

# DR11-C

DEVICE REGISTER TEST  
MD-11-DDDRA-A

EP-DDDRA-A-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN USA



-LJWU-

.REM !

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDRA-A-D
PRODUCT NAME:	DR11C DEVICE REGISTER TEST
DATE RELEASED:	21 DECEMBER 1975
MAINTAINER:	DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH A LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975 DIGITAL EQUIPMENT CORPORATION

## 1. ABSTRACT

THIS IS A LOGIC TEST OF THE DR11C. FOR THIS TEST TO OPERATE  
A SPECIAL MAINTENANCE CABLE MUST BE CONNECTED (BCOBR).  
THIS TEST WILL CHECK UP TO 32 SEQUENTIAL DR11C'S.

## 2. REQUIREMENTS

## 2.1 EQUIPMENT

POP-11 STANDARD COMPUTER

DR11C

BCOBR FOR EACH DR11C

## 2.2 STORAGE

2.2.1 THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

## 3. LOADING PROCEDURE

## 3.1 METHOD

ABSOLUTE LOADER

## 4. STARTING PROCEDURE

## 4.1 CONTROL SWITCH SETTING

STARTING AT SA 200 ALL SWITCHES SHOULD BE DOWN OR ZERO.  
(IF NOT ZERO, BIT 0 TO 8 WILL BE STARTING VECTOR.)

## 4.2 STARTING ADDRESS OR ADDRESSES

(A) 200 = START OF TEST--FOR NORMAL TESTING  
(B) 204 = SPECIAL ENTRANCE --FOR TESTING UNIQUE DR11C  
(C) 210 = RESTART--FOR STARTING AFTER SHUT DOWN

## 4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY.  
IF THERE IS NO TTY IN THE SYSTEM,  
PATCH THE TAGS STPS AND STPB WITH  
177570  
SET SWITCH REGISTER TO STARTING ADDRESS.  
LOAD ADDRESS.  
PRESS START.  
THE PROGRAM WILL STAY IN SECTION AND LOOP.

## 4.3.1 FOR SPECIAL ENTRANCE - SA204

001

11:53 AM OCT 15 1976 15:53 PAGE 4  
CITY OF CHICAGO

1ST HALT SET SWITCH REGISTER EQUAL TO CSR ADDRESS OF DR11C  
PRESS CONTINUE  
2ND HALT SET SWITCH REGISTER EQUAL TO VECTOR ADDRESS OF DR11C  
PRESS CONTINUE  
RAISE SWITCH ID TO "1" TO INHIBIT SEQUENCING TO NEXT DR11C

## 5. OPERATING PROCEDURE

## 5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 ... THE INSTRUCTION AND LOGIC TEST. WITH ALL SWITCHES DOWN THE PROGRAM WILL PRINT OUT ON ERRORS AND CONTINUE IN TEST. (" / DEVICE ADDRESS VECTOR ADDRESS" WILL BE PRINTED AT COMPLETION OF TESTING EACH DR11C IF SW06 IS DOWN. IF SW06 IS UP THEN NOTHING WILL BE PRINTED. INSTEAD THERE WILL BE "5" ON THE DISPLAY LIGHTS FOR COUPLE OF SECONDS AND THEN PROGRAM WILL HALT AT LOCATION 252. PRESS CONTINUE TO RESTART.)

## 5.1.2 SWITCH SETTINGS ARE

SW15 = 1 OR UP ... HALT ON ERROR  
 SW14 = 1 OR UP ... SCOPE LOOP  
 SW13 = 1 OR UP ... INHIBIT PRINTOUT  
 SW12 = 1 OR UP ... NOT USED  
 SW11 = 1 OR UP ... INHIBIT ITERATION LOOP  
 SW10 = 1 OR UP ... DO NOT ADVANCE TO NEXT DR11C  
 SW09 = 1 OR UP ... INHIBIT PRINTOUT OF DEVICE TESTED  
 SW06 = 1 OR UP ... HALT ON END OF PASS.

## 5.2. SUBROUTINE ABSTRACTS

## 5.2.1 BEGIN SA 200

## 5.2.2 SCOPE

-----  
 THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR.

## 5.2.3 HALT

-----  
 IS A ROUTINE THAT PRINTS-OUT AN ADDRESS THAT TAGS THE FAILING SUBTEST, AND THE INCORRECT DATA AT THE TIME OF THE FAILURE, IF THERE IS A TTY. IT ALSO STORES AWAY THE PC+2 OF THE DETECTED ERROR AND THE PROCESSOR STATUS REGISTER STARTING AT LOCATION 017400.

## .5. OPERATING PROCEDURE CONT'D)

## 5.3 PROGRAM AND/OR OPERATOR ACTION

EITHER A) ONLY IF THERE IS A TTY ---

5.3.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

5.3.2 PATCH THE TAGS \$TPS AND \$TPB  
WITH 177570

5.3.3 LOAD ADDRESS 000200

5.3.4 CLEAR CONSOLE SWITCHES

5.3.5 SET SW15=1, SW11=1, SW07=1, SW06=1

5.3.6 PRESS START

IF THERE IS NO ERROR, PROGRAM WILL RUN  
FOR A FEW SECONDS, AT THE END OF WHICH THERE WILL BE "5"  
ON THE DISPLAY LIGHTS FOR COUPLE OF SECONDS AND THEN  
PROGRAM WILL HALT AT ADDRESS 252 FOR END OF PASS.  
PRESS CONTINUE TO CONTINUE THE PROGRAM.

IF ANY ERROR ENCOUNTERED, PROGRAM WILL HALT AT ADDRESS  
3746.

TO DETERMINE TYPE OF ERROR:

5.3.7 LOAD ADDRESS 017400

5.3.8 PRESS EXAMINE

CONTENT OF THIS LOCATION IS ---PC+2 OF THE DETECTED ERROR

5.3.9 PRESS EXAMINE

CONTENT IS --- PROCESSOR STATUS REGISTER

TO TEST IF THERE IS ANY MORE ERROR IN ANY OTHER TEST:

5.3.10 LOAD ADDRESS 000200

5.3.11 CLEAR CONSOLE SWITCHES

5.3.12 SET SW15=1, SW11=1, SW07=1, SW06=1

5.3.13 PRESS START

PROGRAM WILL HALT FOR THE FIRST ERROR THAT WAS JUST OBSERVED

5.3.14 PRESS CONTINUE

IF THE PROGRAM HALTS AT ADDRESS 3746:

5.3.15 LOAD ADDRESS 017400

5.3.16 PRESS EXAMINE

CONTENT IS THE NEW PC+2 OF THE DETECTED ERROR

NEXT LOCATION CONTAINS PROCESSOR STATUS REGISTER

STEPS 5.3.10 THROUGH 5.3.16 SHOULD BE REPEATED

FOR ANY FURTHER ERROR.

(R B) IF THERE IS A TTY ---

5.3.1 LOAD PROGRAM INTO MEMORY

5.3.2 DO NOT PATCH ANY WHERE

5.3.3 CLEAR CONSOLE SWITCHES

5.3.4 PRESS START

PROGRAM WILL TYPE THE DEVICE ADDRESS FOLLOWED BY THE VECTOR  
ADDRESS.

THEN IF THERE IS NO ERROR AND IF SW06 IS DOWN,

" / DEVICE REGISTER VECTOR ADDRESS" WILL BE

TYPED OUT FOR END OF PASS. THERE WILL BE "5"

ON THE DISPLAY LIGHTS BEFORE THE TYPE OUT.

IF SW06 IS UP PROGRAM WILL HALT AT 252 AND

THERE WILL BE NO TYPE OUT.

IF THERE IS ANY ERROR PROGRAM WILL TYPE OUT:

PC+2 OF DETECTED ERROR PROCESSOR STATUS REGISTER

DEVICE ADDRESS VECTOR ADDRESS. (ALL FOUR IN ONE LINE)

IF SW15 IS UP PROGRAM WILL HALT AT 3746 AFTER TYPE OUT.

PRESS CONTINUE TO CONTINUE THE PROGRAM.

6. ERRORS

6.1 ERROR PRINTOUT

ARE IN A FOUR WORD FORMAT. THE 1ST IS THE PC+2 OF THE  
DETECTED ERROR. THE 2ND IS THE PROCESSOR STATUS  
REGISTER. THE 3RD IS DEVICE ADDRESS. THE 4TH IS  
VECTOR ADDRESS.

5.2 ERROR RECOVERY

DEPRESS CONTINUE TO RESTART SECTION



7. RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

THE DR11C MUST HAVE THE BCOBR CABLE TO RUN THIS TEST.

NOTE THAT THE DR11C HAS FLOATING VECTORS:

THE BELOW IS THE ASSIGNMENT OF FLOATING VECTORS, THE ASSIGNED SEQUENCES ARE:

1. STARTING AT 300 AND WORKING UPWARD ALL DC11'S WILL BE ASSIGNED.
2. THEN ANY EXTRA KL11 CALLED FOR (VTO5, VTO6, LC11)
3. THEN ANY DP11 CALLED FOR.
4. THEN ANY DM11 CALLED FOR.
5. THEN ANY DN11 CALLED FOR.
6. THEN ANY DM11BB CALLED FOR.
7. THEN ANY DR11A CALLED FOR.
8. THEN ANY DR11C CALLED FOR.

THE DR11A AND DR11C DEVICE ADDRESSES WILL BE ASSIGNED IN THE USER AREA OF 767776 TO 764000. THE ASSIGNMENT OF ADDRESSES WILL START AT THE HIGH ADDRESS LIMIT AND PROCEED DOWNWARD. USERS AND SPECIAL SYSTEMS SHOULD START THEIR ASSIGNMENTS OF SPECIAL DEVICES AT THE LOW ADDRESS LIMIT AND WORK UP. AFTER ASSIGNING ALL DR11A'S, ASSIGN DR11C'S

767776 TO 767770	DR11C #0	;ASSUMING NO DR11A'S
767766 TO 767760	DR11C #1	
:		
:		
767706 TO 767700	DR11C #7	
:		
:		
767606 TO 767600	DR11C #15	

## 8. MISCELLANEOUS

WHERE THERE ARE MULTIPLE DR11C OR A SYSTEM AND IT IS DESIRED TO TEST ONLY ONE OF THEM. THIS MAY BE ACHIEVED BY USING THE SPECIAL STARTING ADDRESS AND PLACING SW10 ON A ONE (UP) TO INHIBIT SEQUENCING TO THE NEXT DR11C. SEE 4.3.1.

