

-REM -

IDENTIFICATION

PRODUCT CODE: AC-E709I-MC
PRODUCT NAME: CXDLA10 DL11 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT 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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT:

DLA IS AN IOMOD THAT EXERCISES UP TO SIXTEEN DL11 ASYNCHRONOUS INTERFACES. IT IS CAPABLE OF EXERCISING ALL DL11 MODELS. IT USES MAINTENANCE MODE TO TRANSMIT AND RECEIVE A BINARY COUNT PATTERN OUTPUT AND RECEIVED IN 64 CHARACTER BURSTS. THE MAJOR PORTION OF THE ERROR CHECKING IS DEFERRED TO PRIORITY LEVEL 0. ALL LINES SELECTED FOR TEST (UP TO 16 DL11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) ARE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED ON THE CONSOLE TTY.

NOTE:

THIS MODULE IS NOT DESIGNED TO EXERCISE ANY TYPE OF TERMINAL DEVICE. IT IS DESIGNED ONLY TO EXERCISE THE DL. ANY INFORMATION DISPLAYED OR PRINTED IS INCIDENTAL AND SHOULD BE IGNORED. IN MOST CASES THE TERMINAL DEVICE SHOULD BE PLACED IN THE OFF-LINE MODE.

2. REQUIREMENTS:

HARDWARE: AT LEAST ONE DL11 INTERFACE
NOTE: FOR THE DLV11, THE TEST CONNECTOR
MUST BE INSTALLED !

STORAGE:: DLA REQUIRES:
1. DECIMAL WORDS: 917
2. OCTAL WORDS: 1625
3. OCTAL BYTES: 3452

3. PASS DEFINITION:

ONE PASS OF THE DLA MODULE CONSISTS OF TRANSMITTING AND RECEIVING SOME CHARACTERS. THE EXACT NUMBER DEPENDS ON THE BAUD RATE

4. EXECUTION TIME:

VARIABLES WITH BAUD RATE BUT SHOULD TAKE ABOUT ONE
MINUTE TO COMPLETE ONE PASS WHEN RUNNING ALONE.
SR1 MUST BE SET TO MATCH THE BAUD RATE OR THE
PASS TIMES WILL BE OFF.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 176500, VCT: 1, BR1: 4, BR2: 0, DVC: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

VCT: VECTOR ADDRESS OF FIRST DL11
DVC: NO OF DL11'S IF GREATER THAN 1

6. DEVICE OPTION SETUP:

NONE REQUIRED

7. MODULE OPERATION:

7.1 TEST SEQUENCE:

- A. START: USING THE DEVICE SELECTION PARAMETER "DVID1" THIS SECTION OF CODE SETS UP THE VECTORS OF ALL SELECTED LINES TO POINT TO THE APPROPRIATE JSR INSTRUCTION IN THE JSR LINKING TABLE.
- B. SETCSR: THIS PIECE OF CODE INSERTS THE PROPER CSR ADDRESS OF EACH ACTIVE LINE INTO THE THIRD WORD OF EACH JSR TABLE ENTRY.
- C. STOP: THIS ROUTINE INITIALIZES ALL TABLES, BUFFERS, FLAGS AND COUNTERS, THEN PROCEEDS TO TURN ON THE INTERRUPTS FOR ALL ACTIVE LINES. IT USES THE CONTENTS OF THE ACTIVE DEVICE TABLE TO FIND OUT WHICH LINES TO KICK OFF. AFTER INITIALIZING ALL LINES IT WAITS FOR COMPLETION OF 64 TRANSMITTER AND RECEIVER INTERRUPTS VIA A BREAK LOOP. IF THE 64 INTERRUPTS HAVE OCCURRED ON BOTH TRANSMITTER AND RECEIVER, OR IF THE BREAK LOOP TIMES OUT, CONTROL PASSES TO ERRCHK.
- D. TINT: THE TRANSMITTER SERVICE ROUTINE SIMPLY QUEUES UP THE REQUEST FOR SERVICE IN A FIFO QUEUE, UPDATES THE POINTER, AND RETURNS CONTROL BACK TO THE MONITOR WITH A PIR0. THE ELEMENT THAT GETS STORED IN THE QUEUE IS A POINTER TO THE INTERRUPTING CSR ADDRESS. THE ACTUAL SERVICING IS DONE LATER WHERE THE SERVICE CODE IS EXECUTED AT PRIORITY LEVEL 0.
- E. TSERV: THIS CODE RETRIEVES A POINTER FROM THE FIFO QUEUE AND BUILDS THE CSR ADDRESS. THE FOLLOWING SEQUENCE IS EXECUTED:
1. TEST FOR END OF 64. CHAR BURST - IF END EXIT - IF NOT GO TO 2
 2. TEST READY FLAG - IF NOT ASSERTED GO REPORT FALSE INTERRUPT - IF ASSERTED PROCEED TO STEP 3
 3. COUNT THE INTERRUPT FOR INDIVIDUAL LINE
 4. GENERATE AND OUTPUT NEXT CHARACTER,

KEEP TRACK OF THE NUMBER OF CHARACTERS
OUTPUT ON THE LINE, AND THEN EXIT BACK TO
THE MONITOR.

