

# DC11

OFF LINE DIAGNOSTIC TEST  
CZDCAC0

AH-8430C-MC  
COPYRIGHT 70-78  
FICHE 1 OF 1

DEC 1978  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames, each displaying diagnostic test data. The data is organized into columns and rows, with some frames containing headers and others containing numerical or alphanumeric values. The frames are arranged in a grid that is approximately 12 frames wide and 15 frames high. The data appears to be organized into several columns, with some frames containing headers and others containing numerical or alphanumeric values. The frames are arranged in a grid that is approximately 12 frames wide and 15 frames high. The data appears to be organized into several columns, with some frames containing headers and others containing numerical or alphanumeric values.



.REM @

IDENTIFICATION

PRODUCT CODE: AC-8428C-MC  
PRODUCT NAME: CZDCACO DC11 OFLNE DIAG TST  
DATE: FEB 1978  
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.

NO RESPONSIBIOITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR IT'S AFFILIATED COMPANIES.

COPYRIGHT (C) 1975,1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:  
DIGITAL PDP UNIBUS MASSBUS  
DEC DECUS DECTAPE

PAGE 1

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR THE DC-11 (ASYNCHRONOUS MODEM INTERFACE), CZDCA (DC-11 OFF LINE TESTS) AND CZDCB (DC-11 ON LINE TESTS). THE OFF LINE TESTS TEST ALL DC11 LOGIC AND MAY BE USED TO INDIVIDUALLY TEST UP TO 32 DC-11'S. THE OFF LINE TESTS DO NOT REQUIRE THE USE OF A MODEM, HOWEVER A SPECIAL JUMPER CONNECTOR IS REQUIRED. THE ON LINE TESTS ARE ESSENTIALLY DATA RELIABILITY TESTS REQUIRING THE USE OF MODEMS AND A SUITABLE TERMINAL DEVICE.

THIS DOCUMENT DESCRIBES THE OFF LINE TESTS.

THE AVAILABLE TESTS ARE:

PRG0	INPUT/OUTPUT LOGIC TESTS
PRG1	TRANSMITTER SCOPE LOOP
PRG2	RECEIVER SCOPE LOOP
PRG3	SINGLE CHARACTER MAINT. MODE DATA TEST
PRG4	SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST

2. REQUIREMENTS

2.1 EQUIPMENT

- A. PDP 11/20 SYSTEM
- B. DC11
- C. SPECIAL JUMPER CONNECTOR (SEE DC11 MAINTENANCE MANUAL FOR DETAILED DESCRIPTION)

2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (4K) EXCEPT THAT AREA RESERVED FOR THE BOOTSTRAP AND ABSOLUTE LOADERS.

3. LOADING PROCEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

4. USE PROCEDURE  
BEFORE STARTING ANY OF THE SELECTABLE PROGRAMS MAKE SURE  
THAT THE TTY IS IN REMOTE MODE; AND THAT THE PROGRAM SELECTED  
IS A LEGAL PROGRAM, IE. SR 0-2=0-4, OTHERWISE PROGRAM OP-  
ERATION IS UNSPECIFIED. RELOAD PROGRAM AND START OVER.
- 4.1 PRGO INPUT/OUTPUT LOGIC TESTS
- A. LOAD ADDRESS = 000200 (RESTART LOAD ADDR. = 000200)  
LOAD SR 0-2 = 0, AND PRESS START SWITCH.  
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED.  
'PRGO-INPUT-OUTPUT LOGIC TESTS. DISCONNECT DC11 FROM MODEM  
AND CONNECT JUMPER TO CABLE'  
DISCONNECT THE DC11 FROM THE MODEM AND INSERT THE JUMPER CON-  
NECTOR IN THE MODEM END OF THE CABLE, AND PRESS CONTINUE.  
NOTE, IF THE CABLE IS LEFT CONNECTED TO THE MODEM THE FOL-  
LOWING TESTS WILL FAIL:  
AT5, AT34-AT42, AT44, AND AT144
- B. THE PROGRAM WILL NOW REQUEST THE LINE # (8) YOU WISH TO  
TEST. LOAD THE LINE # AS REQUESTED AND PRESS CONTINUE.  
LINE NUMBER REFERS TO THE ADDRESSES TO WHICH THE DC11 RESPONDS.
- |        |        |         |        |         |        |         |        |
|--------|--------|---------|--------|---------|--------|---------|--------|
| LINE 0 | 77400X | LINE 10 | 77410X | LINE 20 | 77420X | LINE 30 | 77430X |
| LINE 1 | 77401X | LINE 11 | 77411X | LINE 21 | 77421X | LINE 31 | 77431X |
| LINE 2 | 77402X | LINE 12 | 77412X | LINE 22 | 77422X | LINE 32 | 77432X |
| LINE 3 | 77403X | LINE 13 | 77413X | LINE 23 | 77423X | LINE 33 | 77433X |
| LINE 4 | 77404X | LINE 14 | 77414X | LINE 24 | 77424X | LINE 34 | 77434X |
| LINE 5 | 77405X | LINE 15 | 77415X | LINE 25 | 77425X | LINE 35 | 77435X |
| LINE 6 | 77406X | LINE 16 | 77416X | LINE 26 | 77426X | LINE 26 | 77436X |
| LINE 7 | 77407X | LINE 17 | 77417X | LINE 27 | 77427X | LINE 27 | 77437X |
- C. THE PROGRAM WILL TYPE OUT INSTRUCTIONS TO SET IN THE DESIRED  
SR OPTIONS. PRESS CONTINUE WHEN THE OPTIONS ARE IN THE SR.  
THE AVAILABLE OPTIONS ARE:  
SR 0-6 ROUTINE TO BE RUN (IF ENABLED BY SR9)  
SR8 DISABLE STALL MODE  
SR9 LOOP SELECTED ROUTINE  
SR10 HALT AT END OF PROGRAM  
SR11 INHIBIT ITERATION  
SR13 INHIBIT PRINTOUT  
SR14 SCOPE  
SR15 HALT ON ERROR.
- D. THE PROGRAM WILL NOW BEGIN TESTING THE DC11 YOU SELECTED.
- E. REFER TO SECTION 5.1.2 FOR ERROR DESCRIPTION
- F. AFTER ONE COMPLETE PASS PRGEND WILL BE TYPED OUT

PAGE 3

4.2 PRG1 - TRANSMITTER SCOPE LOOP

A. LOAD ADDRESS = 000200  
LOAD SR 0-2 = 1, AND PRESS START SWITCH.  
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND  
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS RE-  
QUESTED AND PRESS CONTINUE.

B.  
THE PROGRAM WILL REQUEST THE DC11 LINE PARAMETERS. LOAD THE  
PARAMETERS AS REQUESTED AND PRESS CONTINUE.

C. THE PROGRAM WILL REQUEST A CHARACTER CODE, AND A DELAY  
TIME. THE CHARACTER CODE IS THE DATA THE DC11 WILL TRANSMIT  
AND THE DELAY IS THE TIME ELAPSED BETWEEN SUCCESSIVE TRANS-  
MISSIONS OF ONE CHARACTER. PRESS CONTINUE WHEN THIS IS DONE.

D. THE PROGRAM WILL RUN WITHOUT ERROR OR END TYPEOUTS.

4.3 PRG2 - RECEIVER SCOPE LOOP

A. LOAD ADDRESS = 000200  
LOAD SR 0-2 = 2, AND PRESS START.  
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND  
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS REQ-  
UESTED AND PRESS CONTINUE.

