem display for word processors may be garbled	D200079004	572
	64274S004	01.10	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080697	573
	64274S004	01.10	Using Emulation across RFA can give incomplete symbol information	D200080994	573
	64274S004	01.10	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081943	574
	64274S004	01.10	Processes sometimes left running after parent has stopped.	D200082289	574
	64274S004	01.10	Loading a trace file from a different processor may cause core dump	D200083295	574
	64274S004	01.10	Tracelist symbols dissappear.	D200086074	575
	64274S004	01.10	Software breakpoint in target memory will hang system.	D200087270	575
	64274S004	01.10	Monitor fails w/word processot if DADA not mapped to low memory	D200090324	575
	64274S004	01.10	rd/wr ioport at odd addr accesses to many ports for word processors	D200090621	575
	64274S004	01.10	Code disp. with trace not right if code changed w/o ending emul. session	D200090910	576
	64274S004	01.00	No mnemonic display during stepping of the NEC 7500	D200077966	572
			- USER DEF EMULATION - SSB ISSUE DATE: 09/01/88		
********none********	64274	01.05	64000-UX UDE Using "HOME" key causes trace list problems	5000296855*	79
			- USER DEFIN ASM - SSB ISSUE DATE: 09/01/88		
********none********	64851-90904	01.00	DE must be defined before being referenced.	5000153981	578
	64851-90904	00.70	64000 station resets when linking if SKELETON command used improperly	D200079558	578
MANUAL	64851-90904	00.70	'&' is comment field of a macro causes a parameter error.	5000152892	578
			- USER INTERFACE - SSB ISSUE DATE: 09/01/88		
********none********	64808-90901	01.00	Need to add Note saying that 64100 Terminal Mode is not supported.	5000267468	580

```
                                                - USER INTERFACE -        SSB ISSUE DATE:  09/01/88

Keyword              Product number   uu.ff Description                                                  KPR number page

********none******** 64808-90901      01.00 PMON doesn't allow a file to begin with a numeric value.      5000291427  580

                                                - USER INTERFACE -        SSB ISSUE DATE:  09/01/88


********none******** 64808S004        02.10 Pmon flags legitimate option for lnk (for 64859) as syntax error       5000296921*  80
                     64808S004        02.10 A command file containing these three characters in that order #'! fails D200090613  581
                     64808S004        01.10 PMON not interpreting a command file correctly.              5000178301  581

                                                - USER INTERFACE -        SSB ISSUE DATE:  09/01/88


********none******** 64808S001        01.40 PMON not interpreting a command file correctly.              D200069369  582

                                                 - UTILITIES PKG -        SSB ISSUE DATE:  09/01/88


********none******** 64888S003        01.40 Record attributes on VMS files are incorrect.                5000267005  583
                     64888S004        01.00 Enhancements are not displayed correctly in ANSI mode.       D200078048  584

                                                 - VMS 6500 ASSM -        SSB ISSUE DATE:  09/01/88


********none******** 64843-90904      01.00 .LIS file should be put in same directory as .A and .R files.  D200067124  585

                                                 - VMS 6800 ASSM -        SSB ISSUE DATE:  09/01/88


********none******** 64841-90907      01.20 Assembler should denote an error on non-absolute .SET expressions.  D200046813  586
                     64841-90907      01.00 .LIS file should be put in same directory as .A and .R files.  D200067082  586

                                                 - VMS 6800-03 C -        SSB ISSUE DATE:  09/01/88


********none******** 64821-90903      01.50 Declaring a function which returns a ptr to a function causes error.  D200055798  587
                     64821-90903      01.00 .LIS file should be put in same directory as .A and .R files.  D200066969  588

                                               - VMS 68000/08/10 C -M      SSB ISSUE DATE:  09/01/88


********none******** 64819-90904      01.50 Declaring a function which returns a ptr to a function causes error.  D200055715  589
                     64819-90904      01.00 .LIS file should be put in same directory as .A and .R files.  D200066928  590

                                             - VMS 68000/8/10 ASM M -      SSB ISSUE DATE:  09/01/88


********none******** 64845-90906      01.60 Document REG pseudo                                          5000160143  591
                     64845-90906      01.30 LR error flagged for correct offset using PC+INDEX+OFFSET mode of addr.  D200046268  592
                     64845-90906      01.00 .LIS file should be put in same directory as .A and .R files.  D200067165  592
```

- VMS 68000/8/10 ASM M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
MANUAL	64845-90906	01.60	Manual explains linker options incorrectly.	5000220764	591
	64845-90906	01.60	Manual states incorrectly that EXT is a pseudo op.	5000220772	591

- VMS 68000/8/10 P - SSB ISSUE DATE: 09/01/88

********none********	64815-90908	01.00	.LIS file should be put in same directory as .A and .R files.	D200066860	594
TYPE CONVERSION	64815-90908	01.20	Signed_8 to Unsigned_16 is incorrect.	D200036921	594

- VMS 6805/9/9E ASM -M SSB ISSUE DATE: 09/01/88

********none********	64844-90906	01.20	Assembler should denote an error on non-absolute .SET expressions.	D200046904	595
	64844-90906	01.00	.LIS file should be put in same directory as .A and .R files.	D200067140	595

- VMS 6809 PASCAL - SSB ISSUE DATE: 09/01/88

********none********	64813-90905	00.01	.LIS file should be put in same directory as .A and .R files.	D200066829	596

- VMS 6809/09E C - SSB ISSUE DATE: 09/01/88

********none********	64822-90903	01.20	Declaring a function which returns a ptr to a function causes error.	D200055830	597
	64822-90903	01.00	Clarification of interface for USER_DEFINED and real number routines.	D200063669	598
	64822-90903	01.00	.LIS file should be put in same directory as .A and .R files.	D200066985	600

- VMS 8051 ASSM - SSB ISSUE DATE: 09/01/88

********none********	64855-90904	01.40	The $ operand does not work as defined.	D200053793	601
	64855-90904	01.20	Assembler should denote an error on non-absolute .SET expressions.	D200047118	601
	64855-90904	01.00	.LIS file should be put in same directory as .A and .R files.	D200067264	601

- VMS 8080/85 ASSM - SSB ISSUE DATE: 09/01/88

********none********	64840-90903	01.40	.LIS file should be put in same directory as .A and .R files.	D200067066	603

- VMS 8085 C - SSB ISSUE DATE: 09/01/88

********none********	64826-90903	01.60	Declaring a function which returns a ptr to a function causes error.	D200055905	604
	64826-90903	01.00	.LIS file should be put in same directory as .A and .R files.	D200067041	605

- VMS 8085 PASCAL - SSB ISSUE DATE: 09/01/88

********none********	64825-90903	01.00	.LIS file should be put in same directory as .A and .R files.	D200067025	606

- VMS 8086/88 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64818-90904	03.10	Declaring a function which returns a ptr to a function causes error.	D200055673	607
	64818-90904	03.00	.LIS file should be put in same directory as .A and .R files.	D200066902	608

- VMS 8086/88 PASCAL M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64814-90905	03.00	.LIS file should be put in same directory as .A and .R files.	D200066845	609

- VMS 9900 ASSM - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64847-90905	01.00	.LIS file should be put in same directory as .A and .R files.	D200067181	610

- VMS FILE FORMATS - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64882-90903	01.02	VAX file format manual doesn't give clear explantion of VAX file types.	5000238543	611
	64882-90903	01.01	Linker symbol file format (Chapter 14) Word #6 not defined	D200053132	611

- VMS M-STD1750A ASM M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64857-90903	01.00	.LIS file should be put in same directory as .A and .R files.	D200067280	612

- VMS SYSTEM INSTAL -M SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64882-90904	01.60	Number of errors in Appendix A of the manual, please read submit. text.	5000195701	613
	64882-90904	01.60	Need instructions to MACRO and link ibdriver for single high speed link	D200062844	613
	64882-90904	01.03	Need setting for rear panel of old HP 64000 and 64110(with jumper jacks)	D200055202	613
	64882-90904	01.00	For VMS=>4 using HP 64000 as VMS terminal need to correct manual (pg8-3)	5000167601	613

- VMS TMS 320 ASSM - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64858-90903	01.00	.LIS file should be put in same directory as .A and .R files.	D200067306	614

- VMS USER DEFIN ASM M - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64851-90907	01.20	Assembler should denote an error on non-absolute .SET expressions.	D200047027	615
	64851-90907	01.00	.LIS file should be put in same directory as .A and .R files.	D200067207	616
MANUAL	64851-90907	01.40	'&' is comment field of a macro causes a parameter error.	D200064048	615

- VMS USERS GUIDE - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64882-90902	01.60	Page 3-9 states vt52 emulation using 64100 but does not perform functs.	5000222489	617
	64882-90902	01.01	Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.	D200045492	617

```
                                             - VMS Z80/NSC800 ASM M -     SSB ISSUE DATE:  09/01/88

Keyword              Product number   uu.ff Description                                                     KPR number page

********none******** 64842-90905      01.20 Assembler should denote an error on non-absolute .SET expressions.    D200046847  618
                     64842-90905      01.00 .LIS file should be put in same directory as .A and .R files.        D200067108  618

                                               - VMS Z80/NSC800 C -       SSB ISSUE DATE:  09/01/88

********none******** 64824-90903      01.50 Declaring a function which returns a ptr to a function causes error.  D200055863  619
                     64824-90903      01.00 .LIS file should be put in same directory as .A and .R files.        D200067009  620

                                               - VMS Z80/NSC800 P -       SSB ISSUE DATE:  09/01/88

********none******** 64823-90903      01.00 .LIS file should be put in same directory as .A and .R files.        5000163295  621

                                                - VMS Z8001/02 C -        SSB ISSUE DATE:  09/01/88

********none******** 64820-90903      01.50 Declaring a function which returns a ptr to a function causes error.  D200055756  622
                     64820-90903      01.00 .LIS file should be put in same directory as .A and .R files.        D200066944  623

                                               - VMS Z8001/2 ASSM -       SSB ISSUE DATE:  09/01/88

********none******** 64854-90904      01.00 .LIS file should be put in same directory as .A and .R files.        D200067249  624

                                             - VMS Z8001/2 PASCAL M -     SSB ISSUE DATE:  09/01/88

********none******** 64816-90905      01.00 .LIS file should be put in same directory as .A and .R files.        D200066886  625

                                             - VMS6800/01/02/03 P M -     SSB ISSUE DATE:  09/01/88

********none******** 64811-90904      01.00 .LIS file should be put in same directory as .A and .R files.        D200066803  626

                                                    - Z8 ASSEMB -         SSB ISSUE DATE:  09/01/88

********none******** 64850            00.01 Assembler generates Phase Error of forward referenced EQU            D200091645  627
                     64850            00.00 Assembler not generating error message when attempt to load label.   2700005918  627

                                                       - Z80 -            SSB ISSUE DATE:  09/01/88

********none******** 64753            00.01 Emulator may be confused about running/monitor state                 D200088047  628
                     64753            00.00 In quickbreak mode, the Z80 could break without generic emul's knowledge D200085290  628

                                                       - Z80 -            SSB ISSUE DATE:  09/01/88

********none******** 64753S006        01.00 Invalid COM port in 64700tab file                                    D200091256  629
```

- Z80 ASSEMB - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64842	01.12	Using HEX psuedo is causing bad address calculations.	5000139535	630
	64842	01.12	Xref lists symbols which are under False conditional assembly blocks.	5000239939	630
	64842	01.12	Difference between 64000 and host in XREF when no symbols.	D200086686	631
	64842	01.11	Revision number on output listing is incorrect.	5000152819	630
	64842	01.10	Complex macro interaction causing invalid errors.	5000264986	631
PROBLEM ON VAX	64842	01.12	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093310*	81

- Z80 EMULATION - SSB ISSUE DATE: 09/01/88

********none********	64252S004	01.00	HPIB 64120 I/O AND POWER FAILED WHEN MODIFYING TARGET MEMORY	1650047167	632
	64252S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069542	632
	64252S004	01.00	pwd truncates the /net7system portion of the path when RFA'ed to system.	D200080655	632
	64252S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080952	633
	64252S004	01.00	PC contents lost over continuation if in break state	D200081489	633
	64252S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081901	633
	64252S004	01.00	Processes sometimes left running after parent has stopped.	D200082230	634
	64252S004	01.00	Loading a trace file from a different processor may cause core dump	D200083246	634
	64252S004	01.00	IMPROPER IDENTIFICATION OF THE SECOND Z80 CONTRL CARD IF TWO Z80 PRESENT	D200085332	634
	64252S004	01.00	Tracelist symbols dissappear.	D200086033	634
	64252S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090860	635
	64252S004	00.00	EMULATION SOFTWARE STATUS DOES NOT RECOGNIZE THE "HALT" INSTRUCTION	1650047340	632

- Z80 PASCAL - SSB ISSUE DATE: 09/01/88

CODE GENERATOR	64812	00.00	$ORG directive can cause incorrect code to be generated.	1650041624	636

- Z80 PASCAL - SSB ISSUE DATE: 09/01/88

RUN-TIME LIBRARY	64812-90903	00.00	Library routine 'Zintabs' (DLIBZ80:CZ80) destroys the accumulator.	D200010363	637

- Z80/NSC800 C - SSB ISSUE DATE: 09/01/88

********none********	64824	02.10	Wrong code generated for assignment operator <<= if used with arrays.	1650058123	638
	64824	02.10	Certain Byte additions with word results may fail in error #1009	D200085472	641
	64824	02.10	Indirect comparison of parameter bytes may fail	D200090175	641
	64824	02.10	Certain set operations with explicit type changes may fail.	D200090217	642
	64824	01.04	Error 1006 given for > test condition.	5000245704	638
	64824	01.04	Reference to non-existient library in manual.	5000259150	639
	64824	01.04	Real variable used as a test condition cause error.	D200081554	640
NO PROBLEM/ PISCES I	64824	02.10	Compilers do not list complete information about source file path name.	D200092924*	82
PASS 3	64824	01.03	Conditional compile fails if it suceeds a fixed parm function call.	D200069906	639
PROBLEM ON 9000/S300	64824	02.10	Compilers do not list complete information about source file path name.	D200092924*	82
PROBLEM ON 9000/S500	64824	02.10	Compilers do not list complete information about source file path name.	D200092924*	82
PROBLEM ON VAX	64824	02.10	Compilers do not list complete information about source file path name.	D200092924*	82

- Z80/NSC800 C - SSB ISSUE DATE: 09/01/88

********none********	64824-90901	01.02	Declaring a function which returns a ptr to a function causes error.	D200055590	643

- Z80/NSC800 P - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64823-90901	01.00	Documentation and examples for Z80 I/O port	5000170191	645

- Z80/NSC800PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64823	01.90	Certain set operations with explicit type changes may fail.	D200090209	649
	64823	01.04	Error 1006 for complex statement using mod operator.	1650049163	646
	64823	01.03	Register may be incorrectly remembered following byte negate.	D200071357	647
CODE GENERATOR	64823	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087346	648
NO PROBLEM/ PISCES I	64823	01.90	Compilers do not list complete information about source file path name.	D200092866*	83
NOT ON 64100 SYSTEM	64823	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087346	648
PASS 1	64823	01.02	$Range ON$ causes incorrect code to be generated for a test operation.	D200059600	646
PASS 3	64823	01.02	Incorrect data offsets in listing file.	D200060186	647
PROBLEM ON 9000/S300	64823	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087346	648
	64823	01.90	Compilers do not list complete information about source file path name.	D200092866*	83
PROBLEM ON 9000/S500	64823	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087346	648
	64823	01.90	Compilers do not list complete information about source file path name.	D200092866*	83
PROBLEM ON VAX	64823	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087346	648
	64823	01.90	Compilers do not list complete information about source file path name.	D200092866*	83
	64823	01.90	VAX Pascal xref prints many garbage characters of first line xref list	D200092999*	83

- Z8000 C - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64820	01.06	Logical AND produces a multiply operation.	5000160671	651
	64820	01.06	Local parms not accessed properly when func called via pointer.	5000246983	653
	64820	01.06	Oversized data segment not being flagged as an error.	D200078873	657
	64820	01.06	Real variable used as a test condition cause error.	D200081521	659
	64820	01.06	Function calls via pointers with parameters mess up subsequent calls.	D200085381	660
	64820	01.05	Superfluous register load in switch statement on the 64000	D200064808	655
	64820	01.05	Illegal initialization causes error 1113.	D200068155	656
	64820	01.04	$OPTIMIZE$ compiler directive works differently for signed and unsigned.	5000181545	652
	64820	01.04	Inconsistient error message when linking ASM.R files versus COMP.R files	D200061762	654
	64820	00.01	Code generated for unsigned multiply is the same as for signed multiply.	1650006544	651
NO PROBLEM/ PISCES I	64820	02.10	Compilers do not list complete information about source file path name.	D200092890*	84
PASS 3	64820	01.05	Conditional compile fails if it suceeds a fixed parm function call.	D200069781	657
PROBLEM ON 9000/S300	64820	02.10	Compiler does not create an 'array too large' error when size > 32k.	5000280958	654
	64820	02.10	Compilers do not list complete information about source file path name.	D200092890*	84
	64820	01.06	If condition is tested with a CMP D1,D1	D200079616	658
PROBLEM ON 9000/S500	64820	02.10	Compilers do not list complete information about source file path name.	D200092890*	84
PROBLEM ON VAX	64820	02.10	Compilers do not list complete information about source file path name.	D200092890*	84

- Z8000 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64816	01.12	$RANGE$ & type conversion of UNSIGNED_32 var may cause error 1006.	D200085282	665
	64816	01.11	BA address mode may attempt to use RR0 Illegally as source	D200073015	665
	64816	01.10	Inconsistient error message when linking ASM.R files versus COMP.R files	D200061721	664
	64816	01.09	Jump table may generate code which accesses wrong data space.	5000123497	661

- Z8000 PASCAL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64816	01.04	"Downto" used in a for statement generates incorrect code.	5000150151	662
CODE GENERATOR	64816	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087338	665
NO PROBLEM/ PISCES I	64816	01.90	Compilers do not list complete information about source file path name.	D200092858*	85
NOT ON 64100 SYSTEM	64816	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087338	665
PASS 3	64816	01.10	Calling func. twice in statement causes return value to be overwritten	5000134916	661
	64816	01.10	Compiler $FAR ON$, creates incorrect data offsets in listing	D200060145	663
PROBLEM ON 9000/S300	64816	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087338	665
	64816	01.90	Compilers do not list complete information about source file path name.	D200092858*	85
PROBLEM ON 9000/S500	64816	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087338	665
	64816	01.90	Compilers do not list complete information about source file path name.	D200092858*	85
PROBLEM ON VAX	64816	01.90	"Too many errors pass3" err msg, if use duplicate labels.Need better msg	D200087338	665
	64816	01.90	Compilers do not list complete information about source file path name.	D200092858*	85
	64816	01.90	VAX Pascal xref prints many garbage characters of first line xref list	D200092981*	85

- Z8001 EMUL - SSB ISSUE DATE: 09/01/88

********none********	64232S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080580	667
	64232S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080887	667
	64232S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081869	667
	64232S004	01.00	Processes sometimes left running after parent has stopped.	D200082164	668
	64232S004	01.00	Loading a trace file from a different processor may cause core dump	D200083170	668
	64232S004	01.00	Tracelist symbols dissappear.	D200085977	668
	64232S004	01.00	Using simio, then continuing , may not be possible	D200086330	669
	64232S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088294	669
	64232S004	01.00	Software breakpoint in target memory will hang system.	D200088443	669
	64232S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090795	669

- Z8001/02 C - SSB ISSUE DATE: 09/01/88

********none********	64820-90901	01.04	Declaring a function which returns a ptr to a function causes error.	D200055731	671

- Z8001/2 ASSEMB - SSB ISSUE DATE: 09/01/88

********none********	64854	01.80	*PRODUCT # CHANGE on the VAX* From= 64xxxS003 To=64xxxM003	D200093724*	86

- Z8001/2 EMUL - SSB ISSUE DATE: 09/01/88

********none********	64980-90923	01.00	Need more info on sharing user system calls & monitor interaction.	5000131573	673

- Z8002 EMUL - SSB ISSUE DATE: 09/01/88

********none********	64233S004	01.00	Incorrect breakpoint behaviour on continuing emulation.	D200072462	674
	64233S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080598	674
	64233S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080895	674

- Z8002 EMUL - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64233S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081877	675
	64233S004	01.00	Processes sometimes left running after parent has stopped.	D200082172	675
	64233S004	01.00	Loading a trace file from a different processor may cause core dump	D200083188	675
	64233S004	01.00	Tracelist symbols dissappear.	D200085985	676
	64233S004	01.00	Using simio, then continuing , may not be possible	D200086348	676
	64233S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088302	676
	64233S004	01.00	Software breakpoint in target memory will hang system.	D200088450	677
	64233S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090803	677

- Z80H EMULATION - SSB ISSUE DATE: 09/01/88

Keyword	Product number	uu.ff	Description	KPR number	page
********none********	64253S004	01.00	CANNOT ACCESS COMPILER GENERATED SYMBOLS IN HP64000-UX EMUL ENVIRONMENT	5000258616	678
	64253S004	01.00	Measurement System end_released when terminal cannot be initialized	D200069559	678
	64253S004	01.00	pwd truncates the /net/system portion of the path when RFA'ed to system.	D200080663	678
	64253S004	01.00	Using Emulation across RFA can give incomplete symbol information	D200080960	678
	64253S004	01.00	The Inter-Module-Bus trigger signal latches when set to drive & receive	D200081919	679
	64253S004	01.00	Processes sometimes left running after parent has stopped.	D200082248	679
	64253S004	01.00	Loading a trace file from a different processor may cause core dump	D200083253	680
	64253S004	01.00	Tracelist symbols dissappear.	D200086041	680
	64253S004	01.00	Using simio, then continuing , may not be possible	D200086389	680
	64253S004	01.00	"end" softkey after HP-IB error does not clear command line	D200088344	681
	64253S004	01.00	Code disp. with trace not right if code changed w/o ending emul. session	D200090878	681

KPR #: D200092122 Product: 6301V/03R EMUL 64206 01.01

One-line description:
Illegal opcode error occur when displaying memory repetetively

KPR #: D200082065 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200085886 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086272 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088237 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. . The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090696 Product: 6301X EMULATION 300 64207S004 00.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200082073 Product: 6301Y EMULATION 300 64208S004 00.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 04/29/88 in release A01.01

KPR #: D200085894 Product: 6301Y EMULATION 300 64208S004 00.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

Signed off 04/29/88 in release A01.01

KPR #: D200086280 Product: 6301Y EMULATION 300 64208S004 00.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

KPR #: D200086280 **CONTINUED**

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

Signed off 04/29/88 in release A01.01

KPR #: D200088245 Product: 6301Y EMULATION 300 64208S004 00.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

Signed off 04/29/88 in release A01.01

KPR #: D200090704 Product: 6301Y EMULATION 300 64208S004 00.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 5000269779 Product: 650X ASSEMB 64843 01.00

Keywords: PROBLEM ON 9000/S300 LINKER

One-line description:
LNK does load NOLOAD files.

Problem:
linker DOES load and link NO LOAD objects.

EXAMPLE

```
    object ?  abc_a.R,(abc_b.R)
    library?
    load address ... ... ....

  abc_b.R relocatable file is loaded and linked.
```

Temporary solution:
The linker is operating correctly. The linker listing file is in error in that it shows that the (noload) file has been loaded when it actually has not.

KPR #: D200093328 Product: 650X ASSEMB 64843 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.80

KPR #: D200092908 Product: 6800 C 64821 02.10

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.20

KPR #: D200092817   Product: 6800 PASCAL   64811   01.90

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Temporary solution:
None.

Signed off 08/31/88 in release A02.00

KPR #: D200092940   Product: 6800 PASCAL   64811   01.90

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Temporary solution:
None.

Signed off 08/31/88 in release A02.00

KPR #: D200093534   Product: 6800 PASCAL   64811   02.00

One-line description:
Type casting the ADDR function to SET for masking may cause error

Problem:
Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
    Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

KPR #: D200093534 **CONTINUED**

will geneate incorrect code.

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

HERE is an expanded example:

```
 "PASCAL"
 "6800"
 PROGRAM Error;
 $EXTENSIONS$
 TYPE
     BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
     SET_OF_BITS = SET OF BITS;
 VAR
     S : SET_OF_BITS;
     Byte1,Byte2: BYTE;
     I : SIGNED_16;
 PROCEDURE BadADDRsetMASK;
  BEGIN
   Byte1:=BYTE(
   (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
             ^505
        LDAA #000H
        LDAB #0FFH
        JSR  Zuintmul           <----------Should be AND function
        STAA DLSDqf02563
        STAB DLSDqf02563+00001H
        LDAB DLSDqf02563
        STAB DPTEST110+00002H
   END;
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
    Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
    TempADDR := ADDR(variable);

    Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: D200093542 Product: 6800 PASCAL 64811 02.00

One-line description:
Large Sets may produce invalid results for elements outside set range

Problem:
The set inclusion operation may test undefined bit when the element being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

```
TYPE
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT   {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}     { Branch should always be FALSE,         }
    ELSE   {...}     {  But the result is due to invalid bit test}
END.
```

Temporary solution:
  WORKAROUND:

The workaround for this defect is to separate the use of the the full 256 bit set implementation.

Instead of defining the large set as:

```
   DIG = SET OF '0'..'9';
```

It could be rewritten:

```
   digch = SET OF CHAR;
```

With the sets now using a full 256 bits, all bits will be set and tested properly.

```
------------
TYPE
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT   {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}     { Branch should always be FALSE,         }
    ELSE   {...}     {  But the result is due to invalid bit test}
END.
```

KPR #: D200093682 Product: 6800 PASCAL 64811 01.90

One-line description:
Compare using var pointer to first record item fails.

Problem:
When accessing the first item in a record (passed as a VAR pointer parameter), in a comparison expressions, the 6800 code generator fails to call the run time comparison routine properly.

This defect can occur with any date item of size greater than 1 byte (i.e. INTEGER, REAL, LONGREAL).

This defect is only reproducible in the HOSTED compilers:
(64811S001 S500/HPUX, 64811S004 S300/HPUX, 64811S003 VAX/VMS)
It occurs on compilers Rev. 1.90 & Rev. 2.00.

This defect appears to create correct code on the HP64811A 64100 (Rev. 1.90 & Rev 2.000) compiler.

The following source program produces the incorrect code.

```
PROGRAM PTEST32;
TYPE
   WORD = RECORD
            KEY : SIGNED_16;
            LEFT, RIGHT : REF;
          END; (* RECORD *)
    REF = ^WORD;

PROCEDURE POINTERROR ( X : SIGNED_16; VAR P : REF );
BEGIN
   IF X < P^.KEY THEN  ;  (P^.KEY passed to Zintles incorrectly)
END;
```

Temporary solution:
WORKAROUND:
  Use a temporary variable:

```
    temp:= P^.KEY;
    IF X < temp THEN ...
```

KPR #: D200093708 Product: 6800 PASCAL 64811 01.90

One-line description:
Asignment of string to double dereference string pointer causes error

Problem:
When attempting to assign a string to a double dereferenced string pointer, the 6800 code generator fails to call the run time string move routine properly.

This defect is only reproducible in the HOSTED compilers:
(64811S001 S500/HPUX, 64811S004 S300/HPUX, 64811S003 VAX/VMS)
It occurs on compilers Rev 1.90 & Rev 2.00.

This defect appears to create correct code on the HP64811A 64100

KPR #: D200093708 **CONTINUED**

(Rev 1.90 & Rev 2.00) compiler.

The following source program produces the incorrect code.

```
PROGRAM doublepointererror;
PROCEDURE BAD_STR_ASN;
TYPE
   STR_ARR = PACKED ARRAY [0..7] OF CHAR;
   ARR_PTR = ^STR_ARR;
VAR
   PTR2 : ^ARR_PTR;
BEGIN
   PTR2^^ := "XY";  {======= BAD call to STmove routine ======}
END;    {BAD_STR_ASN}
BEGIN
END.
```

Temporary solution:
Use a temporary pointer to the final string:

```
  VAR PTR1: ARR_PTR;
  ...
  PTR1:= PTR2^;  PTR1^:= "XY"
```

KPR #: D200093716   Product: 6800 PASCAL   64811   01.90

One-line description:
Pointer dereference of VAR pointer to structure as a parameter fails.

Problem:
A parameter passed in as a VAR pointer to a structure can not be passed properly as a parameter to another routine. The 6800 code generator fails to call the routine properly.

This defect is only reproducible in the HOSTED compilers:
(64811S001 S500/HPUX, 64811S004 S300/HPUX, 64811S003 VAX/VMS)
It occurs on compilers Rev 1.90 & Rev 2.00.

This defect appears to create correct code on the HP64811A 64100 (Rev 1.90 & Rev 2.00) compiler.

The following source program produces the incorrect code.

```
PROGRAM VARerror;
TYPE
  ARTIKEL = RECORD
            ELE1 : INTEGER;
            ELE2 : INTEGER;
            END;
  ARTIKEL_PTR = ^ARTIKEL;
PROCEDURE VarParamRec(VAR ART : ARTIKEL);  EXTERNAL;
PROCEDURE VarParam(VAR ART : ARTIKEL_PTR);
BEGIN
VarParamRec(ART^);  { Parameter NOT passed properly }
END;
```

KPR #: D200093716 **CONTINUED**

```
BEGIN
END.
```

Temporary solution:
Use a temporary pointer variable.

```
  VAR Temp_ptr: ARTIKEL_PTR;
  ...
  Temp_ptr:= ART;
  VarParamRec(Temp_ptr^);
```

KPR #: D200093302 Product: 6800/2 ASSEMB 64841 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.80

KPR #: 5000297879 Product: 68000 12MHZ EMUL FW 64742 00.05

One-line description:
Measurements between the external/internal analyzers aren't synchronized

Problem:
The 68000 emulator does not "synchronize" measurements between the external and internal analyzers when the "xtmo -e" command is issued. The Terminal Interface User's Reference explains that the xtmo -e will ".. synchronize measurements made by the two analyzers."

An example that shows that this is not the case can be shown by plugging into the CSA DEMO BOX and mapping all resources to target. Use "xtmo -e" to synchronize the external analyzer with the internal analyzer. Monitor the R/W- line with one of the external bits, and it will always show as a one, even on write cycles.

Duplicate Service Requests: 5000397877

KPR #: D200093435 Product: 68000 12MHZ EMUL DOS 64742S006 01.01

One-line description:
Invalid COM port in 64700tab file

Problem:
There is an invalid COM port in the 64700ta b file. The second COM port in the 64700tab file should be COM2 instead of COM1.

Temporary solution:
There is no workaround available.

KPR #: D200092882 Product: 68000 C 64819 02.10

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.20

KPR #: D200092106 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
"tlist" can oferwrite memory it did not allocate, causing core dumps

Problem:
The tlist package does not necessarily reallocate its string buffers even if the size of a mnemonic field increases. Consequently, if a field starts out small, then is increased in size, it is very likely that memory not allocated by tlist will be overwritten. This occurs when the "stringbytes" parameter to tlialpp() is increased from call to call.

In HP64000-UX emulation, the problem occurs if the first trace display to appear has a small mnemonic field (display trace width mnemonic 4). A display with a larger field (display trace width mnemonic 40) will then cause a core dump.

This problem exists in all HP64000-UX emulators. Other consumers of the tlist package may have similar problems.

Signed off 08/19/88 in release A01.30

KPR #: D200092841 Product: 68000 PASCAL 64815 01.90

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.00

KPR #: D200092973 Product: 68000 PASCAL 64815 01.90

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Signed off 08/31/88 in release A02.00

KPR #: D200093450 Product: 68000 PASCAL 64815 02.00

One-line description:
Type casting the ADDR function to SET for masking may cause an error.

Problem:
Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
    Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

will geneate incorrect code.

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

KPR #: D200093450 **CONTINUED**

HERE is an expanded example:

```
 "PASCAL"
 "68000"
 PROGRAM Error;
 $EXTENSIONS$
 TYPE
     BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
     SET_OF_BITS = SET OF BITS;
 VAR
     S : SET_OF_BITS;
     Byte1,Byte2: BYTE;
     I : SIGNED_16;
 PROCEDURE BadADDRsetMASK;
  BEGIN
   Byte1:=BYTE(
   (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
                       ^505                                        ^505
         LEA      DPTEST110+00004H[A5],A0
         MOVE.L   A0,-[A7]
         CLR.L    D0
         MOVE.W   #000FFH,D0
         MOVE.L   D0,-[A7]
         JSR      Zunsmult[PC]
         MOVE.W   D7,-2[A6]
         MOVE.B   -2[A6],DPTEST110+00002H[A5]
   END;
```

Temporary solution:
WORKAROUND:

The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
TempADDR := ADDR(variable);
Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: D200092148 Product: 68000C AXLS COMP M 64902-90901 01.00

Keywords: MANUAL

One-line description:
Cport68k documentation has incorrect path name for executable

KPR #: D200092155 Product: 68000C AXLS COMP M 64902-90901 01.00

Keywords: MANUAL

One-line description:
Recursive includes not allowed with lister.

KPR #: D200092171 Product: 68000C AXLS COMP M 64902-90901 01.00

Keywords: MANUAL

One-line description:
lister does not allow recursive includes

KPR #: D200092163 Product: 68000C AXLS COMP 300 64902S004 01.00

Keywords: CODE GENERATOR

One-line description:
'Cannot open file' message coming from lister, if open many includes.

Problem:
The lister will issue the following message: 'Cannot open file' under the following circumstances:

- File A includes many files, which may in turn include many files, and the number of files included is larger that 54, and file A is still open

- None of the include files generate assembly code. (In other words, include files that only define variables, will cause this problem).

ODE TRAILER LINE

Temporary solution:
Change include files to include less that 54 files, as they branch out from original include, or put in some dummy code to cause lister to close files.

KPR #: D200093013 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
Cpp looks in the wrong directory for local include files.

Problem:
When searching for relative local include files (those specified with quotes which don't begin with a '/') cpp always searches relative to the directory the original C source file is in. cpp should be searching relative to the directory that the file doing the including is in. In the following example, cpp won't fine t1.h when it should.

Example:

File t.c contains:
```
#include "temp/t.h"
```

File temp/t.h contains:
```
#include "t1.h"
```

File temp/t1.h contains:
```
int i;
```

Temporary solution:
Use full path names.

KPR #: D200093039 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
Cpp requires white space after #define macro name.

Problem:
The following case is currently errored by cpp for having an

KPR #: D200093039 **CONTINUED**

illegal macro name.

```
#define abc/def 1
```

The ANSI draft doesn't indicate an error in this case. The result should be a macro "abc" defined to be "/def 1".

Temporary solution:
None.

KPR #: D200092346 Product: 68008 EMULATION 300 64244S004 01.00

One-line description:
Doing a wait while tracing MAY cause subsuquent traces to never complete

Signed off 08/19/88 in release A01.30

KPR #: D200092353 Product: 68008 EMULATION 300 64244S004 01.20

One-line description:
"bba unload" causes memory growth problems in emmulators

Problem:
Detailed Listing for Defect Number LSDqf03817
Text:
  bbaunload causes memory growth problems in emulators

The bbaunload library (/lsd/p2/cmd/bba/prod/unload) allocates memory and never frees it.

Please expand to 64244S004 and 64245S004 as well.

Bruce E

FIXED:
The new library fixes these problems as well as making the whole unload process much faster. It is fixed as of 03June88.

Signed off 08/19/88 in release A01.30

KPR #: D200093443 Product: 68010 16MHZ EMUL DOS 64745S006 01.01

One-line description:
Invalid COM port in 64700tab file

Problem:
There is an invalid COM port in the 64700ta b file. The second COM port in the 64700tab file should be COM2 instead of COM1.

Temporary solution:
There is no workaround available.

KPR #: D200092338 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
Doing a wait while tracing MAY cause subsuquent traces to never complete

Signed off 08/19/88 in release A01.30

KPR #: D200092361 Product: 68010 EMUL 12.5M 300 64245S004 01.20

One-line description:
"bba unload" causes memory growth problems in emmulators.

Problem:
Detailed Listing for Defect Number LSDqf03817
Text:
bbaunload causes memory growth problems in emulators

The bbaunload library (/lsd/p2/cmd/bba/prod/unload) allocates memory and never frees it.

Please expand to 64244S004 and 64245S004 as well.

Bruce E

FIXED:
The new library fixes these problems as well as making the whole unload process much faster. It is fixed as of 03June88.

Signed off 08/19/88 in release A01.30

KPR #: D200092312 Product: 68020 ASSEMB 300 64870S004 01.00

Keywords: MACROS

One-line description:
>37 parameters in a MACRO heading and it silently does not expand.

Problem:
If more than 37 parameters are declared in the MACRO heading, it insidiously declines to expand without generating any warnings.

Temporary solution:
None.

KPR #: D200092189 Product: 68020C AXLS COMP 300 64903S004 01.10

Keywords: INCLUDE

One-line description:
Nested #INCLUDE's cause too many files to be open.

Problem:
The lister will issue the following message: 'Cannot open file' under the following circumstances:

- File A includes many files, which may in turn include many files, and the number of files included is larger that 54, and file A is still open

- None of the include files generate assembly code. (In other words, include files that only define variables, will cause this problem).

ODE TRAILER LINE

Temporary solution:
Change include files to include less that 54 files, as they branch out from original include, or put in some dummy code to cause lister to close files.

KPR #: D200093021 Product: 68020C AXLS COMP 300 64903S004 01.10

One-line description:
Cpp looks in the wrong directory for local include files.

Problem:
When searching for relative local include files (those specified with quotes
which don't begin with a '/') cpp always searches relative to the directory
the original C source file is in. Cpp should be searching relative to the
directory that the file doing the including is in. In the following example,
cpp won't find t1.h when it should.

Example:

File t.c contains:
```
#include "temp/t.h"
```

File temp/t.h contains:
```
#include "t1.h"
```

File temp/t1.h contains:
```
int i;
```
When searching for relative local include files (those specified with quotes which don't begin with a '/') cpp always searches relative to the directory the original C source file is in. cpp should be searching relative to the directory that the file doing the including is in. In the following example, cpp won't fine t1.h when it should.

Example:

KPR #: D200093021 **CONTINUED**

File t.c contains:
```
#include "temp/t.h"
```

File temp/t.h contains:
```
#include "t1.h"
```

File temp/t1.h contains:
```
int i;
```

Temporary solution:
Use full path names.

KPR #: D200093047 Product: 68020C AXLS COMP 300 64903S004 01.10

One-line description:
Cpp requires white space after #define macro name.

Problem:
The following case is currently errored by cpp for having an illegal macro name.

```
#define abc/def 1
```

The ANSI draft doesn't indicate an error in this case. The result should be a macro "abc" defined to be "/def 1".
The following case is currently errored by cpp for having an illegal macro name.

```
#define abc/def 1
```

The ANSI draft doesn't indicate an error in this case. The result should be a macro "abc" defined to be "/def 1".

Temporary solution:
None.

KPR #: 5000294207 Product: 6805/9 ASSEMB 64844 01.40

Keywords: CODE GENERATOR PROBLEM ON 9000/S300

One-line description:
BRSET range not checked.

Problem:

```
                        1"6805"
              <0001>    2RSW          EQU        1
                        4             ORG        0FFH
     00FF               5FLSW0        RMB        1
     0100               6FLSW1        RMB        1
                        7             ORG        250H
     0250 02FF 05       8             BRSET      RSW,FLSW0,C1011
(1)->0253 0200 02       9             BRSET      RSW,FLSW1,C1011
     0256 A6 01         10            LDA        #1
     0258 9D            11            NOP
  Errors= 0
```

Assembler should output error on (1).

Temporary solution:
None.

KPR #: D200093336 Product: 6805/9 ASSEMB 64844 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.90

KPR #: D200092916 Product: 6809 C 64822 01.80

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A01.90

KPR #: D200093575 Product: 6809 C 64822 01.80

One-line description:
Switch statement using unsigned int values 0 and 0xFFFF creates error

Problem:
Using a switch statement where the expression is an unsigned int, and the case values include small numbers and very large numbers, may create bad object code.

It appears that the compiler is attempting to decide whether to generate a jump table for the switching instead of individual case tests, which would be better for this example.

The code generated will not be able to jump to the "large", although apparently small (0xffff may look like -1) actual values.

For the following example, the generated code will not correctly jump to the case 0xffff statement.

```
main()
{
   unsigned I;
   I = 0xffff;
   switch( I )
   {
      case 0xfffd:
           break;
      case 0x0000:
           break;
      case 0x0001:
           break;
      case 0xffff: break;
      default :
           break;
```

KPR #: D200093575 **CONTINUED**

```
   }
}
```

Temporary solution:
The workaround for this switch statement would be to test the cases with individual if statements.

```
   if (I == 0xfffd) ...
    elseif (I == 0xffff) ...
    elseif (I == 0) ...
    elseif (I == 1) ...
```

KPR #: D200092825 Product: 6809 PASCAL 64813 01.60

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A01.70

KPR #: D200092957 Product: 6809 PASCAL 64813 01.60

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Signed off 08/31/88 in release A01.70

KPR #: D200093468 Product: 6809 PASCAL 64813 01.70

One-line description:
Type casting the ADDR function to SET for masking may cause an error.

Problem:
Expressions which try to perform masking operations on addresses using the ADDR function type cast to set may cause error.

Expressions in the form:

```
Byte := BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

will geneate incorrect code.

The context of the pascal expression is clear that the AND operation is desired. The compiler generates a call to unsigned integer multiply instead of generating an AND instruction.

KPR #: D200093468 **CONTINUED**

HERE is an expanded example:

```
 "PASCAL"
 "68000"
 PROGRAM Error;
 $EXTENSIONS$
 TYPE
     BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
     SET_OF_BITS = SET OF BITS;
 VAR
     S : SET_OF_BITS;
     Byte1,Byte2: BYTE;
     I : SIGNED_16;
 PROCEDURE BadADDRsetMASK;
  BEGIN
   Byte1:=BYTE(
   (SET_OF_BITS(ADDR(I))*SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]))
               ^505
         LDD  #DPTEST110+00004H
         LDX  #000FFH
         LBSR Zuintmul                  <----- Should be AND operation
         STB  DPTEST110+00002H
   END;
```

Temporary solution:
The workaround for this defect is to separate the use of the ADDR function from the actual MASKING expression.

Expressions in the form:

```
    Byte = BYTE( SET_OF_BITS( ADDR(variable) ) * SET_MASK );
```

could be rewritten:

```
    TempADDR := ADDR(variable);
    Byte = BYTE( SET_OF_BITS(TempADDR) * SET_MASK );
```

KPR #: D200093526   Product: 6809 PASCAL   64813   01.70

One-line description:
Large Sets may produce invalid results for elements outside set range

Problem:
The set inclusion operation may test undefined bit when the element being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

```
TYPE
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
```

KPR #: D200093526 **CONTINUED**

```
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {    'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,         }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```

Temporary solution:
Detailed Listing for Defect Number LSDqf04487
Text:
Large sets may produce invalid results for elements outside set range

....

The set inclusion operation may test undefined bit when the element being tested is outside the defined set range.

Normally it is expected that Pascal will produce a FALSE result for any element outside the defined boundaries of a defined set.

The following source code illustrates the problem.

```
TYPE
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {    'A' can NEVER be in the set DIGIT!}
    THEN   {...}      { Branch should always be FALSE,         }
    ELSE   {...}      {  But the result is due to invalid bit test}
END.
```

WORKAROUND:

The workaround for this defect is to separate the use of the the full 256 bit set implementation.

Instead of defining the large set as:

```
 DIG = SET OF '0'..'9';
```

It could be rewritten:

```
 digch = SET OF CHAR;
```

With the sets now using a full 256 bits, all bits will be set and tested properly.

TYPE

KPR #: D200093526 **CONTINUED**

```
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
VAR
   DIGIT : DIG;
BEGIN
  DIGIT:= DIG['1','3','5']
   IF 'A' IN DIGIT    {   'A' can NEVER be in the set DIGIT!}
    THEN   {...}     { Branch should always be FALSE,        }
    ELSE   {...}     {  But the result is due to invalid bit test}
END.
```

KPR #: D200092445 Product: 80186 EMUL FW 64764 00.01

One-line description:
Regnumarray[0] is not being allocated

KPR #: D200093419 Product: 80186 EMUL DOS 64764S006 01.01

One-line description:
Invalid COM port in 64700tab file

Problem:
There is an invalid COM port in the 64700ta b file. The second COM port in the 64700tab file should be COM2 instead of COM1.

Temporary solution:
There is no workaround available.

Signed off 08/31/88 in release A01.02

KPR #: 5000286591 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
80186 DISFUNCTION WHEN MONITOR NOT LOADED ON 64000-UX

Problem:
The 80186 emulator may not function properly if the MONITOR program is not loaded. For example, a program does a JMP FAR PTR, sets up a valid stack, and does an INT 0. The vector has been loaded to point to 1234H, where a IRET is located. When the processor reads locations 0 and 2, the values 006EH and 002EH are read respectively, instead of the 1234H, and 0 that shoudl be read. If the monitor program is loaded before this sample program, the code executes as expected. NOTE THAT NO MONITOR CODE IS RUN, JUST LOADED.

Signed off 08/31/88 in release A01.20

KPR #: D200093427 Product: 80188 EMUL DOS 64765S006 01.00

One-line description:
Invalid COM port in 64700tab file

Problem:
There is an invalid COM port in the 64700ta b file. The second COM port in the 64700tab file should be COM2 instead of COM1.

Temporary solution:
There is no workaround available.

Signed off 08/31/88 in release A01.02

KPR #: D200093252 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
NO warning message if parts of the monitor are in target memory

Problem:
When the monitor is loaded, only the exact address of a symbol is checked to make sure it is in emulation memory. For example, the transfer buffer can start in emulation memory, but extend into guarded memory, and no warning will be issued. Display/modify target memory will not work, but it is not obvious why it fails.

Signed off 08/31/88 in release A01.20

KPR #: D200092734 Product: 80286B ASSEMB 64859 01.40

One-line description:
MODULE pseudo generates random relocation type

Problem:
MODULE pseudo-op in 80286 extensions does not set the TYPE variabl to a known value prior to generating code and so may yield different checksums when running through regression tests.

This results in extra work when running tests because the checksums may not be the same as previously. THIS DOES NOT AFFECT THE RESULTANT EXECUTABLE CODE. This will be modified prior to the next release to make running regression tests easier.

KPR #: D200093732 Product: 80286B ASSEMB 64859 01.40

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Signed off 08/31/88 in release A01.40

KPR #: D200093351 Product: 8048 ASSEMB 64846 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.80

KPR #: D200092098 Product: 8051 ASSEMB 64855 01.08

One-line description:
Cross reference goes into endless loop on macro reference.

KPR #: D200093385 Product: 8051 ASSEMB 64855 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.90

KPR #: D200093278 Product: 8080/5 ASSEMB 64840 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.80

KPR #: D200093005 Product: 8085 B PASCAL 64825 01.90

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Signed off 08/31/88 in release A02.00

KPR #: D200093641 Product: 8085 B PASCAL 64825 01.90

One-line description:
Type casting the ADDR function to SET causes error #1006 on the VAX

Problem:
Type casting the ADDR function to type SET causes 1006 error on VAX.

Temporary solution:
Break up the expression by isolating ADDR:

```
TempADDR := ADDR(variable);
Byte := BYTE(SET_OF_BITS(TempADDR)*SET_MASK);
```

KPR #: D200092932 Product: 8085 C 64826 02.10

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.20

KPR #: 5000398396 Product: 8085 EMULATION 64203 01.07

One-line description:
64203A (8085) MEMORY MAPPING PROBLEMS

Duplicate Service Requests: 5000404988 5000298398

KPR #: D200093104 Product: 8085 EMULATION 64203 01.00

One-line description:
Config memory map is corrupted if examined during modify config

Problem:
After setting up a memory map in configuration for the PI 8085 emulator, if you modify configuration and examine that memory map it becomes corrupted in various and sundry ways: all but one entries are deleted, addresses become masked, addresses become changed.

Signed off 08/19/88 in release A01.08

KPR #: 1650063636 Product: 8085 EMULATION 300 64203S004 01.03

One-line description:
64000-UX 8085 EMULATION SW - BLOCK BOUNDARY PROBLEM

Problem:
WHEN READING A CONTINOUS TARGET MEMORY BLOCK ABOVE ADDRESS 8000H
THE TRANSITION IF THE FIRST 256 BYTES BOUNDARY IS NOT REFLECTED IN SETTING THE APPROPIATE BIT IN THE ADDRESS BUS TO A "1". INSTEAD THE BIT REMAINS 0, MEANING THE FIRST 256 BYTE BLOCK IS READ AGAIN.
FOR INSTANCE, IF YOU DO A "DISPLAY MEMORY 8000H" AND STEP THROUGH MEMORY WITH THE "NEXT" KEY, YOU WILL SEE THE ERROR WHEN THE ADDRESS 8100 IS CROSSED. A "DISPLAY MEMORY 8100", HOWEVER, WILL YIELD THE CORRECT MEMORY CONTENTS.
THE PROBLEM IS NOT THAT SEVERE WHEN YOU ARE JUST AIMING AT DISPLAYING MEMORY CONTENTS, BUT UNFORTUNATELY THE SAME EFFECT HAPPENS WHEN DOING A "STORE MEMORY TO <FILE>" COMMAND.

KPR #: D200093112 Product: 8085 EMULATION 300 64203S004 01.30

One-line description:
Display target mem shows incorrect data when crossing 256 byte boundary.

Problem:
Display target memory does not increment the upper byte of the address when reading target memory so incorrect data is displayed whenever a 256 byte boundry is crossed. This boundry is only crossed for display (or copy) target memory commands which do not begin on a block boundry.

For example "display memory 100h thru 2ffh" will show the correct data, but "display memory 180h thru 27fh" will correctly show the data from 180h thru 1ffh and then incorrectly show the data from memory 100h thru 17fh for the address range 200h thru 27fh.

Signed off 08/19/88 in release A01.40

KPR #: D200093237 Product: 8086 DQ EMUL 300 64220S004 01.20

One-line description:
NO warning message if parts of the monitor are in target memory

Problem:
When the monitor is loaded, only the exact address of a symbol is checked to make sure it is in emulation memory. For example, the transfer buffer can start in emulation memory, but extend into guarded memory, and no warning will be issued. Display/modify target memory will not work, but it is not obvious why it fails.

Signed off 08/31/88 in release A01.30

KPR #: D200092114 Product: 8086/8 ASSEMB 64853 02.70

One-line description:
JMP immediate instructions do not work on the 8086 Assembler

Signed off 08/31/88 in release A02.80

KPR #: D200092726 Product: 8086/8 ASSEMB 64853 02.70

One-line description:
MODULE pseudo generates random relocation type

Problem:
MODULE pseudo-op in 80286 extensions does not set the TYPE variabl to a known value prior to generating code and so may yield different checksums when running through regression tests.

This results in extra work when running tests because the checksums may not be the same as previously. THIS DOES NOT AFFECT THE RESULTANT EXECUTABLE CODE. This will be modified prior to the next release to make running regression tests easier.

Signed off 08/31/88 in release A02.80

KPR #: 5000296947 Product: 8086/8 C 64818 03.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Unsigned Short with bit field aligned on word boundary.

Problem:
The given code does align the unsigned short on a word boundary.
The stack when "unsigned short aa:4, bb:4" is used looks
like the following:

```
   0011 0000  0000 0000    <---- vec[2]
   0001 0010  0000 0000    <---- vec[1]
   0000 0000  0000 0000
```

While the stack after using "unsigned short aa, bb" with no bit
field looks like:

```
   0000 0011  0000 0000    <---- vec[2]
   0000 0001  0000 0010    <---- vec[1]
   0000 0000  0000 0000
```

It seems that the second representation is correct since the bit
fields are defined to be unsigned short (which should be 8 bits).

Temporary solution:
It seems defining the variables of type "unsigned short aa, bb" is
sufficient as is "unsigned short aa:8, bb:8" . Both of these will
align the variables on byte boundaries.

KPR #: 5000297754 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S500

One-line description:
Call to function using LONGS uses wrong segment.

Problem:
The compiler does not use correct segment. The DS segment is used
instead of SS segment. The examples is as follows.

```
"C"
"80188"
main()
{
 long c;
 long sub();
 sub(c);
   /*PUSH    SS:[BP-00004H]
     PUSH    SS:[BP-00006H]
     LEA     BA,SS:[BP-0000AH]     <---- SS is used here.
     PUSH    BX
     CALL    NEAR PTR sub
     ADD     SP,#+00006H  */
}
 long  sub(c);
```

KPR #: 5000297754 **CONTINUED**

```
 long  c;
 {
   return (c);
     /*PUSH    SS:[BP+00008H]
       PUSH    SS:[BP+00006H]
       POP     SS:[BP-00006H]
       POP     SS:[BP-00004H]
       JMP     NEAR PTR sub03_0   */
 }
/* sub03_0
       PUSH    SS:[BP-00004H]
       PUSH    SS:[BP-00006H]
       MOV     BX,SS:WORD PTR [BP+00004H]
       POP     DS:[BX]                         <-----| DS is used here.
       POP     DS:[BX+00002H]                  <-----| SS should be used.
 */
```

Duplicate Service Requests: 5000397752

KPR #: 5000402214 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300

One-line description:
Wrong code generated for structure in while loop.

Problem:
Compiler generates incorrect codes to pointer operation when
it is in 'while' loop.

```
 EXAMPLE
      "C"
      "8086"
      struct {
             unsigned char a;
             unsigned char b;
             unsigned char c;
             }ggg[10];
      unsigned char   i,j,k;
      main()
      {
             while (i <= k){
                          j=ggg[i].b;   <--- incorrect codes generated
                          }
      }
```

The above code does indeed generate incorrect code:

```
   while (i<=k) {
        j=ggg[i].b;
   }
        .
        .
        .
        MOV   AL,#+00003H
        MUL   AL            <---- Should MUL with AL and I
        MOV   BX,AX
```

KPR #: 5000402214 **CONTINUED**

```
        MOV    AL,DS:BYTE PTR Dstatic[BX+00001H]
        MOV    DS:BYTE PTR Dstatic+0001FH,AL
         .
         .
         .
```

If ggg[i].b is repeated, the correct code is generated.

KPR #: D200092080 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300

One-line description:
Long arithmetic expression generates incorrect code.

Problem:
The following expression generates incorrect code:

```
"C"
"8086"

main()
{
  int A, B, C, D, E;

  E = (A - (B/2)) - ((C+D) /2);
}
-----------------------------------------------------
     generates:
        MOV     AX,SS:WORD PTR [BP-00008H]     <-- B
        CWD
        MOV     CX,#+00002H
        IDIV    CX                             <-- B/2
        MOV     DX,SS:WORD PTR [BP-0000AH]
        SUB     DX,AX                          <-- (A-(B/2))
        MOV     AX,SS:WORD PTR [BP-00006H]     <-- C
        ADD     AX,SS:WORD PTR [BP-00004H]     <-- C+D
        MOV     AX,DX                   <---- overwrites (C+D)
        CWD
```

Temporary solution:
The temporary fix is to place sections of the long expression into temporary variables, then evaluate:

```
F = A - (B/2);
E = (C+D) /2;
E = F - E;
```

KPR #: D200092874 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A03.80

KPR #: D200093054 Product: 8086/8 C 64818 03.70

One-line description:
ES reg used instead of SS when assign string to structure

Problem:
NOTE: This problem was reported as fixed on SR#5000195628 in revision 3.70. The following was generated on 3.70 and thus is being re-submitted as a bug.

This is also a HOTSITE !!!! Epic # 1266

8086 C produces wrong code for assigning a character array to a complex data structure. Example Prog:

```
"C"
"8086"
$OPTIMIZE OFF$
$FIXED_PARAMETERS ON,EXTENSIONS ON,FAR_LIBRARIES ON$
$FAR_PROC ON,POINTER_SIZE 32,SEPARATE_CONST OFF,RECURSIVE ON$
$FAR_EXTVARS ON$

struct fibtab {
char name[20];
char typ;
char att;
int first;
int max;
int last;
int byte,date,use,reserve; };

char directory[64];
```

KPR #: D200093054 **CONTINUED**

```
struct dir {
int maxanz;
struct fibtab fib[1];
int link;
};

main()
{
 int i,wert;
 char *string1,*string2,zeil[10];
 struct dir *zei2;

 zei2 = (struct dir *) directory;

 for (i=0;i<10;i++)
     zeil[i] = '\0' ;

 for (i=0;i<4;i++)
     zeil[i] = 'A' ;

 string1 = zeil ;
 string2 = zei2 ;
wert = 0;
i = 0;

while ((zei2->fib[wert].name[i] = zeil[i] ) != '\0' )
                 i++ ;    /* this works fine ! */



while ((zei2->fib[wert].name[i] = string1[i] ) != '\0' )
                 i++ ;    /* this should do the same, it's just using */
                          /* a pointer instead of an array.*/
                          /* However the 8086 C compiler  */
                          /* produces bad code.  */
        .........
        MOV   DI,SS:WORD PTR [BP-0001AH]  <-- move i into DI
        PUSH  ES
        LES   BX,SS:DWORD PTR [BP-00016H] <-- offset of Ziel
        ADD   DI,BX                       <-- add offset and i
        MOV   AL,ES:BYTE PTR [DI]     <-- Problem! Offset into ES
        POP   ES
        ADD   SI,SS:WORD PTR [BP-0001AH]
        MOV   ES:BYTE PTR[SI],AL      <-- Moves value taken from
                                          ES and moves it into
                                          ES!

        .........

 }
```

Zeil is loaded into the SS segment. But, when string1 is used to access the values stored in SS, ES is used instead. This is evident because DI is assigned the correct offset into SS, but ES is used.

KPR #: D200093054 **CONTINUED**

Temporary solution:
Use an array name instead of a pointer.

KPR #: D200092833 Product: 8086/8 PASCAL 64814 03.50

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A03.60

KPR #: D200092965 Product: 8086/8 PASCAL 64814 03.50

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Signed off 08/31/88 in release A03.60

KPR #: D200093476 Product: 8086/8 PASCAL 64814 03.50

One-line description:
Boolean Index into array generates bad code

Problem:
Detailed Listing for Defect Number LSDqf04464
Text:
  boolean index into array generates bad code

```
...........................................................
$EXTENSIONS$

VAR
       bool   :  BOOLEAN;
       int    :  INTEGER;
       arr3 :  ARRAY[BOOLEAN,1..4,BOOLEAN] OF BYTE;
{ ====== END of Declarations for LSD1a01274  ====== }

PROCEDURE LSD1a01247;
```

KPR #: D200093476 **CONTINUED**

```
      {Initialize array arr3}
   FOR int:= 1 TO 4 DO
    FOR bool:= TRUE DOWNTO FALSE DO
     BEGIN
      arr3[bool,int,FALSE]:= 2;
      arr3[bool,int,TRUE]:= -2;
     END;
   bool := FALSE;
                   { variable::constant::variable }
  IF  arr3[bool,2,bool] = 2  { arr3[FALSE,2,FALSE] = 2}
    THEN
       {OK}
    ELSE
       {  "** ERROR LSD1a01274 #16 in file TEST110:Plus_P **."  };

                   { constant::constant::variable }
  bool := TRUE;
  IF  arr3[FALSE,4,bool] = -2
    THEN
       {OK}
    ELSE
       {  "** ERROR LSD1a01274 #19 in file TEST110:Plus_P **."  };
END;
```

KPR #: D200093484 Product: 8086/8 PASCAL 64814 03.50

One-line description:
Test for set inclusion checks beyond the set boundary.

Problem:
Detailed Listing for Defect Number LSDqf04465
Text:
  test for set inclusion checks beyond the set boundary.

```
...........................................................
$EXTENSIONS$

TYPE
        {DIG :: Set only up to character '9'; 64 bit set takes 8 bytes }
   DIG = SET OF '0'..'9';
        {digch :: Set of character; 256 bit set takes 8 bytes }
   digch = SET OF CHAR;

   Array256= ARRAY [0..255] OF BYTE;
VAR
   DIGIT : DIG;
   PADDING: Array256;
   DA : STRING;
{------------------------Possible FIX & test ------------}
   digitset : digch;
   ch : CHAR;

{ ====== BEGIN Test Procedure for LSD1a00270  ====== }
 PROCEDURE LSD1a00270;
  VAR index: INTEGER;
```

KPR #: D200093484 **CONTINUED**

```
  BEGIN
{------*Problems occur due to testing unrelated/unitialized values ---}
{           OUTSIDE of the legal boundaries for the declared SET.      }
{        In this case all the letters have values > '9' and should     }
{        never be tested and should always fail.                       }
{       It appears that this overflow will be detected if $RANGE$ is   }
{       on, but it is entirely ignored.                                }
{  DEMONSTRATE DEFECT by filling the PADDING array                     }
{          with 1's to make sure NO elements erroneously identified!   }


   DIGIT := DIG['0','1','2','3','4','5','6','7','8','9'];
   DA := 'A09z';

   FOR index:= 0 TO 255 DO
       PADDING[index]:= 0FFH;

   IF DA[1] IN DIGIT            { DA[1]='A' is NOT in DIGIT set }
    THEN
      {  "** ERROR LSD1a00270 #9 in file TEST111:Plus_P **."  }
  END;
```

KPR #: D200093518   Product: 8086/8 PASCAL   64814   01.90

One-line description:
Error 1006 for complex statement using MOD operator

Problem:
ERROR 1006 WHEN USING TYPE CONVERSION WITH MODULO OPERATION.

THE FOLLOWING SAMPLE PROGRAM WILL PRODUCE AN ERROR 1006 :

```
"PASCAL"
"8086"

$EXTENSIONS$

PROGRAM ERR1006 ;

VAR I : INTEGER ;
    B : BYTE ;
BEGIN
  B := BYTE((SIGNED_16(ADDR(I))) MOD 100H)
END.
```

Temporary solution:
Break this statement into two separate statements as follows.

```
VAR I,temp: INTEGER;
    B:  BYTE;

   BEGIN
      temp := SIGNED_16(ADDR(I));
      B := BYTE (I MOD 100H);
```

KPR #: D200093518 **CONTINUED**

```
   END.
```

KPR #: D200093245 Product: 8088 DQ EMUL 300 64221S004 01.10

One-line description:
NO warning message if parts of the monitor are in target memory

Problem:
When the monitor is loaded, only the exact address of a symbol is checked to make sure it is in emulation memory. For example, the transfer buffer can start in emulation memory, but extend into guarded memory, and no warning will be issued. Display/modify target memory will not work, but it is not obvious why it fails.

Signed off 08/31/88 in release A01.20

KPR #: D200093740 Product: 8096 ASSEMB 64860 01.70

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Signed off 08/31/88 in release A01.40

KPR #: D200093765 Product: HI SPD RS422 INTF 64037 00.01

One-line description:
The product does not generate a proper XON/XOFF, in "handshake" mode.

Problem:
When operated in software handshake mode, the product can generate a very earily XOFF request. Although this does not prevent the product from working, it does slow it down a lot.

Signed off 09/01/88 in release A00.02

KPR #: D200093591 Product: HOST SOFTWARE / VAX 64882 02.40

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

KPR #: D200093625 Product: HOST SOFTWARE / VAX 64882 02.40

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request
with
the following options may be ignored:

```
        using any of the following options to transfer:

                -cl[h]

                -fhl

                -thl

        note the -l option in combination with any option requesting
        high speed link (-c or -h).
```

KPR #: D200093609 Product: HOST SOFTWARE / 300 64883 01.10

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

KPR #: D200093633 Product: HOST SOFTWARE / 300 64883 01.10

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

KPR #: D200093583 Product: HOST SOFTWARE / 500 64880 01.90

One-line description:
Cluster to cluster transfers have a strange err.msg if >47 files in list

Problem:
Cluster to cluster transfer requests on series 300 may have DISC DRIVER error messages appearing after the 47th file in a list file.
The problem may also appear in any file list transfer after the 47th file in the list.

KPR #: D200093617 Product: HOST SOFTWARE / 500 64880 01.90

One-line description:
Break or ^C may not abort a foreground transfer with a file list

Problem:
Attempting to <BREAK> or interrupt out of a foreground transfer request with
the following options may be ignored:

using any of the following options to transfer:

-cl[h]

-fhl

-thl

note the -l option in combination with any option requesting high speed link (-c or -h).

KPR #: D200090118 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
DOMAIN -SQRT ERROR generated when Data Flows become tangential to bubble

Problem:
When a process bubble in a data flow diagram is moved to where the data flow is tangential to it, the error:

    DOMAIN -SQRT ERROR

is generated from the HP-UX system. In other words, the teamwork window is scrolled up and the error message is generated in the new line. A control L refreshes the screen and no data is lost.

Temporary solution:
As there is no loss of data, and the screen can be repained, there is no temporary work around.

KPR #: D200093088 Product: NETWORK TRANSFER 300 64887S004 01.00

One-line description:
The transferII utility does not work using nft as the transport

Problem:
When using hostcopy or get64 and nft as the transport an errror message of hostcopy:untranslateable NFT response, more information: usage: [-p -s -d -L -r -P -F -B -A] fromnode#user#file tonode#user#file aborting.

KPR #: 5000170118 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Xref cannot be generate as an independent listing

Problem:
We want to list out only xref. But assembler do not generate xref and any other list, when using nolist and xref option. This occures only HP9000/500.

Signed off 08/31/88 in release A02.11

KPR #: 5000214189 Product: OPERATING SYSTEM 64100 02.10

Keywords: CODE GENERATOR

One-line description:
Condtl code not assembled if condtl statmnt is false and missing ENDIF

Problem:
If a conditional statement is missing the ENDIF, and evaluates false, the assembler does not flag it as an error, and does not assemble the program.
The following code will demonstrate this problem:

```
        "Z80"
  TEST: LD  A,[HL]
        NOP
        IF  0
        LD  B,128
  LOOP: DJNZ LOOP
        END
```

This does not assemble, and does not produce an error.

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A02.11

KPR #: 5000401349 Product: SOFTKEY EDITOR 300 64790S004 02.10

One-line description:
Status line does not change after file is written for the save command.

Problem:
The SK editor does not update the status line after it completes storing a file caused by a save command. Example:

- if you issue a "save command, the status line will read "writing /users/joe/file" until another key is pressed.

This does not effect the operation of the editor itself.

KPR #: 5000401372 Product: SOFTKEY EDITOR 300 64790S004 02.10

One-line description:
sk editor replace command does not work properly with anystring (*).

Problem:
The sk editor does not properly handle replace commands involving anystring (*) and a limiting range.
Example:
The current line contains the letters aa in columns one and two and the range is set to one. The command "replace ^* ^ with ^*!^" should change the string to "aa!", instead it changes the line to "aa!!....".

KPR #: D200093401 Product: TMS 320 ASSEMB 64858 00.00

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.80

KPR #: D200092619 Product: USER DEF ASSEMB 64851 00.70

One-line description:
COPY :asmb_sym to display behaves like disc_image on.

Problem:
Please contact Caren Johnson x5714 for the supporting software:

After assembling the given UDA code, the two given sample programs display the problem.
The output from

```
   copy file:asmb_sym to display
```

uses half disc_image and half normal output.

Example after assembling sample code, and using "copy file:asmb_sym to display" command:

```
    Record # 1   size = 121
    Asmb_sym record:
    8053  594D  5F50  4F52  5431  0000 .........
    5F50  4F52  5430  ..........................
    5433  0000  ............................         <-- DISC
    ............................................         IMAGE
    ............................................


    Record # 2   size = 102
    Asmb_sym record:
    SYM_PMGB     0000H Absolute      SYM_PMGA    0000H Absolute
    SYM_BSB2     0000H Absolute      SYM_IPS     0000H Absolute
    SYM_SIO      0000H Absolute      SYM_BSB1    0000H Absolute
    ..........................................................
    ..........................................................

                              ^
                              |
                              |
                            NORMAL
```

Temporary solution:
For some reason if the two statements:

```
  SYM_PORT10       SET      PORT10
  SYM_PORT11       SET      PORT11
```

are added to the sample programs the problem does not occur.

KPR #: D200093781 Product: USER DEF ASSEMB 64851 00.70

Keywords: CODE GENERATOR

One-line description:
Problem with parameter passing in macros

KPR #: D200093781 **CONTINUED**

Problem:
The following program causes the errors shown:

```
 "Z80"
 AAA:   MACRO   &A, &B

   AAA  abc, def  ghi
+  ;abc
+  ;def  ghi       The ghi is a comment here, not part of the parameter



 "Z80"
 AAA:   MACRO   &A,&B,&C
        ;&A
        ;&B
        ;&C

        AAA         {,+,}
        ;{
        ;=
        ;=,}        This is only a problem with the "}".  Other
                    special characters do not cause this problem.
```

Temporary solution:
There is no known work around at this time.

KPR #: D200092288 Product: USER DEF ASSEMB 300 64851S004 00.70

One-line description:
Jump to MACRO label causing expression type error.

Problem:
NOTE: This problem occurs with the QA version (Rev. 2.20) of software sent to Matsushita for hotsite #1487.

```
 "Z80"
 SMB     MACRO   &DATA
         LD      A,&DATA
         MEND

         ORG     0
 CMP01   SMB     1
         JR      CMP01
                 ^
                 | EXPRESSION error here.
```

Assembling with TRACE 3 inserted into the code shows that TYPE does indeed get set to 80. It seems this is then flagged as an ET-Expression Type error.

Signed off 08/31/88 in release A02.20

KPR #: 5000402701 Product: USER DEF ASSEMB VAX 64851S003 02.10

One-line description:
NO LOAD files are not handled correctly.

Problem:
NOTE: This problem was reported fixed with revision 1.60.
Please refer to SR's #5000143370, and #D200063230.

The SR is being re-submitted because the problem still exists. Compile two files and link as follows:

```
segment
object files   FILE1.R
library files  HP$DISK:[HP64000.CLIB.NS8086]lib.R
load addresses   10H,20H,30H
segment
object files   (FILE2.R)
library files
load addresses   40H,50H,60H
absolute file name  FILE.X
```

Only FILE2.R is no-loaded. However, the output listing shows both FILE2.R and all the lib.R file are also no-loaded. The lib.R library files should not be no-loaded.

NOTE: When the above is attempted on the 9000/300 like follows:

```
segment
object files  FILE1.R
library files  /usr/hp64000/lib/clib/ns8086/lib.R
load addresses 10H,20H,30H
segment
object files (FILE2.R)
library files
load addresses 40H,50H,60H
absolute file name  FILE.X
```

FILE2.R should be no-loaded. But, no files in the output listing are marked as no-load.

Signed off 08/31/88 in release A02.20

KPR #: 5000266684 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
UDE CLK SOURCE DIFFERENT BETWEEN 64000/64000-UX CONFIGURATION FILES

Problem:
UDE software accesses unexpected clock source from configuration.
After transfering UDE configuration file developed on 64100 to 64000UX, we must change pod clock selection in the configuration file as follows.