- F. RINT: THE RECEIVER SERVICE ROUTINE STORES
DATA AND STATUS INFORMATION IN A RE-
CEIVER STARTUP TABLE, TESTS FOR THE
END OF A 64. CHAR XFR SEQUENCE AND
THEN EXECUTES AN "RTI" - IT ALSO COUNTS
RECEIVE INTERRUPTS IN A SEPARATE
COUNTER FOR EACH LINE.
SEPARATE COUNTER.
- G. FRRCHK: THE BULK OF THE ERROR CHECKING AND RE-
PORTING IS DONE HERE AT THE END OF EACH
64. CHAR. BURST. THE FOLLOWING SEQUENCE
IS EXECUTED:
1. TURN OFF RCVR AND XMTR INTR. ENABLES
FOR ALL ACTIVE LINES
 2. SCAN THROUGH THE RECEIVER STATUS TABLE
(64 ENTRIES OF TWO WORDS EACH) TO CHECK
FOR AND REPORT:
 - A.) PARITY, FRAMING AND OVER-
RUN ERRORS.
 - B.) RCVR FALSE INTERRUPTS
 - C.) DATA COMPARE ERRORS, ONLY
IF A AND B DID NOT OCCUR.
 3. CHECK RECEIVER AND TRANSMITTER INTERRUPT
COUNTS FOR EACH LINE TO BE SURE THAT NO LINES
WERE DROPPED OR HAD TOO MANY INTERRUPTS.
 4. GO TO THE ENPS ROUTINE AFTER CHECKING
ALL 64 ENTRIES.
- H. ENPS: COUNT THE 64. CHAR BURST AND TEST FOR
128. BURSTS (8192 CHARS). IF NOT END
OF PASS GO TO I. IF END REPORT END
OF PASS AND GO TO C.
- I. RESYNC: RESYNC THE DATA BUFFERS AND THEN RESTART
AT STEP C.

7.2 DESCRIPTION OF TABLES, QUEUES, AND BUFFERS

- A. RSTAR: THIS IS A 128. WORD STATUS TABLE CON-
SISTING OF 64. TWO WORD ENTRIES. IT
GETS LOADED DURING RECEIVER INTERRUPT
SERVICE AND CHECKED AT THE END OF EACH
64. CHAR BURST. EACH ENTRY HAS THE

FOLLOWING FORMAT:

- 1ST WORD: CONTENTS OF RCSR
2ND WORD: LO BYTE = RCVD DATA BYTE
HI BYTE = LINE NUMBER
- B. RCNT: 16 BYTE TABLE CONTAINING AN 8 BIT INTERRUPT COUNTER FOR EACH RCVR. THE APPROPRIATE BYTE GETS INCREMENTED DURING RCVR INTR SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.
- C. TCNT: 16 BYTE TABLE CONTAINING AN 8-BIT INTERRUPT COUNTER FOR EACH TRANSMITTER. THE APPROPRIATE BYTE GETS INCREMENTED DURING DEFERRED INTR. SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.
- D. DLNT: 16 BYTE TABLE CONTAINING AN 8-BIT DATA COUNTER FOR EACH LINE. THE APPROPRIATE BYTE GETS INCREMENTED EACH TIME A CHARACTER IS TRANSMITTED ON THE LINE, AND CLEARED BEFORE THE BEGINNING OF EACH 64. WORD BURST.
- E. TQ: 16 WORD FIFO QUEUE FOR TRANSMITTER SERVICE. LOADED DURING XMTR INTERRUPT SERVICE WITH A POINTER TO THE CSR ADDRESS AND UNLOADED DURING DEFERRED XMTR SERVICE.
- F. XBUF: 16 BYTE XMTR DATA BUFFERS - ONE BYTE/XMTR
- G. RBUF: 16 BYTE RCVR DATA BUFFERS - ONE BYTE/RCVR.
- H. JSRTAB: A 128 WORD TABLE THAT CONTAINS 64 JSR INSTRUCTIONS WITH TWO TRAILING ARGUMENTS. EACH RECEIVER AND EACH XMTR HAS AN ASSIGNED JSR IN THE TABLE OF THE FOLLOWING FORMAT:

JSR R5,RINT(TINT)
0
N

WHERE THE 0 GETS OVERLAYED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL (00-17)

8. OPERATOR OPTIONS:

- A. THE USER CAN USE THE "MOD" COMMAND TO DUMP THE TABLES

BUFFERS DESCRIBED IN 7.2 TO OBTAIN MORE DETAILED ERROR INFORMATION.

- B. THE USER CAN MODIFY (DLA 14) "DVID1" TO SELECT OR DESELECT INDIVIDUAL DL11'S.
- C. SRI IS USED TO SPECIFY THE BAUD RATE AND CHARACTER SIZE WITH WHICH THE DLA MODULE IS TO WORK. THE DEFAULT IS 9600 BAUD WITH AN 8-BIT CHARACTER SIZE.
 - 1. CHAR. SIZE:

SRI	BIT1	BIT0	CHAR SIZE
0	0	0	8-BIT
0	0	1	7-BIT
1	0	0	6-BIT
1	1	1	5-BIT

- 2. BAUD RATE:
BITS 2-14 ARE USED. ONE AT MOST SHOULD BE SET; IF MORE THAN ONE BIT IS SET, THE RIGHT-MOST WILL BE USED.

BAUD	BIT SET
9600	NONE
7200	2
4800	3
2400	4
1800	5
1200	6
600	7
300	8
200	9
150	10
134.5	11
110	12
75	13
50	14

- 3. EXAMPLES:

SRI=5 MEANS 7-BIT CHARACTER AT 7200 BAUD
SRI=12 MEANS 6-BIT CHARACTER AT 4800 BAUD
SRI=1000 MEANS 8-BIT CHARACTER AT 200 BAUD
SRI=0 (DEFAULT) MEANS 8-BIT CHARACTER AT 9600 BAUD