B. THE PROGRAM WILL REQUEST THE DC11 LINE PARAMETERS. LOAD THE  
PARAMETERS AS REQUESTED AND PRESS CONTINUE.

C. THE PROGRAM WILL REQUEST A TEST CHARACTER CODE, AND A DELAY  
TIME. THE CHARACTER CODE IS THE DATA THAT THE DC11 WILL BE  
TRANSMITTING AND THE DELAY IS THE ELAPSED TIME BETWEEN SUCCES-  
SIVE CHARACTERS. PRESS CONTINUE WHEN THIS IS DONE.

D. THE PROGRAM WILL NOW RUN WITHOUT ERROR OR END TYPEOUTS.

PAGE 4

4.4 PRG3 - SINGLE CHARACTER MAINT MODE DATA TEST

- A. LOAD ADDRESS = 000200  
LOAD SR 0-2 = 3, AND PRESS START.  
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND  
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS REQ-  
UESTED AND PRESS CONTINUE.
- B. THE PROGRAM WILL REQUEST THE DC11 LINE PARAMETERS. LOAD THE  
PARAMETERS AS REQUESTED AND PRESS CONTINUE.
- C. THE PROGRAM WILL REQUEST A TEST CHARACTER. LOAD THE TEST CHAR-  
ACTER AND PRESS CONTINUE.
- D. THE PROGRAM WILL NOW RUN CONTINUOUSLY REPORTING ANY DATA FAIL-  
URES.

4.5 PRG4 - SPECIAL BINARY COUNT MAINT. MODE DATA TEST

- A. LOAD ADDRESS - 000200  
LOAD SR 0-2 = 4, AND PRESS START.  
THE DIAGNOSTIC WILL IDENTIFY THE PROGRAM YOU SELECTED, AND  
REQUEST THE LINE # YOU WISH TO TEST. LOAD THE LINE # AS REQ-  
UESTED AND PRESS CONTINUE.
- B. THE PROGRAM WILL NOW REQUEST THE DC11 LINE PARAMETERS. LOAD THE  
PARAMETERS AS REQUESTED AND PRESS CONTINUE.
- C. THE PROGRAM WILL BEGIN TESTING THE LINE YOU SELECTED.  
AND REPORT ANY DATA ERRORS.

5. PROGRAM DESCRIPTIONS

5.1 PRG0 - INPUT/OUTPUT LOGIC TESTS

THE INPUT/OUTPUT LOGIC TESTS CONSIST OF 103(8) ROUTINES WHICH  
MAY BE RUN IN SEQUENTIAL ORDER OR INDIVIDUALLY LOOPED (SEE  
SECT 4.1, C FOR SWITCH SETTINGS). THE JUMPER CONNECTOR MUST  
BE INSERTED BEFORE STARTING.

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
AT0-AT3 AT4-AT42	ADDRESSABILITY OF CSRS & DBRS DIDDLES ALL BITS IN THE CSRS AND CHECKS THAT THEY CAN BE READ/WITTEN PROPERLY.
AT43-AT44 AT45-AT51 AT52-AT63	PROPER OPERATION OF RESET INSTRUCTION PROPER OPERATION OF READY BIT PROPER OPERATION OF CHAR LENGTH, SPEED CONTROL AND STOP CODE BITS.
AT64	PROPER OPERATION OF DATA OVERFLOW BIT
AT65-AT74	PROPER OPERATION OF INTERRUPTS
AT75	DATA OVERFLOW CLEARS DONE
AT76	ERROR CAUSES INTERRUPT
AT77	PROPER OPERATION OF PARITY BIT
AT100-AT137	DATA TESTS THESE TESTS TEST ALL POSSIBLE COMBINATIONS OF CHARACTER LENGTH SPEED AND STOP CODES USING MAINT. MODE.
AT140	DATA TEST HIGH SPEED (JUMPER)
AT141	PROPER OPERATION OF BREAK BIT

PAGE 6

### 5.1.2 ERROR DESCRIPTION

IF A ROUTINE FAILS AND THE INHIBIT PRINTOUT SWITCH IS NOT ENABLED (SR13) A PRINTOUT RESULTS. THE PRINTOUT FORMAT IS:

P(PROG#) T(ROUTINE#) PC-(PC OF ERROR CALL) AND AN ADDITIONAL/MESSAGE (IF APPLICABLE)

P00 T005 PC=XXXX INDICATING THAT TXCSR BIT 1 WAS SET (SHOULD'VE BEEN CLEAR)

P00 T122 PC XXXX DATA ERR S/B:---WAS---  
INDICATING A DATA ERROR WHEN DC11 PARAMETERS WERE SET AT CHAR LENGTH=6 SPEED=00, AND STOP CODE=1

TO RESUME TESTING PRESS CONTINUE. IF ROUTINES 65 OR 71 FAIL DUE TO A BAD TRAP VECTOR, I.E. THE VECTOR PROVIDED BY THE INTERRUPTING DC11 IS INCORRECT THE PROGRAM WILL HALT AND DISPLAY THE VECTOR+2 THAT WAS PROVIDED BY THE INTERRUPTING DC11. TO RECOVER FROM THIS TYPE OF ERROR IT WILL BE NECESSARY TO PUT INTO THE INCORRECT VECTOR ADDRESS THE ADDRESS TO RUN THE ROUTINE. I.E. ADDRESS ATAA AND AXAA FOR ROUTINES 65 AND 71 RESPECTIVELY.

### 5.1.3 JUMPER CONNECTOR

THE JUMPER CONNECTOR TESTS THOSE F/F'S, GATES (RING INDICATOR, CARRIER TRANSITION, CLEAR TO SEND, AND SUPERVISORY RECEIVE DATA) WHICH CANNOT BE TESTED UNLESS A DATA SET IS ACTUALLY CONNECTED TO THE DC11. IN ADDITION TO TESTING DC11 LOGIC THE JUMPER ALSO TESTS CABLE WIRING TO/FROM THE DC11/DATA SET. THE FOLLOWING TESTS WILL FAIL IF THE CABLE IS NOT INSTALLED IN THE DC11:

AT5,AT34-AT42,AT44  
AT140 WILL LOOP CONTINUOUSLY

IF THE JUMPER IS REMOVED FROM THE END OF THE CABLE AND THE CABLE IS LEFT CONNECTED TO THE DC11 THE ABOVE TESTS WILL FAIL WITH THE PROBABLE EXCEPTIONS OF AT35 AND AT36.



PAGE 7

5.2 PRG1-TRANSMITTER SCOPE LOOP

THE PURPOSE OF PRG1 IS TO ALLOW SCOPING OF TRANSMITTER FUNCTIONS IN A RUN CONDITION USING USER SPECIFIED DC11 PARMATERS AND DATA. NO ERROR PRINTOUTS ARE PROVIDED.

5.3 PRG2-RECEIVER SCOPE LOOP

THE PURPOSE OF PRG2 IS TO ALLOW SCOPING OF RECEIVER FUNCTIONS IN A RUN CONDITION USING USER SPECIFIED DC11 PARAMETERS AND DATA. NO ERROR PRINTOUTS ARE PROVIDED.

5.4 PRG3-SINGLE CHARACTER MAINT MODE DATA TEST

PRG3 TRANSMITS, RECEIVES AND CHECKS RECEIVED DATA USING USER SPECIFIED DC11, DC11 PARAMETERS, AND DATA.

5.4.1 ERROR PRINTOUTS

SELF EXPLANATORY ERROR PRINTOUTS ARE PROVIDED.

5.5 PRG4-SPECIAL BINARY COUNT MAINT MODE DATA TEST

PRG4 IS THE SAME AS PRGO ROUTINE 77 AND 100 EXCEPT THAT THE USER SPECIFIES DC11 RUNNING PARAMETERS.