## 8.1 EXECUTION TIME

FOR EACH DR11C ABOUT 1 MINUTE

## 8.2 UNTESTED LOGIC

SIGNALS TO USER NOT TESTED:  
"NEW DATA READY"  
"DATA TRANSMITTED"  
"INIT" TO THE USER

## 9. PROGRAM DESCRIPTION

THIS PROGRAM WHEN STARTED AT 200 CHECKS THE STANDARD DR11-C'S

THE PROGRAM THEN PERFORMES AN INCREMENTAL LOGIC CHECK FOR THE SELECTED DR11C.

THE DATA REGISTER IS TESTED TO SEE IF "RESET" CLEARS IT, AND IF IT WILL HOLD ALL COMBINATIONS OF NUMBERS.

THE READ/WRITE BITS OF THE STATUS REGISTER ARE ALSO TESTED.

BOTH THE "A" AND "B" INTERRUPTS ARE TESTED TO SEE IF THEY INTERRUPT AT THE CORRECT BUS REQUEST LEVEL BR-5.

AT THE END OF THE TEST AN '/' IS TYPED AND ALSO THE ADDRESSES OF THE DR11-C CONTROL STATUS REGISTER AND IT'S SIDE INTERRUPT VECTOR IS TYPED (IF SELECTED VIA SWITCH 9.). THE PROGRAM THEN RETESTS THE UNIT (IF SELECTED VIA SWITCH 10) OR SCANS TO THE NEXT DR11-C. IF ANOTHER DR11-C IS ON THE SYSTEM THEN THE PROGRAM RESTARTS TESTING THE NEW DR11-C.

AFTER ALL DR11-C'S HAVE BEEN TESTED THE PROGRAM WILL TYPE '/' AND RESTART TESTING WITH THE INITIAL DR11-C.

IF NO ERRORS OCCUR AND THREE DR11-C'S ARE AVAILABLE AND SWITCH 9 IS DOWN THE PROGRAM WILL TYPE.

160000 770 \*  
157770 1000 \*  
157760 1010 \*

/

ETC.

IF SWITCH 9 IS UP THEN

\*  
\*  
\*

K01

.MAIN. MACY11 27(732) 15-OCT-76 15:53 PAGE 11  
DDDRAA.P11

IF A POWER FAIL OCCURS THE PROGRAM WILL RESTART AT "START".

10. LISTING

[

:GENERAL REGISTER LOGIC TEST

PSW=177776  
 HLT=104000  
 SR=177570  
 CSR=167770  
 STKPTR=1200  
 :REGISTER DEFINITIONS

R0=R0  
 R1=R1  
 R2=R2  
 R3=R3  
 R4=R4  
 R5=R5  
 SP=R6  
 PC=R7

:SWITCHES

SW9=1000  
 SW10=2000  
 SW11=4000  
 SW13=20000  
 SW14=40000

419  
 420  
 421  
 422  
 423  
 424  
 425  
 426  
 427  
 428  
 429  
 430  
 431  
 432  
 433  
 434  
 435  
 436  
 437  
 438  
 439  
 440  
 441  
 442  
 443  
 444  
 445  
 446  
 447

.ENABLE ABS,AMA  
 .MCALL \$TYPE,\$STYPOCT,\$STRAP,\$EQUAT,\$POWER  
 .MCALL \$SETUP,\$SETUP,\$SCATCH

.SBTTL BASIC DEFINITIONS

;\*INITIAL ADDRESS OF THE STACK POINTER \*\*\* 1100 \*\*\*

001100

STACK= 1100  
 .EQUIV EMT,ERROR ;BASIC DEFINITION OF ERROR CALL  
 .EQUIV IOT,SCOPE ;BASIC DEFINITION OF SCOPE CALL  
 PS= 177776 ;PROCESSOR STATUS WORD  
 .EQUIV PS,PSW  
 STKLMT= 177774 ;STACK LIMIT REGISTER  
 PIRQ= 177772 ;PROGRAM INTERRUPT REQUEST REGISTER  
 SWR= 177570 ;SWITCH REGISTER  
 DISPLAY=SWR

;\*GENERAL PURPOSE REGISTER DEFINITIONS

000000  
 000001  
 000002  
 000003  
 000004  
 000005  
 000006  
 000007

R0= %0 ;GENERAL REGISTER  
 R1= %1 ;GENERAL REGISTER  
 R2= %2 ;GENERAL REGISTER  
 R3= %3 ;GENERAL REGISTER  
 R4= %4 ;GENERAL REGISTER  
 R5= %5 ;GENERAL REGISTER  
 R6= %6 ;GENERAL REGISTER  
 R7= %7 ;GENERAL REGISTER  
 .EQUIV R6,SP ;STACK POINTER  
 .EQUIV R7,PC ;PROGRAM COUNTER

448  
449 100000  
450 040000  
451 020000  
452 010000  
453 004000  
454 002000  
455 001000  
456 000400  
457 000200  
458 000100  
459 000040  
460 000020  
461 000010  
462 000004  
463 000002  
464 000001  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477 100000  
478 040000  
479 020000  
480 010000  
481 004000  
482 002000  
483 001000  
484 000400  
485 000200  
486 000100  
487 000040  
488 000020  
489 000010  
490 000004  
491 000002  
492 000001  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503

;"SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000  
SW14= 40000  
SW13= 20000  
SW12= 10000  
SW11= 4000  
SW10= 2000  
SW09= 1000  
SW08= 400  
SW07= 200  
SW06= 100  
SW05= 40  
SW04= 20  
SW03= 10  
SW02= 4  
SW01= 2  
SW00= 1  
.EQUIV SW09,SW9  
.EQUIV SW08,SW8  
.EQUIV SW07,SW7  
.EQUIV SW06,SW6  
.EQUIV SW05,SW5  
.EQUIV SW04,SW4  
.EQUIV SW03,SW3  
.EQUIV SW02,SW2  
.EQUIV SW01,SW1  
.EQUIV SW00,SW0

;"DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000  
BIT14= 40000  
BIT13= 20000  
BIT12= 10000  
BIT11= 4000  
BIT10= 2000  
BIT09= 1000  
BIT08= 400  
BIT07= 200  
BIT06= 100  
BIT05= 40  
BIT04= 20  
BIT03= 10  
BIT02= 4  
BIT01= 2  
BIT00= 1  
.EQUIV BIT09,BIT9  
.EQUIV BIT08,BIT8  
.EQUIV BIT07,BIT7  
.EQUIV BIT06,BIT6  
.EQUIV BIT05,BIT5  
.EQUIV BIT04,BIT4  
.EQUIV BIT03,BIT3  
.EQUIV BIT02,BIT2  
.EQUIV BIT01,BIT1  
.EQUIV BIT00,BIT0

504			.*BASIC "CPU" TRAP VECTOR ADDRESSES
505		000004	ERRVEC= 4 ; TIME OUT AND OTHER ERRORS
506		000010	RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS
507		000014	TBITVEC= 14 ; "T" BIT
508		000014	TRTVEC= 14 ; TRACE TRAP
509		000014	BPTVEC= 14 ; BREAKPOINT TRAP (BPT)
510		000020	IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
511		000024	PWRVEC= 24 ; POWER FAIL
512		000030	EMTVEC= 30 ; EMULATOR TRAP (EMT) **ERROR**
513		000034	TRAPVEC= 34 ; "TRAP" TRAP
514		000060	TKVEC= 60 ; TTY KEYBOARD VECTOR
515		000064	TPVEC= 64 ; TTY PRINTER VECTOR
516		000240	PIRQVEC= 240 ; PROGRAM INTERRUPT REQUEST VECTOR
517		000000	\$TN=0
518		000000	\$\$WR=0
519			
520		000000	. =0
521	000000	000002	.+2
522	000002	000000	HALT
523	000004	000006	.+2
524	000006	000000	HALT
525	000010	000012	.+2
526	000012	000000	HALT
527	000014	000016	.+2
528	000016	000000	HALT
529	000020	000022	.+2
530	000022	000000	HALT
531	000024	000026	.+2
532	000026	000000	HALT
533	000030	000032	.+2
534	000032	000000	HALT
535	000034	000036	.+2
536	000036	000000	HALT
537	000040	000042	.+2
538	000042	000000	HALT
539	000044	000046	.+2
540	000046	000000	HALT
541	000050	000052	.+2
542	000052	000000	HALT
543	000054	000056	.+2
544	000056	000000	HALT
545	000060	000062	.+2
546	000062	000000	HALT
547	000064	000066	.+2
548	000066	000000	HALT
549	000070	000072	.+2
550	000072	000000	HALT
551	000074	000076	.+2
552	000076	000000	HALT
553	000100	000102	.+2
554	000102	000000	HALT
555	000104	000106	.+2
556	000106	000000	HALT
557	000110	000112	.+2
558	000112	000000	HALT
559	000114	000116	.+2

560	000116	000000	I
561	000120	000122	I
562	000122	000000	I
563	000124	000126	I
564	000126	000000	I
565	000130	000132	I
566	000132	000000	I
567	000134	000136	I
568	000136	000000	I
569	000140	000142	I
570	000142	000000	I
571	000144	000146	I
572	000146	000000	I
573	000150	000152	I
574	000152	000000	I
575	000154	000156	I
576	000156	000000	I
577	000160	000162	I
578	000162	000000	I
579	000164	000166	I
580	000166	000000	I
581	000170	000172	I
582	000172	000000	I
583	000174	000176	I
584	000176	000000	I
585	000200	000202	I
586	000202	000000	I
587	000204	000206	I
588	000206	000000	I
589	000210	000212	I
590	000212	000000	I
591	000214	000216	I
592	000216	000000	I
593	000220	000222	I
594	000222	000000	I
595	000224	000226	I
596	000226	000000	I
597	000230	000232	I
598	000232	000000	I
599	000234	000236	I
600	000236	000000	I
601	000240	000242	I
602	000242	000000	I
603	000244	000246	I
604	000246	000000	I
605	000250	000252	I
606	000252	000000	I
607	000254	000256	I
608	000256	000000	I
609	000260	000262	I
610	000262	000000	I
611	000264	000266	I
612	000266	000000	I
613	000270	000272	I
614	000272	000000	I
615	000274	000276	I