9. NON-STANDARD PRINTOUTS:

THERE ARE TWO ERROR PRINTOUTS WHICH SUPPLY SPECIAL INFORMATION
IN THE CSRC AND STATC VALUES (CONSULT LISTING).


```

;DL11 A-D DEC/X11 EXERCISER MODULE
000000- IOMOD <DLAI > 176500,1,4,0,26
000000- MODULE 140000,DLAI,176500,1,4,0,26
; -TITLE DLAI DEC/X11 SYSTEM EXERCISER MODULE
; DDICOM VERSION 6 -LIST BIN
*****
000000- BEGIN: .ASCII /DLAI / ;MODULE NAME
000000- 046104 044501 040 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000005- 000 ADDR: 176500+0 ;1ST DEVICE ADDR.
000006- 176500 VECTOR: 1-0 ;1ST DEVICE VECTOR.
000010- 000001 BR1: .BYTE PRTV4+0 ;1ST BR LEVEL.
000013- 000 BR2: .BYTE PRTV+0 ;2ND BR LEVEL.
000014- 000001 DVID1: +5 ;DEVICE INDICATOR 1.
000016- 000000 SR1: OPEN ;SWITCH REGISTER 1
000020- 000000 SR2: OPEN ;SWITCH REGISTER 2
000022- 000000 SR3: OPEN ;SWITCH REGISTER 3
000024- 000000 SR4: OPEN ;SWITCH REGISTER 4
*****
000026- 140000 STAT: 140000 ;STATUS WORD.
000030- 000224 INIT: START ;MODULE START ADDR.
000032- 000224 SPOINT: MODDSP ;MODULE STACK POINTER.
000034- 000000 PASCNT: 0 ;PASS COUNTER.
000036- 000000 ICOUNT: 0 ;# OF ITERATIONS PER PASS=0
000040- 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
000042- 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044- 000000 SRFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046- 000000 HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050- 000000 SVSCHK: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052- 000000 RANMUM: 0 ;# OF SVS ERRORS ACCUMULATED
000054- 000000 CNPFC: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
*****
000056- 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060- 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000064- 000000 SVR0: OPEN ;LOC TO SAVE R0.
000066- 000000 SVR1: OPEN ;LOC TO SAVE R1.
000068- 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070- 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072- 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074- 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076- 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100- 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102- 000000 SADR: OPEN ;ADDR OF GOOD DATA, OR
000104- 000000 BASADR: OPEN ;CONTENTS OF CSR.
000106- 000000 ASADR: OPEN ;ADDR OF BAD DATA, OR
000108- 000000 ASADR: OPEN ;CONTENTS OF CSR.
000110- 000000 ERRTYP: 0 ;STATUS REG CONTENTS.
000112- 000522 AMAS: OPEN ;TYPE OF ERROR.
000114- 000000 RSTRT: RSTRT ;EXPECTED DATA.
000116- 000000 WDT0: OPEN ;ACTUAL DATA.
;RESTART ADDRESS AFTER END OF PASS
;WORDS TO MEMORY PER ITERATION

```

```

000116- 000000 WDRP: OPEN ;WORDS FROM MEMORY PER ITERATION
000120- 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122- 000026 IDNUM: 26 ;MODULE IDENTIFICATION NUMBER=26
; -REPT SPSIZ ;MODULE STACK STARTS HERE.
; -NLIST
; -WORD 0
; -LIST
; -FNDR
000224- MODDSP:
*****
404 ;THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
405 ;TO THE APPROPRIATE JSR IN THE JSR LINK TABLE AND SETS
406 ;THE ITERATION COUNT TO MATCH BAUD
407 000224- 012767 000100 177662 START: MOV #64,WDTO ;64 (10) WORDS FROM MEM PER ITERATION
408 000232- 012767 000100 177656 MOV #128,INTR ;64 INTERRUPTS PER ITERATION
409 000240- 012767 000200 177656 MOV SR1,R1 ;GET SR1 VALUE
410 000246- 016701 177544 BIC #17774,R1 ;MASK OUT SIZE PART
411 000252- 042701 177774 BEQ 85 ;DEFAULT SETTING
412 000256- 001406 7S: MOV #200,R0 ;SET UP MASK FOR 7-BIT
413 000260- 112700 000200 DPC R1 ;COUNT BITS MASKED
414 000264- 005301 7S: BEQ 85 ;EXIT WHEN DONE
415 000266- 001402 8S: ASR R0 ;SHIFT AND REPLICATE HIGH BIT
416 000270- 006200 8S: BR 7S ;AGAIN?
417 000272- 000774 002404 8S: MOV R0,SIZE ;SET SIZE MASK
418 000274- 010067 002404 CLR R0 ;INIT POINTER
419 000300- 005000 8S: MOV SR1,R1 ;GET SR1
420 000302- 016701 177510 ASR R1 ;SHIFT SR1 TO
421 000306- 006201 177510 ASR R1 ;GET BAUD RATE
422 000310- 006701 177510 BEQ 19S ;9600 BAUD SET
423 000312- 001405 18S: ADD #2,R0 ;ELSE RAMP POINTER
424 000314- 062700 000002 ROR R1 ;LOOK FOR SR1 BIT
425 000320- 006001 18S: BCS 19S ;LEAVE IF WE FOUND IT
426 000322- 103401 BR 19S ;ELSE DO IT AGAIN
427 000324- 000773 19S: MOV CNTRL(R0),ICONT ;SET UP ITER COUNT
428 000326- 016967 001556- 177502 17S: MOV VECTOR,R0 ;SET R0 TO POINT TO THE 1ST VECTOR
429 000334- 016700 177450 DVID1,R1 ;LOAD R1 WITH DEVICE SELECTION PARAMETER
430 000340- 016701 177450 MOV R1,DEVICE ;INITIALIZE TO RECORD ANY LINES DROPPED
431 000344- 019167 002500 MOV #JSRTAB,R2 ;SET UP R2 TO POINT TO JSR TABLE
432 000350- 012702 003052- 1S: MOV BR1,(R0)+ ;SHIFT SELECT BIT INTO "C" TABLE
433 000354- 006201 1S: ASR R1 ;RR IF NOT SELECTED
434 000356- 103020 MOV R2,(R0)+ ;SET UP RCVR INTR POINTER
435 000360- 010220 177424 MOV BR1,(R0)+ ;SET UP RCVR PRIORITY LEVEL
436 000362- 116720 TSTR (R0)+ ;MOVE POINTER
437 000366- 105720 ADD #10,R2 ;POINT R2 TO XNTR ENTRY IN JSR TABLE
438 000370- 062702 000010 MOV R2,(R0)+ ;SET UP XNTR INTR POINTER
439 000374- 019220 177410 MOV BR1,(R0)+ ;SET UP XNTR PRIORITY LEVEL
440 000376- 116720 TSTR (R0)+ ;MOVE POINTER
441 000402- 105720 ADD #10,R2 ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
442 000404- 062702 000010 CMP #JSRTAB+400,R2 ;IS THE POINTER AT THE END OF THE TABLE?
443 000406- 027202 003452- 2S: BNE 1S ;BR IF NOT
444 000414- 001357 3S: BR SETCSR ;GO SET UP CSR ADDRESSES
445 000416- 000405 3S: ADD #10,R0 ;UPDATE VECTOR POINTER
446 000420- 062700 000010 ADD #20,R2 ;UPDATE JSR TABLE POINTER
447 000424- 062702 000020 BR 2S ;GO CHECK FOR END OF TABLE
448 000430- 000767
449