5.5.1 ERROR PRINTOUTS

SELF EXPLANATORY PRINTOUTS ARE PROVIDED.

6.0 POWER FAIL

A POWER FAIL ROUTINE IS INCLUDED IN THE PROGRAM. WHEN THE POWER FAILS THE PROGRAM WILL AUTOMATICALLY RESTART USING THE PRESENT SR OPTIONS AND THE LINE PREVIOUSLY SELECTED. NOTE: THE POWER MAY FAIL WHEN THE PROGRAM IS EXECUTING A 'RESET' INSTRUCTION, IN THIS CASE OPERATOR INTERVENTION IS NEEDED TO PRESS CONTINUE. AN ERROR TYPEOUT RESULTS AND WILL TYPE THE PROGRAM #, THE ROUTINE THAT WAS RUNNING AT THE TIME THE POWER FAILED (PROGRAM 0 ONLY), AND THE PC OF THE POWER FAIL ERROR CALL.

P(PRG#) T(ROUTINE #) PC (ADDRESS OF ERROR CALL)



```

408 .LIST SEQ,BIN
409 :DC11 DIAGNOSTIC PROGRAM (OFF LINE TESTS)
410 :COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
411 :PRG0- INPUT-OUTPUT LOGIC TESTS
412 :PRG1- TRANSMITTER SCOPE LOOP
413 :PRG2- RECEIVER SCOPE LOOP
414 :PRG3- SINGLE CHARACTER MAINTENANCE MODE DATA TEST
415 :PRG4- SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST
416 :STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
417 :
418 :SR15- HALT ON ERROR
419 :SR14- SCOPE.
420 :SR13- INHIBIT PRINTOUT
421 :SR12- INHIBIT TRACE
422 :SR11- INHIBIT ITERATION.
423 :SR10- HALT AT END OF PROGRAM
424 :SR9- SELECT ROUTINE.
425 :SR8- DISABLE STALL MODE AND RUN FULL SPEED.
426 :SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE SELECTED.
427 :DATA TEST PARAMETERS

```

	CHAR	LENGTH	SPEED	STOP CODE
429	:			
430	:NOTE0	8	00	2
431	:NOTE1	7	00	2
432	:NOTE2	6	00	2
433	:NOTE3	5	00	2
434	:NOTE4	8	01	2
435	:NOTE5	7	01	2
436	:NOTE6	6	01	2
437	:NOTE7	5	01	2
438	:NOTE10	8	10	2
439	:NOTE11	7	10	2
440	:NOTE12	6	10	2
441	:NOTE13	5	10	2
442	:NOTE14	8	11	2
443	:NOTE15	7	11	2
444	:NOTE16	6	11	2
445	:NOTE17	5	11	2
446	:NOTE20	8	00	1
447	:NOTE21	7	00	1
448	:NOTE22	6	00	1
449	:NOTE23	5	00	1
450	:NOTE24	8	01	1
451	:NOTE25	7	01	1
452	:NOTE26	6	01	1
453	:NOTE27	5	01	1
454	:NOTE30	8	10	1
455	:NOTE31	7	10	1
456	:NOTE32	6	10	1
457	:NOTE33	5	10	1
458	:NOTE34	8	11	1
459	:NOTE35	7	11	1
460	:NOTE36	6	11	1
461	:NOTE37	5	11	1

Address	Bin	Seq	Instruction	Description
462			.LIST	BIN,SEQ
463		000000	.=0	
464	000000	000002	.+2	:UNASSIGNED TRAP
465	000002	000000	HALT	
466	000004	000006	MACHER: .+2	:SP OVERFLOW, BUS ERROR TRAP
467	000006	000000	HALT	
468	000010	000012	.+2	:RESERVED INSTRUCTION TRAP
469	000012	000000	HALT	
470	000014	000016	.+2	:TRACE TRAP
471	000016	000000	HALT	
472	000020	000022	.+2	:TRAP TO CALL IOX
473	000022	000002	2	
474	000024	000026	.+2	:POWER FAIL TRAP
475	000026	000000	HALT	
476	000030	002112	EMTINT	:EMT TRAP
477	000032	000340	PRTY7	
478	000034	000036	.+2	
479	000036	000000	HALT	
480	000040	000042	.+2	
481	000042	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
482	000044	000046	.+2	
483	000046	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
484	000050	000052	.+2	
485	000052	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
486	000054	000056	.+2	
487	000056	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
488	000060	000062	.+2	
489	000062	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
490	000064	000066	.+2	
491	000066	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
492	000070	000072	.+2	
493	000072	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
494	000074	000076	.+2	
495	000076	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
496	000100	000102	.+2	
497	000102	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
498	000104	000106	.+2	
499	000106	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
500	000110	000112	.+2	
501	000112	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
502	000114	000116	.+2	
503	000116	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
504	000120	000122	.+2	
505	000122	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
506	000124	000126	.+2	
507	000126	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
508	000130	000132	.+2	
509	000132	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
510	000134	000136	.+2	
511	000136	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
512	000140	000142	.+2	
513	000142	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
514	000144	000146	.+2	
515	000146	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.
516	000150	000152	.+2	
517	000152	000000	HALT	:TRAPPED TO PREVIOUS ADDRESS.



518	000154	000156	.+2	
519	000156	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
520	000160	000162	.+2	
521	000162	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
522	000164	000166	.+2	
523	000166	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
524	000170	000172	.+2	
525	000172	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
526	000174	000176	.+2	
527	000176	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
528	000200	000202	.+2	
529	000202	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
530	000204	000206	.+2	
531	000206	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
532	000210	000212	.+2	
533	000212	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
534	000214	000216	.+2	
535	000216	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
536	000220	000222	.+2	
537	000222	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
538	000224	000226	.+2	
539	000226	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
540	000230	000232	.+2	
541	000232	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
542	000234	000236	.+2	
543	000236	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
544	000240	000242	.+2	
545	000242	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
546	000244	000246	.+2	
547	000246	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
548	000250	000252	.+2	
549	000252	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
550	000254	000256	.+2	
551	000256	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
552	000260	000262	.+2	
553	000262	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
554	000264	000266	.+2	
555	000266	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
556	000270	000272	.+2	
557	000272	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
558	000274	000276	.+2	
559	000276	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
560	000300	000302	.+2	
561	000302	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
562	000304	000306	.+2	
563	000306	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
564	000310	000312	.+2	
565	000312	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
566	000314	000316	.+2	
567	000316	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
568	000320	000322	.+2	
569	000322	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
570	000324	000326	.+2	
571	000326	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.
572	000330	000332	.+2	
573	000332	000000	HALT	;TRAPPED TO PREVIOUS ADDRESS.

```
574 000334 000336 .+2  
575 000336 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
576 000340 000342 .+2  
577 000342 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
578 000344 000346 .+2  
579 000346 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
580 000350 000352 .+2  
581 000352 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
582 000354 000356 .+2  
583 000356 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
584 000360 000362 .+2  
585 000362 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
586 000364 000366 .+2  
587 000366 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
588 000370 000372 .+2  
589 000372 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.  
590 000374 000376 .+2  
591 000376 000000 HALT ;TRAPPED TO PREVIOUS ADDRESS.
```

```
592  
593 ;EQUATE STATEMENTS  
594 177570 SR=177570  
595 177776 CC=177776  
596 177776 PSW=177776  
597 001076 SPBOT=1076  
598 000240 NOP=240  
599 000000 OPEN=0  
600 100000 MANUAL=BIT15  
601 100000 BIT15=100000  
602 040000 BIT14=40000  
603 020000 BIT13=20000  
604 010000 BIT12=10000  
605 004000 BIT11=4000  
606 002000 BIT10=2000  
607 001000 BIT9=1000  
608 000400 BIT8=400  
609 000200 BIT7=200  
610 000100 BIT6=100  
611 000040 BIT5=40  
612 000020 BIT4=20  
613 000010 BIT3=10  
614 000004 BIT2=4  
615 000002 BIT1=2  
616 000001 BIT0=1  
617 005726 POPSP=5726 ;POP THE STACK. SAME AS TST (6)+  
618 022626 POPSP2=022626 ;POP STACK TWICE. SAME AS CMP (6)+,(6)+  
619 000340 PRTY7=340 ;PRIORITY LEVEL DEFINITIONS  
620 000300 PRTY6=300  
621 000240 PRTY5=240  
622 000200 PRTY4=200  
623 000140 PRTY3=140  
624 000100 PRTY2=100  
625 000040 PRTY1=40  
626 000000 PRTY0=0  
627 104000 TYPE=EMT+0  
628 104001 TYPES=EMT+1  
629 104002 STALL=EMT+2
```