```
64100                      64000UX
  1 -external clock           0 -external clock
  0 -internal clock           1 -internal clock
```

If not so, on 64000UX, when we select 0 for internal clock source, UDE accesses external clock.

KPR #: 5000296855 Product: USER DEF EMULATION 64274 01.05

One-line description:
64000-UX UDE Using "HOME" key causes trace list problems

Problem:
I'm using UDE which is made under hp64000 (PI) system on 64000ux (9000/350 rev 5.5). While I'm doing 'display trace' , I enter shift+'home-key' then trace list is not displayed correctly and the display gets confused. And when I enter 'home-key' as soon as I use 'display trace' then the mnemonic of line 1 is deleted.

KPR #: 5000296921 Product: USER INTERFACE 300 64808S004 02.10

One-line description:
Pmon flags legitimate option for lnk (for 64859) as syntax error

Problem:
the 64859 linker for the 80286B (protected mode) has to be invoked with a special option -b (not -h).
The "pmon" interface revision 2.10 however will flag this option as a syntax error.

Temporary solution:
Invoke linker using shell escape: !lnk -b

KPR #: D200093310 Product: Z80 ASSEMB 64842 01.12

Keywords: PROBLEM ON VAX

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Problem:
This Service Request has been entered to inform users of the product

THAT:

The *PRODUCT NUMBER has CHANGED on the VAX version of this product

```
    FROM (OLD Product Number)= 64xxxS003 < The real change being
                                         < the "S" changed to "M"
    TO   (NEW Product Number)= 64xxxM003 < in this Product Series
```

(The "xxx" in the above to be filled in with the Product Number against which this SR is entered... This text applies to many SR's and is generic in nature.)

The above event happend without a change to the REVISION CODES on the PRODUCT.

This event happend on the revision code that was used to sign off this Service Request.

Signed off 08/23/88 in release A01.90

KPR #: D200092924 Product: Z80/NSC800 C 64824 02.10

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.20

KPR #: D200092866 Product: Z80/NSC800PASCAL 64823 01.90

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.00

KPR #: D200092999 Product: Z80/NSC800PASCAL 64823 01.90

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Signed off 08/31/88 in release A02.00

KPR #: D200092890 Product: Z8000 C 64820 02.10

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.20

KPR #: D200092858 Product: Z8000 PASCAL 64816 01.90

Keywords: PROBLEM ON 9000/S300 PROBLEM ON 9000/S500 PROBLEM ON VAX
NO PROBLEM/ PISCES I

One-line description:
Compilers do not list complete information about source file path name.

Problem:
All HOSTED (HPUX and VAX) Pascal & C compilers do not list the full path name of the source file being compiled in the output listing.

The hosted compilers only list the last 15 characters of the source file (and path) in the output listing.

Since the full file path name is included in the relocatable file name record, it should also appear in the listing file. This would make it much easier to determine the true source file which produced the output.

Signed off 08/31/88 in release A02.00

KPR #: D200092981 Product: Z8000 PASCAL 64816 01.90

Keywords: PROBLEM ON VAX

One-line description:
VAX Pascal xref prints many garbage characters of first line xref list

Problem:
All VAX Pascal compilers print a long line of garbage characters as the first line of the cross reference listing.

These extraneous characters cause problems when trying to print or edit the cross reference listing file.

Signed off 08/31/88 in release A02.00

KPR #: D200093724 Product: Z8001/2 ASSEMB 64854 01.80

One-line description:
PRODUCT # CHANGE on the VAX From= 64xxxS003 To=64xxxM003

Signed off 08/31/88 in release A01.00

KPR #: D200082057 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200085878 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                            ; end locks the emulation session
3. <system name> <module name>    ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086264 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088229 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090688 Product: 6301V EMULATION 300 64206S004 00.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200088088 Product: 6301V/03R EMUL 64206 01.01

One-line description:
6301V/03R module cannot be accessed with HP-UX 6.01

Problem:
In A 131 bundled system the 6301V/03R fails to operate with HP-UX 6.01. Error message reads: 6301-00c: could not open module - check HPIB and power to card cage.

Temporary solution:
The current workaround requires that the emulation software be put in the software debug mode by creating a file:

```
usr/hp64000/log/adb.X.0
```

where "X" represents the select code for the HPIB interface.

Then standard error must be redirected to /dev/null when the emulator is invoked.

KPR #: D200081596 Product: 6301Y/03Y EMUL 64208 01.00

One-line description:
Emulator can't work when external clock is selected and E clock = 160khz

Problem:
Emulator can not work when external clock is selected and system clock rate (E clock) is 160 KHz.
Same configuration with 64100A (Pisces I) can perform correct emulation.

KPR #: D200067322 Product: 64 HP-UX VMS 8096 AM 64860-90901 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200087445 Product: 64000 UX GENERIC 64003S004 01.00

One-line description:
Option_test does not support set (environmental variable) command

Problem:
Detailed Listing for Defect Number LSDqf02728
Text:
  option_test does not support set (environmental variable) command

In keeping with the emulation upgrade, option_test should suport the set command, which sets an environmental variable...

This is being added now, in the files layer0/grammar/scanner.c and layre0/grammar/optgram.y.

- Bruce E

Temporary solution:
no temporary solution at this time.

KPR #: D200086900 Product: 64000-UX OP-ENV 300 64801S004 01.80

One-line description:
You can not enter ICC for IMB stimulus when in "option_test"

Problem:
Detailed Listing for Defect Number LSDqf02517
Text:
cannot enter ICC for IMB stimulus in option_test

In option_test:
If somebody tests an analyzer (or other board which wants an IMB-compatible board), the ICC key does not show up on the softkeys, nor can it be entered manually.

Signed off 03/25/88 in release A02.00

KPR #: D200090472 Product: 64000-UX OP-ENV 300 64801S004 01.80

One-line description:
EDB problems with scoping of locals from new com/asm/linker

Problem:
Detailed Listing for Defect Number LSDqf03783
Text:
edb problems with scoping of locals from new comp/asm/linker.

Symbols that are expected to be scoped relative to function are sometimes included with the file local symbols, causing unexpected and varying results in inverse-assembly of tracelists as well as confusion in command entry.

Several of the function entry link symbols are not scoped to the function, while others are. This seems to be random depending on the file being used. Everything "appears" to be ok from the language tools and in the .A files.

Temporary solution:
There is no workaround available.

KPR #: D200091280 Product: 64000-UX OP-ENV 300 64801S004 01.80

One-line description:
"makecdf(1)" error may occur in msinit on a new discless cluster

Problem:
When "msinit -s" is run on a newly created discless cluster, the user may see the following message:

```
        makecdf: /usr/hp64000/etc: Permission denied
```

Temporary solution:
When a new discless cluster is brought up, "msinit -s" should be executed by the super-user first before any other users try to to execute it. Subsequent executions of "msinit [-s]" by non-super-users should work without failure.

KPR #: 5000240580 Product: 64HP-UXVMS8086/8 A M 64853-90908 02.03

Keywords: MANUAL

One-line description:
Need Manual change to explain AC and PH phase errors.

Duplicate Service Requests: 5000240960

KPR #: D200067223 Product: 64HP-UXVMS8086/8 A M 64853-90908 02.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 5000219865 Product: 6800 C 64821 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Libraries cause write to ROM

Problem:
The PUSHX:D6800 library will cause a write to ROM if the library is linked in a ROM section. The library declares a local variable SAVEX using the RMB directive in an area defined as PROG. It then writes to that variable. This variable should be in a DATA section. An example of code that produces this problem:

```
"C"
"6800"
$RECURSIVE-$
int  func1(i)
int  i;
{    return(i);
}
main()
{     int i;
      i = func1(i);
}
```

Temporary solution:
Rex Mayne - Atlanta Response Center

See submitter text

WORKAROUND

Use the slib libraries, or link the dlib libraries in a RAM section.
Rex Mayne - Atlanta Response Center

See submitter text

WORKAROUND

Use the slib libraries, or link the dlib libraries in a RAM section.

KPR #: 5000293779 Product: 6800 C 64821 02.10

One-line description:
Switch statement causes infinite loop.

Problem:
Switch statement causes infinite loop.

Example code:

```
"C"
"6800"
#$FIXED_PARAMETERS ON$
#$INIT_ZEROS OFF$
#$RECURSIVE OFF$
#$SEPERATE ON$
#$UPPER_KEYS ON$
```

KPR #: 5000293779 **CONTINUED**

```
#$SHORT_ARITH ON$
#$WARN OFF$

#define ZLT_ALOCK_LI  0x0001
#define ZLT_POLL_LI   0x0002
#define ZLT_PLU_LI    0x0004
#define ZLT_PARAM_LI  0x0008
#define ZLT_MBT_LI    0x0010
#define ZLT_ACSH_LI   0x0020
#define ZLT_ICSH_LI   0x0040
#define ZLT_SALP_LI   0x0080
#define ZLT_DEPT_LI   0x0100
#define ZLT_SURVEY_LI 0x4000
#define ZLT_CONFIG_LI 0x8000
#define ZDC_ACK_C     6

PROC1 () {

        UNSIGNED *TYP_UP;

        SWITCH (*TYP_UP & ~ZLT_ALOCK_LI)  {

             CASE ZLT_PARAM_LI:

             CASE ZLT_MBT_LI:

             CASE ZLT_ACSH_LI:

             CASE ZLT_ICSH_LI:

             CASE ZLT_SALP_LI:

             CASE ZLT_DEPT_LI:

             CASE ZLT_SURVEY_LI:

             CASE ZLT_CONFIG_LI:
                 *LAN_BTBUF_AC = 0x80;


             DEFAULT:       ;
            }
}
```

Temporary solution:
This only fails for this specific example. So, the only workaround is to simply not use this specific code.

Duplicate Service Requests: D200091348

KPR #: D200068197 Product: 6800 C 64821 01.06

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"

struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200069823 Product: 6800 C 64821 01.06

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */
```

KPR #: D200069823 **CONTINUED**

```
#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200074989 Product: 6800 C 64821 01.07

One-line description:
USE OF MANY FUNCTION CALLS WITH CONSTANT PARAMATERS MAY CAUSE ERR #1007

Problem:
C programs with many function calls with constant parameters may cause Pass 2 Error 1007 - expression too complicated.

The workaround is to use variables to pass parameters. Since the error is related to the compiler attempting to create logical temporaries in order to have these parameters passed as the proper size, the problems can be avoided by using variables instead of constants as parameters.

If only a few instances of constants being passed as parameters is encountered, the use of type casting the constants to int may be sufficient to allow the program to compile.

This is related to the known 6800 compiler limitation which can cause Pass 2 Error 1010 - Too many constants.

When passing constants as parameters, the 6800 code generator will eventually run into the 256 constant limit which produces the 1010 error. The use of variables is the only solution in this instance.

In most instances which produced the 1010 error in previous revisions of the C/6800 compiler, the 1007 error will now be produced. In some instances which previously produced no errors, the 1007 error may be produced.

The following program fragment when duplicated can cause the 1007 error:

```
"C"
"6800"
 extern funct();
 main()
```

KPR #: D200074989 **CONTINUED**

```
 {
   funct(1,2,3,4,5,6,7,8);
     /*  ...  */
   funct(1,2,3,4,5,6,7,8);
 }
```

The following program fragment illustrates the workaround solutions:

```
"C"
"6800"
 extern funct();
                /* Variables initialized to constants */
 int  one=1; int two=2; int three=3; int four=4;
 int five=5; int six=6; int seven=7; int eight=8;

 workaround()
 {

      /* Use functional type change of short int const to int */
   funct((int)1,(int)2,(int)3,(int)4,(int)5,(int)6,(int)7,(int)8);

      /* Use int variables with constant values */
   funct(one,two,three,four,five,six,seven,eight);

 }
```

Temporary solution:
See problem text.

KPR #: D200079624   Product: 6800 C   64821   01.07

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}
```

KPR #: D200079624 **CONTINUED**

```
memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
       }
   }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function
memory test. This will cause slightly more code to
be generated.

KPR #: D200081539   Product: 6800 C   64821   01.07

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself
as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
          ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;

  OR

cast the variable to an int:

if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

KPR #: D200081539 **CONTINUED**

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200085779   Product: 6800 C   64821   02.10

One-line description:
Posedecrement operator used on structure pointer may fail

Problem:
Text:
  Postdecrement operator used on structure pointer may fail

Problem: Use of "--" operator on variable after use in if expression may generate bad object code.

This occurrence is not easily duplicated. "If" statments comparing variables to variables should not cause the problem. The error appears to need a variable compared to a constant value to generate the defect.

This problem only occures on the VAX/VMS release of 64821S003(VAX 2.10). Problem does not exist on 64821A(HP64000 2.10), 64821S001(/500 2.10) or 64821S003(/300 2.10).

The following example illustrates the problem:

```
  "C"
  "6800"
  main() {
      struct s1 {
          struct s1 *msp;
          int *mp1,*mp2;
      }*p, sdef, sdef2;
      int *intp;

      if (p->mp2 != (int*)0) call_error();

      sdef.mp2--;
              CLRB                #should be LDAB Dmain+00007H
              CLRA                #should be LDAA Dmain+00006H
              SUBB #002H
              SBCA #000H
```

KPR #: D200085779 **CONTINUED**

```
              STAA Dmain+00006H
              STAB Dmain+00007H
  }
```

Temporary solution:
  The workaround solution is to write the expression with a temporary.

```
     intp = sdef.mp2;
     intp--;
     sdef.mp2= intp;
```

or write the statement as an explicit subtraction:

```
     sdef.mp2 = sdef.mp2 - 1;
```

Signed off 08/31/88 in release A02.20

KPR #: D200085787   Product: 6800 C   64821   02.10

One-line description:
Type cast of constant to (char *) in pointer expression error

Problem:
Text:
  Type cast of constant to (char *) in pointer expression error

Problem: Use of CHAR pointer type casting may cause error in pointer expressions.

```
"C"
"6800"
 char *st_ptr;
main(){
 st_ptr = "any_string" - (char *)1; /* Computes bad address */
/* WORKAROUND 1 */
 st_ptr = "any_string";
 st_ptr -= 1;              /* Correct decrement -1 */
/* WORKAROUND 2 */
 st_ptr = "any_string" - 1;     /* Correct decrement -1 */
}
```

EXPANDED CODE EXAMPLE:

```
  st_ptr = "any_string" - (char *)1; /* Computes bad address */
             LDX  #CONST_prog
             LDAA #0FEH          ;Should be #0FFH
             LDAB #0FBH          ;Should be #0FFH
             JSR  LEAX_D_X
             STX  Dstatic
 /*WORKAROUND*/
  st_ptr = "any_string";
```

KPR #: D200085787 **CONTINUED**

```
              LDX  #CONST_prog
              STX  Dstatic
  st_ptr -= 1;
              DEX
              STX  Dstatic
  /* or this */
  st_ptr = "any_string" - 1; /* Computes correct address */
              LDX  #CONST_prog
              DEX
              STX  Dstatic
?                            ^515 /* Ignore warning  code OK*/
 }
515: Warning: integer not pointer size
```

Temporary solution:

```
"C"
"6800"
 char *st_ptr;
main(){
 st_ptr = "any_string" - (char *)1; /* Computes bad address */



/* WORKAROUND 1 */
 st_ptr = "any_string";
 st_ptr -= 1;            /* Correct decrement -1 */
/* WORKAROUND 2 */
 st_ptr = "any_string" - 1;     /* Correct decrement -1 */


}
```

KPR #: D200085803 Product: 6800 C 64821 02.10

One-line description:
SHORT_ARITH OFF for some short experssions used as conditional branch

Problem:
Text:
SHORT_ARITH OFF for some short expressions used as conditional branch

Problem: With the SHORT_ARITH option OFF, the 6800 compiler does not execute full K&R C code correctly for certain mixed aritmetic operations when used in "if" expressions. Problems occur when 8-bit (short) aritmetic is used, rather than full expansion to 16 bit values to performa operations as in the standard K&R.

EXAMPLE:

```
"C"
"6800"
short s,ss;
main(){
```

KPR #: D200085803 **CONTINUED**

```
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */
 /* WORKAROUND */
 if ((int)s<<4); /*Result is 64*4=1024 which is <>0 should branch here*/
  else  ;
}
```

The 6800 C compiler computes mixed expressions correctly, as in assignment statements and parameter expressions.
This defect appears only when mixed expressions are used without assignment as conditional branching expressions.

This problem may be generated with other operators besides the "<<" as in the example, such as ">>" , "/" and "%

EXPANDED example:

```
  "C"
                               EXTERNAL entry
  "6800"
  short s,ss;
  main(){
          main
   $SHORT_ARITH OFF$
   if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
              LDAB Dstatic  ; This is correct ONLY $SHORT_ARITH OFF$
              ASLB
              ASLB
              ASLB
              ASLB
              BNE  main01_7
              JMP  main01_1
          main01_7
              JMP  main01_2
          main01_1
    else  ;  /* Code branches here, due to use of byte arithmetic. */
          main01_2
   /* WORKAROUND */

   if ((int)s<<4); /*Result is 64*4=1024 which <>0 should branch here*/
              LDAB Dstatic
              JSR  SEXtend
              JSR  TFR_DtoX
              LDAB #004H
              JSR  Zwshift
              JSR  TFR_DtoX
              CPX  #00000H
              BNE  main01_8
              JMP  main01_3
          main01_8
              JMP  main01_4
          main01_3
```

KPR #: D200085803 **CONTINUED**

```
    else  ;
          main01_4
```

Temporary solution:
EXAMPLE:

```
"C"
"6800"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */



 /* WORKAROUND */
 if ((int)s<<4); /*Result is 64*4=1024 which is <>0 should branch here*/
  else  ;
}
```

KPR #: D200055772 Product: 6800 C M 64821-90901 01.05

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

        for (cntr=1; cntr<4; cntr++)
          { tmp = func5(cntr);
        }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.07

KPR #: D200059980   Product: 6800 PASCAL   64811   01.09

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:
I am expanding this to all pascal compilers. The C compilers list the correct offset. $FAR ON$ only applies to the 68000 cross compiler. The other compilers exhibit the defect w/o any options on.

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200087189   Product: 6800 PASCAL   64811   01.90

One-line description:
Bad code generated fot ADDR of first record element used as a parameter.

Problem:
Bad code for ADDR of first element of record used as parameter.

Pascal is generating bad code if the parameter passed to a procedure is the address of the first element of a record, and that record is specified in a WITH statement.

The compiler is erroneously generating an indirect flag preceding the parameter specifier in the calling sequence.

KPR #: D200087189 **CONTINUED**

This problem exists on the all 64811 products before Rev 1.90.
This problem exists on the 64811A(HP64000Rev 1.90) and the 64811S003(VAX Rev 1.90).
It does not occur on the HPUX products 64811S001(/500 Rev 1.90) and 64811S004(/300 Rev 1.90)

Problem source program:

```
"PASCAL"
"6800"
 PROGRAM ParmPointerBug;
$EXTENSIONS ON$
{USE of ADDR function is an extension, also SHIFT & ROTATE}
$RECURSIVE ON$
 TYPE  PTR = ^INTEGER;

VAR  V: RECORD
          element_1:  INTEGER;
          element_2:  INTEGER;
        END;
PROCEDURE RECURproc (pointer: PTR); EXTERNAL;

BEGIN
  WITH V DO
    BEGIN
                          {bad code - addr passed with indirection}
      RECURproc (ADDR (element_1));
      RECURproc (ADDR (element_2)); {good code}
    END;
END.
```

Temporary solution:
No temporary solution at this time.

Signed off 08/31/88 in release A02.00

KPR #: D200087304   Product: 6800 PASCAL   64811   01.90

Keywords:	CODE GENERATOR	PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
	PROBLEM ON VAX	NOT ON 64100 SYSTEM

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example:

```
"8086"
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
LABEL  100;
BEGIN
100:
     GOTO   100;
```

KPR #: D200087304 **CONTINUED**

```
END;
PROCEDURE  TWO;      { pass 3 error - too many errors in pass 3 }
LABEL  100;          { is generated, without any indication as to }
BEGIN                { what the problem is                        }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get this error message, be aware that you may have duplicate labels in the program.

Signed off 08/31/88 in release A02.00

KPR #: 5000151050 Product: 6800-03 ASSM M 64841-90905 01.15

One-line description:
Mask pseudo works incorrectly in certain cases.

Problem:
The mask psuedo does not work in the following program.

```
"processor"

        MASK    0FFH
        LDX     #'AB'       ;'A' is masked with 0, and B is masked
                            ;with F.
```

It appears that the assembler is using the required leading zero as a mask value.

Temporary solution:
Specify a four-byte mask value.

```
"processor"

        MASK    0FFFFH
        LDX     #'AB'
```

KPR #: 5000221200 Product: 6800-03 ASSM M 64841-90905 01.15

Keywords: MANUAL

One-line description:
Support OIM, AIM, EIM, TIM

Problem:
This request was originally that we support the AIM, OIM, TIM, and EIM instructions. We do support those instructions, but, this fact is not documented in the manual. The manual should be updated to include this information.

Temporary solution:
No temporary solution.

KPR #: 5000117002 Product: 6800/2 ASSEMB 64841 01.13

One-line description:
Comments are listed in the xref table when not delimited by a ;

Problem:
Comments are listed in the cross reference table as labels when the comments are not delimited from the code with a semi-colon(;).

```
       "processor name"
                    opcode
                    opcode
                    opcode          This is a comment
                    opcode
```

"This", "is", "a", and "comment" will be listed in the cross reference table as labels.

Temporary solution:
To avoid having comments listed in the cross reference table, used a semi-colon (;) to delimit the comment from the code.

Duplicate Service Requests: D200065664 D200065854 D200065862

KPR #: 5000166983 Product: 6800/2 ASSEMB 64841 01.15

One-line description:
External MASKS are not handled properly by the assembler.

Problem:
The bit instructions (BSET, BCLR, BTST) do not handle externally defined masks properly.

```
"6301"

MASK            EQU             7
                EXTERN          MASK2


                BCLR            MASK    30
                BCLR            MASK2   30
                BCLR            7       30
                END
```

MASK2 is an external symbol, but, the assembler interprets it as a zero value therefore it generates a mask of FEH. The linker then adds the actual mask value and a legal range error is generated.

Temporary solution:
Use an include file to define the MASK (bit) values. This does not accomodate the linker XREF desires of the customer.

KPR #: 5000226563 Product: 6800/2 ASSEMB 64841 01.40

Keywords: PROBLEM ON 9000/S500

One-line description:
Xref table is not listing all symbol references.

Problem:
The following program when assembled with option XREF will demonstrate a problem. The problem is that the XREF will not list all symbol references.

```
"6301"

          ORG          80H
MAP       RMB          1
          ORG          0C000H
          AIM          80H  MAP
          OIM          80H  MAP
          TIM          80H  MAP
```

Temporary solution:
No temporary solution.

KPR #: 5000255752 Product: 6800/2 ASSEMB 64841 00.00

Keywords: PROBLEM ON 9000/S500

One-line description:
Very long file causes problems with xref listing on a 2563B

Temporary solution:
No temporary solution at this time.

KPR #: 5000273458 Product: 6800/2 ASSEMB 64841 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
6301 AIM instruction with ".NT." operator causes LR error.

Problem:
6301 AIM instruction with ".NT." operator causes LR error.

Example assemble list

```
                   1 "6301"
            <00E2> 2 RXFLG   EQU   0E2H
     0000 867F     3         LDAA  #.NT.80H
     0002 717FE2   4         AIM   #.NT.80H RXFLG
 ERROR-LR                               ^
     0005 717FE2   5         AIM   #07FH RXFLG
                   6         END
```

I talked to Dave Ritchie about this - he said it WAS a problem. However, I was unable to obtain a manual for the 6301 to verify if this is correct op-code syntax.

KPR #: 5000273458 **CONTINUED**

David Landoll

Temporary solution:
No workaround available.

KPR #: 5000273474 Product: 6800/2 ASSEMB 64841 01.10

One-line description:
Assembler allows the inst. "LDA A". "LDA A" isn't a valid instruction.

Problem:

```
"6800"
LAB0  EQU  0
LAB1  EQU  1
      LDA A LAB0
      LDA A LAB1
```

Assembler generates the xref of "LDA A LAB0(1)" on 64000, but not on 64000ux. If I change "LDA A" to "LDAA", it generates the xref on 64000ux ,too.

Temporary solution:
There is no workaround available.

KPR #: D200031088   Product: 6800/2 ASSEMB   500 64841S001   01.20

One-line description:
Assembler flagging out of range error when it should not.

Problem:
There is a descrepency on how out of range errors are handled. The below line will load the lower sixteen bits into register D (this seems appropiate):

```
            LDD          #10000000H
```

While the following line will flag an out of range error:

```
            LDAA         #10000000H
```

Temporary solution:
And the operand with 0FFH. This will force it to eight bits.

```
"6800"

        LDAA       (#10000000H).AN.0FFH
```

Signed off 08/25/86 in release 01.50

KPR #: D200048199   Product: 6800/2 ASSEMB   500 64841S001   01.30

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly. The following program demonstrates this problem:

```
       BUG           MACRO             &VAR
                     .IF &VAR .LE. 0 SUB&&&&
                     NOP
                     NOP
       SUB&&&&       NOP
                     NOP
                     MEND

                     BUG -3
                     BUG  1
                     BUG  0
                     END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 08/25/86 in release 01.50

KPR #: D200053298   Product: 6800/2 ASSEMB   500 64841S001   01.30

One-line description:
Macro def. including .IF, within a IF causes assembler to stop code gen.

Problem:
If you have a ".IF" in a macro definition and that macro definition is within a conditional assembly "IF" then no code is generated. The program provided demonstrates the problem (see submitter text).

Temporary solution:
Pull the macro definition outside of the conditional if. No code will be generated for the definition.

```
"processor name"

ESSAI        EQU     0

MAC          MACRO
             .IF     ESSAI.EQ.0   FIN
LABEL        LD      A,0
FIN          MEND


             IF      ESSAI
             MAC
             ENDIF

START        LD      A,3
```

Signed off 08/25/86 in release 01.50

KPR #: D200085357 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
Stepping says "PC=123456@sp"; reg says "PC=12345678"

Problem:
When the pc has an "invalid" value (greater than 24 bits), the pc displayed at the end of stepping shows only the 6 most-significant characters:

```
    next PC = 123456@sp
```

The pc actually contains 32 bits of value (as given by the command "reg pc"):

```
    pc=12345678
```

The "next PC" display should show either

```
     next PC = 12345678@sp
  or next PC = 345678@sp
```

Signed off 02/02/88 in release A00.03

KPR #: D200085571 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
Trace list mne heading doesn't indicate base (hex)

Problem:
The heading for the mnemonic field in the trace list should indicate what number base is being used, eg. "68000 Mnemonic,H. See the Z80 or 8018x for examples.

Signed off 02/02/88 in release A00.03

KPR #: D200085662 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
Overlapping IAL output if 6800 cycle and GRD or ROM

Problem:
The IAL output for a 6800 cycle which is a write to ROM or an access to guarded memory will look like:

```
        0000  user data rd byte (6800)OM
        0000  user data rd byte (6800)RD
```

The problem is that first part, ending with "(6800)" is too long and overlaps "ROM" or "GRD". Since the field cannot be made wider, the suggested fix is to drop the parens, like so:

```
        0000  user data rd byte 6800 ROM
        0000  user data rd byte 6800 GRD
```

Note that in practice this situation should occur rarely.

Signed off 02/02/88 in release A00.03

KPR #: D200085696 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
In RESET state, try to b(reak), end up running and unable to break

Problem:

```
    >b
   M>cf clk=ext
   c>cmb -e
   c>x
   ASYNC-ERR: Unable to break
   ASYNC-ERR: Run failed from CMB execute
   c>cf clk=int
   ASYNC-STAT: CMB execute break
   R>b
   ERROR: Unable to break
   U>
```

The last "Unable to break" error message should not appear, and execution should not end up in user code. Proper operation is

```
   R>b
   M>
```

Sometimes the error occurs, sometimes it doesn't.

Signed off 02/02/88 in release A00.03

KPR #: D200086876 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
Stepping in user space does not work with foreground monitor

Problem:
Stepping in the user address space does not work with a foreground monitor. Example, if "map 0..0@u" then "s 2 0@u" gives errors:

```
!ERROR   680! Stepping failed
!ERROR   684! Failed to disable step mode
```

Signed off 02/25/88 in release A00.05

Duplicate Service Requests: D200087007

KPR #: D200086884 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
Emulator stays in monitor after the run command without giving a message

Problem:
If the stack pointer in the 68000 emulator is set up in such a way that a bus error is returned by the target system for stack operations a run command will not be successfully executed. The emulator will stay in the monitor. The emulator gives no message indicating why it did not run, it simply returns the "M>" prompt again.

Signed off 02/25/88 in release A00.05

KPR #: D200087015 Product: 68000 12MHZ EMUL FW 64742 00.00

One-line description:
Can do a "load/display target memory" with no target system

Problem:
If you attempt to load or display target memory, when no target system is connected, the command seems to complete properly, but with no error message that it really didn't do what you expected. The monitor runs, so software obviously knows that the access is to the target system memory. You would expect a "slow device" error in cases where we attempt to access target memory, but no target is connected.

We suspect this may be like something we observed in delta, where the "dtack" line may not be pulled up inside the emulator hardware. This allows the 68000 cycles to terminate even though nothing was specifically driving dtack low (to terminate the cycle).

Signed off 02/25/88 in release A00.05

KPR #: D200089631 Product: 68000 12MHZ EMUL FW 64742 00.04

One-line description:
Slow Clock interferes with configuring monitor... Poor error messages.

Problem:
When an external clock is selected, but the target system is not powered-up, the configuration command

```
  cf mon=fg..1000H@s
```

fails, with the message

```
  !ERROR ! Invalid configuration value: fg..1000H@s
```

The message does not give any hint that the real problem is the fact that there's no clock.

This is a problem for the HPUX interface, because there is no message indicating slow clock. Clock source and monitor type are set up within configuration, and incompatibilities are not found until the entire config has been entered; the user has no clue to the order in which the config commands were sent to the pod.

Temporary solution:
There is no workaround.

KPR #: D200091587 Product: 68000 12MHZ EMUL FW 64742 00.05

One-line description:
All states requested from emtrdata should be valid

Problem:
There is a problem with the emtrdata() input structure. The values contained by "startline" and "endline" should always be valid, regardless of their magnitude. Any value that is out of bounds should be accepted and changed to the maximum range acceptable by the specific analyzer.

Temporary solution:

KPR #: D200091587 **CONTINUED**

There is no workaround available.

KPR #: D200089649 Product: 68000 12MHZ EMUL DOS 64742S006 00.01

One-line description:
Performance problem in the PC interface

Problem:
Communications between the PC host computer and the 64742 emulator are unnecessarily slow. Performance of the PC interface is poor as a result (command responses take too long).

Signed off 05/20/88 in release A01.00

KPR #: D200089805 Product: 68000 12MHZ EMUL DOS 64742S006 00.01

One-line description:
Invalid expressions can corrupt PC memory

Signed off 05/20/88 in release A01.00

KPR #: D200090134 Product: 68000 12MHZ EMUL DOS 64742S006 00.01

One-line description:
The "stty" command doesn't work correctly for baud rate <= 1200.

Problem:
If you toggle the xon parameter when running at 1200 baud and below, the stty command will return invalid characters.

```
>stty
 stty A 1200 xon
>stty -xon
 [#!,*&^junk characters
```

Since the PC interface calls the stty command upon startup, this problem will make the PC interface fail at startup with a datacomm error at 1200 baud (all lower baud rates are not supported by the PC interface).

Temporary solution:
To get around this problem, just set switch 13 on the emulator's back panel (enable xon). The stty parameter will not be toggled and PC interface will startup successfully.

From the terminal-mode interface, just enter another carriage-return to regain proper communications.

KPR #: D200090605 Product: 68000 12MHZ EMUL DOS 64742S006 00.00

One-line description:
OR'ing more than 4 label values in analyzer specification doesn't work

Problem:
Detailed Listing for Defect Number LSDqf03816
Text:
  Or'ing more than 4 label values in analyzer spec doesn't work

.labnotes

KPR #: D200090605 **CONTINUED**

The following trace spec will not work correctly:

```
trace after addr=1 or addr=2 or addr=3 or addr=4 or addr=5
```

This doesn't work when there is a level 1 tree where level 1 contains an "or" node and level 0 contains point or range leaves. This will work correctly if you have a level 2 tree containing "and" nodes in level 1.

.submitter

The following trace spec in the PC user interface will not work correctly:

```
trace after addr=1 or addr=2 or addr=3 or addr=4 or addr=5
```

OTHER STARS PRODUCTS AFFECTED: 64753S006, 64764S006, 64765S006

Resp engr: Cheryl Brown

Temporary solution:
There is no known workaround available.

KPR #: D200091546 Product: 68000 12MHZ EMUL DOS 64742S006 01.00

One-line description:
Remove stack checking programa in production build code

Problem:
The pragma for stack checking should be removed in the FUI code along with the symbol libraries and programmatic interface. This must be done at compile time.

Temporary solution:
There is no workaround available.

KPR #: D200091561 Product: 68000 12MHZ EMUL DOS 64742S006 01.00

One-line description:
gregnumarray[0] not allocated in regdisp() or regmod ()

Problem:
The structure r->param.readreg.renumarray[] needs to have one member allocated before called.

The gregnumarray[0] is not allocated in regdisp() or regmod(). The structure r->param.readreg.renumarray[] needs to have one member allocated before it is called.

Temporary solution:
There is no workaround available.

KPR #: 5000216051 Product: 68000 ASSEMB 64845 01.10

One-line description:
ILLEGAL OPCODE IS BEING GENERATED FOR LEA INSTRUCTION.

Problem:
Assembler generates incorrect code for LEA instruction .
EXAMPLE

```
                      1 "68000"
              <1800>  2 USTACK      EQU       1800H
                      3             PROG
  000000 5FFB 1800    4             LEA       USTACK,USP
```

The code "5F" is incorrect.
This occurs only when using USP in LEA.

Temporary solution:
No temporary solution known at this time.

Signed off 01/14/88 in release Z02.00

KPR #: 5000216267 Product: 68000 ASSEMB 64845 01.12

One-line description:
Incorrect code generated for Bit family of instructions when .L specif.

Problem:
Bit instructions with dot long (.L) extensions generate bad code.
Example:

```
"68008 | 10"
        BTST   #7,D5 ;generates correct code  (0805 0007)
                     ;the .L is implicit, because the destination
                     ;is a register
        BTST.L #7,D5 ;generates incorrect code (0805 00000007)
        END
```

The BCLR.L and BSET.L also generate incorrect code.

This defect is only present in the 68008 and 68010 tables.

Temporary solution:
Do not specify the implicit .L (i.e use BTST, not BTST.L)

Signed off 01/14/88 in release Z02.00

KPR #: 5000243048 Product: 68000 ASSEMB 64845 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
Missing whitespace is not flagged.

Problem:
The following code should cause an error to be generated because no whitespace is included. Correct code is generated.

KPR #: 5000243048 **CONTINUED**

```
"68000"

        DC.B08
        DC.W10
        DC.L15
```

KPR #: 5000247437 Product: 68000 ASSEMB 64845 01.12

One-line description:
Size qualifiers in cross reference.

Problem:
The instruction size qualifiers are listed in the xref as undefined symbols (64000 only).

```
"68000"

                MOVE.B   D0,D1
```

'B' is listed in xref.

KPR #: 5000258590 Product: 68000 ASSEMB 64845 01.13

One-line description:
Math operators not working on 64100.

Problem:
Math operator not working properly on 68000 assembler on 64000 host. Example:

```
"68000"
LAB    DC.W   ((-1*12A9H.SL.0.SR.0).SL.15)/10000
  ^ ERROR LR
LAB1   DC.W   0F6AB8000H/10000  ; should be same answer as above
  ^ ERROR LR
LAB3   DC.W   -156532736/10000 ;works OK
       END
```

These all assemble without error on the 300 and VAX, and return the correct result (0C2DB).

Temporary solution:
Make the calculation yourself as is done in the last expression (for LAB3).

KPR #: 5000270637 Product: 68000 ASSEMB 64845 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
No A5 prompt when non-existient .R file specified.

Problem:
The error output is not correct when the following linker command

KPR #: 5000270637 **CONTINUED**

file is used.

```
 segment
 object files        test1.R test2.R test3.R
 library files
 load address        1000H,2000H,0,0 <----------68000 file A5
```

The problem is that test3.R does not exist. Rather than reporting that test3.R does not exist the linker gives the following error:

```
   3 load address must be specified.
```

Temporary solution:
No temporary solution.

Duplicate Service Requests: 5000270215

KPR #: D200018747 Product: 68000 ASSEMB 64845 00.70

Keywords: ASSEMBLER

One-line description:
The legal range for data in the MOVEQ instruction is incorrect

Problem:
The legal range for the immediate data in the MOVEQ instruction is from 127 to -128. If any other data value is used a Legal Range error should be generated by the assembler. But, no Legal Range error is generated for immediate data ranges 128 to 255 and -129 to -254.
For example,

```
                 1 "68000"
000000 7E7F      2             MOVEQ    #127,D7
000002 7E80      3             MOVEQ    #128,D7
000004 7E00      4             MOVEQ    #256,D7
ERROR-LR
```

The error occurs on line 3. The code generated for this instruction corresponds to immediate value #-128 rather than #128. A legal range error should have been generated like the one that occurs in line 4. This error message would alert the user to the invalidity of the instruction.

Temporary solution:
No temporary solution known at this time.

Signed off 01/14/88 in release Z02.00

KPR #: D200064998 Product: 68000 ASSEMB 64845 01.10

One-line description:
RORG may cause generation of invalid errors and warnings.

Problem:
When RORG is turned on invalid warnings and errors may be flagged.

KPR #: D200064998 **CONTINUED**

```
"68000"

        CMPI.B          #3,LABEL[A5]
        MOVE            [A5],[A3]
        MOVE            8[A5],8[A3]
        MOVE            8[A3,D0],8[A4,D1]
```

Temporary solution:
Turn RORG of (NO_RORG) around sections of code which generate invalid errors.

Signed off 01/14/88 in release Z02.00

KPR #: D200079319 Product: 68000 ASSEMB 64845 01.12

One-line description:
Assembler produces incorrect code for several instructions

Problem:
The current 68008 assembler adds extra bytes to four instructions. They are as follows:

```
      Instruction                Data
      -----------                ----

     BTST.L  #1F,D4  old:      0804001F
                     current:  08040000001F

     BCHG.L  #01,D3  old:      08430001
                     current:  084300000001

     BCLR.L  #0A,D7  old:      0887000A
                     current:  08870000000A

     BSET.L  #05,D1  old:      08C10005
                     current:  08C100000005
```

The 68000 family user's manual indicates that these are all supposed to be four byte instructions.

Signed off 01/14/88 in release Z02.00

Duplicate Service Requests: 5000232058

KPR #: D200081794 Product: 68000 ASSEMB 64845 01.30

Keywords: PROBLEM ON 9000/S300

One-line description:
Bcc causes linker error if incorrect syntax is used.

Problem:
The Bcc instruction is causing problems if illegal syntax is used for the destination address.

KPR #: D200081794 **CONTINUED**

```
"68010"

    BPL         2
    MOVE.L      D0,D1
```

Causes a linker error "Displacement > 32k".

Temporary solution:
The correct syntax for the sample instruction is:

```
      BPL          $+2

or
      BPL          LABEL
LABEL
```

KPR #: D200086678 Product: 68000 ASSEMB 64845 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
TITLE directive inserting garbage control characters.

Problem:
The following program demonstrates a problem with the titlte directive. If you 'vi' the output file you will see that garbage characters are in the output file.

```
"68000"


         TITLE      "i changelength"

         MOVE        D0,D1
```

Temporary solution:
No temporary solution at this time.

KPR #: D200090498 Product: 68000 BBA 300 64380S004 01.00

One-line description:
Complex conditional assignment delcarations cause bbacpp to core dump

Problem:
Detailed Listing for Defect Number LSDqf03782
Text:
complex conditional assignment declarations cause bbacpp to core dump

Some complex conditional assignments will cause bbacpp to core dump if they are in declaration statements. For example:

```
main()
{
   int digit = 6 - ((__ctype[c]&01) ? '0' : ((__ctype[c]&02)
               ? 'A' - 10)
               : ('a' - 10));
}
```

Causes a problem.

Temporary solution:
Move the assignment statement out of the of declaration:

```
main()
{
   int digit;
   digit = 6 - ((__ctype[c]&01) ? '0' : ((__ctype[c]&02)
            ? 'A' - 10)
            : ('a' - 10));
}
```

KPR #: D200090506 Product: 68000 BBA 300 64380S004 01.00

One-line description:
Switch statement followed immediately by a label cases bbacpp to fail

Problem:
06/02/88 LSD STARS DTS LINK COPIED FROM D200090456 64381S004
Detailed Listing for Defect Number LSDqf03784
Text:
switch statement followed immediately by a label cases bbacpp to fail

If a switch statement of the form

```
   switch(var)
   label:
      if (alpha)
      {
         case 5 :
         alpha++;
      }
```

is encountered, bbacpp will generate incorrect code that changes the logic of the program. Note that this is a *very* unusual construct.

KPR #: D200090506 **CONTINUED**

Workaround:
Place braces around the label:

```
switch(var)
{         /* added brace */
label:
   if (alpha)
   {
      case 5 :
      alpha++;
   }
}         /* added brace */
```

Temporary solution:
Place braces around the label:

```
switch(var)
{         /* added brace */
label:
   if (alpha)
   {
      case 5 :
      alpha++;
   }
}         /* added brace */
```

KPR #: D200090514   Product: 68000 BBA   300 64380S004   01.00

One-line description:
A switch statement with no statement causes bbacpp to fail

Problem:

Detailed Listing for Defect Number LSDqf03785
Text:
a switch statement with no statement causes bbacpp to fail

If a switch statement of the form

```
        switch(5) ;
```

is encountered by bbacpp, bbacpp will issue an incorrect warning and refuse to continue parsing the file.

Note that this statement is somewhat non-sensical; there are no 'case' statements for the switch to go to.

Temporary solution:
No workaround is needed.

KPR #: 5000163048   Product: 68000 C   64819   01.08

One-line description:
Pass 3 error 1113 flagged.

Problem:
Error 1113 is flagged in the following program.

```
"C"
"68000"

main()
{

  int  intc;
  int  *int_ptr;

  int_ptr = & intc;

  *(int_ptr + intc)--;            /* Error 1113 flagged. */
  intc++;                         /* Doesn't appear in listing. */
  intc--;                         /* Appears in listing. */

}
```

Temporary solution:
Break the instruction into two parts as demonstrated below.

```
"C"
"68000"

main()
{
  int   intc;
  int   *temp,*int_ptr;

  int_ptr = &intc;

  temp = int_ptr + intc;
  *(temp)--;

}
```

KPR #: 5000173815   Product: 68000 C   64819   01.09

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"
```

KPR #: 5000173815 **CONTINUED**

```
struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: 5000176065   Product: 68000 C   64819   01.10

One-line description:
Terenary expression causing incorrect code to be generated.

Problem:
The terenary expression in the following program causes incorrec code to be generated if the condition is false.

```
"C"
"68000"


main() {
   double   x[10],a,b;
   int      c,d,i;

   for (i=0; i<= 9; i++) {

       x[i] = ((!c) && (d)) ? (a/b) : 1000;
   }

}
```

If the condition is true the compiler correctly generates the array index, however, if the condition is false the compiler never calculates the array index.

Temporary solution:
Replace the terenary expression with an 'if' statement.

```
"C"
"68000"

main() {

   double  x[10],a,b;
```

KPR #: 5000176065 **CONTINUED**

```
   int     c,d,i;

   for (i=0; i<=9; i++) {

       if ( (!c) && (d) )
          x[i] = (a/b);

       else  x[i] = 1000;
   }
}
```

Signed off 08/31/88 in release A02.20

KPR #: 5000192054   Product: 68000 C   64819   01.09

One-line description:
Fields of a structure are dereferenced incorrectly (if fields are big).

Problem:
Structure ponters are not being calculated correctly when relative addressing requires offsets of large sizes. See following code.

```
"C"
"68000"

$FAR$

struct this{
          unsigned short int first[256][256];
          unsigned short int second[256][256];
} one,*bufptr;

unsigned short int *desptr;

main()
{
 bufptr = &one;
 destptr = bufptr->first[0][0];
 destptr = bufptr->second[0][0];   /* Same address assigned. */
 }
```

NOTE: The 8086 line of compilers will generate a pass three error for this code. "Program Counters Disagree. "

Temporary solution:
For the 68000 family of cross compilers you may use the '.' operator instead.

```
"C"
"680XX"

$FAR$
```

KPR #: 5000192054 **CONTINUED**

```
struct  this {
         unsigned short int  first [256][256];
         unsigned short int  second[256][256];
} Structure,*ptrToStruct;

unsigned short int *destptr;
main()
{

 destptr = &Structure.second[0][0];
}
```

KPR #: 5000209742   Product: 68000 C   64819   01.10

One-line description:
Logical operators '&&' and '||' causing bad code to be generated.

Problem:
The 64000-UX 68000 C compiler creates incorrect code in the following example:

```
                                Linker listing:
"C"                             *b=c&&1;
"68000"                                 TST.W     Dstatic[A5]
int c,*b;     these should be ---->     BEQ       main01_1
main()        swapped!        ---->     MOVEA.L   Dstatic+00002h[A5],A0
{                                       MOVE.W    #00001h,[A0]
*b=c&&1;                                BRA       main01_2
*b=c||1;                            main01_1
}                                       CLR.W     [A0]
                                    main01_2
                                        UNLK      Rmain
The listing info. show the case for
&&.  The same happens for ||.       c   EQU       Dstatic+0h
If TST.W equals 0, A0 never gets    b   EQU       Dstatic+02h
initialized.  To clear it first
would be ineffective.
```

Temporary solution:
Use the alternate if-then-else.

```
"C"
"68000"

int c,*b;
main()
{
/* For the and operator. */
if (c&&1)
   *b=1;
else *b=0;

/* For the or operator. */
if (c)
   *b=1;
```

KPR #: 5000209742 **CONTINUED**

```
else if (1 )
   *b=1;
else
   *b=0;
}
```

Signed off 08/31/88 in release A02.20

KPR #: 5000220418   Product: 68000 C   64819   01.10

One-line description:
Address is not incremented past 0xFFFF for data areas > 32k.

Problem:
In the expanded listing the address of the variable declarations is not shown when the size of the data area is greater than FFFFH. I am refering to the top of the listing where the C declarations of the arrays are made.
In the expanded listing the address of the variable declarations is not shown when the size of the data area is greater than FFFFH. I am refering to the top of the listing where the C declarations of the arrays are made.

Temporary solution:
No temporary solution at this time.

KPR #: 5000222307   Product: 68000 C   64819   01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Optimize directive causing bad code to be generated.

Problem:
Optimize option may cause bad code to be generated when arrays of pointers to arrays are used. Example:

```
"C"
"68000"
$OPTIMIZE+$
typedef int MEAS_MARK[4];
MEAS_MARK *MEAS_INFO[4];
main()
{       int  request;
        (*MEAS_INFO[3])[0] = 0;
        request = (*MEAS_INFO[3])[3];
/* the D0 register is cleared to do the first statement, then the
  compiler erroneously assumes that D0 has the correct value for
  (*MEAS_INFO[3])[3] - it has the value of (*MEAS_INFO[3])[0]   */
}
```

Temporary solution:
Turn optimize off and on again around the problem statement.

KPR #: 5000222307 **CONTINUED**

Signed off 08/31/88 in release A02.20

KPR #: 5000223107 Product: 68000 C 64819 01.10

One-line description:
float += float(unsigned -unsigned) hangs compiler.

Problem:
The following program effects the compilers is different ways.

```
Z80  Error 1006
6800 Error 1001
6809 Error 1001
Z8001  Works
8086   Error 1001
8085   Error 1006
```

```
"C"
"processor"

unsigned int aa,b;

float c;

int func()
{
  aa += c*(b-a);
}
```

Temporary solution:
Use signed integers rather than unsigned.

```
"C"
"processors"

int a,b;
float c;

int try()
{
a += c*(b-a);
}
```

Signed off 08/31/88 in release A02.20

KPR #: 5000226530 Product: 68000 C 64819 01.10

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

KPR #: 5000226530 **CONTINUED**

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
           ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;

  OR
```

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: 5000229237 Product: 68000 C 64819 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Inconsistient error messages for too large of data area.

Problem:
68000 C compiler will generate a pass 3 error number 1113 when the data space exceeds 32K, and the FAR directive is used. Error 1113 is described as "Program counters to not agree". The correct error message is 1105 "Data size too large". This is true on the host machines, however, the 64000 incorrectly does not generate an error.

Temporary solution:
No temporary solution.

KPR #: 5000229237 **CONTINUED**

Signed off 08/31/88 in release A02.20

KPR #: 5000236828 Product: 68000 C 64819 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Calling a function w/o assigning result causes stack to get messed up.

Problem:
Bad code generated when procedure that returns a double is called without assignment to a result, and $FIXED_PARAMETERS ON$ is used. Example:

```
"C"
"68000"
$FIXED_PARAMETERS ON$
extern double dummy();
main()
{
       dummy();
}      /* the problem occurs after the call to dummy - an ADDQ.L,A7 */
       /* is done to adjust the stack unnecessarily                 */
```

Temporary solution:
Create a dummy variable and assign the result of the function to this variable.

```
main()
{
double dummy;

dummy = doubleFunc();
}
```

Signed off 08/31/88 in release A02.20

KPR #: 5000264481 Product: 68000 C 64819 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Problem with Type Name cast - causes Pass 1 error.

Problem:
Type Name cast causes Pass 1 error with no info other than:
"comp: C Pass 1 cannot recover from errors, parsing stopped at line .."
Example:

```
"C"
"68010"
extern  unsigned short *list[];
int   main()
{
    unsigned short (*maddr)[6];
```

KPR #: 5000264481 **CONTINUED**

```
    maddr = (unsigned short(*)[6])list; /* this line causes the */
}                                       /* problem              */
```

This operation is discussed on pages 199 and 200 of K&R.

The problem seems to come from an error in the way the pointers are assigned, i.e. if the program is changed to:

```
. . .
{
    unsigned short *maddr[6];
    maddr = (unsigned short (*)[6])list; /* . . .
. . .
```

it seems to compile.
But, according to K. & R., this should be an lvalue error.

Temporary solution:
Change program structure to form a new block - then redefine the variable type as needed inside this block. Upon exiting this block the original type cast will resume.

e.g.

```
"C"
"68010"
extern unsigned short *list;
int main()
{
      {           /* new block */
         unsigned short (*maddr)[6];
         unsigned short (*list)[6]; /* redefined var. type */
         maddr = list ;  /* simplified assignment statement */
      }
     /* Rest of program
        goes here . . . note that the global type defn.
        for the variable "list" now takes over */
```

This isn't pretty, but it is functional.

KPR #: 5000269407 Product: 68000 C 64819 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
The EXT.L command does not work properly.

Problem:
The problem seems to be with the EXT.L instruction itself - when the code is executed D0 contains 'FFFF8000h' (it should contain '00008000h'. Somehow the negative flag in the status register is getting set. Then, when the EXT.L is executed, the computer thinks the number it is expanding is negative - and it expands it with 2's compliment using 'F's instead of zeros.

KPR #: 5000269407 **CONTINUED**

NOTE: This problem has come up in other places, too. See
SR #5000271957.

Temporary solution:
Cast to an unsigned long;

```
    ul = (unsigned long)0x8000;
```

KPR #: 5000269415 Product: 68000 C 64819 02.10

One-line description:
Bad code is generated when a char var is compared to a negative number

Problem:
Bad code generated when a var that is type char is compared to a negative number. Example:

```
"C"
"68000"
char  c;
main()
{
    if( c == -1 )
              c = 1;
}
```

the problem is that 8 bit value is moved to a register via a MOVE.B instruction, but then compared to 0FFFFH via a CMPI.W instruction.

Temporary solution:
Cast the -1 to a char:

```
     if( c == (char)-1 )
```

KPR #: 5000271957 Product: 68000 C 64819 02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
Problem with EXT.L command.

Problem:
When we assigns the address value to the pointer, the address is converted to a long by EXT.L. The address is 8000H thru 0FFFFH. In result, the pointer points to unexpected memory location.
example;

```
           "C"
           "68000"
            main()
              {
                int *a;
                a=0x8000;
                   EXT.L is used here.
              }
```

The problem seems to be with the EXT.L instruction itself - when the code is executed D0 contains 'FFFF8000h' (it should contain '00008000h'. Somehow the negative flag in the status register

KPR #: 5000271957 **CONTINUED**

is getting set. Then, when the EXT.L is executed, the computer thinks the number it is expanding is negative - and it expands it with 2's compliment using 'F's instead of zeros.

NOTE: This problem has come up in other places, too. See
SR #5000269407.

Temporary solution:
workaround:
Use a cast and change the type of address.

```
   a=(long)0x8000
```

This statement generates MOVE.L instruction instead of MOVE.W and EXT.L.

KPR #: 5000273730 Product: 68000 C 64819 01.10

Keywords: PREPROCESSOR

One-line description:
Station reset during preprocessor pass.

Problem:
Station reset during processor pass.

Temporary solution:
None.

Signed off 08/31/88 in release A02.20

KPR #: D200004929 Product: 68000 C 64819 00.21

Keywords: CODE GENERATOR

One-line description:
Multiple assignments may cause compiler to reuse an overwritten reg.

Problem:
Statements requiring extensive use of address reisters may cause the compiler to use all available registers and then reuse a register that has been over-written. Such a situation is encountered in the following source line where address register A0 originally contains the base address of the array of structures xx[]. By the time that the assignment xx[i].x = xx[i].y is made A0 has been over written to contain the address of xx[i], but it is used as if it still contained xx[0].

```
     struct { double x, y, z;
            } xx[2];
     int i;
     double d;
     main()
     {
       xx[i].x = xx[i].y = xx[i].z = d;
     }
```

Temporary solution:

KPR #: D200004929 **CONTINUED**

Break such complex statements up into simpler statements; i.e., three separate assignments.

KPR #: D200014399   Product: 68000 C   64819   01.07

Keywords: CODE GENERATOR

One-line description:
Bad code using $OPTIMIZE$ and successive uses of the same pointer.

Problem:
Sometimes bad code is produced when $OPTIMIZE ON$ is in effect and one dereferences a pointer, updates the pointer, and dereferences the pointer again. For example,

```
   char *p,c;
   main()  {  $OPTIMIZE ON$
      c = *p;    /* Dereference a pointer */
        MOVEA.L Dstatic[A5],A0      ; p is loaded into A0
        MOVE.B  [A0],Dstatic+4[A5]
      p += 1;    /* Update the pointer */
        MOVEA.L Dstatic[A5],A1
        LEA     1[A1],A2            ; updated value of p is in A2
        MOVE.L  A2,Dstatic[A5]
      c = *p;    /* Dereference the pointer again */
        MOVE.B  [A0],Dstatic+4[A5]  ; ERROR - A0 contains the old value
                                    ; of p, not the updated value.
```

Temporary solution:
Turn $OPTIMIZE OFF$ around the operations of the above type.

KPR #: D200032045   Product: 68000 C   64819   01.07

One-line description:
Compiler uses MSB of word containing char value rather than LSB.

Problem:
When a character is passed as a parameter, its value is loaded into the lower byte of a word. The most significant byte of this word is passed as the address of this character. Our compiler doesn't realize that the actual ascii value is in the lower byte. So, when this character is referenced, the compiler uses the contents of the most significant byte rather than the lower byte which contains the ascii value. See code and comments below.

```
"C"
"68000"
/*The following sequence of function calls will duplicate the problem.*/
main()
{ putchar('5');
}

putchar(parm)
char parm;
{
write(&parm);
```

KPR #: D200032045 **CONTINUED**

```
}

write(p1)
char *p1;
{
 char temp;
 temp = *p1;       /* temp is loaded with zero rather than the ascii
}                     value for '5' */
```

The intermediate function call to putchar is needed to duplicate this problem. Simply calling write from main will not duplicate the problem.

Temporary solution:
Do not declare parameter to be of type character. Instead define it as an integer and declare a local variable which is intialized to the parameters. SEE CODE BELOW:

```
            "C"
            "68000"
            main L)
            {
              putchar ('5');
            }

            putchar (parm)
            int  parm;          *1 Here is the change *1
            {
              local_variable=parm;

              Write (&local_variable);
            }
            Write (p1);
            char *p1;
            {
              char temp;
              temp=*p1;
            }
```

KPR #: D200063115   Product: 68000 C   64819   01.09

One-line description:
Shift of wrong sized value in register.

Problem:
Shift right sometimes generates incorrect result.

```
"C"
"68000"

int fct1(param)
unsigned short param;
{
unsigned short data;
data = param;
```

KPR #: D200063115 **CONTINUED**

```
/* Zeroes are shifted in for the right shift below. */

data = data >> 7 | data << 1;
return (data);
}


int fct2(param)
unsigned short data;
data = param; .

/* The word size of param is in a register. The next statement
   uses this word size value of param and this results in the
   upper byte of the word being shifted into data instead
   of zeros. */

data = data >> 7 | data << 1;
return(data);
}
```

Temporary solution:
No temporary solution at this time.

KPR #: D200065193 Product: 68000 C 64819 01.09

One-line description:
An "if" statement may cause the compiler to go astray.

Problem:
An if statement causes a branch to be generated in case the condition is false. If the code space below the if statement is >32 in size then the branch goes off into the weeds. Use the following program to demonstrate the problem.

```
"C"
"680000"

main()
{
   int i;

   if(i==1) {

       /*This code must be >32K */
       i++;i++;i++;i++;i++;i++;i++;i++;   /*Repeat 1035 times. */
  }
}
```

Temporary solution:
Don't create such large blocks within an if statement. Perhaps the code below the if can be put in a sub-routine.

KPR #: D200067587 Product: 68000 C 64819 01.09

One-line description:
A0 register not initialized properly when using ||.

Problem:
The following program will generate bad code. The A0 register is not initialized at the correct time.

```
"C"
"6800"

test()
{
  extern int mlvar1,mlvar2;
  extern int *mlchar;          /* It's not necessary that these are
                                  external var's, but, they must be
                                  ints for the discrepency to occur. */

 *mlchar = mlvar1 || mlvar2;
}
```

Temporary solution:
Use an equivalent "if then else" construct.

```
"C"
"6800"

test()
{
  extern   int mlvar1,mlvar2;
  extern   int *mlvar;

  if (mlvar1)
     *mlvar = 1;
  else if (mlvar2)
       *mlvar = 1;
       else  *mlvar = 0;

}
```

Signed off 08/31/88 in release A02.20

KPR #: D200069674 Product: 68000 C 64819 01.09

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"
```

KPR #: D200069674 **CONTINUED**

```
$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function.
For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200071829 Product: 68000 C 64819 01.09

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory,

KPR #: D200071829 **CONTINUED**

remove the DATA pseudo, and reassemble.

KPR #: D200074997 Product: 68000 C 64819 01.10

One-line description:
USE OF FUNCTION POINTER TYPE CAST ON MULTIPLE EXTERNAL ARRAYS CAN ERROR

Problem:
C/68000 program direct call to external code array using function type cast causes error.

The use of the C type cast operator to call an array of integers as if it were a subroutine, may not work when more than 1 external array is used in this manner.

Although the 68000 compiler will generate proper code for the first instance of an external array used for this type cast operation, the use of other external arrays for this feature may not work.

The following example shows the problem. The call to intarrayFIRST works OK, but the calls to intarraySECOND, intarrayTHIRD & intarrayFOUR
TH
call intarrayFIRST by mistake.

Notice that a workaround solution is to use an embedded assignment to a function pointer, which will cause the proper call to be made.

```
 "C"
 "68000"
  extern int intarrayFIRST[];
  extern int intarraySECOND[];
  extern int intarrayTHIRD[];
  extern int intarrayFOURTH[];

 typedef int(*PFI)();  /* PFI a pointer to int functions */
  PFI p;               /* p a pointer of type PFI */
  int answer;

 main()
 {
     main
         LINK     A6,#0
    answer = (*((PFI) intarrayFIRST))();
         JSR      intarrayFIRST+0[A5]
         MOVE.W   D7,Dstatic+00004H[A5]
    answer = (*((PFI) intarraySECOND))();

         JSR      intarrayFIRST+0[A5]    ***NOTE call to intarrayFIRST
         MOVE.W   D7,Dstatic+00004H[A5]
    answer = (*((PFI) intarrayTHIRD))();
         JSR      intarrayFIRST+0[A5]    ***NOTE call to intarrayFIRST
         MOVE.W   D7,Dstatic+00004H[A5]
    answer = (*((PFI) intarrayFOURTH))();
         JSR      intarrayFIRST+0[A5]    ***NOTE call to intarrayFIRST
```

KPR #: D200074997 **CONTINUED**

```
        MOVE.W   D7,Dstatic+00004H[A5]

             /* WORKAROUND solution: use embedded assignment to pointer
 */
   answer = (*( (p = (PFI) intarrayFOURTH )))();
        LEA      intarrayFOURTH+0[A5],A0    ***NOTE call to intarrayFOU
RTH
        MOVE.L   A0,Dstatic[A5]
        MOVEA.L  Dstatic[A5],A1
        JSR      [A1]
        MOVE.W   D7,Dstatic+00004H[A5]
 }

        UNLK     A6
    Rmain
        GLOBAL   Rmain

        RTS
```

Temporary solution:
See problem text.

Signed off 08/31/88 in release A02.20

KPR #: D200076455 Product: 68000 C 64819 01.10

One-line description:
Terenary expression causing error 1113 or "Too many errors. "

Problem:
When compiled the following program compiles correctly on the 64100, causes error 1113 on the 300 and pass two error "Too many errors" on the VAX and 9000/500.

```
"C"
"68000"

error_check(a)
int a[2];

{
  a[0] = (a[0] == 0)?1:a[0]+1;
}
```

Temporary solution:
Use the alternate if-then-else conditional statement.

Signed off 08/31/88 in release A02.20

KPR #: D200076513 Product: 68000 C 64819 01.10

One-line description:
Address comparisons for variables located on negative base-page may fail

Problem:
Address comparisons may not work with $BASE_PAGE$ variables loaded in the address range 0FF8000H-0FFFFFFH.

In particular, addresses in that range generated when initializing pointers will be created by the linker with the upper byte set to 00H for the ABSOLUTE address. They will fail if compared to ABSOLUTE_SHORT addresses (which perform sign extension on the address setting the upper byte to 0FFH) generated with PEA or LEA instructions executed at run time.

The following program illustrates the problem:

```
      "C"
      "68000"
      $BASE_PAGE$
       int other[16];
       int *p =other;
       unsigned int mask16=0xffff;
       main()
       {
0000      main
0000          LINK     A6,#0
       if ( p!=other )  ;
0004          ABSOLUTE_SHORT
0004          MOVE.L   Dstatic+00020H,D0  ### 00FF8XXXH
0008          LEA      Dstatic,A0         ### FFFF8XXXH
000C          CMP.L    A0,D0             #FAIL Compare HERE#
000E          BEQ      main01_1

       /* Workaround # mask addresses if $BASE_PAGE$ */
0012      main01_1
       if ( ( (int)p&(mask16) ) != ( (int)other&(mask16) ) )  ;
0012          MOVE.L   Dstatic+00020H,D1  ### 00FF8XXXH

0016          AND.W    Dstatic+00024H,D1  ### 0000FFFFH
001A          LEA      Dstatic,A1         ### FFFF8XXXH
001E          MOVE.L   A1,D2
0020          AND.W    Dstatic+00024H,D2  ### 0000FFFFH
0024          CMP.W    D2,D1             #OK# 0000XXXXH (D2)
0026          BEQ      main01_2          #==# 0000XXXXH (D1)
002A      main01_2
       }
0000          DATA
0000      Dstatic
0000          DC.B     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0010          DC.B     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0020          DC.L     Dstatic      ### Produces 00FFXXXXH value in link
0024          DC.W     -1           ### 0FFFFH
```

KPR #: D200076513 **CONTINUED**

Temporary solution:
See problem text for a work around.

KPR #: D200079590   Product: 68000 C   64819   01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
      }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081505   Product: 68000 C   64819   01.10

One-line description:
Libraries generate incorrect code 68010 processor.

Problem:
The 68000/08/10 compilers all share the same library routines. In the function Zenter_trap an ADDA.L #6 instruction is used to point to the parameters in the trap procedure. The problem is the 68010 pushes the Vector base register so the instruction should use an immediate value of 8.

Temporary solution:
Contact your local Applications Engineer to obtain a copy of the sources for the libraries.

KPR #: D200085373   Product: 68000 C   64819   01.10

One-line description:
SHORT_ARITH OFF use of mixed short int in conditionals may not work

Problem:
With the SHORT_ARITH option OFF, the 68000 compiler does not execute full K&R C code correctly for certain mixed arithmetic operations when used in "if" expressions.

EXAMPLE:

```
  main() {
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */
```

Detailed Listing for Defect Number LSDqf02058
Text:
SHORT_ARITH OFF use of mixed short int in conditionals may not work

Problem: With the SHORT_ARITH option OFF, the 68000 compiler does not execute full K&R C code correctly for certain mixed arithmetic operations when used in "if" expressions.

EXAMPLE:

```
  main() {
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */
  /* WORKAROUND */
              /* Result will be (0xFD -> 253)+3 = 256 */
```

KPR #: D200085373 **CONTINUED**

```
  if ( (unsigned int)us +3) ;
   else;
  }
```

The 68000 C compiler computes mixed expressions correctly, as in assignment statments and parameter expression.
This defect appears only when mixed expressions are used without assignment as conditional branching expressions.

This problem may be generated with other operators besides the "+" as in the example.

EXPANDED example:

```
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
          MOVE.W   #0FFFDH,D0
          MOVE.B   D0,-2[A6]
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
          ADDQ.B   #3,D0
          BEQ      main01_1
          BRA      main01_2
      main01_1
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */

  /* WORKAROUND */
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
          CLR.L    D1
          MOVE.B   -2[A6],D1
          ADDQ.W   #3,D1
          BEQ      main01_3
          BRA      main01_4
      main01_3
   else;
      main01_4
```

Temporary solution:
EXAMPLE:

```
  main() {
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */
```

KPR #: D200085373 **CONTINUED**

```
  /* WORKAROUND */
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
   else;
  }
```

KPR #: D200085399   Product: 68000 C   64819   01.10

One-line description:
SHORT_ARITH OFF with unsigned short int in conditional branch error

Problem:
This problem may be generated with other operators besides the "+" as in the example.

EXAMPLE:

EXPANDED example:

```
   unsigned short us;
  $SHORT_ARITH OFF$
  us = -3;
          MOVE.W   #0FFFDH,D0
          MOVE.B   D0,-2[A6]
  if (us+3);  /* Result should be (0xFD -> 253)+3 = 256 */
          ADDQ.B   #3,D0
          BEQ      main01_1
          BRA      main01_2
      main01_1
  else
    /* This statement executes erroneously */
              /* But using byte arithmetic  produces FALSE=0 */
```

Temporary solution:

```
              /* Result will be (0xFD -> 253)+3 = 256 */
  if ( (unsigned int)us +3) ;
          CLR.L    D1
          MOVE.B   -2[A6],D1
          ADDQ.W   #3,D1
          BEQ      main01_3
          BRA      main01_4
      main01_3
   else;
      main01_4
```

KPR #: D200087940   Product: 68000 C   64819   02.10

One-line description:
Arrays of 64k Multiples of size caus 210 error unnecessarily

Problem:
Detailed Listing for Defect Number LSDqf02937
Text:
  Arrays of 64k multiples of size cause 210 error unnecessarily

KPR #: D200087940 **CONTINUED**

more-info

```
dtssub: -fmq nam="Bob Gronlund" pho=590-5699 add=hplsdla!bob editor=vi
dtssub: en=- sys=hplsdla
dtssub: ver=current sf=a rep=stars act=cu an=n en=.dtsrc in=more-info
dtssub: rf=more-info ph=e

dtsrec: -dtp eng="" proj="" editor=vi how=screen pro=language_maint

dtsrep: -fmp  report=ms pro=language_maint where=f dn="LSDqf000" bs=n
dtsrep: dest=/users/bob/dtsdir/unsort.lm

dtsupd: -ftp editor=vi us=no ua=yes another dn="LSDqf000" how=e
dtsupd: mod=""
```

.submitter

An array declared as:

```
short data[512][128];
```

will cause a spurious 210 error. This problem occurs for any array (single or multiple dimensioned) that has a size that is a precise multiple of 64k.

This problem applies to all C compilers (pass 1 defect).

BG, 880316

Temporary solution:
There is no workaround.

Signed off 08/31/88 in release A02.20

KPR #: 5000184374 Product: 68000 C M 64819-90902 01.09

One-line description:
List library link range in manuals.

Problem:
The 68000 library routines must reside within +/- 32K of each other. This applies to both A5_LIB and ABS_LIB. This caused problems when a customer linked in some of the libraries and burned them into EPROM. He then used that link_sym when linking subsequent code. A new library routine, not previously linked, was loaded. It made reference to a previously loaded library using a BSR which is limited to a +/-32k branch.

Either the libraries should be changed, or the manuals should be updated to reflect this restriction.

Temporary solution:
No temporary solution at this time.