MACY: 27,732) 15-OCT-76 15:53 PAGE 16  
 00000.P11 BASIC DEFINITIONS

616	000276	000000	HALT
617	000300	000302	.+2
618	000302	000000	HALT
619	000304	000306	.+2
620	000306	000000	HALT
621	000310	000312	.+2
622	000312	000000	HALT
623	000314	000316	.+2
624	000316	000000	HALT
625	000320	000322	.+2
626	000322	000000	HALT
627	000324	000326	.+2
628	000326	000000	HALT
629	000330	000332	.+2
630	000332	000000	HALT
631	000334	000336	.+2
632	000336	000000	HALT
633	000340	000342	.+2
634	000342	000000	HALT
635	000344	000346	.+2
636	000346	000000	HALT
637	000350	000352	.+2
638	000352	000000	HALT
639	000354	000356	.+2
640	000356	000000	HALT
641	000360	000362	.+2
642	000362	000000	HALT
643	000364	000366	.+2
644	000366	000000	HALT
645	000370	000372	.+2
646	000372	000000	HALT
647	000374	000376	.+2
648	000376	000000	HALT
649	000400	000402	.+2
650	000402	000000	HALT
651	000404	000406	.+2
652	000406	000000	HALT
653	000410	000412	.+2
654	000412	000000	HALT
655	000414	000416	.+2
656	000416	000000	HALT
657	000420	000422	.+2
658	000422	000000	HALT
659	000424	000426	.+2
660	000426	000000	HALT
661	000430	000432	.+2
662	000432	000000	HALT
663	000434	000436	.+2
664	000436	000000	HALT
665	000440	000442	.+2
666	000442	000000	HALT
667	000444	000446	.+2
668	000446	000000	HALT
669	000450	000452	.+2
670	000452	000000	HALT
671	000454	000456	.+2



MAIN. MACY11 27.732) 15-OCT-76 15:53 PAGE 17  
 000RAA.P11 BASIC DEFINITIONS

672	000456	000000	HALT
673	000460	000462	.+2
674	000462	000000	HALT
675	000464	000466	.+2
676	000466	000000	HALT
677	000470	000472	.+2
678	000472	000000	HALT
679	000474	000476	.+2
680	000476	000000	HALT
681	000500	000502	.+2
682	000502	000000	HALT
683	000504	000506	.+2
684	000506	000000	HALT
685	000510	000512	.+2
686	000512	000000	HALT
687	000514	000516	.+2
688	000516	000000	HALT
689	000520	000522	.+2
690	000522	000000	HALT
691	000524	000526	.+2
692	000526	000000	HALT
693	000530	000532	.+2
694	000532	000000	HALT
695	000534	000536	.+2
696	000536	000000	HALT
697	000540	000542	.+2
698	000542	000000	HALT
699	000544	000546	.+2
700	000546	000000	HALT
701	000550	000552	.+2
702	000552	000000	HALT
703	000554	000556	.+2
704	000556	000000	HALT
705	000560	000562	.+2
706	000562	000000	HALT
707	000564	000566	.+2
708	000566	000000	HALT
709	000570	000572	.+2
710	000572	000000	HALT
711	000574	000576	.+2
712	000576	000000	HALT
713	000600	000602	.+2
714	000602	000000	HALT
715	000604	000606	.+2
716	000606	000000	HALT
717	000610	000612	.+2
718	000612	000000	HALT
719	000614	000616	.+2
720	000616	000000	HALT
721	000620	000622	.+2
722	000622	000000	HALT
723	000624	000626	.+2
724	000626	000000	HALT
725	000630	000632	.+2
726	000632	000000	HALT
727	000634	000636	.+2

MAIN. MACY11 27,732) 15-OCT-76 15:53 PAGE 18  
 000000.P11 BASIC DEFINITIONS

728	000636	000000	HALT
729	000640	000642	.+2
730	000642	000000	HALT
731	000644	000646	.+2
732	000646	000000	HALT
733	000650	000652	.+2
734	000652	000000	HALT
735	000654	000656	.+2
736	000656	000000	HALT
737	000660	000662	.+2
738	000662	000000	HALT
739	000664	000666	.+2
740	000666	000000	HALT
741	000670	000672	.+2
742	000672	000000	HALT
743	000674	000676	.+2
744	000676	000000	HALT
745	000700	000702	.+2
746	000702	000000	HALT
747	000704	000706	.+2
748	000706	000000	HALT
749	000710	000712	.+2
750	000712	000000	HALT
751	000714	000716	.+2
752	000716	000000	HALT
753	000720	000722	.+2
754	000722	000000	HALT
755	000724	000726	.+2
756	000726	000000	HALT
757	000730	000732	.+2
758	000732	000000	HALT
759	000734	000736	.+2
760	000736	000000	HALT
761	000740	000742	.+2
762	000742	000000	HALT
763	000744	000746	.+2
764	000746	000000	HALT
765	000750	000752	.+2
766	000752	000000	HALT
767	000754	000756	.+2
768	000756	000000	HALT
769	000760	000762	.+2
770	000762	000000	HALT
771	000764	000766	.+2
772	000766	000000	HALT
773	000770	000772	.+2
774	000772	000000	HALT
775	000774	000776	.+2
776	000776	000000	HALT
777		000020	.=20
778	000020	004264	.SCOPE
779	000022	000340	340
780	000024	004524	PFAIL
781	000026	000340	340
782	000030	003626	.HLT
783	000032	000340	340

784	000034	005462			\$TRAP		
785	000036	000340			340		
786		000046			. =46		
787	000046	003610			LOPIC		
788		000200			. =200		
789	000200	000137	001242		JMP	@#START1	: INITIAL START
790	000204	000137	001256		JMP	@#SPEC	: TO SELECT UNIQUE ADDRESS AND VECTOR
791	000210	000137	001366		JMP	@#START	: RESTART
792		000250			. =250		
793	000250	000000			EOPHLT: HALT		: THIS IS AN END OF PASS HALT
794							: NOT AN ERROR HALT.
795							: PRESS CONTINUE TO CONTINUE THE
796							: PROGRAM.
797							
798	000252	000207			RTS	PC	
799		001200			. =1200		
800							
801							; THIS TABLE CONTAINS INITIAL REGISTER AND VECTOR ADDRESSES
802							
803	001200	167770			RCSR: CSR		
804	001202	167772			CSR+2		
805	001204	167774			CSR+4		
806	001206	167773			CSR+3		
807	001210	000300			RCSR1: 300		
808	001212	000302			302		
809	001214	000304			304		
810							
811							; THIS TABLE CONTAINS REGISTER AND VECTOR ADDRESSES OF THE DR11-C UNDER TEST
812							
813	001216	167770			DRCSR: 167770		: ADDRESS OF DR11-C STATUS REGISTER
814	001220	167772			DROBUF: 167772		: ADDRESS OF DR OUTPUT BUFFER REG.
815	001222	167774			DRISUF: 167774		: ADDRESS OF DR INPUT BUFFER REG.
816	001224	167773			DRBH10: 167773		: HIGH BYTE OF OUTPUT BUFFER REG.
817							
818	001226	000300			DRVECA: 300		: INTERRUPT VECTOR OF UNIT UNDER TEST
819	001230	000302			DRLVL: 302		
820	001232	000304			DRVECB: 304		: INTERRUPT VECTOR
821	001234	000000			XORFLG: 0		
822							
823	001236	000000			COUNT: 0		: COUNT LOCATION
824	001240	000240			PL: 240		: PRIORITY LEVEL
825							
826	001242	012706	001200		START1: MOV	#STKPTR,R6	
827	001246	004737	001272		JSR	PC,FIRST	
828	001252	000137	001366		JMP	@#START	
829	001256	012706	001200		SPEC: MOV	#STKPTR,R6	
830	001262	004737	001272		JSR	PC,FIRST	
831	001266	000137	004612		JMP	SPEC0	
832	001272	013746	000004		FIRST: MOV	@#4, -(R6)	
833	001276	012737	001352	000004	MOV	#XORA,@#4	
834	001304	012737	000031	177060	MOV	#31,@#177060	
835	001312	012637	000004		MOV	(R6)+,@#4	
836	001316	012737	177777	001234	MOV	#-1,@#XORFLG	
837	001324	012701	160000		MOV	#160000,R1	
838	001330	004737	004642		JSR	PC,@#SPEC1	
839	001334	012701	000770		MOV	#770,R1	

```

940 001340 004737 004672      JSR      PC, @#SPEC2
941 001344 104400              TYPE
942 001346 004712              MESS1
943 001350 000207              RTS      PC
944 001352 022626              XORA:   CMP      (R6)+, (R6)+
945 001354 012637 000004      MOV      (R6)+, @#4
946 001360 005037 001234      CLR      @#XORFLG
947 001364 000207              RTS      PC
948
949 001366 012700 001200      ; INITIALIZE ADDRESS AND VECTORS
950 001372 012701 001216      START:  MOV      #RCSR, R0      ;GET ADDRESS OF FIRST POSSIBLE DR11-C'S
951 001376 012021              MOV      #DRCSR, R1
952 001400 012021              MOV      (R0)+, (R1)+      ;LOAD INITIAL TEST ADDRESSES
953 001402 012021              MOV      (R0)+, (R1)+
954 001404 012021              MOV      (R0)+, (R1)+
955 001406 012021              MOV      (R0)+, (R1)+
956 001410 012021              MOV      (R0)+, (R1)+
957 001412 012021              MOV      (R0)+, (R1)+
958 001414 012706 001200      RSTART: MOV      #STKPTR, R6      ;SET UP STACK
959 001420 012737 001446 004372  MOV      #BEGIN, RETURN      ;SET SCOPE RETURN
960 001426 005037 004370      CLR      @#SCOPEF
961
962      ; DOES RESET CLEAR REGISTER?
963 001432 032737 001000 177570  BIT      #SW9, @#SR
964 001440 001002              BNE      BEGIN
965 001442 004737 004410      JSR      PC, @#MOREID
966 001446 013705 001216      BEGIN:  MOV      DRCSR, R5      ;GET ADDRESS OF STATUS REGISTER
967 001452 012777 000240 176316  MOV      #240, @PSW      ;SET PRIORITY LEVEL 6
968 001460 012737 001516 000004  MOV      #15, @#4      ;SET TIME OUT TRAP VECTOR
969 001466 012737 000010 004366  MOV      #10, ICOUNT
970 001474 012777 177777 177516  MOV      #-1, @DROBUF      ;PRESET OUTPUT BUFFER
971 001502 000005              RESET      ;CLEAR DATA REGISTER
972 001504 017700 177510      MOV      @DROBUF, R0      ;GET RESULT OF RESET
973 001510 001403              BEQ      2$
974 001512 104000              HLT
975 001514 000401              BR       2$      ;DATA REGISTER NOT CLEAR
976 001516 104000      1$:      HLT      ;ERROR! TIMED OUT WHEN REFERENCING DROBUF.
977 001520 012706 001200      2$:      MOV      #STKPTR, SP      ;RESET STACK POINTER
978 001524 012737 000006 000004  MOV      #6, @#4      ;RESTORE TIME OUT TRAP
979
980      SCOPE
981 001532 000004              MOV      #4000, ICOUNT
982 001534 012737 004000 004366  MOV      #-1, @DROBUF      ;ALL ONES TO REGISTER
983 001542 012777 177777 177450  MOV      @DROBUF, R0
984 001550 017700 177444      MOV      #-1, R0
985 001554 022700 177777      CMP
986 001560 001401              BEQ      .+4
987 001562 104000              HLT      ;REG WILL NOT HOLD ONES
988
989      SCOPE
990 001564 000004              MOV      #10, ICOUNT
991 001566 012737 000010 004366  MOV      #-1, @DROBUF
992 001574 012777 177777 177416  MOV
993 001602 000005              RESET      ;SET DATA TO ALL ONES
994 001604 005777 177412      TST      @DRIBUF      ;SHOULD CLEAR REGISTER
995 001610 001401              BEQ      .+4
996 001612 104000              HLT      ;REG FAILED TO CLEAR