630	104003	ERROR=EMT+3
631	104004	WATCHK=EMT+4
632	104005	EHALT=EMT+5
633	104006	STRXV=EMT+6
634	104007	STTXV=EMT+7
635	104010	EHALT=EMT+10
636	104011	SRESET=EMT+11
637	104012	SCOPE=EMT+12
638	104013	SAVREG=EMT+13
639	104014	RSTREG=EMT+14
640	104015	ERROR1=EMT+15
641	104016	DELAY=EMT+16
642	000007	BELL=007
643	177777	ATLAST=-1

644						
645		000200		.=200		
646	000200	000137	001562	JMP	START	:GO TO START OF PROGRAM.
647		001100		.=1100		
648	001100	174000		RXCSR:	174000	:RECEIVER CSR
649	001102	174002		RXBUF:	174002	:RECEIVER BUFFER
650	001104	174004		TXCSR:	174004	:TRANSMITTER CSR
651	001106	174006		TXBUF:	174006	:TRANSMITTER BUFFER
652	001110	000000		RXVTR:	OPEN	:RECEIVER VECTOR
653	001112	000240		RXLVL:	PRTY5	:RECEIVER PRIORITY LEVEL
654	001114	000304		TXVTR:	304	:TRANSMITTER VECTOR
655	001116	000240		TXLVL:	PRTY5	:TRANSMITTER PRIORITY LEVEL
656	001120	177560		TKS:	177560	:LSR CSR
657	001122	177562		TKB:	177562	:LSR BUFFER
658	001124	177564		TPS:	177564	:LSP CSR
659	001126	177566		TPB:	177566	:LSP BUFFER
660	001130	000060		TKVTR:	60	:LSR INTERRUPT VECTOR
661	001132	000200		TKLVL:	PRTY4	:LSR PRIORITY LEVEL
662	001134	000064		TPVTR:	64	:LSP INTERRUPT VECTOR
663	001136	000200		TPLVL:	PRTY4	:LSP PRIORITY LEVEL
664	001140	000000		PRGNUM:	OPEN	:CONTAINS CURRENT PROGRAM#
665	001142	000000		KSTART:	OPEN	:CURRENT PROGRAM START ADDRESS.
666	001144	000000		CURTST:	OPEN	:CONTAINS ADDR OF CURRENT TEST.
667	001146	000000		RTNNO:	OPEN	:CONTAINS CURRENT TEST #.
668	001150	000000		NXTST:	OPEN	:CONTAINS ADDR OF NEXT TEST.
669	001152	000000		ICTR:	OPEN	:CONTAINS CURRENT ITERATION COUNT
670	001154	000000		SCOPTR:	OPEN	:CONTAINS CURRENT SCOPE POINTER.
671	001156	003726		PRGTAB:	PRG0	:PRG0 START ADDRESS
672	001160	014422			PRG1	:PRG1 START ADDRESS
673	001162	014466			PRG2	:PRG2 START ADDRESS
674	001164	014562			PRG3	:PRG3 START ADDRESS
675	001166	014620			PRG4	:PRG4 START ADDRESS
676	001170	002456		EMTTAB:	TYP	:POINTER TO TIMEOUT ROUTINE
677	001172	002600			TYP5	:POINTER TO CHAINED MESSAGES ROUTINE
678	001174	002732			STAL	:POINTER TO RANDOM STALL ROUTINE
679	001176	001406			ERR	:POINTER TO ERROR ROUTINE
680	001200	001344			DTCHK	
681	001202	000000			OPEN	
682	001204	002302			STLSRV	
683	001206	002332			STLSPV	
684	001210	001332			EHLT	:POINTER TO ERROR HALT ROUTINE.
685	001212	002362			SRSETT	
686	001214	001756			CHAINN	
687	001216	002202			SAVRG	
688	001220	002242			RSTRG	
689	001222	001430			ERR1	
690	001224	002664			DLY	
691						
692						
693						



```
694
695 001226 000000 RCNT: OPEN
696 001230 000000 CRBUF: OPEN
697 001232 000000 CRBUFA: OPEN
698 001234 000000 CARMSK: OPEN
699 001236 000000 CHR1: OPEN
700 001240 000000 CHR2: OPEN
701 001242 000000 CHR3: OPEN
702 001244 000000 ERCTR: OPEN
703 001246 000000 CTRA: OPEN
704 001250 000000 CTRB: OPEN
705 001252 000000 CTRC: OPEN
706 001254 000000 CTRD: OPEN
707 001256 000000 TXCSRT: OPEN
708 001260 000000 RXCSRT: OPEN
709 001262 000000 RXBUFT: OPEN
710 001264 000000 TEMP: OPEN
711 001266 000000 SRT: OPEN
712 001270 177740 STLMSK: 177740
713 001272 104000 SETSR: TYPE ;TYPE SELECT OPTION MESSAGE.
714 001274 016433 ASETSR
715 001276 000000 HALT ;COMMON HALT.
716 001300 000207 RTS %7 ;EXIT.
717 001302 104000 INCRTN: TYPE ;TYPE INCORRECT ROUTINE SELECTED.
718 001304 016531 AINCR
719 001306 000000 HALT ;COMMON HALT.
720 001310 000207 RTS %7 ;EXIT.
721 001312 104000 PRGEND: TYPE ;TYPE PROGRAM END.
722 001314 016566 APGEND
723 001316 032737 002000 177570 BIT #BIT10,SR ;TEST END OF PROGRAM HALT OPTION
724 001324 001401 BEQ .+4 ;BRANCH IF NOT SELECTED
725 001326 000000 HALT
726 001330 000207 RTS %7 ;EXIT.
727
728 ;CONDITIONAL ERROR HALT ROUTINE.
729 001332 005737 177570 EHLT: TST SR ;CHECK FOR HALT ON ERROR.
730 001336 100001 BPL EHLTA ;BRANCH IF NO HALT DESIRED.
731 001340 000000 HALT ;HALT.
732 001342 000002 EHLTA: RTI ;IN DATA LIGHTS.
733
734 ;DATA CHECK ROUTINE.
735 001344 023737 001230 001232 DTCHK: CMP CRBUF,CRBUFA ;COMPARE EXPECTED AND RECEIVED
736 001352 001414 BEQ DTCHKA ;CHARS. BRANCH IF SAME.
737 001354 004537 003300 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
738 001360 001230 CRBUF ;SOURCE ADDR.
739 001362 016424 AWAS ;DESTINATION ADDR.
740 001364 000003 3 ;#OF DIGITS TO CONVERT.
741 001366 004537 003300 JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
742 001372 001232 CRBUFA ;SOURCE ADDR.
743 001374 016412 AASB ;DESTINATION ADDR.
744 001376 000003 3 ;#OF DIGITS TO CONVERT.
745 001400 104015 ERROR1
746 001402 016371 ERDAT
747 001404 000002 DTCHKA: RTI ;EXIT.
748
749
```