KPR #: D200055699 Product: 68000 C M 64819-90902 01.08

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
```

KPR #: D200055699 **CONTINUED**

```
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.10

KPR #: 5000279117 Product: 68000 DQ EMUL 300 64243S004 01.10

One-line description:
Enabling DMA to emulation memory halts emulator.

Problem:
If the emulation configuration question "Enable DMA transfers to emulation memory" is answered "yes" the emulator releases from reset immediately into a "halt" state.

This problem appears always; whether a target system is hooked up or not.

Signed off 08/19/88 in release A01.30

KPR #: D200078188 Product: 68000 DQ EMUL 300 64243S004 01.00

One-line description:
Questions not asked when switch from real-time to non-real.

Problem:
When modifying an emulation configuration and switching from restricted to real time to NOT restricted to real time, the simulated I/O and simulated interrupt configuration questions are not asked. You must be modifying an existing configuration file.

Signed off 08/28/87 in release 99.99

KPR #: D200078196 Product: 68000 DQ EMUL 300 64243S004 01.00

One-line description:
Wrong breakpoint behavior on continuing emulation

Problem:
When using software breakpoints and doing the following sequence, emulation does not behave as expected.

```
run form LOOP
modify software_breakpoints set LOOP
#breaks into monitor, displays breakpoint and clears breakpoint
run
end
#return to emulation
modify software_breakpoints set LOOP
#breaks into monitor, but does not display or clear breakpoint
# at second iteration, things return to normal
```

Signed off 08/28/87 in release 99.99

KPR #: D200078972 Product: 68000 DQ EMUL 300 64243S004 01.00

One-line description:
EBPP as analyzer fails intermittently

Signed off 08/28/87 in release 99.99

KPR #: D200087197 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
Mnemonic scroll and step display may not read memory

Problem:
Detailed Listing for Defect Number LSDqf02659
Text:
mnemonic scroll and step display may not read memory

Memory may not be read (but rather data in a buffer from a prior read would be used) when a mnemonic memory display is scrolled up, or a step is displayed. The buffer is used when instructions are short. For stepping, the problem will occur if a step is done from the same address as before. Possible scenario:

```
step from 3000h
modify memory 3000h to XXXX
step from 3000h
```

The display would show the original instruction, not the one memory was modified to. In any case, memory should always be read.

NOTE: THIS DEFECT APPLIES TO THE 64244 (68008DP EMUL) AS WELL.

Temporary solution:
There is no known workaround available.

Signed off 08/19/88 in release A01.30

KPR #: D200087759 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
Analyzer break fails when stepping over software breakpoint

Problem:
Detailed Listing for Defect Number LSDqf02819
Text:
Analyzer break fails when stepping over software breakpoint

Problem only occurs when using the analyzer to step.
If a software breakpoint point is hit while stepping, the analyzer is not correctly cleared. As a result, the next step command fails to return back into the monitor and the processor is left running user code.

This defect applies to:

64243	68000DP Emulator
64244	68008DP Emulator
64245	68010DP Emulator

Temporary solution:
There is no known workaround available.

Signed off 08/19/88 in release A01.30

KPR #: D200088039 Product: 68000 DQ EMUL 300 64243S004 01.00

One-line description:
Doing a wait while tracing MAY cause subsuquent traces to never complete

Signed off 08/19/88 in release A01.30

KPR #: D200088062 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
Multiple commands on the line after "set" will not be executed

Problem:
Multiple commands after "set" shall not be executed.

Signed off 08/19/88 in release A01.30

KPR #: D200088310 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

Signed off 08/19/88 in release A01.30

KPR #: D200090357 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
Display memory w/part of the line not readable may hang the system

Problem:
Detailed Listing for Defect Number LSDqf03714
Text:
display memory w/part of line not readable may hang system

Doing a display memory where part of a line is not readable (for example target memory without monitor loaded) will cause the system to hang.
This defect exists in all of the following products:

64243	68000DP	Emul
64244	68008DP	Emul
64245	68010DP	Emul
64286	F9450	Emul
64203	8085	Emul

workaround

Typing <CTRL>-C will make the system return to normal.

KPR #: D200090357 **CONTINUED**

Temporary solution:
Typing <CTRL>-C will make the system return to normal.

Signed off 08/19/88 in release A01.30

KPR #: D200090548 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
bbaunload causes memory growth problems in emulators

Problem:
Detailed Listing for Defect Number LSDqf03817
Text:
bbaunload causes memory growth problems in emulators

The bbaunload library (/lsd/p2/cmd/bba/prod/unload) allocates memory and never frees it.

Please expand to 64244S004 and 64245S004 as well.

Bruce E

FIXED:
The new library fixes these problems as well as making the whole unload process much faster. It is fixed as of 03June88.

Signed off 08/19/88 in release A01.30

KPR #: D200090555 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
_display_message causes 'access guarded memory'

Problem:
Detailed Listing for Defect Number LSDqf03818
Text:
_display_message cuases 'access guarded memory'

3) The routine __display_message does not always work. According to the source, it expects the host to initialize a global variable XEnv_68k_except. The 1.10 emulator software does not always do this, so there is the wrong value in that variable causing the routine to attempt a TRAP 0. The TRAP vectors are empty, so the program gets an address error and dies.

It seems to occur when ending the emulation session, restarting it, and reloading the absolue file.

This problem can be avoided by manually doing
modify memory XEnv_68k_except words to 0

This is from notes, reported by Jerry Metz at LID. Bryce has duplicated the problem here with a routine that just does an exit(1). Please see him for details.

KPR #: D200090555 **CONTINUED**

Please expand to the 64244 and 64245 also.

Temporary solution:
This problem can be avoided by manually doing
modify memory XEnv_68K_except words to 0.

Signed off 08/19/88 in release A01.30

KPR #: D200090829 Product: 68000 DQ EMUL 300 64243S004 01.20

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/19/88 in release A01.30

KPR #: D200031872 Product: 68000 DQ SW ANALYZER 64341G 01.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 01.02

KPR #: D200081372 Product: 68000 EMUL 12.5 MHZ 64243 01.01

One-line description:
State IA generates wrong instruction for Adr Reg. Indirect w/Indexing

Problem:
The assembler syntax of the address register indirect with index addressing mode is d(An,Rn). Rn (the index register) can be either an address or a data register. If the code uses an address register for Rn, the disassembler incorrectly displays the data register of the same number. Refer to the following example:

```
Source Code:        MOVE.W  0[A0,A1].D0
Disassembled Code:  MOVE.W  000H[A0,D1.W],D0
                                    ^----------incorrect value
```

Note that the index register was incorrectly disassembled as D1 instead of A1.

Temporary solution:
No workaround at this time.

KPR #: D200067637 Product: 68000 EMULATION 64242 01.07

One-line description:
Load of more than 1 abs. to targ. mem. not allowed when restricted to RT

Problem:
When restricted to real time, loading more than one absolute file to target memory gives an error when attempting the second load, "ERROR: Command causes break- runs restricted to real time".

The emulator is actually running in the monitor at this time but the emulation software seems to think that it is executing target memory.

Example:

```
emulate CONFIG load MON ;restricted to real-time, load monitor
load FILE1              ;FILE1 mapped to target memory
load FILE2              ;FILE2 mapped to target memory <<<<<
                                       Error occurs here >>^
```

Temporary solution:
Add a "break" between each load

Example

```
emulate CONFIG load MON
load FILE1              ;mapped to target memory
break
load FILE2              ;mapped to target memory
break
load FILE3              ;mapped to target memory
etc.
```

KPR #: 5000187674 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Ehalt occurring too often while running user code.

Signed off 10/02/87 in release 01.10

KPR #: D200069484 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Duplicate Service Requests: D200069500 D200069492

KPR #: D200080606 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080903 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as

KPR #: D200080903 **CONTINUED**

well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081679 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Tracing on status int_ack does not work.

Problem:
Tracing on status int_ack does not work.
To observe an interrupt service routine, "trace about status int_ack" does not work. One must instead trace about an address in the interrupt vector table (0h thru 2ffh).

Temporary solution:
Trace on an address (range) in the interrupt vector table. This will result in capturing nearly the same amount of information.

KPR #: D200081885 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082180 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

KPR #: D200082180 **CONTINUED**

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
     cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082594 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Memory breaks during stepping are not detected

Problem:
Memory breaks (write to ROM, etc.) which occur while stepping will not be noted on the status line.

Temporary solution:
No workaround at this time.

KPR #: D200082610 Product: 68000 EMULATION 300 64242S004 01.00

Keywords: BREAKPOINT

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

Temporary solution:
Do not use software breakpoints in target memory. If software breakpoints are required to debug a certain section of code, make sure that section is mapped to emulation memory.

KPR #: D200083196 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:

KPR #: D200083196 **CONTINUED**

Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085993 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                           ; end locks the emulation session
3. <system name> <module name>   ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200090811 Product: 68000 EMULATION 300 64242S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

KPR #: D200090811 **CONTINUED**

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200013110 Product: 68000 HL SOFT ANAL M 64331-90902 01.00

One-line description:
Tracing a variable declared as a pointer to a function doesn't work in C

Problem:
If a variable is declared as a pointer to a function (using C) and you then try and trace that variable in Spider an "Invalid type encountered" message will occur. No known workaround situation is currently known.

KPR #: 1650006700 Product: 68000 PASCAL 64815 00.00

One-line description:
Immediate operand's value is altered when doing a logical and.

Problem:
Incorrect code is generated for the below program. Specifically, the immediate value is altered when the logical AND is done.

```
"68000"
$EXTENSIONS$

PROGRAM TEST;

TYPE
  MASK = 0..15;
  INTEGER = SIGNED_16;

VAR
  CNTL_REG, I : INTEGER;


BEGIN
  IF INTEGER(MASK(CNTL_REG)*MASK(SIGNED_16(0FH)) > 9
     THEN;

END.
```

Temporary solution:
The April SMS has fixed this problem. The revisions involved are 1.3 on the 9000, 1.4 on the VAX and 1.10 on the 64100A.

KPR #: 2700005561 Product: 68000 PASCAL 64815 01.09

Keywords: DIV

One-line description:
UNSIGNED_32 division with dividend or divisor > 8000,0000H may not work.

Problem:
RE: UNSIGNED_32 DIVISION.
0E000,0000H / 09000,0000H return the wrong value. Dividend and divisor values less than 08000,0000H appear OK.

Signed off 08/25/86 in release 03.01

KPR #: 5000094615 Product: 68000 PASCAL 64815 01.02

Keywords: CODE GENERATOR

One-line description:
B := ABS(B) fails to write to the data area.

Problem:

```
VAR I : INTEGER;  B : BYTE;

BEGIN
```

KPR #: 5000094615 **CONTINUED**

```
I := B;
IF I < 0 THEN
I := ABS(I);
       ^ Although I is complimented here, it is kept in the register
         and not rewritten to the data area.
```

Temporary solution:
IF I < 0 THEN I := -(I);

Signed off 08/25/86 in release 03.01

KPR #: 5000119933 Product: 68000 PASCAL 64815 01.09

One-line description:
Compiler allows non-standard funct. with EXTENSIONS OFF or ANSI ON

Problem:
With the compiler option $EXTENSIONS OFF$ or $ANSI ON$ some of the extensions, such as ADDR,SHIFT, and ROTATE, are permitted without an error message being created. The following sample program causes no error messages and generates assembly code when compiled.

```
       "68000"
       $EXTENSIONS OFF$  {or $ANSI ON$}

       PROGRAM TEST;

       VAR  V : INTEGER; P : ^INTEGER;

       BEGIN
         P := ADDR (V);
         V := SHIFT (V,1);
         V := ROTATE (V,1);
       END.
```

However, when testing other Pascal extensions, such as type changing, BYTE data type, and non-decimal constants, errors 455 and/or 504 were generated. This same problem occured on th 8086 and 6800.

NOTE: 455 - extensions used in extensions off mode
504 - non-standard feature used

Signed off 08/25/86 in release 03.01

KPR #: 5000161182 Product: 68000 PASCAL 64815 01.12

One-line description:
Bad code when taking ADDR of record element when using WITH.

Problem:
Code generated by the 68000 Pascal compiler is wrong in the following example:

"68000"

KPR #: 5000161182 **CONTINUED**

```
PROGRAM  test;
$EXTENSIONS ON, EXTVAR ON$

TYPE    x_type = RECORD
          f1   : BYTE;
          f2   : BYTE;
        END;

VAR  x    :  x_type;
     i    :  INTEGER;

BEGIN

   WITH  x  DO
      i :=  INTEGER (ADDR(F2)) - INTEGER(ADDR(F1));

END.
```

The value of F1 is subtracted rather than the address of F1. This problem seems to be unique to the first element of the record.

Temporary solution:
Add a negative first element rather than subtracting it.

```
"68000"
PROGRAM  test;

$EXTENSIONS ON, EXTVAR ON$

TYPE    x_type  =   RECORD
           F1   :   BYTE;
           F2   :   BYTE;
        END;

VAR    x  :  x_type;
       i  :  INTEGER;

BEGIN
   WITH  x  DO
     i = - INTEGER(ADDR(F1)) + INTEGER(ADDR(F2));

END.
```

KPR #: 5000169250 Product: 68000 PASCAL 64815 01.11

One-line description:
Declaring a boolean array may cause an out of bounds error.

Problem:
The Pascal compiler generates an invalid out of bounds error for the following program.

KPR #: 5000169250   **CONTINUED**

"processor"

```
PROGRAM  OUTOFBOUNDS;
$RANGE+$

TYPE    I                 =       INTEGER;
        BOOL_ARRAY        =       ARRAY[1..2] OF BOOLEAN;

VAR     B_ARRAY           :       BOOL_ARRAY;
        B                 :       BOOLEAN;

BEGIN

        B   := B_ARRAY[1];
END.
```

If the array type is not boolean then this code compiles correctly. Also, you can put the array declaration first and the invalid error goes away.

Temporary solution:
Declare the boolean array first.

```
PROGRAM  BOOLEAN_ARRAY;

TYPE     BOOL_ARRAY       =       ARRAY[1..2] OF BOOLEAN;

VAR      B_ARRAY          :       BOOL_ARRAY;
         BOOL             :       BOOLEAN;
         INT              :       INTEGER;

BEGIN
   BOOL := B_ARRAY[1];
END.
```

KPR #: 5000183913   Product: 68000 PASCAL   64815   01.11

One-line description:
Casting address to int and adding a signed_16 var generates bad code.

Problem:
In the following program the address to integer conversion is not always done correctly. In particular if you cast an address to an integer and add a signed_16 variable incorrect code is generated.

"68000"

```
$EXTENSIONS ON$
$GLOBPROC ON$

PROGRAM TEST;
```

KPR #: 5000183913   **CONTINUED**

```
VAR   INT  :  INTEGER;
      SHORT:  SIGNED_16;


PROCEDURE  PROC(VAR PARAM: INTEGER);

VAR   INT1,INT2    : INTEGER;
      SHORT        : SIGNED_16;

BEGIN

   SHORT := 4;
   INT2  := 8;

   INT1  := INTEGER(ADDR(PARAM)) + SHORT;
END;

BEGIN
END.
```

Instead of adding the address of PARAM to SHORT we add the value of PARAM.

Temporary solution:
Cast the signed_16 variable, in this example SHORT, to an integer.

KPR #: 5000196428   Product: 68000 PASCAL   64815   01.11

One-line description:
Bytes sign extened in a case statement.

Problem:
The foloowing program generates wrong code on the 64000 (old) system (Seems to be correct on 300 series host)

```
"68000"
PROGRAM T;
$EXTENSIONS ON$;
VAR A : BYTE;

BEGIN
  CASE A OF
    BYTE(55H) : ;
    BYTE(0F5)H: ;
  END;
END.
```

"A" is extended to a word in 64000 Pascal (longword on 300) and the comp are is done on a word too (byte on 300).If A=0F5H then it is extended to 0FFF5H and compared with 0F5H, which fails.

Temporary solution:
Use a signed_16 variable for the test case.

```
$EXTENSIONS ON$
```

KPR #: 5000196428 **CONTINUED**

```
PROGRAM  TEST;

VAR   A:  SIGNED_16;

BEGIN

    CASE  A OF
      0F5H:  ;
    END;
END.
```

KPR #: 5000244368   Product: 68000 PASCAL   64815   01.05

One-line description:
Specific file causes pass three error when compiled. (Too many errors)

Problem:
pascal-68000 on hp9000/500 , 64815S/001, V1.5:
compiler generates a pass 3 error: too many errors.

but source did not contain any error, compiler generated listing has no error-messages. a x-ref can't be generated. the same program compiled with version 1.4 lists no error, but can't be used by customer, because this version is delimited by 128 include-files.

dave, see ux-mail gru.shar (xaa..xae) file g_curs for duplicating this error.
burkhard hp-ism/muc-north

Signed off 08/31/88 in release A02.00

KPR #: D200012104   Product: 68000 PASCAL   64815   01.08

Keywords: CODE GENERATOR

One-line description:
$DEBUG$ may cause undesired TRAPV.

Signed off 08/25/86 in release 03.01

KPR #: D200013359   Product: 68000 PASCAL   64815   01.08

Keywords: CODE GENERATOR

One-line description:
Compiler generates incorrect code for set inclusion check.

Problem:
The following program will display a compiler code generation problem when testing for SET inclusion.

```
  PROGRAM TEST;
  $EXTENSIONS$
```

KPR #: D200013359 **CONTINUED**

```
     TYPE
        DIG = SET OF '0'..'9';

     VAR
        DIGIT : DIG;
        DA : STRING;
        A, B : INTEGER;

     BEGIN
        DIGIT := DIG['0','1','2','3','4','5','6','7','8','9'];
        DA := 'AAA';
        IF DA[1] IN DIGIT THEN
     END.
```

Temporary solution:
As a temporary work-around use the following TYPE definition for DIG.

```
  TYPE
     DIG = SET OF CHAR;
```

KPR #: D200014332   Product: 68000 PASCAL   64815   01.09

Keywords: CODE GENERATOR

One-line description:
Bad code using $RANGE$ or $DEBUG$ with $CALL_PC_LONG$ or LIB_PC_LONG

Problem:
Bad code is generated when calling functions and the compiler directives $RANGE ON$ or $DEBUG ON$ are used in combination with the directives $CALL_PC_LONG$ or LIB_PC_LONG. For example,

```
   $DEBUG ON, LIB_PC_LONG$     VAR I:SIGNED_16;
   FUNCTION F:SIGNED_16; BEGIN F := 0; END;
   BEGIN   I := F * 2;  {PRODUCES BAD CODE}
        BSR   F            ;CALL F
        MULS  #2,D7        ;MULTIPLY RESULT IN D7 TIMES 2
        MOVE.L D7,-[A7] ;PUSH PARAMETER FOR Zoverflow_s16
        MOVE.L #Zoverflow_s16[PC],D7 ;ERROR!! D7 DESTROYED!!
        JSR   -6[PC,D7.L] ;CALL Zoverflow_s16 VIA PC LONG METHOD
        MOVE.W D7,I        ;WRONG VALUE STORED, D7 CONTAINS BAD DATA!!
```

Temporary solution:
Avoid the combination of functions, $RANGE$ or $DEBUG$, and $CALL_PC_LONG$ or LIB_PC_LONG. The example above may be rewritten to achieve the same functionality.

```
   I := F;   {STATEMENT DOES NOT CAUSE CALL TO OVERFLOW ROUTINE}
   I := I * 2;   {OVERFLOW ROUTINE CALLED HERE BUT DATA IS NOT IN D7}
```

KPR #: D200030585 Product: 68000 PASCAL 64815 01.09

Keywords: BOOLEAN

One-line description:
NOT(function) as boolean expression in "IF" statement doesn't work.

Problem:

```
"68000"
PROGRAM TEST;
FUNCTION X : BOOLEAN;EXTERNAL;
BEGIN
IF NOT X THEN ;     {THE RETURN VALUE IS NEVER TESTED.}
                    {COMPARE THE CODE TO:}
IF X THEN;
END.
```

Temporary solution:
Assign the function to an intermediate variable an test the variable.

Signed off 08/25/86 in release 03.01

KPR #: D200036749 Product: 68000 PASCAL 64815 01.09

Keywords: INCLUDE

One-line description:
Nested INCLUDE files 3 or more deep cause 64000 to "hang" in pass 3.

Problem:
Nested INCLUDE files 3 or more deep cause 64000 to hang in pass 3.

Temporary solution:
None at this time.

Signed off 08/25/86 in release 03.01

KPR #: D200036889 Product: 68000 PASCAL 64815 01.09

Keywords: PASS 2

One-line description:
K := K + K + K; causes too many pass 2 errors to continue.

Problem:

```
PROCEDURE TEST (VAR K : SIGNED_16);
BEGIN
K := K + K + K;     Causes 64000 to hang in pass 2.  Causes the HOST to
                    abort in pass 2 with too many errors.
```

Temporary solution:
None at this time.

Signed off 08/25/86 in release 03.01

KPR #: D200047423 Product: 68000 PASCAL 64815 01.09

One-line description:
TOO MANY ERRORS IN PASS 3 IF >127 PROCEDURES

Signed off 08/25/86 in release 03.01

KPR #: D200049882 Product: 68000 PASCAL 64815 01.10

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200050997 Product: 68000 PASCAL 64815 01.10

One-line description:
Program causes compiler to hang up.

Problem:
A program containing a complicated expression causes the compiler to hang up in pass 2. No listing file is created and no error message is generated.

KPR #: D200050997 **CONTINUED**

Temporary solution:
Break the complicated expression up into two or more simpler expressions.

Signed off 08/25/86 in release 03.01

KPR #: D200052563 Product: 68000 PASCAL 64815 01.10

One-line description:
Missing semicolon causes compiler to hang in Pass 1.

Problem:
The following code causes the 64000 to hang in pass 1. An error is generated on the hosts stating that parsing has stopped at a particular line number.

```
"BZ80"
PROGRAM MAIN;
TYPE
STRUCTURED= RECORD
            INT1:INTEGER;
            INT2:INTEGER;
            END;

PROCEDURE OUTER(VAR P1:STRUCTURED; VAR P2:INTEGER);
VAR I:INTEGER;
BEGIN
I:=P1       <--This missing semicolon causes the problem
I:=P1.2;
I:=P2;
END;

BEGIN
END.
```

Temporary solution:
If the compiler hangs, look for a statement without a semicolon. On the 64000, the status line will show which line of code it stopped on. On the hosts, the error message generated indicates which line of code parsing stopped on.

Signed off 08/25/86 in release 03.01

KPR #: D200053157 Product: 68000 PASCAL 64815 01.10

One-line description:
Real number library routine "ROUND" not working in some cases.

Problem:
The ROUND routine will not always work properly. I have below listed a sample program which I plugged different values in.

"68000"

KPR #: D200053157 **CONTINUED**

```
PROGRAM  ROUNDTEST;

VAR   rtemp,itemp,temp      :REAL;

BEGIN
   itemp := ROUND(rtemp);
END.
```

```
If  rtemp equals  5.5    itemp will equal  6.0
                  6.5                      6.0
                  7.5                      8.0
                  100.5                    100.0
```

Temporary solution:
No known work around at this time.

Signed off 08/25/86 in release 03.01

KPR #: D200053165 Product: 68000 PASCAL 64815 01.10

One-line description:
Library routine TRUNC will sometimes return wrong value.

Problem:
The below code will demonstrate the problem.

```
"68000"
PROGRAM  TRUNCTEST;
VAR   rtemp,itemp,temp        :   REAL;

BEGIN
   rtemp := temp;       {some value }
   rtemp := TRUNC(rtemp + 0.5);
END.
```

If rtemp is equal to 8388607 the result of TRUNC is correct. That is the answer is 8388607. However, at 8388607 and above the value of rtemp is altered by TRUNC.

The cases I tried were:

```
8388608.0 (4B00 0000)  after TRUNC  (4B00 0001)
8400000.0 (4B00 2C80)   " "    ""   (4B00 2C81)
8388609.0 (4B00 0001)  after TRUNC  (4B00 0002)
```

Temporary solution:
No known work around at this time.

Signed off 08/25/86 in release 03.01

KPR #: D200060103   Product: 68000 PASCAL   64815   01.10

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200060343   Product: 68000 PASCAL   64815   01.10

One-line description:
Compiler generates a LEA instruction with an illegal source operand.

Problem:
The following program causes an illegal 68000 instruction to be generated.

```
"68000"
PROGRAM  CGR_1;
$EXTENSIONS ON$


TYPE   REC     =      RECORD;
       REG0,
       REG1,
       REG2,
       REG3,
       REG4,
```

KPR #: D200060343 **CONTINUED**

```
       REG5,
       REG6,
       REG7:            UNSIGNED_8;
       END;

VAR    REC1,
       REC2:           REC;

       V1  :           UNSIGNED_8;

BEGIN
     REC1.REG0 := SHIFT(V1,5);
     REC2 := REC1;
END.
```

The 'REC2 := REC1' instruction causes a 'LEA D0,A0' 68000 instruction to be generated. This is an illegal instruction because the LEA instruction cannot have a data register as its source operand.

Temporary solution:
Turn $AMNESIA ON$ above the instruction which makes the structure access.

```
"68000"

PROGRAM  CGR_1;
$EXTENSIONS$

TYPE      REC      =         RECORD
                   REG0,
                    .
                    .
                    .
                   REG7:     UNSIGNED_8;
          END;

VAR       REC1,
          REC2:   REC;

          V1  :   UNSIGNED_8;

BEGIN
$AMNESIA OFF$

      REC1.REG0 = SHIFT(V1,5);
      REC2 := REC1;
$AMNESIA OFF$

END.
```

KPR #: D200065045 Product: 68000 PASCAL 64815 01.11

One-line description:
The WARN option cannot be turned off.

Problem:
The WARN option cannot be turned off on the hosts. With WARN off, the host machines (9000/500, 9000/320 and VAX) still generate warning messages on the screen and in the listing file. The 64100 will generate some warning messages (only the duplicate macro warning in the following example).

```
"processor"  PREPROCESS

$WARN OFF$

#define byte  SHORT
#define byte  UNSIGNED_8

PROGRAM GENERATE_WARN;

VAR      LSD:                                                        :INT

{The above is intended to generate warning 512 }

$LSD +$

BEGIN
END.
```

Temporary solution:
No temporary solution at this time.

KPR #: D200071696 Product: 68000 PASCAL 64815 01.11

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

KPR #: D200071696 **CONTINUED**

Temporary solution:
The only work around is to obtain the sources from the factory, remove the DATA pseudo, and reassemble.

KPR #: D200073007 Product: 68000 PASCAL 64815 01.11

One-line description:
Problems with routine STRWRITE & $BASE_PAGE$ mode with ASPIOLIB

Problem:
I/O error # 2 may occur when using STRWRITE in a program compiled with $BASE_PAGE$ option, in conjunction wtih using the the Pascal IO library ABSPIOLIB.

If the user allows the DATA area for Pascal program variables to be loaded in the memory range 00FF8000H to 00FFFFFFH, an unexpected run time I/O error #2 may occur.

When using the Pascal/64000 string function STRWRITE, the compiler generates a global variable STRfile and calls the routines Pstringopen and Pwrite_string. As part of its error checking mechanism, Pwrite_string performs an address comparison on the variable STRfile as passed by the caller (done as a sign extened SHORT address due to the $BASE_PAGE$ option [0FFFF6238H]) and as created by Pstringopen (done as a LONG address due to the FAR option used in the compililng the ABSPIOLIB routines [000FF6238H]). Since the addresses are found to be different the routine generates the I/O error #2 message.

This error can be avoided by always using the FAR option for any user program which wants to be used with the ABSPIOLIB.

More Information:

I/O error # 2 may occur when using STRWRITE in a program compiled with $BASE_PAGE$ option, in conjunction wtih using the the Pascal IO library ABSPIOLIB.

If the user allows the DATA area for Pascal program variables to be loaded in the memory range 00FF8000H to 00FFFFFFH, an unexpected run time I/O error #2 may occur.

When using the Pascal/64000 string function STRWRITE, the compiler generates a global variable STRfile and calls the routines Pstringopen and Pwrite_string. As part of its error checking mechanism, Pwrite_string performs an address comparison on the variable STRfile as passed by the caller (done as a sign extened SHORT address due to the $BASE_PAGE$ option [0FFFF6238H]) and as created by Pstringopen (done as a LONG address due to the FAR option used in the compililng the ABSPIOLIB routines [000FF6238H]). Since the addresses are found to be different the routine generates the I/O error #2 message.

KPR #: D200073007 **CONTINUED**

This error can be avoided by always using the FAR option for any user program which wants to be used with the ABSPIOLIB.

Workaround solution for this defect:

This error can be avoided by consistent use of the $BASE_PAGE$ and FAR options in user compiled code with the proper A5 or ABS libraries.

Always using the FAR option for any user program which will be used with the ABSPIOLIB.

Always using the $BASE_PAGE$ option for any user program which will be used with the A5_PIOLIB.

Temporary solution:
This error can be avoided by consistent use of the $BASE_PAGE$ and FAR options in user compiled code with the proper A5 or ABS libraries.

Always using the FAR option for any user program which will be used with the ABSPIOLIB.

Always using the $BASE_PAGE$ option for any user program which will be used with the A5_PIOLIB.

KPR #: D200076562 Product: 68000 PASCAL 64815 01.12

One-line description:
Subrange parameter not passed properly when function returning integer

Problem:
A multi - parameter procedure or function with a value parameter of type subrange may not be passed properly in $COMMON$ option mode when the parameter expression includes a function returning a subrange value.

The problem appears to be improper conversion of the 32 bit integer returned from the function and the truncation to 16 bits for the parameter passing.

###################### Problem source text ###################

The defect can be reproduced with a call to routine PARMS defined:

KPR #: D200076562 **CONTINUED**

```
PROCEDURE PARMS(SP:SUBRANGE; IP:INTEGER; RP:REAL; LP:LONGREAL);

FUNCTION FI1:INTEGER; BEGIN FI1:=1; END;

BEGIN
 PARMS(FI1,FI1,FI1,FI1);  {Fails to pass the first parameter properly}

END;
```

Temporary solution:
The workaround would seem to be performing a functional type change on the integer function call:

```
 PARMS( SUBRANGE(FI1) ,FI1,FI1,FI1);  {--Will pass the first parameter
                                            properly. -- }
```

KPR #: D200087320 Product: 68000 PASCAL 64815 01.90

Keywords: CODE GENERATOR PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
PROBLEM ON VAX NOT ON 64100 SYSTEM

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example:

```
"8086"
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
LABEL  100;
BEGIN
100:
     GOTO   100;
END;
PROCEDURE  TWO;      { pass 3 error - too many errors in pass 3 }
LABEL  100;          { is generated, without any indication as to }
BEGIN                { what the problem is                        }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get

KPR #: D200087320 **CONTINUED**

this error message, be aware that you may have duplicate labels in the program.

Signed off 08/31/88 in release A02.00

```
KPR #: D200087718   Product: 68000 PASCAL         64815            01.90
```

Keywords: PROBLEM ON 9000/S300

One-line description:
Array offset is incorrect if BOOLEAN is the data type for the indices.

Problem:
Using the boolean data type as an array indice causes incorrect code to be generated.

```
"68000" PREPROCESS

$EXTENSIONS ON$
$GLOBVAR ON$
PROGRAM BADCODE;

TYPE
  TS = SET OF 0...35;
  TA = ARRAY [BOOLEAN] OF TS;

VAR

  S:TS;
  A:TA;
  B: BOOLEAN;

BEGIN
   B:= TRUE;
   S:= A[B];   {THE OFFSET CALCULATED FOR THIS ARRAY ACCESS IS INCORRECT

END.
```

Temporary solution:
Use a subrange of the integer data type.

```
PROGRAM  BADCODE;

TYPE
   TS = SET OF 0..35;
   TA = ARRAY[0..1] OF TS;

VAR

   S:TS;
   A:TA;
   B:INTEGER;

BEGIN
```

KPR #: D200087718 **CONTINUED**

```
    B=1;
   S:=A[B];
END.
```

Signed off 08/31/88 in release A02.00

KPR #: D200031823 Product: 68000 SW ANALYZER 64341B 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.02

KPR #: 5000182006 Product: 68000/08 EMUL M 64242-90906 01.07

One-line description:
Monitor is not reentrant. Please document consequences and limitations.

Problem:
Because the monitor program is not re-entrant, the following combination may cause program crashes during emulation:

- A software breakpoint is reached and monitor code starts executing
- A user interrupt occurs (interrupts do not need to be re-enabled in the monitor, because the monitor enters via a TRAP which does not alter the interrupt mask)
- While processing the user interrupt, a break into the monitor occurs to service a display memory repetitively command

After returning from the NMI and the user interrupt the code executes in the monitor, but the register variables are no longer valid because they were last modified by the break to service the display memory.

KPR #: 5000169037 Product: 68000/08 EMULHP-UX M 64243-90903 01.00

One-line description:
Explain useage/definition in manual of INTR character to terminate wait

Problem:
IT WOULD BE HELPFUL TO HAVE THE FOLLOWING ADDED TO THE DESCRIPTION OF THE "wait" COMMAND IN THE "COMMAND SUMMARY" SECTION OF 64000_UX EMULATION MANUALS. PERHAPS IT SHOULD ALSO BE ADDED TO THE "CREATING AND USING COMMAND FILES" CHAPTER OF THE 64000_UX USER'S GUIDE.

IN THE PRESENT 64243 EMULATION MANUAL IT SAYS THAT A "wait" COMMAND WILL CAUSE COMMAND FILE EXECUTION TO WAIT FOR CONTROL C BEFORE ACCEPTING THE NEXT COMMAND.

IT WOULD BE VERY HELPFUL TO POINT OUT THAT THIS IS TRUE IF INTERRUPT CHARACTER ("intr") IS SET TO BE CONTOL C. THE DEFAULT IS "DEL". THE "stty -a" COMMAND CAN BE USED TO CHECK THIS SETTING.

Temporary solution:
This has been documented in the HP 64000-UX 16 bit emulation manual update (64200-90912 E0187 U0987). It will also be documented in the next update of the HP 64000-UX User's Guide.

Signed off 11/17/87 in release X01.01

KPR #: 5000173716 Product: 68000/08 EMULHP-UX M 64243-90903 01.00

One-line description:
Page 4-9 Fig 4-3 The first ORG statement should be UNCOMMENTED

Problem:
In the Getting Started chapter of the manual, on page 4-9 in figure 4-3, the FIRST ORG statement is commented.

Temporary solution:
In the figure, the first ORG statement should be uncommented. If this is not done, the desired results will not be obtained by the linking instructions given.

Fix information:
Fix is documented in Software Notice 5959-2128 R2707.

Signed off 08/05/87 in release 01.01

KPR #: 5000182519 Product: 68000/08 EMULHP-UX M 64243-90903 01.00

One-line description:
Single Stepping slow when in "display trace source only mode".

Problem:
This is an operating characteristic of our product. When "display trace source only" is selected, there will be a decrease in performance whenever the analyzer has captured only assembly language generated code.

Temporary solution:

KPR #: 5000182519 **CONTINUED**

This has been documented in the HP 64000-UX 16 bit emulation reference manual update pg 3-44 (64200-90912 E0187 11087). It will also be documented in the next update of the HP 64000-UX User's Guide.

Signed off 11/17/87 in release X01.01

KPR #: 5000125229 Product: 68000/08/10 ASM M 64845-90904 01.09

One-line description:
RORG ONLY WORKS IF THE CODE IS NOT PC INDEPENDENT.

Problem:
If you use RORG around code that is relocatable, an invalid operand error is flagged.
For example:

```
"68000"

                RORG
                MOVE     D0,D1         ; NO PROBLEM
                MOVE     LABEL[PC]     ; IO ERROR GENERATED.
```

Temporary solution:
Turn RORG off around sections of code that are already relocatable.
For example:

```
"68000"

         RORG
         MOVE           D0,D1
         NO_RORG
         MOVE          LABEL[PC]
LABEL
```

Fix information:
Fix is documented in Software Notice 5958-8824 R2707.

Signed off 08/05/87 in release 01.10

KPR #: 5000239012 Product: 68000/08/10 ASM M 64845-90904 01.10

Keywords: MANUAL

One-line description:
Alter all assembler manuals to reflect new syntax.

KPR #: 5000242032 Product: 68000/08/10 ASM M 64845-90904 01.10

Keywords: MANUAL

One-line description:
Manual indicates EXT is a legal psuedo for an external declaration.

Problem:
The assembler outputs an error when assembling:

```
"68000"
       EXT        L1
                   ^
                error -IO
```

KPR #: 5000242032 **CONTINUED**

The 68000 assembler manual states EXT is an acceptable pseudo-op.

Temporary solution:
Use "EXTERNAL" rather than "EXT" when making external symbol declarations.

KPR #: D200045864 Product: 68000/08/10 ASM M 64845-90904 01.00

One-line description:
Wrong offset calculated when using PC+index reg+ offset mode of addr.

Problem:
When using the PC relative with offset and index register mode of addressing the assembler may generate a legal range error. The error will be made if the offset symbol is at an absolute location greater than FFH. The correct address is generated.

```
"68000"

     ORG     010H
     MOVE    #0,D0
     JMP     TABLE[PC,D0]
     ORG     100H
TABLE        DS.W     10
```

Temporary solution:
No temporary solution.

Duplicate Service Requests: D200045351

KPR #: D200055947 Product: 68000/08/10 ASM M 64845-90904 01.08

One-line description:
Incorrect opcode is generated for Move from CCR instruction.

Problem:
The 68000 assembler generates an incorrect opcode for the Move from CCR instruction.

```
"68000"


        MOVE    CCR,D0
```

A F0C1 is generated rather than a F2C1.

Temporary solution:
The opcode that is generated is the opcode for a Move from SR instruction. A move from SR will still move the CCR as the user expects.

Fix information:
Fix is documented in Software Notice 5958-8824 R2707.

Signed off 08/06/87 in release 01.20

KPR #: D200062521 Product: 68000/08/10 ASM M 64845-90904 01.00

Keywords: MANUAL

One-line description:
The MOVE example for program counter with index address mode is wrong.

Problem:
In Chapter 2, Special Address Modes, Program Counter with Index, the example:

```
MOVE  SAM[D4.L],CHARLIE
```

is incorrect.

Temporary solution:
The example should read:

```
MOVE SAM[PC,D4.L],CHARLIE
```

KPR #: D200065565 Product: 68000/08/10 ASM M 64845-90904 01.00

One-line description:
Include support for BHS and BLO.

Problem:
BHS should assemble the same as BCC and BLO the same as BCS. This is more logical for the programmer and matches Motorola's assembler.

Temporary solution:
No temporary solution.

KPR #: D200065607 Product: 68000/08/10 ASM M 64845-90904 01.00

One-line description:
Clear up confusion on correct symbols.

Problem:
The manuals could give a better discription of the mneumonics and perhaps describe the differences between HP's assembler and Motorolas. Two specific examples:

External references must be declared "EXTERNAL" not "EXT".

The manual states that all lables must begin with an upper case alpha character. This is not true, both "_" and small alpha characters are accepted.

Signed off 12/02/86 in release 01.01

KPR #: 5000214148 Product: 68000/10 EMUL M 64245-90903 01.00

One-line description:
Manual has error on fig 3-1, should be "cp" and "sample.*".

Fix information:
The documentation has been fixed manual part number 65245-90901 print date Feb87 E0287, on page 3-2 located on bottom of page.

Signed off 08/18/87 in release 01.01

KPR #: 5000163808   Product: 68000/10 RT S-ANAL M 64341-90903   02.00

One-line description:
Non-adjacent symbols not traceable in some conditions.

KPR #: 5000291914   Product: 68000C AXLS COMP   M 64902-90901   01.00

One-line description:
Fig. 4.1 does not compile as the manual indicates.

Problem:
AxLS C Compiler does not cause the optimization for space.
I can not duplicate the example on the page 4-5 of "68000 C Compiler Operating Manual".
Although the "optimize" option for time is not appended, the compiler always generates the optimized code for time instead of space.

Temporary solution:
There is no workaround needed as this is a problem with the manual.

KPR #: 1650055251 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
Option for inverse C code only works if it is specified first.

Problem:
The option for surrounding the C source statements in the output listing with inverse video escape sequences only works if it specified as the first option.

Eg:

```
cc68000 -Lac -WL,ti  simple.c    #OK will generate inverse video

cc68000 -Lac -WL,it  simple.c    #BAD won't   "        "       "
```

Temporary solution:
There is no workaround available.

KPR #: D200083584 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
When cpp macros expansion includes new-line, debug line numbers get off.

Problem:
In the absence of -O and -s, the -g flag is passed to cpp68000 which causes it to pass information to ccom68000 regarding macro expansions (namely, what the original text was and what it was replaced with). When this information is being passed and the original text of the macro invocation includes a new-line, ccom68000 counts that new-line as a line in the original source file but cpp68000 does not. This causes ccom68000's line counting mechanism to be off-by-one until the next #line directive from cpp68000 is received.

Temporary solution:
First of all, this defect is only noticeable when accessing line numbers in the emulator in the vicinity of a macro expansion which included a new-line character in its invocation. Should this ever occur, it can be worked around by using the disired line number minus one.

Signed off 04/29/88 in release A01.10

Duplicate Service Requests: D200083576

KPR #: D200085423 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
All variables following an ORG are put at same address.

Problem:
All variables following an ORG are put at same address.

If you ORGed several variables beneath one section pragma they are all given the starting address.

```
#pragma  SECTION  DATA=0x1000
```

KPR #: D200085423 **CONTINUED**

```
char   letter;
char   number;
int    count;

#pragma SECTION UNDO

main()
{

int i;

   i = count;
}
```

Signed off 04/29/88 in release A01.10

KPR #: D200085431 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
Operations on bitfields > 17 bits wide may fail.

Problem:
The runtime routines bfget6/bfget6u return the value of a signed/unsigned bitfield. This value will then be used as an operand for some operation. Under most circumstances neither of these 2 routines will return the correct value from the bitfield. A bitfield declaration of width 18-32 bits may cause one of these routines to be used in getting that field's value.

Temporary solution:
Make sure that the compiler, cc68000, does not generate calls to either bfget6 or bfget6u. This may accomplished in either of the following 2 ways.

1.) Do not use bitfields which are wider than 17 bits. By limiting the width of the fields used to being no wider than 17 bits, one can be certain that no calls will be made to either defective routine.

2.) Use bitfields of any width, but adhere to one of the following constraints when defining a field which is wider than 17 bits.

    a.) Make the field the first defined within a structure.

    b.) If the field cannot be the first defined or there are multiple fields which are wider than 17 bits then the workaround is as follows:
    Force the field to be allocated memory within 4 bytes of the preceding word boundary. For example, the following structure declaration will yield the following memory allocation.

KPR #: D200085431 **CONTINUED**

```
struct  /* Allocation will begin on word boundary */
{
        int     a: 26;
        int     b: 22;
        int     c: 4;
} field;

Address           Memory
field+8  ----------------------------------
         |c|c|c|c| | | | || | | | | | | | |
field+6  ----------------------------------
         |b|b|b|b|b|b|b|b||b|b|b|b|b|b|b|b|
field+4  ----------------------------------
         |a|a|a|a|a|a|a|a||a|a|b|b|b|b|b|b|
field+2  ----------------------------------
         |a|a|a|a|a|a|a|a||a|a|a|a|a|a|a|a|
field+0  ----------------------------------
```

Here, field.a is the first declared field within the structure, ensuring correct operations which involve field.a.
Also, field.b is contained within the first 4 bytes from the word-aligned address field+2, ensuring correct operations involving field.b.

Signed off 04/29/88 in release A01.10

KPR #: D200085449   Product: 68000C AXLS COMP 300 64902S004   01.00

One-line description:
Compiler won't take address of a function pointer.

Problem:
The compiler issues a warning that it is ignoring the & operator when it is applied to a function pointer. The following code would result in an error message about the assignment since the address of 'fp' has not been taken.

```
int (*fp)();
int (**fpp)();

main()
{
        fpp = &fp;
}
```

Signed off 04/29/88 in release A01.10

KPR #: D200086918   Product: 68000C AXLS COMP 300 64902S004   01.00

One-line description:
Compiler erroneously constant folds pointer offsets

Problem:
Text:

KPR #: D200086918 **CONTINUED**

  Compiler erroneously constant folds pointer offsets.

The compiler would error when trying to handle negative pointer offsets from constant pointer values.

```
int a[10];

main()
{
        a[-1] = 1;
}
```

This would produce an error.

Signed off 04/29/88 in release A01.10

KPR #: D200086926   Product: 68000C AXLS COMP 300 64902S004   01.00

One-line description:
Function entry code over writes stack when widened param is not passed.

Problem:
Text:
  Function entry code overwrites stack when widened param is not passed.

Code is generated to unwiden params at function entry when the param is a widened type. This code will trash the stack frame if the parameter was not actually passed.

Signed off 04/29/88 in release A01.10

KPR #: D200089417   Product: 68000C AXLS COMP 300 64902S004   01.00

One-line description:
#progma BBA_IGNORE still produces warrnings

Problem:
Detailed Listing for Defect Number LSDqf03370
Text:
  #pragma BBA_IGNORE still produces warnings

The following pragmas still generate warnings in the AxLS compilers:

```
#pragma BBA_IGNORE
#pragma BBA_ALERT
```

Although this does not result in any problems, it does look ugly.
                                    - Bruce Erickson

Temporary solution:
There is no workaround needed.

KPR #: D200089516 Product: 68000C AXLS COMP 300 64902S004 01.00

One-line description:
Runtime and support libraries contain loadtime initializers.

Problem:
Compiler libraries lib.a, libpi.a and env.a contain loadtime initializers. The lack of runtime initialization in an embedded system may cause library routines to behave unexpextedly.

Signed off 04/29/88 in release A01.10

KPR #: D200069518 Product: 68008 EMULATION 300 64244S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Signed off 08/28/87 in release 99.99

KPR #: D200072470 Product: 68008 EMULATION 300 64244S004 01.00

One-line description:
Incorrect breakpoint behaviour on continuing emulation.

Problem:
When using software breakpoints, and doing the following sequence, emulation does not behave as expected.

```
run LOOP
modify software_breakpoints set LOOP
# breaks into monitor, displays breakpoint and clears breakpoint
run
end  locked
# return to emulation
modify software_breakpoints set LOOP
# breaks into monitor, but does not display or clear breakpoint
# at second iteration, things return to normal.
```

Signed off 08/28/87 in release 99.99

KPR #: D200078170 Product: 68008 EMULATION 300 64244S004 01.00

One-line description:
Questions not asked when switch from real-time to non-real.

Problem:
Detailed Listing for Defect Number LSDqf00461

```
  Submission Number: 00469LSDqf     Date Found:    870702
  Defect Status: RESOL              Date Arrived:  870702
  Prod/SCMS:64244 (68008DP Emul)    Date Received: 870706
  Version : 3.82                    Date Resolved: 870313
  Severity: 4
```

KPR #: D200078170 **CONTINUED**

Showstopper: No
Workaround: No
Defect/Enhancement:
* defect

Number of Duplicates:
Additional Files: 2

Text:
Questions not asked when switch from real-time to non-real.

Submitter Supplied Information

Submitter name: Kathy Moser
Submitter phone:
Submitter address: km
Activity used to find defect: casual use

Responder Supplied Information

Responsible site: LSD
Responsible project: stars
Responsible engineer: STARS II

Resolution:
? ot (unknown resolution code)

SCMS Project: /lsd/p2/cmd/emul/m68008dp
SCMS proj version fix: 4.0

.submitter

When modifying an emulation configuration and switching from restricted to real time to NOT restricted to real time, the simulated I/O and simulated interrupt configuration questions are not asked. You must be modifying an existing configuration file.

.labnotes

File changed:

/lsd/p2/cmd/emul/m68008dp/config.qs

Signed off 08/28/87 in release 99.99

KPR #: D200078964 Product: 68008 EMULATION 300 64244S004 01.00

One-line description:
EBPP as analyzer fails intermittently

Signed off 08/28/87 in release 99.99

KPR #: D200087213 Product: 68008 EMULATION 300 64244S004 01.20

One-line description:
Emulator does not work reliably with 64155b Memory

Problem:
Emulator does not work properly with 64155B memory control card.

Signed off 08/19/88 in release A01.30

KPR #: D200088328 Product: 68008 EMULATION 300 64244S004 01.20

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

Signed off 08/19/88 in release A01.30

KPR #: D200090837 Product: 68008 EMULATION 300 64244S004 01.20

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then

KPR #: D200090837 **CONTINUED**

moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/19/88 in release A01.30

KPR #: D200091116 Product: 68008 EMULATION 300 64244S004 01.20

One-line description:
Multiple commands on the line after "set" will not be executed

Problem:
Multiple commands after "set" shall not be executed.

Signed off 08/19/88 in release A01.30

KPR #: D200091132 Product: 68008 EMULATION 300 64244S004 01.20

One-line description:
Enabling DMA to emulation memory halts emulator.

Problem:
If the emulation configuration question "Enable DMA transfers to emulation memory" is answered "yes" the emulator releases from reset immediately into a "halt" state.

This problem appears always; whether a target system is hooked up or not.

Signed off 08/19/88 in release A01.30

KPR #: D200069567 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200077545 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
State inverse assembler for 6801 does not work

Problem:
Todd Hatfield@Logic Systems Division

The problem with the state inverse assembler for the 6801 has been fixed by the lab engineer. The software being shipped currently with the product has been fixed to work. The bad software which was shipped will be updated with an SMS shipment in the latter part of the summer or the first part of the fall, 1987.

Duplicate Service Requests: D200077537

KPR #: D200080671 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080978 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081927 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082255 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

KPR #: D200082255 **CONTINUED**

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082727 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Under certain conditions the 6801 may not work correctly with SPA

Problem:
In certain configurations SPA doesn't work with the 68XX series emulators.

There was a similar problem with the JLO 68XX clones, the 63XX series. They implemented a software fix to take care of this problem. The lab is implementing the same fix for the 68XX series.

KPR #: D200083261 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086058 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                             ; end locks the emulation session
3. <system name> <module name>     ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the

KPR #: D200086058 **CONTINUED**

problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200086397 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088351 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090886 Product: 6801/3 EMULATION 300 64256S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles,

KPR #: D200090886 **CONTINUED**

relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200068874 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
simulated I/O, Ints can't be modified when changing from rtrt to nrtrt

Problem:
When running restricted to real time, if you change to NOT restricted to real-time, the simulated I/O and simulated Interrupts configuration questions are not asked while modifying the configuration.

Temporary solution:
Use the vi or sk editor to edit the ASCII version of the emulation configuration file. This file is denoted by the .EA extension.

Signed off 08/28/87 in release 99.99

KPR #: D200069526 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Signed off 08/28/87 in release 99.99

KPR #: D200072488 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
Incorrect breakpoint behaviour on continuing emulation.

Problem:
When using software breakpoints, and doing the following sequence, emulation does not behave as expected.

```
run LOOP
modify software_breakpoints set LOOP
# breaks into monitor, displays breakpoint and clears breakpoint
run
end  locked
# return to emulation
modify software_breakpoints set LOOP
# breaks into monitor, but does not display or clear breakpoint
```

KPR #: D200072488 **CONTINUED**

```
# at second iteration, things return to normal.
```

Signed off 08/28/87 in release 99.99

KPR #: D200078055 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
Ehalt occurring too often while running user code.

Signed off 08/28/87 in release 99.99

KPR #: D200078220 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
SPA emulator title exceeds 9 character limit and will be truncated

Signed off 08/28/87 in release 99.99

KPR #: D200078956 Product: 68010 EMUL 12.5M 300 64245S004 01.00

One-line description:
EBPP as analyzer fails intermittently

Signed off 08/28/87 in release 99.99

KPR #: D200081844 Product: 68010 EMUL 12.5M 300 64245S004 01.10

One-line description:
Function Code not correct for user mem. access during display registers

Problem:
When single stepping through target memory, the read of target memory to perform mnemonic disassembly during "display registers" is performed using a Function Code of 000B. This is an "undefined (reserved)" function code for the 68010 processor. The correct function code should be 110 (Supervisor Program) or 010 (User Program) for access to program memory. This customer's target system is sensitive to the function code signals and does not complete memory cycles successfully with a function code of 000b.

Temporary solution:
The following modification to the monitor will test for a function code of 000b and modify it to 110 (or 010). This workaround will only work for target systems that would not normally use a function code of 000b.

/usr/hp64000/monitor/mon_68010.S

```
566  *   GET SOURCE FUNCTION CODE FROM UPPER 3 BITS
567       MOVE.L    SRC_ADDR,D2
568       ASR       #8,D2
569       ASR       #5,D2
NEW       ANDI.L    #00000007H,D2 ;CHECK FOR FC=000B
NEW       BNE       FC_OK         ;FC NOT ZERO, SO LEAVE AS IS
```

KPR #: D200081844 **CONTINUED**

```
NEW       MOVE.B    #110B,D2       ;SET FC TO SUPERVISOR PROGRAM
NEW                                ;(OR 010B FOR USER PROGRAM)
NEW  FC_OK
570       MOVEC     D2,SFC
```

Signed off 08/19/88 in release A01.30

KPR #: D200088336 Product: 68010 EMUL 12.5M 300 64245S004 01.20

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

Signed off 08/19/88 in release A01.30

KPR #: D200090845 Product: 68010 EMUL 12.5M 300 64245S004 01.20

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200090845 **CONTINUED**

Signed off 08/19/88 in release A01.30

KPR #: D200091124 Product: 68010 EMUL 12.5M 300 64245S004 01.20

One-line description:
Multiple commands on the line after "set" will not be executed

Problem:
Multiple commands after "set" shall not be executed.

Signed off 08/19/88 in release A01.30

KPR #: D200091140 Product: 68010 EMUL 12.5M 300 64245S004 01.20

One-line description:
Enabling DMA to emulation memory halts emulator.

Problem:
If the emulation configuration question "Enable DMA transfers to emulation memory" is answered "yes" the emulator releases from reset immediately into a "halt" state.

This problem appears always; whether a target system is hooked up or not.

Signed off 08/19/88 in release A01.30

KPR #: D200069534 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200072496 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Incorrect breakpoint behaviour on continuing emulation.

Problem:
When using software breakpoints, and doing the following sequence, emulation does not behave as expected.

```
run LOOP
modify software_breakpoints set LOOP
# breaks into monitor, displays breakpoint and clears breakpoint
run
end  locked
# return to emulation
modify software_breakpoints set LOOP
# breaks into monitor, but does not display or clear breakpoint
# at second iteration, things return to normal.
```

KPR #: D200080648 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080945 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081893 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082222 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

KPR #: D200082222 **CONTINUED**

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082776 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Memory breaks during stepping are not detected

Problem:
Memory breaks (write to ROM, etc.) which occur while stepping will not be noted on the status line.

Temporary solution:
No workaround at this time.

KPR #: D200083238 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086025 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

KPR #: D200086025 **CONTINUED**

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200087288 Product: 68010 G.P. EMUL 300 64249S004 01.00

Keywords: BREAKPOINT

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

Temporary solution:
Do not use software breakpoints in target memory. If software breakpoints are required to debug a certain section of code, make sure that section is mapped to emulation memory.

KPR #: D200090852 Product: 68010 G.P. EMUL 300 64249S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200031781 Product: 68010 SW ANAL 64334 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.03

KPR #: D200031849 Product: 68010 SW ANALYZER 64341D 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.02

KPR #: 5000199059 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Packed BCD constants cause incorrect code to be generated.

Problem:
The ".P" operator causes incorrect code to be generated. Example:

```
    FMOVE.P  #1.234567898765,FP0

generates:   3FF3 C0CA 44DE 319E 00D5 0000   as the PACKED DECIMAL
REAL constant. It should be:
             0000 0001 2345 6789 8765 0000
```

It appears that anytime a constant is used with a .P qualifier incorrect code is generated.

Temporary solution:
Don't use .P qualifier.

Signed off 12/07/87 in release Z01.10

KPR #: 5000285742 Product: 68020 ASSEMB 300 64870S004 01.00

Keywords: PROBLEM ON 9000/S300

One-line description:
NOPAGE option does not work for the 68000 assembler.

Problem:
This is the 68000 assembler directive option problem.
Please refer 68000/10/20 Assembler/Linker/Librarian reference manual.
(Manual part number : 64870-90901)
Chapter-6 assembler directive : 'NOPAGE' option could not work.
This option used , but all page eject and page headers were printed.
'NOPAGE' option were entered inthe file in the second(operation) field.

Temporary solution:
There is no workaround available at this time.

KPR #: D200089276 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Using asm psued END with numeric expression causes linker error.

Problem:
If you use a numeric expression in the assembler END pseudo the linker reports error 318.

```
     SECT  prog
main:  move.l d0,d1
     END  $1000
```

Temporary solution:
Rather than using a numeric expression use the form

```
                END    LABEL
```

KPR #: D200089276 **CONTINUED**

and then load that file starting at the address you wish.

KPR #: D200089714 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Ar68k can not handle long list in command line options.

Problem:
DETAILED DESCRIPTION: Ar68k allows the -a, -d, -r, and -e options on the command line. These options may be followed by a list of files or modules.
If the list becomes too long OR if too many of the above options are issued
than ar68k acts badly, usually core dumps or sometimes giving a spurious error. The list seems to be limited to a few hundred characters

It would be better if a very long list were allowed. This would allow an easier interface with "make" as shown in the following example.

```
MODLIST = very long list of modules ...

all:    $(MODLIST)
        rm -f lib.a
        ar68k -a "$(MODLIST)" lib
```

The following shell script illustrates the problem. It makes about 30 modules with long names. It then tries to archive them using the command line.

A test of the fix could be done using the script below and then comparing
the librarian listing.

```
list1=
list2=
for i in 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21
22 23
4 25 26 27 28 29 30 31 32
do
        list1="$list1 ar68klist$i.o"
        list2="$list2 -a ar68klist$i.o"
        as68k -o ar68klist$i <<EOF
        sect    a,,c
        xdef    g$i
g$i     dc.w    0
        end
EOF
done
        rm -f lib.a
        # try it as one option with long list
        echo $list1
        ar68k -a "$list1" lib
        # try it as many options with short list
```

KPR #: D200089714 **CONTINUED**

```
        echo $list2
        rm -f lib.a
        ar68k $list2 lib
```

Temporary solution:
Workaround: Create a libraian command file. For example,

```
MODLIST = very long list of modules ...

all: $(MODLIST)
        rm -f lib.a
        echo create lib > arcmd
        for i in ar*.o ;\
        do echo addmod $i >> arcmd ;\
        done
        echo save >> arcmd
        ar68k < arcmd
```

KPR #: D200089722   Product: 68020 ASSEMB   300 64870S004   01.00

One-line description:
PLEN directive does not work properly

Problem:
DETAILED DESCRIPTION: The PLEN directive is supposed to adjust the number of lines in a listing page. The manual says that the number specified in the directive should be the total number of lines on the page, including headings. Two things are wrong.

1. The number specified does not include heading lines. If I specify PLEN 50 then I get about 54 lines per page, 49 lines of source and 5 heading lines.

2. If I specify a PLEN greater than 55, then the first page is short. Subsequent pages are OK obeying the rule stated in number 1 above.

Temporary solution:
None.

KPR #: D200089730   Product: 68020 ASSEMB   300 64870S004   01.00

One-line description:
LLEN directive does not work properly with tab characters

Problem:
DETAILED DESCRIPTION: The LLEN directive is used to specify the width of the listing. However, when the source file contains tab characters, these tabs are counted as 1 character for the purposes of truncation. The output device usually expands these tabs into more than one character. This ma

KPR #: D200089730 **CONTINUED**

kes
the listing wider than specified sometimes causing jamming problems with certain printers.

Temporary solution:
None.

KPR #: D200089748   Product: 68020 ASSEMB   300 64870S004   01.00

One-line description:
Temporary files should be created in /tmp directory

Problem:
DETAILED DESCRIPTION: Ld68k (and perhaps other tools) create temprorary files in the pwd.

1. If the program terminates abnormally (core dump) than the temporary file is left around. (A interrupt and some kill signals seem to clean up properly.

2. The program cannot be executed in a directory without write permission (where the output files are created elsewhere).

We would like as68k, ld68k, and ar68k to create ALL temporary files in /tmp. Tmpfile(3S) and tmpname(3S) are available to expedite this process on HPUX.

Temporary solution:
None.

KPR #: D200089763   Product: 68020 ASSEMB   300 64870S004   01.00

One-line description:
Incremental link and strip results in corrupted relocatable

Problem:
When performing an incremental link (-i option) in combination with the strip flag (-f nos), the resulting IEEE relocatable file is corrupt. The corruption seems to be in the IMAGE part.

The fix may be tested in the incremental link directory by comparing ieee relocatable files. Example files follow.

```
-------------------------- eco242a.s ----------------------------
        sect    a
        xdef    ga
        xref    gb
ga      dc.l    gb
        end
-------------------------- eco242b.s ----------------------------
```

KPR #: D200089763 **CONTINUED**

```
        sect    a

        xdef    gb
        xref    ga
gb      dc.l    ga
        end
-------------------------- eco242.lc ----------------------------
* strip & incremental link produce corrupt relocatable output
* incremental link (-i option) must be specified on command line
nlist s
load eco242a,eco242b
end
------- partial prnieee dump of resulting relocatable file -------
PRNIEEE: Printer version 3.0, reader version 3.0.0
         Reading file eco242.o
...
IMAGE LOAD PART:
----------------
(0142) SB : Current section for loading is L01.

       VVVVV !!!!!!!!!!!
(0144) Record warning: (01) 0144 AS: extra fields found at end of record
.(0144
 ASP: Set load address for section L01 to (R01 + 0000)
(014D) ASR: R01 Base offset is (R01 + 0004)
(0154) SB : Current section for loading is L01.

       VVVVV !!!!!!!!!!!
(0156) Record warning: (01) 0156 AS: extra fields found at end of record
.(0156
 ASP: Set load address for section L01 to (R01 + 0000)
(015F) ASR: R01 Base offset is (R01 + 0004)
...
```

Temporary solution:
None.

KPR #: D200089771 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Reference to label in empty section causes ld68k error

Problem:
Detailed Listing for Defect Number LSDqf03193

ONE LINE DESCRIPTION: Ref. to label in empty sect. causes ld68k error.

DETAILED DESCRIPTION: Originally Microtec's eco #243. I am entering this so that I can keep track of it in the usual way.

An empty section is one which does not contain any code or data. It may, however, contain a label. If code in some other section refers to the label in the empty section, then ld68k generates an INTERNAL ERROR (318) when attempting to link the resulting relocatable file.

KPR #: D200089771 **CONTINUED**

The problem is seen while linking while the fix, I am told, will be made in the assembler. The fix can be tested in either of two ways. By comparing the relocatable files produced by the assembler or by comparing linker listing files.

DEFECT OWNED BY: Paul Malek

Temporary solution:
There is no known workaround.

KPR #: D200089789 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Section mismatch causes bad info in HP link_sym file

Problem:
Section mismatch causes bad info in HP link_sym file

Ld68k flags a "Section Mismatch" when a global symbol
is defined in one section and referenced using a different section name.
Under certain conditions when this warning occurs, there is bad informat
ion
in the HP link_sym file. Specifically, there is a bad "memory space"
record for the referencing module.

This bad memory space record has a range from 0x00000000 thru 0xffffffff
.
This causes problems for the HP emulators because they think the whole memory belongs to this module.

The only way to explain this is with several examples...

Temporary solution:
None.

KPR #: D200089888 Product: 68020 ASSEMB 300 64870S004 01.00

One-line description:
Embedded assembly code will not substitute defined variables correctly.

Problem:
The following code will result in a ERROR:(525)

```
#define USR_STATE,SR   DFFFH
.
.
#include "system/local/pragma_asm_h"
   ANDI   #USR_STATE,SR;
```

the assembler will substitute this way:

```
   ANDI   #$DFFFUSR_STATE,SR;
```

and generate a message:

KPR #: D200089888 **CONTINUED**

```
  ERROR:(525) Invalid character
```

Temporary solution:
This feature will be included in later Rev's of the product.
Until then, use the EQU directive for embedded assembly.

KPR #: D200090449 Product: 68020 BBA 300 64381S004 01.10

One-line description:
Complex conditional assignment delcarations cause bbacpp to core dump

Problem:
Detailed Listing for Defect Number LSDqf03782
Text:
complex conditional assignment declarations cause bbacpp to core dump

Some complex conditional assignments will cause bbacpp to core dump if they are in declaration statements. For example:

```
main()
{
   int digit = 6 - ((__ctype[c]&01) ? '0' : ((__ctype[c]&02)
                 ? 'A' - 10)
                 : ('a' - 10));
}
```

Causes a problem.

Temporary solution:
Move the assignment statement out of the declaration:

```
main()
{
    int digit;

    digit = 6 - ((__ctype[c]&01) ? '0' : ((__ctype[c]&02)
               ? 'A' - 10)
               : ('a' -10));
}
```

KPR #: D200090456 Product: 68020 BBA 300 64381S004 01.10

One-line description:
Switch statement followed immediately by a label cases bbacpp to fail

Problem:
Detailed Listing for Defect Number LSDqf03784
Text:
switch statement followed immediately by a label cases bbacpp to fail

If a switch statement of the form

```
   switch(var)
   label:
      if (alpha)
      {
         case 5 :
         alpha++;
      }
```

is encountered, bbacpp will generate incorrect code that changes the logic of the program. Note that this is a *very* unusual construct.

KPR #: D200090456 **CONTINUED**

Temporary solution:
Place braces around the label:

```
   switch(var)
   {         /* added brace */
   label:
      if (alpha)
      {
         case 5 :
         alpha++;
      }
   }         /* added brace */
```

KPR #: D200090464   Product: 68020 BBA   300 64381S004   01.10

One-line description:
A switch statement with no statement causes bbacpp to fail

Problem:

Detailed Listing for Defect Number LSDqf03785
Text:
a switch statement with no statement causes bbacpp to fail

If a switch statement of the form

```
        switch(5) ;
```

is encountered by bbacpp, bbacpp will issue an incorrect warning and refuse to continue parsing the file.

Note that this statement is somewhat non-sensical; there are no 'case' statements for the switch to go to.

Temporary solution:
No workaround is needed.

KPR #: D200086801   Product: 68020 EMUL   300 64410S004   02.00

One-line description:
"at_execution run" may fail to run upon execution,

Problem:
"at_execution run ..." may ignore a subsequent execute command and fail to initiate a run.

Please contact the factory if you encounter this problem.