```

```

896 001614 000004          SCOPE
897 001616 012737 004000 004366  MOV      #4000,ICOUNT
898 001624 012777 052525 177366  MOV      #52525,@DROBUF
899 001632 017700 177362          MOV      @DROBUF,R0
900 001636 022700 052525          CMP      #52525,R0
901 001642 001401          BEQ      .+4
902 001644 104000          HLT
                                     ;DATA NOT=52525
903
904 001646 000004          SCOPE
905 001650 012777 125252 177342  MOV      #125252,@DROBUF
906 001656 017700 177336          MOV      @DROBUF,R0
907 001662 022700 125252          CMP      #125252,R0
908 001666 001401          BEQ      .+4
909 001670 104000          HLT
                                     ;DATA NOT=125252
910
911                                     ;TEST RELIABILITY OF DR11-C OUTPUT BUFFER REGISTER
912 001672 000004          SCOPE
913 001674 012737 000040 004366  MOV      #40,@ICOUNT
914 001702 010502          BUFTST: MOV      R5,R2          ;GET ADDRESS OF DRCSR
915 001704 005722          TST      (R2)+          ;R2=ADDRESS OF OUTPUT BUFFER REG.
916 001706 012703 000401          MOV      #401,R3          ;LOAD CONSTANT
917 001712 012704 000400          1$: MOV      #256.,R4          ;SET COUNT
918 001716 005000          CLR      R0          ;PRESET EXPECTED RESULT
919 001720 005012          CLR      (R2)          ;CLEAR REGISTER
920 001722 060300          2$: ADD      R3,R0
921 001724 060312          ADD      R3,(R2)
922 001726 021200          CMP      (R2),R0
923 001730 001401          BEQ      .+4
924 001732 104000          HLT
925 001734 005304          DEC      R4
926 001736 001371          BNE      2$
927 001740 006303          ASL      R3
928 001742 001363          BNE      1$
929
930                                     ;TEST THAT BYTE REFERENCE TO DROBUF AFFECT PROPER BYTE ONLY
931
932 001744 000004          SCOPE
933 001746 012777 177777 177244  TAG: MOV      #-1,@DROBUF
934 001754 105077 177240          CLR      @DROBUF          ;CLEAR LOW BYTE
935 001760 017700 177234          MOV      @DROBUF,R0
936 001764 022700 177400          CMP      #177400,R0
937 001770 001401          BEQ      .+4
938 001772 104000          HLT
                                     ;BYTE LOW FAILED TO CLEAR
939
940 001774 000004          SCOPE
941 001776 012777 177777 177214  MOV      #-1,@DROBUF
942 002004 105077 177214          CLR      @DROBH10          ;CLEAR HIGH BYTE
943 002010 017700 177204          MOV      @DROBUF,R0
944 002014 022700 000377          CMP      #377,R0
945 002020 001401          BEQ      .+4
946 002022 104000          HLT
                                     ;HIGH BYTE CLEAR FAILED
947
948 002024 000004          SCOPE
949 002026 005037 002076          CLR      @#2$
950 002032 012704 002076          MOV      #2$,R4
951 002036 005077 177156          CLR      @DROBUF

```

952	002042	105077	177156		CLRB	2DRBH10	
953	002046	105277	177152		INCB	2DRBH10	; INCREMENT HIGH BYTE
954	002052	105264	000001		INCB	1(R4)	
955	002056	027714	177136		CMP	2DR0BUF, (R4)	
956	002062	001401			BEQ	+.4	
957	002064	104000			HLT		; HIGH BYTE HAS BAD DATA
958	002066	105764	000001		TSTB	1(R4)	
959	002072	001402			BEQ	3\$	
960	002074	000764			BR	1\$	
961	002076	000000				0	
962	002100	000004			2\$: .WORD		
963					3\$: SCOPE		
964	002102	005015			; CONTROL STATUS REGISTER (DRCSR) TESTS.		
965	002104	011500			CLR	(R5)	
966	002106	001401			MOV	(R5), R0	
967	002110	104000			BEQ	+.4	
968	002112	012715	000140		HLT		
969	002116	011500			MOV	#140, 2R5	; INTERRUPT ENABLE FOR A+B
970	002120	022700	000140		MOV	2R5, R0	
971	002124	001401			CMP	#140, R0	; ENABLE BITS
972	002126	104000			BEQ	+.4	
973					HLT		
974	002130	000004			SCOPE		
975	002132	012737	000010	004366	MOV	#10, ICOUNT	
976	002140	012715	000140		MOV	#140, 2R5	; SET INTERRUPT ENABLE FLOPS
977	002144	000005			RESET		; CLEAR THOSE FLOPS
978	002146	011500			MOV	2R5, R0	
979	002150	001401			BEQ	+.4	
980	002152	104000			HLT		; RESET DID NOT CLEAR INTERRUPT ENABLE BITS
981							
982	002154	000004			SCOPE		
983	002156	052715	000001		BIS	#1, 2R5	; SHOULD SET REQ A ALSO
984	002162	021527	000201		CMP	2R5, #201	
985	002166	001401			BEQ	+.4	
986	002170	104000			HLT		
987	002172	005015			CLR	2R5	
988							
989	002174	000004			SCOPE		
990	002176	052715	000002		BIS	#2, 2R5	; SHOULD SET REQ B
991	002202	021527	100002		CMP	2R5, #100002	
992	002206	001401			BEQ	+.4	
993	002210	104000			HLT		
994	002212	005015			CLR	2R5	
995							
996	002214	000004			SCOPE		
997	002216	052737	000340	177776	BIS	#340, 2#PSW	
998	002224	052715	177777		BIS	#-1, 2R5	
999	002230	022715	100343		CMP	#100343, (R5)	
1000	002234	001401			BEQ	+.4	
1001	002236	104000			HLT		
1002	002240	042715	000003		BIC	#3, 2R5	
1003	002244	022715	000140		CMP	#140, 2R5	
1004	002250	001401			BEQ	+.4	
1005	002252	104000			HLT		; WRONG BITS SET
1006							
1007	002254	000004			SCOPE		

```

1008 002256 012737 000340 177776      MOV      #340, @#PSW
1009 002264 052715 000003          BIS      #3, @R5
1010 002270 000005          RESET
1011 002272 005715          TST     @R5
1012 002274 001401          BEQ     .+4
1013 002276 104000          HLT                      ;RESET DID NOT CLEAR
1014
1015 002300 000004          SCOPE
1016 002302 012737 004000 004366      MOV     #4000, ICOUNT
1017 002310 005015          CLR     @R5
1018 002312 005215          INC     @R5
1019 002314 105715          TSTB   @R5
1020 002316 100401          BMI     .+4
1021 002320 104000          HLT                      ;BIT 0 DID NOT SET BIT 7
1022
1023 002322 000004          SCOPE
1024 002324 012715 000002      MOV     #2, @R5
1025 002330 005715          TST     @R5
1026 002332 100401          BMI     .+4
1027 002334 104000          HLT                      ;BIT 1 DID NOT SET BIT 15
1028
1029                      ;TEST EXTERNAL TRANSFERS - CABLE MUST BE CONNECTED
1030 002336 000004          SCOPE
1031 002340 005077 176654          CLR     @DROBUF
1032 002344 017777 176652 176646      MOV     @DRIBUF, @DROBUF ;TEST TRANSFER OF ZERO
1033 002352 001401          BEQ     .+4
1034 002354 104000          HLT                      ;NOT EQUAL TO ZERO
1035
1036 002356 000004          SCOPE
1037 002360 012777 177777 176632      MOV     #-1, @DROBUF
1038 002366 017777 176677 176624      MOV     @DRIBUF, @DROBUF ;MOV ALL ONES
1039 002374 022777 177777 176616      CMP     #-1, @DROBUF
1040 002402 001401          BEQ     .+4
1041 002404 104000          HLT                      ;NOT ALL ONES
1042
1043 002406 000004          SCOPE
1044 002410 005037 004366          CLR     ICOUNT
1045 002414 005000          CLR     RO
1046 002416 010077 176576          MOV     RO, @DROBUF      ;TEST ALL NUMBERS
1047 002422 017777 176574 176570      MOV     @DRIBUF, @DROBUF
1048 002430 020077 176564          CMP     RO, @DROBUF
1049 002434 001401          BEQ     .+4
1050 002436 104000          HLT                      ;ERROR - CHECK RO FOR GOOD
1051 002440 005200          INC     RO              ;DROBUF FOR BAD
1052 002442 001403          BEQ     TST9
1053 002444 005077 176550          CLR     @DROBUF
1054 002450 000762          BR      TST6
1055 002452 000004          SCOPE
1056 002454 012737 007005 004366      MOV     #5, @#ICOUNT
1057                      ;TEST DATA FROM BLACK BOX (NOT CONNECTED)
1058 002462 012777 177777 176530      MOV     #-1, @DROBUF
1059 002470 017777 176524 176524      MOV     @DROBUF, @DRIBUF ;STATIC LINES EQUAL ONES
1060 002476 017700 176520          MOV     @DRIBUF, RO     ;DATA REGISTER TO RO
1061 002502 022700 177777          CMP     #-1, RO
1062 002506 001401          BEQ     .+4
1063 002510 104000          HLT                      ;REG 0 SHOULD = ALL ONES