.MAIN. MACY11 30A(1052) 17-MAR-78 13:05 PAGE 18  
 CZDCAC.P11 17-MAR-78 13:04

750	001406	012737	177777	001542	ERR:	MOV	#-1,ERRB		;SET UP ONE MESSAGE CALL.
751	001414	012737	000240	001544		MOV	#240,ERRB+2		
752	001422	005037	001560			CLR	ERRE		
753	001426	000413				BR	ERRA		
754	001430	011637	001542		ERR1:	MOV	@%6,ERRB		;DEVELOP ADDT'L MESSAGE ADDR.
755	001434	017737	000102	001542		MOV	@ERRB,ERRB		;STORE AT ERRB.
756	001442	012737	177777	001544		MOV	#-1,ERRB+2		
757	001450	012737	000002	001560		MOV	#2,ERRE		
758	001456	032737	020000	177570	ERRA:	BIT	#BIT13,SR		;INHIBIT ERROR PRINT?
759	001464	001030				BNE	ERRC		;BRANCH TO INHIBIT PRINT.
760	001466	011637	001556			MOV	@%6,ERRD		;DEVELOP CALLING ADDR.
761	001472	162737	000002	001556		SUB	#2,ERRD		
762	001500	004537	003300			JSR	%5,OACNV		;GO TO OCTAL TO ASCII CONVERT.
763	001504	001556				ERRD			;SOURCE ADDR.
764	001506	015300				APC			;DESTINATION ADDR.
765	001510	000006				6			;#OF DIGITS TO CONVERT.
766	001512	004537	003300			JSR	%5,OACNV		;GO TO OCTAL TO ASCII CONVERT.
767	001516	001140				PRGNUM			;SOURCE ADDR.
768	001520	015263				APNUMB			;DESTINATION ADDR.
769	001522	000002				2			;#OF DIGITS TO CONVERT.
770	001524	004537	003300			JSR	%5,OACNV		;GO TO OCTAL TO ASCII CONVERT.
771	001530	001146				RTNNO			;SOURCE ADDR.
772	001532	015271				ATNUMB			;DESTINATION ADDR.
773	001534	000003				3			;#OF DIGITS TO CONVERT.
774	001536	104001				TYPES			;TYPE:
775	001540	015260				EMO			;ERROR HEADER,
776	001542	000000			ERRB:	OPEN			;ADDT'L ERROR MESSAGE IF ANY.
777	001544	177777				-1			
778	001546	104010			ERRC:	EHALT			;GO ERR HALT IF DESIRED.
779	001550	063716	001560			ADD	ERRE,@%6		
780	001554	000002				RTI			;EXIT.
781	001556	000000			ERRD:	OPEN			
782	001560	000000			ERRE:	OPEN			

```
783
784 001562 004737 002144      START: JSR    PC,DCACHE      ;DISABLE CACHE ;:++C
785 001566 012706 001076      MOV    #SPBOT,%6      ;SET BOTTOM OF SP STACK.
786 001572 005037 177776      CLR    PSW
787 001576 012737 000006 000004  MOV    #6,MACHER
788 001604 005037 001146      CLR    RTNNO
789 001610 013700 177570      MOV    SR,%0          ;(SR) TO R0
790 001614 042700 177770      BIC    #177770,%0     ;LIMIT (SR) TO BITS 3-0
791 001620 010037 001140      MOV    %0,PRGNUM      ;SAVE PROGRAM #
792 001624 006300
793 001626 012737 003644 000024  MOV    #PFAIL,24
794 001634 012737 000340 000026  MOV    #PRTY7,26
795 001642 000170 001156      JMP    @PRGTAB(0)     ;GO TO SELECTED PROGRAM.
796 001646 013737 001142 001150  GETRDY: MOV    KSTART,NXTST  ;ADDR OF 1ST ROUTINE TO NXTST
797 001654 012737 000006 000004  GTRDYX: MOV    #6,MACHER  ;RESET MACHER TRAP.
798 001662 005037 177776      CLR    PSW
799 001666 012706 001076      MOV    #SPBOT,%6     ;SET BOTTOM OF STACK.
800 001672 104011      SRESET                ;ISSUE RESET.
801 001674 004737 002042      GTRDYA: JSR    %7,FORWD  ;ROLL FORWARD TO 'NEXT' ROUTINE.
802 001700 032737 001000 177570  GTRDYB: BIT    #BIT9,SR   ;CHECK SELECT ROUTINE SWITCH
803 001706 001003      BNE    GTRDYC         ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
804 001710 000177 177230      JMP    @CURTST        ;GO RUN CURRENT ROUTINE.
805 001714 000437      BR     CHNB           ;NO GO. MANUAL RTN BYPASSED.
806 001716 013700 177570      GTRDYC: MOV    SR,%0     ;(SR) TO R0
807 001722 042700 177600      BIC    #177600,%0     ;MASK UNDESIRED BITS
808 001726 123700 001146      CMPB   RTNNO,%0      ;COMPARE RTNNO TO (R0)
809 001732 001002      BNE    GTRDYD         ;BRANCH IF ROUTINE NOT FOUND YET.
810 001734 000177 177204      JMP    @CURTST        ;GO RUN ROUTINE.
811 001740 022737 177777 001150  GTRDYD: CMP    #-1,NXTST ;NO. CHECK FOR LAST ROUTINE.
812 001746 001352      BNE    GTRDYA         ;BRANCH IF NOT LAST ROUTINE.
813 001750 004737 001302      JSR    %7,INCRN      ;YES. INCORRECT ROUTINE SELECTED.
814 001754 000734      BR     GETRDY         ;START OVER.
815 001756 032737 040000 177570  CHAINN: BIT    #BIT14,SR  ;CHECK FOR SCOPE OPTION.
816 001764 001403      BEQ    CHNA          ;BRANCH IF SCOPE SW NOT SET.
817 001766 013716 001154      CHNAB: MOV    SCOPTR,@%6 ;SET UP TO RETURN TO ROUTINE.
818 001772 000002      RTI                  ;RETURN TO ROUTINE.
819 001774 032737 004000 177570  CHNA:  BIT    #BIT11,SR   ;TEST INHIBIT ITERATION SWITCH
820 002002 001003      BNE    CHNAA         ;BRANCH IF INHIBIT ITERATION SW SET.
821 002004 005337 001152      DEC    ICTR          ;DECREMENT ITERATION COUNT.
822 002010 001366      BNE    CHNAB         ;BRANCH IF COUNT NOT 0.
823 002012 022626      CHNAA: POPSP2       ;POP STACK TWICE
824
825 002014 032737 001000 177570  CHNB:  BIT    #BIT9,SR   ;CHECK SELECT ROUTINE SWITCH
826 002022 001311      BNE    GETRDY        ;BRANCH IF SELECT RTN SW SET
827 002024 022737 177777 001150  CMP    #-1,NXTST     ;LAST TEST?
828 002032 001310      BNE    GTRDYX        ;BRANCH IF NOT LAST TEST.
829 002034 004737 001312      JSR    %7,PRGEND     ;PROGRAM END.
830 002040 000702      BR     GETRDY
831 002042 013705 001150      FORWD: MOV    NXTST,%5  ;ADDR OF NEXT ROUTINE TO R5.