```
        Joanne Carlson (719) 590-5840
                         -or-  590-5576
```

KPR #: D200091306   Product: 68020 EMUL   300 64410S004   02.00

One-line description:
Leading comma in some addtess indirect assembly is not needed

Problem:
The leading comma in some indirect address disassembly is not needed.

Example code listing:

```
*** ORIGINAL SOURCE FILE ***
     CHIP 68020
     ORG 0000H

     CLR.B          ([A0],$12345678)
     CLR.B          ([A0.W],$12345678)

*** ASSEMBLY LISTING ***
1                                     CHIP 68020
2                                     ORG 0000H
3
4    00000000 4230 8193 1234          CLR.B ([A0],$12345678)
              5678
5    00000008 4230 8193 1234          CLR.B ([A0.W],$12345678)
              5678

*** DELTA 68020 INVERSE ASSEMBLY ***
         0  42308193+  CLR.B          ([,A0.W],$12345678)
         8  42308193+  CLR.B          ([,A0.W],$12345678)

   ^
```

An unnecessary comma is displayed when no offset is present.

Temporary solution:
There is no workaround available.

KPR #: 5000198952 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
FMOVE instructions are disassembled incorrectly.

Problem:
The 68020 disassembler does not properly disassemble the FMOVE instruction. For example, the assembly listing file displays the following:

```
00006000  F23C 4400 3F9D       FMOVE.S  #1.234,FP0
          F3B6
00006008  4E71                 NOP
0000600A  4E71                 NOP
```

But, when displaying memory mneumonic, the following is displayed:

```
6000 F23C4400+  FMOVE.S   #$3F9D,FP0
6006 F3B6       rsvd 68881 instr type
6008 4E71       NOP
600A 4E71       NOP
```

Note that the FMOVE.S instruction is a 4 word instruction. But, the disassembler only uses the first 3 words of the instruction, and treats the fourth word as an additional instruction. The same failure occurs when using the FMOVE.D and FMOVE.X instructions.

The FMOVE.W and FMOVE.L instructions also disassemble incorrectly. For example, the assembly listing file displays the following:

```
0000603A  F23C 5000 1234       FMOVE.W  #$1234,FP0
00006040  4E71                 NOP
00006042  4E71                 NOP
00006044  F23C 4000 1234       FMOVE.L  #$12345678,FP0
          5678
0000604C  4E71                 NOP
0000604E  4E71                 NOP
```

But, when displaying memory mnemonic, the following is displayed:

```
603A F23C5000+   FMOVE.W   #$34,FP0
6040 4E71        NOP
6042 4E71        NOP
6044 F23C4000+   FMOVE.L   #$34,FP0
604A 56784E71    ADDQ.W    #3,$00004E71
604E 4E71        NOP
```

Note that the disassembler is getting out of sync because the immediate data present in the FMOVE instructions is not being correctly disassembled.

Signed off 02/17/88 in release A02.00

KPR #: 5000211599 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Error may result when executing "run from a until b" for target address.

Problem:
An intermittent failure occurs when executing the instruction "run from a until b" if address a and/or b reside in target system memory. The error message "too many states pushed onto stack" will appear on the status line.

Note that this failure was entered against version 1.0 of 68020 emulation software and has already been fixed in the current version of 68020 emulation software (version 1.10).

Signed off 02/17/88 in release A02.00

KPR #: 5000213983 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Enhancement to the dissassemble feature of the trace display.

Problem:
The customer would like the 68020 trace data to be disassembled automatically from opcode boundaries. He does not want to be required to specify "dissassemble from line number" or "from high word" or "from low word".

Signed off 02/17/88 in release A02.00

KPR #: 5000214452 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Failures when running the 68020 and any other emulator in same cardcage.

Problem:
The 68020 emulator will not function reliably if another emulator from the same cardcage is invoked. The 68020 may also affect the operation of the other emulator. Varying error messages appear on the status line of both emulators. The most common messages are "slow device" or "HALT".

The failure will not occur if one of the emulators in the card cage is not being used.

Temporary solution:
Assume that you have already entered into emulator A and are in the middle of an emulation session. Now you want to begin an emulation session with emulator B. Perform the following steps to ensure that no failures occur.
1. End lock from emulator A.
2. Enter emulator B and load the appropriate configuration file.
3. Now you can re-enter emulator A without affecting the operation of emulator A or B.

Signed off 02/17/88 in release A02.00

KPR #: 5000214841 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Failure occurs with "modify memory <address> to <procedure name>".

Problem:
The following modify memory operation does not work correctly:

modify memory long <address> to <procedure name>

Rather than loading the value of the starting address of the procedure, this operation will load the return address of the procedure.

For example,

modify memory long 6000H to _towers

where,
"_towers" is the name of a procedure in a "C" program.
_towers = 22BAH
return address of _towers = 2318H

The previous command will incorrectly load the value 00002318H into address 6000H. The value that should have been loaded is 000022BAH.

Temporary solution:
To work around this problem the procedure can be specified as "procedure + 0":

```
        modify memory long 6000h to _hanoi+0
```

or the absolute numeric data can be used:

```
        modify memory long 6000h to 22BAH
```

Signed off 02/17/88 in release A02.00

KPR #: 5000215558 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Improve comments in the emulation monitor regarding the "TRACE" vector.

Problem:
This report contains two enhancement requests:

1. The comments in the mon_68020.s file regarding the TRACE exception vectors are confusing. The comment reads:
   " THE TRACE-EXCEPTION ENTRY FOR THE MONITOR (ORG $24) SHOULD NOT
     BE ALTERED OR COMMENTED OUT IF THE USER WISHES TO USE THE
     SINGLE-STEP FACILITIES OF THE EMULATOR"
   There are two ORG $24 statements in our monitor, one reads

KPR #: 5000215558 **CONTINUED**

   "MONITOR SINGLE-STEP ENTRY" the other "---TRACE---". It would be better to have separate comments above each of these entries.

2. The second request was regarding the explanation of the TRACE vector in the manual. The manuals have already been updated with the correction.

Signed off 02/17/88 in release A02.00

KPR #: 5000234849 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Sporadic HP64120 I/O failures.

Problem:
Customer has a 64000_UX system with a 64416 68020 emulator. Sometimes when taking a trace with the emulator a error will occur and the following message will appear: "I/O Error HP64120 - Check power and HP-IB connections"

This occurs sometimes on the first page of the trace display and occurs more often when the next page key is pressed.

Signed off 02/17/88 in release A02.00

KPR #: 5000236844 Product: 68020 EMUL 300 64416S004 01.10

One-line description:
Failure occurs when executing a software breakpoint in user state

Problem:
The 68020 emulator does not properly handle breakpoints that are executed in user program space:

Example:
1. Write a program that clears the "SUPERVISOR" bit and then executes a small loop of any instructions.
2. Run in the monitor, and issue the command:
   "modify software_breakpoints set XXXXH" where X is the address of one of the instructions being executed in user state.
3. Then issue the command:
   "run from XXXXH" where XXXXH is the address where the user state is entered.

The emulator will execute the breakpoint, but return the message:
"could not disable breakpoint at address XXXX".

Temporary solution:
Explicitly clear the breakpoint after the error message appears:
  "modify software_breakpoints clear XXXXXH"
Or, use "run until" whenever possible, which still reports an error, but does disable the breakpoint.

Signed off 02/17/88 in release A02.00

KPR #: 5000239905 Product: 68020 EMUL 300 64416S004 01.10

One-line description:
OR instruction is not disassembled properly

Problem:
The 68020 disassembler does not properly disassemble the OR instruction. For example, the assembly listing file displays the following:

```
00002000    207C 0000 3000          MOVE.L  #$3000,A0
00002006    223C 0000 FFFF          MOVE.L  #$FFFF,D1
0000200C    8350                    OR      D1,(A0)
0000200E    66F0                    BNE     START
```

But, when displaying memory mneumonic, the following is displayed:

```
2000 207C0000+   MOVEA.L     #$00003000,A0
2006 223C0000+   MOVE.L      #$0000FFFF,D1
200C 835066F0    PACK        D0,D1,#$66F0
2010 00000000    ORI.B       #$00,D0
```

The mneumonic display in the trace list is also incorrect. Note that the OR instruction is a one word instruction. But, the disassembler uses two words to incorrectly generate a PACK instruction.

Signed off 02/17/88 in release A02.00

KPR #: D200077024 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Memory Mapper deletes the map_overlay definition when address bit 31=1.

Problem:
The memory mapper eliminates an overlay definition if the address range specified in the map and map_overlay definition has the highest bit (bit 31) set to a one. The failure does not appear until the configuration is modified. The error message "ERROR: No overlay defined for the given address" appears on the status line at the time that the "Modify memory configuration?" question is asked. If you answer "yes" to this configuration question and display the memory mapping screen, the map_overlay command has been eliminated from the screen. The .EA file also no longer contains the map_overlay command.

Following is the memory map portion of the .EA file before and after the "Modify memory configuration?" is answered "yes".

Before:
```
BEGIN MEMORY MAP
modify default guarded
modify valid_codes all
map fcode 6 080000000H thru 080001fffH emulation ram width32
map_overlay fcode 5 080000000H thru 080001fffH ram over fcode 6
  080000000H
END MEMORY MAP
```

KPR #: D200077024 **CONTINUED**

After (note that the map_overlay definition is missing):
```
BEGIN MEMORY MAP
modify default guarded
modify valid_codes all
map fcode 6 080000000H thru 080001fffH emulation ram width32
END MEMORY MAP
```

The map_overlay command was also deleted when I substituted 080000000H for 0f0000000H and 080001fffH for 0f0001fffH in the above example. The map_overlay command was not deleted when I used 070000000H and 040000000H in the above example. This seems to indicate that the failure only appears when bit 31 of the address is set to a one.

Temporary solution:
Do not perform map overlays in an address range where address bit 31 is set to a one.

Signed off 02/17/88 in release A02.00

KPR #: D200078071 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Unable to access transfer address of .X file if .L file does not exist.

Problem:
In HP OLS file formats, the absolute file (.X) defines the transfer address (if it exists) while the link_sym file (.L) defines symbolic information.

When using 68020 emulation, the following situation has been encountered:

1. First a program (called "mon_68020") consisting solely of the emulation monitor module is loaded. Both absolute (.X) and link_sym (.L) files exist for this program. The monitor program DOES NOT define a transfer address.

2. Next a second program (called "myprog") is loaded. This program is the user's code. The absolute (.X) file exists for this file. But the link_sym (.L) file does not exist. The absolute file DOES define a transfer address.

3. The command "run from transfer_address" is issued. The emulator generates the following error: "ERROR: transfer address not defined".

The reason the user does not I have a link_sym file is that the program was not made with the HP linker. Rather, it was made with 64888S, the HP file format converter product. This product translates absolute files with different formats (Tektronix, Motorola, Intel, etc.) into HP absolute files (including a transfer address if possible.) This product never creates a link_sym file.

KPR #: D200078071 **CONTINUED**

Signed off 02/17/88 in release A02.00

KPR #: D200078105 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Use of disassemble instructions_only may show no data with new trace.

Problem:
When a trace is disassembled with the instructions_only option and then a new trace is executed, a new disassemble command is required to re-tag instructions. Nothing is displayed in the trace list before the new disassemble command is executed.

The instructions_only option should be cleared when a new trace is executed, thus requiring the user to re-specify the new disassemble command.

Temporary solution:
If no data appears in the trace list after the trace measurement is complete, then issue a "display trace disassemble_from_line 0" command. This command will cause the trace data to be displayed.

Signed off 02/17/88 in release A02.00

KPR #: D200078113 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Modify memory map attributes does not release mapped memory.

Problem:
If more emulation memory is mapped than is available the following error message will appear on the status line when you attempt to end from the memory mapping session: "existing memory map exceeds available memory, free entries". If the user attempts to fix the problem by modifying the attributes of some of the entries to target or guarded memory, the error message described above will still appear when the user tries to end from the memory mapping session.

Temporary solution:
Delete the particular memory map entry and re-enter rather than modifying its attributes.

Signed off 02/17/88 in release A02.00

KPR #: D200078907 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
A DISPLAY MEMORY MNEUMONIC DISASSEMBLY ERROR.

Problem:
A DISSASEMBLY PROBLEM OCCURS WITH THE FOLLOWING INSTRUCTION:

```
   INSTRUCTION: MOVE.L #$D000,$C000
   DISASSEMBLY: MOVE.L #$0000D000,$00000000
```

THE DISASSEMBLY IS INCORRECT ONLY FOR THE MNEUMONIC MEMORY DISPLAY.

KPR #: D200078907 **CONTINUED**

THE DISASSEMBLED VERSION OF THE TRACE DISPLAY IS CORRECT.

THE FOLLOWING LINE FROM THE DISPLAY MEMORY MNEUMONIC DISPLAY IS INCORRECT (THE $C000 OPERAND IS INCORRECTLY DISPLAYED AS $00000000):

```
3000  23FC0000+  MOVE.L    #$0000D000,$00000000
```

THE FOLLOWING LINE FROM THE TRACE DISPLAY IS CORRECT:

```
trigger 00003000 23FC0000 MOVE.L    #$0000D000,$0000C000     0.40us
```

Signed off 02/17/88 in release A02.00

KPR #: D200078915 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Overlapped memory entries may not be resolved properly.

Problem:
During memory map configuration, if a new map entry overlaps an existing map entry, the following question is asked:

"A map entry has been overlapped, remove overlapped section?"

If the user answers "yes" to the above question, the two entries should be automatically combined by the emulation software.

A defect exists that prevents proper resolution if one of the map entries includes the upper end of the address space, i.e. 0ffffff00h to 0ffffffffh. If this is the case, the entries may not be combined as expected or additional erroneous entries may automatically be added.

Temporary solution:
The user should study the map to determine whether any entries are incorrect. Any erroneous entries should be deleted and re-specified.

Signed off 02/17/88 in release A02.00

KPR #: D200078931 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Modify memory using symbolic data incorrectly loads "value -1".

Problem:
When a modify memory command is issued and the new data to be used is specified by a symbol instead of numeric data, the actual value loaded will be the value of the symbol minus 1. For example:

```
        if MONITOR_ENTRY is 1000h, then

        modify memory long 24h to MONITOR_ENTRY

        actually loads address 24 with 0fffh.
```

KPR #: D200078931 **CONTINUED**

Temporary solution:
To work around this problem the symbol can be specified as "symbol + 0":

```
        modify memory long 24h to MONITOR_ENTRY+0
```

or the absolute numeric data can be used:

```
        modify memory long 24h to 1000h
```

Signed off 02/17/88 in release A02.00

KPR #: D200078949 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Can not map_overlay memory if range includes 0ffffff00h-0ffffffffh.

Problem:
When configuring memory with function codes on, one emulation memory map entry can not be overlayed onto another entry if either range includes the upper end of the address space, 0ffffff00h thru 0ffffffffh. If this is attempted, the error message "No overlay defined for given address" results.

Signed off 02/17/88 in release A02.00

KPR #: D200079061 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
68020 PV incorrectly defaults co-cards for 2nd emulator in a cage.

Problem:
If the user has 2 68020 emulators in a cardcage, and selects the second control card for PV, the software selects the analysis card from the first emulator in the cage as the default analysis co-card for the second emulator. The user must enter the slot number of the correct analysis card before continuing.

Temporary solution:
Do not select the default analysis card by pressing the return softkey. It is necessary to enter the actual slot number of the analysis card before pressing the return softkey.

Signed off 02/17/88 in release A02.00

KPR #: D200079707 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Load/copy/store using files over net may fail with 68020 emulator.

Problem:

KPR #: D200079707 **CONTINUED**

If you netunam to another HP-UX system, and attempt to load/store/copy using files on this remote system, the operations may fail without producing any warning messages.

Signed off 02/17/88 in release A02.00

KPR #: D200080408 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Some interaction problems with "at_execution run/trace".

Problem:
Several problems exist with interaction between "at_execution run" and "at_execution trace".

If the module is using intermodule bus functions, the execute softkey should always appear. If an "at_execution run ..." followed by a "run ..." is issued, the pending "at_execution" is killed and the execute softkey is removed even though the IMB is being used.

Killing a pending "at_execution run ..." also kills a pending trace. This is incorrect operation.

Loading a trace specification may remove the execute key needed for a pending run.

Signed off 02/17/88 in release A02.00

KPR #: D200080416 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Analyzer GLOBAL_CONTEXT specification core dumps when running HP-UX 5.5.

Problem:
When running HP 64416 Emulation Software with HP-UX operating system version 5.5, any specification of GLOBAL_CONTEXT using the "set" or "modify" commands will cause a core dump and an automatic "end released" out of the 68020 emulation session.

Temporary solution:
The only work-around for this defect is to continue using HP-UX operating software version 5.3 or 5.22 until version 2.0 of the 68020 emulation software is available.

Signed off 02/17/88 in release A02.00

KPR #: D200080424 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Trace data not updated if trace command follows at_execution run.

Problem:
When the following commands are entered:

```
at_execution run ...
```

KPR #: D200080424 **CONTINUED**

trace

The tracelist display is not properly updated and may contain old data.

Signed off 02/17/88 in release A02.00

KPR #: D200080440 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Some command file sequences may not work.

Problem:
When using command files, the follow command sequences do not work properly:

```
  trace ...
  end ...
```

causes the trace to be halted,

```
  end select <modulename>  or  select <modulename>
  halt
```

does not halt the analyzer (halt is ignored).

Temporary solution:
Inserting a delay in these sequences using the "wait" command will solve the problem:

```
   trace
   wait 5
   end ...
```

Signed off 02/17/88 in release A02.00

KPR #: D200080457 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Turning off BNC usage does not update the trace display

Problem:
After specifying a measurement using bnc ports, a subsequent "set bnc_ports off" command does not remove the previous bnc spec from the current trace specification display.

Signed off 02/17/88 in release A02.00

KPR #: D200081059 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
PV for coordinated emulation start may miss failure.

Problem:
The "coordinated emulation start" test within the 68020 emulator

KPR #: D200081059 **CONTINUED**

performance verification software may fail to report a problem with the hardware. The test will always pass if the IMB cable is not attached. This test should always fail when the IMB cable is not hooked-up.

Signed off 02/17/88 in release A02.00

KPR #: D200081125 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
MON_XFR_BUF area in monitor not properly defined.

Problem:
With version 1.10 of the 68020 emulation monitor, it is possible to define an emulation memory area for the monitor that is actually too small to contain the monitor. It is common to allocate 4096 bytes (1000h-1fffh for example) for the monitor, when it actually requires 4164 bytes (version 1.10). No errors result when loading the monitor and usually no strange behavior is evident.

This results from the use of "DS.W" directives to define the monitor transfer buffer. "DS.W 325" informs the assembler to define 325 words of storage. This generates an uninitialized (the values are not specified, nor are they predictable) area of memory beginning at the address of the "DS.W" directive. When the assembler encounters this statement, it simply increments its internal program counter by 325 words (in this case), and continues with the next instruction. This causes the relocatable (.o) file and finally the executable file (.X) to contain information indicating that there is a 325 word GAP at a particular address. But, the presence of a gap, as opposed to 325 words of data, means that nothing will be loaded into memory for the address range covered by the gap. In fact, during a "load memory ..." command, such an area is reserved in memory, but is left unmodified by the load procedure.

During a "load memory ..." command, since no memory accesses will occur for an area defined by "DS.W" there will be no error message even if the area crosses into a guarded memory range. The last 327 words (transfer buffer) of the version 1.10 monitor is defined with "DS.W", which means that this area (or any part thereof) could be mapped as guarded, but would not generate error messages during a "load memory ..." command. This situation would probably go unnoticed until a monitor command referenced this portion of the transfer buffer, at which time a guarded access in the monitor would occur.

The "DS.W" directives should be replaced with "DC.B" directives in the transfer buffer area. The "DC.B" directives will correct this problem by initializing the transfer buffer to specific values. "DCB.W 325,0" tells the assembler to generate a "Constant Buffer", 325 words long, will all words initialized to 0. Here, the relocatable and executable files contain 325 words of actual data, rather than a gap. During a "load memory ..." command, the area covered by a DCB.W is modified to the values specified by the programmer (0 in this case).

Since memory accesses do occur during a "load memory ..." command, an

KPR #: D200081125 **CONTINUED**

error message will be properly displayed if the monitor overflows into guarded space. The remedy is simple - extend the memory map entry for the area occupied by the emulation monitor. Remember that this area can be extended in as-little-as 256 byte increments.

Temporary solution:
The monitor should be changed to use DCB.W for the transfer buffer definition:

```
MON_XFR_BUF
               DCB.W         325,0
MON_XFR_END
               DCB.W         2,0
```

Signed off 02/17/88 in release A02.00

KPR #: D200081968 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Tracelist symbols disappear from the trace display in certain conditions

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                           ; end locks the emulation session
3. <system name> <module name>   ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

Signed off 02/17/88 in release A02.00

KPR #: D200082354 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

KPR #: D200082354 **CONTINUED**

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
      cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 02/17/88 in release A02.00

KPR #: D200086421 Product: 68020 EMUL 300 64416S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

Signed off 02/17/88 in release A02.00

KPR #: D200088427 Product: 68020 EMUL 300 64416S004 02.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200085456 Product: 68020C AXLS COMP 300 64903S004 01.00

One-line description:
Compiler won't take address of a function pointer.

Problem:
The compiler issues a warning that it is ignoring the & operator when it is applied to a function pointer. The following code would result in an error message about the assignment since the address of 'fp' has not been taken.

```
int (*fp)();
int (**fpp)();

main()
{
        fpp = &fp;
}
```

Signed off 04/29/88 in release A01.30

KPR #: D200087502 Product: 68020C AXLS COMP 300 64903S004 01.10

One-line description:
#include files are not searched in order of -I options

Problem:
Text:
  #include files not searched in order of -I options

In the present AxLS cpp, directories searched for #include files are in opposite order than the -I options are specified on the command line.

A more natural searching order (and the one Ft. Collins' uses) is to search directories in the same order as the -I options are specified. For example:

If the file <stdio.h> existed in both ./Adir and ./Bdir, and the command

```
   cpp68020 -I./Adir -I./Bdir <file>
```

is given, you would expect that the stdio.h in ./Adir would be included; in fact the file in ./Bdir is included.

Temporary solution:
Workaround:
1) don't overload #include file names
2) specify include directories in reverse order you want them searched.

Signed off 04/29/88 in release A01.30

KPR #: D200089524 Product: 68020C AXLS COMP 300 64903S004 01.10

One-line description:
Runtime and support libraries contain loadtime initializers.

Problem:
Compiler libraries lib.a, libpi.a and env.a contain loadtime initializers. The lack of runtime initialization in an embedded system may cause library routines to behave unexpextedly.

Signed off 04/29/88 in release A01.30

KPR #: D200080150 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080762 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081430 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

KPR #: D200081430 **CONTINUED**

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

KPR #: D200081976 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200082008 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082503 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085241 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Load <file> noupdate without database should give error message.

Problem:
Load <file> noupdate without database should give error message

The following load command gives a somewhat confusing result if no database (file.Y) exists:

```
     load <file> noupdate
```

The noupdate command says to not rebuild the edb database if it is out of date. If no database exists the result is that the absolute file gets loaded, without any warning or error message, and any old symbols get destroyed. The correct response should be to at least issue the following warning message - "No database: file".

This enhancement request should be expanded to all of the PII emulators.

KPR #: D200085795 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous

KPR #: D200085795 **CONTINUED**

error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200086934 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Core dump can happen when displaying guarded memory mnemonic

KPR #: D200088104 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090639 Product: 6805 U&R EMUL 300 64192S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 1650057893 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Can't load a program into target memory or emul. mem. with slow clock.

Problem:
Same as submitter text.

Temporary solution:
There is no workaround available.

KPR #: D200080481 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080796 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081455 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

KPR #: D200082032 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083071 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:

KPR #: D200083071 **CONTINUED**

Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085852 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                          ; end locks the emulation session
3. <system name> <module name>  ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086249 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088203 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090662 Product: 6805 E EMUL 300 64195S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200080473 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080788 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200082024 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can

KPR #: D200082024 **CONTINUED**

release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083063 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085845 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086231 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

KPR #: D200086231 **CONTINUED**

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088195 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090654 Product: 6805 G EMUL 300 64194S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200080465 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080770 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081448 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

KPR #: D200081448 **CONTINUED**

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

KPR #: D200082016 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
       cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083055 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085837 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                          ; end locks the emulation session
```

KPR #: D200085837 **CONTINUED**

```
3. <system name> <module name>    ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086223 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088187 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090647 Product: 6805 P EMUL 300 64193S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

KPR #: D200090647 **CONTINUED**

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 1650020396 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
LR error flagged for legal expression of the form 'label-value'.

Problem:
The 6809 assembler flags a legal range error if you use an external label in the following manner.

```
"6809"

      EXT     LSD
      JMP     LSD-2
```

The JMP LSD-2 causes a legal range error to be generated.

Temporary solution:
Jump to an equivalent positive offset. In this example you would use

```
"6809"

         EXT   LSD
         JMP   LSD+0FFFEH
```

Duplicate Service Requests: 1650044578

KPR #: 5000143628 Product: 6805/9 ASSEMB 64844 01.10

One-line description:
Label in IF stmnt. does not appear in XREF

Problem:
The following shows a condition where a label is not listed in the cross reference table that should be there.

```
"processor name"
TOTO                EQU            0
                    IF             TOTO
LABEL0              LDA            1000H
                    ELSE
LABEL1              LDA            2000H
                    ENDIF
```

When TOTO equals 1, everything is correct. The cross reference list both TOTO and LABEL0. When TOTO equals 0, the cross reference only list TOTO, not LABEL1.

KPR #: 5000150292 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
HEX pseudo causes byte counter to quit incrementing in certain cases.

Problem:
The byte counter is not incremeted after the 1EH in the following program.

"6809"

```
LABEL           HEX             1B,EC,1E,20,30
```

Temporary solution:
Use the FCB pseudo instead.

"6809"

```
LABEL       FCB     1BH,0ECH,1EH,20H,30H
```

KPR #: 5000164012 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
Arithmetic expression is not being evaluated correctly.

Problem:
When you offset a relocatable label by -1 the assembler flags an out of range error.

"6809"

```
        EXT     TABLE,ENDTABLE

        LDA     TABLE
        LDA     ENDTABLE-1                      ;LR error flagged
        LDA     TABLE+0FFFFH                    ;No error flagged
```

KPR #: D200037267 Product: 6805/9 ASSEMB 64844 00.15

One-line description:
No error generated when overflow occurs.

Problem:
No overflow error message is generated. See following text.
"6809"

```
    LDD     10000H        ;SHOULD GENERATE ERROR
```

Signed off 08/25/86 in release 01.15

KPR #: D200063164 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
NT operator not operating consistiently.

Problem:

KPR #: D200063164 **CONTINUED**

In the following program the .NT. operator will work in the first case, but, not in the second.

"6809"

```
START   EQU     80H
        LDA     #.NT.00001111B
        LDA     #.NT.11110000B          ;LR ERROR FLAGGED
        LDA     #.NT.START              ;LR error flagged.
```

Temporary solution:
AND THE VALUE WITH 0FFH.

"6809"

```
       LDA        #.NT.11110000B.AN.0FFH
```

KPR #: D200076950 Product: 6805/9 ASSEMB 64844 01.11

One-line description:
BEXT address is not calculated correclty.

Problem:
In the following program the base page external DISP_MASK is given two different values in the two STA commands.

"6809"

```
        BEXT    DISP_MASK
        EXT     CLOCK,PROC

N_CALC  EQU     CLOCK+2
S_CALC  EQU     CLOCK_+1

        PROG
        SETDP   0

CALC:   CLR     S_CALC
        CLS     N_CALC
        STA     DISP_MASK
        JSR     PROC
        STA     DISP_MASK
        RTS
```

"6809"

```
        GLB     CLOCK,DISP_MASK,PROC

        DATA
        BASE_SEG
CLOCK   RMB     10
DISP_MASK       RMB     1
        BASE_END
```

KPR #: D200076950 **CONTINUED**

```
        PROG
PROC:  NOP
       RTS
```

NOTE: In the first module all lines are necessary. If you remove
the JSR PROC, for example, the problem goes away.

If you look at the .X file created you will note that the
STA DISP_MASK's instructions have different destinations.

Temporary solution:
In module one (ie the one that declares DISP_MASK as BEXT and
has the STA commands) declare DISP_MASK as a regular external
and turn on the DIRECT pseudo above the STA instructions.

```
"6809"

        EXT     DISP_MASK,CLOCK,PROC

N_CALC          EQU     CLOCK+1
S_CALC          EQU     CLOCK+2

                PROG
                CLR     S_CALC
                CLR     N_CALC

                DIRECT
                STA     DISP_MASK
                EXTEND
                JSR     PRO
                DIRECT
                STA     DISP_MASK
                RTS
```

KPR #: 1650056838 Product: 6809 C 64822 01.80

Keywords: PROBLEM ON 9000/S300

One-line description:
There is a problem with incrementing pointer structures using '++'.

Problem:
The code generated by the C compiler 6809 is incorrect for the following
instructions:

```
"C"
"6809"
struct { char m1;
         char m2;
         char m3  } *struc;
func1()
{
(*struc).m1++;
(*struc).m2++;
(*struc).m3++;
}
```

For the last 2 incrementations the compiler generate:

```
      STX [000000002H,S]
```

which has no sense.

Temporary solution:
Use the expanded notation for incrementing the structure variables.
For example, change the code to read :

```
"C"
"6809"
struct { char m1;
         char m2;
         char m3;  } *struct;
func1()
{
   (*struc).m1 = (*struc).m1 + 1;
   (*struc).m2 = (*struc).m2 + 1;
   (*struc).m3 = (*struc).m3 + 1;
}
```

Signed off 08/31/88 in release A01.90

KPR #: 5000243907 Product: 6809 C 64822 01.40

Keywords: PROBLEM ON 9000/S500

One-line description:
Compiler generating bad code which may cause run-time crash.

Problem:
The following instructions generate bad code:

```
 extern char tab[50];
 f()
 {
 short i; char j;
```

KPR #: 5000243907 **CONTINUED**

```
 tab[i] &= ~j;
 }
```

An extra instruction STX [4,S] is generated and then a run-time error occurs.

Signed off 08/31/88 in release A01.90

KPR #: D200007237 Product: 6809 C 64822 00.00

Keywords: PASS 1

One-line description:
COMP_SYM file not purged when COMP_SYM option not selected.

Problem:
If COMP_SYM option is not selected when compiling, previously created COMP_SYM file is not purged. Since this file can only cause trouble, it should be purged if the COMP_SYM option is not specified. This only happens if you have 64330.

Temporary solution:
Purge the COMP_SYM file before compiling.

Signed off 06/19/85 in release 00.01

KPR #: D200055558 Product: 6809 C 64822 01.06

Keywords: PASS 3

One-line description:
Illegal opcode generated when assigning value to a char. array pointer.

Problem:
The 6809 C compiler is generating an illegal opcode. The program below demonstrates this problem.

```
"C"
"6809"
$LIST_OBJ$
enum  COMMAND {Nocmd, Special} command;

main() {

char *disp_cmnd[];
      switch(command){
                 case  Nocmd    :   disp_cmnd= "No cmd " ;  break;
                 case  Special  :   disp_cmnd= "Special ";  break;
      }

}
```

The expanded listing shows that the compiler is generating a 'CD' for STD   0xxH,S instructions when it should generate a 'ED'.

KPR #: D200055558 **CONTINUED**

Temporary solution:
Define the pointer as a pointer to a character rather than a pointer to an array of chars.

```
"C"
"6809"

$LIST_OBJ$

enum   COMMAND {Nocmd, Special} command;

main() {

char   *disp_cmnd;

        switch(command) {
                case   Nocmd    :   disp_cmnd = "No cmd"  ; break;
                case   Special  :   disp_cmnd = "Special "; break;

        }

}
```

Signed off 09/15/86 in release 01.07

KPR #: D200068239 Product: 6809 C 64822 01.07

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"

struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200068239 **CONTINUED**

KPR #: D200069864 Product: 6809 C 64822 01.07

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200073171 Product: 6809 C 64822 01.07

One-line description:
Use of address (&) stack vars on right side of conditional expression

Problem:
C 6809 defect with the address(&) function:

Comparisons using the address(&) function with local variables or parameters in $RECURSIVE ON$ may generate incorrect code.

The use of the &(variable) function on the right hand side of

KPR #: D200073171 **CONTINUED**

comparison expressions can cause incorrect code to be generated.

eg. The statement:

```
IF ( pointer <> &local_var ) ...
```

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of functions compiled with $RECURSIVE ON$(the default value for C functions).

No problem occurs for static variables, any external variables, or local variables and parameters of functions compiled with $FIXED_PARAMETERS ON$ and $RECURSIVE OFF$.

The simple work around solution is to only use the &(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

The following listing illustrates the problem.

```
"C"
"6809"
extern int a,b,c;
Recursive_function(p,q,r)
 int p,q,r;
{  int i,j,k; $LIST_CODE ON$
              /* Static variables */
 if (&a !=  b) ; /* Works */
          LDX  #a
          CMPX b
          LBEQ Recursi01_1
      Recursi01_1
 if ( a != &b) ; 7* Works */
          LDX  a
          CMPX #b
          LBEQ Recursi01_2
      Recursi01_2
 if (&a != &b) ; 7* Works */
          LDX  #a
          CMPX #b
          LBEQ Recursi01_3
      Recursi01_3
              /* Local parameters */
 if (&p !=  q) ; /* Works with & on left side */
          LEAX 0000000CH,S

          CMPX 0000000EH,S
          LBEQ Recursi01_4
      Recursi01_4
 if ( p != &q) ; 7* Fails with & on right side */
          LDX  0000000CH,S
```

KPR #: D200073171 **CONTINUED**

```
              CMPX 00000000EH,S *** Value of q, NOT address of q ***
              LBEQ Recursi01_5
          Recursi01_5
   if (&p != &q) ; /* Fails with & on both sides */
              LEAX 00000000CH,S
              CMPX 00000000EH,S *** Value of q, NOT address of q ***
              LBEQ Recursi01_6
          Recursi01_6
                 /* Local variables */
   if (&i != &j) ; /* Fails with & on both sides */
              LEAX 000000002H,S
              CMPX 000000004H,S *** Value of j, NOT address of j ***
              LBEQ Recursi01_9
          Recursi01_9
  }
```

Workaround:

The simple work around solution is to only use the &(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

Temporary solution:
See problem text.

Duplicate Service Requests: D200073163

KPR #: D200075036   Product: 6809 C   64822   01.08

One-line description:
Some C programs using pointer & structure dereferences cause error #1006

Problem:
Some C programs using structure with pointers in expressions may cause pass 2 error 1006 - Compiler Error.

Some expressions with multiple use of pointer and structure dereferences may cause this error.

These errors did not appear on previous versions of the compiler.

The workaround solution is to break up the expression into smaller statements.

The following program illustrates the problem:

```
  "C"
```

KPR #: D200075036 **CONTINUED**

```
  "6809"

  typedef unsigned int (*FP)();
  typedef struct hs {
      FP         load;
      FP         reset;
  } HEADER;

  int _app_set_valid(h)
  HEADER *h;
  {  FP p;  /* Temporary variable */
    h->load = (char*)h+(int)h->load;  /* This works */
    h->reset = (char*)h+(int)h->reset; /*  This causes 1006 error */
**** Pass 2 ERROR ?? 1006
  }
```

The following program shows some workaround solutions to the problem:

```
"C"
"6809"

typedef unsigned int (*FP)();
typedef struct hs {
    FP         load;
    FP         reset;
} HEADER;

int _app_set_valid(h)
HEADER *h;
{  FP p;  /* Temporary variable */
  h->load = (char*)h+(int)h->load;  /* This works */
  /*h->reset = (char*)h+(int)h->reset;     This causes 1006 error */
    h->reset += (int)(char*)h;                 /* This works: using += */
    p =  h->reset; /* This works: temporary assignment */
    h->reset =  (int)p+(int)(char*)h; /* This works: temprorary assignme
nt */
}
```

Temporary solution:
See problem text.

KPR #: D200075663   Product: 6809 C   64822   01.08

One-line description:
Programs with duplicate goto labels may fail in Pass 3 on VAX&HPUX C

Problem:
C programs with duplicate user labels(for goto's)may fail in pass3.

The current SUDS C compilers may produce the error
   "comp: failed; too many errors in pass 3."
   from some C programs which previously compiled correctly.

KPR #: D200075663 **CONTINUED**

This problem did not appear in any C compilers before April 1987.

In C it is valid to use the same goto label symbol in different functions, since they have a logical different scope.

However, the HP64000 C cross will inform the user that these symbols are duplicate in the pass3 on the compiler. These symbols would produce duplicate label definitions when defined the ASM_FILE output is assembled. In addition the emulation products will only find one of these symbols.

The duplicate symbol detection algorithm on the HPUX/300, HPUX/500 and VAX/VMS C language compilers has an error which causes the compiler to fail.

However, the duplicate symbol checking is done after all of the relocatable and asmb_sym files have been produced. These output files are equivalent to those produced in the HP64000 version compilers. Thus, the output of the compilers is still correct, except for some trailing lines in the listing file.

The following program will cause this defect to occur:

```
"C"
"6800"
/***********************************************************/
/*   TEST  file for problem with duplicate local labels     */
/*---------------------------------------------------------*/
/*   This program fails in pass 3 on VAX & HPUX/500 &/300   */
/*      While checking for duplicate asmb_sym symbols       */
/*      due to the "duplicate" error_exit labels            */

int i;
test1()
{
    /* ... */
  if (i == 77) goto error_exit;
    /* ... */
  error_exit:

      i = -1;
    /* ... */
}
/* duplicate symbol should be created */

test2()
{
    /* ... */
  if (i == 137) goto error_exit;
    /* ... */
  error_exit:

      i = -1;
```

KPR #: D200075663 **CONTINUED**

```
    /* ... */
}
```

Temporary solution:
Do not use a local symbol more than once per module.

KPR #: D200079632 Product: 6809 C 64822 01.08

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
        }
   }
 }
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to

KPR #: D200079632 **CONTINUED**

be generated.

KPR #: D200081497 Product: 6809 C 64822 01.20

Keywords: PROBLEM ON 9000/S300

One-line description:
Incorrect code is generaaated for while statment.

Problem:
The following program causes the 6809 C compiler to generate bad code.
A write is made to an invalid location.

```
"C"
"6809"

char  tos[20];

strcpy(to,from)
char *to,*from;

{

   while (*to++ = *from++);
}

main()
{
strcpy(tos,"0");
}
```

Within the strcpy procedure an instruction

```
STX  [000002H,S]
```

is generated. This instruction is incorrect, the store should
be to location 08H,S.

Temporary solution:
Do the increment of the pointers in the while body.

```
strcpy(to,from)
char *to,*from;

{
   while (*to = *from) {
        to++;
        from++;
}
```

Signed off 08/31/88 in release A01.90

KPR #: D200081547 Product: 6809 C 64822 01.08

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself
as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
           ;
}
```

Customer feels that this variable should be evaluated to see if it
is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;

  OR
```

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200086603 Product: 6809 C 64822 01.80

One-line description:
Use of "+=" accessing first element of structure using pointer error

Problem:
Text:
  Use of "+=" accessing first element of structure using pointer error
.submitter

Problem: Compiler generates bad temporary store instruction for

KPR #: D200086603 **CONTINUED**

"+=" instruction accessing first element of a structure using pointer.

Use of other "-=", "++" and "--" in similar pointer expressions may also cause this error.

```
"C"
"6809"
typedef struct {
    short s;
    int i;
    long l;
} OBJ_STRUCT1;

typedef struct {
    OBJ_STRUCT1 *ptr1;
    OBJ_STRUCT1 *ptr2;
    OBJ_STRUCT1 *ptr3;
} PTR_STRUCT1;

typedef PTR_STRUCT1 *PTR_PTR1;

PTR_PTR1 pp1;
short *sptr;
main(){
    pp1 -> ptr2 -> s += 1;

  /*WORKAROUND*/
   sptr = &(pp1-> ptr2 -> s);
   *sptr += 1;
}
```

EXPANDED EXAMPLE:

```
 main(){
     pp1 -> ptr2 -> s += 1;
             LDX  Dstatic
             LEAX 000000002H,X
             LDX  ,X
             STX  [000000002H,S] ; This instruction is wrong. It will
                                 ; cause an improper write to memory
                                 ; It should be STX 2,X  no indirects.
             LDB  ,X
             INCB
             STB  ,X

   /*WORKAROUND*/
    sptr = &(pp1-> ptr2 -> s);
             LDX  Dstatic
             LEAX 000000002H,X
             LDX  ,X
             STX  Dstatic+00002H
    *sptr += 1;
             INC  [Dstatic+00002H]
  }
```

KPR #: D200086603 **CONTINUED**

Temporary solution:

```
"C"
"6809"
typedef struct {
    short s;
    int i;
    long l;
} OBJ_STRUCT1;

typedef struct {
    OBJ_STRUCT1 *ptr1;
    OBJ_STRUCT1 *ptr2;
    OBJ_STRUCT1 *ptr3;
} PTR_STRUCT1;

typedef PTR_STRUCT1 *PTR_PTR1;

PTR_PTR1 pp1;
short *sptr;
main(){
    pp1 -> ptr2 -> s += 1;



  /*WORKAROUND*/
   sptr = &(pp1-> ptr2 -> s);
   *sptr += 1;
}
```

Signed off 08/31/88 in release A01.90

KPR #: D200086611 Product: 6809 C 64822 01.80

One-line description:
Compare error using address of local variable on right of expression

Problem:
Problem: Compiler generates illegal instruction when performing an address compare of a stack relative local variable on the right hand side of an expression.

The compiler needs to use a load effective address instruction to create the proper address. This can not be done in one instruction with a compare.

```
"C"
"6809"
test(){
 int t,*q;
  q = &t;
```

KPR #: D200086611 **CONTINUED**

```
  $AMNESIA$
  if (q != &t) ; /* This will not work */
/*WORKAROUND*/
  if (&t != q) ;
  }
```

EXPANDED EXAMPLE :

```
 test(){
  int t,*q;
   q = &t;
             LEAX 000000002H,S
             STX  000000004H,S
   $AMNESIA$
   if (q != &t) ;
             LDX  000000004H,S
             CMPX 000000002H,S   /* This is not correct */
             LBEQ test01_1       /* Comparing to contents NOT address */
         test01_1
 /*WORKAROUND*/
   if (&t != q) ;
             LEAX 000000002H,S
             CMPX 000000004H,S
             LBEQ test01_2
         test01_2
   }
```

Temporary solution:

```
"C"
"6809"
test(){
 int t,*q;
  q = &t;
  $AMNESIA$
  if (q != &t) ; /* This will not work */


/*WORKAROUND*/
  if (&t != q) ;
  }
```

KPR #: D200086629   Product: 6809 C   64822   01.80

One-line description:
SHORT_ARITH OFF expressions in branches may not work as K&R

Problem:
Problem: With the SHORT_ARITH option OFF, the 6809 compiler does not execute full K&R C code correctly for certain mixed arithmetic operations when used in "if" expressions.
Problems occur when 8-bit (short) arithmetic is used, rather than full expansion to 16 bit values to perform operations as in the standard K&R.

KPR #: D200086629 **CONTINUED**

EXAMPLE:

```
"C"
"6809"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */
 /* WORKAROUND */
 if ((int)s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
  else  ;
}
```

The 6809 C compiler computes mixed expressions correctly, as in assignment statements and parameter expressions.
This defect appears only when mixed expressions are used without assignment as conditional branching expressions.

This problem may be generated with other operators besides the "<<" as in the example, such as ">>" , "/" and "%

EXPANDED example:

```
  $SHORT_ARITH OFF$
  if (s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
             LDB  Dstatic
             LSLB
             LSLB
             LSLB
             LSLB
             LBEQ main01_1
             LBRA main01_2
         main01_1
   else  ;  /* Code branches here, due to use of byte arithmetic. */
         main01_2
/* WORKAROUND */
  if ((int)s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
             LDB  Dstatic
             SEX
             TFR  D,X
             LDB  #004H
             LBSR Zwshift
             CMPD #00000H
             LBEQ main01_3
             LBRA main01_4
         main01_3
   else  ;

         main01_4
```

Temporary solution:
"C"

KPR #: D200086629 **CONTINUED**

```
"6809"
short s,ss;
main(){
 s= 0x40;
 $SHORT_ARITH OFF$
 if (s<<4); /*Result should be 64*16=1024 => <>0 should branch here*/
  else  ;  /* Code branches here, due to use of byte arithmetic. */


 /* WORKAROUND */
 if ((int)s<<4); /*Result 64*16=1024 which is <>0 should branch here*/
  else  ;
}
```

KPR #: 5000152439 Product: 6809 C M 64822-90901 01.06

One-line description:
Clarification of interface for USER_DEFINED and real number routines.

Problem:

In the example below, 6809 libraries cannot be explicitly called. If they are called explicitly as routines, the stack is built differently than when a compiler generated call is made.

Example:

```
"C"
"6809"

$FIX_PARAMETERS ON$

main() {

int x;
double xx;
extern double LONGREAL_FLOAT();
int *px;
int *pxx;
x =5;
px = &x;
pxx = &xx;
LONGREAL_FLOAT(px,pxx);             /* Conversion is not made */
}
```

Temporary solution:
For explicit use of ALL the real number library routines, declare your routines as in the following example with $FIXED_PARAMETERS ON$ and $RECURSIVE OFF$ (Chapter 4 in manual). The compiler will then generate the proper form of parameter passing to satisfy the real number library. Note, $RECURSIVE OFF$ is also necessary when using the USER_DEFINED interface method (Chapter 2 in manual).

Example:

```
"C"
"6809"

extern int xint;
extern double xdouble;
extern int *pxint;
extern int *pxdouble;

extern recursive_variable_func();
$FIXED_PARAMETERS ON$
extern recursive_FIXED_PARM_func();
$RECURSIVE OFF$
extern LONGREAL_FLOAT();
/*NOTE do not declare these functions double. It will cause extra
```

KPR #: 5000152439 **CONTINUED**

```
parameters to be passed*/

main() {

$LIST_CODE ON$
  /* NOTE: Parameter passing method for standard C function*/
recursive_variable_func(&xint,&xdouble);
            LDU   #xdouble
            LDY   #xint
            LDD   #00004H
            PSHS  X,Y,U
            LBSR  recursive_varia
            LEAS  000000006H,S
  /* NOTE: Parameter passing method for FIXED_PARAMETER(Pascal) function
*/
recursive_FIXED_PARAM_func(&xint,&xdouble);
            LDU   #xdouble
            LDY   #xint
            LBSR  recursive_FIXED
xint = 5;
            LDD   #00005H
            STD   xint
pxint = &xint;
            LDD   #xint
            STD   pxint
pxdouble = &xdouble;
            LDD   #xdouble
            STD   pxdouble
/* NOTE: Parameter passing method for STANDARD REAL NUMBER LIBRARY funct
ion */
/* A (Pascal) function with $FIXED_PARAMETERS$ and $RECURSIVE OFF$ !*/
LONGREAL_FLOAT(pxint,pxdouble);
            TFR   D,X
            LDD   pxint
            LBSR  LONGREAL_FLOAT
LONGREAL_FLOAT(&xint,&xdouble);
            LDX   #xdouble
            LDD   #xint
            LBSR  LONGREAL_FLOAT

/* Compare the stack build on this assignment call which uses
LONGREAL_FLOAT versus the explicit call above */

xint = xdouble;
            LDX   #xint
            LDD   #xdouble
            LBSR  LONGREAL_TRUNC
xdouble = xint;
            LDX   #xdouble
            LDD   #xint
            LBSR  LONGREAL_FLOAT
}
        Rmain
            GLOBAL   Rmain
```

KPR #: 5000152439 **CONTINUED**

```
            RTS
        Dmain
            RMB      0000EH
            GLOBAL   main
        Emain        EQU $-1

            GLOBAL   Emain

            EXTERNAL LONGREAL_TRUNC
            EXTERNAL LONGREAL_FLOAT
```

Fix information:
Fix is documented in Software Notice 5959-2129 R2707.

Signed off 08/05/87 in release 01.08

KPR #: D200055814 Product: 6809 C M 64822-90901 01.06

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int   cntr;
  int   (*tmp)();
```

KPR #: D200055814 **CONTINUED**

```
        for (cntr=1; cntr<4; cntr++)
          { tmp = func5(cntr);
          }
}

pfi func5(tmp2)
int tmp2;
{
    if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.08

KPR #: D200015636 Product: 6809 C 500 64822S001 01.00

Keywords: PASS 1

One-line description:
Incorrect code is generated when complementing a parm. in a return stmt.

Problem:
In the following program the incorrect code is generated for the complement of the parameter to be returned.

```
"C"
"6809"
unsigned short bug()
{
    return(~x);
}
```

The compiler generates a "NEGB" when it should be a "COMB"

Temporary solution:
Set up a temporary variable and assign the complement of the parameter to it and then return the temporary. For example,

```
    unsigned short temp;
    temp = ~x;
    return temp;
```

Signed off 08/25/86 in release 01.50

KPR #: D200029702 Product: 6809 C 500 64822S001 01.00

One-line description:
File fails to compile. Error 1113 is generated.

Problem:
The submitted file does not compile. In pass three error 1113 "Program counters disagree" is flagged. The file will not compile on any system.

Signed off 08/25/86 in release 01.50

KPR #: D200035873 Product: 6809 C 500 64822S001 00.00

Keywords: CODE GENERATOR

One-line description:
16 bit comparison on a 8 bit unsigned short field.

Problem:
IMPROPER CODE GENERATED FOR STATEMENT INVOLVING unsigned short VARIABLE UNLESS EXPLICITLY RE-CAST AS unsigned short.

```
main()
{
static unsigned short digit_index;
static unsigned short digit[12];
int a,b;
```

KPR #: D200035873 **CONTINUED**

```
if (digit[digit_index]--){
a=4;
b=4;}
else{
a=5;
b=5;}
}
```

IMPROPER CODE IS GENERATED FOR THE COMPARISON (ie THE COMPARISON IS DONE ON 16 BITS (8 OF WHICH HAVE BEEN CLEARED) AGAINST #0FFFFH.
12/10/85: The problem also arises if you compare a constant against an unsigned short. For example if you declared:

```
#define constant ~0
unsigned short  var;
```

and later compared these two the compiler will zero out the upper byte of the variable var and then compare it to FFFFH. Thus, the condition is never met.

12/16/85: Another example of incorrect code being generated when a char variable is used in a test condition is as follows:

```
char a;
main()
{
  a = -1;
  if(a == -1)
    a ='A';
}
```

Temporary solution:
IF THE LINE IN QUESTION IS CHANGED TO:

```
if ((unsigned short)digit[digit_index]--){
```

CORRECT CODE IS GENERATED ALTHOUGH digit[] HAS ALREADY BEEN DECLARED unsigned short.
12/10/85: Declare the constant as a short. In other words:
#define constant 0FFH.
12/16/85: If only 128 valid characters are required the variable can be declared as a short int.

Signed off 08/25/86 in release 01.50

KPR #: D200037135   Product: 6809 C   500 64822S001   00.00

Keywords: PASS 3

One-line description:
Compiler option $LIST_OBJ ON$ generates wrong output information.

Problem:
Use of the compiler option $LIST_OBJ ON$ may result in incorrect data being output to the list file. In selected cases, machine code will be incorrectly listed. For example, consider the following Pascal program.

KPR #: D200037135 **CONTINUED**

```
  $EXTENSIONS ON$
  $LIST_OBJ ON$
  PROGRAM test;

     VAR
        a, b : BOOLEAN;

     PROCEDURE one;

        BEGIN
           a := b;
        END;
```

In the example listed above, the output file will denote machine code of the form FFFFC00001 for one of the generated assembly statements. The correct value should be C8000001. This problem is caused by an incorrect "printf" mask when generating the output file.

NOTE: THIS DEFECT IS ONLY PRESENT IN THE GENERATED LISTING FILE.
THE GENERATED CODE IS CORRECT.

Signed off 08/25/86 in release 01.50

KPR #: D200040766   Product: 6809 C   500 64822S001   00.00

Keywords: PASS 3

One-line description:
Pass 3 fails to detect relative jump address out-of-range.

Problem:
Pass 3 of the compilation process may fail to detect a relative jump which is out of range. In the test program submitted the relative jump is generated for an IF..THEN statement while the compiler option OPTIMIZE is enabled. [BLINK_TAS:BUG]

Temporary solution:
As a temporary work around disable the compiler option OPTIMIZE around those sections of code which are suspect.

Signed off 08/25/86 in release 01.50

KPR #: D200041335   Product: 6809 C   500 64822S001   00.00

One-line description:
Problem with integer pointer in conditional statement.

Problem:
In the following example, two loads are performed, but no other code is generated to check for zero value.

```
"C"
"processor name"
#define  NULL  0
```

KPR #: D200041335 **CONTINUED**

```
fct(parm)
int *parm;
{
  if (parm - NULL)
     parm = 10;
}
```

Signed off 08/25/86 in release 01.50

KPR #: D200045971   Product: 6809 C   500 64822S001   00.00

One-line description:
Title description is incorrect.

Signed off 08/25/86 in release 01.50

KPR #: D200047613   Product: 6809 C   500 64822S001   00.00

One-line description:
TOO MANY ERRORS IN PASS 3 IF >127 PROCEDURES

Signed off 08/25/86 in release 01.50

KPR #: D200051276   Product: 6809 C   500 64822S001   01.20

One-line description:
++ and -- operators evaluated with improper precedence.

Problem:
According to Kernighan and Ritchie, page 43, the following expressions are equivalent:

```
Example 1:  array[index++] = 1;
Example 2:  array[index] = 1;
            index++;
```

However, different code is generated for these expressions. The second example is compiled correctly, but the first one increments index before setting array[index] equal to 1. Furthermore, when these statements are executed in a main program, an unintialized and unknown variable, Dmain, is used to index into array when the variable index is supposed to be used.

Temporary solution:
Separate the expression as shown in example 2.

Signed off 08/25/86 in release 01.50

KPR #: D200059030   Product: 6809 C   500 64822S001   01.20

One-line description:
Host compilers do not put absolute pats specifications in relocatables

Problem:
Host compilers do not specify the full path name in the relocatable file.

KPR #: D200059030 **CONTINUED**

Signed off 08/25/86 in release 01.50

KPR #: D200069401 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200070557 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

Duplicate Service Requests: D200070581

KPR #: D200080507 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080812 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200082081 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083097 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A

KPR #: D200083097 **CONTINUED**

good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085902 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086298 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088252 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.

KPR #: D200088252 **CONTINUED**

This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090712 Product: 6809 EMULATION 300 64215S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 1650051649 Product: 6809 PASCAL 64813 01.30

One-line description:
If >39 functions declared; following funcs may include bad code.

Problem:
Define 39 procedures with PASCAL 6809, and define this procedure

```
PROCEDURE A40;
VAR CARAC_1:CHAR;
    BEGIN
         WRITE(CARAC_1);
    END;
```

and define another procedure A41 with the same instructions. If you look at the code generated by the compiler there is one more instruction with A40. This instruction is STD DA40+0394BH, between a SEX and a TFR D,Y.
You have the same problem with WRITE(REAL), WRITE(INTEGER), WRITE(UNS.), etc...
You also have the same problem with procedures #42,44,46,48 but not with procedures # 41,43,45,47,...

Temporary solution:
Limit each module of your project to under 40 functions.

s

KPR #: 5000184317 Product: 6809 PASCAL 64813 01.10

One-line description:
Records of pointers to text not handled correctly.

Problem:
The following program causes incorrect code to be generated for writes to TEXT files.

```
"6809"
$EXTENSIONS ON$
$RECURSIVE OFF$
$SEPARATE OFF$$
$GLOBPROC ON$

PROGRAM test;

TYPE
   files     = RECORD  F0,F1 : ^TEXT;
               END;

VAR      CH    :CHAR;
         f     :files;
         g     :TEXT;


PROCEDURE  doit (VAR f:files);

BEGIN
    WRITE( f.f0^,CH);      {LOOKS OK }
    WRITE( f.f1^,CH);      {LOOKS BAD.  NO CALL IS EVEN MADE TO
```

KPR #: 5000184317 **CONTINUED**

```
                                     Pwrite_char. }
END;


BEGIN { MAIN }

WRITE(f.f1^,CH);
WRITE(g,CH);

END.
```

Temporary solution:
No temporary solution.

KPR #: D200060020 Product: 6809 PASCAL 64813 01.09

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200073155 Product: 6809 PASCAL 64813 01.10

One-line description:
ADDR function for stack relative variables in right side conditionals

Problem:
ADDR function for stack relative variables in right side of conditionals

Pascal 6809 defect with the ADDR function:

Comparisons using the ADDR() function with local variables or parameters with $RECURSIVE ON$ may generate incorrect code.

The use of the ADDR(variable) function on the right hand side of comparison expressions can cause incorrect code to be generated.

eg. The statement:

IF pointer <> ADDR(local_var) THEN ...

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of procedures compiled with $RECURSIVE ON$(the default value).

No problem occurs for static variables, which may be outer PROGRAM block variables, any external variables, or local variables and parameters of procedures compiled with $RECURSIVE OFF$.

The simple work around solution is to only use the ADDR(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

The following listing illustrates the problem.

```
"PASCAL" PREPROCESS
"6809"
PROGRAM ADDRbug;
$EXTENSIONS$
$RECURSIVE ON$
  VAR GLBi, GLBj: ^INTEGER;
PROCEDURE RecursiveON;
 VAR
    I,J:^INTEGER;
 BEGIN  {Procedure RecursiveON}
$LIST_CODE ON$
   IF ADDR(I)<>     J  THEN ;  { With ADDR on left, This works. }
            LEAX 000000002H,S
            CMPX 000000004H,S
            LBEQ RecursiveON01_1
        RecursiveON01_1
   IF      I <>ADDR(J) THEN ;  { With ADDR on right, it FAILS ! }
```

KPR #: D200073155 **CONTINUED**

```
              LDX  000000002H,S
              CMPX 000000004H,S  *** Value of J, NOT address of J ***
              LBEQ RecursiveON01_2
          RecursiveON01_2
     IF ADDR(I)<>ADDR(J) THEN ;  { With ADDR both sides ALWAYS FAILS !!
}
              LEAX 000000002H,S
              CMPX 000000004H,S  *** Value of J, NOT address of J ***

              LBEQ RecursiveON01_2
          RecursiveON01_3
  $LIST_CODE OFF$
   END;
  BEGIN
  $LIST_CODE ON$
     IF ADDR(GLBi)<>     GLBj  THEN ;  { With static vars it works. }
              LDX  #DADDRbug
              CMPX DADDRbug+00002H
              LBEQ ADDRbug00_4
          ADDRbug00_4
     IF      GLBi <>ADDR(GLBj) THEN ;  { With static vars it works. }
              LDX  DADDRbug
              CMPX #DADDRbug+00002H
              LBEQ ADDRbug00_5
          ADDRbug00_5
     IF ADDR(GLBi)<>ADDR(GLBj) THEN ;  { With static vars it works. }
              LDX  #DADDRbug
              CMPX #DADDRbug+00002H
              LBEQ ADDRbug00_6
          ADDRbug00_6
  $LIST_CODE OFF$
  END.
```

Workaround:

Pascal 6809 defect with the ADDR function:

eg. The statement:

IF pointer <> ADDR(local_var) THEN ...

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of procedures compiled with $RECURSIVE ON$(the default value).

No problem occurs for static variables, which may be outer PROGRAM block variables, any external variables, or local variables and parameters of procedures compiled with $RECURSIVE OFF$.

KPR #: D200073155 **CONTINUED**

The simple work around solution is to only use the ADDR(local_var) function on the left hand side of the comparison expression.

If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

Temporary solution:
IF pointer <> ADDR(local_var) THEN ...

will not generate correct code if the local_var is on the stack. This will occur for local variables or parameters of procedures compiled with $RECURSIVE ON$(the default).

No problem occurs for static vars, which may be outer PROGRAM block variables, any external variables, or local variables and parameters of procedures compiled with $RECURSIVE OFF$.

The simple work around solution is to only use the ADDR(local_var) function on the left hand side of the comparison expression.
If two local_var addresses must be compared, then use of a temporary (pointer) variable to hold the value of one of the two addresses will be required.

KPR #: D200075010 Product: 6809 PASCAL 64813 01.11

One-line description:
With statements used in FOR loops on records may cause error #1006

Problem:
With statements used in FOR loops may cause pass 2 error 1006 on VAX & HPUX 6809 Pascal compilers.

This problem does not occur on 64000 versions of the 6809 Pascal compilers.

The following program illustrates the problem:

```
"PASCAL"
"6809"
PROGRAM P1006;
$EXTENSIONS$
TYPE
    RECORDTYPE = RECORD
       FIELD1, FIELD2, FIELD3 : BYTE ;
    END ;
VAR
    VARTYPE : ARRAY [1..5] OF RECORDTYPE ;
    J,K,L,t1,t2   : BYTE ;

BEGIN   { MAIN }
     FOR J := 1 TO 5 DO
     BEGIN
        WITH VARTYPE[J] DO
          BEGIN
```

KPR #: D200075010 **CONTINUED**

```
               FOR K := FIELD2 TO FIELD3 DO
**** Pass 2 ERROR ?? 1006
                  L := L + 1 ;
            END;
         END;
  END.    { MAIN }
```

Temporary solution:
The workaround is to assign the WITH variables to temporary variables for use in the FOR loop boundary conditions.

```
        WITH    VARTYPE[J]  DO
          BEGIN
             t1 := FIELD2;
             t2 := FIELD3;
             FOR K := t1 TO t2 DO
                 L := L +1;
          END;
```

Duplicate Service Requests: D200075002

KPR #: D200082446 Product: 6809 PASCAL 64813 01.11

One-line description:
Compiler incorrectly assumes the value of a var is in the D register.

Problem:
The compiler assumes it know the value of a variable which it has loaded in the D register, but, the D register is modified by a library call to Zwinset. The pascal code has the following logic:

```
   IF (VARIABLE in arrayOfRecords[].set)  THEN

       IF(array[VARIABLE] = someValue)
```

When the compiler tests the first condition VARIABLE is loaded into the D register and a call is made to Zwinset. Zwinset modifies the D registers. Next, when the compiler is testing the second condition it assumes VARIABLE is still in register D.

Temporary solution:
Anytime the 64000 compilers incorrectly assume a value is in a register try turning AMNESIA on around the offending statements.

$AMENESIA ON$

KPR #: D200082446 **CONTINUED**

```
      IF (VARIABLE in arrayOfStructures[].set)

         IF (array[VARIABLE] = someValue)

$AMNESIA OFF$
```

KPR #: D200087312   Product: 6809 PASCAL   64813   01.60

Keywords: CODE GENERATOR   PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
PROBLEM ON VAX   NOT ON 64100 SYSTEM

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example: "8086"

```
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
LABEL  100;
BEGIN
100:
     GOTO   100;
END;
PROCEDURE  TWO;     { pass 3 error - too many errors in pass 3 }
LABEL  100;         { is generated, without any indication as to }
BEGIN               { what the problem is                        }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get this error message, be aware that you may have duplicate labels in the program.

Signed off 08/31/88 in release A01.70

KPR #: 5000093708   Product: 6809 PASCAL   M 64813-90903   00.02

One-line description:
Parameter passing thru the registers has changed.

Problem:
Further explanation on how we use registers for parameter passing is needed.

Temporary solution:
No temporary solution.

KPR #: D200069419 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200080515 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080820 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

KPR #: D200080820 **CONTINUED**

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081851 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082099 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083105 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

KPR #: D200083105 **CONTINUED**

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085910 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086306 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088260 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090720 Product: 6809E EMULATION 300 64216S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 1650048355 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
68HC11 will work alone as a measurement system.

KPR #: D200082271 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083287 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086066 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

KPR #: D200086066 **CONTINUED**

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086413 Product: 68HC11 EMUL 300 64265S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088377 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090902 Product: 68HC11 EMUL 300 64265S004 01.10

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

KPR #: D200090902 **CONTINUED**

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 5000256867 Product: 68HCII ASSEMB 64865 01.00

One-line description:
Incorrect object code generated for BSET external_sym,mask

Problem:

```
"6811"
DATA
       GLOBAL     SINTAN
       GLOBAL     BIT0
WASTE  RMB        54H
SINTAN RMB        01H
BIT0   EQU        001H
```

File 2:

```
"6811"

       EXTERNAL   SINTAN,BIT0

       BSET       SINTAN,BIT0
```

The object code generated for the BSET instruction should be 145401 instead it is 140101.

Temporary solution:
No temporary solution at this time.

Signed off 08/31/88 in release A01.40

KPR #: D200072397 Product: 68HCII ASSEMB 64865 01.00

One-line description:
Illegal and incorrect object code for STAA, STD operators.

Problem:
For the STAA and STD operators the 68HC11 assembler generates illegal object and incorrect object code respectively if the destination is an immediate operand. An error should be generated.

```
"6811"

       STAA          #33
       STD           #33
```

This file will assembler without error.

Temporary solution:
No known temporary solutions.

Signed off 08/31/88 in release A01.40

KPR #: D200091835 Product: 68HCII ASSEMB 64865 01.00

One-line description:
BAD CODE GENERATED FOR "JSR" INSTRUCTION.

Problem:
The following fragment produces bad code for the first JSR, but not for the second:

```
START   LDY    #1111H
        JMP    0,Y
        CPX    08,Y
        JSR    0,Y
        LDX    07,Y
        CPY    0,Y
        JSR    0,Y
        LDS    0,Y
        END    START
```

Temporary solution:
No workaround is currently available.

Signed off 08/31/88 in release A01.40

KPR #: D200082305 Product: 70016 EMUL (JLO) 300 64294S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200086090 Product: 70016 EMUL (JLO) 300 64294S004 01.10

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                            ; end locks the emulation session
3. <system name> <module name>    ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200088385 Product: 70016 EMUL (JLO) 300 64294S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.

KPR #: D200088385 **CONTINUED**

This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200082313 Product: 70108 EMUL (JLO) 300 64295S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200086108 Product: 70108 EMUL (JLO) 300 64295S004 01.10

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200088393 Product: 70108 EMUL (JLO) 300 64295S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.

KPR #: D200088393 **CONTINUED**

This problem exists for all HP64000-UX emulators built with
/lsd/p2/cmd/emul/gencore.

KPR #: D200082339 Product: 70208 EMUL 64297S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200088419 Product: 70208 EMUL 64297S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with
/lsd/p2/cmd/emul/gencore.

KPR #: 5000242818 Product: 70216 EMUL 64296 01.00

One-line description:
V50 Disassembler generates "illegal" opcode for "POP PS" instruction

---

KPR #: 5000251363 Product: 70216 EMUL 64296 01.00

One-line description:
Can not specify needed trigger specification.

KPR #: D200082321 Product: 70216 EMUL 64296S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200088401 Product: 70216 EMUL 64296S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200086645 Product: 80186 EMUL FW 64764 00.00

One-line description:
Each "init" command eats memory - crash after 20 "init"'s

KPR #: D200087114 Product: 80186 EMUL FW 64764 00.01

One-line description:
Guarded memory might not cause gaurded-mem break

KPR #: D200087601 Product: 80186 EMUL FW 64764 00.00

One-line description:
Help message for the "loc config" item is incomplete

KPR #: D200087676 Product: 80186 EMUL FW 64764 00.01

One-line description:
Invalid "cf mon" setting in firmware gives PC intfc. problems

---

KPR #: D200087684 Product: 80186 EMUL FW 64764 00.00

One-line description:
Invalid "cf mon" setting in firmware gives PC intfc. problems

Problem:
When "cf mon=ufg" or "cf mon=ubg" is selected, a monitor must first be loaded. The firmware currently returns "unknown" status (a question mark) if the user attempts to set either value without first loading a monitor.

The impact on option S006 (the PC user interface) operation is that the configuration screen can't be called up if the configuration in the emulator firmware has any "unknown" settings in it. It would be preferable for the 64764 firmware to reject the operand as invalid if the monitor has not yet been loaded. This way, the emulator firmware will retain the previous setting, rather than marking it as "unknown".

Temporary solution:
The user should be sure to load the monitor BEFORE giving the command to change to user-foreground or user-background monitor.

If the user changes the configuration first and finds that the configuration contains an "unknown" setting for "mon", the user should modify the configuration in the emulator to a valid monitor setting, then load the monitor, and THEN change the configuration.

KPR #: D200087916 Product: 80186 EMUL FW 64764 00.01

One-line description:
Incorrect report of bp when breakpoint feature is disabled

Problem:
It is possible to get the emulator to report that a software

KPR #: D200087916 **CONTINUED**

breakpoint occurred even when the command "bc -d bp" has disabled the breakpoint feature:

```
map -d *
map other grd
map 0..0ffff eram
m 0..1f=99     #make vector table point into grd memory
bc -d bp       #disable the bp feature
m 1000=0cc     #set a "user" int3 at address 1000
r 1000         #run that int3
```

result is message "Unknown software breakpoint: 09999:09999"

Signed off 04/14/88 in release A00.02

KPR #: D200088112 Product: 80186 EMUL FW 64764 00.01

One-line description:
Bad Background monitor

Problem:
During single stepping, if the user's code changes the ES register (so it's different from the DS register value), the registers could end up with incorrect values.

Temporary solution:
If the user steps through an instruction and finds that the register values are wrong, the "reg" command can be used to set the registers to their correct values.

Signed off 04/14/88 in release A00.02

KPR #: D200087957 Product: 80186 EMUL DOS 64764S006 00.01

One-line description:
Modify memory with an invalid string 1..100 could fail

Problem:
If a modify memory command is entered without any data, then the command could hang.

```
ex.  Modify Memory Byte 0..100h <return>
```

Signed off 05/20/88 in release A01.00

KPR #: D200089813 Product: 80186 EMUL DOS 64764S006 00.01

One-line description:
Invalid expressions can corrupt PC memory

Signed off 05/20/88 in release A01.00

KPR #: D200090167 Product: 80186 EMUL DOS 64764S006 00.01

One-line description:
The "stty" command doesn't work correctly for baud rate <= 1200.

Problem:
If you toggle the xon parameter when running at 1200 baud and below, the stty command will return invalid characters.

```
>stty
 stty A 1200 xon
>stty -xon
 [#!,*&^junk characters
```

Since the PC interface calls the stty command upon startup, this problem will make the PC interface fail at startup with a datacomm error at 1200 baud (all lower baud rates are not supported by the PC interface).

Temporary solution:
To get around this problem, just set switch 13 on the emulator's back panel (enable xon). The stty parameter will not be toggled and PC interface will startup successfully.

From the terminal-mode interface, just enter another carriage-return to regain proper communications.

KPR #: D200089847 Product: 80186 EMUL FW M 64764-90901 01.00

One-line description:
The Manual says that step is not allowed in real time mode.

KPR #: 1650044016 Product: 80186 EMULATION 64224 00.00

One-line description:
"run from ‹addr›", "modify reg ‹reloc›" generates 16 extra I/O writes.

Problem:
The commands "Run From ‹addr›", "modify register ‹name›", where ‹name› is a relocatable register such as UMCS, LMCS, generates a series of 16 I/O writes on addresses from 0 to 1EH incrementing by 2 (0,2,4,6,8....) from the start of the Perpheral Control Block.

KPR #: 2700005280 Product: 80186 EMULATION 64224 00.00

Keywords: EBPP

One-line description:
80186 Emulator with EBPP and State calls the wrong disassembler.

Temporary solution:
Use the following procedure to avoid using a "disassemble using " command every time the analysis session is reentered.

```
     measurement_system
     state_#             ;(# = slot number)
     show tracelist
     disassemble using I80186E
     show trace_specification
     configuration save_in C80186_EX:HP write_protect
     end
     end
     purge C80186_E:HP:trace
     rename C80186_EX:HP:trace to C80186_E:HP:trace
```

Signed off 02/23/87 in release 01.04

KPR #: 5000211557 .Product: 80186 EMULATION 64224 01.04

One-line description:
"disp. memory mnemonic" shows incorrect inv. assembly for JMP NEAR inst

Problem:
"display memory mnemonic" shows incorrect inverse assembly for JMP NEAR instruction when MOV for segment register appears before the JMP NEAR.

```
 EXAMPLE source code:  MOV DS,BX
                       JMP NEAR PTR 1000H

   disassembled code:  MOV DS,BX    JMP 0FFFH
```

Temporary solution:
There is no workaround available.

KPR #: 5000225748 Product: 80186 EMULATION 64224 01.05

One-line description:
LODS instructions with segment override not properly disassembled.

Problem:
LODS instructions with segment override not properly disassembled. Example:

```
The code
        2EH,0ACH  represents a   LODSB  CS:[SI] instruction
when displayed mnemonic the opcode is shown as:
        LODS   ES:BYTE PTR[DI],CS:[SI]
```

other LODS instructions with segment override disassemble incorrectly also.

Temporary solution:
There is no workaround available.

KPR #: D200015123 Product: 80186 EMULATION 64224 01.03

One-line description:
PROBLEMS APPEAR WHEN LISTING MEMORY INCLUDING ADDRESSES 0FFFEH & 0FFFFH.

Problem:
In emulation, attempting to list certain addresses or ranges of addresses to a file results in the following problem:

When listing mnemonically a range with either 0FFFEH or 0FFFFH as the end address, the list operation never appears to finish. Instead, the listing loops back around to address 0H and proceeds to list all memory locations.
Examples of the command format:
a) LIST ‹file name› MEMORY XXXXH THRU 0FFFEH mnemonic
b) LIST ‹file name› MEMORY XXXXH THRU 0FFFFH mnemonic

The problem occurs with these addresses when the number of address bits designated in the configuration question is 16. If 20 address bits are assigned, the problem occurs with addresses 0FFFFEH and 0FFFFFH.

Signed off 02/23/87 in release 01.04

Duplicate Service Requests: 5000210799

KPR #: D200033647 Product: 80186 EMULATION 64224 01.03

Keywords: DISASSEMBLER

One-line description:
Disassembler displays segment override though it is not coded.