```

```

450 ;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
451 ;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
452 SETCSR: MOV ADDR,R0 ;GET THE FIRST CSR ADDRESS INTO R0
453 MOV DVIDL,R1 ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
454 BNE 1S ;BR IF DVC OK
455 ;
456 1S: MOV #JSRTAB+4,R2 ;POINT R2 TO CSR ADDRESS ENTRY
457 2S: ASR R1 ;SHIFT SELECT BIT INTO "C"
458 BCC 4S ;BR IF LINE NOT SELECTED
459 MOV R0,(R2) ;PUT RCVR CSR ADDRESS IN TABLE
460 CMP (R0)+,(R0)+ ;GENERATE XMTX CSR ADDRESS IN R0
461 ADD #10,R2 ;POINT TO XMTX SLOT IN JSR TABLE
462 MOV R0,(R2) ;PUT XMTX CSR ADDRESS IN TABLE
463 CMP (R0)+,(R0)+ ;GENERATE NEXT RCVR CSR ADDRESS IN R0
464 ADD #10,R2 ;POINT TO RCVR SLOT IN JSR TABLE
465 3S: CMP #JSRTAB+404,R2 ;IS POINTER BEYOND END OF TABLE?
466 BNE 1S ;BR IF NOT
467 REST ;RR SET UP ACTIVE DEVICE TABLE.
468 4S: ADD #10,R0 ;UPDATE CSR ADDRESS
469 ADD #20,R2 ;UPDATE JSR TABLE POINTER
470 BR 3S ;GO TEST FOR END OF TABLE
471 ;
472 ;THIS ROUTINE CLEARS BUFFERS AND TABLES, INITIALIZES FLAGS, AND STARTS
473 ;UP ALL SELECTED LINES.
474 ;
475 000522* 005767 177306 RESTR: TST PASCNT ;HAVE WE BEEN THRU START?
476 000526* 001636 BEQ START ;BR THERE IF WE HAVEN'T
477 000530* 004767 002170 RFST: JSR PC,CLRBUF ;GO CLEAR XMTX, AND RCVR, BUFFERS
478 000534* 004767 002170 STUP1: JSR PC,DTAB ;SET UP THE ACTIVE DEVICE TABLE.
479 000540* 004767 002176 JSR PC,CLRTAB ;GO CLEAR TABLES AND QUEUES
480 000544* 005067 002140 CLR TXCNT ;CLEAR TX TOTAL INTERRUPT COUNTER.
481 000550* 005067 002136 CLR RXCNT ;CLEAR RX TOTAL INTERRUPT COUNTER.
482 000554* 005767 002140 MOV #RTAB,SVPTR ;INITIALIZE RCVR. STATUS TABLE POINTER
483 000562* 012767 002604 MOV #TQ,QPTR1 ;SET UP XMTX FIFO QUEUE POINTERS
484 000570* 012767 002604 MOV #TQ,QPTR2
485 000576* 016700 002244 MOV ACDEV,R0 ;GET COUNT OF ACTIVE DEVICES
486 000582* 016700 002244 1S: MOV DDEV,R0 ;GET AN ACTIVE LINE NO.
487 000586* 004767 001152 JSR PC,ETAADR ;GET BUILD CSR ADDRESS IN R3
488 000592* 005763 000002 TST 2(R3) ;READ RCVR DRR TO FLUSH DONE BIT
489 000596* 052713 000100 BIS #100,(R3) ;ENABLE RECEIVER INTERRUPTS
490 000602* 016700 000004 INCR #BUF(R2) ;ENABLE MAINT. MODE
491 000606* 005763 000002 MOVX #BUF(R2),6(R3) ;OUTPUT CHAR ONTO TX.
492 000612* 116263 002644 INCR TXCNT ;UP COUNT OF CHARS OUTPUT.
493 000616* 105267 002042 DCR TXCNT ;COUNT CHARACTERS OUTPUT ON THIS LINE
494 000622* 105267 002544 INCR #100,4(R3) ;ENABLE TX INTERRUPT.
495 000626* 005763 000100 DEC R0 ;COUNT ONE KICKED OFF
496 000632* 005300 BNE 1S ;GO TEST FOR NEXT ONE
497 000636* 100347 000006 2S: BPL 1S ;INITIALIZE COUNTER TO WAIT AT LEAST
498 000642* 012767 000006 MOV #6,CNTR ;30 SECONDS BEFORE TIMING OUT
499 ;
500 10S: CLR R4
501 2S: ;
502 BREAK$,REGIN ;TEMPORARY RETURN TO MONITOR...
503 BREAK$,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
504 CNPR #64, TXCNT+1 ;64 TRANSMITTER INTERRUPTS?
505 000712* 003004 8GT 3S ;NO- BRANCH TO WAIT