832 002046 012537 001146      MOV    (5)+,RTNNO    ;GET NEXT ROUTINE NUMBER.
833 002052 012537 001150      MOV    (5)+,NXTST    ;GET ADDR OF NEXT 'NEXT' ROUTINE.
834 002056 012537 001152      MOV    (5)+,ICTR     ;GET ITERATION COUNT.
835 002062 012537 001154      MOV    (5)+,SCOPTR   ;GET SCOPE LOOP ENTRY POINTER.
836 002066 010537 001144      FORWDA: MOV    %5,CURTST ;ADDR OF NOW CURRENT TEST TO CURTST.
837 002072 000207      RTS    %7            ;EXIT FORWD SUBROUTINE.
838 002074 012737 177777 001154  FORWDB: MOV    #-1,SCOPTR ;FORCE 'NO SCOPE'
```

```
839 002102 012737 000001 001152      MOV    #1,ICTR      ;FORCE I COUNT OF 1
840 002110 000766                      BR     FORWDA
841 002112 011646      EMTINT: MOV    @%6,-(6)  ;GET SAVED PC.
842 002114 162716 000002      SUB    #2,@%6      ;DECREMENT PC BY 2.
843 002120 017616 000000      MOV    @%6,@%6
844 002124 006316      EMTA:  ASL    @%6      ;EMT ARG X 2.
845 002126 042716 177001      BIC    #177001,@%6 ;REMOVE 7 MSB.
846 002132 062716 001170      ADD    #EMTTAB,@%6 ;FORM EMT RTN ADDR.
847 002136 017616 000000      MOV    @%6,@%6
848 002142 000136      JMP    @%6+        ;GO TO EMT ROUTINE.
849
850      ;SUBROUTINE TO SIZE FOR AN 11/70 CENTRAL PROCESSOR      ;:++C
851      IF IT IS AN 11/70 CPU, CACHE WILL BE DISABLED
852      IF NOT AN 11/70 CPU, NO ACTION TAKEN
853
854      ;CALLED BY          JSR    PC,DCACHE
855      NO ARGUEMENTS PASSED
856
857
858 002144 013746 000004      DCACHE: MOV    @%4,-(SP)  ;SAVE TRAP INFO
859 002150 012737 002172 000004      MOV    #1$,@%4      ;SETUP FOR TIMEPUT
860 002156 005737 177746      TST    @%177746     ;TEST FOR CACHE
861 002162 012737 000014 177746      MOV    #14,@%177746 ;DISABLE CACHE
862 002170 000401      BR     2$          ;EXIT,CACHE DISABLED.
863 002172 022626      1$:  CMP    (SP)+,(SP)+ ;CLEAN UP STACK
864 002174 012637 000004      2$:  MOV    (SP)+,@%4
865 002200 000207      RTS    PC          ;RETURN
866
867      ;SAVE REGS 0 TO 4 SUBROUTINE.
868 002202 012637 002236      SAVRG: MOV    (6)+,SVRPC ;SAVE PC AND PSW.
869 002206 012637 002240      MOV    (6)+,SVRPSW
870 002212 010446      MOV    %4,-(6)     ;SAVE REGS 0 - 4
871 002214 010346      MOV    %3,-(6)     ;IN STACK.
872 002216 010246      MOV    %2,-(6)
873 002220 010146      MOV    %1,-(6)
874 002222 010046      MOV    %0,-(6)
875 002224 013746 002240      MOV    SVRPSW,-(6) ;RESTORE PC AND PSW.
876 002230 013746 002236      MOV    SVRPC,-(6)
877 002234 000002      RTI              ;EXIT.
878 002236 000000      SVRPC: OPEN
879 002240 000000      SVRPSW: OPEN
880      ;RESTORE REGS 0 TO 4 SUBROUTINE.
881 002242 012637 002276      RSTRG: MOV    (6)+,RSTPC ;SAVE PC AND PSW.
882 002246 012637 002300      MOV    (6)+,RSTPSW
883 002252 012600      MOV    (6)+,%0     ;RESTORE REGS 0 - 4
884 002254 012601      MOV    (6)+,%1     ;FROM STACK.
885 002256 012602      MOV    (6)+,%2
886 002260 012603      MOV    (6)+,%3
887 002262 012604      MOV    (6)+,%4
888
889 002264 013746 002300      MOV    RSTPSW,-(6) ;RESTORE PC AND PSW.
890 002270 013746 002276      MOV    RSTPC,-(6)
891 002274 000002      RTI              ;EXIT
892 002276 000000      RSTPC: OPEN
893 002300 000000      RSTPSW: OPEN
894      ;ROUTINE TO SET RECEIVER INTERRUPT VECTOR AND PRIORITY
```



```
895 002302 017637 000000 002322 STLSRV: MOV @ (6), STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2
896 002310 062716 000002          ADD #2, @%6 ;SET UP EXIT
897 002314 013701 001110          MOV RXVTR, %1
898 002320 012721 000000          STPRA: MOV #OPEN, (1)+ ;SET VECTOR ADDRESS
899 002324 013721 001112          MOV RXLVL, (1)+ ;SET PRIORITY
900 002330 000002          RTI ;EXIT
901          ;ROUTINE TO SET TRANSMITTER INTERRUPT VECTOR AND PRIORITY.
902 002332 017637 000000 002352 STLSPV: MOV @ (6), STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
903 002340 062716 000002          ADD #2, @%6 ;SET UP EXIT
904 002344 013701 001114          MOV TXVTR, %1
905 002350 012721 000000          STPPA: MOV #OPEN, (1)+ ;SET VECTOR ADDRESS.
906 002354 013721 001116          MOV TXLVL, (1)+ ;SET PRIORITY
907 002360 000002          RTI ;EXIT.
908          ;ROUTINE TO ISSUE RESET.
909 002362 012700 052525          SRSETT: MOV #52525, %0 ;DATA TO R0.
910 002366 005100          COM %0 ;COMPLEMENT (R0).
911 002370 010037 002364          MOV %0, SRSETT+2 ;(R0) TO SRSETT+2.
912 002374 000005          RESET ;ISSUE RESET. (R0) IS
913 002376 004737 002144          JSR PC, DCACHE ;DISABLE CACHE. ;++;+C
914 002402 000002          RTI ;DISPLAYED. EXIT.
915
916          ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
917 002404 013700 002452          RNGEN: MOV RP1, %0
918 002410 006100          ROL %0
919 002412 006100          ROL %0
920 002414 063700 002454          ADD RP2, %0
921 002420 010037 002452          MOV %0, RP1
922 002424 006100          ROL %0
923 002426 006100          ROL %0
924 002430 063700 002454          ADD RP2, %0
925 002434 006100          ROL %0
926 002436 006100          ROL %0
927 002440 010037 002454          MOV %0, RP2
928 002444 013700 002452          MOV RP1, %0