Signed off 02/23/87 in release 01.04

KPR #: 1650042606 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

Signed off 08/31/88 in release A01.20

KPR #: 1650042630 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
"trace only status INTACK" always displays interrupt type 0.

Problem:
When using "trace only status INTACK", the display always shows an interrupt type 0.

Signed off 08/31/88 in release A01.20

KPR #: D200081166 Product: 80186 EMULATION 300 64224S004 01.00

One-line description:
Loading/modifing configuration after continue may cause reset.

Problem:
Loading or modifying configuration without changing certain key items may cause a processor reset if it is done after exiting emulation, then continuing the emulation session whereas no reset would occur if the user had not done an exit/continue.

Temporary solution:
Temporary work around: avoid using exit/continue in conjunction with modifing or loading configuration in order to avoid a processor reset.

Signed off 04/07/88 in release A01.10

KPR #: D200081208 Product: 80186 EMULATION 300 64224S004 01.00

One-line description:
Modify/Store memory abort at physical addr 0 for seg/offset procs

Problem:
Modify memory for a large range (> 4096 bytes) and store memory (> 250 bytes) that crosses the physical address 0 boundary will fail somewhere after physical address 0 on segment:offset processors. For example, on the 8086, the following command will modify only part of the requested range:

```
        modify memory 0FF00H:0 thru 0FF00H:0FFFFH to 0
```

since address 0FF00H:1000H is in fact physical address 0.

KPR #: D200081208 **CONTINUED**

Temporary solution:
Temporary work around:

Do not attemt to modify/store memory through physical address 0H.

Signed off 04/07/88 in release A01.10

KPR #: D200082131 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 08/31/88 in release A01.20

KPR #: D200083147 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

Signed off 08/31/88 in release A01.20

KPR #: D200085951 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                         ; end locks the emulation session
3. <system name> <module name>   ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

Signed off 08/31/88 in release A01.20

KPR #: D200086314 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

Signed off 08/31/88 in release A01.20

KPR #: D200088278 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

Signed off 08/31/88 in release A01.20

KPR #: D200089839 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
Support baunload feature in the emulator.

Problem:
Detailed Listing for Defect Number LSDqf03472
Text:
  Support bbaunload feature in the emulator.

.submitter

Add the unload capability of the Basis Branch Analyzer to the I80X8X family of emulators.

THIS ENHANCEMENT APPLIES TO THE FOLLOWING PROCESSORS:

64224	80186DQ Emulation
64225	80188DQ Emulation
64220	8086DQ Emulation
64221	8088DQ Emulation

Signed off 08/31/88 in release A01.20

KPR #: D200089854 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
SOURCE LINES are missing from "absolute trace display" with "SOURCE ON"

Problem:
Detailed Listing for Defect Number LSDqf03582
Text:
  Source lines missing from absolute trace display with source on

"display trace absolute source on" does not show source lines in the dis play.
"display trace mnemonic source on" will show the source lines. Source l ines
should appear for both types of display.

Signed off 08/31/88 in release A01.20

KPR #: D200090761 Product: 80186 EMULATION 300 64224S004 01.10

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the

KPR #: D200090761 **CONTINUED**

source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/31/88 in release A01.20

KPR #: D200031799 Product: 80186 SW ANAL 64335 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.03

KPR #: D200031856 Product: 80186 SW ANALYZER 64341E 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.02

KPR #: D200087973 Product: 80188 EMUL FW 64765 00.01

One-line description:
Incorrect report of bp when breakpoint feature is disabled

Problem:
It is possible to get the emulator to report that a software breakpoint occurred even when the command "bc -d bp" has disabled the breakpoint feature:

```
map -d *
map other grd
map 0..0ffff eram
m 0..1f=99      #make vector table point into grd memory
bc -d bp        #disable the bp feature
m 1000=0cc      #set a "user" int3 at address 1000
r 1000          #run that int3
```

result is message "Unknown software breakpoint: 09999:09999"

Signed off 04/14/88 in release A00.02

KPR #: D200088120 Product: 80188 EMUL FW 64765 00.01

One-line description:
Bad Background monitor

Problem:
During single stepping, if the user's code changes the ES register (so it's different from the DS register value), the registers could end up with incorrect values.

Temporary solution:
If the user steps through an instruction and finds that the register values are wrong, the "reg" command can be used to set the registers to their correct values.

KPR #: D200065805 Product: 80188 EMULATION 64225 01.03

Keywords: USER MEMORY

One-line description:
Emulator would not recover from errors during display memory repetitive.

Problem:
The problem occurs when displaying user memory repetitively. An error condition such as slow clock or guarded memory access would cause the 64000 station to reboot or to display extraneous data at the top of the screen. When the screen had been written to at the top, the only action to delete the characters was resetting the station.

Temporary solution:
There is no workaround other than avoiding the error conditions during a repetitive display of user memory.

KPR #: D200081265 Product: 80188 EMULATION 300 64225S004 01.00

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:
Display Memory and Modify Memory will be incorrect at the segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the center of a line.

Modify Memory will be incorrect if done beyond the end of a segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment in the middle of the line being displayed.

Modify Memory will be correct if it is not extended through the end of a segment. For example:

```
        modify memory OFFFEH to 1,2 will be correct.

        modify memory OFFFEH to 1,2,3 will NOT be correct, because
          the third entry is in the next segment.
```

Signed off 08/31/88 in release A01.20

KPR #: D200082149 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 08/31/88 in release A01.20

KPR #: D200083154 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

Signed off 08/31/88 in release A01.20

KPR #: D200084939 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

Signed off 08/31/88 in release A01.20

KPR #: D200084954 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
"trace only status INTACK" always displays interrupt type 0.

Problem:
When using "trace only status INTACK", the display always shows an interrupt type 0.

Signed off 08/31/88 in release A01.20

KPR #: D200086132 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
Software Breakpoints don't work in target memory.

Problem:
Software breakpoints do nct work in target memory.
There is no workaround; updated software is required.

Temporary solution:
Software breakpoints do not work in target memory.
There is no workaround; updated software is required.

Signed off 08/31/88 in release A01.20

Duplicate Service Requests: D200089904

KPR #: D200086322 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

Signed off 08/31/88 in release A01.20

KPR #: D200088286 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

Signed off 08/31/88 in release A01.20

KPR #: D200090779 Product: 80188 EMULATION 300 64225S004 01.10

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the

KPR #: D200090779 **CONTINUED**

source line.

Temporary solution:
End out of emulation, and reenter before loading the new program
or executing the trace.

Signed off 08/31/88 in release A01.20

KPR #: D200031807 Product: 80188 SW ANAL 64336 02.01

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer
to lock up. The analyzer looks through the dynamic data area without
ever realizing that the variable is in the static data area, thus an
infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use
local static variables until new software is available.

Signed off 11/06/87 in release 02.04

KPR #: D200031864 Product: 80188 SW ANALYZER 64341F 01.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 01.02

KPR #: 5000240259 Product: 80286 EMULATION 64228 01.02

One-line description:
trace only ‹Odd Address› data 0: analyzer doesn't qualify properly.

Problem:
Trigger and store function in 80286 emulator does not function properly when an odd address is specified with a data qualifier. The problem is that the data quailifier appears to be ignored.
For example:
   "trace only address 1463DH data 0"
will show ALL accesses to 1463DH, not just those with data = 0H

also:
   "trace only address 1463Dh data 03EXXH"
will show all accesses to 1463DH

also:
    "trace only address 1463DH data 0XX3EH"
will not capture any data, however
    "trace only address 1463DH data 0XX00H" will capture all accesses
(The program used to test this writes consectutive values from 0-FFH to location 1463DH)

KPR #: 5000244343 Product: 80286 EMULATION 64228 01.02

One-line description:
80286 emul. fails to run programs mapped as user memory at the target.

Problem:
The 80286 emulator fails to run programs mapped as user memory at the target system if the target system makes frequent hold requests.
This problem results from a cpu misinterpretation of an ACK signal.
The emulator generates a HOLD-ACK signal at the end of every emulator controller's hold cycle. If the target system makes a hold request near the HOLD-ACK signal, the target system may misuse this ACK as its own ACK, and immediately initiates BUS master operation. But the 80286 cpu continues to execute normal bus cycles. The CPU's read/write instruction fails because the data/address bus is used by another bus master.

Temporary solution:
There is no workaround available at this time.

KPR #: 5000273250 Product: 80286 EMULATION 64228 01.02

One-line description:
80286 Emulator may not display proper Interrupt Type number.

Temporary solution:
There is no workaround available at this time.

KPR #: 5000273268 Product: 80286 EMULATION 64228 01.02

One-line description:
trace abt addr 0:0E0H status rd mem triggers on addresses 0E0h, 0C0H.

Problem:

KPR #: 5000273268 **CONTINUED**

Special combination of address and status causes improper analyzer spec in 80286 emulator.

The trace command:

```
 "trace about address 0:0E0H status read_mem"  will trigger on addresses
0E0H and 0C0H. It should only trigger on 0E0H.
```

The commands:

```
  "trace about address 0:0E0H"    AND
  "trace about address 0E0H status read_mem"
```

work properly.

KPR #: 5000275727 Product: 80286 EMULATION 64228 01.02

One-line description:
"list printer memory" command gives wrong addresses using seg:offset.

Problem:
The "list printer memory" command does not function properly. Two errors have been observed.

1. If the "list printer memory" command is issued with the address specified in the segment:offset format the addresses on the printout do not have a colon separating the segment from the offset. The resulting addresses are not the addresses specified.
2. If the "list printer memory" command is issued multiple times after a "display memory" command (again with the address specified as segment:offset) the addresses in the printout are not the addresses specified.

KPR #: D200080127 Product: 80286 EMULATION 64228 01.02

One-line description:
First PV cycle shows failure with some 64155B cards, if PV'd 1st on 228.

Problem:
When the 64228 is in the same cage as a 64155B, and PV is run on the 228 before being run on the 155B, the first cycle of PV will fail with certain 155B cards, and not with others. PV passes when perfomed on the 155B card before the 228.

Temporary solution:
This problem does not influence the operation of the emulator, and is merely an inconvenience, especially however, for those who do not expect it to occur.

KPR #: 5000141747 Product: 80286 UDE 64227 01.00

Keywords: DISPLAY MEMORY

One-line description:
Inverse assembler does not work properly during display memory mnemonic.

Problem:
The "MUL " instruction is not correct when shown by a "display memory mnemonic" command. The inverse assembler cosiders it a 3 byte instruction rather than a 2 byte instruction. For example

```
3015    MOV BX,#0010H
3018    MUL BX
301B    MOV 3002H,AX          * the correct address is 301A
301E    JMP s 03004H          * this address is correct
```

The trace disassembly is correct.

KPR #: 5000162651 Product: 80286 UDE 64227 01.00

Keywords: DISPLAY MEMORY

One-line description:
The IDIV instruction is not correct during a display memory mnemonic.

Problem:
The trace disassembly is correct. The problem is only with display memory mnemonic. An example follows.

```
3009    MOV BX,#0100H
300C    IDIV BX
300F    MOV 3000H,AX     * address should be 300E
3012    MOV ax,#0100H    * this address is correct
```

KPR #: 5000181131 Product: 80286 UDE 64227 01.00

Keywords: DISASSEMBLER

One-line description:
Incorrect data is returned on a trace about an I/O port.

KPR #: D200046714 Product: 80286 UDE 64227 01.00

Keywords: INSTRUCT. EXECUTION

One-line description:
Single step function does not work after a software breakpoint.

KPR #: D200068775 Product: 80286B ASSEMB 64859 01.02

One-line description:
Aliases not allowed in the linker to specify library paths.

Problem:
Path specifications are not allowed for libraries in the linker on old assembler, you could use aliases; you can not use aliases now.

Temporary solution:
No known temporary solutions.

KPR #: D200085316 Product: 80286B ASSEMB 64859 01.02

One-line description:
Address in 8086 family assemblers lost segment information.

Problem:
Due to changes to the hosted assembler, all assemblers in product 64853 and 64859 (80x86 family and 80286, B version) were truncating the segment information in addresses passed to the linker.

Signed off 08/31/88 in release A01.50

KPR #: 5000132662 Product: 8048 ASSEMB 64846 01.00

One-line description:
Error message LR generated on valid JMP instruction

Problem:
The 8042 processor allows jumping through 2K blocks called pages
The following example generates a LR error for a valid JMP opcode.

```
   "8042"
        ORG    401H
   LABEL   NOP
           NOP
           PROG
           JMP     LABEL       {opcode 8401 - is valid, p.14-19
                   ^LR error       Micontroller Handbook}
```

Temporary solution:
No known temporary solution.

KPR #: 5000169995 Product: 8051 ASSEMB 64855 01.08

One-line description:
Assembler inconsistant in permitting forward referencing

Problem:
The assembler does not always allow forward referencing. It is not clear why it allows forward referencing sometimes but not others.

```
"8051"

     MOV SYMBOL,C  ; no error
     MOV C,SYMBOL  ; DE error why?
     EXT SYMBOL

     ORL A,SYMBOL2 ; no error
     EXT SYMBOL2
     END
```

Temporary solution:
Define all externals before referencing them.
(In this case customer does not like this workaround, because a large amount of code was written under rev 1.06 and these errors did not occur.)

KPR #: 5000171470 Product: 8051 ASSEMB 64855 01.08

One-line description:
Defining a transfer address causes an ET error

Problem:
The following program generates an ET error.

```
"8051"

start   NOP
        NOP
        END start
```

Therefore, a transfer address cannot be defined. This is a critical need for our customers for emulation

Temporary solution:
No known temporary solution.

KPR #: 5000240929 Product: 8051 ASSEMB 64855 01.20

Keywords: CODE GENERATOR PROBLEM ON 9000/S300

One-line description:
Special operator "HIGH" does not work with DS pesudo opcode

Problem:
The special operator "HIGH" does not work correctly when the label is defined using the DS pseudo opcode: Example:

KPR #: 5000240929 **CONTINUED**

```
"8051"
       ORG   1234H
LABEL1 EQU   $
LABEL2 EQU   3344H
LABEL3 DS    1
       MOV   A,#HIGH(LABEL1)  ;correct - moves 12H into A
       MOV   A,#HIGH(LABEL2)  ;correct - moves 33H into A
       MOV   A,#HIGH(LABEL3)  ;WRONG - moves 34H into A
       MOV   DPTR,#LABEL3     ; correct
       END
```

Temporary solution:
There is no known work around at this time.

KPR #: D200049833 Product: 8051 ASSEMB 64855 00.00

One-line description:
Link maps produced on VAX are different than on 64000 and are wrong.

Problem:
If files are assembled and linked on the VAX, the link map produced is different than if they are assembled and linked on the 64000. The length of object files other than the first one listed is too long.

Here is an example of the link maps produced:

On the 64000:

```
FILE/PROG NAME         PROGRAM
file1                   1000
file2                   10C8

next address            1190
```

On the VAX:

```
FILE/PROG NAME         PROGRAM
file1                   1000
file2                   10C8

next address            1258
```

If the code linked on the VAX is transferred to the 64000 and run on an emulator, the file doesn't run, and the message "accesses to guarded memory" is displayed. The code linked on the 64000 runs with no problems on the emulator. The state analyzer also has problems with code linked on the VAX. When a trace is done, the modules are displayed relative to the monitor.

Signed off 04/05/88 in release A01.08

KPR #: D200068379 Product: 8051 ASSEMB 64855 01.08

One-line description:
HIGH operator does not function correctly

Problem:
The HIGH operator does not work when accessing data memory. It assumes that external ram is 8 bits instead of the 16-bits. So no matter what you use as an example, you always get the LOW byte.

Temporary solution:
No known temporary solution.

KPR #: D200081570 Product: 8051 ASSEMB 64855 01.08

Keywords: CODE GENERATOR

One-line description:
HIGH does not work

Problem:
HIGH operator does not work

Temporary solution:
There is no known work around.

KPR #: D200091710 Product: 8051 ASSEMB 64855 01.08

One-line description:
CONT in linker will overwrite addresses of variables in different module

Problem:
The CONT command used during link will overwrite variable addresses declared in different modules.

```
----------------------  MOD1 -----------------------------
"8051"
        EXT LAB1,LAB2

        PROG
        MOV     LAB1,#01H
        MOV     LAB2,#02H

---------------------   MOD2  ------------------------------
"8051"
        GLB LAB1,LAB2,LAB3

LAB1    DS      1
LAB2    DS      1
LAB3    DS      1

---------------------   MOD3  ------------------------------
"8051"
        GLB LAB4,LAB5
LAB4    DS      1
LAB5    DS      1
```

KPR #: D200091710 **CONTINUED**

After assembling, link all three modules together using the CONT command for the addresses of the last two modules. The XREF listing will look like this:

```
           LAB1       D   0020  <-----
           LAB2       D   0021
           LAB3       D   0022
           LAB4       D   0020  <----- Should continue, not start
           LAB5       D   0021         over
```

Temporary solution:
Define all variables in one module, or declare the addresses during link and don't use CONT.

KPR #: 5000206458 Product: 8051 ASSM M 64855-90902 01.05

Keywords: CODE GENERATOR

One-line description:
In the manula pg 8-2 states the BIT instruc. shows operand is address.

KPR #: D200086439 Product: 8051 ASSM M 64855-90902 01.07

Keywords: MANUAL

One-line description:
The assmblr manual needs to be updated w/ information in reference manul

KPR #: 5000135855 Product: 8051 ASSM + AL REF M 64855-90905 01.05

One-line description:
The $ operand does not work as defined.

Problem:
If the $ operand is used in a multi-byte instruction, it should specify the value of the PC at the beginning of that instruction. In the following example, it represents the value of the PC in the middle of the MOV instruction:

```
"8051"
      ORG     10H
      MOV     A,#$        ; moves 11H into A instead of 10H

      END
```

Temporary solution:
Use $-x instead of $ where x represents the offset back to the first byte of the multi-byte instruction:

```
"8051"
      ORG    10H
      MOV    A,#$-1    ; this will move 10H into A
      END
```

Signed off 04/07/88 in release Z00.00

KPR #: 5000183475 Product: 8051 EMUL M 64264-90901 01.01

One-line description:
Manual enhancement to reflect Port display info in more detail.

KPR #: 1650042655 Product: 8051 EMULATION 64264 00.00

One-line description:
Cannot load absolute file using remote file access.

Problem:
Cannot load absolute file using remote file access(RFA).

KPR #: 5000280750 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
"Core dumps" when displaying user memory<odd address> blocked word.

Problem:
8051 emulator will "end_release" with a core dump if a display of external_data_memory blocked word is requested with an odd address and the memory is mapped as target. Example:

```
  display external_data_memory 1 blocked words
```

will cause the emulator to end release if address 1 is mapped as "external_data_memory target ram"

KPR #: D200069575 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200075788 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Msg "Monitor must reside in emul pgm mem" is flaky

KPR #: D200077438 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Monitor is not recognized when overwritten,re-entered after "end-lock".

KPR #: D200080689 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in

KPR #: D200080689 **CONTINUED**

the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080986 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081935 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082263 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

KPR #: D200082263 **CONTINUED**

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200082453 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Emulator end_releases when displaying int. data mem. repet. at odd addr.

Problem:
When displaying internal_data_memory at an odd address repetitively, the emulator end_releases with the error "HP 64120 I/O failed - Bad Address"

Temporary solution:
Specify an even address or a range that does not exceed the internal data memory space (maximum 7fh).

KPR #: D200083279 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200084913 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

KPR #: D200086405 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

KPR #: D200086405 **CONTINUED**

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088369 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090894 Product: 8051 EMULATION 300 64264S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 5000219220 Product: 8080/5 ASSEMB 64840 01.00

Keywords: CODE GENERATOR

One-line description:
xref incorrect with conditional assmbly IF when code generated for false

Problem:
Cross reference listing is incorrect when using IF statement of conditional assembly. When code is generated for FALSE part of IF, the cross reference may be for the wrong value.

Temporary solution:
No known solution at this time.

KPR #: 5000129023 Product: 8085 B PASCAL 64825 01.01

Keywords: PASS 1

One-line description:
$Range ON$ causes incorrect code to be generated for a test operation.

Problem:
The following program when compiled with the $RANGE ON$ option wil cause incorrect code to be generated.

```
"B8085" | "BZ80"
$EXTENSIONS$
$RANGE ON$

PROGRAM  BOOLREAL;

VAR  A,B,C    :   REAL;
     L        :   BOOLEAN;

BEGIN
     A := 10.0;
     B := 15.0;
     C := 12.0;


     L := (C < (B+.5)) AND ((C + .5) > A);
END.
```

The two intermediate results "(C < (B +.5))" and "((C+.5) >A)" are anded together and this result is compared with the value two. Thus the case is never true. With RANGE OFF correct code is generated.

Temporary solution:
It is necessary to turn $RANGE OFF$ to obtain correct code. Simply breaking up the expression will not work.

KPR #: D200060228 Product: 8085 B PASCAL 64825 01.02

One-line description:
Incorrect data offsets in listing file.

Problem:
I am expanding this to all pascal compilers. The C compilers list the correct offset. $FAR ON$ only applies to the 68000 cross compiler. The other compilers exhibit the defect w/o any options on.

```
"processor name"
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
```

KPR #: D200060228 **CONTINUED**

```
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

```
KPR #: D200087353   Product: 8085 B PASCAL        64825            01.90

Keywords: CODE GENERATOR       PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
          PROBLEM ON VAX       NOT ON 64100 SYSTEM
```

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example:

```
"8086"
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
LABEL  100;
BEGIN
100:
     GOTO   100;
END;
PROCEDURE  TWO;     { pass 3 error - too many errors in pass 3 }
LABEL  100;         { is generated, without any indication as to }
BEGIN               { what the problem is                        }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get this error message, be aware that you may have duplicate labels in the program.

KPR #: D200087353 **CONTINUED**

Signed off 08/31/88 in release A02.00

KPR #: D200069948 Product: 8085 C 64826 01.03

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                  /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200081562 Product: 8085 C 64826 01.04

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
          ;
}
```

KPR #: D200081562 **CONTINUED**

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;

  OR
```

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200090258 Product: 8085 C 64826 02.10

One-line description:
Indirect func calls fail following calls with temporary results

Problem:
Copied from Lab text:

Text:
Indirect func calls fail following calls with temporary results.

When a function is called indirectly, the call is performed by pushing the return address onto the stack, loading the call address into the HL register, and then moving HL into the PC. If a temporary is located in a register (as when the result of a previous call is a temporary), then the temporary is stored to memory after the call address has been loaded into HL.

In order to store the temporary in memory it may be neccessary to use the HL register. The following code illustrates how the temporary in DE is swapped with the call address in HL (during the store of the temporary), but the call address is not returned to HL prior to loading the PC from HL, thus causing the call to fail to access the correct

KPR #: D200090258 **CONTINUED**

location.

```
"C"
"8085"
$RECURSIVE OFF$
$EXTENSIONS ON$

typedef int  (*PFI)();
typedef struct {
                   char proc_name[10];
                   PFI  proc_ptr;
           } proc_entry;

proc_entry proc_tab[10];

  LSDqf01386()
   {
     int i;
      if (  (*proc_tab[i].proc_ptr)() <=  (*proc_tab[4].proc_ptr)()  )
      {
      }
    }
```

NOTE: Repair was effected in GEN_CALL/call.P by setting PROTECT_HL to true while HL contains the call address.

Signed off 08/31/88 in release A02.20

KPR #: D200055889 Product: 8085 C M 64826-90901 01.02

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor name"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int    cntr;
  int    (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.04

KPR #: D200052316 Product: 8085 C 300 64826S004 00.00

Keywords: CODE GENERATOR

One-line description:
Incorrect opcode "MOV A,ACC" allowed by our assembler

Problem:
The instruction "MOV A,ACC" was assemble and emulated by our products; however, the Intel 8051 goes into the weeds at this instrcution. At first glance the machine code in the asembler listing appears valid (MOV A,ACC ->0000 E5E0), but the bottom of page 8-35 in Intel's microcontroller handbook states: *MOV A,ACC is not a valid instruction.

Neither our manuals nor AMD's user manual mention this instruction.

Signed off 08/25/86 in release 01.00

KPR #: 1650037556 Product: 8085 EMULATION 300 64203S004 01.10

One-line description:
Display user memory causes a time-out, requiring end-release to recover.

Signed off 11/10/87 in release 01.20

KPR #: 1650053660 Product: 8085 EMULATION 300 64203S004 01.00

One-line description:
I/O Failure message when "run from <> until <>"; HPUX 6.0.

Problem:
When the HP-UX operating system is updated to version 6.0 problems occur in emulation.
As an example "run from ADDRESS until ADDRESS" causes the error message "HP64120 I/O failed - check HP-IB and power to 64120". The only softkey that is left is the "end release_system" softkey.

workaround: run emulation software in the debug mode by creating the file /usr/hp64000/log/adb.X.0, where X represents the select code for the HPIB interface.
In addtion the stderr must be redirected to /dev/null each time the emulator is invoked.
example: ms8085 em8085 2>/dev/null

Signed off 08/31/88 in release A01.40

KPR #: D200080226 Product: 8085 EMULATION 300 64203S004 01.00

One-line description:
Absolute code part user,part emul, will be overwritten at boundary.

Signed off 11/10/87 in release 01.20

KPR #: D200080804 Product: 8085 EMULATION 300 64203S004 01.10

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

KPR #: D200080804 **CONTINUED**

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

Signed off 11/10/87 in release 01.20

KPR #: D200081406 Product: 8085 EMULATION 300 64203S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

Signed off 11/10/87 in release 01.20

KPR #: D200090670 Product: 8085 EMULATION 300 64203S004 01.30

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and rerums the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and rerums the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the

KPR #: D200090670 **CONTINUED**

source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/19/88 in release A01.40

KPR #: D200013334 Product: 8085 PASCAL 64810 00.70

Keywords: CODE GENERATOR

One-line description:
Compiler generates incorrect code for BOOLEAN assignment statement.

Problem:
The following program displays a code generation error with regard to BOOLEAN assignment statements. The BOOLEAN assignment statement overwrites a value in the H&L register pair which is relied upon later.

```
PROGRAM BAD_LOOP;

   VAR
      A, B : BYTE;
      ERROR : BOOLEAN;

   BEGIN
      REPEAT
      UNTIL A < B;
      ERROR := FALSE;     { overwrites H&L which contain 'A' }
      FOR A := 1 TO B DO  { uses H&L assuming 'A' still in register }
   END.
```

Temporary solution:
No known workaround at this time.

KPR #: D200040600 Product: 8085 PASCAL 64810 00.70

One-line description:
Run time UNDERFLOW error using ZDSBSUB library if result has even parity

Problem:
Byte subtraction with $DEBUG ON$ will cause an underflow error if the result has even parity. An underflow will be incorrectly flagged if the result has even parity. No error will be indicated, even if one exists, if the result has odd parity. The problem is in ZDsbsub (Debug signed byte subtraction). The 8085 interprets PE exclusively as a parity bit, while the library is anticipating that the bit can be interpreted as an overflow bit.

SAMPLE CODE:

```
"C"
"8085"
$DEBUG ON$   /*This is required for the error to occur*/
main()
 {
    short small;
    short zero;
    small = -128;
    zero = small - small;   /* causes error */
 }
```

This problem affects 8085 C and Pascal compilers on 64000 and hosts.

Temporary solution:

KPR #: D200040600 **CONTINUED**

Turn $DEBUG OFF$ around signed byte subtractions.

KPR #: D200069427 Product: 8086 DQ EMUL 300 64220S004 01.10

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Signed off 11/05/87 in release 01.10

KPR #: D200075838 Product: 8086 DQ EMUL 300 64220S004 01.10

Keywords: DISASSEMBLER

One-line description:
8087 FSUB & FDIV instructs are not disassembled properly.

Problem:
The 8087 coprocessor FSUB and FDIV instructions are not disassembled properly in certain cases. The absolute information is correct.

Signed off 08/31/88 in release A01.30

KPR #: D200080432 Product: 8086 DQ EMUL 300 64220S004 01.10

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:
Display Memory and Modify Memory will be incorrect at the segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the center of a line.

Modify Memory will be incorrect if done beyond the end of a segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment in the middle of the line being displayed.

KPR #: D200080432 **CONTINUED**

Modify Memory will be correct if it is not extended through the end of a segment. For example:

modify memory 0FFFEH to 1,2 will be correct.

modify memory 0FFFEH to 1,2,3 will NOT be correct, because the third entry is in the next segment.

Signed off 08/31/88 in release A01.30

KPR #: D200081067 Product: 8086 DQ EMUL 300 64220S004 01.10

One-line description:
Loading/modifing configuration after continue may cause reset.

Problem:
Loading or modifying configuration without changing certain key items may cause a processor reset if it is done after exiting emulation, then continuing the emulation session whereas no reset would occur if the user had not done an exit/continue.

Temporary solution:
Temporary work around: avoid using exit/continue in conjunction with modifing or loading configuration in order to avoid a processor reset.

Signed off 04/07/88 in release A01.20

KPR #: D200082107 Product: 8086 DQ EMUL 300 64220S004 01.20

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 08/31/88 in release A01.30

KPR #: D200083113 Product: 8086 DQ EMUL 300 64220S004 01.20

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

Signed off 08/31/88 in release A01.30

KPR #: D200085928 Product: 8086 DQ EMUL 300 64220S004 01.20

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

Signed off 08/31/88 in release A01.30

KPR #: D200090738 Product: 8086 DQ EMUL 300 64220S004 01.20

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles,

KPR #: D200090738 **CONTINUED**

relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/31/88 in release A01.30

KPR #: D200031757 Product: 8086 DQ SW ANAL 64332B 01.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 01.02

KPR #: 1650034066 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Software Breakpoints lock up terminal, when mapped to target.

Signed off 10/09/87 in release 01.20

KPR #: 1650034082 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Paging at a segment end produces a confusing CS:IP.

Signed off 10/09/87 in release 01.20

KPR #: 1650038240 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

Signed off 10/09/87 in release 01.20

KPR #: D200069443 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

KPR #: D200069443 **CONTINUED**

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Signed off 10/09/87 in release 01.20

KPR #: D200080549 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

Signed off 10/09/87 in release 01.20

KPR #: D200081190 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Modify/Store memory abort at physical addr 0 for seg/offset procs

Problem:
Modify memory for a large range (> 4096 bytes) and store memory (> 250 bytes) that crosses the physical address 0 boundary will fail somewhere after physical address 0 on segment:offset processors. For example, on the 8086, the following command will modify only part of the requested range:

```
        modify memory 0FF00H:0 thru 0FF00H:0FFFFH to 0
```

since address 0FF00H:1000H is in fact physical address 0.

Temporary solution:
Temporary work around:

Do not attemt to modify/store memory through physical address 0H.

Signed off 10/09/87 in release 01.20

KPR #: D200081240 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:

KPR #: D200081240 **CONTINUED**

Display Memory and Modify Memory will be incorrect at the segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the center of a line.

Modify Memory will be incorrect if done beyond the end of a segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment in the middle of the line being displayed.

Modify Memory will be correct if it is not extended through the end of a segment. For example:

```
        modify memory 0FFFEH to 1,2 will be correct.

        modify memory 0FFFEH to 1,2,3 will NOT be correct, because
          the third entry is in the next segment.
```

KPR #: D200081414 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

KPR #: D200082123 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

KPR #: D200082123 **CONTINUED**

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083139 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200084921 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

KPR #: D200085944 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                            ; end locks the emulation session
3. <system name> <module name>    ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the

KPR #: D200085944 **CONTINUED**

problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200090753 Product: 8086 EMUL 300 64222S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200031740 Product: 8086 SW ANAL 64332 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.03

KPR #: D200031815 Product: 8086 SW ANALYZER 64341A 01.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 01.02

KPR #: 5000254730 Product: 8086-89 ASSM M 64853-90907 02.01

Keywords: MANUAL

One-line description:
8086 Asm/linker manual doesn't doc. valid DQ and DT directives.

Problem:
The 8051 assembler/linker reference manual has paragraph with errors relating to the CSEG directive. The errors are typographical, but can lead to confusion:
Page 8-4 of the HP-UX hosted manual talks about the CSEG directive with a paragraph:

The code segment counters can be charged (changed) with the DS, DW, and DW (DB) pseudos, and with each instruction encoded. Each unit in the program relocatable counter represents one byte in the code address space within the range of 0 to 64. (0 to 64K)

(corrections are indicated in parenthesis)

KPR #: 5000135905   Product: 8086/8 ASSEMB   64853   02.00

One-line description:
Assembler does not allow [SI] as operand for OUTS

Problem:
The assembler will not generate code for the following instructions:

```
           OUTS ,CS:WORD PTR [SI]
           OUTS DX,DS:WORD PTR[SI]
```

According to the Intel manual, DX addresses the port and SI addresses the string.

Temporary solution:
No known temporary solution.

Signed off 08/31/88 in release A02.80

KPR #: 5000172221   Product: 8086/8 ASSEMB   64853   02.03

One-line description:
Labels used in the operand field of a DBS instr causes ET error

Problem:
THE ASSEMBLER CAN NOT ASSEMBLE A FOLLOWING PROGRAM:

```
"70116"
          ORG     10000000H
          ASSUME  DS0:ORG
CCC       LABEL   BYTE
          DBS     1000H
DDD       LABEL   BYTE
          DBS     DDD-CCC     >>> ET ERROR  ..ABSOLUTE-ABSOLUT=ABSOLUTE
EEE       EQU     DDD-CCC
          DBS     EEE         >>> ET ERROR            SAME
          DATA
          ASSUME   DS1:DATA
FFF       LABEL   BYTE
          DBS     1000H
GGG       LABEL   BYTE
          DBS     GGG-FFF     >>>  ET ERROR   RELOC-RELOC=ABSOLUTE
HHH       EQU     GGG-FFF
          DBS     HHH         >>>  ET ERROR            SAME
```

Temporary solution:
No known workaround at this time.

Signed off 08/31/88 in release A02.80

KPR #: 5000215913   Product: 8086/8 ASSEMB   64853   02.03

Keywords: CODE GENERATOR

One-line description:
Incorrect code generated when EQU offset used in MOV REG,REG/MEM

Problem:

KPR #: 5000215913   **CONTINUED**

Incorrect code generated when MOV REG,REG/MEM with EQUated offset is used. For example:

```
"80186"
       PROG
LABL   DB    38H
BASE   EQU    5
       MOV   AL,CS:LABL+5[BX]  ;works correctly-generates code to load
                               ;AL with data contained at [BX+5+address
                               ;of LABL]
       MOV   AL,CS:LABL+BASE[BX]  ;generates wrong code - (B005H) which
       END                     ;is for a MOV  AL,IMMED8 instruction.
```

No error indication is given.

Temporary solution:
There is no work around at this time.

Signed off 08/31/88 in release A02.80

KPR #: 5000227991   Product: 8086/8 ASSEMB   64853   02.02

Keywords: CODE GENERATOR

One-line description:
Using arithmetic to calc address of mem loc of destin. of MOV causes err

Problem:
Assembler causes ET error for the following relocatable format.

```
  "70116"

         ASSUME        PS:PROG,DS0:PROG
  AD     DBS           10
  LAB    EQU           #0FH
         MOV           AD+LAB,#91H
                       ^
                       ET occures
```

Temporary solution:
Use LAB+AD instead of AD+LAB

Signed off 08/31/88 in release A02.80

KPR #: 5000247783   Product: 8086/8 ASSEMB   64853   02.30

Keywords: CODE GENERATOR   PROBLEM ON 9000/S300

One-line description:
Assembler does not handle all string comparisons correctly.

Problem:
Assembler does not handle all string comparisons correctly:

```
"70108"
DEMO1   MACRO   &PARM1
VAL1    SET     1
```

KPR #: 5000247783 **CONTINUED**

```
        .IF     "&PARM1" .EQ. ":char" saw_char
VAL2    SET     2
        .IF     "&PARM1" .EQ. ":long" saw_long
VAL3    SET     3
        .IF     "&PARM1" .EQ. ":int" saw_int
VAL4    SET     4
        .IF     "&PARM1" .EQ. ":ch" saw_ch
VAL5    SET     5
saw_char  .NOP
saw_long  .NOP
saw_int   .NOP
saw_ch    .NOP
        MEND
        DEMO1    char  ;shouldn't match ":char", but does
     DEMO1  :char   ;should match and does
     DEMO1  long    ;shouldn't match, but does
     DEMO1  :long   ; should match and does
     DEMO1  int     ; shouldn't match - and doesn't INTERESTING
     DEMO1  :int    ;should match and does
     DEMO1  ch      ;shouldn't match - and  doesn't
     DEMO1  :ch     ; should match and does
     DEMO1  nothing ; shouldn't match and doesn't
     END
```

It appears that if the original macro string includes a colon, and the passed string has more than 3 chars that match the first 3 chars after the colon, the comparison will always be true.

Temporary solution:
Pass strings that are less that 4 characters.

KPR #: 5000250274   Product: 8086/8 ASSEMB   64853   02.30

Keywords: CODE GENERATOR   PROBLEM ON 9000/S300

One-line description:
Cannot use DS for a var that is EQU'd to another var that used "SET".

Problem:
Variable that is EQU'ed to another variable that used "SET" cannot be referenced in DBS, DWS or DDS directives. It can be used in DW or in instructions such as MOV CX,#LABEL. Example:

```
"8086"
NUMSET   SET   4
NUM_EQU  EQU   NUMSET
EQU4     EQU   4
         PROG
         MOV   CX,#NUMSET    ; OK
         DATA
         DBS   EQU4          ; OK
         DBS   NUM_EQU       ; ERROR ET ; expression type
ERROR ET       ^     EXPRESSION TYPE
         DW    NUMSET        ; OK
         DB    NUMSET
         END
```

KPR #: 5000250274 **CONTINUED**

Temporary solution:

```
WORKAROUND
  Use "SET" directive in place of EQU
       NUMSET   SET   4
       NUMEQU   SET   NUMSET
                DATA
                DBS   NUMEQU
```

Signed off 08/31/88 in release A02.80

KPR #: 5000250456   Product: 8086/8 ASSEMB   64853   02.30

Keywords: CODE GENERATOR   PROBLEM ON 9000/S300

One-line description:
External EQU'ed veriables may not be resolved properly.

Problem:
EXTernal EQU'ed variables may not be resolved properly. Example:

```
"8086"
         EXT   q_ext
         EXT   foo
foo_equ  EQU   foo
Q_eq_ptr EQU   q_ext
         MOV   AX,SEG foo_equ    ; moves in SEG for q_ext
         MOV   BX,OFFSET foo_equ ; moves in OFFSET for q_ext
         MOV   AX,SEG Q_eq_ptr   ; works fine
         MOV   BX,OFFSET Q_eq_ptr ; works fine
         END    ;
```

it appears that the q_ext reference worked because it was the first EXT defined variable - swapping the position of q_ext and foo caused both references to show the SEG and OFFSET for foo instead. NOTE:this info is based on the output from the linker. foo and q_ext are defined using DD's in another file and defined as GLOBALS. If foo and q_ext are used instead of foo_equ and Q_eq_ptr then the code generated is correct.

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A02.80

KPR #: 5000260869   Product: 8086/8 ASSEMB   64853   00.00

Keywords: CODE GENERATOR   PROBLEM ON 9000/S300

One-line description:
Using 'WORD PTR' to a EQU'd constant can result in bad code

Problem:
The expression MOV AX,DS:A, when A is a EQU constant, will do a move immediate, instead of considering A a variable.
See following example

KPR #: 5000260869 **CONTINUED**

```
     ZERO  EQU  0
           MOV  AX,DS:WORD PTR 0        --> var at DS:0
           MOV  AX,DS:WORD PTR ZERO     --> constant 0
```

Temporary solution:
Use this expression:

```
ZERO   EQU DS:WORD  PTR 0
       MOV AX,ZERO
```

Signed off 08/31/88 in release A02.80

KPR #: 5000283077   Product: 8086/8 ASSEMB   64853   02.70

Keywords: PROBLEM ON 9000/S300

One-line description:
CMP statement is producing wrong label address.

Problem:
The assembler generates a bad code, which counts a wrong address.

```
          "8086"
                PROG CS:PROG,DS:DATA
                GLOBAL POWER_ON
          POWER_ON
E80100          CALL    BBB
80BC000000      CMP     DS:AAA[SI],0
90        BBB   PROC
33S0            XOR     AX,AX
C3              RET
                DATA
0000      AAA   DBS     1
                END
```

In above program, "CALL BBB" generates a wrong code "E80100". This is a mistake of counting its address. ASM 8086 rev2.3 generates PH error, but we can not find where the wrong part is.

NOTE: For the CMP statement the assembler generates 1 byte during pass 1, and 7 bytes during pass 2.

Temporary solution:
Replace "CMP DS:AAA[SI],0" with "CMP DS:BYTE PTR AAA[SI],0" will achieve the desired results however forward referencing is still not recommended.

KPR #: D200079566   Product: 8086/8 ASSEMB   64853   02.03

Keywords: LINKER

One-line description:
will not link if segment address not equal 0

Problem:
The following program causes the linker to give message:
"Address out of range" when data segment address greater than 0

KPR #: D200079566 **CONTINUED**

```
module1
"70108"
   GLB    N1
   EXTRN  N2
   PROG
N1 NOP
   RET

MODULE 1
"70108"
   GLB   N2
   EXTRN N1
N2 NOP
   RET

MODULE3
"70108_80"
   GLB   E1,D1
   EXT   E2
   PROG
E1  NOP
   LXI   H,D1
   RET
   DATA
D1  DS   10

MODULE4
"70108_80"
   GLB   E2
   EXT   E1
   EXT  D1
E2  NOP
   LXI   H,D1
   RET

LINK  N1  0F0000000H,00000100H,0
      E1  50000000H,50000100H,0
      E2  CONT,CONT,CONT
      N2  0,0,0
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A02.80

KPR #: D200079582   Product: 8086/8 ASSEMB   64853   02.03

Keywords: CODE GENERATOR

One-line description:
intra segment indirect calls

Problem:

KPR #: D200079582 **CONTINUED**

Intra segment indirect call does not generate the correct instructions:

```
"70108"
   CALL   POINTER_TABLE[IX]
   CALL   DS0:POINTER_TABLE[IX]
   CALL   OFFSET POINTER_TABLE[IX]

POINTER_TABLE
   DWS    10
```

Temporary solution:

```
USE:
CALL  DS0:WORD PTR POINTER_TABLE[IX]
ASSUMING THAT POINTER TABLE IS IN THE DATA SEGMENT
OR,
CALL   PS: WORD PTR  POINTER_TABLE[IX]
IF TABLE IS IN THE CODE SEGMENT.
```

Signed off 08/31/88 in release A02.80

KPR #: D200085035   Product: 8086/8 ASSEMB   64853   02.03

Keywords: CODE GENERATOR

One-line description:
LXI E,addr and LXI C,addr are not flagged as errors in 8080 mode

Problem:
The 8080 compatiblity modes for the V20/v30 processors to not flag expressions of the form LXI C,addr or LXI E,addr as being erroneous. The directives associated with this are "70108_80" and "70116_80".

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A02.80

KPR #: D200085308   Product: 8086/8 ASSEMB   64853   02.03

One-line description:
Address in 8086 family assemblers lost segment information.

Problem:
Due to changes to the hosted assembler, all assemblers in product 64853 and 64859 (80x86 family and 80286, B version) were truncating the segment information in addresses passed to the linker.

Signed off 08/31/88 in release A02.80

KPR #: D200088435   Product: 8086/8 ASSEMB   64853   02.70

Keywords: PROBLEM ON 9000/S300

One-line description:
INSTALLATION PROBLEM

KPR #: D200088435 **CONTINUED**

Problem:
An INSTALLATION problem thet will cause catistrophic results has been corrected.

Signed off 04/05/88 in release A02.71

KPR #: D200090340   Product: 8086/8 ASSEMB   64853   02.70

One-line description:
OUTM does not allow override prefix.

Problem:
While using the 8086 assembler with the "70116" directive, the override prefix is not allowed while using the OUTM command. All documentation seems to suggest override is permitted. The default segment register is DS0, and the user would like to use the override DS1.

NOTE:
Using the following code I could not even get the OUTM command to assemble:

```
   "70116"
           ASSUME     DS1:DATA
           OUTM       DW,DS1:[IX]
```

It does not like the second operand in the OUTM statement.

NOTE: The customer also believes a similar problem exists with MOVBK and CMPBK.

Temporary solution:
Since the OUTM command is not behaving correctly, using the OUT command is probably sufficient. Here is a possible substitute for using the DS1 override:

```
"70116"
        MOV    AW,DS1:[IX]
        OUT    DW,AW
```

Signed off 08/31/88 in release A02.80

KPR #: D200091918   Product: 8086/8 ASSEMB   64853   02.70

Keywords: PROBLEM ON 9000/S300

One-line description:
Linker locks for no apparent reason.

Problem:
The following code will assemble fine, but will cause the linker to lock. NOTE: This is an inconsistent error, sometimes it works fine sometimes it does not!
This problem is being raised to a hotsite.

KPR #: D200091918 **CONTINUED**

The code works fine on the original 64000 and 300 hosts.
The problem occurs with the "80186" option also.

```
"8086"
        NAME    "STARTUP"
        ASSUME  CS:PROG
LAB:    MOV     AX,22H
```

Signed off 08/31/88 in release A02.80

KPR #: 1650055806   Product: 8086/8 C   64818   03.07

One-line description:
ES register used but never defined.

Temporary solution:
Use $FAR_EXTVARS$ in order to load the ES segment prior to accessing the variable.

Signed off 08/31/88 in release A03.80

Duplicate Service Requests: 5000297762 5000397760

KPR #: 1650061572   Product: 8086/8 C   64818   03.70

One-line description:
Constant divided by short in function call generates wrong code.

Problem:

```
 "C"
 "8086"
 #define A 800
 extern short B
 main()
 {
    nothing(A/B);
 }

 nothing(D)
 short D;
 {
   int E;
   E = D;
 }


 generates the following (wrong) code

 rev 3.70                                rev 3.20
 MOV   AX,#0320H                         MOV  AL,3018H
 CWD                                     CBW
 MOV   AL,3018H (AL OVERWRITTEN !)       MOV CX,AX
 CBW                                     MOV AX,#0320H
 MOV CX,AX   (DIVISION RESULT IS         CWD
 IDIV  CX     ALWAYS ONE )               IDIV CX
 PUSH  AX                                PUSH AX
```

Temporary solution:
The customer is using Rev. 3.20 as a temporary fix. But, in Rev. 3.70 A/B can be placed into a temporary variable and then passing the temporary to the function.

KPR #: 5000128751 Product: 8086/8 C 64818 02.00

Keywords: CODE GENERATOR

One-line description:
Assignment operator used with mult. arrays of double/float type - bad cd

Problem:
Incorrect code is generated when using an assignment operator with multiple arrays of double or float and the right side of the equation is another type.

```
"C"
"8088"
$FIXED_PARAMETERS ON$
$FAR_EXTVARS ON$
$FAR_LIBRARIES ON$
$FAR_PROC ON$
$POINTER_SIZE 32$

struct i3 {
           int    ff[2];
        double  gg [2];
           };
static int  i,j;
ststic struct   i3  gs[2];

main()
{   j=1;
    i=0;
    gs[i].gg[j] /= j;
}
```

The problem occurs with the calculation of the field address.

```
The line:  gs[i].gg[j] /= j;   generates
         MOV    AX,#+00014H          ;AX contains 14H
         IMUL   DS:WORD PTR Dstatic  ;AX contains 14H * 0 = 0H
         MOV    BX,AX                ;BX contains 0H
         LEA    BX,DS:Dstatic[BX+8H] ;BX loaded with 8H.
                                     ;This is the address of gs[0].gg[0]
                                     ;(Dstatic +8H).  It should be the
                                     ;address of gs[0].gg[1] (Dstatic+10
                                     ;H).
```

Summary:

This problem only happens when:

1. The structure is an array
2. The field is an array of double/float
3. The referenced field is not the first element in the array of double/float (i.e. not gs[*].gg[0])
4. The two operands of the operation are of different types (i.e. gs[i].gg[j] - double, j - int)

Note: It does not matter if "gg" is the first or second field in the

KPR #: 5000128751 **CONTINUED**

structure.

Temporary solution:
Write out the assignment instead of using an assigment operator.
For example:

```
        gs[i].gg[j] = gs[i].gg[j] / j;
```

Signed off 08/31/88 in release A03.80

KPR #: 5000135285 Product: 8086/8 C 64818 03.00

One-line description:
1006 message generated when referenced to unspecified array element

Problem:
Misleading error message generated when incorrect reference to an array in a structure is made. An 1006 (contact HP error) is generated

```
"C"
"8088"
struct CN {int f; int cnt[5];} cn;
struct CM {int i; struct CN *p;} cm;
main(){ cm.i = ++cm.p->cnt; }
                             ^1006 {invalid C, but misleading error msg}
```

Temporary solution:
Use array index in stmnt:

```
      cm.i = ++cm.p->cnt[3];
```

KPR #: 5000146779 Product: 8086/8 C 64818 03.00

Keywords: CODE GENERATOR

One-line description:
Divide operation byte divisor & quotient › a byte which causes 0 Int.

Problem:
The following program causes the processor to generate an interrupt 0.

```
"C"
"8088"
main(){ unsigned        n;
        unsigned short  j;
            n = 25178;
              MOV        SS:WORD PTR [BP-05H],#+0625AH
            j = n % 10;
              MOV   CL,#+0AH     (10 loaded into byte -divisor)
              MOV   AX,SS:WORD PTR [BP-05H]  (loads n)
              DIV   CL (dividend in AX, result in AL, remainder in AH
                          quotient ‹ FFH  or a INT 0 is executed)
              MOV   SS:BYTE PTR [BP-03H],AH
      }
```

The extensions listed in the 8086 C manual on pg. 1-3 states that arithmetic can be performed with short's without converting them

KPR #: 5000146779 **CONTINUED**

int's. Also, the processor states on pg 3-69 that if the quotient exceeds FFH for a byte source (10), an INT 0 will be generated. The compiler should have generated the DIV instruction with a word source by loading 10 into CX instead of CL.

Temporary solution:
A temporary solution would be to type cast the divisor.

```
j = n % (unsigned) 10;
```

This forces the 10 to be loaded into CX (a word) for the DIV operation. This means that the quotient must be <= FFFFH.

KPR #: 5000176891 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Bad code generated when casting a real constant into an integer

Problem:
Bad code generated when casting a real constant into an integer.
Example:

```
"C"
"processor"
main() {
        int   i;
        i = (int)0.0; /* generates code to move some value other
                         than 0 into i  ----- BUG ------  */
}
```

Temporary solution:
There is no known work around at this time.

KPR #: 5000191361 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
When $POINTER_SIZE 32$ generates 32 bit arithmetic for 16 bit variables

Problem:
32 bit arithmetic used when 16 bit called for. Example:

```
"C"
"80186"
$POINTER_SIZE 32$
main() {
        int    *p;
        unsigned int  i,j;
        if ( p && (( i - j ) > j )
                               ;
}
```

The left and right sides of the && are both evaluated using 32 bit arithmetic.

KPR #: 5000191361 **CONTINUED**

Temporary solution:
WORKAROUND

```
    use:
      if ( p ==0 && (( i - j ) > j )    /*which causes the right
    hand side to be evaluated using 16 bit arithmetic  */
```

KPR #: 5000214858 Product: 8086/8 C 64818 03.01

Keywords: CODE GENERATOR

One-line description:
Bad code generated when left shift short variable & AND w/ unsigned int

Problem:
Bad source code generated with left shift:

```
"C"
"8086"
$LIST_OBJ$
unsigned short i;unsigned int j;
main()
{
if (j & (1<<i));
       MOV CL,DS:BYTE PTR Dstatic+1H
       MOV AL,#1H
       CBW
       SHL AX,CL   <----- This is incorrect.  The correct code is:  |
D2E0 <---------  SHL AL,CL  <---------------|
       ......
}
```

Temporary solution:
There is no known fix at this time.

KPR #: 5000228023 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Error 1113 generated in PASS 3, when using "case 0xffff" in switch stmt

Problem:
Compiler generates 1113 error in PASS 3 of switch statement. This only happens when the "case 0xffff" is the last statement in the switch

The example:

```
 "C"
 "8086"
 main()
 {
    unsigned I;
              switch( I )
              {
               case 0xfffd:
```

KPR #: 5000228023 **CONTINUED**

```
                case 0x0000:
                case 0x0001:
                case 0xffff:
                default :;
               }
}
```

Temporary solution:
Do not make the case statement the last one.

Signed off 08/31/88 in release A03.80

KPR #: 5000229245   Product: 8086/8 C   64818   03.02

Keywords: CODE GENERATOR

One-line description:
Compiler generates MOV SP, BP and LEAVE. This is redundant.

Problem:
The 8086 C Cross Compiler generates redundant code when compiled with the 80286 option. When a Function is exited the compiler generates the the following instructions. MOV SP, BP and LEAVE. MOV SP, BP is implicit to the LEAVE instruction. This minor problem only appears in the 64100A version of the Compiler. This redundancy does not appear when compiled on the HP-UX Version of the Cross Compiler.

Temporary solution:
There is no know work around at this time.

KPR #: 5000247536   Product: 8086/8 C   64818   03.20

Keywords: CODE GENERATOR

One-line description:
Err 1006 generated if passing address of array into array of pointers.

Problem:
The following C program generates a pass 2 compilation error number 1006: Compiler error, contact Hewlett-Packard

```
"C"
"70108"
$POINTER_SIZE=32$

struct ARGUMENT {
   int argc;
   char *argv[];
};

test(com,arg,j)
char com[];
stuct ARGUMENT *arg;
int j;
{
```

KPR #: 5000247536 **CONTINUED**

```
   arg->argv[j] = &com[0];
}
```

Temporary solution:
Workaround:

```
"C"
"70108"
struct ARGUMENT {
   int argc;
   char *argv[];
}

test(com,arg,j)
char com[];
struct ARGUMENT *arg;
int j;
{
   arg->argv[j] = com;
}
```

Signed off 08/31/88 in release A03.80

KPR #: 5000278127   Product: 8086/8 C   64818   03.70

One-line description:
Bad code generated on 64000 with "80286" directive.

Problem:
Bad Code generated when "80286" directive is used with C compiler on 64000. The following function generates bad code:

```
"C"
"80286"
$POINTER_SIZE 32$
int  slength(s)
char  *s;
{     int   i;
      while(*s++)
            i++;
}
```

Temporary storage for the compiler overwrites values pushed on the stack as parameters to an Add32 library call.

NOTE: The ENTER command is not used on the 64000, but the offsets look as if it had been. Also, the LEAVE command is used without the ENTER command.

Temporary solution:
No known fix at this time.

KPR #: 5000291930 Product: 8086/8 C 64818 03.70

One-line description:
Use of structure causing fatal 1006 error.

Problem:

```
"C"
"80186"
$POINTER_SIZE = 32$

struct  EXAMP_MA
{
  int measmode;
  char  accssort;
};

struct DT_MEAS
{
  int mode;
  char accssort;
};

static char remdtst (fdata,measdata)
struct EXAMP_MA fdata[];
struct DT_MEAS measdata[];
{
  int mcount;
  int dtcount;

  measdata[mcount].accssort = fdata[dtcount].accssort; <---
};                                                         |
                                                           |
                              Causes Fatal 1006 error  -------
```

It seems error 1006 (contact Hewlett-Packard) is a fatal pointer error. The lab tested the problem on the latest revision of the C compiler in the lab and the problem still occurred.

The problem does not seem to occur when POINTER_SIZE = 16, or if the order of the elements in struct EXAMP_MA are reversed.

Temporary solution:
Instead of using

```
  measdata[mcount].accssort = fdata[dtcount].accssort;
```

try breaking it up into:

```
char  temp;
temp = fdata[dtcount].accssort;
measdata[mcount].accssort = temp;
```

Signed off 08/31/88 in release A03.80

KPR #: 5000294199 Product: 8086/8 C 64818 03.70

Keywords: PROBLEM ON 9000/S300

One-line description:
& address operator generates PUSH DS1 when DS1 not defined.

Problem:
The compiler uses a wrong register(DS1),which is not defined before using it.

```
"C"
"70116"
$EXTENSION  ON$ $WARN OFF$ $SEPARATE CONST OFF$ $POINTER SIZE 32$
$RECURSIVE OFF$ $OPTIMIZE ON$ $INIT ZEROS OFF$ $FAR LIBRALIES ON$
$FAR PROC ON$
extern unsigned int wtest,wtest1;
test()
{
       unsigned int *lptr;
       lptr=&wtest+wtest1;  -----> This generates PUSH DS1
                                   DS1 is not defined any place.
}
```

We can not use Rev 3.7,we decide to use Rev 3.2 again.

KPR #: D200008342 Product: 8086/8 C 64818 00.56

Keywords: CODE GENERATOR

One-line description:
Vars ORGed in seg. 0 in SHORT env. access current DS seg with no warning

Problem:
In the SHORT environment (16-bit pointers) all variables ORGed in segment zero (0) will be accessed as an offset from DS. If DS <> 0 the actual location will not have an address equal to the value selected in the ORG statement in the source. This feature of the compiler permits the address of variables ORGed in segment 0 to be taken in the short environment. However, if the system under design has an I/O port in segment 0, but DS <> 0, there will be problems the cause of which may not be readily apparent. A warning message should be given whenever a variable is ORGed to segment 0, stating that that variable will be located in the current DS segment.

Temporary solution:
If the address of an ORGed variable does not have to be taken (assigned to a pointer, or variable an array, etc.) and the address value of the variable >= 10H, then the variable can be ORGed in a segment other than 0 and the addressed accessed will reflect the value in the ORG statement

KPR #: D200027995 Product: 8086/8 C 64818 02.00

One-line description:
One's complement operator ~ causes incorrect code when used in if stmt.

Problem:

KPR #: D200027995 **CONTINUED**

When the one's complement operator, ~ , is used in an if statement, a NOT instruction is generated to perform the one's complement. Then a jump instruction is executed which is dependent on certain flags being set. The NOT instruction does not set any flags, so the jump never occurs. The following code is an example of this:

```
"C"
"processor name"
extern test;
main() {
  unsigned int x;
   if (~x)
     MOV     AX,SS:WORD PTR [BP-00002H]
     NOT     AX
     JNE     SHORT $+00005H
     JMP     NEAR PTR main01_0
    test;
}
```

Temporary solution:
If only negation is required, use the logical negation operator, !. If one's complement is required, then turn $ASM_FILE ON$ and edit the resulting assembly file by changing the NOT AX instruction to XOR AX,0FFFFH. This will set the flags so that the JNE may be executed.

Signed off 08/31/88 in release A03.80

KPR #: D200038836 Product: 8086/8 C 64818 02.00

One-line description:
Compiler using unacceptable amount of stack space for procedure returns.

Problem:
When procedures are called that return values greater than 16 bits long (32 bit pointers), temporary variables are used to store the result. These temporary variables are created on the stack and are used only for temporary result storage. The current version of the 8086 C compiler creates a new temporary every time it needs one, even though it will only be used once. The previous version (1.05) generated only as many temporaries as it required and reused them throughout the procedure.

Temporary solution:
No known temporary solution at this time.

KPR #: D200049908 Product: 8086/8 C 64818 03.00

One-line description:
Illegal instruction generated by ASM_FILE

Problem:
An illegal assembly instruction is generated for an assignment statement. The ASM_FILE contains a move of a byte register to a word register.

KPR #: D200049908 **CONTINUED**

```
struct {
        char cnt;
        char chain;
        char param[62];
      } parmabuf[20];
stuct  tache{
            struct{
                  struct{
                        genre        :1;
                                     :1;
                        fonction     :4;
                        classe       :2;
                        destination  :3;
                        origine      :3;
                        type         :2;
                        } entete;
                  char sapi;
                  char ces_tei;
                  union{
                        char shparam[2];
                        int  numbuf;
                       } prim[6];
           struct{
                            :8;
                 prm        :6;
                            :1;
                 libcount   :1;
                } event;
           char pr,pv;
          } tache2;
main()
{
int i,k;
tache2.prim[k].sapi = parambuf[j].param[2];
.
.
     MOV  AX,CL     ; illegal assmbly instruction
.
.
}
```

Temporary solution:
No known temporary solution.

KPR #: D200063057 Product: 8086/8 C 64818 03.01

One-line description:
Conditional expressions with unsigned mixed operands may fail

Problem:
Conditional expressions with unsigned mixed operands may fail.

Temporary solution:
There is no known work around at this time.

KPR #: D200068080 Product: 8086/8 C 64818 03.01

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"

struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200068700 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Casting ptr. to int as short & incrementing it generates bad code

Problem:

The following table describes the nature of each compiled file and on the 64100.

			number of	increment	
			increments;	and gets	
test	"if"	Ptr	statement	separate	
case	used	size	separation	statements	BUG DESRCIPTION
TEST1	yes	32	2 ;	no	Reboots system
TEST2	no	32	2 ;	no	No increment in listing
TEST3	yes	32	2 ,	no	No increment in listing
TEST4	yes	16	2 ;	no	Reboots system
TEST5	no	16	2 ;	no	compiles correct
TEST6	yes	16	2 ,	no	Reboots system
TEST7	yes	32	1	no	No increments in listing
TEST8	yes	16	1	no	Reboots system
TEST9	no	32	1	yes	Error in factor message

KPR #: D200068700 **CONTINUED**

TEST10	no	16	1	yes	Error in factor message
TEST11	no	32	1	yes	No increment in listing
TEST12	no	16	1	yes	No increment in listing

The following table describes the nature of each compiled file and on the 64100.

			number of	increment	
			increments;	and gets	
test	"if"	Ptr	statement	separate	
case	used	size	separation	statements	BUG DESRCIPTION
TEST1	yes	32	2 ;	no	Reboots system
TEST2	no	32	2 ;	no	No increment in listing
TEST3	yes	32	2 ,	no	No increment in listing
TEST4	yes	16	2 ;	no	Reboots system
TEST5	no	16	2 ;	no	compiles correct
TEST6	yes	16	2 ,	no	Reboots system
TEST7	yes	32	1	no	No increments in listing
TEST8	yes	16	1	no	Reboots system
TEST9	no	32	1	yes	Error in factor message
TEST10	no	16	1	yes	Error in factor message
TEST11	no	32	1	yes	No increment in listing
TEST12	no	16	1	yes	No increment in listing

Temporary solution:
There is no known fix at this time.

KPR #: D200069716 Product: 8086/8 C 64818 03.01

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
```

KPR #: D200069716 **CONTINUED**

```
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function.
For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200071787 Product: 8086/8 C 64818 03.01

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory, remove the DATA pseudo, and reassemble.

KPR #: D200076695 Product: 8086/8 C 64818 03.02

One-line description:
Fields of a structure are dereferenced incorrectly (if fields are big).

Problem:
Structure ponters are not being calculated correctly when relative addressing requires offsets of large sizes. See following code.

```
"C"
"68000"
```

KPR #: D200076695 **CONTINUED**

```
$FAR$

struct this{
         unsigned short int first[256][256];
         unsigned short int second[256][256];
} one,*bufptr;

unsigned short int *desptr;

main()
{
 bufptr = &one;
 destptr = bufptr->first[0][0];
 destptr = bufptr->second[0][0];   /* Same address assigned. */
 }
```

NOTE: The 8086 line of compilers will generate a pass three error for this code. "Program Counters Disagree. "

Temporary solution:
For the 68000 family of cross compilers you may use the '.' operator instead.

```
"C"
"680XX"

$FAR$

struct  this {
        unsigned short int  first [256][256];
        unsigned short int  second[256][256];
} Structure,*ptrToStruct;

unsigned short int *destptr;
main()
{

 destptr = &Structure.second[0][0];
}
```

KPR #: D200079343 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
~, &, | and ^ may not correctly expand shorts in condidionals

Problem:
The complement operator and the bitwise AND, OR, and XOR operators do not correctly extend signed 8-bit quantities. For example, in the following code segment for Z80 "s" is extended as if it were an unsigned quantity before the OR operation is performed. NOTE: no extend is

KPR #: D200079343 **CONTINUED**

done, and an 8-bit OR performed if $SHORT_ARITH ON$ is in effect.

```
short s;

main()
{
   if (s&0x01)
       LDA s
       MVI H,000H
       MOV L,A
       LXI D,00001H
       CALL Zwor
           .
           .
           .
}
```

KPR #: D200079608 Product: 8086/8 C 64818 01.10

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
```

KPR #: D200079608 **CONTINUED**

```
          }
         addr =addr+1;
       }
  }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081513 Product: 8086/8 C 64818 03.02

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
          ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;

  OR
```

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200082628 Product: 8086/8 C 64818 03.02

Keywords: CODE GENERATOR

One-line description:
Case stment nested in With stment w/ variant records generates bad code

Problem:
The following program generates bad code in the case statement:

```
"PASCAL"
"80186"
$FAR_PROC       +$
$FAR_LIBRARIES  +$
$POINTER_SIZE  32$
$EXTENSIONS     +$
$SEPARATE_CONST -$
$RECURSIVE      +$

PROGRAM  IWG_01P;

CONST
  ENQ    =      05H;
  MAK    =        2;
  ANZ_MWERTE =   20;

TYPE
  INTEGER     = SIGNED_16;
  IWK_BUFFER  =  RECORD
       KENNUNG     : CHAR;
       KENNZAHL    : UNSIGNED_8;
       CASE INTEGER OF
           1       : (STARTKRIT    :  UNSIGNED_16;
                      DELAY        :  UNSIGNED_32;
                      SAMPLE_FAC   :  UNSIGNED_16;
                      ANZ_KANAELE  :  UNSIGNED_8;
                      KANAL_NR     :  ARRAY[1..MAK] OF UNSIGNED_8);
           2       : (BLOCK_NR     :  UNSIGNED_8;
                       MW          :  ARRAY[1..ANZ_MWERTE] OF INTEGER);
           3       : (INTEGR       :  SIGNED_32;
                       KURVEN_MAX  :  INTEGER);
           4       : (FEHLER       :  UNSIGNED_8);
           END;

IWKB_POINTER        =    ^IWK_BUFFER;

$GLOBVAR          +$
VAR
  IWKB_P   :  IWKB_POINTER;
$GLOBVAR          -$

FUNCTION KANALNR(ZEILE:  INTEGER);                 EXTERNAL;
FUNCTION SET_VALUE(ZEILE: INTEGER): UNSIGNED_32;   EXTERNAL;
FUNCTION SOLLWERTNR(ZEILE: INTEGER): INTEGER;      EXTERNAL;
```

```
PROCEDURE IWKA_INIT(Z1,Z2,Z3,Z4,Z5: INTEGER; K: UNSIGNED_8; FAC:
                    UNSIGNED_16);

VAR
  I    : INTEGER;

BEGIN
   WITH IWKB_P^ DO
   BEGIN
          KENNUNG              := CHAR(ENQ);
          KENNZAHL             := UNSIGNED_8(1);
          STARTKRIT            := UNSIGNED_16(SOLLWERTNR(Z1));
          DELAY                := SET_VALUE(Z2)*UNSIGNED_32(100);
          SAMPLE_FAC           := FAC;
          ANZ_KANAELE          := K;
          FOR I := 0 TO INTEGER(K) DO
           CASE I OF
                1 : IF (Z3>0) THEN
                        KANAL_NR[I] := UNSIGNED_8(KANALNR(Z3));
{This code, in fact all of the CASE statements produce the following
bad code:
                  .
                  .
                  ADD  SI,SS:WORD PTR [BP-00004H]
                  MOV  ES,SS:WORD PTR [BP-2H]
                  .
                  .
{BP-2H contains the display, not the Segment for KANAL_NR[I] }
                2 : IF (Z4>0) THEN
                        KANAL_NR[I] := UNSIGNED_8(KANALNR(Z4));
                3 : IF (Z5>0) THEN
                        KANAL_NR[I] := UNSIGNED_8(KANALNR(Z5));
                OTHERWISE;
          END;
     END;
END;
.
```

Temporary solution:
The problem is in the function calls. The work around is to make the function call in a spearate line, into a temporary variable, and then put the temporary variable in the equation.

Signed off 08/31/88 in release A03.80

KPR #: D200085738 Product: 8086/8 C 64818 03.70

Keywords: CODE GENERATOR

One-line description:
PASS 2 error when pntr type used to invoke code stored in array.

Problem:
When the typedef statement is used to define pointers to functions,

KPR #: D200085738 **CONTINUED**

and this pointer type is used in a cast of a variable array to invoke code stored in that array, program execution may transfer to the wrong location. For example, in the following code the simple call to code_array fails while the call and assignment to p works correctly:

```
     typedef int(*PFI)();  /* PFI a pointer to int functions */
     int code_array[100];  /* array contains code */
     PFI p;                /* p a pointer of type PFI */

     pfibug()
     {
       (*((PFI) code_array))();  /* fails in JSR to code_array */
       (*(p=(PFI)code_array))(); /* assignment and JSR successful */
     }
```

Temporary solution:
Set up a dummy variable and perform an assignment to it when doing this type of operation.

KPR #: D200086942 Product: 8086/8 C 64818 03.70

Keywords: CODE GENERATOR

One-line description:
ADDR routine causes "Access to guarded mem" msg, due to prob w/ FINDMARK

Problem:
Due to a problem with FINDMARK routine, it is possible for the ADDR call to result in an "Access to guarded memory" failure message in emulation.

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.80

KPR #: D200090332 Product: 8086/8 C 64818 03.70

One-line description:
IF statement loads wrong segment for compare statement.

Problem:
Code generated by the IF statement in the following program loads the wrong segment at the CMP level. It seems it is the extra segment being overwritten.

```
"C"
"70108"
$SEPARATE_CONST OFF$
$FAR_LIBRARIES ON$
$RECURSIVE ON$
$AMNESIA ON$
$FAR_PROC ON$
$POINTER_SIZE 32$
$FAR_EXTVARS ON$
```

KPR #: D200090332 **CONTINUED**

```
struct S1
{
  char *p1;
  int entier;
  char c[10];
} s1;

struct S1 *p0;

main()
{
  toto(p0);
}

toto(pointeur)
struct S1 *pointeur;
{
  if (*pointeur->p1 == pointeur->c[pointeur->entier])  {};
}
```

Temporary solution:
A temporary fix may be to break-up the statements being compared:

```
   toto(pointeur)
   struct S1 *pointeur;
   {
      char  temp1,temp2;
      temp1 = *pointeur->p1;
      temp2 = pointeur->c[pointeur->entier];

      if (temp1 == temp2) {}
   }
```

KPR #: 5000110098 Product: 8086/8 PASCAL 64814 01.10

Keywords: CODE GENERATOR

One-line description:
Incorrect code generated when CASE stmnt. uses an arrayed record field.