```

```

1064
1065 ;READY BIT IS IN A ONE STATE
1066 002512 000004 SCOPE
1067 002514 012715 000003 MOV #3,DR5 ;CSRO AND CSRI
1068 002520 011500 MOV (R5),R0
1069 002522 022700 100203 CMP #100203,R0
1070 002526 001401 SEQ .+4
1071 002530 104000 HLT
1072
1073 ;CAN WE RAISE INTERUPT "A"
1074 002532 000004 SCOPE
1075 002534 052737 000340 177776 BIS #340,@PSW ;LOCK OUT INTERRUPTS
1076 002542 012706 001200 MOV #STKPTR,R6
1077 002546 012777 002570 176452 MOV #TST4,@DRVECA ;INTERRUPT RETURN POINTER
1078 002554 012715 000101 MOV #101,DR5 ;INTERRUPT ENABLE AND CSPO
1079 002560 005037 177776 CLR @PSW
1080 002564 000240 NOP
1081 002566 104000 HLT ;NO "A" INTERRUPT
1082 002570 005015 TST4: CLR DR5
1083 002572 013777 001230 176426 MOV DRLVL,@DRVECA ;MOVE .+2 TO "A" INTERRUPT VECTOR
1084
1085 ;RAISE INTERRUPT "B"
1086 002600 000004 SCOPE
1087 002602 012706 001200 MOV #STKPTR,R6
1088 002606 052737 000340 177776 BIS #340,@PSW
1089 002614 012777 002640 176410 MOV #TST5,@DRVECB
1090 002622 012715 000042 MOV #42,DR5 ;IE AND CSRI
1091 002626 042737 000377 177776 BIC #377,@PSW
1092 002634 000240 NOP
1093 002636 104000 HLT ;NO B INTERRUPT
1094 002640 005015 TST5: CLR DR5
1095
1096 ;TEST FOR INTERRUPT FROM DEVICE
1097 002642 013777 001240 176360 MOV PL,@DRLVL
1098 002650 042737 000340 177776 BIC #340,@PSW ;PROCESSOR LEVEL ZERO
1099 002656 012777 002710 176342 MOV #TINT1,@DRVECA
1100 002664 012706 001200 MOV #STKPTR,R6 ;STACK POINTER
1101 002670 042777 000100 176320 BIC #100,@DRCSR ;CLEAR INTERRUPT ENABLE
1102 002676 052777 000101 176312 BIS #101,@DRCSR ;SET INTERRUPT ENABLE-AND CSRO
1103 002704 000240 NOP
1104 002706 104000 HLT ;NO DEVICE INTERRUPT OCCURED
1105 002710 000004 TINT1: SCOPE
1106
1107 ;TEST FOR INTERRUPT FROM THE DEVICE
1108 002712 042737 000340 177776 BIC #340,@PSW
1109 002720 052737 000040 177776 BIS #040,@PSW ;SET TO PRIORITY LEVEL 1
1110 002726 012777 002760 176272 MOV #TINT2,@DRVECA ;INTERRUPT VECTOR ADDRESS
1111 002734 012706 001200 MOV #STKPTR,R6 ;SET UP STACK POINTER
1112 002740 042777 000100 176250 BIC #100,@DRCSR ;CLEAR INTERRUPT ENABLE
1113 002746 052777 000101 176242 BIS #101,@DRCSR ;SET INTERRUPT ENABLE-AND CSRO
1114 002754 000240 NOP
1115 002756 104000 HLT ;NO DEVICE INTERRUPT OCCURED
1116
1117 TINT2: SCOPE
1118 002762 042737 000340 177776 BIC #340,@PSW
1119 002770 052737 000100 177776 BIS #100,@PSW ;SET TO PRIORITY LEVEL 2

```



```

1120 002776 012777 003030 176222      MOV      #TINT3,ADRVECA ; INTERRUPT VECTOR ADDRESS
1121 003004 012706 001200                MOV      #STKPTR,R6    ; SET UP STACK POINTER
1122 003010 042777 000100 176200      BIC      #100,ADRCSR   ; CLEAR INTERRUPT ENABLE
1123 003016 052777 000101 176172      BIS      #101,ADRCSR   ; SET INTERRUPT ENABLE-AND CSRC
1124 003024 000240                NOP
1125 003026 104000                HLT      ; NO DEVICE INTERRUPT OCCURED
1126
1127 003030 000004      TINT3:  SCOPE
1128                                ; TEST FOR INTERRUPT FROM THE DEVICE
1129 003032 042737 000340 177776      BIC      #340,#PSW
1130 003040 052737 000140 177776      BIS      #140,#PSW    ; SET TO PRIORITY LEVEL 3
1131 003046 012777 003100 176152      MOV      #TINT4,ADRVECA ; INTERRUPT VECTOR ADDRESS
1132 003054 012706 001200                MOV      #STKPTR,R6    ; SET UP STACK POINTER
1133 003060 042777 000100 176130      BIC      #100,ADRCSR   ; CLEAR INTERRUPT ENABLE
1134 003066 052777 000101 176122      BIS      #101,ADRCSR   ; SET INTERRUPT ENABLE-AND CSRC
1135 003074 000240                NOP
1136 003076 104000                HLT      ; NO DEVICE INTERRUPT OCCURED
1137 003100 000004      TINT4:  SCOPE
1138                                ; TEST FOR INTERRUPT FROM DEVICE
1139
1140 003102 042737 000340 177776      BIC      #340,#PSW
1141 003110 052737 000200 177776      BIS      #200,#PSW    ; RAISE PROCESSOR PRIORITY TO LEVEL 4
1142 003116 012777 003160 176102      MOV      #TINT5,ADRVECA ; IN CASE OF INTERRUPT
1143 003124 012706 001200                MOV      #STKPTR,R6    ; SET STACK POINTER
1144 003130 042777 000100 176060      BIC      #100,ADRCSR   ; CLEAR INTERRUPT ENABLE
1145 003136 052777 000101 176052      BIS      #101,ADRCSR   ; SET INTERRUPT ENABLE AND CSRC
1146 003144 000240                NOP      ; LET INTERRUPT OCCUR
1147 003146 042777 000100 176042      BIC      #100,ADRCSR
1148 003154 000240                NOP
1149 003156 104000                HLT      ; NO DEVICE INTERRUPT OCCURED
1150 003160 000004      TINT5:  SCOPE
1151                                ; TEST FOR NO INTERRUPT FROM DEVICE (HIGHEST PROCESSOR PRIORITY)
1152
1153 003162 052737 000340 177776      BIS      #340,#PSW    ; RAISE PROCESSOR PRIORITY TO HIGHEST LEVEL
1154 003170 012777 003230 176030      MOV      #TINT6,ADRVECA ; IN CASE OF INTERRUPT
1155 003176 012706 001200                MOV      #STKPTR,R6    ; SET STACK POINTER
1156 003202 042777 000100 176006      BIC      #100,ADRCSR   ; CLEAR INTERRUPT ENABLE
1157 003210 052777 000101 176000      BIS      #101,ADRCSR
1158 003216 000240                NOP
1159 003220 042777 000100 175770      BIC      #100,ADRCSR
1160 003226 000401                BR      .+4            ; WITH NO INTERRUPT, BRANCH OVER HALT
1161 003230 104000                HLT      ; INTERRUPT OCCURED
1162 003232 000004      TINT6:  SCOPE
1163                                ; TEST FOR NO INTERRUPT FROM DEVICE
1164
1165 003234 042737 000340 177776      BIC      #340,#PSW
1166 003242 052737 000240 177776      BIS      #240,#PSW    ; RAISE PROCESSOR PRIORITY TO LEVEL 5
1167 003250 012777 003310 175750      MOV      #TINT7,ADRVECA ; IN CASE OF INTERRUPT
1168 003256 012706 001200                MOV      #STKPTR,R6    ; SET STACK POINTER
1169 003262 042777 000100 175726      BIC      #100,ADRCSR   ; CLEAR INTERRUPT ENABLE
1170 003270 052777 000101 175720      BIS      #101,ADRCSR   ; SET INTERRUPT ENABLE AND CSRC
1171 003276 000240                NOP
1172 003300 042777 000100 175710      BIC      #100,ADRCSR   ; DON'T LEAVE IT SET
1173 003306 000401                BR      .+4            ; WITH NO INTERRUPT, BRANCH OVER HALT
1174 003310 104000                HLT      ; INTERRUPT OCCURED
1175 003312 000004      TINT7:  SCOPE

```

```

1176
1177
1178 003314 042737 000340 177776 ;TEST FOR NO INTERRUPT FROM DEVICE
1179 003322 052737 000300 177776 BIC #340, @#PSW
1180 003330 012777 003370 175670 BIS #300, @#PSW ;RAISE PROCESSOR PRIORITY TO LEVEL 5
1181 003336 012706 001200 MOV #TINT8, @DRVECA ;IN CASE OF INTERRUPT
1182 003342 042777 000100 175646 MOV #STKPTR, R6 ;SET STACK POINTER
1183 003350 052777 000101 175640 BIC #100, @DRCSR ;CLEAR INTERRUPT ENABLE
1184 003356 042777 000100 175632 BIS #101, @DRCSR ;SET INTERRUPT ENABLE-AND CSRD
1185 003364 000240 NOP ;DON'T LEAVE IT SET
1186 003366 000401 BR .+4 ;WITH NO INTERRUPT, BRANCH OVER HALT
1187 003370 104000 TINT8: HLT ;INTERRUPT OCCURED
1188 003372 000004 SCOPE
1189
1190 003374 013777 001230 175624 MOV DRLVL, @DRVECA ;FOR FALSE INTERRUPT
1191 003402 005077 175620 CLR @DRVECA
1192
1193 ;END OF TEST ROUTINE
1194 003406 010237 003750 END: MOV R2, SAVR2
1195 003412 010337 003752 MOV R3, SAVR3
1196 003416 012702 000052 MOV #'*, R2
1197 003422 012703 000062 MOV #50., R3
1198 003426 000005 2$: RESET
1199 003430 005303 DEC R3
1200 003432 001375 BNE 2$
1201 003434 032737 000100 177570 BIT #SW06, SWR
1202 003442 001402 BEQ 1$
1203 003444 004737 000250 JSR PC, EOPHLT
1204 003450 104400 000002 1$: TYPE R2
1205 003454 013702 003750 MOV SAVR2, R2
1206 003460 013703 003752 MOV SAVR3, R3
1207 003464 032737 002000 177570 BIT #SW10, @#SR ;LOOP ON SELECTED DR?
1208 003472 001402 BEQ 4$
1209 003474 000137 001414 JMP @#RSTART ;REPEAT TEST ON DR11C SELECTED
1210 ;STEP TO NEXT DR11-C
1211 003500 012700 000010 4$: MOV #10, R0 ;STEPPING CONSTANT
1212 003504 012737 003554 000004 MOV #5$, @#4 ;SET TIME OUT TRAP
1213 003512 160005 SUB R0, R5 ;STEP TO NEXT DR11-C ADDRESS
1214 003514 005715 TST (R5) ;WILL TIME OUT IF NOT AVAILABLE
1215 003516 012705 001216 MOV #DRCSR, R5 ;SET TABLE POINTER
1216 003522 160025 SUB R0, (R5)+
1217 003524 160025 SUB R0, (R5)+
1218 003526 160025 SUB R0, (R5)+
1219 003530 160025 SUB R0, (R5)+
1220 003532 060025 ADD R0, (R5)+
1221 003534 060025 ADD R0, (R5)+
1222 003536 060025 ADD R0, (R5)+
1223 003540 000137 001414 JMP @#RSTART ;RESTART TEST USING NEXT DR11-C
1224 003544 032737 001000 177570 BIT #SW9, @#SR
1225 003552 001013 BNE 8$
1226 003554 012777 000057 001304 5$: MOV #' /, @#STPB ;TYPE '/'
1227 003562 105777 001276 6$: TSTB @#STPS
1228 003566 100375 BPL 6$
1229 003570 005077 001272 CLR @#STPB
1230 003574 105777 001264 7$: TSTB @#STPS
1231 003575 100375 BPL 7$