929 002450 000207          RTS %7 ;EXIT. NUMBER IN R0
930 002452 001233          RP1: 1233
931 002454 007622          RP2: 7622
932          ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
933 002456 011600          TYP: MOV @%6, %0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
934 002460 062716 000002          ADD #2, @%6 ;SET UP EXIT.
935 002464 011000          MOV @%0, %0 ;ADDRESS OF MESSAGE TO R0.
936 002466 112037 002576          TYPA: MOVB (0)+, TYPDAT ;GET CHARACTER
937 002472 122737 000100 002576          CMPB #100, TYPDAT ;CHECK FOR 'a' CHARACTER
938 002500 001001          BNE TYPC ;BRANCH IF NOT 'a'.
939 002502 000002          RTI ;TERMINATOR CHAR. DONE. EXIT.
940 002504 122737 000045 002576          TYPC: CMPB #45, TYPDAT ;CHECK FOR '%'.
941 002512 001416          BEQ TYPF ;BRANCH IF '%'.
942 002514 122737 000043 002576          CMPB #43, TYPDAT ;NOT '%'. CHECK FOR '#'.
943 002522 001417          BEQ TYPG ;BRANCH IF '#'.
944 002524 004737 002532          JSR %7, TYPD ;TYPE CHAR IN TYPDAT
945 002530 000756          BR TYPA
946 002532 113777 002576 176366          TYPD: MOVB TYPDAT, @TPB ;OUTPUT CHARACTER TO PRINTER
947 002540 105777 176360          TSTB @TPS ;WAIT FOR DONE FLAG.
948 002544 100375          BPL -4
949 002546 000207          RTS %7 ;EXIT
950 002550 112737 000015 002576          TYPF: MOVB #15, TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
```

```

951 002556 004737 002532          JSR    %7,TYPD      ;GO TYPE CHAR.
952 002562 112737 000012 002576  TYPG:  MOVB   #12,TYFDAT  ;MOVE LF CODE TO TYPDAT.
953 002570 004737 002532          JSR    %7,TYPD      ;GO TYPE CHAR.
954 002574 000734                BR     TYPA
955 002576 000000          TYPDAT: OPEN
956                ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
957 002600 011600          TYP5:  MOV    @%6,%0      ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
958 002602 062716 000002          ADD    #2,@%6        ;UPDATE TO NEXT MESSAGE ADDRESS
959 002606 011037 002626          MOV    @%0,TYPSB     ;ADDRESS OF MESSAGE TO TYPSB
960 002612 022737 177777 002626          CMP    #-1,TYPSB    ;CHECK FOR TERMINATOR
961 002620 001001          BNE    TYP5A        ;BRANCH IF NOT TERMINATOR.
962 002622 000002          RTI
963 002624 104000          TYP5A: TYPE        ;CALL ON TYP SUB TO TYPE MESSAGE
964 002626 000000          TYPSB: OPEN      ;ADDRESS OF MESSAGE GOES HERE
965 002630 000763          BR     TYP5        ;GO PROCESS NEXT MESSAGE
966
967 002632 012701 000300          OVRLAY: MOV   #300,%1   ;GET DC11 VECTOR BASE ADDRESS
968 002636 012702 000302          MOV   #302,%2       ;GET NEXT ADDRESS
969 002642 010221          OVRLYA: MOV   %2,(1)+ ;LOAD VECTOR WITH ADDRESS OF NEXT ADDRESS
970 002644 005021          CLR   (1)+         ;PUT A HALT IN THE NEXT ADDRESS
971 002646 020237 001000          CMP   %2,1000      ;ALL VECTORS BEEN LOADED
972 002652 001403          BEQ   OVRLYB
973 002654 062702 000004          ADD   #4,%2        ;GET NEXT VECTOR ADDRESS
974 002660 000770          BR   OVRLYA
975 002662 000207          OVRLYB: RTS    7    ;EXIT
976
977                ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
978 002664 011637 002730          DLY:  MOV   @%6,DLCNT ;GET DELAY COUNT ADDRESS.
979 002670 062716 000002          ADD   #2,@%6        ;SET UP EXIT ADDRESS
980 002674 017746 000030          MOV   @DLCNT,-(6)  ;DELAY COUNT TO STACK
981 002700 001411          BEQ   DLYC
982 002702 005037 177776          CLR   PSW         ;SET PRIORITY 0
983 002706 012746 000554          DLYA: MOV   #554,-(6);1 MSEC COUNT TO STACK
984 002712 005316          DLYB: DEC   @%6     ;DECREMENT 1 MSEC COUNT
985 002714 001376          BNE   DLYB        ;BRANCH IF NOT 0.
986 002716 005726          POPSP           ;ZERO UNCOVER MSECS. COUNT.
987 002720 005316          DEC   @%6        ;DECREMENT IT
988 002722 001371          BNE   DLYA        ;BR IF NOT DONE DELAYING
989 002724 005726          DLYC: POPSP      ;DONE
990 002726 000002          RTI             ;EXIT.
991 002730 000000          DLCNT: OPEN     ;CONTAINS MILLISECONDS COUNT ADDRESS.
992                ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
993                ;DETERMINED BY CONTENTS OF LOC STLMSK.
994 002732 004737 002404          STAL: JSR    %7,RNGEN ;GO GET RANDOM NUMBER.
995 002736 043700 001270          BIC   STLMSK,%0    ;# IN R0. APPLY STALL MASK.
996 002742 001404          BEQ   STALB       ;BRANCH IF RESULT IS 0.
997 002744 010037 002752          MOV   %0,STALA
998 002750 104016          DELAY
999 002752 000000          STALA: OPEN     ;DELAY COUNT
1000 002754 000002          STALB: RTI      ;DONE. EXIT.
1001                ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
1002 002756 004737 002404          GRCNT: JSR   %7,RNGEN ;GET RANDOM NUMBER
1003 002762 043700 002776          BIC   RCMSK,%0    ;APPLY MASK
1004 002766 001773          BEQ   GRCNT       ;TRY AGAIN IF RESULT 0
1005 002770 010037 003000          MOV   %0,RNCNT   ;COUNT TO RNCNT
1006 002774 000207          RTS    %7        ;EXIT.