```

```

506 000714* 022767 000100 001770 CMP #64, RXCNT ;YES- 64 RECEIVER INTERRUPTS?
507 000722* 003405 BLE 4S ;YES- GO CHECK FOR ERRORS
508 000724* 005304 3S: DEC R4 ;TIMEOUT?
509 000726* 001362 BNE 2S ;NO- WAIT SOME MORE
510 000730* 005367 001760 DEC CNTR ;EACH PASS OF THE SMALL LOOP TAKES
511 ; AT LEAST 5 SECONDS
512 000734* 001356 BNE 10S ;BRANCH IF NOT DONE WITH 6 PASSES OF
513 000736* 000167 000306 4S: JMP ERRCHK ;THE SMALL COUNTER
514 ;
515 ;TRANSMITTER INTERRUPT SERVICE - ENTERED VIA APPROPRIATE JSR TABLE
516 ;ENTRY WITH R5 POINTING TO THE CSR ADDRESS - CONTENTS OF R5
517 ;GETS QUEUED UP IN FIFO QUEUE AND ROUTINE RETURNS CONTROL BACK TO
518 ;THE MONITOR VIA A PING TO DEPR
519 ;
520 000742* 010577 001752 INT: MOV #R5,QPTR1 ;STORE CONTENTS OF R5 IN THE QUEUE
521 000746* 052767 000002 ADD #2,QPTR1 ;UPDATE THE QUEUE POINTER
522 000754* 022767 002644 001736 CMP #TQ+40,QPTR1 ;POINTER AT END OF QUEUE?
523 000762* 001003 BNE 1S ;BR IF NOT
524 000764* 012767 002604 001726 MOV #TQ,QPTR1 ;RESET THE POINTER
525 000772* 012605 MOV (R0)+,R5 ;RESTORE THE OTHER CUV'S R5
526 ;
527 000774* 000004 000000 001002* 1S: ;
528 ;PIRQ$,BEGIN,TSERV ; QUEUE UP TO CONTINUE AT TSERV AND RTI
529 ;-----
530 ;DEFERRED XMTX SERVICE - THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS
531 ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0
532 TSERV: MOV #QPTR2,R0 ;GET POINTER FROM THE QUEUE
533 ADD #2,QPTR2 ;UPDATE THE QUEUE POINTER
534 001014* 022767 002644 001700 CMP #TQ+40,QPTR2 ;POINTER AT HIGH LIMIT
535 BNE 1S ;BR IF NOT
536 001024* 012767 002604 001670 MOV #TQ,QPTR2 ;RESET THE POINTER
537 001032* 012001 1S: MOV (R0)+,R1 ;MOV CSR ADDRESS INTO R1
538 001034* 011000 MOV R1,R0 ;MOV LINE # INTO R0
539 001036* 105267 001647 INCR TXCNT+1 ;COUNT TOTAL TX INTERRUPTS.
540 001042* 105260 002544 INCR CNT(R0) ;COUNT THE INTERRUPT
541 001046* 105711 RPL (R1) ;XMTX READY FLAG ASSERTED?
542 001050* 100011 BPL 4S ;RR IF NOT
543 001052* 122767 000100 001630 CNPR #64, TXCNT ;64 CHARACTERS TRANSMITTED?
544 001060* 061427 BEQ 5S ;YES- BRANCH TO EXIT
545 001062* 105260 002644 INCR #BUF(R0) ;GENERATE NEXT DATA BYTE
546 001066* 116061 002644 000002 MOVX #BUF(R0),2(R1) ;LOAD THE XMTX BUFFER
547 001074* 105267 001510 INCR TXCNT ;UP TOTAL COUNT OF CHARS OUTPUT.
548 001100* 105260 002544 INCR DCNT(R0) ;COUNT CHARACTERS OUTPUT ON THIS LINE
549 001104* 104400 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
550 001110* 010167 176764 4S: MOV R1,CSRA ;SAVE CSR ADDRESS
551 001114* 011167 176762 MOV (R1),ACSR ;SAVE CONTENTS OF THE CSR
552 001120* 142711 000100 BICB #100,(R1) ;DISABLE XMTX INTERRUPT
553 001124* 012767 000011 176754 MOV #11,ERRTYP ;ILLEGAL INTERRUPT
554 ;*****
555 001132* 104405 000000 000000 5S: ;*****
556 ;RODRS$,REGIN,NULL ;XMTX FALSE INTERRUPT
557 001140* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
558 ;
559 ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
560 ;STORES PERTINENT INFORMATION IN THE RECEIVER STATUS TABLE THAT WILL