```

```

1232 003602 013700 000042      SS:      MOV      Q#42,RO
1233 003606 001405                      BEQ      END1
1234 003610 004710      LOGIC:   JSR      PC,(RO)
1235 003612 000240                      NOP
1236 003614 000240                      NOP
1237 003616 000240                      NOP
1238 003620 000240                      NOP
1239 003622 000137 001366      END1:    JMP      Q#START
1240
1241                      ;ENTERED WITH SYSTEM TRAP CALL(HLT)
1242                      ;PRINT OUT THE ERROR PC AND STATUS REGISTER
1243 003626 033727 177570 020000 .HLT:    BIT      SR,#SW13      ;TEST FOR INHIBIT PRINT OUT
1244 003634 001401                      BEQ      .+4            ;BRANCH TO PRINT
1245 003636 000002                      RTI                     ;INHIBIT, RETURN TO MAIN STREAM
1246 003640 012637 017400      MOV      (6)+,SAVPC    ;PC OF FAILING ROUTINE
1247 003644 012637 017402      MOV      (6)+,SAVCC    ;CC OF ERROR CONDITION
1248 003650 024646      CMP      -(6),-(6)    ;REPOSITION THE STACK
1249 003652 104400 004747      TYPE    ,MCRLF
1250 003656 013746 017400      MOV      SAVPC,-(SP)   ;SAVE SAVPC FOR TYPEOUT
1251 003662 104404                      TYPOS    ;GO TYPE--OCTAL ASCII
1252 003664 006                      .BYTE   6              ;TYPE 6 DIGITS
1253 003665 000                      .BYTE   0              ;SUPPRESS LEADING ZEROS
1254 003666 104400 004752      TYPE    ,MSPACE
1255 003672 013746 017402      MOV      SAVCC,-(SP)   ;SAVE SAVCC FOR TYPEOUT
1256 003676 104404                      TYPOS    ;GO TYPE--OCTAL ASCII
1257 003700 006                      .BYTE   6              ;TYPE 6 DIGITS
1258 003701 000                      .BYTE   0              ;SUPPRESS LEADING ZEROS
1259 003702 104400 004752      TYPE    ,MSPACE
1260 003706 013746 001216      MOV      DRCSR,-(SP)   ;SAVE DRCSR FOR TYPEOUT
1261 003712 104404                      TYPOS    ;GO TYPE--OCTAL ASCII
1262 003714 006                      .BYTE   6              ;TYPE 6 DIGITS
1263 003715 000                      .BYTE   0              ;SUPPRESS LEADING ZEROS
1264 003716 104400 004752      TYPE    ,MSPACE
1265 003722 013746 001226      MOV      DRVECA,-(SP) ;SAVE DRVECA FOR TYPEOUT
1266 003726 104404                      TYPOS    ;GO TYPE--OCTAL ASCII
1267 003730 006                      .BYTE   6              ;TYPE 6 DIGITS
1268 003731 000                      .BYTE   0              ;SUPPRESS LEADING ZEROS
1269 003732 104400 004747      TYPE    ,MCRLF
1270 003736 005737 177570      TST     SR
1271 003742 100001                      BPL     .+4
1272 003744 000000      HALT
1273 003746 000002      RTI                     ;HALT ON ERROR SET
1274 003750 000000                      ;RETURN TO MAIN STREAM
1275 003752 000000      SAVR2:  0
1276 003754 000000      SAVR3:  0
1277                      SAVR4:  0
1278 003756 005037 004234      PRTAB:  CLR      BINCT
1279 003762 005037 004232      CLR      WGTCT
1280 003766 012704 004240      MOV      #LIST,R4      ;GET LIST ADDRESS
1281 003772 012737 000005 004236      MOV      #5,ASCNT
1282 004000 012737 000007 004226      MOV      #7,SEVEN
1283 004006 012737 000001 004230      MOV      #1,DECML
1284 004014 105777 001044      WAIT1: TSTB     Q#1PS
1285 004020 100375                      BPL     WAIT1
1286 004022 005702                      TST     R2
1287 004024 100404                      BMI     MINUS          ;NEG SIGN PRINT 1
  
```

```

1288 004026 012777 000260 001032      MOV      #260, @STPB      ;POS SIGN PRINT 0
1289 004034 000403                BR          STAR
1290 004036 012777 000261 001022  MINUS:  MOV      #261, @STPB
1291 004044 013703 004226                STPR:    MOV      SEVEN, R3      ;PUT MASK IN R3
1292 004050 010237 004224                MOV      R2, TODDLE      ;GET READY TO DOODLE NUMBER IN TODDLE
1293 004054 005137 004224                COM      TODDLE          ;COMPENSATES FOR COMPLEMENT DURING BIC
1294 004060 043703 004224                BIC      TODDLE, R3      ;AND IN OCTAL CHARACTER
1295 004064 001410                BEQ      WRTCC           ;ZERO, WRITE 0 IN LIST
1296 004066 063737 004230 004232  MANJM:  ADD      DECML, WGTCT      ;COUNT UP TO
1297 004074 005237 004234                INC      BINCT           ;AND RECORD
1298 004100 023703 004232                CMP      WGTCT, R3       ;SAME BINARY WEIGHT
1299 004104 001370                BNE      MKNUM          ;KEEP COUNTN
1300 004106 062737 000260 004234  WRTCC:  ADD      #260, BINCT      ;ADD ASCII PREFIX
1301 004114 013724 004234                MOV      BINCT, (4)+     ;WRITE ASCII CHAR IN LIST
1302 004120 063737 004226 004230                ADD      SEVEN, DECML    ;EXPAND BINARY WEIGHT
1303 004126 005037 004232                CLR      WGTCT
1304 004132 005037 004234                CLR      BINCT
1305 004136 005337 004236                DEC      ASCNT
1306 004142 001410                BEQ      XLIST           ;5 CHAR IN LIST
1307 004144 012703 000003                MOV      #3, R3         ;SET X3 FOR ADD LOOP
1308 004150 063737 004226 004226  MOADC:  ADD      SEVEN, SEVEN    ;MAKING SEVENTY BY SEVEN
1309 004156 005303                DEC      R3
1310 004160 001373                BNE      MOADD
1311 004162 000730                BR          STAR
1312 004164 012737 000005 004236  XLIST:  MOV      #5, ASCNT      ;NX SEVEN SET GET NX OCTAL
1313 004172 105777 000666                WAIT2:  TSTB     @STPS         ;SEND 5 CHAR TO TTY
1314 004176 100375                BPL      WAIT2
1315 004200 014477 000662                MOV      -(4), @STPB
1316 004204 005337 004236                DEC      ASCNT
1317 004210 001401                BEQ      HDFHM          ;FINISH PRINTING GET NXT NUM
1318 004212 000767                BR          WAIT2
1319 004214 105777 000644                HDFHM:  TSTB     @STPS
1320 004220 100375                BPL      -4
1321 004222 000207                RTS      R7              ;HEAD FOR HOME
1322 004224 000000                TODDLE: 0
1323 004226 000000                SEVEN:  0
1324 004230 000000                DECML:  0
1325 004232 000000                WGTCT:  0
1326 004234 000000                BINCT:  0
1327 004236 000000                ASCNT:  0
1328 004240 000000                LIST:   0
1329 004242 000000                0
1330 004244 000000                0
1331 004246 000000                0
1332 004250 000000                0
1333                ;SCOPE LOOP ROUTINE ENTERED BY USER TRAP
1334 004252 022606                SCOPEB: CMP      (6)+, R6      ;REPOSITION THE STACK
1335 004254 012637 177776                MOV      (6)+, @PSW
1336 004260 000177 000106                JMP      @RETURN        ;SCOPE RETURN
1337
1338                ;SCOPE OR/AND ITERATION LOOP FOR EACH TEST 4000 TIMES
1339 004264 032737 040000 177570  ;SCOPE: BIT      #5, SR          ;TEST SR FOR SCOPE
1340 004272 001367                BNE      SCOPEB         ;YES SCOPE
1341 004274 005737 001234                TST     @XORFLG
1342 004300 100012                BPL      IS
1343 004302 013746 000004                MOV     @4, -(R6)