Problem:
Following pascal program generates bad code:

```
$EXTENSIONS ON$
TYPE INTEGER=SIGNED_16;
     contour=RECORD  length:INTEGER; dirar:ARRAY[0..20] OF INTEGER;
PROCEDURE peri(VAR c:contour);
VAR  i, dir : INTEGER;
BEGIN
   CASE c.dirar[i] OF  { Generated wrong code }
   END;

   dir := c.dirar[i]   { Workaround }
   CASE dir OF
   END;
END;
.
```

The wrong code is:

```
    SHL  BX
    ADD  BX,BX   for calculating the base address of the array in the
                 record.
```

The workaround using the following to calculate the address.

```
          SHL    AX
          ADD    BX,AX
```

Temporary solution:
Use IF statement instead of CASE.

Signed off 08/31/88 in release A03.60

KPR #: 5000138941 Product: 8086/8 PASCAL 64814 03.00

One-line description:
Out of expression storage error generated on code that ran on old ver.

Problem:
Customer has some code that originally compiled on both the 9000 and the 64100. With the latest released version (3.00) he can no longer compile the same file on the 64100. When he tries to compile, he gets a 406:Out of expression tree storage;simply expression.

For example:

```
  VAR X : ARRAY [0..50] OF BOOLEAN;
      A ; BOOLEAN;
  BEGIN
    A := (X[0] OR X[1] OR X[2] ...X[18]);
END;
```

Temporary solution:
Break up expression into simplier pieces.

KPR #: 5000146829 Product: 8086/8 PASCAL 64814 03.00

One-line description:
Libraries load constants into the data area

Problem:
Some of the library routines contain constants which reside in DATA space. This prevents these libraries from being used in a ROM based system.

For example:

The file SINCOSC:NS8086 is an assembly file containing constants that are used by the routine SINCOS:NS8086. The "DATA" pseudo opcode is used and all constants reside in DATA area!!!

There are a few variables (i.e. monitor_message) which need to be in the DATA area, but the majority of the constants are also being loaded in the DATA area. Since the libraries are shipped in relocatable form only, the customer must wait for the factory to send the sources to him just so he can take out the DATA pseudo and reassemble.

Please place all constants in the PROG area.

Temporary solution:
The only work around is to obtain the sources from the factory, remove the DATA pseudo, and reassemble.

KPR #: 5000170175 Product: 8086/8 PASCAL 64814 03.01

Keywords: CODE GENERATOR

One-line description:
DIV of array of signed_16 by signed_16 in FOR loop produces bad code

Problem:
Incorrect code generated for following situation:

1. Division of signed_16 by signed_16 using the Pascal DIV command.
2. The dividend must be an array of signed_16.
3. Division is taking place within a loop where the subscript of the array is the looping variable. (ie. N is the subscript and looping variable.)
4. The error occurs with the first instance of the Pascal statement referencing array[subscript] DIV var. Any statements with the same format following the first in the loop will create good code.

```
"70108"
$EXTENSIONS ON$
PROGRAM HERE;
VAR
  X     :SIGNED_16;
  V       :ARRAY[1 .. 3] OF SIGNED_16;
  I,N     :SIGNED_8;
BEGIN
  FOR N := 1 TO 3 DO
  BEGIN
```

KPR #: 5000170175 **CONTINUED**

```
     I := V[N] DIV X;  => Generates incorrect code.
     I := V[N] DIV X;  => Generates correct code.
   END
END.
```

Temporary solution:
Create a variable for array element. Break the equation into two pieces:

```
"70108"
$EXTENSIONS ON$
PROGRAM HERE;
VAR
  W,X    :SIGNED_16;
  V      :ARRAY[1 .. 3] OF SIGNED_16;
  I,N    :SIGNED_8;
BEGIN
  FOR N := 1 TO 3 DO
  BEGIN
     W := V[N];         => Generates correct code and is safest work arou
     I := W DIV X          nd.
   END
END.
```

Signed off 08/31/88 in release A03.60

KPR #: 5000232744   Product: 8086/8 PASCAL   64814   03.02

Keywords: CODE GENERATOR

One-line description:
Array reference overrides DX register

Problem:
The following program overrides DX reqister when using the IMUL instruction:

```
"8086"
PROGRAM TEST;
TYPE
   REC=RECORD
     R1,R2,R3:SIGNED_16;
   END;
VAR SOURCE  : ARRAY [ 1..20] OF REC;
    V1      : SIGNED_16;
BEGIN
   FOR V1 := 1 TO 19 DO
       SOURCE[V1].R1  := SOURCE[V1+1].R1;
       .
       .
       .
       MOV  DX,DS:WORD PTR [BX]   ; DX = DS + SOURCE[V1].R1
       MOV  AX,#+6H
       IMUL DS:WORD PTR DTEST+78H  ; DS = V1 * 6 This will put V1*6
       .                             in DX:AX and re-use the DX regis-
```

KPR #: 5000232744 **CONTINUED**

```
       .                             ter that was holding the right side
END.                                 of the equation.
```

Temporary solution:
Add a temporary variable to hold SOURCE[V1+1].R

```
"8086"
PROGRAM TEST;
TYPE
   REC=RECORD
     R1,R2,R3:SIGNED_16;
   END;
VAR SOURCE  : ARRAY [ 1..20] OF REC;
    V1      : SIGNED_16;
    TEMP    : SIGNED_16;
BEGIN
   FOR V1 := 1 TO 19 DO
    BEGIN
       TEMP := SOURCE[V1+1].R1;
       SOURCE[V1].R1  := TEMP;
END.
```

Signed off 08/31/88 in release A03.60

KPR #: 5000244392   Product: 8086/8 PASCAL   64814   03.02

Keywords: CODE GENERATOR

One-line description:
ERROR 117 generated, but does not indicate variable in error

Problem:
Pascal Compiler error 117 "Unsatisfied Forward Reference" does not provide information as to which reference is unsatisified when the reference is a pointer to an undefined type. The following program gives an example:

```
         "8086"
         PROGRAM ERR117;
         TYPE
             UNSAT = ^UNSAT_TYPE;
         BEGIN
         END.
** ERROR??  ^117
117: Unsatisfied forward reference
```

There is no indication as to which reference is unsatisfied.

Temporary solution:
There is no known work around at this time.

```
KPR #: 5000246157   Product: 8086/8 PASCAL        64814             03.50

Keywords: CODE GENERATOR       PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
          PROBLEM ON VAX       NOT ON 64100 SYSTEM
```

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example:

```
"8086"
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
LABEL  100;
BEGIN
100:
     GOTO   100;
END;
PROCEDURE  TWO;     { pass 3 error - too many errors in pass 3 }
LABEL  100;         { is generated, without any indication as to }
BEGIN               { what the problem is                        }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get this error message, be aware that you may have duplicate labels in the program.

Signed off 08/31/88 in release A03.60

Duplicate Service Requests: 5000264499

```
KPR #: 5000259176   Product: 8086/8 PASCAL        64814             03.20

Keywords: CODE GENERATOR
```

One-line description:
MOD operative in complex equation generates bade code.

Problem:
Bad code generated when MOD function is used and result is stored in array. Example:

```
"8086" PREPROCESS
$EXTENSIONS ON$
$FAR_EXTVARS$
$POINTER_SIZE 32$
$RECURSIVE ON$
PROGRAM DISPLY;
TYPE   LINE = ARRAY[1..80] OF CHAR
VAR
$EXTVAR ON$
    DISPLAY_Q : ARRAY[1..52] OF LINE;
```

KPR #: 5000259176 **CONTINUED**

```
    DISP_LINE : SIGNED_16;
$EXTVAR OFF$
IS : SIGNED_16 ;
PROCEDURE  D_DIU_STATUS ;
  DISPLAY_Q[DISP_LINE,43] := CHR((IS MOD 10)+48);   END;.
```

The DX register is used to hold the result of the
CHR((IS MOD 10) + 48);
but then is later overwritten when the compiler uses an IMUL instruction to calculated the index into the array.

Temporary solution:
WORKAROUND:
A valid workaround is to use a temp var to hold the result and then assign the temp to the array.

Signed off 08/31/88 in release A03.60

```
KPR #: 5000259598   Product: 8086/8 PASCAL        64814             03.50

Keywords: CODE GENERATOR
```

One-line description:
Err 1006 generated in compex equation w/ in 2 FOR stmts in an IF statmnt

Problem:
The following program will cause an error:

```
"8086" PREPROCESS
$EXTENSIONS ON$
$FAR_EXTVARS$
$FAR_PROC$
$FAR_LIBARARIES$
$POINTER_SIZE=32$
$SEPARATE_CONST OFF$

PROGRAM DISPUTILS;

  TYPE
   DISPLINE  = ^D_LINE;
   D_LINE = RECORD
          LENGTH : UNSIGNED_8;
          DATA   : ARRAY [1..50] OF CHAR;
          END;
  VAR
  $GLOBVAR ON$
    CLEAR_INDEX    :  ARRAY [1..15] OF SIGNED_16;
  $GLOBVAR OFF$
  $EXTVAR ON$
    MENU_MODE      :  BOOLEAN;
  $EXTVAR OFF$

  $EXTVAR ON$
   STATE          : ARRAY [1..18] OF DISPLINE;
```

KPR #: 5000259598 **CONTINUED**

```
   DISP_TBL         : ARRAY [1..70] OF DISPLINE;
  $EXTVAR OFF$

  $EXTVAR ON$
    DISPLAY_Q_FLAG  :  BOOLEAN;
    AUDIT_Q         :  ARRAY [0..699] OF CHAR;
    NUM_TO_CLEAR    :  SIGNED_16;
  $EXTVAR OFF$

  $GLOBPROC ON$
  PROCEDURE CLEAR_WHAT;
  $GLOBPROC OFF$
   VAR
    I,J,K,AUDIT_I : SIGNED_16;

   BEGIN
    IF (NUM_TO_CLEAR <> 0) AND (MENU_MODE = TRUE)
    THEN BEGIN
       FOR I := 0 TO 4 DO AUDIT_Q[I] := STATE[1]^.DATA[I+1];
       BEGIN
         FOR J := 1 TO SIGNED_16(DISP_TBL[K]^.LENGTH) DO
              AUDIT_Q[AUDIT_I + J] := DISP_TBL[K]^.DATA[J];
       END;
    END;
END;
.
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: 5000272021   Product: 8086/8 PASCAL   64814   03.02

One-line description:
CASE statement produces bad code for complicated expression

Temporary solution:
A temporary fix for the problem is to place the complicated expression used in the CASE statement into a temporary variable. Then, use this temporary variable in the CASE statement instead of the complicated expression.

KPR #: D200006080   Product: 8086/8 PASCAL   64814   00.46

Keywords: CODE GENERATOR

One-line description:
Data structures larger than 64K are not flagged as an error.

Problem:
Variables (data structures) which require more than 64K of contiguous memory are not flagged as an error. Please refer to the example shown below.

KPR #: D200006080 **CONTINUED**

```
   $POINTER_SIZE 32$
   VAR
      RAM : ARRAY [0..3FFFFH] OF BYTE;
      I : INTEGER;

   BEGIN
      FOR I := 0 TO 3FFFFH DO
         RAM[I] := 0FFH
   END.
```

In the above example, the first 64K bytes (one segment) of memory are assigned three times.

Temporary solution:
No known workaround at this time.

KPR #: D200010280   Product: 8086/8 PASCAL   64814   00.60

Keywords: CODE GENERATOR

One-line description:
Byte values may be converted to 16-bit before comparison with byte var.

Problem:
Byte variables may be converted to 16-bit before being compared with another byte variable (constant). Please refer to the following example for further details.

```
PROGRAM TEST;
$EXTENSIONS ON$
VAR A,B : SIGNED_8;

BEGIN               { contents of A:  0H  |          0FFH              }
  CASE A OF         { A->AL, CBW   :  0H  |         0FFFFH             }
    0FFH : B:=0;    {                     | 000FF<>0FFFFH : Case error }
    0    : B:=1     {                 ok  |                            }
  END
END.
```

In the example above, 0FFH is converted to a 16-bit value before being compared to the byte variable A. When 0FFH is converted to a 16-bit quantity, the resulting value is 0FFFFH, which will never equal the value of the byte variable A. If the value that is to be compared to A is less than 080H, then conversion to a 16-bit value is not performed.

Temporary solution:
No known workaround at this time.

KPR #: D200014944   Product: 8086/8 PASCAL   64814   01.10

Keywords: RUN-TIME LIBRARY

One-line description:
Failed to detect out-of-bounds case.

KPR #: D200014944 **CONTINUED**

Problem:
A run time out-of-bounds error is not indicated for the following program.

```
  PROGRAM TEST;

     CONST
        CON_1 = 32768;

     PROCEDURE PASS_1 (CON_1 : SIGNED_16);

        VAR
           LOCAL_1 : SIGNED_16;

        BEGIN
           LOCAL_1 := CON_1;
        END;

     BEGIN
        PASS_1 (CON_1);
     END.
```

Temporary solution:
No known workaround at this time.

KPR #: D200027516 Product: 8086/8 PASCAL 64814 03.02

One-line description:
Nested IFs inside a WITH may generate incorrect code.

Problem:
Incorrect code may be generated by the 8086 pascal compilers in the UNIX environment when nested IF's are used inside a WITH statement.

Temporary solution:
Use simpler, less compilex nesting.

KPR #: D200047779 Product: 8086/8 PASCAL 64814 02.01

One-line description:
80186Generates wrong offset within CONST_data area

Problem:
The statement:

```
    ELA    BX,DS:CONST_data+000183H
```

causes a wrong condition as the CONST_data area goes up to approximately 100H offset.

Temporary solution:
There is no know work around at this time.

KPR #: D200060061 Product: 8086/8 PASCAL 64814 03.00

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200085019 Product: 8086/8 PASCAL 64814 03.02

Keywords: CODE GENERATOR

One-line description:
Complex data structure produces bad code

Problem:
The following program produces bad code. ES register is PUSHED too late:

```
"C"
"80186"

$POINTER_SIZE 32$
$FAR_EXTVARS$

struct update_msg{char mode_id; short neigh_devid[16];};
```

KPR #: D200085019 **CONTINUED**

```
struct TDB { short age; short link_count; short neighbors[16];
             short attribute[16]; short cons_DSO[16];
             short neigh_devid[16]; unsigned int seq_no; };
extern struct TDB tdb[100];
extern node_id;

main()
{
int temp_devid;
struct update_msg *buffer;

buffer -> neigh_devid[temp_devid] = tdp[node_id].neigh_devid[temp_devid]

   LES   BX, SS:DWORD PTR [BP-00006H]
   ADD   BX, #+00001H        /*BX = buffer->neigh_devid
   MOV   AX, SEG node_id
   MOV   ES, AX  /*ES = SEG node_id
   IMUL  AX, ES:WORD PTR node_id,#+00044H /*AX=tdb[node_id]
   MOV   SI,AX   /* SI = tdb[node_id]
   LEA   SI,DS:tdb[SI+00032H]  /*ES:SI = tdb[node_id].neigh_devid
   MOV   AX,SEG  tdb
   MOV   ES,AX   /* loose ES
   PUSH  ES         /* it is too late to PUSH ES, it is already lost
   ADD   SI,SS:WORD PTR [BP-00008H] /* BX=tdb[node_id].neigh_devid[temp_
                                              devid]
   .
   .
   .
   .
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200085712 Product: 8086/8 PASCAL 64814 02.01

Keywords: CODE GENERATOR

One-line description:
WITH statement generating bad code

Problem:
The following program generates incorrect code:

```
   "processor name"
   $EXTENSIONS ON$
   PROGRAM TEST;
   TYPE NUM_REC =RECORD  NUM_BUF : ARRAY [1..24] OF BYTE;
                         TOT_NUM : BYTE; END;
        PTR = ^INTEGER;
   VAR  KEY:BYTE; NUM_INP : NUM_REC; POINTER: PTR;

   PROCEDURE DISPLAY(ROW,COLUMN,LENGTH : BYTE; START:PTR;); EXTERNAL;

   PROCEDURE IN;
```

KPR #: D200085712 **CONTINUED**

```
   BEGIN
     WITH NUM_INP DO BEGIN
       NUM_BUF[TOT_NUM] := KEY;
                :
           ADD  BX,AX        {BX WILL HOLD ADDR OF TOT_NUM}
                :
       POINTER:=  ADDR(NUM_BUF);
                :
           MOV   DS:WORD PTR DTEST+01AH,BX {Assumes BX contains addr of
                :                              NUM_BUF, IT DOESN'T}
       DISPLAY (5,25-TOT_NUM,TOT_NUM,POINTER);
                :
          MOV   AL,DS:BYTE PTR [BX+00018H] {also assumes this. wrong!}
                :
     END;
   END; .
```

Temporary solution:
Do not use the WITH statement. Reference all record members directly.

Signed off 08/31/88 in release A03.60

KPR #: D200085720 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
Compiler produces bad code for accessing parameters in nested procedures

Problem:
Compiler produces bad code when accessing parameters in nested procedures. Register are used twice and address are lost.

Temporary solution:
There is no known fix at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200085746 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
Bad code created when assgn ext real valu to real variable in a procedur

Problem:
Assigning externally declared real value to variable defined in a procedure as a real type, produces bad code.
The following program produces bad code:

```
"8086"  PREPROCESS

$EXTENSIONS+$
$SEPARATE_CONST-$
$ASM_FILE+$
```

KPR #: D200085746 **CONTINUED**

```
$POINTER_SIZE=32$

PROGRAM CODE_TEST;

$EXTVAR+$
$FAR_EXTVARS+$

  VAR
    SOMETHING:  REAL;
$FAR_EXTVARS-$
$EXTVAR-$

PROCEDURE ASSIGNMENT;

  VAR
    GIVE_ME_VALUE:   REAL;

BEGIN

 GIVE_ME_VALUE := SOMETHING;
      MOV   AX,SEG SOMETHING
      MOV   EX,AS
      LEA   SI,DS:SOMETHING
      PUSH  DS;     push eff addr of something
      PUSH  ES;     push seg of something
      LEA   DI,DS:ASSIGNMENT
      MOV   CX,#+00004H
      PUSH  DS;   push give_me_value
      POP   ES;   es = give_me_value
      CLD
      POP   DS;   ds = seg something
      REP   MOVSB;  move ds:si into es:di
      POP   DS;   ds = eff addr something

   GIVE_ME_VALUE := SOMETHING;
      LEA    SI,DS:SOMETHING;  this is the error, ds= the eff addr
              of something, not the dataseg for something.
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200085753 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
Addr Function for ORG'ed integer generating bad code

Problem:
Addr function for ORG'ED integer generating bad code.
The following program produces bad code; using ADDR function on a integer variable ORG'ed to a location:

KPR #: D200085753 **CONTINUED**

```
"PASCAL"   PREPROCESS
"8086"

$ALIGN OFF$
$AMNESIA ON$
$ASM_FILE_OFF$
$ASMB_SYM ON$
$DEBUG OFF$
$EMIT_CODE ONT$
$EXTENSIONS ON$
$EXTVAR OFF$
$FAR_LIBRARIES ON$
$FAR_PROC ON$
$FAR_EXTVARS$
$FULL_LIST OFF$
$GLOBPROC ON$
$GLOBVAR OFF$
$INTERRUPT OFF$
$IOCHECK ON$
$LINE_NUMBERS ON$
$lIST ON$
$LIST_CODE OFF$
$LIST_QBJ OFF$
$OPTIMIZE OFF$
$POINTER_SIZE 32$
$RANGE OFF$
$RECURSIVE OFF$
$SEPARATE_CONST OFF$

PROGRAM TEST_ADDR_FUNC;

#DEFINE ORD INTEGER

TYPE
  INTEGER = SIGNED_16;

VAR
 $ORG 020000000h$
   I : INTEGER;
 $END_ORG$

  INT_PTR_1  :^INTEGER;
  INT_PTR_2  :^INTEGER;
PROCEDURE MY_PROC;
BEGIN
  IF INT_PTR_2 = ADDR (I)
      PUSH   DS:DTEST_ADDR_FUNC+00006H
      PUSH   DS:DTEST_ADDR_FUNC_00004H
      MOV    AX,#+02000H
      MOV    ES,AX
      LEA    BX,DS:0000H
      PUSH   DS; should be PUSH ES
      PUSH   BX
```

Temporary solution:

KPR #: D200085753 **CONTINUED**

There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200085811 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
IMUL instruction will overwrite a value in DX register.

Problem:
Under certain circumstances the IMUL instruction will overwrite a value contained in the DX register which is needed at a later time.

```
"PASCAL" PREPROCESS
"8086"

$EXTENSIONS ON$
$SEPARATE_CONST ON$
$FAR_LIBRARIES OFF$
$POINTER_SIZE 16$

$ALIGN OFF$
$DEBUG OFF$
$INTERRUPT OFF$
$OPTIMIZE OFF$
$DS_EXTVAR$
$FAR_PROC OFF$
$SHORT_LIBRARIES ON$

PROGRAM TEST102;

CONST
 MAXINT = 32767;
TYPE
  INTeGER = SIGNED_16;
  Tilst_txt_lintyp = RECORD
                      Pos_L   : BYTE;
                      Pos_H   : BYTE;
                      Li      : BYTE;
                      Txtstreng : ARRAY[1..42] OF BYTE;
                     END;
Tilst_txt_element = RECORD
                      Tilst_txt_lin : ARRAy[1..2] OF Tilst_txt_lintyp;
                     END;

Tilst_txt_type    = ARRAY[0..16] OF Tilst_txt_element;

VAR
$ORG 00000650H$
  X1310 : BYTE;
$END_ORG$

$EXTVAR ON$
```

KPR #: D200085811 **CONTINUED**

```
#ifdef INTEL_es_un_joto
$FAR_EXTVARS$
#endif
Tilst_txt       : Tilst_txt_type;
$EXTVAR OFF$
J1310 : INTeGER;

$GLOBPROC ON$

PROCEDURE FAST_TXT_TILST(Tilst_txtnr,Linnie_ofset:INTeGER);

VAR
    N    : INTeGER;

BEGIN
    FOR N := 1 TO 2 DO
    BEGIN
     Tilst_txt[Tilst_txtnr].Tilst_txt_lin[N].Pos_L := N;
         MOV   AX,#+0005AH
         IMUL  SS:WORD PTR [BP+0006H]
         MOV   BX,AX
         LEA   BX,DS:Tilst_txt[BX-0002DH]
         MOV   DL,DS:BYTE PTR DFAST_TXT_TILST
         MOV   AX,#+0002DH
         IMUL  DS:WORD PTR DFAST_TXT_TILST
         ADD   BX,AX
         MOV   DS:BYTE PTR [BX],DL
  END; {END FOR}
END;
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200085829 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
Bad cd gen if proc declared EXT in another proc is called w/ FAR PROC ON

Problem:
Bad code is generated when a procedure declared external within another procedure is called, when the $FAR_PROC ON$ directive is used.

```
"8086" PREPROCESS
$FAR_PROC ON$
PROGRAM HAROLD;
$EXTENSION+$
VAR   B:BYTE;
PROCEDURE ERROR;
   PROCEDURE CLR_PORTB(MASK:BYTE); EXTERNAL;
BEGIN
    CLR_PROTB(B);
```

KPR #: D200085829 **CONTINUED**

```
   {CALL NEAR PTR CLR_PORTB <<< should be CALL FAR PTR CLR_PORTB}
END;
. {the problem is that a near call is made to CLR_PROTB rather than
a far call}
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200086579 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
Procedure Environ_init is missing from the simlib.R library

Problem:
Procedure Environ_init is missing from the simlib.R library

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200087882 Product: 8086/8 PASCAL 64814 03.50

Keywords: CODE GENERATOR

One-line description:
WITH statement generates bad code.

Problem:

```
"PASCAL" PREPROCESS
"80186"
$EXTENSIONS ON$
$SEPARATE_CONST OFF$
$SEPARATE ON$
$FAR_PROC  ON$
$GLOBPROC ON$
$FAR_LIBRARIES$
$POINTER_SIZE 32$
$FAR_EXTVARS$
$RECURSIVE ON$
$OPTIMIZE ON$
$DEBUG OFF$
$IOCHECK OFF$
$FULL_LIST OFF$
$LIST_CODE OFF$
$LIST_OBJ OFF$

PROGRAM Err_16;

$RANGE OFF$
```

KPR #: D200087882 **CONTINUED**

```
CONST

 No_of_tasks=24;
 No_of_mb = 16;
 No_of_mbq = 32;
TYPE
ADDRESS=RECORD
  CASE SIGNED_16 OF
    1: (i    :SIGNED_32);
    2: (ofs,seg  : UNSIGNED_16);
    3: (lo,hi    : SIGNED_16);
    4: (b0,b1,b2,b3 : BYTE);
    5: (P            : ^ADDRESS);
  END;
MAIL    = ADDRESS;
TASK_NO = BYTE;
TASK_STATUS = SET OF (
  dormant,
  suspended,
  delayed,
  wait_sema,
  wait_mail,
  wait_qmail);
TASK_CTRL_BLOCK = RECORD
  stack_adr   : ADDRESS;
  priority    : BYTE;
  status      : TASK_STATUS;
  next        : TASK_NO;
  delay_next  :  TASK_NO;
  delay       : UNSIGNED_16;
  FILLER      : ARRAY[0..5] OF CHAR;
  END;
MAILBOX_QUEUE = RECORD
  mail          : ARRAY[0..Length_of_mbq] OF ADDRESS;
  length        : SIGNED_16;
  inp           : SIGNED_16;
  outp          : SIGNED_16;
  waiting_task  : TASK_NO;
  FILLER        : BYTE;
  END;
VAR
$GLOBVAR ON$
$ALIGN ON$
  TCB     : ARRAY[0..No_of_tasks] OF TASK_CTRL_BLOCK;
  RUNNING : TASK_NO;
  READY   : TASK_NO;
  MBQ     : ARRAY[1..No_of_mbq] OF MAILBOX_QUEUE;
$ALIGN OFF$
$GLOBVAR OFF$

PROCEDURE MQ_IPOST( Queue  : SIGNED_16;
                    Message : MAIL;
                  VAR Error :SIGNED_16);
VAR
  WT : TASK_NO;
```

KPR #: D200087882 **CONTINUED**

```
  NXT : SIGNED_16;
BEGIN
WITH MBQ[Queue] DO
  BEGIN
  NXT := SUCC(inp) MOD length;
$OPTIMIZE OFF$
  IF NXT <> outp THEN
$OPTIMIZE ON$
  BEGIN
    mail[inp].lo := Message.lo;
    mail[inp].hi := Message.hi;
    inp := NXT;
    WT := WAITING_TASK;
    IF WT <> 0 THEN
      WITH TCB[WT] DO
         BEGIN
           waiting_task := next;
               MOV   CH,DS:BYTE PTR [BX+00006H]
               MOV   DS:BYTE PTR ]BX+00008AH],CH  did not load BX
                                              with MBQ[Queue]
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200087890   Product: 8086/8 PASCAL   64814   03.50

Keywords: CODE GENERATOR

One-line description:
WITH statement generates bad code.

Problem:
The following program produces bad code:

```
"PASCAL" PREPROCESS
"80186"
$EXTENSIONS ON$
$SEPARATE_CONST OFF$
$SEPARATE ON$
$FAR_PROC  ON$
$GLOBPROC ON$
$FAR_LIBRARIES$
$POINTER_SIZE 32$
$FAR_EXTVARS$
$RECURSIVE ON$
$OPTIMIZE ON$
$DEBUG OFF$
$IOCHECK OFF$
$FULL_LIST OFF$
$LIST_CODE OFF$
$LIST_OBJ OFF$

PROGRAM Err_16;
```

KPR #: D200087890 **CONTINUED**

```
$RANGE OFF$

CONST

 No_of_tasks=24;
 No_of_mb = 16;
 No_of_mbq = 16;
TYPE
ADDRESS=RECORD
  CASE SIGNED_16 OF
    1: (i   :SIGNED_32);
    2: (ofs,seg  : UNSIGNED_16);
    3: (lo,hi    : SIGNED_16);
    4: (b0,b1,b2,b3 : BYTE);
    5: (P           : ^ADDRESS);
  END;
MAIL    = ADDRESS;
TASK_NO = BYTE;
TASK_STATUS = SET OF (
  dormant,
  suspended,
  delayed,
  wait_sema,
  wait_mail,
  wait_qmail);
TASK_CTRL_BLOCK = RECORD
  stack_adr   : ADDRESS;
  priority    : BYTE;
  status      : TASK_STATUS;
  next        : TASK_NO;
  delay_next  :  TASK_NO;
  delay       : UNSIGNED_16;
  FILLER      : ARRAY[0..5] OF CHAR;
  END;
MAILBOX_QUEUE = RECORD
  mail         : ARRAY[0..Length_of_mbq] OF ADDRESS;
  length       : SIGNED_16;
  inp          : SIGNED_16;
  outp         : SIGNED_16;
  waiting_task : TASK_NO;
  FILLER       : BYTE;
  END;
VAR
$GLOBVAR ON$
$ALIGN ON$
  TCB     : ARRAY[0..No_of_tasks] OF TASK_CTRL_BLOCK;
  RUNNING : TASK_NO;
  READY   : TASK_NO;
  MBQ     : ARRAY[1..No_of_mbq] OF MAILBOX_QUEUE;
$ALIGN OFF$
$GLOBVAR OFF$

PROCEDURE MQ_PEND ( Queue   : SIGNED_16;
                    Message : MAIL)
BEGIN
```

KPR #: D200087890 **CONTINUED**

```
WITH MBQ[Queue] DO
  BEGIN
  IF inp = outp THEN
  BEGIN
    WITH TCB[RUNNING] DO
      status := status + [wait_qmail];
  END;
  Message.lo := mail[outp].lo;
  Message.hi := mail[outp].hi;    {BX gets overwritten here}
 outp := SUCC(outp) MOD length;   {BX not available}
.
.
.
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A03.60

KPR #: D200090597 Product: 8086/8 PASCAL 64814 03.50

Keywords: PROBLEM ON 9000/S300

One-line description:
Assignment of constant into array of 3 elements does not work.

Problem:
When the following code is compiled CONST_data is never defined even though it is referenced in the assembly code.

```
"8086"
$EXTENSIONS ON$
PROGRAM test;
CONST
  SYNC_PATTERN = 20F3FAH;      <---------  32 Bits
TYPE
  UB = UNSIGNED_8;
  UB3 = ARRAY[1..3] OF UB;     <---------  24 Bits
VAR
  SYNC : UB3;
BEGIN
  SYNC := UB3(SYNC_PATTERN);  <----------  LEA  SI,DS:CONST_data
END.
```

Note: The array element is not an even multiple of bytes, therefore it is unclear what the compiler is supposed to do with the constant. The assembly code generated uses CONST_data without defining it.

Temporary solution:
In this particular situation a fix would be to change the array declaration of [1..3] to [1..4]. That is, to use 32 bits instead of 24. The user must be very careful and make sure the proper code is being generated. The compiler will

KPR #: D200090597 **CONTINUED**

save these bits in a particular order - which may not be the order the customer had desired! Refering to page 10-12 in the HP-UX Hosted Pascal Cross Compiler - 8086 manual might be helpful. Refer to the section on Functional Type Change.

KPR #: D200055657 Product: 8086/88 C M 64818-90905 03.00

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int   cntr;
  int   (*tmp)();

        for (cntr=1; cntr<4; cntr++)
          { tmp = func5(cntr);
        }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 03.02

KPR #: 5000238337 Product: 8086/88 PASCAL M 64814-90903 03.00

Keywords: MANUAL

One-line description:
Change manual to say that libraries need to be in same segment

KPR #: 5000131029 Product: 8086/88/186/188HLSAM 64332-90902 02.00

One-line description:
Display variable may result in "ERROR:E64".

KPR #: 5000141150 Product: 8086/88/186/188HLSAM 64332-90902 02.00

One-line description:
Data structures too large to display in "display variable" command.

Temporary solution:
Reducing the ascii string size of the variable names
shall cause less space to be used in the 64340 analyzer.
Therefore development can continue with the freed space.

Duplicate Service Requests: 5000141143

KPR #: D200031831 Product: 8088 DQ SW ANALYZER 64341C 01.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 01.02

KPR #: D200031765 Product: 8088 SW ANAL 64333 02.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 02.03

KPR #: D200081232 Product: 8088 DQ EMUL 300 64221S004 01.10

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:
Display Memory and Modify Memory will be incorrect at the segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the center of a line.

Modify Memory will be incorrect if done beyond the end of a segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment in the middle of the line being displayed.

Modify Memory will be correct if it is not extended through the end of a segment. For example:

```
        modify memory 0FFFEH to 1,2 will be correct.

        modify memory 0FFFEH to 1,2,3 will NOT be correct, because
          the third entry is in the next segment.
```

Signed off 08/31/88 in release A01.20

KPR #: D200082115 Product: 8088 DQ EMUL 300 64221S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 08/31/88 in release A01.20

KPR #: D200083121 Product: 8088 DQ EMUL 300 64221S004 01.10

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

Signed off 08/31/88 in release A01.20

KPR #: D200085936 Product: 8088 DQ EMUL 300 64221S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

Signed off 08/31/88 in release A01.20

KPR #: D200086116 Product: 8088 DQ EMUL 300 64221S004 01.10

One-line description:
Software Breakpoints don't work in target memory.

Problem:
Software breakpoints do not work in target memory.
There is no workaround; updated software is required.

Temporary solution:
Software breakpoints do not work in target memory.
There is no workaround; updated software is required.

KPR #: D200086116 **CONTINUED**

Signed off 08/31/88 in release A01.20

Duplicate Service Requests: D200089912

KPR #: D200090746 Product: 8088 DQ EMUL 300 64221S004 01.10

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/31/88 in release A01.20

KPR #: D200031773 Product: 8088 DQ SW ANAL 64333B 01.00

One-line description:
Using local static variables in C causes a lockup in the analyzer

Problem:
Using local static variables in the C language causes the analyzer to lock up. The analyzer looks through the dynamic data area without ever realizing that the variable is in the static data area, thus an infinite loop results.

Temporary solution:
The problem itself has no workaround, the customer must not use local static variables until new software is available.

Signed off 11/06/87 in release 01.03

KPR #: D200069476 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

Signed off 10/09/87 in release 01.20

KPR #: D200080218 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Absolute code part user,part emul, will be overwritten at boundary.

Signed off 10/09/87 in release 01.20

KPR #: D200080275 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Apparent error during disassembly of the offset at an intrasegment jump.

Signed off 10/09/87 in release 01.20

KPR #: D200080325 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Paging at a segment end produces a confusing CS:IP.

Signed off 10/09/87 in release 01.20

KPR #: D200080572 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX

KPR #: D200080572 **CONTINUED**

applications.

Signed off 10/09/87 in release 01.20

KPR #: D200080879 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

Signed off 10/09/87 in release 01.20

KPR #: D200081224 Product: 8088 EMULATION 300 64226S004 01.00

One-line description:
Modify/Store memory abort at physical addr 0 for seg/offset procs

Problem:
Modify memory for a large range (> 4096 bytes) and store memory (> 250 bytes) that crosses the physical address 0 boundary will fail somewhere after physical address 0 on segment:offset processors. For example, on the 8086, the following command will modify only part of the requested range:

```
        modify memory 0FF00H:0 thru 0FF00H:0FFFFH to 0
```

since address 0FF00H:1000H is in fact physical address 0.

Temporary solution:
Temporary work around:

Do not attemt to modify/store memory through physical address 0H.

Signed off 10/09/87 in release 01.20

KPR #: D200081273   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Display memory line crossing segment boundary will be wrong

Problem:
Display Memory and Modify Memory will be incorrect at the segment wrap around, under the following conditions:

Display Memory will be wrong when the segment end is in the center of a line.

Modify Memory will be incorrect if done beyond the end of a segment.

Temporary solution:
Temporary workaround for each situation is as follows:

Display Memory should not be set to have the end of the segment in the middle of the line being displayed.

Modify Memory will be correct if it is not extended through the end of a segment. For example:

```
        modify memory 0FFFEH to 1,2 will be correct.

        modify memory 0FFFEH to 1,2,3 will NOT be correct, because
          the third entry is in the next segment.
```

KPR #: D200081422   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Relative path names (e.g. ./cmd) should not search PATH

Problem:
A new feature was added to the core feature set to search for command files using the users PATH variable for a search path. A defect has been introduced such that specifying a relative path with a command file still has the command file looked for in the search path.

Relative path names should override the PATH variable like they do in a standard shell. Only names not containing any '/' should be searched for using PATH. All others (especially ./name) should be used relative to the current directory.

Temporary solution:
Specify command files with full path names if the application is unable to find your command file.

KPR #: D200082156   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:

KPR #: D200082156 **CONTINUED**

Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
          cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083162   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200084947   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
"modify memory" command results in an "end release".

Problem:
The "Modify Memory" command results in an "end release".

KPR #: D200085969   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                             ; end locks the emulation session
3. <system name> <module name>     ; continues the emulation session
4. display trace
```

KPR #: D200085969 **CONTINUED**

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200089920   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Emulator does not work reliably with 64155B memory controller

Problem:
Detailed Listing for Defect Number LSDqf03557
Text:
  emulator does not work reliably with 64155B memory controller

Any 8-bit processor which uses a foreground monitor may have strange problems when using the 64155B memory controller. This is due to the MONITOR_CONTROL word being 16 bits wide. The 8-bit processors require two bus cycles to modify this word. Unfortunately, it is possible to read the value between those two bus cycles, resulting in a bogus value being passed up to the host. The likelihood of this happening depends on the timing of the monitor - using a different assembler/linker or changing the monitor code can cause the problem to appear or disappear. A simple example is the 68008DP: running a particular program in target memory, then attempting to modify target memory produces an undefined software breakpoint message.

Detailed Listing for Defect Number LSDqf03557
Text:
  emulator does not work reliably with 64155B memory controller

Any 8-bit processor which uses a foreground monitor may have strange problems when using the 64155B memory controller. This is due to the MONITOR_CONTROL word being 16 bits wide. The 8-bit processors require two bus cycles to modify this word. Unfortunately, it is possible to read the value between those two bus cycles, resulting in a bogus value being passed up to the host. The likelihood of this happening depends on the timing of the monitor - using a different assembler/linker or changing the monitor code can cause the problem to appear or disappear. A simple example is the 68008DP: running a particular program in target memory, then attempting to modify target memory produces an undefined software breakpoint message.

KPR #: D200090787   Product: 8088 EMULATION   300 64226S004   01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.

KPR #: D200090787 **CONTINUED**

For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

  End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
  End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 5000134619 Product: 8096 ASSEMB 64860 01.00

One-line description:
display GLOBAL/LOCAL does not work when using the ROM emulator

Signed off 06/02/86 in release 01.01

KPR #: 5000180000 Product: 8096 ASSEMB 64860 01.03

One-line description:
Rom emulator does not display local/global symbols correctly w/ 8096 cod

Signed off 12/02/87 in release Z01.04

KPR #: 5000191767 Product: 8096 ASSEMB 64860 01.03

Keywords: CODE GENERATOR

One-line description:
Linker does not allocate the file at even addresses

Problem:
The 8096 linker does not allocate files at even addresses.
The following example shows this problem:

```
file_a     "8096"                      file_b      "8096"
            L1  DSL  2                              L3  DSL  2
            L2  DSW  1                              L4  DSW  1
            ST  SP,L2                               ST  SP,L4
```

Link above files with load address prog=20H.

```
FILE/PROG NAME            PROGRAM      DATA
_____________________________________________
file_a                    0020
next address              002F

file_b                    002F  <----  Customer wants to allocate
next address              003E         this file from an even
                                       address, automatically
                                       from linker.
```

Temporary solution:
Two possible work-arounds exists. The first is to use an ORG statement in each file to place the file on an even boundry. The second work-around is to place the statement "DSL 0" at the end of each file, thus leaving the file on an even boundry.

KPR #: 5000225078 Product: 8096 ASSEMB 64860 01.03

Keywords: CODE GENERATOR

One-line description:
Using ORG statemnts can generate ERR_LR errors

Problem:

KPR #: 5000225078 **CONTINUED**

The following program will produce a Legal Range error on line 9,10 and 11.

```
 1 "8096"
 2            ORG        1800H
 3 CPTG1      DSW        2
 4            ORG        1C00H
 5 TEST       DSB        4
 6            PROG
 7 AX         EQU        10H
 8 AL         EQU        20H
 9            ST         AX,CPTG1
10            STB        AL,CPTG1+1
11            LDB        AL,TEST
```

Temporary solution:
Move ORG statement to end of the program.

```
 1 "8096"
 2           PROG
 3 AX        EQU       10H
 4 AL        EQU       20H
 5           ST        AX,CPTG1
 6           STB       AL,CPTG1+1
 7           LDB       AL,TEST
 8           ORG       1800H     <-*
 9 CPTG1     DSW       2         <-*
10           ORG       1C00H     <-*
11 TEST      DSB       4         <-*
```

KPR #: 5000275305 Product: 8096 ASSEMB 64860 01.04

One-line description:
Pseudo instruction DCB treats absolute variable as relocatable.

Problem:
The pseudo instruction DCB is not treated correctly by linker. The label which is defined by DSW is used as operand of DCB. After linked, the label is not assigned the proper value. The value is not absolute but relocatable. The following is a example;

```
"8096"
AX    DSW  1
      ST   SP,AX
      DCB  AX
```

After the above program is linked, AX remains as relocatable value at DCB line. But the AX in ST SP,AX is assinged the absolute value.

NOTE: Since no emulator exists for this processor, the problem can be verified by looking at the :absolute file. When the program is linked with PROG address set to 80H, the :absolute file shows:

```
C301  8000  1800
```

KPR #: 5000275305 **CONTINUED**

```
                                   ||
                                   ||
         This 00H is not the absolute value.
         Load address of PROG is 80H, so the code should be 80H.
```

Therefore, the correct code must be "C301 8000 1880".

KPR #: 5000151241 Product: C COMPILER REF M 64800-90907 00.07

Keywords: MANUAL

One-line description:
Add note to compiler supplements regarding 3000 symbol limit.

Problem:
When compiling a program on the HP 64100A or HP 64110A, there is a limit on the amount of symbols an entire program can have. This limit is about 3000 without a XREF and around 1500 if an XREF is generated.

Temporary solution:
Avoid using more than 3000 symbols (1500 if a XREF is generated) in your entire program.

Fix information:
Fix is documented in Software Notice 5958-6068 R2707.

Signed off 08/05/87 in release 01.02

KPR #: D200033399 Product: EBPP 64304 01.03

One-line description:
Configuration file not automatically loaded when using EBPP

Problem:
The configuration file is normally loaded automatically for a specific emulator when using the EBPP. This fails to happen when using the following emulators.

emulator	prod. #	config. file
Z80	64253A	CZ80H_E:HP
dequeued 80C86.	64220C	C8687D_E:HP
dequeued 80C88	64221C	C8887D_E:HP
dequeued 8086/87	64220B	C8687D_E:HP
dequeued 8088/87	64221B	C8887D_E:HP
dequeued 68000	64243A	C68000D_E:HP
dequeued 68008	64244A	C68008D_E:HP
dequeued 68010	64245A	C68010D_E:HP

Temporary solution:
Load the configuration file when in the state analyzer by hitting the softkeys "configuration load_from <file>". <file> being the configuration file for that emulator.

Signed off 07/18/86 in release 99.99

KPR #: D200090928 Product: F9450 EMUL 300 64286S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200075150 Product: F9450 EMULATION 64286 01.04

One-line description:
RS232 Simulated IO will overrun the user's read buffer sometimes.

Problem:
R232 Simulated IO will overrun the user buffer under certain conditions. If the Last Byte Address Pointer coincidentally is pointing to the location of the Read Buffer End Address Pointer, and the 64000 station has received characters input to the RS232 port, when the "Update Read Buffer" (8CH) command is given, the 64000 will write the new bytes to the memory following the Buffer End Address. The 64000 should write the new characters to the Buffer End Address location and then wrap around to the Buffer Begin Addrss location. Instead the new characters continue to be written to ever increasing address locations.

Duplicate Service Requests: 5000194373

KPR #: D200087395 Product: GENERIC ANALYSIS M 64740-90909 01.00

One-line description:
Errors in xtt help screen.

KPR #: D200085084 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
Analyzer ROM PV fails about 1 time every 6 hours.

Problem:
The analyzer ROM PV fails about once in every 6 hours of continious running using the "pv 0" command.

Signed off 04/14/88 in release A00.03

KPR #: D200085092 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
"tck" Command does not give errors for invalid options.

Problem:
The tck command is inconsistant with other odyssey commands in that it does not give an error message if invalid options are given.

An example that does not work:

```
        tck junk
```

Signed off 04/14/88 in release A00.03

KPR #: D200085605 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
Help tsq in easy mode still showes -t option

Problem:
The help for tsq in easy mode still shows the -t option, but this option is not valid in easy mode.

Signed off 02/02/88 in release A00.02

KPR #: D200085613 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
The use of the xttq command can cause the storage of incorrect trans

Problem:
The use of the xttq command when in transitional mode timing will result in insufficient transitions being stored by the analyzer. In addition to storing transitions on the channels specified it is necessary to store any channel which could cause the analyzer to trigger to ensure that the trigger will be in memory.

Signed off 02/02/88 in release A00.02

KPR #: D200085621 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
CMB-exec trace started message should be flagged ASYNC-STAT

Problem:

KPR #: D200085621 **CONTINUED**

The message printed as
"!STATUS 1305! CMB execute; emulation trace started"

should be changed to
"!ASYNC-STAT 1305! CMB execute; emulation trace started"

Same change for message printed for CMB external anly trace.

Printing an ASYNC-STAT message as STATUS could possibly confuse the programmatic/host, since they assume that any message that isn't ASYNC is a direct result of the command that was just processed. The two messages described above are always ASYNC.

Signed off 02/02/88 in release A00.02

KPR #: D200086637 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
When "tg arm and addr=4" command then "Label not defined:any"

Problem:
Detailed Listing for Defect Number LSDqf02436
Text:
when "tg arm and addr=4" command then "Label not defined:any"
.submitter

When the user entered a command in the tcf -e mode and the command started with "arm" and had more arguments then the wrong error message was printed.

For example

```
tsto arm and data=4
Label not defined: any
```

This was fixed in Jan 88 by Steve Warntjes by a 2 line change in lib/parnor.c

Signed off 04/14/88 in release A00.03

KPR #: D200087023 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
Arm to trigger time count is off by 120ns

Signed off 04/14/88 in release A00.03

KPR #: D200087387 Product: GENERIC ANALYSIS FW 64740 00.00

One-line description:
"xtarm always" generates an error message

Signed off 04/14/88 in release A00.03

KPR #: D200087619 Product: GENERIC ANALYSIS FW 64740 00.02

One-line description:
"ts" after "init -c" shows incorrect "trigger in memory"

Problem:
When executing the "init -c" command and then executing the "ts" command, the user will receive the following status:

```
  --- Emulation Trace Status ---
  User trace halted
  Arm received
  Trigger in memory
  Arm to trigger <0.04 uS
  States 0 (0) ?..?
  Sequence term ?
  Occurrence left ?
```

Since the user just initialized the hardware, it is not understandable that a trigger is in memory, nor that the arm is recieved. The ver command returned the following information:

```
  HP64700 Series Emulation System
    Version:    A.00.04 27Jan88

  HP64764A Intel 80186 Emulator
    Version:    A.00.01 27Jan88
    Speed:     10 MHz
    Memory:    126 KBytes

  HP64740 Emulation Analyzer
    Version:    A.00.02 23Dec87
```

Signed off 04/14/88 in release A00.03

KPR #: D200088013 Product: GENERIC ANALYSIS FW 64740 00.02

One-line description:
Changing the trace configuration causes error with the fast clock speed

Problem:
Changing the trace configuration using the tcf command causes an error if tck -s is set to F or VF.

Signed off 04/14/88 in release A00.03

KPR #: D200088021 Product: GENERIC ANALYSIS FW 64740 00.02

One-line description:
Arm to trigger time can be incorrect if: clock is set to the fast mode

Problem:
The Arm to trigger time displayed with the ts command can be wrong if the clock speed (tck -s) is set to F or VF. This is because the trigger must propigate through the input pipeline

KPR #: D200088021 **CONTINUED**

(this requires subsequent user clocks) before the arm to trigger time counter is stoped.

Signed off 04/14/88 in release A00.03

KPR #: D200088138 Product: GENERIC ANALYSIS FW 64740 00.02

One-line description:
HELP xteq scrolls off screen

Problem:
Help for xteq scrolls off the screen, but looks like it shouldn't have to.

KPR #: D200088153 Product: GENERIC ANALYSIS FW 64740 00.02

One-line description:
short help for trc says telif is "seq glb restart"

Problem:
The short help for the trace commands say telif is "seq glb restart". While this is true in easy mode, it's not really accurate for complex mode.

KPR #: D200090290 Product: GENERIC ANALYSIS FW 64740 00.02

One-line description:
Incorrect absolute time count when trigger is not found

Problem:
Detailed Listing for Defect Number LSDqf03610
Text:
Incorrect absolute time count when trigger not found

The absolute count is sometimes incorrect when trigger is not found. On several occasions the '-1' state had a time count of -4 uS, which is incorrect behavior.

The absolute time count should always be 0 for the '-1' state when trigger is not found.

Temporary solution:
There is no known workaround available.

Signed off 07/22/88 in release A00.03

KPR #: D200085258 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Baud rate setting of 38400 changes to 19200

Problem:
When one port has a baud rate of 38400 and the other port has its baud rate changed to any setting other than 38400, the first port changes to a baud rate of 19200.

This problem was found because Odyssey could not power up with the rear panel switch set for 38400 on Port A.

Signed off 02/02/88 in release A00.02

KPR #: D200085274 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
The command "bc -d bp" can result in strange behaviour.

Problem:
"bc -d bp" can result in strange behavior

```
  >cf rrt=dis          #not restricted to real-time runs
  >map A..B tram       #we'll put a breakpoint in this mem block
 R>map C..D eram       #we'll put a loop-to-self in this mem block
 R>b
 M>mem C=loop-to-self  #modify mem to install the loop-to-self
 M>bc -e bp           #enable the break-on-software breakpoints condition
 M>bp A               #define a breakpoint anywhere in the block A..B
 M>r C                #run user code
 U>bc -d bp           #disable the break condition
 !ERROR !: Unable to run
 U>                   #the command actually succeeded; the error message
                      # should not have been printed
```

On the 64753 (z80), if "cf qbrk=en", the last four lines change to:

```
 M>r C                #run user code
 U>bc -d bp           #disable the break condition
 U>                   #no error message, and we are now unning user code
```

Proper operation is supposed to be:

```
 M>r C                #run user code
 U>bc -d bp           #disable the break condition
 U>                   #no error message, maybe a tempbreak for memory
                      # accesses; return to running user code
```

Temporary solution:
The user should be warned to always break to monitor before giving the "bc -d bp" command. This is the temporary workaround to avoid the peculiar error message; the final fix will require the same warning.

Signed off 02/02/88 in release A00.03

KPR #: D200085365 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Main Help screen has emul listed twice.

Problem:
When the command "help" command is issued with no options, a list of VALID <group> NAMES is listed. Under this list "emul" is listed twice.

Signed off 02/02/88 in release A00.03

KPR #: D200085530 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Ending value of data stream does not report proper error.

Problem:
When performing a memory modify like the following:

```
   >m 0..3=1,2,3,4:junk
```

where the stream of data values is equal to the buffer of memory we which to fill, no error is reported on the last value of 4:junk. The memory does get correctly modified to the values 1,2,3,4 however an error should be reported on the 4:junk value.

KPR #: D200085563 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Failure to run from R state after rx rst, x,x,rst,r

Problem:
When setting up a execute run from reset, two execute pulses were received, the emulator given a rst command then a run was attempted. An error message unable to run was received and the prompt was that the emulator was in the monitor. If done with a single execute in the sequence the run was successful.

Signed off 02/02/88 in release A00.02

KPR #: D200085597 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Can't ser up CMB to run-from-power-up

Problem:
Command sequence:

```
  cf clk=ext    #set clock to external; target system power is off
  rx rst        #set run-at-execute to run from target system reset
  x             #emit execute pulse
```

Result:

```
  !ASYNC-ERROR ! Unable to break
```

Reason:
In the rxint routine, if the routine "statbreak" returns a

KPR #: D200085597 **CONTINUED**

failure (e.g., no clock, therefore no monitor, therefore unable to break), rxint won't call break_handler (because we can't clear break causes unless we're in monitor), and won't call run. The run will not occur.

Signed off 02/02/88 in release A00.02

KPR #: D200085639 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Too many Commands in command line causes command truncation

Problem:
If the command line length is exceeded, then the command in process is truncated, that truncated command is executed and if erroneous, no errors appear. There are three separate problems here--one is truncated commands which can cause unexpected actions to occur, one is no error messages result from the truncated command, and the most serious is the automatic execution of all commands in the line even if no return is hit!! Thus, the person is not allowed to recognize that his line is too long and edit it--it just executes the command as soon as character overflow occurs. The line length isn't very long (about 255 chars) and is independent of number of commands in the line. This limit is easily reached by putting in a comment with the command.

Signed off 02/02/88 in release A00.02

KPR #: D200085647 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Improper coverage calculation of overlapping ranges

Problem:
If coverage is used, and multiple ranges are used which overlap, the calculation of coverage sometimes gets an incorrect value. For example, if only location 0fffh has been accessed, the following command is wrong (a cov -r has been done to initialize coverage):

```
cov 0ffe..1000 0fff..1000
percentage of memory accessed: % 40.0
```

The correct result is given by

```
cov 0ffe..1000
percentage of memory accessed: % 33.3
```

Since multiple ranges are allowed, the coverage algorithm should be fixed so that overlaps are correctly computed.

KPR #: D200086173 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Incorrect documentation in help screen for grammer

Problem:
In the help screen for the grammar, the statement is made that the base for repetition counts defaults to decimal, all others are hex. This is incorrect. The number of seconds in a wait command is also defaulted to decimal.

Signed off 04/14/88 in release A00.05

KPR #: D200086199 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
stty command changes on 1200 and 300 will not return a prompt.

Problem:
When the current baud rate on the HP64700 is 1200 or 300 baud and an stty command is given to turn off echo - a prompt is not returned by the HP64700. This prevents the "programmatic interface" from powering up at these baud rates.

Signed off 03/25/88 in release A00.03

KPR #: D200086207 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Extended hex format symbol records cause download problems

Problem:
When downloading Extended hex format all symbol records are to suppose to be ignored. The HP64700 fails to do this properly - causing the next data record to be missed.

Signed off 03/25/88 in release A00.03

KPR #: D200086215 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
"Stepping aborted" status message may or may not appear.

Problem:
1) When stepping is aborted on the last requested step (9th of 9, first of 1, etc.), the status message
"Stepping aborted; number steps complete: xx"
is printed, and xx shows the correct number. The message should not be displayed here.

2) When stepping is aborted on next-to-last step (8th of 9, 14th of 15, 1st of 2, etc), the message will NOT be displayed. It should be displayed here.

The consequences are that a debugger interface that extensively uses the "step once" command, looking for some condition will run slower than it should, because it is processing all the extra

KPR #: D200086215 **CONTINUED**

characters that are being transmitted when that condition is found (e.g., software breakpoint).

In addition, the user may be misled if the "Stepping aborted" msg does not appear in case 2. Stepping will abort, and the accompanying status message that states the condition (e.g., "Software breakpoint: 1234") will appear, but without the message, the user may think that stepping did complete.

Signed off 02/04/88 in release A00.03

KPR #: D200086512 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
The help screen says load-load emulation memory

Problem:
The help screens say : load - load emulation memory when it really should say : load - load memory

Signed off 04/14/88 in release A00.05

KPR #: D200086520 Product: GENERIC EMULATION FW 64700 00.03

One-line description:
"Cntl C" after a power-on can crash the emulator

Problem:
If a control C is hit just after the powerup initialization the emulator might not work.

Signed off 03/25/88 in release A00.04

KPR #: D200086595 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
"ser" gives incorrect pattern match address for TMS32020

Signed off 04/14/88 in release A00.05

KPR #: D200086652 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Command "map -d 0" hangs the system

Signed off 04/14/88 in release A00.05

KPR #: D200086660 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
A cntl-C break after "init" may not inititialize properly

Signed off 04/14/88 in release A00.05

KPR #: D200086868 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
The "run" and "step" commands do not check for ambigous addr obj

Signed off 04/14/88 in release A00.05

KPR #: D200087031 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Reading into garded memory can't return less than 16 bytes

Problem:
Detailed Listing for Defect Number LSDqf02554
Text:
  Reading into guarded memory can't return less than 16 bytes

When attempting to do a read memory range that crosses good memory into guarded memory (or target memory with the processor reset) can never return less than 16 bytes of good data.

```
Example:
          With the memory map is as follows:
              0h..1fffh is eram
              other is grd

          The following behavior occurs:
              r 1ff0..200f  returns 16 bytes of good data
              r 1ff1..200f  returns 0 bytes of good data
```

This is a problem for the user interfaces, because they may be trying to do an absolute memory display and valid addresses will appear as if they were guarded memory.

Manfred Arndt

Responsible engineer: Eric Kuzara

Signed off 04/14/88 in release A00.05

KPR #: D200087452 Product: GENERIC EMULATION FW 64700 00.00

One-line description:
Incorrect info on "help io", can't display IO in long words

KPR #: D200087742 Product: GENERIC EMULATION FW 64700 00.04

One-line description:
Some problems with the "step" command

Problem:
If
   a command is given to step several instructions
and then
   some step AFTER the first is able to be initiated but
   unable to complete,
the result is

KPR #: D200087742 **CONTINUED**

the error/status messages are misleading, because they give the user the idea that "n" instructions were executed when only "n-1" instructions were executed.

Other messages indicate that the emulator is trying (and failing) to display the mnemonic and the current program counter, which should not be displayed if the step failed to complete.

Also,
If
the step command is to be executed in "quiet" mode (no display)
and then
the first step command is able to be initiated, but unable to complete,
the result is
no error or status message at all.

If the command is not executing in "quiet" mode, there are other messages being printed, so the user can see that something is wrong.

If the user is using a host interface, the stepping failure might not show up as a command failure, even though it should.

In most cases, the user would get some kind of "asynchronous" error message. If a transient condition prevents a step from completing properly, but the condition goes away before the next attempt to communicate with the monitor, stepping will proceed as though nothing went wrong. This combination of factors is very unlikely to occur.

Temporary solution:
If the user executes one step at a time, NOT in quiet mode, even a transient condition will, in MOST cases, result in an error message.

If the condition that prevents step completion is very short-term, and its effect is limited to the instruction being stepped, there is no workaround.

Signed off 04/14/88 in release A00.05

KPR #: D200088054 Product: GENERIC EMULATION FW 64700 00.04

One-line description:
Break, Breakpoint, Mem Access cause confusion if they occur simultaneous

Problem:
If a software breakpoint is detected during a memory access, the emulator may get confused about whether it's supposed to be running or not. A subsequent "reg" command may result in the emulator going back to running user code, when it should stay in the monitor.

Signed off 04/14/88 in release A00.05

KPR #: D200088070 Product: GENERIC EMULATION FW 64700 00.04

One-line description:
Error message priority of break messages needs to be changed.

Problem:
If the vector table for the INT 3 interrupt points to GUARDED memory, the emulator generates a guarded memory access when a software breakpoint is encountered. The error is reported on the status line as an unknown software breakpoint instead of a guarded access (both errors show up in the error log).

Signed off 04/14/88 in release A00.05

KPR #: D200090308 Product: GENERIC EMULATION FW 64700 00.04

One-line description:
Restricted load fails when file loads to guarded memory

Problem:
Detailed Listing for Defect Number LSDqf03679
Text:
Restricted load fails when file loads to guarded memory

.submitter

Doing a load command "load -e" or "load -t" fails if guarded memory gets accessed. This is incorrect behavior, because of the restricted load, guarded memory should not be accessed at all.
Manfred Arndt

.labnotes

In memcmds/accessmem.c the function accessmapmem() rejects an access to guarded memory without checking to see if the access is supposed to be restricted to emulation-only or target-only. If the access is restricted, then requests to access guarded memory should just be ignored (no failure).

Pisces 2 works in the expected way.

Resp engr: Eric Kuzara

KPR #: D200091264 Product: GENERIC EMULATION FW 64700 00.05

One-line description:
Odd byte format records may cause an extra byte written to memory

Problem:
An odd number of bytes contained in an HP format absolute record can cause an extra byte to be written to memory. This problem can be seen on the Z80 emulator and possibly the 186. (not seen on the 68000).

This does not effect the execution of a program which has been loaded. The primary problem seen by the user is when calculating a checksum over the data loaded into memory. The extra byte written with an odd record is random in value.

KPR #: D200091264 **CONTINUED**

Temporary solution:
There is no known workaround available.

KPR #: 5000163303 Product: HOST PASCAL 64817 01.04

One-line description:
IOERROR not generated.

Problem:
The following program does not result in an IOERROR # 11 when a string instead of an integer is entered from the keyboard. Using $IOCHECK ON$, the error is detected.

```
VAR I,J : INTEGER;
    F   : TEXT;
BEGIN
RESET(F,'keyboard');
$IOCHECK OFF$
READLN(F,I);
IF IOERROR <> 0 THEN
   BEGIN
     J := IORESULT;
     WIRTELN('ERR');
   END;
END.
```

KPR #: D200014357 Product: HOST PASCAL 64817 01.04

One-line description:
Spurious run-time error doing WRITE(REAL_VAL) after previous I/O error

Problem:
WRITE(REAL_VAL) fails to reset the I/O error indicator. The result is a spurious run-time error if one writes a REAL value immeadiately after a previous I/O error or end-of-file condition. For example,

```
   VAR R:REAL;
   BEGIN
   WHILE NOT EOF DO {Eventually produces end-of-file, an error}
     READLN;
   WRITELN(R);      {Write real immeadiately after EOF causes run-time
                     error erroneously.}
```

Temporary solution:
Put a dummy I/O operation before the write of the real. For example, one could write the null string before writing the real. The intervening I/O call resets the error indicator.

```
   WHILE NOT EOF DO
     READLN;
   WRITELN('',R);  {Writing null string resets error indicator}
```

KPR #: D200015305 Product: HOST PASCAL 64817 01.04

One-line description:
STRWRITE function may produce run time error in specific case.

Problem:
The following HOST Pascal program will produce a run time error based on the STRWRITE function. This is incorrect since only one

KPR #: D200015305 **CONTINUED**

item is being written into the string 's'.

```
  program test (input, output);

     var
        s : string[3];
        d : integer;

     begin
        setstrlen (s, 3);
        strwrite (s, 3, dummy, 'c');
     end.
```

Temporary solution:
As a temporary work-around check the value of 'dummy' and reset to STRMAX (s) or less if necessary.

KPR #: 1650016618 Product: HOST SOFTWARE / VAX 64882 01.60

One-line description:
Transfer fails when downloading relocatable libraries

Problem:
Libraries of relocatables are created by appending the files together. Relocatables uploaded from the HP64000 station can be mixed with relocatables created on the host. Transfer does not handle the mixed source relocatable libraries correctly during download. The result of initiating such a transfer is unpredictable.

Temporary solution:
Each relocatable file could be transfered seperately and combined on the 64000 with the library command.

Duplicate Service Requests: 5000187922 1650025270

KPR #: 5000149724 Product: HOST SOFTWARE / VAX 64882 01.60

Keywords: RCMAIN HIGH SPEED LINK

One-line description:
HSL transfer from within RCMAIN does not return control to RCMAIN.

Problem:
An HSL transfer from within RCMAIN, although it complete successfully, does not return control to RCMAIN.

If a 64000 was not selected prior to the transfer, it will additionally return an error message:
"Unrecognized flag (z) option."

Temporary solution:
This problem does not effect any transfer outside of RCMAIN nor RS232 transfers from within.

KPR #: 5000151290 Product: HOST SOFTWARE / VAX 64882 01.60

One-line description:
RCMAIN corrupts RCDEVICE.dat file when aborted with Cntl C or Y

Problem:
If a vax terminal hangs up while in rcmain utility and I do a cntl c or a cntl y the rcmain.dat file gets corrupted and an orderly exit from rc main is not accomplished. need to enhance the software to allow for cntl c or cntl y to exit from rcmain if you cannot use the exit command in rc main.
We do need to enhance the software to accept the cntl y or c characters to allow for an orderly exit from the rcmain routine. at present the .da t file does get trashed when cntl c or y is executed
customer needs to be copied on this response...
If a vax terminal hangs up while in rcmain utility and I do a cntl c or a cntl y the rcmain.dat file gets corrupted and an orderly exit from rc main is not accomplished. need to enhance the software to allow for cntl

KPR #: 5000151290 **CONTINUED**

c or cntl y to exit from rcmain if you cannot use the exit command in rc main.

Temporary solution:
NO KNOWN WORK-AROUND

KPR #: 5000180323 Product: HOST SOFTWARE / VAX 64882 01.70

Keywords: RCMAIN

One-line description:
RCDEVICE.DAT is not properly maintained.

Problem:
RCMAIN does not update the RCDEVICE.DAT file properly under certain conditions. Example: A small file with only two lines defining two station entries such as:

```
    edp1     tty1
    edp2     tty3
```

will sometimes not show a busy status when browsed even when a station is selected. The file size has also changed after exiting RCMAIN, and on one occasion placed a B in the busy field.

Temporary solution:
None.

KPR #: 5000239921 Product: HOST SOFTWARE / VAX 64882 02.00

Keywords: TRANSFER

One-line description:
Transfer of files over DECnet causes program to crash

Problem:
High Speed Link transfer may fail when file is accessed via DECnet on a remote node. Error messages such as:
   Improperly handled exit condition......   are displayed
and the transfer fails to complete.

Temporary solution:
   Use DECnet to first copy the file to the node where the high speed link resides, then perform the transfer.

KPR #: D200045096 Product: HOST SOFTWARE / VAX 64882 01.20

One-line description:
Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.

Problem:
None of the HP programs react well to the normal VAX terminal control commads - CNTRL Z; CNTRL Y; CNTRL C. The programs are not consistant in how they react.

For example if rcmain hangs it is necessary to edit the rcmain file.

KPR #: D200045096 **CONTINUED**

The only file to cause real damage was the RCMAIN. I used cntrl Y to exit while connected to the HP. The program left the HP in a busy state that was not cleaned up. A data file had to be edited by hand to correct.

Temporary solution:
None.

KPR #: D200047217 Product: HOST SOFTWARE / VAX 64882 01.20

One-line description:
LONG COMMANDS GREATER THAN 1024 CHAR. MALFUNCTION WITH DMF-32 I/O CARD

Problem:
LONG COMMANDS (>1024 CHAR) INPUT TO REMOTE CONTROL CAUSES CORE DUMP OR OR LOSS OF FUNCTIONALITY OF TERMINAL INUSE WHEN 64000 I/O CARD IS DMF-32

KPR #: D200059428 Product: HOST SOFTWARE / VAX 64882 01.60

Keywords: RCMAIN

One-line description:
Vax rcdevice file not updated correctly

Problem:
On the VAX, the rcdevice.dat file used by remote control is not being updated correctly when there is no comment at the end of a device entry. It is also affected by the position of the entry in the file. The error is noticed when the entry is the lase entry in the device file.

KPR #: D200059444 Product: HOST SOFTWARE / VAX 64882 01.60

Keywords: RCMAIN

One-line description:
VAX remote control dumps when a very long command is entered

Problem:
On the VAX, one of the regression tests for remote control asks you to enter an 11 line command. This causes the remote control session to end in a stack and register dump.

KPR #: D200064055 Product: HOST SOFTWARE / VAX 64882 01.70

Keywords: RCMAIN

One-line description:
/DEVICES= does not work with a list of stations.

Problem:
When RCMAIN is invoked as:

```
    rcmain/dev=(x,y)
```

where x and y are devices listed in RCDEVICE.DAT file, the program

KPR #: D200064055 **CONTINUED**

goes directly into interactive mode. When invoked with a single device, the program goes directly into remote control.

Temporary solution:
Use only device lists that consist of one device.

KPR #: D200079483 Product: HOST SOFTWARE / 300 64883 01.00

Keywords: TRANSFER

One-line description:
Transfer does not handle extra line-feeds in file.

Problem:
Transfer needs to correctly handle "extra" line-feed characters which may be in the host file. These line-feeds appear in files which have been transferred from the VAX to the 9000 series 300.

KPR #: D200079681 Product: HOST SOFTWARE / 300 64883 01.00

Keywords: TRANSFER

One-line description:
Incorrect syntax/usage may not result in warning or error message.

Problem:
High Speed Link (transfer -h) software may not always catch the use of invalid file names or illegal syntax. For example, the command

```
        $ transfer -tah file1 file1@1
```

will transfer the file "file1" into "FILE1::source@0". Note that the file is transferred to the wrong cluster. Transfer should (1) copy the file to cluster 1, or (2) flag the transfer statement as syntactically incorrect (the 64000 file name is lower case).

KPR #: D200085076 Product: HOST SOFTWARE / 300 64883 01.00

One-line description:
Cluster - Cluster Transfer does not work with filelist

Problem:
Cluster - Cluster transfer via High Speed Link with a filelist does not work. The first file is transfered and a message indicates this is so, however, the second file is not transfered, and the only message is that it is NOT transfered - no other messages. None of the other files in the list are transfered or attempted to transfer. If the second file in the list does not exist, the message is:

```
     ZZZZZZ:userid@n     NOT transfered to ZZZZZZ:userid@m
```

Temporary solution:
Transfer the files one at a time.

KPR #: D200087148 Product: HOST SOFTWARE / 300 64883 01.10

One-line description:
Transfer may abort on >32K files.

Problem:
Transfer does not properly use updated free list after releasing and regaining disc. The transfer may try to write to a sector which is no longer free, causing the process to be killed.

KPR #: D200087148 **CONTINUED**

Temporary solution:
NO TEMPORARY SOLUTION AT THIS TIME.

Signed off 04/29/88 in release A01.60

KPR #: 5000169698 Product: HOST SOFTWARE / 500 64880 01.06

Keywords: TRANSFER

One-line description:
Transfer does not correctly parse "FILE:USERID:@HSL".

Problem:
Entering the command:

```
transfer -hast file  FILE:USERID:@1
```

transfers the file to HSL0 with no comments. The leading colon in front onf @1 should generate a syntax error and produce no transfer instead. The correct syntax is:

```
transfer -hast file  FILE:USERID@1
```

KPR #: 5000191544 Product: HOST SOFTWARE / 500 64880 01.60

Keywords: TRANSFER

One-line description:
Transfer may not move library files.

Problem:
Create a relocatable file on a 64000 workstation using one of the available compilers or assemblers. Transfer this file to the 9000 computer.

Next, create a relocatable file on the 9000. Merge the 64000 created file and the 9000 created file into a library file using the cat(1) command, i.e.