```

```
562 ;RE CHECKED IF 64. CHARACTERS HAVE BEEN RECEIVED
563 RINT: MOV R0,-(R6) ;SAVE R0 AND R1 ON THE STACK
564 MOV R1,(R6)
565 CMP #64,RXCNT ;64 CHAR'S RECEIVED?
566 MOV SVPTR,R0 ;YES- BRANCH
567 MOV (R5),R1 ;NO- GET RCVR STATUS TABLE POINTER
568 MOV (R1),R1 ;GET RCVR CSR ADDRESS
569 MOV (R1),R1 ;SAVE THE CSR CONTENTS
570 ASRB -1(R0) ;SHIFT ERROR WORD
571 BISR 3(R1),-1(R0) ;GET MORE ERROR STATUS
572 MOV 2(R1),-1(R0) ;SAVE THE RCVD DATA
573 BISR 2(R1),-3(R0) ;REREAD CSR IN CASE ERROR SET BETWEEN THE
574 ;LAST TWO INSTRUCTIONS
575 MOV 2(R5),(R0)+ ;SAVE THE LINE NUMBER
576 MOV R0,SVPTR ;SAVE THE UPDATED STATUS TABLE POINTER
577 MOV 2(R5),R5 ;GET LINE NO. INTO R5
578 INC RCNT(R5) ;COUNT THE INTERRUPT FROM THIS LINE
579 INC RCNT ;INCREMENT RX INTERRUPT TOTAL COUNT.
580 MOV (R6)+,R1 ;RESTORE THE OTHER GUY'S REGISTER
581 MOV (R6)+,R0
582 MOV (R6)+,R5
583 RTI ;RETURN CONTROL BACK TO OTHER GUY
584
585 ;RECEIVER ERROR CHECKING AND CLEANUP ROUTINES
586
587 ;THIS ROUTINE DISABLES INTERRUPTS FROM ALL ACTIVE LINES
588 ERRCHK: MOV ACTDEV,R0 ;GET COUNT OF ACTIVE LINES
589 MOV DEVTAB(R0),R2 ;GET ACTIVE LINE NO.
590 PC,GETADR ;GO TO ADDRESS IN R3
591 BIC #100,(R3) ;TURN OFF RECEIVER.
592 BIC #100,4(R3) ;TURN OFF TRANSMITTER.
593 DEC R0 ;COUNT ONE GUY OFF
594 BPL 15 ;BR TIL ALL OFF
595
596 ;THIS ROUTINE SCANS THROUGH THE 64 ENTRY RECEIVER STATUS TABLE
597 ;CHECKING FOR AND REPORTING ANY ERRORS
598 CHK1: CMP RCNT,#64. ;MAKE SURE COUNT IS NO LARGER THAN TABLE
599 BLE 15
600 MOV #64,RXCNT
601 MOV #RSTAB,R1 ;GET STATUS TABLE POINTER
602 CLR R0 ;INDICATE NO HARDWARE FAILURES YET.
603 MOV 3(R1),R2 ;GET LINE NO. INTO R2
604 TST (R1) ;ERROR SET?
605 BPL 45 ;BRANCH IF NOT
606 JSR PC,RCVERR ;GO SETUP TO REPORT ERROR
607 CLR ERRTYP
608 ;*****
609 HDRS,REGIN,NULL ;CARRIER TRANS + RING + OVERRUN
610 BIT #BIT13,(R1) ;RING INDICATOR SET?
611 BEQ 45 ;BRANCH IF NOT
612 MGENS,BRGIN,RING ;ASCII MESSAGE CALL WITH COMMON HEADER
613 MOV #1,R1 ;SETUP TO DROP LINE
614 BR 3S ;SHIFT BIT TO ALLIGN WITH INDICATOR IN DVICE
615 ASL R1
616 DEC R2
617
```

```
618 BPL 3S
619 BIC R1,DVCF ;DROP THE LINE
620 JMP EMPS ;SKIP REST OF CHECKING SINCE RING INDICATOR
621 ;SET WILL CAUSE ALL COUNTS TO BE BAD
622 4S: TSTR (R1) ;POINT TO LO BYTE OF CSR
623 BML 52 ;BR IF DONE WAS SET
624 JSD PC,RCVERR ;SETUP FOR ERROR REPORT
625 MOV #1,ERRTYP ;ILLEGAL INTERRUPT
626 ;*****
627 HDRS,REGIN,NULL ;RECEIVER FALSE INTERRUPT
628 ;*****
629 INCR RBUF(R2) ;BUMP EXPECTED DATA
630 TST R0 ;HARDWARE ERRORS?
631 BNE 6S ;DO NOT REPORT DATA ERRORS THEN.
632 JSR SIZE,RBUF(R2) ;WASH OFF BITS <'S> TO CHECK ONLY
633 BICR SIZE,2(R1) ; SPECIFIED SIZE
634 CNPR RBUF(R2),2(R1) ;DID RCVD DATA CHECK OK?
635 BEQ 6S ;BR IF YES
636 JSR PC,DATBAD ;GO REPORT DATA ERROR
637 CMP (R1)+,(R1)+ ;POINT R1 TO NEXT TABLE ENTRY
638 DEC RCNT ;ALL CHARS RECEIVED CHECKED?
639 BNE 2S ;NO- GO CHECK NEXT ENTRY
640
641 ;THIS ROUTINE REPORTS ANY LINE RECEIVING AN INCORRECT NUMBER OF INTERRUPTS
642 CKLINS: MOV ACTDEV,R1 ;GET ACTIVE DEVICE COUNT
643 MOV DEVTAB(R1),R2 ;GET ACTIVE DEVICE LINE NO.
644 CNPR RCNT(R2),DCNT(R2) ;CORRECT NUMBER OF RCVR INTERRUPTS?
645 BEQ 4S ;BR IF YES
646 JSR PC,BADR ;GO REPORT BAD RCVR
647 CNPR ICNT(R2),DCNT(R2) ;CORRECT NUMBER OF XNTR INTERRUPTS?
648 BEQ 4S ;BR IF YES
649 JSR PC,RADT ;GO REPORT BAD XNTR
650 DEC R1 ;COUNT ONE GUY CHECKED
651 BPL 3S ;BR TIL ALL CHECKED
652 BR EMPS ;GO CHECK FOR END OF PASS
653
654 ;TABLE USED FOR DIFFERENT BAUD RATES
655 CNTTBL: 1450
656 1140
657 121
658 310
659 230
660 141
661 141
662 30
663 20
664 14
665 13
666 11
667 6
668
669 001610 000004 4
670 001612 000004 4 ;NOT A LEGAL SRI SETTING
671
672 RING: MRING
673 -1
```

```

674
675 001620 051045 047111 020107 MRING: .ASCIZ /XRING SET- BAD LINE DROPPED%/
676 001626 042523 026524 041040
677 001634 042101 046040 047111
678 001642 020105 051104 050111
679 001650 042526 022504 000
680
681
682
683
684 001655 004767 000102 ;ROUTINE TO REPORT BAD LINES (TOO MANY OR TOO FEW INTERRUPTS)
685 001666 116267 002564 176206 BADR: JSR PC,GETADR ;GO BUILD CSR ADDRESS
686 001674 116267 002524 176202 MOV R3,CSRA ;SAVE CSR ADDRESS
687
688 MOV R2,DCNT(R2),ACSR ;CHARACTERS XMTD
689 MOV R2,RCNT(R2),ASTAT ;# OF RCVR INTERRUPTS
690
691 MOV #14,ERRTYP ;WRONG # OF INTERRUPTS
692 *****
693 HRDERS,BEGIN,NULL ;INCORRECT NUMBER OF RCVR INTERRUPTS
694 *****
695 ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
696 ;TRANSMITTED, AND STATC VALUE IS # OF
697 ;RECEIVER INTERRUPTS
698
699 001710 104405 000000 000000 RTS PC ;RETURN TO CALLER
700
701 001720 004767 000040 BADT: JSR PC,GETADR ;GO BUILD CSR ADDRESS
702 001724 022323 CMP (R3)+,(R3)+ ;MAKE IT A XMTR CSR ADDRESS
703 001732 116267 176146 MOV R3,DCNT(R2),ACSR ;SAVE CSR ADDRESS
704 001740 116267 002564 176142 MOV R2,DCNT(R2),ACSR ;CHARACTERS XMTD
705 001746 116267 002544 176136 MOV R2,RCNT(R2),ASTAT ;# OF XMTR INTERRUPTS
706
707 MOV #14,ERRTYP ;WRONG # OF INTERRUPTS
708 *****
709 HRDERS,BEGIN,NULL ;INCORRECT NUMBER OF XMTR INTERRUPTS
710 *****
711 ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
712 ;TRANSMITTED, AND STATC VALUE IS # OF
713 ;RECEIVER INTERRUPTS
714
715 001762 000207 RTS PC ;RETURN TO CALLER
716
717 001764 010203 GETADR: MOV R2,R3 ;GET LINE NO.
718 001766 006303 ASL R3 ;BUILD CSR ADDRESS
719 001770 006303 ASL R3
720 001774 006703 ADD ADDR,R3
721 002000 000207 RTS PC ;RETURN TO CALLER
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785