```

```

1344 004306 012737 004400 000004      MOV      #XOR,2#4
1345 004314 012737 000031 177060      MOV      #31,2#177060
1346 004322 012637 000004      MOV      (R6)+,2#4
1347 004326 032737 004000 177570 15:      BIT      #SW11,SR      ;NO - TEST FOR ITERATION
1348 004334 001011      BNE      SCOPEA      ;INHIBIT ITERATION
1349 004336 023737 004370 004366      CMP      SCOPEF,ICOUNT
1350 004344 001403      BEQ      SCOPEG      ;EXIT - DONE
1351 004346 005237 004370      INC      SCOPEF      ;INCREMENT COUNT
1352 004352 000737      BR       SCOPEB      ;LOOP SOME MORE
1353 004354 005037 004370      SCOPEG: CLR      SCOPEF      ;CLEAR COUNT
1354 004360 011637 004372      SCOPEA: MOV      @R6,RETURN ;SAVE SCOPE RETURN POINTER
1355 004364 000002      RTI      ;RETURN INLINE-NEXT TEST
1356 004366 004000      ICOUNT: 4000
1357 004370 000000      SCOPEF: 0      ;COUNT LOCATION FOR ITERATION LOOP
1358 004372 001446      RETURN: BEGIN ;ADDRESS OF LAST TEST
1359 004374 000137 000200      JMP      200
1360
1361 004400 022626      XOR:      CMP      (R6)+,(R6)+
1362 004402 012637 000004      MOV      (R6)+,2#4
1363 004406 000721      BR       SCOPEB
1364      ;PRINT DEVICE ADDRESS AND VECTOR
1365 004410 012777 000240 000450  MOREID: MOV      #240,@STPB
1366 004416 105777 000442      TSTB    @STPB
1367 004422 100375      BPL     .-4
1368 004424 013702 001216      MOV     @DRCSR,R2
1369 004430 004737 003756      JSR    R7,PRTAB
1370 004434 012777 000240 000424      MOV     #240,@STPB
1371 004442 105777 000416      TSTB    @STPB
1372 004446 100375      BPL     .-4
1373 004450 013702 001226      MOV     DRVECA,R2
1374 004454 004737 003756      JSR    R7,PRTAB
1375 004460 012777 000215 000400      MOV     #215,@STPB
1376 004466 105777 000372      TSTB    @STPB
1377 004472 100375      BPL     .-4
1378 004474 012777 000212 000364      MOV     #212,@STPB
1379 004502 105777 000356      TSTB    @STPB
1380 004506 100375      BPL     .-4
1381 004510 005077 000352      CLR     @STPB
1382 004514 105777 000344      TSTB    @STPB
1383 004520 100375      BPL     .-4
1384 004522 000207      RTS     R7      ;BACK TO PRINT
1385
1386      ;ENTER HERE FOR POWER FAIL
1387
1388 004524 010046      PFAIL:  MOV     R0,-(6)      ;SAVE REGISTER OR STACK
1389 004526 010146      MOV     R1,-(6)      ;WHEN POWERING DOWN
1390 004530 010246      MOV     R2,-(6)
1391 004532 010346      MOV     R3,-(6)
1392 004534 010446      MOV     R4,-(6)
1393 004536 010546      MOV     R5,-(6)
1394 004540 013746 000024      MOV     24,-(6)
1395 004544 010637 004560      MOV     R6,@SAVR6      ;STORE STACK POSITION
1396 004550 012737 004562 000024      MOV     #RESTAR,2#24
1397 004556 000000      HALT    ;HALT ON POWER DOWN NORMAL
1398 004560 000000      SAVR6: 0      ;STACK IS SAVED HERE
1399 004562 013706 004560      RESTAR: MOV     SAVR6,R6 ;RESTORE REGISTER OFF STACK
  
```

```

1400 004566 012637 000024      MOV      (6)+,R4      ;WHEN POWERING UP
1401 004572 012605      MOV      (6)+,R5
1402 004574 012604      MOV      (6)+,R4
1403 004576 012603      MOV      (6)+,R3
1404 004600 012602      MOV      (6)+,R2
1405 004602 012601      MOV      (6)+,R1
1406 004604 012600      MOV      (6)+,R0
1407 004606 000137 001414      JMP      @RSTART
1408
1409      ;ENTER HERE FOR UNIQUE SELECTION OF DF11C
1410
1411 004612 000000      SPEC0:  HALT          ;PLACE ADDRESS OF DR11-C CONTROL STATUS
1412 004614 013701 177570      MOV      @SR,R1
1413 004620 004737 004642      JSR      PC,@SPEC1
1414 004624 000000      HALT
1415 004626 013701 177570      MOV      @SR,R1
1416 004632 004737 004672      JSR      PC,@SPEC2
1417 004636 000137 001366      JMP      @START
1418
1419 004642 012700 001200      SPEC1:  MOV      @RCSR,R0      ;SET TABLE ADDRESS
1420 004646 010120      MOV      R1,(R0)+      ;LOAD INTO TABLE STARTING AT RCSR
1421 004650 062701 000002      ADD      #2,R1          ;STEP TO ADDRESS OF DROUTBUF
1422 004654 010120      MOV      R1,(R0)+      ;LOAD INTO TABLE
1423 004656 062701 000002      ADD      #2,R1          ;STEP TO ADDRESS OF DRINBUF
1424 004662 010120      MOV      R1,(R0)+      ;LOAD INTO TABLE
1425 004664 005301      DEC      R1             ;FORM ADDRESS OF DROUTBUF+1
1426 004666 010120      MOV      R1,(R0)+      ;LOAD INTO TABLE
1427 004670 000207      RTS      PC
1428
1429 004672 012700 001210      SPEC2:  MOV      @RCSR1,R0
1430 004676 010120      MOV      R1,(R0)+      ;LOAD INTO TABLE
1431 004700 005721      TST      (R1)+
1432 004702 010120      MOV      R1,(R0)+
1433 004704 005721      TST      (R1)+
1434 004706 010120      MOV      R1,(R0)+
1435 004710 000207      RTS      PC
1436
1437
1438
1439 004712 005015 047531 020125      MESS1:  .ASCIZ  <15><12>'YOU ARE ON AN XOR TESTER'<15><12>
1440 004720 051101 020105 047117
1441 004726 040440 020116 047530
1442 004734 020122 042524 052123
1443 004742 051105 005015      000
1444 004747      015 000012
1445 004752 020040 000040      MCRLF:  .ASCIZ  <15><12>
1446      MSPACE: .ASCIZ  / /
1447      .EVEN
1448      ;*****
1449      .SBTTL  TYPE ROUTINE
1450
1451      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1452      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1453      ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1454      ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1455      ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

```



```

1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524 005074 017646 000000
1525 005100 116637 000001 005317
1526 005106 112637 005321
1527 005112 062716 000002
1528 005116 000406
1529 005120 112737 000001 005317
1530 005126 112737 000006 005321
1531 005134 112737 000005 005316
1532 005142 010346
1533 005144 010446
1534 005146 010546
1535 005150 113704 005321
1536 005154 005404
1537 005156 062704 000006
1538 005162 110437 005320
1539 005166 113704 005317
1540 005172 016605 000012
1541 005176 005003
1542 005200 006105 1$:
1543 005202 000404 BR 3$:
1544 005204 006105 2$:
1545 005206 006105
1546 005210 006105
1547 005212 010503
1548 005214 006103 3$:
1549 005216 105337 005320
1550 005222 100016
1551 005224 042703 177770
1552 005230 001002
1553 005232 005704
1554 005234 001403
1555 005236 005204 4$:
1556 005240 052703 000060
1557 005244 052703 000040 5$:
1558 005250 110337 005314
1559 005254 104400 005314
1560 005260 105337 005316 7$:
1561 005264 003347
1562 005266 002402
1563 005270 005204
1564 005272 000744
1565 005274 012605 6$:
1566 005276 012604
1567 005300 012603

; *
; *STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
; *STYPOS OR STYPOC
; *CALL:
; *      MOV      NUM,-(SP)          ;NUMBER TO BE TYPED
; *      TYPON                    ;CALL FOR TYPEOUT
; *
; *STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
; *CALL:
; *      MOV      NUM,-(SP)          ;NUMBER TO BE TYPED
; *      TYPOC                     ;CALL FOR TYPEOUT
; *
; *STYPOS: MOV      2(SP),-(SP)      ;PICKUP THE MODE
; *          MOVB   1(SP),SOFILL     ;LOAD ZERO FILL SWITCH
; *          MOVB   (SP)+,SOMODE+1   ;NUMBER OF DIGITS TO TYPE
; *          ADD    #2,(SP)          ;ADJUST RETURN ADDRESS
; *          BR     STYPON
; *STYPOC: MOVB   #1,SOFILL         ;SET THE ZERO FILL SWITCH
; *          MOVB   #6,SOMODE+1     ;SET FOR SIX(6) DIGITS
; *STYPON: MOVB   #5,SOCNT          ;SET THE ITERATION COUNT
; *          MOV    R3,-(SP)         ;SAVE R3
; *          MOV    R4,-(SP)         ;SAVE R4
; *          MOV    R5,-(SP)         ;SAVE R5
; *          MOVB   SOMODE+1,R4      ;GET THE NUMBER OF DIGITS TO TYPE
; *          NEG    R4
; *          ADD    #6,R4            ;SUBTRACT IT FOR MAX. ALLOWED
; *          MOVB   R4,SOMODE        ;SAVE IT FOR USE
; *          MOVB   SOFILL,R4        ;GET THE ZERO FILL SWITCH
; *          MOV    12(SP),R5        ;PICKUP THE INPUT NUMBER
; *          CLR    R3              ;CLEAR THE OUTPUT WORD
; *          ROL   R5                ;ROTATE MSB INTO "C"
; *          BR     3$              ;GO DO MSB
; *          ROL   R5                ;FORM THIS DIGIT
; *          ROL   R5
; *          ROL   R5
; *          MOV   R5,R3
; *          ROL   R3                ;GET LEFT OF THIS DIGIT
; *          DECB  SOMODE            ;TYPE ...IS DIGIT?
; *          BPL   7$                ;BR IF NO
; *          BIC   #17770,R3        ;GET RID OF JUNK
; *          BNE   4$                ;TEST FOR 0
; *          TST  R4                ;SUPPRESS THIS 0?
; *          BEQ  5$                ;BR IF YES
; *          INC  R4                ;DON'T SUPPRESS ANYMORE 0'S
; *          BIS  #'0,R3            ;MAKE THIS DIGIT ASCII
; *          BIS  #' ,R3            ;MAKE ASCII IF NOT ALREADY
; *          MOVB R3,#$             ;SAVE FOR TYPING
; *          TYPE #8$              ;GO TYPE THIS DIGIT
; *          DECB SOCNT            ;COUNT BY 1
; *          BGT  2$                ;BR IF MORE TO DO
; *          BLT  6$                ;BR IF DONE
; *          INC  R4                ;INSURE LAST DIGIT ISN'T A BLANK
; *          BR   2$                ;GO DO THE LAST DIGIT
; *          MOV  (SP)+,R5          ;RESTORE R5
; *          MOV  (SP)+,R4          ;RESTORE R4
; *          MOV  (SP)+,R3          ;RESTORE R3