```
                $ cat file1.R file2.R > lib.R
```

An attempt at transferring the file 'lib.R' (either RS-232 or HSL) will fail. In the case of the High Speed Link, the error message returned is

```
                WARNING: Memory fault
```

Temporary solution:
Create a file list containing the relocatable file names which belong in the library. Transfer ALL relocatable files to the 64000 system using the '-l' option (list option) of transfer.

Create a relocatable library file on the 64000 using the library command.

KPR #: D200036608 Product: HOST SOFTWARE / 500 64880 01.20

One-line description:
Transfer to blank userid does not translate file names correctly.

Problem:
Translating files into the blank userid (":") results in incorrect

KPR #: D200036608 **CONTINUED**

file name translations. When downloading files for emulation, debug, ..., the blank userid should not be used.

KPR #: D200037275 Product: HOST SOFTWARE / 500 64880 01.20

One-line description:
xx.L TO xx:link_sym translation wrong for 0 length records (types 3 & 4)

Problem:
When transfer (or translate) translates a host linker symbol file (.L file) which has a 0 length type record 3 or 4, the output file is incorrect.

KPR #: D200043877 Product: HOST SOFTWARE / 500 64880 01.20

Keywords: RCMAIN

One-line description:
A session command is req'd before entering the menu in batch jobs.

Problem:
Entering the rcmain menu in a batch job before doing a select command results in hanging remote control. This is only true for select menue commands. Problems can be solved by making sure there is a blank after the menue command, AND doing a remote session command as the first command in the job.

KPR #: D200062539 Product: HOST SOFTWARE / 500 64880 01.50

Keywords: TRANSLATE

One-line description:
C.K.1 and C.K.2 both translate to C_K on the 64000.

Problem:
C.K.1 and C.K.2 both translate to the same file name on the 64000 when a file is transfered.

Temporary solution:
None.

KPR #: D200065938 Product: HP 64020A UPGRADE M 64020-90902 00.00

One-line description:
Retrofit kit does not include fans.

Signed off 07/08/87 in release 201.01

KPR #: D200066241 Product: HP 64120A CARDCAGE M 64120-90902 00.00

One-line description:
Communications PCA is 64120-66508 (new) 64120-69508 (exchange)

Problem:
In Chapter 1 of the manual, in Table 1-1, two fans are listed as being included in the kit.

Temporary solution:
The listing is incorrect. Fans are not included in the kit.

KPR #: 1650032698 Product: HP TEAMWORK 300 64711S004 01.00

One-line description:
The laser printer 'loses' a few columns across page breaks in a DFD.

Problem:
When diagrams are printed on the laserjet which are wider than a page, so that it is cut and printed on several pages, a few milimeters of the diagram are lost. This means that a small column isn't printed on page 1 and not printed on page 2 either.

Temporary solution:
None at this time.

KPR #: 1650033720 Product: HP TEAMWORK 300 64711S004 02.00

One-line description:
Removing models from the index does not delete all its files.

Problem:
When models are removed from the database by selecting them from the model index and selecting "delete" not all files which belong to the model are actually purged from the disc.

Since they can't be accessed from within teamwork anymore they just fill up the disc.

Temporary solution:
Issue a 'dump_twk -d dumpfile -all' command.

'cd $DBPATH/twk_0_files'

'rm dir.*/*'

Return to the dumpfile directory and

'load_twk -d dumpfile -all'

KPR #: 1650054486 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
"background" colour change when text is selected.

Problem:
When Teamwork is used on an SRX system all the text is displayed with a purple background. When text is selected the background turns to blue.

Temporary solution:
None.

KPR #: 1650059162 Product: HP TEAMWORK 300 64711S004 02.03

One-line description:
The funct. of the mid. and rt. button of 46060b needs to be exchanged.

Problem:
THE FUNCTIONALITY OF THE MIDDLE AND RIGHT BUTTON OF THE 46060B 3 BUTTON MOUSE NEEDS TO BE EXCHANGED. THE CURRENT FUNCTION IS:

2 BUTTON MOUSE	3 BUTTON MOUSE
LEFT	LEFT
BOTH	RIGHT
RIGHT	MIDDLE

Temporary solution:
There is no workaround available.

KPR #: 5000263111 Product: HP TEAMWORK 300 64711S004 02.20

One-line description:
Data base error caused by NOTE manipulation. -- fixed in 2.3

Signed off 06/17/88 in release A02.30

KPR #: 5000283184 Product: HP TEAMWORK 300 64711S004 02.20

One-line description:
When moving a large group with "group move", some boxes are left hanging

Problem:
When moving a large number of objects with the "move group" command in the structure chart editor sometimes the invocations are not moved with the couples and modules. The invocations remain at the old location on the display. They could be accessed and then deleted or moved, however, when we re-created the problem using the Cruse Control model with version 2.3 of TEAMWORK at LSD.

Temporary solution:
The only solution we were to develop to this problem was to access the invocation and manually move it to where it was supposed to go during the "group move".

KPR #: D200089342 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
Simultaneous socket connections cause a hang.

Problem:
Simultaneous socket request of the dc_server will cause some requests to be ignored.

Requests are generated by starting HP Teamwork or opening an Index or Diagram.

After a request is "ignored", that session of HP Teamwork can appear to be "hung".

KPR #: D200089342 **CONTINUED**

Temporary solution:
None at this time.

KPR #: D200090480 Product: HP TEAMWORK 300 64711S004 02.30

One-line description:
When a bubble is moved in a data flow diag., the old num. isn't removed.

Temporary solution:
These old bubble numbers are erased when the screen is reprinted.

KPR #: D200066654 Product: HP TEAMWORK SA M 64710-90901 00.01

Keywords: MANUAL

One-line description:
Need clarification in Default Printer section of manual.

Problem:
Clarification is needed in the Default Printer section of the manual (page 11-6). The final two menu selections need further explanation:

apple :postscript

Mentor_DOC:Mentor_DOC

A short paragraph describing which printers are supported by HP Teamwork/SA would be extremely helpful.

Signed off 12/04/86 in release 00.02

KPR #: 5000235143 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
Manual should include guidelines for swap space config when installed.

Signed off 05/12/88 in release X01.01

KPR #: D200077636 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
PRINT OBJECTS from the PI doesn't work correctly.

Problem:
If one of the DFD's is open and you return to the PI window, select any number of objects (as long as it included the DFD that is open), PRINT OBJECTS (from the PI window), the follow error message is printed:

The DFD Context-Diagram;0 was not printed for the following reason: object locked.

(The Context-Diagram was the DFD that was open in this case.)

The DFD was not active. It was partly obscured by the PI window. It should be 'readable' for printing.

Temporary solution:
None at this time.

KPR #: D200077891 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
Spline is too large for binder.

Temporary solution:
no temporary solution.

KPR #: D200090274 Product: HP TEAMWORK SA M 64711-90903 01.00

One-line description:
twk_image -dd misbehaves.

Problem:
twk_image -dd -m Model_name causes an error:

```
    SYNTAX ERROR:  No "object name specified".
```

even though the manual says an object name is not necessary.

Temporary solution:
None.

KPR #: D200055780 Product: HP-UX 6800-03 C M 64821-90902 01.40

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int    cntr;
  int    (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200055780 **CONTINUED**

Duplicate Service Requests: D200055806

KPR #: 1650004499   Product: HP-UX 68000/8/10 A M 64845-90905   01.30

One-line description:
Assembler flagging LR error for correct offset when using PC+IND+OFFSET.

Problem:
The following program shows a problem with PC-relative addressing with displacement. The displacement is taken as the low-order 8 bits of the label instead of relative to the current PC.

```
         ORG      0F8H
         MOVE     LABEL[PC,D6],D6
         ORG      102H
LABEL    DC.W     0FFFFH
```

This results in an error message:

LR - Legal Range, Address or displacement out of range of the instruction's addressing capabilitities.

Temporary solution:
No temporary solution.

KPR #: D200045880   Product: HP-UX 68000/8/10 A M 64845-90905   01.30

One-line description:
Wrong offset calculated when using PC+index reg+ offset mode of addr.

Problem:
When using the PC relative with offset and index register mode of addressing the assembler may caluculate the wrong address. The error will be made if the offset symbol is at an absolute location greater than FFH.
"68000"

```
     ORG     010H
     MOVE    #0,D0
     JMP     TABLE[PC,D0]
     ORG     100H
TABLE        DS.W     10
```

Temporary solution:
No temporary solution.

KPR #: D200081836   Product: HP-UX 68000/8/10 A M 64845-90905   01.04

Keywords: MANUAL

One-line description:
Cannot substitute Macro parameter at beginning of variable.

Problem:
The following macro parameter substitution will not work:
"68000"
 LABELAA
AALABEL

KPR #: D200081836 **CONTINUED**

```
PRO10   MACRO     &P1
        BRA.W     LABEL&P1 ---------> O.K.
        BRA.W     &P1LABEL ---------> DOES NOT WORK, BUT W/ MOTOROLA
        MEND                          ASSEMBLER IT WILL  WORK

        PRO10     AA
```

The manual needs to be changed in section 6, the section dealing with Macros. On pages 6-4, 6-5 we need to explain, that the kind of macro substitution above, will not work. The problem is, the assembler parses the ENTIRE value, &P1LABEL, and cannont find a parameter

KPR #: D200055707   Product: HP-UX 68000/8/10 C M 64819-90903   01.40

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200064386   Product: HP-UX 68000/8/10 C M 64819-90903   01.40

One-line description:
Byte parameters are pushed onto the stack incorrectly.

Problem:
When passing a byte parameter it is not pushed onto the stack as the manual specifies it will be. The Pascal and C manual specify that a byte parameter will be pushed in the upper byte of the word which is pushed on the stack. The C compiler does a Move.W and pushes the char in the lower byte. The pascal compiler does the push correctly.

```
"C"
"68000"

char called_func();

calling_func() {

  char  passed_parm;
  passed_parm = 'b';

  called_func(passed_parm);
}

char  called_func(parm)
char  parm;
{
  char local_var;
  local_var = parm;
}
```

KPR #: D200036913 Product: HP-UX 68000/8/10 P M 64815-90907 01.20

Keywords: TYPE CONVERSION

One-line description:
Signed_8 to Unsigned_16 is incorrect.

Problem:

```
VAR S8 : SIGNED_8;
    US16 : UNSIGNED_16;

BEGIN
US16 := UNSIGNED_16(S8);  This does a sign extend which is incorrect.
```

Temporary solution:
None at this time.

Duplicate Service Requests: D200051623

KPR #: D200046896 Product: HP-UX 6805/9/9E A M 64844-90905 01.00

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix in documented in Software Notice5958-8821 R2707.

Signed off 08/06/87 in release 01.40

KPR #: D200055822   Product: HP-UX 6809/09E C   M 64822-90902   01.20

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int        (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.04

KPR #: D200055822 **CONTINUED**

Duplicate Service Requests: D200055848

KPR #: D200063651   Product: HP-UX 6809/09E C   M 64822-90902   01.00

One-line description:
Clarification of interface for USER_DEFINED and real number routines.

Problem:
In the example below, 6809 libraries cannot be explicitly called. If they are called explicitly as routines, the stack is built differently than when a compiler generated call is made.

Example:

```
"C"
"6809"

$FIX_PARAMETERS ON$

main() {

int x;
double xx;
extern double LONGREAL_FLOAT();
int *px;
int *pxx;
x =5;
px = &x;
pxx = &xx;
LONGREAL_FLOAT(px,pxx);              /* Conversion is not made */
}
```

Temporary solution:
For explicit use of ALL the real number library routines, declare your routines as in the following example with $FIXED_PARAMETERS ON$ and $RECURSIVE OFF$ (Chapter 5 in manual). The compiler will then generate the proper form of parameter passing to satisfy the real number library. Note, $RECURSIVE OFF$ is also necessary when using the USER_DEFINED interface method (Chapter 3 in manual).

Example:

```
"C"
"6809"

extern int xint;
extern double xdouble;
extern int *pxint;
extern int *pxdouble;

extern recursive_variable_func();
$FIXED_PARAMETERS ON$
```

KPR #: D200063651 **CONTINUED**

```
extern recursive_FIXED_PARM_func();
$RECURSIVE OFF$
extern LONGREAL_FLOAT();
/*NOTE do not declare these functions double. It will cause extra
parameters to be passed*/

main() {

$LIST_CODE ON$
  /* NOTE: Parameter passing method for standard C function*/
recursive_variable_func(&xint,&xdouble);
            LDU   #xdouble
            LDY   #xint
            LDD   #00004H
            PSHS  X,Y,U
            LBSR  recursive_varia
            LEAS  000000006H,S
  /* NOTE: Parameter passing method for FIXED_PARAMETER(Pascal) function
*/
recursive_FIXED_PARAM_func(&xint,&xdouble);
            LDU   #xdouble
            LDY   #xint
            LBSR  recursive_FIXED
xint = 5;
            LDD   #00005H
            STD   xint
pxint = &xint;
            LDD   #xint
            STD   pxint
pxdouble = &xdouble;
            LDD   #xdouble
            STD   pxdouble
/* NOTE: Parameter passing method for STANDARD REAL NUMBER LIBRARY funct
ion */
/* A (Pascal) function with $FIXED_PARAMETERS$ and $RECURSIVE OFF$ !*/
LONGREAL_FLOAT(pxint,pxdouble);
            TFR   D,X
            LDD   pxint
            LBSR  LONGREAL_FLOAT
LONGREAL_FLOAT(&xint,&xdouble);
            LDX   #xdouble
            LDD   #xint
            LBSR  LONGREAL_FLOAT

/* Compare the stack build on this assignment call which uses
LONGREAL_FLOAT versus the explicit call above */

xint = xdouble;
            LDX   #xint
            LDD   #xdouble
            LBSR  LONGREAL_TRUNC
xdouble = xint;
            LDX   #xdouble
            LDD   #xint
            LBSR  LONGREAL_FLOAT
```

KPR #: D200063651 **CONTINUED**

```
}
        Rmain
            GLOBAL   Rmain

            RTS
        Dmain
            RMB      0000EH
            GLOBAL   main
        Emain        EQU $-1

            GLOBAL   Emain

            EXTERNAL LONGREAL_TRUNC
            EXTERNAL LONGREAL_FLOAT
```

Fix information:
Fix is documented in Software Notice 5959-2129 R2707.

Signed off 08/06/87 in release 01.40

Duplicate Service Requests: D200063677

KPR #: 5000240937 Product: HP-UX 8051 ASSM M 64855-90903 01.40

Keywords: MANUAL

One-line description:
Change 8051 manual page 8-4

Problem:
The 8051 assembler/linker reference manual has paragraph with errors relating to the CSEG directive. The errors are typographical, but can lead to confusion:
 Page 8-4 of the HP-UX hosted manual talks about the CSEG directive with a paragraph:

The code segment counters can be charged (changed) with the DS, DW, and DW (DB) pseudos, and with each instruction encoded. Each unit in the program relocatable counter represents one byte in the code address space within the range of 0 to 64. (0 to 64K)

(corrections are indicated in parenthesis)

Temporary solution:
Be aware of these changes when using the 8051.

KPR #: D200053785 Product: HP-UX 8051 ASSM M 64855-90903 01.30

One-line description:
The $ operand does not work as defined.

Problem:
If the $ operand is used in a multi-byte instruction, it should specify the value of the PC at the beginning of that instruction. In the following example, it represents the value of the PC in the middle of the MOV instruction:

```
"8051"
      ORG     10H
      MOV     A,#$        ; moves 11H into A instead of 10H

      END
```

Temporary solution:
Use $-x instead of $ where x represents the offset back to the first byte of the multi-byte instruction:

```
"8051"
      ORG    10H
      MOV    A,#$-1    ; this will move 10H into A
      END
```

Signed off 04/07/88 in release X00.00

KPR #: D200053801 Product: HP-UX 8051 ASSM M 64855-90903 01.00

One-line description:
The $ operand does not work as defined.

Problem:
If the $ operand is used in a multi-byte instruction, it should specify the value of the PC at the beginning of that instruction. In the following example, it represents the value of the PC in the middle of the MOV instruction:

```
"8051"
      ORG     10H
      MOV     A,#$        ; moves 11H into A instead of 10H

      END
```

Temporary solution:
Use $-x instead of $ where x represents the offset back to the first byte of the multi-byte instruction:

```
"8051"
      ORG    10H
      MOV    A,#$-1    ; this will move 10H into A
      END
```

Signed off 04/07/88 in release X00.00

KPR #: 1650008128 Product: HP-UX 8085 C M 64826-90902 01.50

One-line description:
New and dispose have inconsistient parameters

Problem:
If you call DISPOSE as the manual states on page 4-10 a run time error is flagged.

```
DISPOSE (&pointer, sizeof (*pointer));  /* as in manual. */

DISPOSE (pointer, sizeof (*pointer));   /* this works.   */
```

This would be acceptable, but, NEW is called with the address of a pointer as the manual states. It seems that NEW and DISPOSE should be called in the same manner.

Temporary solution:
Call DISPOSE with a pointer rather than its address.

```
DISPOSE (pointer, sizeof (*pointer));
```

KPR #: D200055897 Product: HP-UX 8085 C M 64826-90902 01.40

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor name"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
```

KPR #: D200055897 **CONTINUED**

```
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
         }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.60

Duplicate Service Requests: D200055913

KPR #: D200079574 Product: HP-UX 8086/88 ASSM M 64853-90905 02.20

Keywords: CODE GENERATOR MANUAL

One-line description:
.

Problem:
THE FOLLOWING PROGRAM PRODUCES AN error ET.

"70108"

```
       GLB     HPTEST
       PROG
       EXTERN  SYMBOL
HPTEST PROC    FAR
       ASSUME  DS0:DATA,PS:PROG
       MOV     AW,SEG SYMBOL
       MOV     DS0,AW
       MOV     AW,SYMBOL
       INC     AW
       MOV     SYMBOL,AW
       RET
```

Temporary solution:
There is no known solution at this time.

KPR #: 5000211359 Product: HP-UX 8086/88 C M 64818-90903 03.02

One-line description:
Additional info about the $SEPARATE_CONST$ directive works, pg. 2-3.

Problem:
This SR consists of 2 complaints. The first requests that the $SEPARATE_CONST ON/OFF$ option default to off, instad of on. This will not be implemented at this time, because of the installed base using our compilers.

The second is a request to change the manual, making it easier to find information on "preparing the program for prom programming". This request has been turned over to the manual writers, and will be addressed soon.

KPR #: D200055665 Product: HP-UX 8086/88 C M 64818-90903 03.10

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

"C"
"processor"

```
int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

"C"
"processor"

```
int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
```

KPR #: D200055665 **CONTINUED**

```
        { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 03.30

Duplicate Service Requests: D200055681

KPR #: 5000188813 Product: HP-UX 8086/88 PAS M 64814-90904 01.01

One-line description:
DOC. FOR THE PASCAL LIB. ERROR HANDLING ROUTINES NEEDS IMPROVEMENT.

Problem:
Documentation for the Pascal Library error handling routines needs improvement. For Example: The manual documents that a routine "CaseError" is called if an unexpected CASE value is encountered. If the user does not supply his own "CaseError" routine, he/she can eventually determine that our CaseError library routine calls a routine called "Abort", which in turn calls "PASCAL_ERROR". If HP's monitor is linked, PASCAL_ERROR is in the monitor program. If the monitor is not linked, we provide a PASCAL_ERROR routine. This routine never returns control to the calling routine. Neither Abort nor PASCAL_ERROR are documented in the manual.
Please improve the documentation in this area.

Temporary solution:
No known temporary solution at this time.

KPR #: 5000174805 Product: HP-UX OP SYS M 64801-90903 00.09

One-line description:
Setting the TERM variable to vt101a will allow use of pmon

Signed off 01/28/87 in release 00.10

KPR #: 5000182246 Product: HP-UX OP SYS M 64801-90903 01.00

One-line description:
Method of entering this CONTROL-M should be explained to the reader.

Signed off 03/05/87 in release 01.01

KPR #: D200079517 Product: HP-UX OP SYS M 64801-90903 01.00

One-line description:
Meas system unuseable if WINDEX exited without ending measurement.

Problem:
If you run a 64000-UX measurement system
feature under WINDEX, and you exit WINDEX
without first doing an "end" or "end release_system"
within the feature, the measurement system in
question becomes hung in the "in-use" state
(even though noone is using it). It cannot be
released without rebooting the HP-UX system.

Temporary solution:
Do an "end" or "end release_system" within the
measurement system before exiting WINDEX.

KPR #: D200090431 Product: HP-UX OP SYS M 64801-90903 01.00

One-line description:
ftio command for hp-ux 6.01 does not function as documented.

Problem:
ftio command for hp-ux 6.01 does not function as documented.
IE. ftio -ocx / complains about not being able to locate something
and eventually terminates abnormally.

This problem has been fixed for hp-ux 6.2.
Also, this SR was entered under the wrong product code.

Temporary solution:
There is no workaround.

KPR #: 5000182824 Product: HP-UX SYSTEM INST M 64880-90901 01.02

One-line description:
DOC SHOULD INCLUDE LIST OF SUPPORTED CARDS FOR RS232 XFER.

Problem:
64000 II rs232 transfer function and the available cards.
Rs232 transfer from the 9000/300 is only supported through the 98628 rs232 card. It is not supported on the human interface card and there is no documentation that lets you know that. Two requests.

1. Documentation should include list of supported cards for rs232 transfer right next to the suggestion for using rs232 transfer for file transfering from the 64000 to the 64000 II and near all explanations of rs232 transfer. teh suggestion to transfer all files from old ssytem to new using rs232 is in the instalation and configuration manual.

2. Make the human interface card a supported card. Most people get it because it seems to be the most versital. it seems to be a waist to have a card and not be able to use it fully.

Temporary solution:
No temporary solution.

KPR #: 5000269381 Product: HP-UX SYSTEM INST M 64880-90901 01.00

One-line description:
Manual needs to be more explicit about /dev/ttyXX where XX is numeric

KPR #: D200068429 Product: HP-UX SYSTEM INST M 64880-90901 01.02

Keywords: HIGH SPEED LINK

One-line description:
Fails to transfer first passworded file, but doesn't notify the user.

Problem:
When transfering a file list to a passworded HP64000 userid,
transfer will ask you for the password. If you misstype the password,
which happens alot because of UPPER/lower case differences, transfer
will display an error message, tell you that the first file is not
transfered, and prompt you again for the password. If you then type
the password correctly, transfer will go ahead and transfer the
remaining files, BUT NOT THE FIRST ONE.

Temporary solution:
None at this time.

KPR #: D200047019 Product: HP-UX USER DEFIN A M 64851-90906 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8832 R2707.

Signed off 08/06/87 in release 01.40

KPR #: D200064030 Product: HP-UX USER DEFIN A M 64851-90906 01.30

Keywords: MANUAL

One-line description:
'&' is comment field of a macro causes a parameter error.

Problem:
If your comment field in a macro has a '&' in it a parameter error is generated.

"processor"

```
MAC1     MACRO     &P1,&P2
         DC.&P1    &P2          ;P1 & P2
         MEND

         MAC1  W,2
```

If the comment field in a macro contains a '&', a parameter error is generated.

"processor"

```
MAC1     MACRO     &P1,&P2
         DC.&P1    &P2          ;P1 & P2
         MEND

         MAC1  W,2
```

Temporary solution:
Avoid using a '&' in a macro comment field.

Fix information:
Fix is documented in Software Notice 5958-8832 R2707.

Signed off 08/06/87 in release 01.50

KPR #: D200046839 Product: HP-UX Z80/NSC800 A M 64842-90904 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8817 R2707.

Signed off 08/05/87 in release 01.40

KPR #: D200055855   Product: HP-UX Z80/NSC800 C M 64824-90902   01.40

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int    cntr;
  int    (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200055855   **CONTINUED**

Duplicate Service Requests: D200055871

KPR #: D200055749   Product: HP-UX Z8001/02 C   M 64820-90902   01.30

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

        for (cntr=1; cntr<4; cntr++)
          { tmp = func5(cntr);
          }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.40

KPR #: D200055749 **CONTINUED**

Duplicate Service Requests: D200055764

KPR #: 5000283630 Product: INT SNSL BD 8-16 UPM 64404-90901 01.00

One-line description:
The 68020 emul. may not show all source lines assoc. with exec. code.

Problem:
The 68020 emulator may not show all the source lines which are associated with the executed code.

When the first word of an opcode is not aligned on a four byte boundary, the source line for that opcode will not be displayed UNLESS the trace is disassembled. Users that are using "display trace source only" normally do not need to issue any disassembly commands.

Temporary solution:
No workaround available.

KPR #: D200077933 Product: INVERSE ASSEMB 64856 01.01

One-line description:
Can loop forever when a source file contains macros.

Problem:
It is not possible to use macros in the Inverse Assembler source definition.

Temporary solution:
Until this problem is fixed, either avoid using macros, or simply expand all macros that already exist.

KPR #: D200047126 Product: M-STD 1750A ASM M 64857-90901 00.54

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8842 R2707.

Signed off 08/06/87 in release 01.05

KPR #: 5000117507 Product: MS1750A ASSEMB 64857 00.00

One-line description:
'DEFF' generating incorrect code for NAN's.

Problem:
When using the pseudo DEFF the following does not generate any warning or correct hex:

```
                DEFF 1E128
     hex generated  49DD 24AA which is some negative number.
```

Signed off 01/14/88 in release Z02.70

KPR #: 5000231076 Product: MS1750A ASSEMB 64857 01.04

One-line description:
Incorrect code generated for immediate negative data.

Problem:
Bad code generated when variables equated to negative numbers are used. Exmaple:

```
"1750AGD"
neg25k   EQU  -25000
hex25k   EQU  9E58H
         LD   #neg25k,R0
         LD   #-25000,R0    ;both generate 8307H - opcode for
                            ;LD4  #-8,R0
         LD   #hex25k,R0
         LD   #9E58H,R0     ;both generate correct code (8500 9E58)
         LD   #0,R0         ;these
         ADD  #neg25k,R0    ;two statements achieve the desired result
         END
```

Temporary solution:
There are two versions of the IEEE specifications for the 1750A assembler. One version is known as the General Dynamics Version and the other is known as the Proprietary Software Specification (PSS). Our assembler follows the PSS specifications which require double pound (#) signs to indicate immediate long data. This is true for all cases except when immediate data is used with the XIO command. In this case only one pound (#) sign is required.

To correct the problem described in this SR use double pound signs to indicate long immediate data.

```
"1750AGD"

neg25k          EQU     -25000
hex25k          EQU     9E58H

                LD      ##neg25k,R0
                LD      ##hex25k,R0
                END
```

Signed off 01/14/88 in release Z02.70

KPR #: D200060491 Product: MS1750A ASSEMB 64857 01.04

One-line description:
Invalid instruction assembles without error message

Problem:
The following TBR instruction is invalid syntax.

```
   "1750AMS"
   START           TBR        R3,0      opcode - 5730
```

The syntax for This instruction is TBR N,RB (not TBR RB,N). The opcode generated is a valid opcode for the instruction TBR 3,R0 (not TBR R#,0). No error message was generated to point out this mistake.

Temporary solution:
No known temporary solution.

Signed off 01/14/88 in release Z02.70

KPR #: D200091637 Product: OP NOTE 68000C AXLSM 5959-2191 01.00

One-line description:
5959-2191 op note say to use "what" to determine revision code. No!

KPR #: 1650058925 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Problem with Macro code generation.

Temporary solution:
There are several work arounds:
- Use a subroutine
- Use the following code:

```
     M2    MACRO
           .GOTO   FOO
     FOO   .NOP
     L&&&&  NOP
```

It's hard for us to see the exact use of the MACRO sent, so we could probably supply a better workaround with a more specific example.

KPR #: 2700005769 Product: OPERATING SYSTEM 64100 01.39

Keywords: DC600

One-line description:
DC600 backup hangs up when it encounters a defective tape.

Problem:
Customer attempted DC600 backup from keyboard of master. Backup never completed. Error message appeared very briefly claiming a CRC check failure. Then system attempts to restart backup procedure. Situation continues ad nauseum.

Bill Furch has complete details and customer's tape.

Temporary solution:
no temporary solution

KPR #: 5000203620 Product: OPERATING SYSTEM 64100 02.10

One-line description:
The '.' character is causing problems for rev 2.06 of asm.exe.

Problem:
The 2.06 operating system causes errors when a . is used in an assembler statement. Ex:

```
                  MOVE.W    0[A0,D0.W],BUF (ref. to D0.W)
                  or        symbol.OR.symbol2
```

In both cases the "." causes errors in the second example, if there are spaces befor the and after the .OR., "symbol .OR. symbol2", then the . seems to recognized. The only way for all .'s to be recognized is to go back to O.S. 2.05.

This problems may appear on different processors with different symptoms. However, the problem will always arise to to the mishandling of '.'.

Signed off 08/31/88 in release A02.11

KPR #: 5000209007 Product: OPERATING SYSTEM 64100 02.10

Keywords: CODE GENERATOR

One-line description:
Intractive link gives invalid lnk/command (.K) file if full path too big

Problem:
Valid interactive link may yield invalid linker command (.K) file. The following interactive link is an example:

```
$lnk <CR>
object files  lnkbug.R
library files  /users/hp64k/johng/sub1/sub2/sub3/sub_directory4/sub_dire
ctory5/a5_lib.R<CR>
.
.
.
```

The problem is that because the full path to the library file does not fit completely on the original line, the linker creates a .K file that looks like:

```
object files  lnkbug.R
library files
/users/hp64k/johng/sub1/sub2/......  (this wraps around)
```

Relinking with this .K file yields the pass one fatal error:
"*****ERROR: Initialization Syntax error in linker command file"

Temporary solution:
Edit the .K file and combine the lines.

Signed off 08/31/88 in release A02.11

KPR #: 5000214106 Product: OPERATING SYSTEM 64100 02.10

One-line description:
ef directive does not work with mneumonics with "."

Problem:
The following program creates an incorrect xref lisiting on the 300:

```
1   "8051"  XREF
2    SYMA   EQU 1
3    SYMB   SETB   0B0H.SYMA
4          JP    ACC.SYMA,SYMB
5          JC    SYMB
```

XREF FOR 64000 LOOKS LIKE:

```
LINE #   SYMBOL     TYPE    REFERENCES
     2    SYMA        A    3,4
     3    SYMB        P    4,5
```

KPR #: 5000214106 **CONTINUED**

XREF FOR SERIES 300 LOOKS LIKE:

```
LINE#    SYMBOL     TYPE    REFERENCES
    2     SYMA        A
    3     SYMB        P      5
```

Temporary solution:
There is no known solution at this time.

Signed off 08/31/88 in release A02.11

Duplicate Service Requests: 5000212977

KPR #: 5000252825 Product: OPERATING SYSTEM 64100 02.10

Keywords: CODE GENERATOR

One-line description:
Assembler output listing is missing part of line number at EQU statement

Problem:
Assembler output listing is missing the thousand column for EQU instruction.

Example:

```
                1489   label   MOVE ......
         < 1234> 490           EQU
                ^
             missed !!!!!!!!!
```

Temporary solution:
There is no known work around at this time.

Signed off 08/31/88 in release A02.11

KPR #: D200015297 Product: OPERATING SYSTEM 64100 02.00

One-line description:
CDC FLOPPY DRIVE DOESN'T FORMAT CORRECTLY IN A COMMAND FILE.

Problem:
CDC DRIVES DON'T FORMAT CORRECTLY WHEN INITIATED FROM A COMMAND FILE.

Temporary solution:
no temporary solution.

KPR #: D200041178 Product: OPERATING SYSTEM 64100 02.02

One-line description:
Nested macro calls cause incorrect macro expansion.

Problem:
The following code will assemble with no errors on the 64000, but the

KPR #: D200041178 **CONTINUED**

macro expansion is incorrect. When a macro calls another macro, the expansion includes the name of the macro being called on a separate line, then includes the body of the macro itself.

```
"processor name"
DUMMY       MACRO
            NOP
            MEND
MAKRO1      MACRO
            DUMMY
            MEND
VAR1        EQU      40H
MAKRO2      MACRO
            MAKRO1
NULL        .SET      0
            DB       VAR1
            MEND
            MAKRO1
            MAKRO2
```

The call to MAKRO1 will expand as follows:

```
+           DUMMY
+           NOP
```

The call to MAKRO2 will expand as follows:

```
+           MAKRO1
+           DUMMY
+           NOP
+           DB      VAR1
```

Temporary solution:
No temporary solution at this time.

KPR #: D200042036 Product: OPERATING SYSTEM 64100 02.10

One-line description:
ASSEMBLER ISSUES DUPLICATE SYMBOL ERRORS FOR THE 'SEGMENT' PSEUDO

Problem:
Assembling the following file results in duplicate symbol errors for the name SEGMENT EO
.
.
name ENDS
pseudo instruction. The same file assembles without errors on PISCES.

```
ASSUME DS:DATA,SS:DATA
DATA
DATA_SEG        SEGMENT RW
```

The following area is reserved for the stack.

KPR #: D200042036 **CONTINUED**

```
ALIGN
REPT 4        ;If a larger emulation monitor stack is ne
DWS 16        ;change REPT 1 to REPT X.
DWS 15
STACK           DW 0H
TEST0            DW  0000H
TEST1            DW  0000H
TEST2            DW  0000H
TEST3            DW  0000H
TEST4            DW  0000H
TEST5            DW  0000H
TEST6            DW  0000H
TEST7            DW  0000H
TESTB            DB  00H
TESTC            DB  00H
DATA_SEG        ENDS
PROG
PROG_SEG        SEGMENT ER
INIT
START
NOP
MOV AX,#0
MOV DS,AX
MOV ES,AX
NOP
NOP
MOV TEST0,#55H
NOP
JMP START
NOP
NOP
NOP
NOP
NOP
NOP
NOP

SUB
MOV AX,TEST0
NOP
NOP
JMP START

PROG_SEG        ENDS

ORG 0FFFFF0H
RESET_SEG       SEGMENT ER
ASSUME CS:ORG
JMP FAR PTR MON_INIT
RESET_SEG       ENDS

END INIT  DS:DATA_SEG
```

Signed off 08/31/88 in release A02.11

KPR #: D200066308 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Illegal logical expressions are not flagged.

Problem:
Illegal logical operators are not flagged by the assemblers. In most cases the assembler quits when it reaches an illegal expression an does not flag an error. This problem is usually (depending on the processor) confined to EQU statements.

"processor"

```
VALUE          EQU     01001B.or.0110B
VALUE2         EQU     01001B.AND.0110B
VALUE3         EQU     01001B.why.0110B
```

Temporary solution:
There is no known solution at this time.

Signed off 08/31/88 in release A02.11

KPR #: D200069658 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Comment field can not be delimited by whitespace.

Problem:
Whitespace can not be used to delimit comments.

Temporary solution:
Do not use whitespace to delimit comments.

Signed off 08/31/88 in release A02.11

KPR #: D200069989 Product: OPERATING SYSTEM 64100 02.06

Keywords: DC600

One-line description:
store to DC600 causes 64000 to reboot.

Problem:
Store to dc600 causes 64000 to reboot

KPR #: D200074450 Product: OPERATING SYSTEM 64100 02.06

Keywords: DC600

One-line description:
May cause inadvertent overwrite of user's disc.

Problem:
The status line message for the following command:

```
restore from_dc600 <filename>
```

shows the following string:

"restore user's files"

This leads a user to believe that all of the files in the current userid will be restore from the tape.

In reality, all of files on the tape will be restored and will overwrite the disc.

The status line message needs to be changed to reflect this.

Temporary solution:
N.A.

KPR #: D200079368 Product: OPERATING SYSTEM 64100 02.10

Keywords: CODE GENERATOR

One-line description:
Hosted version of GET_ASCII_BYTE strips high order bits of input.

Problem:
The GET_ASCII_BYTE function strips high order bits off of characters on the hosts and does not in Pisces I assembler. The hosted assembler needs to be modified to allow high order bits to be passed.

Temporary solution:
There is no known solution at this time.

Signed off 08/31/88 in release A02.11

KPR #: D200084897 Product: OPERATING SYSTEM 64100 02.07

One-line description:
Recover cmd on 64000(PISCES I) will recover all types on disc's > 150Mb

Problem:
The recover command fails on large disc drives (greater than 150 Mbytes) if no file type is given. This syntax is normally used to recover all possible types of a purged file.

This failure shows in a station hang.

See the Lab Text for more information.
Detailed Listing for Defect Number LSDqf01747
Text:
  recover cmd on 64000(piscesI) will recovering all types on disc's >150 Mb

Large disc drives (greater than 150 Mbytes) will show a defect in the 64000 operating system when a user trys to recover all possible types of a purged file.

```
        Failure syntax:

                            recover X

        Working syntax:

                            recover X:source
```

Defect apparently is in the file manager, and is related to discs with 32 sectors per page as formatted for large disc drives. The failure shows in a station hang. The station will hang AFTER successfully compl
eting
any recovery. The work around, is to specify any file(s) with specific types.

Temporary solution:
The temporary solution to this problem is to specify each file type individually when recovering purged files.

KPR #: D200085043 Product: OPERATING SYSTEM 64100 00.70

One-line description:
Phase error incorrectly reported on 64000 and hosted assemblers.

Problem:
The 1750 assembler reports a spurious error message PH_ERR (Phase error) due to the usage of COUNTER_UPDATE vs. GEN_CODE in passes 1 and 2 of the assembler.

Cause:
This defect has been fixed and this report is being submitted for QA release purposes.

Temporary solution:
None At this Time...

Signed off 08/31/88 in release A02.11

KPR #: D200086694 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Macro use of a label is missing from xref.

Problem:
Another difference between 64000 and 64000-UX assembler found.
The label "LABEL" is missing from the cross reference.

```
   1   "processor"
   2   RMB   MACRO &P1
   3   VALUE .SET  &P1
   4         AND   VALUE
   5         MEND
   6
   7   LABEL EQU   0         <- DEFINITION
```

KPR #: D200086694 **CONTINUED**

```
       8          RMB   LABEL     <- REFERENCE
The X-REF table of this program is
       LINE# SYMBOL    TYPE     REFERENCES
           7 LABEL     A                  <- This field should be 8.
       ***   VALUE     U                  <-                      4.
```

Temporary solution:
No temporary solution.

KPR #: D200086728 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Undefined label not flagged when passed as a parameter to a macro.

Problem:
In the following program an undefined symbol is passed as a parameter to a macro. Rather than flagging an error the assembler assumes it has a value of 0

```
"6301"

         INCLUDE   MACFILE
MAC1      UNDEFLABEL
         END


"MACFILE"

MAC1       MACRO           &P1
           .IF             &P1.EQ.0  NEXT
           .NOP
NEXT       .NOP
           MEND
```

Temporary solution:
No temporary solution at this time.

Signed off 08/31/88 in release A02.11

KPR #: D200087049 Product: OPERATING SYSTEM 64100 02.10

Keywords: CODE GENERATOR

One-line description:
XREF option does not work for instructions that contain a period.

Problem:
On 64000 classic, the xref option does not work for instructions that contain a period, '.'.

Signed off 08/31/88 in release A02.11

KPR #: D200089490 Product: OPERATING SYSTEM 64100 02.10

One-line description:
Erroneaous phase errors generated by Z8000 assembler

Problem:
Phase errors generated by Z8000 assembler on 64100 while assembling the following code:

```
LABEL1:  CP   R0,LABEL2-LABEL1
LABEL2:  NOP
```

Signed off 08/31/88 in release A02.11

KPR #: D200090282 Product: P1750 EMUL 300 64288S004 01.00

One-line description:
Need different monitor names for the F9450 and P1750

Problem:
The P1750 emulator ships its monitor with the same name as the F9450 emulator. This is not a good idea, since an update of only one of these products could break the other product. All installed product file names should be unique.

Signed off 08/19/88 in release A01.10

KPR #: D200090936 Product: P1750 EMUL 300 64288S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

Signed off 08/31/88 in release A01.10

KPR #: 1650028860 Product: PROM PROGRAMMER 300 64501S004 00.00

One-line description:
prommer taking too long to program prom.

Signed off 06/22/87 in release 01.10

KPR #: 5000187617 Product: PROM PROGRAMMER 300 64501S004 01.00

Keywords: SFNONE

One-line description:
8751 does not program in 64000-ux environment.

Signed off 06/22/87 in release 01.10

KPR #: 5000240952 Product: PROM PROGRAMMER 300 64501S004 01.30

One-line description:
PROM programmer has problems in UX envr programming 32 bit system.

Problem:
File expansion to 32 bits causes byte to be dropped.

KPR #: 5000231571 Product: ROM EMULATION 64272 01.04

One-line description:
store command generates 16-bit width absolute file only

Problem:
The store memory command in the 64272 ROM emulator creates an absolute file specifying a bus width of 16 bits, regardless of the emulation configuration. This makes programming EPROMs for systems with 8 bit wide data buses difficult.

The emulator should create the file with the bus width (8 or 16) specified in the emulation configuration.

Temporary solution:
No workaround at this time.

KPR #: 5000267054 Product: RS-232 TRANSFER 300 64885 01.30

One-line description:
Cannot install software on AXE environment machine.

Problem:
The rs232 transfer software cannot be properly installed on an AXE system, because the Programming Environment command "tic" is used to create the file "/usr/lib/terminfo/h/hp64000".

A customer that has an AXE system, does not recieve the proper terminfo file for the hp64000. However, using a TERM type of hp will allow the transfer software to work.

Temporary solution:
Edit the /system/64885/customize file and remove the line accessing the 'tic' command. Then use a term type of hp.

KPR #: D200065219 Product: RS-232 TRANSFER 500 64884 01.10

Keywords: TRANSFER

One-line description:
Transfer hangs after bad options message is displayed.

Problem:
Transfer hangs after bad options message is displayed

KPR #: 5000194951 Product: RS-232 TRANSFER VAX 64886 01.10

Keywords: TRANSFER

One-line description:
Inaccurate specification in HELP for TRANSFER command

Problem:
RS232 transfer on VAX does not work as specified in the VMS help pages. The command syntax shown for a filelist transfer is:

```
    TRANSFER /OPTION(S) <filelist> [<HP64000_file_spec>]
```

which implies that the 64000 filespec is optional. But, if that is omitted, an error message:

```
     ERROR BAD PARAMETERS - NO DESTINATION SPEC
```

is given.

This second parameter should be optional.

KPR #: 5000211375 Product: SOFTKEY EDITOR 64790-90901 01.00

One-line description:
The find command does not work correctly, cannot find string includ '$'.

Problem:
In sk editor mode and when I use command file as follows,the find command does not work correctly.

```
   find "$PAGE$"  all   ----> this change   find "$"  all
```

This means we can not find the string including '$' when using command file.

Temporary solution:
No temporary solution.

KPR #: D200089896 Product: SOFTKEY EDITOR 64790-90901 01.00

One-line description:
AND '$' NEEDS TO BE ESCAPED ON COMMAND LINE TO PREVENT SHELL EXPANSION

Problem:
Text:
any '$' needs to be escaped on command line to prevent shell expansion

If the user wishes to have a command that includes a '$' anywhere in the command, then the user must put a backslash before that '$'. Otherwise, the softkey package does shell variable expansion, which will probably cause the '$' and any text following it to be replaced with either a null string, or whatever the text matches in the shell.

```
ie.  echo $hello ---> echo \$hello
```

KPR #: 1650061580 Product: SOFTKEY EDITOR 300 64790S004 02.00

One-line description:
When 4 retrieves are done, the sk editor jumps to shell.

Problem:
The following command sequence forces sk back to the shell:

```
Make an empty file with only 3069 line feeds (insert line, repeat
3068).
Edit the file, copy one line
retrieve
retrieve
retrieve
retrieve  <<< return to shell
$
Retrieve 4 after 3069 lines doesn't cause any problems , just this
 sequence
```

Also, the following error message is printed:
"Malloc error, unable to continue. Use SK preserve"

Temporary solution:
This problem seems to occur only under these exact conditions. Therefore, the chance of having this problem is minimal. It can, however, be avoided by using "retrieve 4" instead of 4 sequential retrieve commands.

KPR #: 5000205054 Product: SOFTKEY EDITOR 300 64790S004 01.10

One-line description:
Sk may not work when called from pmon.

Problem:
Invoking sk (soft key editor) from within pmon (monitor) may not work if the 64801 Operating Environment is not purchased. The 64801 product contains a file which is required by the sk software.

Signed off 08/28/87 in release 99.99

KPR #: D200090241 Product: SOFTKEY EDITOR 300 64790S004 02.10

One-line description:
When retrieving enough lines to get file exactly 1024 in size; core dump

Problem:
Detailed Listing for Defect Number LSDqf03643
Text:
when retrieving enough lines to get file exactly 1024 in size, core dump

A core dump occurs in the following situation:

```
        enter sk without any filename.
        enter a single line in INSERT mode
        Get out of INSERT mode.
        retrieve the line 1024 times.
        sk goes into infinite loop or core dumps.
```

KPR #: D200090241 **CONTINUED**

There may also be problems with merging files with exactly 1024 lines.

KPR #: D200075028 Product: STATE ANALYZER 64620 00.71

One-line description:
Source referencing will not work with non-zero segments (8086, etc)

KPR #: 5000136135 Product: STATE ANALYZER 64621 01.07

One-line description:
trace file may end up in random userid.

Temporary solution:
Specifying the userid the file shall be created under manually corrects the problem.

Duplicate Service Requests: 5000136069

KPR #: D200091538 Product: STATE ANALYZER 300 64620S004 01.10

One-line description:
File names <8 chars in link_sym will cause translate problems

Problem:
File names in the linker symbol file which have less than 8 characters cause a problem in the translation routines of translate(1) and transfer(1). If this occurs, the user will see several disc transfer error reports from transfer(1) because the malloc(3) buffer space gets trashed in the translation.

Temporary solution:
Be careful to use file names which have at least 8 characers.

KPR #: 5000122374 Product: SW PERF ANALYZER 64310 01.11

One-line description:
"show curr_meas" after measurement change crashes station.

Temporary solution:
Do not show the current measurement after you have changed the measurement setup. This is not very inconvenient, since the user is about to take a new measurement anyway; and presumably knows the results of his last measurement.

KPR #: D200080176 Product: SW PERF ANALYZER 300 64310S004 01.20

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200081026 Product: SW PERF ANALYZER 300 64310S004 01.20

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200082347 Product: SW PERF ANALYZER 300 64310S004 01.20

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can

KPR #: D200082347 **CONTINUED**

```
release the processes by
        cat < ptyxx
This causes the pending output to be flushed, and the processes will die
naturally.
```

KPR #: D200061515 Product: SYSTEM SOFTWARE M 64980-90934 02.00

Keywords: MANUAL

One-line description:
Passing parameters to command files is inconsistient.

Problem:
Parameter passing to the commands file is not consistent. A command file will accept parameters as shown in the following example:

prefix_&parm

but not in the form:

&parm_suffix

The second method will cause a syntax error to be flagged.

Temporary solution:
Avoid using parameters shown in the above example.

Fix information:
Fix is documented in Software Notice 5959-2127 R2707.

Signed off 08/06/87 in release 02.05

KPR #: D200082370 Product: TIMING ANALYZER 300 64610S004 01.60

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: 5000089359 Product: TIMING/STATE 64610 01.00

One-line description:
label cannot be deleted in trigger specification

Temporary solution:
Just change the trigger specification to "trigger on anything" and then delete the label.
****
In order to remove a label from a trigger in this set of circumstances, punch in "trigger on anything" before "trigger received". This will clear the trigger of any labels, and the timing analyzer will then receive the trigger correctly.
****

Duplicate Service Requests: 5000122770

KPR #: 5000089367 Product: TIMING/STATE 64610 01.00

One-line description:
Hitting CLR LINE causes softkeys to return to first level

Temporary solution:
Hit ---ETC--- to get back to the correct level.

KPR #: D200043794 Product: TIMING/STATE 64610 01.00

One-line description:
TIMING PV MODIFICATION TO ALLOW GREATER TIME TOLLERANCE

Problem:
TIMING PV SOFTWARE IS MODIFIED, SO THAT DURATION TESTS HAVE MORE TOLLERANCE FOR VARIOUS CHANGES IN THE SPEED OF THE BPC. TIMING SOFTWARE WILL NOW PASS IN MAINFRAMES CREATED IN MARCH 1986.

Temporary solution:
NONE, OR IGNORE ALL FAILURES ON CONTROL BOARD TEST #4.

KPR #: D200090522 Product: TMS320C25 EMUL FW 64787 00.01

One-line description:
Data words at address 6 & 7 can apparantly be displayed and modified

Problem:
Detailed Listing for Defect Number LSDqf03720

Text:
Data words at address 6 & 7 can apparently be displayed & modified

Temporary solution:
There is no known workaround.

KPR #: D200035261 Product: UPROG 64276 01.00

One-line description:
IN UP_CNTL,"LIST TRACEDATA" SHOWS "AND" EVEN IF NO "ABSOLUTE_IS"

Problem:
In uprog_control context, if list tracedata <LABEL> relative_to <MAP> entered, the softkeys will display and as a valid option (as in and segments or and symbols) even if no asmb_db has been specified with t absolute_is command.

The and segments and and symbols options are only valid if a asmb_db has been specified. Therefore, the and should be suppressed if no asmb_db has been specified.

Temporary solution:
Workaround:
Ignore and option, end command with <RETURN> after specifying map.

KPR #: D200035287 Product: UPROG 64276 01.00

One-line description:
IN UP_CNTL, NO ERRMSG ISSUED IF "RUN UNTIL W/JAM ATTEMPTED W/O JAM LABEL

Problem:
In uprog_control context:
If (1) run until issued without any jam at start;
and (2) on_break action is to jam (rather than stop clocks);
and (3) no default jam label is in effect
then error message No default jam label in effect should be issued
because instrument has no way of determining width of jam desired.

Circumstances causing this defect are very rare because a default jam label is always defined unless the user enters format specification context, deletes the current default jam label, and enters more than one alternative jam labels.

Temporary solution:
Workaround:

Define a default jam label by issuing a run command with some sort of ja start.

KPR #: 5000251322 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Expand Directive not working on 64000.

Problem:
The expand directive is not working in the 64000.

"processor"

```
FOXTROT     MACRO
            LD      B,H
            MEND

            EXPAND
            FOXTROT
            END
```

No expansion is done.

Temporary solution:
Specify expand on the command line rather than in your source.

KPR #: D200068924 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Duplicate Symbols in Symbols Declarations not flagged as an error.

Problem:
Duplicate symbols are not flagged as errors in the UDA definition source.

Example:

```
SYMBOLS = NUMBERS
  ZERO = 0
  ONE  = 1
END

SYMBOLS = TWO_SYM
  ZERO  = 0         <<<Should be flagged as
  ONE   = 1         <<<duplicate symbols
END
```

Temporary solution:
Do not enter duplicate symbol names. Each symbol must have a unique name.

KPR #: D200068932 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Duplicate SYMBOLS Definitions are not flagged as an error

Problem:
Duplicate symbol type names not flagged as an error. Very misleading to user who thinks code assembled correctly.

KPR #: D200068932 **CONTINUED**

Example:

```
SYMBOLS = DSEL_SYM
   INT = 0
   EXT = 1
END

SYMBOLS = DSEL_SYM     <<<Duplicate symbol type
   INT = 0                declaration should be
   EXT = 1                flagged as an error
END
```

Temporary solution:
Do not define duplicate symbol types.

KPR #: D200068940 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Bad table code generated when more than 25 SYMBOLS definitions

Problem:
Only 25 user-defined symbol types allowed. Any more than that are not flagged as an error. Instead, bad code is generated. The last valid SYMBOLS declaration can be detected in the UDA :listing file as follows.

```
00FF  source_line#a   SYMBOLS = NAME1   <<<Last valid Symbols
                                           declaration has 00FF

010F  source_line#b   SYMBOLS = NAME2   <<<First invalid Symbols
                                           declaration has 010F
```

Temporary solution:
Do not define more than 25 User defined symbols types.

KPR #: D200079376 Product: USER DEF ASSEMB 64851 00.70

Keywords: CODE GENERATOR

One-line description:
High order bits stripped from source characters in Pisces I

Problem:
High order bits stripped from source characters in Pisces I.

Temporary solution:
There is no known fix at this time.

KPR #: D200089409 Product: USER DEF ASSEMB 64851 00.70

One-line description:
REPT will only take arguments range 1 thru 32767

KPR #: D200089433 Product: USER DEF ASSEMB 64851 00.70

One-line description:
Page size is different between PI and Hosted assemblers

KPR #: D200089458 Product: USER DEF ASSEMB 64851 00.70

One-line description:
line number only 16-bits in size... This is too small for long files.

KPR #: 5000251348 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
Undefined label not flagged when passed as a parameter to a macro.

Problem:
In the following program an undefined symbol is passed as a parameter to a macro. Rather than flagging an error the assembler assumes it has a value of 0

```
"6301"

         INCLUDE   MACFILE
MAC1      UNDEFLABEL
         END


"MACFILE"

MAC1       MACRO            &P1
           .IF              &P1.EQ.0  NEXT
           .NOP
NEXT       .NOP
           MEND
```

Temporary solution:
No temporary solution at this time.

Signed off 08/31/88 in release A02.20

KPR #: 5000252833 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
Macro use of a label is missing from xref.

Problem:
Another difference between 64000 and 64000-UX assembler found.
The label "LABEL" is missing from the cross reference.

```
      1   "processor"
       2   RMB   MACRO &P1
       3   VALUE .SET  &P1
       4         AND   VALUE
       5         MEND
       6
       7   LABEL EQU   0          <- DEFINITION
       8         RMB   LABEL      <- REFERENCE
The X-REF table of this program is
       LINE# SYMBOL    TYPE     REFERENCES
           7 LABEL     A                <- This field should be 8.
       ***   VALUE     U                <-                      4.
```

Temporary solution:
No temporary solution.

KPR #: 5000252833 **CONTINUED**

Signed off 08/31/88 in release A02.20

KPR #: 5000278606 Product: USER DEF ASSEMB 300 64851S004 02.00

One-line description:
Line number for EQU is not completely dislayed after line 1000

Signed off 08/31/88 in release A02.20

KPR #: 5000294181 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
Undefined Error placed on all macro usage, if just one label undefined

Problem:
The given code does indeed flag P1, P2, and P3 as undefined even though P2 is the only label that is undefined. The SR was submitted for Hotsite Epic #1487. The current version in QA however seems to have fixed this problem while only flagging P2. This pre-released version has been sent to the customer.

KPR #: D200048421 Product: USER DEF ASSEMB 300 64851S004 01.00

Keywords: MACRO

One-line description:
Conditional instr. .IF with rational oper. in Macro creates bad code

Problem:
The use of the conditional instruction, .IF, with rational operator (.EQ.,.NE.,.LT.,.GT.,.LE.,.GE.) in a macro functions incorrectly. The following program demonstrates this problem:

```
       BUG            MACRO           &VAR
                      .IF &VAR .LE. 0 SUB&&&&
                      NOP
                      NOP
       SUB&&&&        NOP
                      NOP
                      MEND

                      BUG  3
                      BUG -1
                      BUG  0
                      END
```

Passing a 3 appears to create correct code, but 0 causes a ML error. Passing -1 to the MACRO creates code which doesn't call the subroutine. This is incorrect since -1 is less than 0. This same problem occured with all the rational operators on all processors. The problem was consistant on the 64000, VAX, and 9000.

Signed off 08/25/86 in release 01.10

KPR #: D200053512 Product: USER DEF ASSEMB 300 64851S004 01.00

One-line description:
Macro def. including .IF, within a IF causes assembler to stop code gen.

Problem:
If you have a ".IF" in a macro definition and that macro definition is within a conditional assembly "IF" then no code is generated. The program provided demonstrates the problem (see submitter text).

Temporary solution:
Pull the macro definition outside of the conditional if. No code will be generated for the definition.

"processor name"

```
ESSAI        EQU     0

MAC          MACRO
             .IF     ESSAI.EQ.0   FIN
LABEL        LD      A,0
FIN          MEND


             IF      ESSAI
             MAC
             ENDIF

START        LD      A,3
```

Signed off 08/25/86 in release 01.10

KPR #: D200055541 Product: USER DEF ASSEMB 300 64851S004 01.00

One-line description:
Comments not delimited by semi-colons appear in the assembler xref.

Problem:
If you do not delimit a comment with a semi-colon it will appear in the assembler xref.

"processor"

```
        MOVE     D0,D1       COMMENT
```

COMMENT appears in the asm xref as an undefined symbol.

Temporary solution:
Delimit all comments with a semi-colon.

Signed off 08/25/86 in release 01.10

KPR #: D200059311 Product: USER DEF ASSEMB 300 64851S004 01.00

One-line description:
Host compilers do not put absolute pats specifications in relocatables

Problem:
Host compilers do not specify the full path name in the relocatable file.

Temporary solution:
No known temporary solution.

Signed off 08/25/86 in release 01.10

KPR #: D200059964 Product: USER DEF ASSEMB 300 64851S004 01.00

One-line description:
QUOTING CHARACTERS WITHIN STRINGS ARE ALL TRANSLATED TO "."

Problem:
When using quoting characters within strings (',",^) they are all translated to "." This was done to facilitate string comparisons but causes a problem when the string is to be part of the generated code.

Signed off 08/25/86 in release 01.10

KPR #: D200065417 Product: USER DEF ASSEMB 300 64851S004 01.10

One-line description:
Conditional assembly for INCLUDE files causes error.

Problem:
The following program has a conditional which causes one of two files to be included. However, if the file that is not included by the conditional is missing, an error is generated when the program is assembled:

```
"processor name"
        DATA
X       RMB             0
        PROG
        IF              X = 0
        INCLUDE         julie1
        ELSE
        INCLUDE         julie2
                        ^ IO - invalid operand error if julie 2 is
                               missing
        ENDIF
```

Temporary solution:
No known workaround at this time.

Signed off 12/10/87 in release Z02.00

KPR #: D200081646 Product: USER DEF ASSEMB 300 64851S004 01.20

One-line description:
expressions of the form 123456.78 cause errors

Problem:
There is a problem with the expression handler on the hosted software when parsing expressions of the form 12345.67, which the assembler thinks is a real number. This problemis shown in sample code supplied to Dave Ritchie by JLO in conjunction with the 64180 assembler.

Signed off 12/10/87 in release Z02.00

KPR #: D200087569 Product: USER DEF ASSEMB 300 64851S004 02.10

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, destroying a pointer to the phase error occurance chain in the MS-DOS version.

Temporary solution:
Do not use forward referencs.

KPR #: D200065391 Product: USER DEF ASSEMB 500 64851S001 01.50

One-line description:
Conditional assembly for INCLUDE files causes error.

Problem:
The following program has a conditional which causes one of two files to be included. However, if the file that is not included by the conditional is missing, an error is generated when the program is assembled:

```
"processor name"
        DATA
X       RMB             0
        PROG
        IF              X = 0
        INCLUDE         julie1
        ELSE
        INCLUDE         julie2
                        ^ IO - invalid operand error if julie 2 is
                                 missing
        ENDIF
```

Temporary solution:
No known workaround at this time.

KPR #: D200086702 Product: USER DEF ASSEMB 500 64851S001 02.10

One-line description:
Macro use of a label is missing from xref.

Problem:
Another difference between 64000 and 64000-UX assembler found. The label "LABEL" is missing from the cross reference.

```
      1   "processor"
       2   RMB   MACRO &P1
       3   VALUE .SET  &P1
       4         AND   VALUE
       5         MEND
       6
       7   LABEL EQU   0          <- DEFINITION
       8         RMB   LABEL      <- REFERENCE
The X-REF table of this program is
       LINE# SYMBOL    TYPE     REFERENCES
           7 LABEL     A                 <- This field should be 8.
       ***   VALUE     U                 <-                      4.
```

Temporary solution:
No temporary solution.

Signed off 08/31/88 in release A02.20

KPR #: D200086736 Product: USER DEF ASSEMB 500 64851S001 02.10

One-line description:
Undefined label not flagged when passed as a parameter to a macro.

Problem:
In the following program an undefined symbol is passed as a parameter to a macro. Rather than flagging an error the assembler assumes it has a value of 0

```
"6301"

         INCLUDE   MACFILE
MAC1      UNDEFLABEL
         END


"MACFILE"

MAC1       MACRO           &P1
           .IF             &P1.EQ.0  NEXT
           .NOP
NEXT       .NOP
           MEND
```

Temporary solution:
No temporary solution at this time.

Signed off 08/31/88 in release A02.20

KPR #: D200087544 Product: USER DEF ASSEMB 500 64851S001 02.10

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, which trashed a pointer to the phase error occurance chain in the MS-DOS version.

KPR #: D200087577 Product: USER DEF ASSEMB DOS 64851S006 00.00

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, destroying a pointer to the phase error occurance chain in the MS-DOS version.

Temporary solution:
Do not use forward reference.

Signed off 03/09/88 in release A01.90

KPR #: D200087726 Product: USER DEF ASSEMB DOS 64851S006 00.00

Keywords: CODE GENERATOR

One-line description:
Do not refer to the serial port config as "emulator config"

Problem:
Don't refer to serial port config as emulator config

The install script for the PC emulator interface installation refers to the serial port comfiguration file as the emulator configuration file. This may cause concern for users familiar with the current emulation products, since the emulation configuration generally refers to the actual setup of the emulator memory, etc.

Temporary solution:
Be aware that the emulator configuration file is the same as the serial port configuration file.

Signed off 08/31/88 in release A02.20

KPR #: D200088492 Product: USER DEF ASSEMB DOS 64851S006 02.10

One-line description:
The "INSTALL.BAT" file has some problems, preventing installation

Problem:
The product floppy has a "bad","INSTALL.BAT" file on it making installation of the product overly difficult.

Temporary solution:
Use the DOS "copy" command to install the product to your system, making the directories the same as found on the product floppy.

DO NOT USE the INSTALL.BAT file...

Signed off 04/08/88 in release A02.11

KPR #: D200090183 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
Linker xref has bad defs & refs - may crash

Problem:
Linker xref does not always work due to a memory allocation error. The symbol table nodes are allocated in chunks of 5 each, but the xref generator frees them on at a time. This results in memory being freed that was not allocated to begin with! On DOS, this can result in nodes being corrupted, or in a system crash. On HPUX it seems to be benign. VAX behavior has not been examined, but this will be repaired there also. The fix is to remove the 'free' statements from the 'symtab.c' module function 'get_next_symb'. The OS will automatically deallocate this process memory upon completion (As it does now if no xref is requested).

Signed off 08/31/88 in release A02.20

KPR #: D200090191 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
MS-DOS linker will not accept .k file as a linker command file

Problem:
The linker on MS-DOS does not recognize a command line input for the command file with a small 'k' extension as a valid command file. (ex: 'test.k' gives err, 'test.K' works)

The reason for this is that the linker doesn't recognize the '.k' as a command file extension, and so appends a '.K' to it to form the default file (eg: 'test.k' becomes 'test.k.K') and so the open fails.

Similar effects are to be expected for '.r', '.l', and '.x' files.

Since MS-DOS is NOT case sensitive, I would recommend modifying the functions that look for these extensions to look for lower or upper case letters (eg: use to_upper() in compare).

Signed off 08/31/88 in release A02.20

KPR #: D200090266 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
REAL number pseudos & SCAN_REAL function no good on MS-DOS

Problem:
Copied from Lab text:

Text:
REAL number pseudos & SCAN_REAL function no good on PC

During porting of 64860 regression tests to Vectra, an error turned up in the use of the REAL pseudo-op and the DCR pseudo for the 8096.

KPR #: D200090266 **CONTINUED**

Invalid values were generated. Example:

```
09B1 F9FE3C00   REAL 1.0E3  ;Should be 447A0000
09B6 FE00003C   DCR  1.0E3  ;Should be 00007A44
```

.fix

NOTE:
Fixes for this problem were put into 'real.c' and 'table.c'. The problem was caused by incorrect type casting and the use of char pointers to short int variables being confused about the index of the most significant and least significant bits of the real number values. This had apparently been encountered before when porting to the VAX and HPUX, as 'ifdef's existed for these processors.

Signed off 08/31/88 in release A02.20

KPR #: D200091272 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
Can not assemble a file on a different disk...(ie: 'A:' from C:)

Problem:
Detailed Listing for Defect Number LSDqf03929
Text:
Can't assemble file on different disk (eg: 'a:') from c:

Found a problem with assembler on DOS where I could not assemble a file on a different disk than the current disk because assembler builds wrong path name for source file.

Example:
current directory is c:\junk and we want to assemble file 'fred' on drive a:

```
C:\JUNK> asm a:fred
```

Assembler cannot open object file because it builds a path for the source file name of: 'C:\JUNK\A:FRED'. This is wrong.

This bug was fixed in the file pathgen.c and will be resolved in the next SUDS release (Jul 88).

This report is for informational purposes only. Please enter in STARS and resolve with code change. Thanks.

KPR #: D200091314 Product: USER DEF ASSEMB DOS 64851S006 02.11

One-line description:
Assembler crashes when directory path name is too long

KPR #: D200065409 Product: USER DEF ASSEMB VAX 64851S003 01.50

One-line description:
Conditional assembly for INCLUDE files causes error.

Problem:
The following program has a conditional which causes one of two files to be included. However, if the file that is not included by the conditional is missing, an error is generated when the program is assembled:

```
"processor name"
        DATA
X       RMB             0
        PROG
        IF              X = 0
        INCLUDE         julie1
        ELSE
        INCLUDE         julie2
                        ^ IO - invalid operand error if julie 2 is
                                    missing
        ENDIF
```

Temporary solution:
No known workaround at this time.

KPR #: D200086710 Product: USER DEF ASSEMB VAX 64851S003 02.10

One-line description:
Macro use of a label is missing from xref.

Problem:
Another difference between 64000 and 64000-UX assembler found.
The label "LABEL" is missing from the cross reference.