```

```

730 002036 111567 176044 MOV R5,ASB ;SAVE GOOD DATA
731 002042 010567 176034 MOV R5,BADR ;SAVE ADDRESS OF GOOD DATA
732 *****
733 002046 104404 000000 DATERS,BEGIN ;DATA ERROR!!!
734 *****
735 002052 000207 RTS PC ;RETURN TO CALLER
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785

```

786 002736 001374
 787 002740 000207
 788
 789
 790 002742 012700 002124
 791 002746 005020
 792 002750 022700 002644
 793 002754 001374
 794 002756 000207
 795
 796
 797
 798
 799 002760 005000
 800 002762 005100
 801 002764 005001
 802 002766 005101
 803 002770 016702 000054
 804 002774 005200
 805 002776 022700 000020
 806 003002 001003
 807 003004 010167 000036
 808 003010 000200
 809 003012 005202
 810 003014 103367
 811 003016 005201 003026
 812 003020 110061
 813 003024 000763
 814
 815 003026 000010
 816 003046 000000
 817 003050 000000
 818
 819
 820
 821 003052 004567 176066
 822 003056 000000
 823 003060 000000
 824 003062 004567 175654
 825 003066 000000
 826 003070 000000
 827 003072 004567 176046
 828 003076 000000
 829 003100 000001
 830 003102 004567 175634
 831 003104 000000
 832 003110 000001
 833 003112 004567 176026
 834 003116 000000
 835 003120 000002
 836 003122 004567 175614
 837 003126 000000
 838 003130 000002
 839 003132 004567 176006
 840 003136 000000
 841 003140 000003

```

BNE PS ;BR TIL ALL CLEAR
RTS PC ;RETURN TO CALLER

;SUBROUTINE TO CLEAR TABLES AND QUEUES
CLR: MOV #RSTAR,R0 ;SET UP R0 TO POINT TO BEGINING
IS: CLR (R0)+ ;CLEAR A WORD
CMP #20,R0 ;END?
BNE PS ;BR IF NOT
RTS PC ;RETURN TO CALLER

;THIS ROUTINE SETS UP AN ACTIVE DEVICE TABLE TO REMEMBER HOW MANY
;AND WHICH LINES WERE ACTIVE DURING TEST - IT IS USED DURING THE
;ERROR CHECKING ROUTINES FOR VARIOUS PURPOSES
DTAB: CLR R0 ;SET UP R0 AS TOTAL LINE COUNTER
COM R0 ;INITIALLY SET TO MINUS ONE
CLR R1 ;SET UP R1 AS ACTIVE LINE COUNTER
COM R1 ;INITIALLY SET TO MINUS ONE
MOV DVICE,R2 ;GET DEVICE SELECTION PARAMETER
IS: INC R0 ;COUNT ONE DEVICE
CMP #20,R0 ;16 LINES CHECKED?
BNE PS ;BR IF NOT
MOV R1,ACTDEV ;SAVE THE COUNT OF ACTIVE LINES
RTS ;RETURN TO CALLER
2S: ASR R0 ;SHIFT SELECT BIT INTO "C"
BCC IS ;BR IF NOT SELECTED
INC R1 ;COUNT ACTIVE LINE
MOV# R0,DEVTAB(R1) ;STORE ACTIVE LINE NO. IN THE TABLE
BR ;GO TEST NEXT LINE

DEVTAB: .BLK# 8. ;16 BYTE ACTIVE DEVICE TABLE
ACTDEV: OPEN ;STORES COUNT OF NO. OF ACTIVE LINES MINUS ONE
DVICE: OPEN ;DEVICE SELECTION INDICATOR

;JSR LINK TABLE CONSISTING OF 32 JSR'S (16 RCVR AND 16 XMTR) THAT
;LINK THE INTERRUPTS TO THE COMMON SERVICE ROUTINES
JSRTAB: JSR R5,TINT ;RECEIVER LINK FOR LINE 0
0 ;SET UP WITH RCVR CSR ADDRESS
0 ;LINE NUMBER
JSR R5,TINT ;XMTR LINK FOR LINE 0
0 ;SET UP WITH XMTR CSR ADDRESS
0 ;LINE NUMBER
JSR R5,PINT ;LINK FOR LINE 1
1
1 JSR R5,TINT
1
1 JSR R5,RINT ;LINK FOR LINE 2
0
2 JSR R5,TINT
2
2 JSR R5,RINT ;LINK FOR LINE 3
0
3