```



```

1568 005302 016666 000002 000004      MOV      2(SP),4(SP)      ;SET THE STACK FOR RETURNING
1569 005310 012616                MOV      (SP)+,(SP)
1570 005312 000002                RTI          ;RETURN
1571 005314 000                $S:      .BYTE 0          ;STORAGE FOR ASCII DIGIT
1572 005315 000                .BYTE 0          ;TERMINATOR FOR TYPE ROUTINE
1573 005316 000                $OCNT:  .BYTE 0          ;OCTAL DIGIT COUNTER
1574 005317 000                $OFILL: .BYTE 0          ;ZERO FILL SWITCH
1575 005320 000000                $OMODE: 0          ;NUMBER OF DIGITS TO TYPE
1576                ;:*****
1577                .SBTTL POWER DOWN AND UP ROUTINES
1578
1579                :POWER DOWN ROUTINE
1580
1581 005322 012737 005444 000024  $PWRDN: MOV      $SILLUP,2#PWRVEC      ;SET FOR FAST UP
1582 005330 012737 000340 000026  MOV      #340,2#PWRVEC+2          ;PRIO:7
1583 005336 010046                MOV      R0,-(SP)                ;PUSH R0 ON STACK
1584 005340 010146                MOV      R1,-(SP)                ;PUSH R1 ON STACK
1585 005342 010246                MOV      R2,-(SP)                ;PUSH R2 ON STACK
1586 005344 010346                MOV      R3,-(SP)                ;PUSH R3 ON STACK
1587 005346 010446                MOV      R4,-(SP)                ;PUSH R4 ON STACK
1588 005350 010546                MOV      R5,-(SP)                ;PUSH R5 ON STACK
1589 005352 010637 005450                MOV      SP,$SAVR6              ;SAVE SP
1590 005356 012737 005370 000024  MOV      $PWRUP,2#PWRVEC          ;SET UP VECTOR
1591 005364 000000                HALT
1592 005366 000776                BR      .-2                      ;HANG UP
1593
1594                :POWER UP ROUTINE
1595
1596 005370 013706 005450  $PWRUP: MOV      $SAVR6,SP          ;GET SP
1597 005374 005037 005450  CLR      $SAVR6                ;WAIT LOOP FOR THE TTY
1598 005400 005237 005450  $S:      INC      $SAVR6          ;WAIT FOR THE INC
1599 005404 001375                BNE      $S                      ;OF WORD
1600 005406 012605                MOV      (SP)+,R5                ;POP STACK INTO R5
1601 005410 012604                MOV      (SP)+,R4                ;POP STACK INTO R4
1602 005412 012603                MOV      (SP)+,R3                ;POP STACK INTO R3
1603 005414 012602                MOV      (SP)+,R2                ;POP STACK INTO R2
1604 005416 012601                MOV      (SP)+,R1                ;POP STACK INTO R1
1605 005420 012600                MOV      (SP)+,R0                ;POP STACK INTO R0
1606 005422 012737 005322 000024  MOV      $PWRDN,2#PWRVEC          ;SET UP THE POWER DOWN VECTOR
1607 005430 012737 000340 000026  MOV      #340,2#PWRVEC+2          ;PRIO:7
1608 005436 104400 005452                TYPE      , $POWER                ;POWER FAIL MESSAGE
1609 005442 000002                RTI
1610 005444 000000                $SILLUP: HALT                    ;THE POWER UP SEQUENCE WAS STARTED
1611 005446 000776                BR      .-2                      ;BEFORE THE POWER DOWN WAS COMPLETE
1612 005450 000000                $SAVR6: 0
1613 005452 005015 047520 042527  $POWER: .ASCIZ <15><12>"POWER"
1614                ;PUT THE SP HERE
1615                .EVEN
1616                ;:*****
1617                .SBTTL TRAP DECODER
1618
1619                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1620                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1621                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
1622                ;*GO TO THAT ROUTINE.
1623

```

```

1624 005462 010046 $TRAP: MUV RO, -(SP) ;SAVE RO
1625 005464 016600 000002 MOV 2(SP),RO ;GET TRAP ADDRESS
1626 005470 005740 TST -(RO) ;BACKUP BY 2
1627 005472 111000 MOVB (RO),RO ;GET RIGHT BYTE OF TRAP
1628 005474 016000 005502 MOV $TRPAD(RO),RO ;INDEX TO TABLE
1629 005500 000200 RTS RO ;GO TO ROUTINE
  
```

1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651

.SBTTL TRAP TABLE

;\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
 ;\*BY THE "TRAP" INSTRUCTION.

: ROUTINE  
 :-----

```

$TRPAD:
$TYPE ;CALL=TYPE TRAP+0(104400) TTY TYPEOUT ROUTINE
$TYPC ;CALL=TYPC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING
$TYPOS ;CALL=TYPOS TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZE
$TYPON ;CALL=TYPON TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST C

.=017400
SAVPC: .WORD 0
SAVCC: .WORD 0
.END
  
```

ASCNT	004236	1281*	1305*	1312*	1316*	1327*													
BEGIN	001446	859	864	866*	1358														
BINCT	004234	1278*	1297*	1300*	1301	1304*	1326*												
BIT0	= 000001	502*																	
BIT00	= 000001	492*	502																
BIT01	= 000002	491*	501																
BIT02	= 000004	490*	500																
BIT03	= 000010	489*	499																
BIT04	= 000020	488*	498																
BIT05	= 000040	487*	497																
BIT06	= 000100	486*	496																
BIT07	= 000200	485*	495																
BIT08	= 000400	484*	494																
BIT09	= 001000	483*	493																
BIT1	= 000002	501*																	
BIT10	= 002000	482*																	
BIT11	= 004000	481*																	
BIT12	= 010000	480*																	
BIT13	= 020000	479*																	
BIT14	= 040000	478*																	
BIT15	= 100000	477*																	
BIT2	= 000004	500*																	
BIT3	= 000010	499*																	
BIT4	= 000020	498*																	
BIT5	= 000040	497*																	
BIT6	= 000100	496*																	
BIT7	= 000200	495*																	
BIT8	= 000400	494*																	
BIT9	= 001000	493*																	
BPTVEC	= 000014	509*																	
BUFTST	001702	914*																	
COUNT	001236	823*																	
CSR	= 167770	358*	803	804	805	806													
DECML	004230	1283*	1296	1302*	1324*														
DISPLA	= 177570	434*																	
DRBHIO	001224	816*	942*	952*	953*														
DRCSR	001216	813*	850	866	1101*	1102*	1112*	1113*	1122*	1123*	1133*	1134*	1144*	1145*					
		1147*	1156*	1157*	1159*	1169*	1170*	1172*	1182*	1183*	1184*	1215	1260	1368					
DRIBUF	001222	815*	892	1032	1038	1047	1059*	1060											
DRLVL	001230	819*	1083	1097*	1190														
DROBUF	001220	814*	870*	872	882*	883	890*	898*	899	905*	906	933*	934*	935					
		941*	943	951*	955	1031*	1052*	1037*	1038*	1039	1046*	1047*	1048	1053*					
		1058*	1059																
DRVECA	001226	818*	1077*	1083*	1099*	1110*	1120*	1131*	1142*	1154*	1167*	1180*	1190*	1191*					
		1265	1373																
DRVECB	001232	820*	1089*																
EMTVEC	= 000030	512*																	
END	003406	1194*																	
END1	003622	1233	1239*																
EOPHLT	000250	793*	1203																
ERRVEC	= 000004	505*																	
FIRST	001272	827	830	832*															
GNS	= ***** U	1640	1641	1642	1643														
HDFHM	004214	1317	1319*																
HLT	= 104000	396*	874	876	886	894	902	909	924	938	946	957	967	972					
		980	986	993	1001	1005	1013	1021	1027	1034	1041	1050	1063	1071					













COURT: DISTRICT OF COLUMBIA  
CASE NO: 93-1019-76  
DATE: 10-15-93  
PAGE: 43  
REFERENCE: --  
MORNING: --  
AFTERNOON: --

STY 1 43 144  
STY 1 43 153



MOV8	1475	1492	1525	1526	1529	1530	1531	1535	1538	1539	1558	1627			
NEG	1536														
NOP	1080	1092	1103	1114	1124	1135	1146	1148	1158	1171	1185	1235	1236	1237	1238
RESET	871	891	977	1010	1198										
ROL	1542	1544	1545	1546	1548										
RTI	1245	1273	1355	1480	1570	1608									
RTS	798	843	847	1321	1384	1427	1435	1493	1629						
SUB	1213	1216	1217	1218	1219										
TRAP	1631	1641	1642	1643											
TST	892	915	1011	1025	1214	1270	1286	1341	1431	1433	1477	1553	1626		
TSTB	958	1019	1227	1230	1284	1313	1319	1366	1371	1376	1379	1382	1469	1490	
.RECIZ	1439	1444	1445	1612											
.BYTE	1252	1253	1257	1258	1262	1263	1267	1268	1496	1497	1498	1499	1571	1572	1573
	1574														
.ENABL	1	419													
.END	1651														
.ENDC	427	503	517	519	1253	1254	1258	1259	1262	1264	1268	1269	1448	1501	1577
	1589	1599	1608	1615	1615	1625	1628	1640	1641	1642	1643	1644			
.EQUIV	427	428	430	445	446	465	466	467	468	469	470	471	472	473	474
	493	494	495	496	497	498	499	500	501	502					
.EVEN	1446	1614													
.IF	425	475	503	519	1252	1253	1257	1258	1262	1263	1267	1268	1447	1500	1576
	1589	1599	1607	1608	1612	1615	1624	1628	1631	1641	1642	1643	1644		
.IFF	425	1252	1253	1257	1258	1262	1263	1267	1268	1448	1501	1577	1608	1616	1625
.IIF	1251	1256	1261	1266	1494	1495	1496	1497	1498	1499	1640	1641	1642	1643	
.IRP	519	1583	1599												
.LIST	1	5	419	517	519	1631	1640	1641	1642	1643	1644				
.MACRO	1	1631													
.MCALL	420	421	517												
.MLIST	1	5	419	517	519	1631	1640	1641	1642	1643	1644				
.REM	6														
.REPT	521														
.SBTTL	423	1449	1502	1578	1617	1632									
.WORD	961	1647	1649												

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DDDRAA.SEQ/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZ=DDDRAA.SML,DDCRAA.P11  
 RUN-TIME: 16 20 2 SECONDS  
 RUN-TIME RATIO: 90/39=2.3  
 CORE USED: 21K (41 PAGES)