```
      1   "processor"
       2   RMB   MACRO &P1
       3   VALUE .SET  &P1
       4         AND   VALUE
       5         MEND
       6
       7   LABEL EQU   0         <- DEFINITION
       8         RMB   LABEL     <- REFERENCE
The X-REF table of this program is
       LINE# SYMBOL    TYPE     REFERENCES
           7 LABEL     A                <- This field should be 8.
       ***   VALUE     U                <-                      4.
```

Temporary solution:
No temporary solution.

Signed off 08/31/88 in release A02.20

KPR #: D200086744 Product: USER DEF ASSEMB VAX 64851S003 02.10

One-line description:
Undefined label not flagged when passed as a parameter to a macro.

Problem:
In the following program an undefined symbol is passed as a parameter to a macro. Rather than flagging an error the assembler assumes it has a value of 0

```
"6301"

         INCLUDE   MACFILE
MAC1      UNDEFLABEL
         END


"MACFILE"

MAC1       MACRO          &P1
           .IF            &P1.EQ.0  NEXT
           .NOP
NEXT       .NOP
           MEND
```

Temporary solution:
No temporary solution at this time.

Signed off 08/31/88 in release A02.20

KPR #: D200087551 Product: USER DEF ASSEMB VAX 64851S003 02.10

One-line description:
DE errors anr not declared in all cases for forward references.

Problem:
This problem was caused by a reference to the array used by the header printing routines. The array used as a buffer is referenced one byte past its end, which destroyed a pointer to the phase error occurance chain in the MS-DOS version.

Temporary solution:
Do not use forward references.

KPR #: 5000232991 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Displaying memory takes much longer in 64000-UX environment than 64000.

Problem:
Migrating from 64100 ude to 64120 ude has had a large time impact on debugging programs. A display of memory sometimes takes 17 seconds! At best it takes about 7 seconds. Much longer than on 64100.

Temporary solution:
No workaround at this time.

Signed off 08/31/88 in release A01.20

KPR #: D200077966 Product: USER DEF EMUL 300 64274S004 01.00

One-line description:
No mnemonic display during stepping of the NEC 7500

Problem:
For some processors, the UDE may not be able to display the opcode and the mnemonic of the instruction during a single step operation. The processor is single stepped correctly and the display of the registers is correct but the instruction being single stepped only gets displayed randomly.

This problem only occurs if the processor is running very slowly, and the single step trace does not have time to complete.

Signed off 08/31/88 in release A01.20

KPR #: D200079004 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
64000-UX UDE mnemonic mem display for word processors may be garbled

Problem:
The mnemonic memory display is garbled for word based processors. An inverse assembler that works properly with the UDE on the 64100 may have this problem under HP64000-UX.

This problem occurs because the Inverse Assembler variable RETURN_COUNT counts words on the 64100, where it now counts bytes under HP64000-UX. HP64000-UX should treat RETURN_COUNT in the same manner as the variable is treated on the 64100.

This problem only occurs on mnemonic memory displays, and can be identified when one notices that the UDE is attempting to disassemble 2nd, 3rd, etc. words of an instruction as the 1st word of the next instruction.

Temporary solution:
If you have the source for your inverse assembler, you can modify your inverse assembler to work around this problem until it is fixed. The workaround is to change the significance of RETURN_COUNT in your Inverse Assembly Language source. Since

KPR #: D200079004 **CONTINUED**

RETURN_COUNT is now expected to always count bytes, you must double all values of RETURN_COUNT. For example, if you initialize RETURN_COUNT to 1, change it so that it is initialized to 2. If you increment RETURN_COUNT, then you must now increment twice.

Signed off 08/31/88 in release A01.20

Duplicate Service Requests: D200078998

KPR #: D200080697 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

Signed off 08/31/88 in release A01.20

KPR #: D200080994 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

Signed off 08/31/88 in release A01.20

KPR #: D200081943 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

Signed off 08/31/88 in release A01.20

KPR #: D200082289 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

Signed off 08/31/88 in release A01.20

KPR #: D200083295 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200083295 **CONTINUED**

Signed off 08/31/88 in release A01.20

KPR #: D200086074 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                             ; end locks the emulation session
3. <system name> <module name>     ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

Signed off 08/31/88 in release A01.20

KPR #: D200087270 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

Signed off 08/31/88 in release A01.20

KPR #: D200090324 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Monitor fails w/word processot if DADA not mapped to low memory

Signed off 08/31/88 in release A01.20

KPR #: D200090621 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
rd/wr ioport at odd addr accesses to many ports for word processors

Problem:
Detailed Listing for Defect Number LSDqf03871
Text:
  rd/wr ioport at odd addr accesses too many ports for word processors

KPR #: D200090621 **CONTINUED**

Defect applies only to UDE word processors:

If a request is made to read or write an I/O port, and its address is odd, the ude processor driver will read 1 too many ports. This can be a real problem if the port immediately following the requested port has undesirable side-effects.

.labnotes

This problem occurs because the UDE aries driver routines that rd/wr i/o ports do not special case word operations with a word processor. The UDE host code needs to send the aries routines an io_command flag that indicates the processor is word oriented.

Temporary solution:
There is no known workaround.

Signed off 08/31/88 in release A01.20

KPR #: D200090910 Product: USER DEF EMUL 300 64274S004 01.10

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200090910 **CONTINUED**

Signed off 08/31/88 in release A01.20

KPR #: 5000152892 Product: USER DEFIN ASM M 64851-90904 00.70

Keywords: MANUAL

One-line description:
'&' is comment field of a macro causes a parameter error.

Problem:
If the comment field in a macro contains a '&', a parameter error is generated.

"processor"

```
MAC1    MACRO    &P1,&P2
        DC.&P1   &P2          ;P1 & P2
        MEND

        MAC1  W,2
```

Temporary solution:
Avoid using a '&' in a macro comment field.

Fix information:
Fix is documented in Software Notice 5959-2131 R2707.

Signed off 08/05/87 in release 01.01

KPR #: 5000153981 Product: USER DEFIN ASM M 64851-90904 01.00

One-line description:
DE must be defined before being referenced.

Problem:
Assembler manual has apparently ambiguous error definition.

Specifically

DE - Indicated symbol must be defined prior to it being referenced.
Symbol may be defined later in program sequence.

The 64000 and 500 seem to agree with the later statement. The series 300 thinks the first one is true.

Temporary solution:
No temporary solution.

KPR #: D200079558 Product: USER DEFIN ASM M 64851-90904 00.70

One-line description:
64000 station resets when linking if SKELETON command used improperly

Problem:
When the SKELETON command is used improperly in the linker, the 64000 station behaves erratically. Symptoms range from resetting of the station to garbage characters displayed when linking the target code.

KPR #: D200079558 **CONTINUED**

Temporary solution:
The SKELETON command should only be used in the linker definition source when a GEN_CODE <RELOC_FMT>,BOTH is used in the assembler definition source. The keyword here is BOTH. If VALUE is used, the SKELETON command in the linker should not be used. A note to this effect will be added to the next revision of the manuals.

KPR #: 5000267468 Product: USER INTERFACE M 64808-90901 01.00

One-line description:
Need to add Note saying that 64100 Terminal Mode is not supported.

KPR #: 5000291427 Product: USER INTERFACE M 64808-90901 01.00

One-line description:
PMON doesn't allow a file to begin with a numeric value.

Problem:
When in PMON on a hpux 9000 s/300, and you try to access a file (editor or emulation), you get an error cannot access file. This occurs if the file begins with a numeric character. If you try to use the same editor on a similar named file while in unix (outside of PMON), you can access the file ok. This was a limitation in the old 64100 system, but is not congruent with the unix naming convention. You can get around this problem if you place an * for the first character followed by a numeric character.

Temporary solution:
You can get around this by placing a '*' for the first character followed by the numeric character.

KPR #: 5000178301 Product: USER INTERFACE 300 64808S004 01.10

One-line description:
PMON not interpreting a command file correctly.

Problem:
In the following example PMON substitutes the value of a shell variable before it executes a command. The substition causes an illegal syntax error because PMON will only allow you to print to | $PRINTER and not | lpr.

```
comp test.c | $PRINTER    is the command file.
```

If executed in PMON a syntax error is flagged. However, if you type this same command PMON accepts it.

Temporary solution:
Invoke the compiler using its full path name.

```
/usr/hp64000/bin/comp  test.c | $PRINTER
```

Signed off 08/20/87 in release 01.30

KPR #: D200090613 Product: USER INTERFACE 300 64808S004 02.10

One-line description:
A command file containing these three characters in that order #'! fails

Problem:
Detailed Listing for Defect Number LSDqf03870
Text:
  command file containing #'! (3 characters, in that order) fails

A command file containing the characters

```
#'!
```

will fail, leaving the command line garbled; subsequent commands in the command file are not executed. The line may have other characters in it, for example

```
#comment - this isn't my idea!
```

which will produce the same results.
The defect was noticed while running regression tests for ptui z80.

Temporary solution:
Be careful not to use that particular combination of characters in a comment line.

KPR #: D200069369 Product: USER INTERFACE 500 64808S001 01.40

One-line description:
PMON not interpreting a command file correctly.

Problem:
In the following example PMON substitutes the value of a shell variable before it executes a command. The substition causes an illegal syntax error because PMON will only allow you to print to | $PRINTER and not | lpr.

```
comp test.c | $PRINTER    is the command file.
```

If executed in PMON a syntax error is flagged. However, if you type this same command PMON accepts it.

Temporary solution:
Invoke the compiler using its full path name.

```
/usr/hp64000/bin/comp  test.c | $PRINTER
```

Signed off 08/20/87 in release 01.30

KPR #: 5000267005 Product: UTILITIES PKG 300 64888S003 01.40

One-line description:
Record attributes on VMS files are incorrect.

Problem:
The xlate utility on the VAX does create Motorola S Record files with the convential "Record Attributes". Normally, the Record Attribute of an S Record file (ASCII file) is "Carriage return carriage control". The xlate utility creates a .H file with "none" as the Record attributes.

This can cause a few problems:

- viewing the file is difficult, because each record is not started on a new line when viewed via the "TYPE" command for example.
- some editors will not accept the file, becuase the one line is too long
- in this customer's case, the Microtec Research 6301 simulator cannot be used because the simultor will not accept the file.

The ASCII files created via xlate should have Record attributes of:
"Carriage return carriage control"

Temporary solution:
- create an fdl file such as "tovarcr.fdl" that looks like:

```
RECORD
     CARRIAGE_CONTROL        carriage_return
     FORMAT                  variable
```

(note that tabs should be used, not spaces, and case should be as shown).
Then issue the command:

```
"convert/fdl=tovarcr <OLDFILE.H> <NEWFILE.H>"
```

The NEWFILE.H will have carriage return carriage control attribuites.

KPR #: D200078048 Product: UTILITIES PKG 300 64888S004 01.00

One-line description:
Enhancements are not displayed correctly in ANSI mode.

Problem:
Detailed Listing for Defect Number LSDqf00423

```
  Submission Number: 00442LSDqf    Date Found:     870622
  Defect Status: OPEN              Date Arrived:   870622
  Prod/SCMS:fastalpha.term         Date Received:  870625
  Version : 0.14                   Date Resolved:          (estimated)
  Severity: 4
  Showstopper: No                  Number of Duplicates:
  Workaround: No                   Additional Files: 1
  Defect/Enhancement:
   * defect
```

Text:
Enhancements are not displayed correctly in ANSI mode.

Submitter Supplied Information

```
  Submitter name:      Blair Pendleton
  Submitter phone:
  Submitter address:   bdp
  Activity used to find defect: casual use
```

Responder Supplied Information

```
  Responsible site:     LSD
  Responsible project:  stars
  Responsible engineer: STARS II
```

.submitter

Enhancements are not updated correctly on ANSI terminals (including vt100, vt220 & vt330). Writing into a previously enhanced field does not always retain the enhancement. I first noticed this in Timing, we have an inverse video title line which names the current display. When I change to a different display, the new display name is printed but the previous enhancements are lost.

To reproduce, put your 2392a terminal into ANSI mode and set your TERM to vt102a. Now run pmon and type in any keyword to make the softkeys track. Examine the softkeys - some are enhanced, others partially enhanced, some not at all. Control-L redraws the display correctly.

Signed off 08/28/87 in release 99.99

KPR #: D200067124 Product: VMS 6500 ASSM M 64843-90904 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200046813 Product: VMS 6800 ASSM M 64841-90907 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8815 R2707.

Signed off 08/05/87 in release 01.50

KPR #: D200067082 Product: VMS 6800 ASSM M 64841-90907 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200055798 Product: VMS 6800-03 C M 64821-90903 01.50

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int    cntr;
  int    (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 RR2707.

Signed off 08/06/87 in release 01.90

KPR #: D200066969 Product: VMS 6800-03 C M 64821-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200055715 Product: VMS 68000/08/10 C M 64819-90904 01.50

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int         func1();
typedef     int       (*pfi)();
pfi         func5();

main() {
  int    cntr;
  int    (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.90

KPR #: D200055715 **CONTINUED**

Duplicate Service Requests: D200055723

KPR #: D200066928 Product: VMS 68000/08/10 C M 64819-90904 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 5000160143 Product: VMS 68000/8/10 ASM M 64845-90906 01.60

One-line description:
Document REG pseudo

Problem:
The REG pseudo is not explained in the manual.

Temporary solution:
The REG pseudo assigns a symbolic name to a register list for future use by the MOVEM instruction.

Example:

```
       MY_LIST   REG     A1-A5
                 MOVEM   MY_LIST,-[A7]
```

Fix information:
Fix is documented in Software Notice 5958-8825 R2707.

Signed off 08/05/87 in release 01.65

KPR #: 5000220764 Product: VMS 68000/8/10 ASM M 64845-90906 01.60

Keywords: MANUAL

One-line description:
Manual explains linker options incorrectly.

Problem:
In all "Using the linker" chapters for VAX hosted manuals the *options definitions are both incorrect and misleading. The /map option states that a load map listing is produced. This is done by default when /output is specified. In fact you cannot specify /map without /output so rather than explaining /map we should explain that /nomap can be used to supress a linker map when /output is specified. Secondly, .LIS is not the default extension as stated in the /output definition. filename.MAP is the default extension for the linker. filename.LIS is used by the assembler and compilers.

* Linker options are typically described on page 3-3 of the Using the linker chapter.

Temporary solution:
No temporary solution.

KPR #: 5000220772 Product: VMS 68000/8/10 ASM M 64845-90906 01.60

Keywords: MANUAL

One-line description:
Manual states incorrectly that EXT is a pseudo op.

KPR #: 5000220772 **CONTINUED**

Problem:
Antoinette Burkett sc:

On page 5-11 of the 68000 cross assembler/linker manual for the vax it states that the either the EXT or EXTERNAL will work as the operator for the external pseudo op, however only EXTERNAL works. EXT generates an error message.

Temporary solution:
Use EXTERNAL in replace of EXT. EXT is an assembler instruction for the 68000 family.

KPR #: D200046268 Product: VMS 68000/8/10 ASM M 64845-90906 01.30

One-line description:
LR error flagged for correct offset using PC+INDEX+OFFSET mode of addr.

Problem:
The following program shows a problem with PC-relative addressing with displacement. The displacement is taken as the low-order 8 bits of the label instead of relative to the current PC.

```
         ORG     0F8H
         MOVE    LABEL[PC,D6],D6
         ORG     102H
LABEL    DC.W    0FFFFH
```

This results in an error message:

LR - Legal Range, Address or displacement out of range of the instruction's addressing capabilitities.

Temporary solution:
Temporary solution:

"68000"

```
        ORG    0FFH
        MOVE   TABLE-($+2)[PC,D0],D1
TABLE   DS     1
```

Duplicate Service Requests: 5000116046 D200045898 5000160754 5000163576 5000270629

KPR #: D200067165 Product: VMS 68000/8/10 ASM M 64845-90906 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these

KPR #: D200067165 **CONTINUED**

files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file
to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200036921 Product: VMS 68000/8/10 P M 64815-90908 01.20

Keywords: TYPE CONVERSION

One-line description:
Signed_8 to Unsigned_16 is incorrect.

Problem:

```
VAR S8 : SIGNED_8;
    US16 : UNSIGNED_16;

BEGIN
US16 := UNSIGNED_16(S8);  This does a sign extend which is incorrect.
```

Temporary solution:
None at this time.

KPR #: D200066860 Product: VMS 68000/8/10 P M 64815-90908 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Makesure that you are in the directory containing the source file
to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200046904 Product: VMS 6805/9/9E ASM M 64844-90906 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8822 R2707.

Signed off 08/06/87 in release 01.70

KPR #: D200067140 Product: VMS 6805/9/9E ASM M 64844-90906 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200066829 Product: VMS 6809 PASCAL M 64813-90905 00.01

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200055830 Product: VMS 6809/09E C M 64822-90903 01.20

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}    .

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200063669 Product: VMS 6809/09E C M 64822-90903 01.00

One-line description:
Clarification of interface for USER_DEFINED and real number routines.

Problem:
In the example below, 6809 libraries cannot be explicitly called. If they are called explicitly as routines, the stack is built differently than when a compiler generated call is made.

Example:

```
"C"
"6809"

$FIX_PARAMETERS ON$

main() {

int x;
double xx;
extern double LONGREAL_FLOAT();
int *px;
int *pxx;
x =5;
px = &x;
pxx = &xx;
LONGREAL_FLOAT(px,pxx);            /* Conversion is not made */
}
```

Temporary solution:
For explicit use of ALL the real number library routines, declare your routines as in the following example with $FIXED_PARAMETERS ON$ and $RECURSIVE OFF$ (Chapter 5 in manual). The compiler will then generate the proper form of parameter passing to satisfy the real number library. Note, $RECURSIVE OFF$ is also necessary when using the USER_DEFINED interface method (Chapter 3 in manual).

Example:

```
"C"
"6809"

extern int xint;
extern double xdouble;
extern int *pxint;
extern int *pxdouble;

extern recursive_variable_func();
$FIXED_PARAMETERS ON$
extern recursive_FIXED_PARM_func();
$RECURSIVE OFF$
extern LONGREAL_FLOAT();
```

KPR #: D200063669 **CONTINUED**

```
/*NOTE do not declare these functions double. It will cause extra
parameters to be passed*/

main() {

$LIST_CODE ON$
  /* NOTE: Parameter passing method for standard C function*/
recursive_variable_func(&xint,&xdouble);
            LDU   #xdouble
            LDY   #xint
            LDD   #00004H
            PSHS  X,Y,U
            LBSR  recursive_varia
            LEAS  000000006H,S
  /* NOTE: Parameter passing method for FIXED_PARAMETER(Pascal) function
*/
recursive_FIXED_PARAM_func(&xint,&xdouble);
            LDU   #xdouble
            LDY   #xint
            LBSR  recursive_FIXED
xint = 5;
            LDD   #00005H
            STD   xint
pxint = &xint;
            LDD   #xint
            STD   pxint
pxdouble = &xdouble;
            LDD   #xdouble
            STD   pxdouble
/* NOTE: Parameter passing method for STANDARD REAL NUMBER LIBRARY funct
ion */
/* A (Pascal) function with $FIXED_PARAMETERS$ and $RECURSIVE OFF$ !*/
LONGREAL_FLOAT(pxint,pxdouble);
            TFR   D,X
            LDD   pxint
            LBSR  LONGREAL_FLOAT
LONGREAL_FLOAT(&xint,&xdouble);
            LDX   #xdouble
            LDD   #xint
            LBSR  LONGREAL_FLOAT

/* Compare the stack build on this assignment call which uses
LONGREAL_FLOAT versus the explicit call above */

xint = xdouble;
            LDX   #xint
            LDD   #xdouble
            LBSR  LONGREAL_TRUNC
xdouble = xint;
            LDX   #xdouble
            LDD   #xint
            LBSR  LONGREAL_FLOAT
}
        Rmain
            GLOBAL   Rmain
```

KPR #: D200063669 **CONTINUED**

```
            RTS
        Dmain
            RMB      0000EH
            GLOBAL   main
        Emain        EQU $-1

            GLOBAL   Emain

            EXTERNAL LONGREAL_TRUNC
            EXTERNAL LONGREAL_FLOAT
```

Fix information:
Fix is documented in Software Notice 5958-6099 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200066985 Product: VMS 6809/09E C M 64822-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200047118 Product: VMS 8051 ASSM M 64855-90904 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8841 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200053793 Product: VMS 8051 ASSM M 64855-90904 01.40

One-line description:
The $ operand does not work as defined.

Problem:
If the $ operand is used in a multi-byte instruction, it should specify the value of the PC at the beginning of that instruction. In the following example, it represents the value of the PC in the middle of the MOV instruction:

```
"8051"
      ORG     10H
      MOV     A,#$        ; moves 11H into A instead of 10H

      END
```

Temporary solution:
Use $-x instead of $ where x represents the offset back to the first byte of the multi-byte instruction:

```
"8051"
      ORG    10H
      MOV    A,#$-1    ; this will move 10H into A
      END
```

Signed off 04/07/88 in release X00.00

KPR #: D200067264 Product: VMS 8051 ASSM M 64855-90904 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble

KPR #: D200067264 **CONTINUED**

is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200067066 Product: VMS 8080/85 ASSM M 64840-90903 01.40

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200055905 Product: VMS 8085 C M 64826-90903 01.60

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor name"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
ODE TRAILER LINE
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int   cntr;
  int   (*tmp)();

        for (cntr=1; cntr<4; cntr++)
          { tmp = func5(cntr);
        }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

KPR #: D200055905 **CONTINUED**

Signed off 08/06/87 in release 01.90

KPR #: D200067041 Product: VMS 8085 C M 64826-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200067025 Product: VMS 8085 PASCAL M 64825-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the souce file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200055673   Product: VMS 8086/88 C   M 64818-90904   03.10

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int        func1();
typedef    int        (*pfi)();
pfi        func5();

main() {
  int    cntr;
  int    (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 03.50

KPR #: D200066902   Product: VMS 8086/88 C   M 64818-90904   03.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compiled command.

Signed off 04/07/88 in release X00.00

KPR #: D200066845 Product: VMS 8086/88 PASCAL M 64814-90905 03.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directoryy containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200067181 Product: VMS 9900 ASSM M 64847-90905 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 5000238543 Product: VMS FILE FORMATS M 64882-90903 01.02

One-line description:
VAX file format manual doesn't give clear explantion of VAX file types.

KPR #: D200053132 Product: VMS FILE FORMATS M 64882-90903 01.01

One-line description:
Linker symbol file format (Chapter 14) Word #6 not defined

Problem:
In chapter 14,. figure 14-5, word #6 is not defined in the manual.

Temporary solution:
This is now documented in the manual 64882-90903 E1086.

Signed off 01/05/88 in release Z01.70

KPR #: D200067280 Product: VMS M-STD1750A ASM M 64857-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 5000167601 Product: VMS SYSTEM INSTAL M 64882-90904 01.00

One-line description:
For VMS=>4 using HP 64000 as VMS terminal need to correct manual (pg8-3)

Problem:

Signed off 12/12/86 in release 01.01

KPR #: 5000195701 Product: VMS SYSTEM INSTAL M 64882-90904 01.60

One-line description:
Number of errors in Appendix A of the manual, please read submit. text.

Signed off 05/06/87 in release 401.70

KPR #: D200055202 Product: VMS SYSTEM INSTAL M 64882-90904 01.03

One-line description:
Need setting for rear panel of old HP 64000 and 64110(with jumper jacks)

Problem:
Apendix B should also include the setting for the rear panel of an old 64100, one with jumper packs. Also an old 64110, one with jumper packs.

Temporary solution:
This information will be added to future revision of the manual (it has never been documented even in the older manuls).

KPR #: D200062844 Product: VMS SYSTEM INSTAL M 64882-90904 01.60

One-line description:
Need instructions to MACRO and link ibdriver for single high speed link

Problem:
The manual does not give instructions to MACRO and link ibdriver except for multiple high speed links. It needs to be done in any case. The customer might not read the section on multiple HSLs if it doesn't apply.

Temporary solution:
Revised manual 64882-90904 E0487 covers single or multiple high speed link installation in chapter 5.

Signed off 01/05/88 in release Z01.70

KPR #: D200067306 Product: VMS TMS 320 ASSM M 64858-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200047027 Product: VMS USER DEFIN ASM M 64851-90907 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8833 R2707.

Signed off 08/06/87 in release 01.60

KPR #: D200064048 Product: VMS USER DEFIN ASM M 64851-90907 01.40

Keywords: MANUAL

One-line description:
'&' is comment field of a macro causes a parameter error.

Problem:
If your comment field in a macro has a '&' in it a parameter error is generated.

"processor"

```
MAC1     MACRO     &P1,&P2
         DC.&P1    &P2          ;P1 & P2
         MEND

         MAC1  W,2
```

If the comment field in a macro contains a '&', a parameter error is generated.

"processor"

```
MAC1     MACRO     &P1,&P2
         DC.&P1    &P2          ;P1 & P2
         MEND

         MAC1  W,2
```

Temporary solution:
Avoid using a '&' in a macro comment field.

Fix information:
Fix is documented in Software Notice 5958-8833 R2707.

KPR #: D200067207 Product: VMS USER DEFIN ASM M 64851-90907 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure tat you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 5000222489 Product: VMS USERS GUIDE M 64882-90902 01.60

One-line description:
Page 3-9 states vt52 emulation using 64100 but does not perform functs.

Temporary solution:
Page 3-9, add Note: VT52 terminal emulation with EDT Does Not apply for VMS 4.0 and higher.

KPR #: D200045492 Product: VMS USERS GUIDE M 64882-90902 01.01

One-line description:
Inconsistent response to ^C,Z,Y among rcmain,transfer, and mapbus.

Problem:
None of the HP programs react well to the normal VAX terminal control commands - CNTRL Z; CNTRL Y; CNTRL C. The programs are not consistant in how they react.

For example if rcmain hangs it is necessary to edit the rcmain file.

The only file to cause real damage was the RCMAIN. I used cntrl Y to exit while connected to the HP. The program left the HP in a busy state that was not cleaned up. A data file had to be edited by hand to correct.

Temporary solution:
No temporary solution.

KPR #: D200046847 Product: VMS Z80/NSC800 ASM M 64842-90905 01.20

One-line description:
Assembler should denote an error on non-absolute .SET expressions.

Problem:
If a nonabsolute expression is used with .SET MACRO instruction, the expression will not assemble correctly.

Temporary solution:
You must use absolute (type=0) expressions with the .SET MACRO instruction.

Fix information:
Fix is documented in Software Notice 5958-8818 R2707.

Signed off 08/05/87 in release 01.60

KPR #: D200067108 Product: VMS Z80/NSC800 ASM M 64842-90905 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200055863 Product: VMS Z80/NSC800 C M 64824-90903 01.50

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.90

KPR #: D200067009 Product: VMS Z80/NSC800 C M 64824-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 5000163295 Product: VMS Z80/NSC800 P M 64823-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before execution the compile command.

Signed off 04/07/88 in release Z00.00

KPR #: D200055756 Product: VMS Z8001/02 C M 64820-90903 01.50

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.90

KPR #: D200066944 Product: VMS Z8001/02 C M 64820-90903 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200067249 Product: VMS Z8001/2 ASSM M 64854-90904 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200066886 Product: VMS Z8001/2 PASCAL M 64816-90905 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Make sure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: D200066803 Product: VMS6800/01/02/03 P M 64811-90904 01.00

One-line description:
.LIS file should be put in same directory as .A and .R files.

Problem:
The .LIS file created with the /OUTPUT option when compiling or assembling is put in the directory where the compile or assemble is invoked from. However, the .A and .R files are put in the directory that the source file resides in. All three of these files should be placed in the same directory.

Temporary solution:
Makesure that you are in the directory containing the source file to be compiled before executing the compile command.

Signed off 04/07/88 in release X00.00

KPR #: 2700005918 Product: Z8 ASSEMB 64850 00.00

One-line description:
Assembler not generating error message when attempt to load label.

Problem:
When an attempt to load a label, which was previously defined as a constant using the EQU pseudo op, is made no error message is generated.

Temporary solution:
No temporary solution at this time.

KPR #: D200091645 Product: Z8 ASSEMB 64850 00.01

One-line description:
Assembler generates Phase Error of forward referenced EQU

Problem:
The assembler generates a Phase Error on forward referenced EQU instruction. The following code is an example program that will produce the phase error:

```
"Z8"

        INC  FRED
FRED    EQU  R15              ;R15 is a register symbol
```

A phase error should not be the only error produced.

Temporary solution:
There is no workaround available.

KPR #: D200085290 Product: Z80 EMUL FW 64753 00.00

One-line description:
In quickbreak mode, the Z80 could break without generic emul's knowledge

Problem:
If "cf qbrk=en", if the monitor finds no command present, it defaults to staying in the monitor. Generic software is not informed of any quickbreak, and therefore thinks emulation is still running user code. Generic software will not read/report breakcauses until it knows that emul is definitely not running user code.

Signed off 02/02/88 in release A00.01

KPR #: D200088047 Product: Z80 EMUL FW 64753 00.01

One-line description:
Emulator may be confused about running/monitor state

Problem:
The emulator may become confused about whether it should be running user code or running in the monitor. This could happen when a command that affects the state of the emulator is in progress when a break comes in. For example, a "reg" command will cause the emulator to temporarily break while the registers are accessed. If a "real" break occurs during the small windows of vulnerability during the temporary break, the emulator will become confused.

The result is that the emulator may be returning to foreground user code prematurely, prior to completing its background task such like displaying registers.

Temporary solution:
This problem can be avoided by selecting "restrict to real time" mode in the emulation configuration (cf rrt=en).

KPR #: D200091256 Product: Z80 EMUL DOS 64753S006 01.00

One-line description:
Invalid COM port in 64700tab file

Problem:
There is an invalid COM port in the 64700ta b file. The second COM port in the 64700tab file should be COM2 instead of COM1.

Temporary solution:
There is no workaround available.

KPR #: 5000139535 Product: Z80 ASSEMB 64842 01.12

One-line description:
Using HEX psuedo is causing bad address calculations.

Problem:

```
"Z80"

TEST1           HEX     16,1A,0E,16
TEST2           HEX     16,1A,0E,00
TEST3           NOP
                NOP
                JP      TEST3
DEMO            DEFW    TEST1
                DEFW    TEST2
                DEFW    TEST3           ;ADDR IS CALCULATED AS 006H
                                        ;WHEN IT SHOULD BE 0008H
```

Temporary solution:
Use the DEFW psuedo instead of the HEX psuedo.

KPR #: 5000152819 Product: Z80 ASSEMB 64842 01.11

One-line description:
Revision number on output listing is incorrect.

Problem:
The revision number printed on the output listing file is incorrect. It is always 1.10 for the Z80 assembler. The correct revision for this product on the 64000 is 1.10; on the 9000 series 500 it is 1.30; on the 9000 series 300 it is 1.00. The revision does not appear on an output listing produced on the VAX.

KPR #: 5000239939 Product: Z80 ASSEMB 64842 01.12

One-line description:
Xref lists symbols which are under False conditional assembly blocks.

Problem:
Using "IF true" "IF failure" instruction on the 64100A system, the Z-80 assembler outputs x-references table from failure routine.

64100AF Rev2.07 64842AF Rev1.12

For example

```
                  1 "Z80"                | 11         IF      FALSE
          <0001>  2 TRUE     EQU    1    | 12 DATA    EQU     1
          <0000>  3 FALSE    EQU    0    | 13         LD      A,DATA
                  4 ;                    | 14         IFEND
                  5          ORG    0    | 15 ;
                  6          IF     TRUE | 16         END
          <0000>  7 DATA     EQU    0    |
0000 3E00         8          LD     A,DATA| LINE# SYMBOL TYPE REFERENCES
                  9          IFEND       |  12  DATA     A   8,13
                 10 ;                    |   3  FALSE    A   11
```

KPR #: 5000239939 **CONTINUED**

```
                                              |   2  TRUE     A  6
The symbol "DATA" is determined at LINE 7 and used only LINE 8.
```

Temporary solution:
No temporary solution at this time.

Duplicate Service Requests: 5000242495

KPR #: 5000264986 Product: Z80 ASSEMB 64842 01.10

One-line description:
Complex macro interaction causing invalid errors.

Problem:
A sample program with a complex macro calling scheme is causing the assembler to generate invalid errors.

Temporary solution:
No temporary solution at this time.

KPR #: D200086686 Product: Z80 ASSEMB 64842 01.12

One-line description:
Difference between 64000 and host in XREF when no symbols.

Problem:
The cross reference tables differ between the 9000 host and pisces I. If there are no symbols the 9000 will generate a header, but, the 64000 will not.

Temporary solution:
No temporary solution at this time.

KPR #: 1650047167 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
HPIB 64120 I/O AND POWER FAILED WHEN MODIFYING TARGET MEMORY

Problem:
VERIFIED THE PROBLEM AS DESCRIBED EXCEPT THE HP-IB 64120 I/O AND POWER FAILURE MESSAGE CAN OCCUR REGARDLESS OF ADDRESS RANGE MODIFIED IN TARGET MEMORY

KPR #: 1650047340 Product: Z80 EMULATION 300 64252S004 00.00

One-line description:
EMULATION SOFTWARE STATUS DOES NOT RECOGNIZE THE "HALT" INSTRUCTION

KPR #: D200069542 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200080655 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080952 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081489 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
PC contents lost over continuation if in break state

Problem:
8085 (64203) and old z80 (64252) emulators do not save the PC value over continuation. This only applies if the emulator is in the break state when ending out of the emulator. On continuation the PC is set to zero. Thus the first run command will start executing at zero if no address is specified, and a registers display will list the Next PC as zero.

KPR #: D200081901 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082230 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
          cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083246 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085332 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
IMPROPER IDENTIFICATION OF THE SECOND Z80 CONTRL CARD IF TWO Z80 PRESENT

KPR #: D200086033 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

KPR #: D200086033 **CONTINUED**

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200090860 Product: Z80 EMULATION 300 64252S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 1650041624 Product: Z80 PASCAL 64812 00.00

Keywords: CODE GENERATOR

One-line description:
$ORG directive can cause incorrect code to be generated.

Problem:
Z80 Pascal compiler generates wrong code when option $ORG $ is switched on.

```
"PASCAL"
"BZ80"
$ASM_FILE$
PROGRAM walesch;
TYPE
  AUF1 = RECORD
    NUMBER  : SIGNED_16;
    DATA_NO : SIGNED_16;
  END;
  AUF = ARRAY[1..100] OF AUF1;
VAR
  $ORG 1000H$
  AUF_INDEX :AUF;
  $END_ORG$
BEGIN
  AUF_INDEX[1].NUMBER := 0;  {This generates wrong code}
  AUF_INDEX[1].DATA_NO :=0;  {LD HL,01000H missing}
END
.
```

Temporary solution:
There is no know work around at this time.

KPR #: D200010363 Product: Z80 PASCAL M 64812-90903 00.00

Keywords: RUN-TIME LIBRARY

One-line description:
Library routine 'Zintabs' (DLIBZ80:CZ80) destroys the accumulator.

Problem:
The library routine 'Zintabs' in the file 'DLIBZ80:CZ80:reloc' destroys the contents of the accumulator. Listed below are the first three assembly statements of this routine. Note the accumulator is not saved.

```
Zintabs    XOR   A
           ADD   A,H
           RET   P
```

Temporary solution:
To save the contents of the accumulator, 'Zintabs' needs to be modified as follows:

```
Zintabs    PUSH  AF
           XOR   A
           ADD   A,H
           POP   AF
           RET   P
```

Fix information:
Fix is documented in Software Notice 5959-2128 R2707.

Signed off 08/05/87 in release 01.01

KPR #: 1650058123 Product: Z80/NSC800 C 64824 02.10

One-line description:
Wrong code generated for assignment operator <<= if used with arrays.

Problem:
In the following sample program, the generated code, after the shift operation, attempts to load the result address from a temporary memory location that has not been set before.

```
"C"
"Z80"
$RECURSIVE-$

int buffer[10], i ;

main()
{
   i = 3 ;
   buffer[i] <<= 2 ;     /* result is not stored properly */
}
```

Temporary solution:
Use the longhand form of the statement. For example, use :

```
 "C"
 "Z80"
 $RECURSIVE-$

 int buffer[10], i ;

 main()
 {
    i = 3 ;
    buffer[i] = buffer[i] << 2 ; /* use expanded notation here */
}
```

Signed off 08/31/88 in release A02.20

KPR #: 5000245704 Product: Z80/NSC800 C 64824 01.04

One-line description:
Error 1006 given for > test condition.

Problem:
The following code causes error 1006 on the 64000.

```
"C"
"Z80"
main(){
 unsigned short a,b;
 if ( (double)a>b ); /* Compiler should generate another error message*/
 }                   /* here!                                        */
```

Temporary solution:
Cast both variables to a double.

KPR #: 5000245704 **CONTINUED**

```
if ( (double) a > (double) b )
```

Signed off 08/31/88 in release A02.20

KPR #: 5000259150 Product: Z80/NSC800 C 64824 01.04

One-line description:
Reference to non-existient library in manual.

Problem:
The Z80/NSC800 C compiler manual incorrectly indicates that a library "ROTATE" can be called from "C". Page 3-20 of the manual 64824-90901 reads: "... by calling the run-time library; ROTATE". There is no such library in Zlibrary:LZ80 or Zreallib:LZ80. The bottom of the page shows an example of calling the rotate function.

The library routine should be added, or the manual updated.

Signed off 08/31/88 in release A02.20

KPR #: D200069906 Product: Z80/NSC800 C 64824 01.03

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

KPR #: D200069906 **CONTINUED**

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function.
For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200081554 Product: Z80/NSC800 C 64824 01.04

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{     if( x )    /* gives "Illegal type of operand(s) */
         ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use      if( x != 0.0 )  ;
```

OR

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200085472 Product: Z80/NSC800 C 64824 02.10

One-line description:
Certain Byte additions with word results may fail in error #1009

Problem:
Certain byte additions having a word result may fail in error #1009 "No free registers". If two bytes are added together and one is left in the A register when the other is to be extended to word size, the compiler may give the #1009 error message. The following code illustrates the problem:

```
"C"
"Z80"
$RECURSIVE OFF$

unsigned int Input_Byte()
{
   unsigned char byte, status;

   return ((status << 8) + byte);
}
```

In this code the temporary value of "status << 8" is stored in the A register. Then an attempt is made to load "byte" in register A prior to extending it into the HL register. This causes the code generator to issue the error message.

Temporary solution:
Try reversing the order of the operands in the addition, as in:

```
   return (byte + (status << 8));
```

Signed off 08/31/88 in release A02.20

KPR #: D200090175 Product: Z80/NSC800 C 64824 02.10

One-line description:
Indirect comparison of parameter bytes may fail

Problem:
Comparisons of indirectly accessed byte parameters may fail. The following code repeatedly compares *right to *right, which is always true, causing the for loop to cycle forever.

```
"C"
"Z80"

func(left,right)
char *left, *right;
{
   for (;*left == *right; left++, right++)
   {
   }
}
```

Temporary solution:

KPR #: D200090175 **CONTINUED**

Use $SHORT_ARITH ON$ or $AMNESIA ON$ compiler directives.

KPR #: D200090217 Product: Z80/NSC800 C 64824 02.10

One-line description:
Certain set operations with explicit type changes may fail.

Problem:
The following code performs arithmetic multiplication and addition, rather than set intersection (AND) and inclusion (OR) operations in the assignments to Bytel.

```
"BZ80"
$EXTENSIONS$
PROGRAM TEST;

TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;

VAR
    Bytel : BYTE;
    I : SIGNED_16;

PROCEDURE ERR_PROC;
BEGIN
   Bytel := BYTE(SET_OF_BITS(ADDR(I)) *
                 SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);

   Bytel := BYTE(SET_OF_BITS(ADDR(I)) +
                 SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);
END;
```

Temporary solution:
Reverse the order of the operands, as in:

```
   Bytel := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] *
                 SET_OF_BITS(ADDR(I)));

   Bytel := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] +
                 SET_OF_BITS(ADDR(I)));
```

KPR #: D200055590 Product: Z80/NSC800 C M 64824-90901 01.02

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

       for (cntr=1; cntr<4; cntr++)
         { tmp = func5(cntr);
       }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.04

KPR #: 5000170191   Product: Z80/NSC800 P   M 64823-90901   01.00

One-line description:
Documentation and examples for Z80 I/O port

Problem:
Customer would manual to provide documentation that would make writing assembly language routines that interface with compiler routines easier. A very common example is the documentration that would be needed for the customer to write his/her own routine that would use the IO port of the Z80. Our compilers do not supply code generation or library routines for that purpose.

Temporary solution:
No temporary solution.

KPR #: 1650049163   Product: Z80/NSC800PASCAL   64823   01.04

One-line description:
Error 1006 for complex statement using mod operator.

Problem:
ERROR 1006 WHEN USING TYPE CONVERSION WITH MODULO OPERATION.

Temporary solution:
Break this statement into two separate statements as follows.

```
VAR I,temp: INTEGER;
    B:  BYTE;

   BEGIN
      temp := SIGNED_16(ADDR(I));
      B := BYTE (I MOD 100H);
   END.
```

Signed off 08/31/88 in release A02.00

KPR #: D200059600   Product: Z80/NSC800PASCAL   64823   01.02

Keywords: PASS 1

One-line description:
$Range ON$ causes incorrect code to be generated for a test operation.

Problem:
The following program when compiled with the $RANGE ON$ option wil cause incorrect code to be generated.

```
"B8085" | "BZ80"
$EXTENSIONS$
$RANGE ON$

PROGRAM  BOOLREAL;

VAR  A,B,C    :   REAL;
     L        :   BOOLEAN;

BEGIN
     A := 10.0;
     B := 15.0;
     C := 12.0;


     L := (C < (B+.5)) AND ((C + .5) > A);
END.
```

The two intermediate results "(C < (B +.5))" and "((C+.5) >A)" are anded together and this result is compared with the value two. Thus the case is never true. With RANGE OFF correct code is generated.

KPR #: D200059600 **CONTINUED**

Temporary solution:
It is necessary to turn $RANGE OFF$ to obtain correct code. Simply breaking up the expression will not work.

KPR #: D200060186 Product: Z80/NSC800PASCAL 64823 01.02

Keywords: PASS 3

One-line description:
Incorrect data offsets in listing file.

Problem:
I am expanding this to all pascal compilers. The C compilers list the correct offset. $FAR ON$ only applies to the 68000 cross compiler. The other compilers exhibit the defect w/o any options on.

```
"processor name"
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:
If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200071357 Product: Z80/NSC800PASCAL 64823 01.03

One-line description:
Register may be incorrectly remembered following byte negate.

Problem:
There are certain situations where a byte negation may cause subsequent operations to fail because a register is remembered to contain a value that it in fact never did contain. The following code illustrates this:

KPR #: D200071357 **CONTINUED**

```
"B8085"
$EXTENSIONS ON$
PROGRAM TEST;
VAR
   S : SIGNED_8 ;
   US : UNSIGNED_8 ;
   S16 : SIGNED_16 ;
   US16 : UNSIGNED_16 ;

BEGIN
   US16 := -US ;
              LDA DTEST+00001H
              CMA
              INR  A
              MVI  H,000H
              MOV  L,A
              SHLD DTEST+00004H
   US := -US ;   { incorrect reuse of register L }
              MOV  A,L
              CMA
              INR  A
              STA  DTEST+00001H
END.
```

This problem does not occur if DEBUG is ON or the declarations of US and US16 are consecutive.

FIX INFO:

This problem is already repaired on the lab system. The fix was to call DELETE_OP(R_A) after the code that complements and increments the A register in the PROCEDURE NEG_. This insures that register L never is tagged as containing the variable US.

Signed off 06/25/87 in release 99.99

KPR #: D200087346 Product: Z80/NSC800PASCAL 64823 01.90

Keywords: CODE GENERATOR PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
PROBLEM ON VAX NOT ON 64100 SYSTEM

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example:

```
"8086"
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
LABEL  100;
BEGIN
```

KPR #: D200087346 **CONTINUED**

```
100:
     GOTO   100;
END;
PROCEDURE  TWO;       { pass 3 error - too many errors in pass 3 }
LABEL  100;           { is generated, without any indication as to }
BEGIN                 { what the problem is                       }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get this error message, be aware that you may have duplicate labels in the program.

Signed off 08/31/88 in release A02.00

KPR #: D200090209 Product: Z80/NSC800PASCAL 64823 01.90

One-line description:
Certain set operations with explicit type changes may fail.

Problem:
The following code performs arithmetic multiplication and addition, rather than set intersection (AND) and inclusion (OR) operations in the assignments to Byte1.

```
"BZ80"
$EXTENSIONS$
PROGRAM TEST;

TYPE
    BITS = (B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,B11,B12,B13,B14,B15);
    SET_OF_BITS = SET OF BITS;

VAR
    Byte1 : BYTE;
    I : SIGNED_16;

PROCEDURE ERR_PROC;
BEGIN
   Byte1 := BYTE(SET_OF_BITS(ADDR(I)) *
              SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);

   Byte1 := BYTE(SET_OF_BITS(ADDR(I)) +
              SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15]);
END;
```

Temporary solution:
Reverse the order of the operands, as in:

```
   Byte1 := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] *
              SET_OF_BITS(ADDR(I)));
```

KPR #: D200090209 **CONTINUED**

```
   Byte1 := BYTE(SET_OF_BITS[B8,B9,B10,B11,B12,B13,B14,B15] +
              SET_OF_BITS(ADDR(I)));
```

KPR #: 1650006544   Product: Z8000 C   64820   00.01

One-line description:
Code generated for unsigned multiply is the same as for signed multiply.

Problem:
Code generated for an unsigned multiplication is the same as for signed multiplication. The following code is an example:

```
"C"
"processor name"
unsigned u1,u2,u3;
int i1,i2,i3;
main()
{
  u3 = u1*u2;        (*LD     R13,Dstatic        *)
                     (*MULT   RR12,Dstatic+00002H*)
                     (*LD     Dstatic+00004H,R13 *)

  i3 = i1*i2;        (*LD     R11,Dstatic+00006H *)
                     (*MULT   RR10,Dstatic+00008H*)
                     (*LD     Dstatic+0000AH,R11 *)
}
```

Temporary solution:
No known temporary solution.

KPR #: 5000160671   Product: Z8000 C   64820   01.06

One-line description:
Logical AND produces a multiply operation.

Problem:

```
"C"
"Z8001"
$SEPARATE ON$
test()
{
             SUB   R15,#00010H
unsigned short *px, *py, *(*psub)();
unsigned long x;
   px = (unsigned long) py & 0x7F00FFFFL;
             LDL   RR12,RR14[#00004H]
             LDL   RR10,#07F00FFFFH
             AND   R12,R10
             AND   R13,R11
             LDL   RR14[#00000H],RR12
/* Multiplication when (and) is called for */
   px = (unsigned long) &x & 0x7F00FFFFL;
             LDA   RR6,RR14[#0000CH]
             MULTL RQ4,#07F00FFFFH
             LDL   RR14[#00000H],RR6
/* Indirect subroutine call */
   px = (*psub)(py,0);
             PUSH  @RR14,#00000H
             LDL   RR12,RR14[#00006H]
             PUSHL @RR14,RR12
```

KPR #: 5000160671   **CONTINUED**

```
             LDL   RR12,RR14[#0000EH]
             CALL  @RR12
             INC   R15,#6
             LDL   RR10,RR2
/* Next line appears to have wrong offset for RR14 */
             LDL   RR14[#0000CH],RR10
             INC   R15,#16
   Rtest
```

Two code generations problems:
1) Logical "and" operation produces a multiply
2) After calling a function indirectly that returns a pointer
   the register offset is incorrect for referencing variables

Temporary solution:
No temporary solution at this time.

KPR #: 5000181545   Product: Z8000 C   64820   01.04

One-line description:
$OPTIMIZE$ compiler directive works differently for signed and unsigned.

Problem:
The $OPTIMIZE$ compiler directive optimizes operations on unsigned types like char, but doesn't optimize operations on signed variables like ints. When this directive is on, the compiler should not reload a register if it has already been loaded with a previous value. The compiler should assume that the value is correct. When this directive is off, then register contents should be forgotten, and reloaded. In the following code example, with $OPTIMIZE ON$, the compiler optimizes the code if the variables are unsigned, but doesn't if they are signed:

```
"C"
"processor name"
$OPTIMIZE ON$
main()
{
  char  *a,*b;     (*if these are declared as int, then no optimization
                         is done*)
  while ((*a=*b) != '\0')
   {
    a++;
    b++;
   }
}
```

Temporary solution:
No known temporary solution at this time.

KPR #: 5000246983   Product: Z8000 C   64820   01.06

One-line description:
Local parms not accessed properly when func called via pointer.

Problem:
In the following program the code generated for the access of func2's local variables is incorrect. This happens only if func2 is called via a pointer and is defined after main.

```
"C"
"Z8001"


int *func2();

main()
{

int i;
int (*func)();


func = func2;
i = (*func)(i);

}

int *func2(p1)
int p1;

{
int j,k,l,m;

      j =1;
      k = 1;
      l = 1;
      m = 1;

/* All of the above references to the local variables will cause
   bad code to be generated.  For example, the reference to j is
   RR14[0004] rather than RR14[0000].                  */
```

Temporary solution:
Define the function and its body prior to making calls to it via a pointer.

```
func2()
{

.
.
.
}

main()
```

KPR #: 5000246983 **CONTINUED**

```
{

int (*func)();
int i;

   func=func2;

  i = (*func)(i);

}
```

KPR #: 5000280958   Product: Z8000 C   64820   02.10

Keywords: PROBLEM ON 9000/S300

One-line description:
Compiler does not create an 'array too large' error when size › 32k.

Problem:
Z8001 C compiler generates 1113 error. When the array area is more than 32k bytes, compiler generates 1113 error. The C compiler on 64100 does not generate that error.

```
 EXAMPLE:
          "C"
          "Z8001"
            char   array[0x8000];
            main()
            {
            }
```

NOTE: according to SR#D200078873 the actual error may exist in
      the 64100 compiler. The Z8001 catches the 'array too large'
      error - the 64100 compiler allows this error to pass unnoticed.

Temporary solution:
There is no workaround available.

KPR #: D200061762   Product: Z8000 C   64820   01.04

One-line description:
Inconsistient error message when linking ASM.R files versus COMP.R files

Problem:
When linking files with DATA addresses defined by more than 16 bits the assembler output causes link errors. If the program below is assembled the linker will flag a legal range error, but, will still generate an absolute file. Flagging the legal range error seems reasonable as the immediate mode of addressing only allows 16 bits for its value (remember the data are must be put in at 10000H or greater for this error to occur).

```
"Z8002"
         DATA
LABEL    RMB        1
```

KPR #: D200061762 **CONTINUED**

```
        PROG
START   PUSH        @R15,#LABEL     ;ONLY 16 BITS ARE ALLOCATED/ALLOWED
                                    ;FOR THE IMMEDIATE VALUE.
```

Now, if the below file is compiled it will generate a similar PUSH instruction is generated. If this file is linked with the data area at 10000H or greater NO error is given. Furthermore, if you turn ASM_FILE on and assemble and link the ASMZ8002 file the legal range error is flagged, but, identical absolutes are generated. In summary, it seems the LR error is appropiate because the immediate mode of addressing only allows 16 bits for its value. If for some reason the LR is inappropriate then the linker should be changed so it is consistient.

```
"C"
"Z8002"

$ASM_FILE ON$
$SEPARATE ON$

float  table[10];

main() {

float  *i;

*i = table[1] + table[2];

}
```

Temporary solution:
No temporary solution at this time.

KPR #: D200064808   Product: Z8000 C   64820   01.05

One-line description:
Superfluous register load in switch statement on the 64000

Problem:
In certain situations, the Z8000 C compiler generates different code on the 64000 than it does on the hosts. The C constructs that cause this to occur are switch statements where the switch value is a four byte quantity accessed via the structure pointer operator. This code difference manifests itself as a superfluous register load that appears to have no adverse effects on the execution of the program. The extra register load occurs only on code compiled on the 64000, not on the hosts.

The following code demonstrates this problem:

```
"C"
"Z8001"
```

KPR #: D200064808 **CONTINUED**

```
struct GROUP {
    char data1;
    long data2;
    char data3;
    int  data4;
    long data5;
    } group_struct = {'A',0x42L,'C',0x44,0x45L};

struct GROUP *grp_ptr = {&group_struct};

main()
{
  switch(grp_ptr->data2)    (*extra code is LDL RR12,RR0 instruction*)
   {
     case 0x42 : break;
     default : break;
    }
}
```

Temporary solution:
No known temporary solution at this time.

KPR #: D200068155   Product: Z8000 C   64820   01.05

One-line description:
Illegal initialization causes error 1113.

Problem:
If you try to initialize a union (illegal per K&R page 198) the compiler does not flag the error. Instead pass three error 1113 is generated (if your target is the 68000, other processors will do the initialization incorrectly.).

```
"C"
"processor"

struct  struct_type {  union  {  int i;
                                 long l; } union_var;
};

static struct struct_type  struct_var = {9,-1};

main() {}
```

The 68000 flags error 1113 and other processor reserve static memory for the structure and try to initialize it. The Z80 initializes three words of memory to 9, -1 and -1.

Temporary solution:
If you get error 1113 check for this illegal construct.

KPR #: D200069781   Product: Z8000 C   64820   01.05

Keywords: PASS 3

One-line description:
Conditional compile fails if it suceeds a fixed parm function call.

Problem:
Conditional compile does not always work properly if you precede the conditional compile with a call to a fixed parameter function.

```
"C"
"processor"

$FIXED_PARAMETERS ON$
extern func1();
$FIXED_PARAMETERS OFF$
#define ibis 0

extern func2();

main()
{
int i;

func1(24);                /* See comment below. */

#if ibis
  func2();
#else if
  i =1;
#endif

}
```

If the fixed parameter function does not have a parameter which is a number I cannot duplicate the problem.

Temporary solution:
Turn $AMNESIA ON$ prior to the call to the fixed parameter function. For efficiency reasons turn $AMNESIA OFF$ after the call.

KPR #: D200078873   Product: Z8000 C   64820   01.06

One-line description:
Oversized data segment not being flagged as an error.

Problem:
Using the z8001 C compiler the compiler does not always flag too large of data area. See the example code below.

```
"C"
"Z8001"

char   array[60000];  /* should cause pass II error "Data
                         segment too large. */
```

KPR #: D200078873 **CONTINUED**

```
main()
{}
```

If you have two arrays of whose sum total size is greater than 32K then the appropriate error message is generated. On the 9000 series of hosts the above program causes a pass III error (program counters disagree.) On the 64100 the file incorrectly reports no errors.

Temporary solution:
No temporary solution.

KPR #: D200079616   Product: Z8000 C   64820   01.06

Keywords: PROBLEM ON 9000/S300

One-line description:
If condition is tested with a CMP D1,D1

Problem:
The following problem will cause a CMP D1,D1 to be generated. This instruction is generated to test an if condition.

```
"C"
"68000"

int dataw,datar;
int *addr;

main()
{
int i,j;

memory_test();
}

memory_test()
{
  long i;

  for (;;) {
      addr = 0x100000;
      for (i=0; i < 0x100000; i++) {
          dataw = (long)addr & 0xffff;
          *aaddr = dataw;
          datar = *addr;

          if (datar != dataw) {
             /* CMP D1,D1 generated here. */
            for(;;);
          }
         addr =addr+1;
```

KPR #: D200079616 **CONTINUED**

```
        }
    }
}
```

Temporary solution:
Turn amnesia on ($AMNESIA ON$) around the function memory test. This will cause slightly more code to be generated.

KPR #: D200081521 Product: Z8000 C 64820 01.06

One-line description:
Real variable used as a test condition cause error.

Problem:
68000 C compiler does not accept a float variable by itself as an expression. Example:

```
float   x;
main()
{      if( x )    /* gives "Illegal type of operand(s) */
          ;
}
```

Customer feels that this variable should be evaluated to see if it is a non-zero float value.

WORKAROUND:

```
Use       if( x != 0.0 )  ;

  OR
```

cast the variable to an int:

```
if ( (int)x);
```

Temporary solution:
Explicitly test the value against zero.

```
"C"
"processor"

main()
{

float i;

if ( i != 0)
;

}
```

KPR #: D200085381 Product: Z8000 C 64820 01.06

One-line description:
Function calls via pointers with parameters mess up subsequent calls.

Problem:
Calls to functions made via a dereferenced pointer where parameters are passed will cause problems in accessing objects on the stack in subsequent functions. The following code illustrates.

```
"C"
"Z8002"

func1()
{
   int (*TEST)();

   (TEST)(11);
}

int func2(parm)
int parm;
{
   return(parm);    /* parm not correctly accessed */
}
```

Temporary solution:
There is no clean solution. You can avoid indirect calls when parameters are involved. Or make sure that the indirect call is the last thing in the source file.

KPR #: 5000123497   Product: Z8000 PASCAL   64816   01.09

One-line description:
Jump table may generate code which accesses wrong data space.

Problem:
The following program will generate a jump table which uses a LD R3,<src> instruction to execute. This instruction causes the status lines to indicate a RAM access when in fact the jump table is in the PROG (ROM) area.

```
"Z8002"
$EXTENSIONS ON$
$SEPARATE ON$

PROGRAM   TEST;

VAR  V    :    INTEGER;

BEGIN
   V  := 10;

   CASE   V   OF

          1:     V:=2;
          2:     V:=3;
          3:     V:=4;
          4:     V:=5;
          5:     V:=6;
          7:     V:=8;
   END;

END.
```

Temporary solution:
No known temporary solution at this time.

KPR #: 5000134916   Product: Z8000 PASCAL   64816   01.10

Keywords: PASS 3

One-line description:
Calling func. twice in statement causes return value to be overwritten

Problem:
In the following program the user defined function "SQR" is called twice in one statement. The result of the first call is overwritten by the results of the second call because RH3 is not saved.

```
"Z8001"
$EXTENSIONS ON$
PROGRAM FUNCTION_CALL;

VAR
```

KPR #: 5000134916   **CONTINUED**

```
$EXTVAR ON$
   RESULT          :    REAL;
   NUMBER1,NUMBER2:     REAL;

$EXTVAR OFF$

FUNCTION SQR( NUM : REAL) : REAL;  EXTERNAL;

PROCEDURE  TESTFUNC;
BEGIN
   RESULT := SQR (NUMBER1) + SQR (NUMBER2);
END.
```

Temporary solution:
Break up the statement into two separate statements with the first call storing the result of "SQR" in a temporary variable.

```
"Z8001"
$EXTENSIONS ON$

PROGRAM FUNCTION_CALL;

VAR
$EXTVAR ON$
RESULT             :REAL;
NUMBER1,NUMBER2    :REAL;
TEMP               :REAL;
$EXTVAR OFF$

FUNCTION  SQR( NUM: REAL ) : REAL; EXTERNAL;

PROCEDURE TESTFUNC;
BEGIN
  TEMP := SQR(NUMBER1);
  RESULT := TEMP + SQR(NUMBER2);
END.
```

KPR #: 5000150151   Product: Z8000 PASCAL   64816   01.04

One-line description:
"Downto" used in a for statement generates incorrect code.

Problem:
Using "downto" in a for statement generates incorrect code. The loop will only be executed once, because the jump condition at the end of the loop jumps on no carry instead of on carry.
The following is an example:

```
"processor name"
PROGRAM TEST;
TYPE
    DAYS = (SUN,MON,TUES,WED,THURS,FRI,SAT);
VAR
    DAY_COUNT:ARRAY[DAYS] of SIGNED_16;
    DAY:DAYS;
```

KPR #: 5000150151 **CONTINUED**

```
BEGIN
    FOR DAY:= SAT DOWNTO SUN DO
        DAY_COUNT[DAY] := 17;      Code generated is:
                                   LDRB   RL7,DTEST+0000EH
                                   SUBB   RH7,RH7
                                   SLL    R7,#1
                                   LD     DTEST[R7],#00011H
                                   LDRB   RL7,DTEST+0000EH
                                   ADDB   RL7,#0FFH
                                   LDRB   DTEST+0000EH,RL7
                                   JR     UGE,TEST00_0
END.
```

The JR UGE,TEST00_0 instruction jumps on no carry instead of on carry, and the loop only executes once.

Temporary solution:
No known temporary solution at this time.

KPR #: D200060145 Product: Z8000 PASCAL 64816 01.10

Keywords: PASS 3

One-line description:
Compiler $FAR ON$, creates incorrect data offsets in listing

Problem:

```
"68000"
$FAR ON$
PROGRAM PROVE;

VAR
  X,Y:INTEGER;
  A: ARRAY[0..99999] OF INTEGER;
BEGIN
$TESTS 1, LIST_CODE ON, LIST_OBJ ON$
(* Comment ON
   Y := A[0];
   Y := A[8000];
   Y := A[9000];
   Comment OFF   *)
   $TESTS 3$
   Y := A[16000];
   Y := A[17000];
   $TESTS 7$
   Y := A[16000];
   Y := A[17000];
   $TESTS 1$
(* Comment ON
   Y := A[32000];
   Y := A[33000];
   Comment OFF   *)
END.
```

Temporary solution:

KPR #: D200060145 **CONTINUED**

If arrays of this size are required download the file to the 64100 and compile.

KPR #: D200061721 Product: Z8000 PASCAL 64816 01.10

One-line description:
Inconsistient error message when linking ASM.R files versus COMP.R files

Problem:
When linking files with DATA addresses defined by more than 16 bits the assembler output causes link errors. If the program below is assembled the linker will flag a legal range error, but, will still generate an absolute file. Flagging the legal range error seems reasonable as the immediate mode of addressing only allows 16 bits for its value (remember the data are must be put in at 10000H or greater for this error to occur).

```
"Z8002"
         DATA
LABEL    RMB          1
         PROG
START    PUSH         @R15,#LABEL       ;ONLY 16 BITS ARE ALLOCATED/ALLOWED
                                        ;FOR THE IMMEDIATE VALUE.
```

Now, if the below file is compiled it will generate a similar PUSH instruction is generated. If this file is linked with the data area at 10000H or greater NO error is given. Furthermore, if you turn ASM_FILE on and assemble and link the ASMZ8002 file the legal range error is flagged, but, identical absolutes are generated. In summary, it seems the LR error is appropiate because the immediate mode of addressing only allows 16 bits for its value. If for some reason the LR is inappropriate then the linker should be changed so it is consistient.

```
"C"
"Z8002"

$ASM_FILE ON$
$SEPARATE ON$

float  table[10];

main() {

float  *i;

*i = table[1] + table[2];

}
```

Temporary solution:
No temporary solution at this time.

KPR #: D200073015 Product: Z8000 PASCAL 64816 01.11

One-line description:
BA address mode may attempt to use RR0 Illegally as source

Temporary solution:
None.

KPR #: D200085282 Product: Z8000 PASCAL 64816 01.12

One-line description:
$RANGE$ & type conversion of UNSIGNED_32 var may cause error 1006.

Problem:
A pass 2 error #1006 on the HP-UX and VMS compilers, or compilation hanging in pass 2 on the HP 64000, may occur under the following conditions:

1) The $RANGE ON$ compiler directive is in effect.
2) An arithmetic operation between two UNSIGNED_32 vars is type converted to SIGNED_32 and used as an array index.
3) The $RANGE OFF$ is then put into effect.
4) There is an array reference inside a conditional.

The following code illustrates:

```
"Z8002"
$EXTENSIONS ON$
PROGRAM ERROR;

PROCEDURE TEST;
VAR
   P,R : UNSIGNED_32;
   ARR : ARRAY [0..5] OF BYTE;
BEGIN
$RANGE ON$
   ARR[SIGNED_32(P+R)] := 95;
$RANGE OFF$
   IF ARR[5] <> 95 THEN ;        (* error 1006 or pass 2 hangs *)
END;
```

KPR #: D200087338 Product: Z8000 PASCAL 64816 01.90

Keywords: CODE GENERATOR   PROBLEM ON 9000/S300 PROBLEM ON 9000/S500
          PROBLEM ON VAX   NOT ON 64100 SYSTEM

One-line description:
"Too many errors pass3" err msg, if use duplicate labels.Need better msg

Problem:
Pascal compiler may generate " too many errors in pass 3 " if two procedures in one module have a label with same name. Example:

```
"8086"
$EXTENSIONS ON$
PROGRAM TOO_MANY;
PROCEDURE  ONE;
```

KPR #: D200087338 **CONTINUED**

```
LABEL  100;
BEGIN
100:
     GOTO   100;
END;
PROCEDURE  TWO;      { pass 3 error - too many errors in pass 3 }
LABEL  100;          { is generated, without any indication as to }
BEGIN                { what the problem is                        }
100:
    GOTO 100
END;
.
```

Temporary solution:
The obvious workaround, is do not use duplicate labels. If you get this error message, be aware that you may have duplicate labels in the program.

Signed off 08/31/88 in release A02.00

KPR #: D200080580 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080887 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081869 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082164 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083170 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085977 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:

KPR #: D200085977 **CONTINUED**

Perform the following steps after executing steps 1-4 listed in the problem text.

5. display trace mnemonic
6. display trace absolute

KPR #: D200086330 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088294 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200088443 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

KPR #: D200090795 Product: Z8001 EMUL 300 64232S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The

KPR #: D200090795 **CONTINUED**

trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: D200055731 Product: Z8001/02 C M 64820-90901 01.04

One-line description:
Declaring a function which returns a ptr to a function causes error.

Problem:
Declaring a function that returns a pointer to a function that returns an integer causes invalid syntax errors to be generated.

```
"C"
"processor"

int  func1();
int  (*func5())();

main () {

int cntr;
int (*tmp)();

   for (cntr=1; cntr<4; cntr++)  {
       tmp=func5(cntr);
   }

func1(){return(1);}
```

Temporary solution:
Break up the declaration by using a typedef.

```
"C"
"processor"

int       func1();
typedef   int       (*pfi)();
pfi       func5();

main() {
  int   cntr;
  int   (*tmp)();

        for (cntr=1; cntr<4; cntr++)
          { tmp = func5(cntr);
          }
}

pfi func5(tmp2)
int tmp2;
{
   if (tmp2==1) return(func1);
}

func1(){return(1);}
```

Fix information:
Fix is documented in Software Notice 5959-2125 R2707.

Signed off 08/06/87 in release 01.06

KPR #: 5000131573 Product: Z8001/2 EMUL M 64980-90923 01.00

One-line description:
Need more info on sharing user system calls & monitor interaction.

Problem:
Customer feels that manual should include more information on the sharing of user system calls with the emulation monitor's system calls. Include a short section which explains how this can be done, including an example of user code which would handle all system calls and fall through to the monitor if appropiate. The customer was able to do this himself but felt it should be explained in the manual.

Also, the chapter "EMULATION CONFIGURATION" subsection "Monitor Interaction" talks of the transparency of the Break system call instruction, including the jamming of the PSA information. Customer felt that this info should be more detailed.

Temporary solution:
See problem text.

KPR #: D200072462 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Incorrect breakpoint behaviour on continuing emulation.

Problem:
When using software breakpoints, and doing the following sequence, emulation does not behave as expected.

```
run LOOP
modify software_breakpoints set LOOP
# breaks into monitor, displays breakpoint and clears breakpoint
run
end  locked
# return to emulation
modify software_breakpoints set LOOP
# breaks into monitor, but does not display or clear breakpoint
# at second iteration, things return to normal.
```

KPR #: D200080598 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080895 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

KPR #: D200080895 **CONTINUED**

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081877 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082172 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
        cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083188 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

KPR #: D200083188 **CONTINUED**

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200085985 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086348 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088302 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal. This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200088450 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Software breakpoint in target memory will hang system.

Problem:
A software breakpoint set in target memory will cause the system to hang.

KPR #: D200090803 Product: Z8002 EMUL 300 64233S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session. For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

KPR #: 5000258616 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
CANNOT ACCESS COMPILER GENERATED SYMBOLS IN HP64000-UX EMUL ENVIRONMENT

Problem:
Compiler generated symbols are specific to the HP64000 software products. While in the HP64000-UX emulation environment, users can access all symbol information in the user program, but can not access these compiler generated symbols for displaying, tracing and modifying purposes. An enhancement investigation may be possible if needs are warranted.

KPR #: D200069559 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Measurement System end_released when terminal cannot be initialized

Problem:
A measurement system will be end_released, resulting in loss of data, when a non-supported terminal is used to enter a currently locked measurement system. This problem will arise if the TERM environment variable is not set to a value that is supported by 64000-UX. This may happen when logging in to a system over a port that is not hard-wired (modem, LAN vt, etc.) where the system prompts you for the Terminal type, which sets the TERM variable.

Temporary solution:
Make sure that the TERM variable is set to a type of terminal that is supported with 64000-UX. This can be verified by typing "echo $TERM".

KPR #: D200080663 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
pwd truncates the /net/system portion of the path when RFA'ed to system.

Problem:
When using the HP 64000-UX products and netunaming across the LAN to another system, such as a compile server, the HP-UX command "pwd" which is used by the HP64000-UX product to tell what the local directory is, truncates the "/net/system" part of the path.

This is a HP-UX operating system defect. It is not a defect in the HP 64000-UX application software. As soon as this defect is fixed in HP-UX, it will work correctly when using the HP 64000-UX applications.

KPR #: D200080960 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Using Emulation across RFA can give incomplete symbol information

Problem:
Accessing symbol data in a remote file across RFA may result in incomplete symbol information being available. This

KPR #: D200080960 **CONTINUED**

problem is a result of read() calls being interrupted during file access over RFA.

This problem can also affect reading absolute data into memory as well.

Temporary solution:
If this problem occurs while loading absolute data, attempting to reload the file again may work.

There are two possible answers to this problem. The first is to move the .Y file to the machine running the emulator.

The second solution is to move the program object to the machine which is running the emulator. This can be done using the get64 program. When you load the emulator, a new .Y file will be created.

KPR #: D200081919 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
The Inter-Module-Bus trigger signal latches when set to drive & receive

Problem:
When two emulators are set to drive and receive trigger, after the trigger signal is driven once over the IMB, the trigger signal latches. Subsequent measurements trigger immediately since the trigger signal on the IMB is still latched from the previous measurement.

Temporary solution:
No workaround at this time.

KPR #: D200082248 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Processes sometimes left running after parent has stopped.

Problem:
Sometimes, when the parent process to a measurement system is killed some of the measurement systems processes are left running. Please change the behaviour of the products so that these processes die nicely.

Temporary solution:
If the tty associated with the process is a pty, then you can release the processes by

```
       cat < ptyxx
```

This causes the pending output to be flushed, and the processes will die naturally.

KPR #: D200083253 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Loading a trace file from a different processor may cause core dump

Problem:
If a trace file is created with "store trace" on a processor that allows multiple analysis modes with some mode other than the default, then is loaded by a processor with only one mode, a core dump will result. A good example is storing an execution mode trace on the dequeued 68000, then trying to load the trace on the non-dequeued 68000.

Temporary solution:
Do not attempt to load a trace file for a mode that is not supported.

KPR #: D200086041 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Tracelist symbols dissappear.

Problem:
The symbols will not be displayed in the trace list if the following commands are executed:

```
1. display trace absolute symbols on
2. end                              ; end locks the emulation session
3. <system name> <module name>      ; continues the emulation session
4. display trace
```

The symbols will not be displayed even if you try to re-execute step number 1.

Temporary solution:
Perform the following steps after executing steps 1-4 listed in the problem text.

```
5. display trace mnemonic
6. display trace absolute
```

KPR #: D200086389 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Using simio, then continuing , may not be possible

Problem:

If simio is used extensively, then the user attempts to end and reenter emulation, reentry will not be possible, and there will be a continuous error message stating that the 64120 will not accept a download. This is a problem with emulation core, and exists in all emulators.

KPR #: D200088344 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
"end" softkey after HP-IB error does not clear command line

Problem:
If there is an HP-IB error and the "end" softkey appears, the command line will not be cleared on any keystroke like it normally is.
In addition there have been instances where NO keystroke would work and the "end" softkey was inoperable, nothing could be typed on the command line. The exit was to kill the process from another terminal.
This problem exists for all HP64000-UX emulators built with /lsd/p2/cmd/emul/gencore.

KPR #: D200090878 Product: Z80H EMULATION 300 64253S004 01.00

One-line description:
Code disp. with trace not right if code changed w/o ending emul. session

Problem:
Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

End out of emulation, and reenter before loading the new program

Source lines displayed with a trace may not be correct if the code is changed without ending out of the emulation session.
For exanple, a user running is windows does a display trace source on, and sees a statement: i = 1; in the trace along with the MOVE.L #1,D0 that accompanies that source line. The user then moves to another window, changes the source line to i = 2;, recompiles, relinks, and runs edbuild. The user then moves back to the emulation window, reloads the file, and reruns the code and the trace. The trace shows MOVE.L #2,D0 as expected, BUT shows i = 1; as the source line.

Temporary solution:
End out of emulation, and reenter before loading the new program or executing the trace.

BIN# 15

#J626	#02749	SSBQ	CSD .STARS	ENVPICA2	THU, SEP 1, 1988, 3:29 PM	U SYS

MESSAGE :

HEWLETT PACKARD – LOGIC SYSTEMS DIVISION

HEWLETT
PACKARD

5958-6018, October 1988
E1088

Printed in U.S.A.