```

842 003142 004567 175574
 843 003146 000000
 844 003150 000003
 845 003152 004567 175766
 846 003156 000000
 847 003160 000004
 848 003162 004567 175554
 849 003166 000000
 850 003170 000004
 851 003172 004567 175746
 852 003176 000000
 853 003200 000005
 854 003202 004567 175534
 855 003206 000000
 856 003210 000005
 857 003212 004567 175726
 858 003216 000000
 859 003220 000006
 860 003222 004567 175514
 861 003226 000000
 862 003230 000006
 863 003232 004567 175706
 864 003236 000000
 865 003240 000007
 866 003242 004567 175474
 867 003246 000000
 868 003250 000007
 869 003252 004567 175666
 870 003256 000000
 871 003260 000010
 872 003262 004567 175454
 873 003266 000000
 874 003270 000010
 875 003272 004567 175646
 876 003276 000000
 877 003300 000011
 878 003302 004567 175434
 879 003306 000000
 880 003310 000011
 881 003312 004567 175626
 882 003316 000000
 883 003320 000012
 884 003322 004567 175414
 885 003326 000000
 886 003330 000012
 887 003332 004567 175606
 888 003336 000000
 889 003340 000013
 890 003342 004567 175374
 891 003346 000000
 892 003350 000013
 893 003352 004567 175566
 894 003356 000000
 895 003360 000014
 896 003362 004567 175354
 897 003366 000000

```

JSR R5,TINT
0
3 JSR R5,RINT ;LINK FOR LINE 4
4
4 JSR R5,TINT
4
5 JSR R5,RINT ;LINK FOR LINE 5
5
5 JSR R5,TINT ;LINK FOR LINE 6
6
6 JSR R5,TINT
6
7 JSR R5,RINT ;LINK FOR LINE 7
7
7 JSR R5,TINT ;LINK FOR LINE 10
10
10 JSR R5,TINT
10
11 JSR R5,RINT ;LINK FOR LINE 11
11
11 JSR R5,TINT ;LINK FOR LINE 12
12
12 JSR R5,RINT ;LINK FOR LINE 13
13
13 JSR R5,TINT
13
14 JSR R5,RINT ;LINK FOR LINE 14
14
0

```

898	003370	000014		14		
899	003372	004567	175546	JSR	R5,RINT	;LINK FOR LINE 15
900	003376	000000		0		
901	003400	000015		15		
902	003402	004567	175334	JSR	R5,TINT	
903	003406	000000		0		
904	003410	000015		15		
905	003412	004567	175526	JSR	R5,RINT	;LINK FOR LINE 16
906	003416	000000		0		
907	003420	000016		16		
908	003422	004567	175314	JSR	R5,TINT	
909	003426	000000		0		
910	003430	000016		16		
911	003432	004567	175506	JSR	R5,RINT	;LINK FOR LINE 17
912	003436	000000		0		
913	003440	000017		17		
914	003442	004567	175274	JSR	R5,TINT	
915	003446	000000		0		
916	003450	000017		17		
917						
918	000001					

.END

CROSS REFERENCE TABLE -- USPR SYMROLS														
ACSR	000102R	386#	511*	685*	701*	741*								
ACTDEV	003046R	485	588	642	807*	816#								
ADDR	000006R	352#	452	718										
ADDR22=	001000	404#												
ASB	000108R	390#	730*											
ASTAT	000104R	388#	686*	702*										
AWAS	000110R	391#	724*											
BADR	001656R	646	683#											
BADT	001720R	649	698#											
BEGIN	000000R	349#	455	502	503	527	549	555	558	609	613	627	690	706
		733	746											
BIT0 =	000001	404#												
BIT1 =	000002	404#												
BIT10 =	002000	404#												
BIT11 =	004000	404#												
BIT12 =	010000	404#												
BIT13 =	020000	404#												
BIT14 =	040000	404#	611											
BIT15 =	100000	404#												
BIT2 =	000004	404#												
BIT3 =	000010	404#												
BIT4 =	000020	404#												
BIT5 =	000040	404#												
BIT6 =	000100	404#												
BIT7 =	000200	404#												
BIT8 =	000400	404#												
BIT9 =	001000	404#												
BREAKS=	104407	404#	502	503										
BR1	000917R	354#	436	440										
BR2	000013R	355#												
BTONS =	104421	404#												
CDATAS=	104412	404#												
CHKL	001302R	595#												
CKLINS	001510R	642#												
CLRRUP	002724R	477	783#											
CLRTAR	002742R	479	790#											
CNTR	002714R	498*	510*	776#										
CNTTBL	001556R	428	655#											
CONFIG	000056R	374#												
COUNT	002706R	772#												
CSRA	000100R	388#	550*	684*	700*	723*	740*							
DATERAD	002002R	636	722#											
DATCKS=	104411	404#												
DATFRS=	104404	404#	733											
DCNT	002564R	494*	548*	644	647	685	701	762#						
DEVTAR	002026R	486	589	643	812*	815#								
DTR	002760R	478	799#											
DVICE	003050R	431*	619*	803	817#									
DVIDI	000014R	356#	430	453										
ENDITS=	104413	404#	746											
ENDS =	104410	404#	455											
ENPS	002074R	620	652	745#										
ERRCHK	001250R	514	548#											
ERRTP	000106R	389#	533*	607*	688*	704*								
EXITS =	104400	404#	549	558	698	714#	722	739						
GETADR	001764R	487	590	683										