```

```
1007 002776 000000          RCMSK: OPEN          ;RANDOM CHARACTER MASK.
1008 003000 000000          RNCNT: OPEN          ;RANDOM CHARACTER COUNT.
1009                          ;SUBROUTINE TO SELECT LINE AND
1010 003002 104000          LINSEL: TYPE
1011 003004 017010          LDLINE
1012 003006 000000          HALT
1013 003010 013701 177570          MOV SR,%1
1014 003014 042701 177407          BIC #177407,%1
1015 003020 010137 001264          MOV %1,TEMP
1016 003024 012702 000770          MOV #770,%2
1017 003030 012703 001100          MOV #RXCSR,%3
1018 003034 012704 000004          MOV #4,%4
1019 003040 040213          BIC %2,(3)
1020 003042 050123          BIS %1,(3)+
1021 003044 005304          DEC %4
1022 003046 001374          BNE .-6
1023 003050 006201          ASR %1          ;POSITION SELECTED LINE
1024 003052 006201          ASR %1          ;
1025 003054 016101 015160          MOV VECTAB(1),%1 ;GET LINE VECTOR ADDRESS
1026 003060 010137 001110          MOV %1,RXVTR    ;LOAD INTO PROGRAM RXVTR
1027 003064 022121          CMP (1)+,(1)+  ;ADD +4
1028 003066 010137 001114          MOV %1,TVVTR    ;LOAD INTO PROGRAM TVVTR
1029 003072 006237 001264          ASR TEMP
1030 003076 006237 001264          ASR TEMP
1031 003102 006237 001264          ASR TEMP
1032 003106 004537 003300          JSR 5,OACNV     ;TYPE LINE #
1033 003112 001264          TEMP
1034 003114 017061          SELINE
1035 003116 000002          2
1036 003120 104000          TYPE
1037 003122 017047          ALINE
1038 003124 000205          RTS 5
1039                          ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
1040 003126 012737 177777 003150 INBIN: MOV #-1,RIND    ;SET ALL VARIABLES
1041 003134 004537 003362          JSR %5,BMOVE    ;TO MINUS 1.
1042 003140 003150          RIND
1043 003142 003151          RIND+1
1044 003144 000013          11.
1045 003146 000207          RTS %7          ;EXIT
1046 003150 000000          RIND: OPEN
1047 003152 000000          PT0: OPEN
1048 003154 000000          PT1: OPEN
1049 003156 000000          PIND: OPEN
1050 003160 000000          PTOP: OPEN
1051 003162 000000          PT1P: OPEN
1052                          ;SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN RO
1053 003164 013737 003152 003154 GTBIN: MOV PT0,PT1    ;PREVIOUS BIN CHAR TO PT1
1054 003172 005137 003154          COM PT1
1055 003176 005137 003150          COM RIND
1056 003202 001002          BNE .+6
1057 003204 005237 003154          INC PT1
1058 003210 042737 177400 003154          BIC #177400,PT1 ;MASK TO 8 BITS
1059 003216 013737 003154 003152          MOV PT1,PT0    ;SAVE BIN CHAR IN PT0
1060 003224 013700 003154          MOV PT1,%0     ;BIN CHAR TO RO.
1061 003230 000207          RTS %7          ;EXIT.
1062 003232 013737 003160 003162 GTBINP: MOV PTOP,PT1P   ;PREVIOUS BIN CHAR TO PT1P
```

```
1063 003240 005137 003162      COM      PT1P
1064 003244 005137 003156      COM      PIND
1065 003250 001002              BNE      .+6
1066 003252 005237 003162      INC      PT1P
1067 003256 042737 177400 003162  BIC      #177400,PT1P      ;MASK TO 8 BITS.
1068 003264 013737 003162 003160  MOV      PT1P,PTOP      ;SAVE BIN CHAR IN PTOP.
1069 003272 013701 003162      MOV      PT1P,%1        ;BIN CHAR TO R1.
1070 003276 000207              RTS      %7              ;EXIT.
1071              ;OCTAL TO ASCII CONVERT ROUTINE
1072 003300 013537 003360      OACNV:  MOV      @(%5)+,OACNVX ;GET OCTAL VALUE.
1073 003304 012501              MOV      (%5)+,%1        ;GET DESTINATION ADDR.
1074 003306 012502              MOV      (%5)+,%2        ;GET CONVERT COUNT.
1075 003310 060201              ADD      %2,%1          ;DEVELOP ADDR TO STORE 1ST CHAR.
1076 003312 013703 003360      OACNVA: MOV      OACNVX,%3
1077 003316 042703 177770      BIC      #177770,%3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
1078 003322 062703 000060      ADD      #60,%3         ;CONVERT DIGIT TO ASCII.
1079 003326 110341              MOVB     %3,-(1)        ;STORE ASCII CHARACTER.
1080 003330 042737 000007 003360  BIC      #7,OACNVX
1081 003336 006037 003360      ROR      OACNVX
1082 003342 006037 003360      ROR      OACNVX
1083 003346 006037 003360      ROR      OACNVX
1084 003352 005302              DEC      %2              ;DONE ALL DIGITS?
1085 003354 001356              BNE      OACNVA         ;BRANCH IF NOT DONE.
1086 003356 000205              RTS      %5              ;DONE. EXIT.
1087 003360 000000      OACNVX: OPEN
1088              ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1089 003362 104013      BMOVE:  SAVREG          ;SAVE REGS.
1090 003364 012501              MOV      (%5)+,%1        ;GET 'FROM' ADDRESS
1091 003366 012502              MOV      (%5)+,%2        ;GET 'TO' ADDRESS
1092 003370 012503              MOV      (%5)+,%3        ;GET COUNT
1093 003372 112122      BMOVA:  MOVB     (1)+,(2)+ ;MOVE BYTE
1094 003374 005303              DEC      %3              ;DECREMENT COUNT
1095 003376 001375              BNE      BMOVA         ;BRANCH IF NOT DONE.
1096 003400 104014              RSTREG          ;RESTORE REGS.
1097 003402 000205              RTS      %5              ;DONE EXIT
1098              ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
1099 003404 012700 003524      BDCNV:  MOV      #DECVAL,%0 ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
1100 003410 013501              MOV      @(%5)+,%1        ;BINARY VALUE TO R1.
1101 003412 012702 003512      MOV      #ADTENP,%2      ;ADDR OF TEN POWER STRING TO R2.
1102 003416 012737 000005 003504  MOV      #5,CNVCTR      ;SET UP FOR 5 POWER CONVERSIONS.
1103 003424 012237 003510      BDCNVA: MOV      (2)+,TENPWR ;MOVE POWER OF TEN VALUE TO TENPWR.
1104 003430 004737 003444      JSR      %7,SUBTEN      ;PERFORM CONVERSION
1105 003434 005337 003504      DEC      CNVCTR          ;DONE 5 CONVERSIONS?
1106 003440 001371              BNE      BDCNVA         ;BRANCH IF NOT YET 5.
1107 003442 000205              RTS      %5              ;YES, EXIT.
1108 003444 005037 003506      SUBTEN: CLR      DIGIT    ;CLEAR DIGIT
1109 003450 163701 003510      SUBTNA: SUB      TENPWR,%1 ;SUBTRACT TEN POWER FROM BINARY VALUE.
1110 003454 103403              BCS      SUBTNB         ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
1111 003456 005237 003506      INC      DIGIT
1112 003462 000772              BR      SUBTNA
1113 003464 063701 003510      SUBTNB: ADD      TENPWR,%1 ;RESTORE SUBTRACTED VALUE.
1114 003470 062737 000060 003506  ADD      #60,DIGIT      ;CONVERT (DIGIT) TO ASCII
1115 003476 113720 003506      MOVB     DIGIT,(0)+     ;MOVE ASCII CHAR TO DECVAL FIELD.
1116 003502 000207              RTS      %7              ;EXIT.
1117 003504 000000      CNVCTR: OPEN
1118 003506 000000      DIGIT:  OPEN
```

```
1119 003510 000000          TENPWR: OPEN
1120 003512 023420          ADTENP: 10000.
1121 003514 001750          1000.
1122 003516 000144          100.
1123 003520 000012          10.
1124 003522 000001          1
1125 003524 040 040 040 DECVAL: .BYTE 040,040,040,040,040,040
1126 003527 040 040 040
1127 003532 012537 001266  DATTST: MOV (5)+,SRT ;GET PARAMETERS
1128 003536 004737 014772  JSR 7,STPARB ;LOAD PARAMETERS
1129 003542 042777 000001 175330 BIC #BIT0,@RXCSR ;CLEAR DATA TERM. READY
1130 003550 052777 000004 175326 BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
1131 003556 012737 000144 001246 MOV #100.,CTRA ;GET CHARACTER COUNT
1132 003564 105777 175314 DATAA: TSTB @TXCSR ;WAIT FOR
1133 003570 100375 BPL .-4 ;READY FLAG
1134 003572 004737 003232 JSR 7,GTBINP ;GET CHARACTER
1135 003576 110137 001232 MOVB %1,CRBUFA ;MOVE CHARACTER
1136 003602 043737 001234 001232 BIC CARMSK,CRBUFA ;MASK OFF NON TRANSMITTED BITS
1137 003610 110177 175272 MOVB %1,@TXBUF ;TRANSMIT CHARACTER
1138 003614 105777 175260 TSTB @RXCSR ;WAIT FOR
1139 003620 100375 BPL .-4 ;DONE FLAG
1140 003622 117737 175254 001230 MOVB @RXBUF,CRBUF ;GET RECEIVED CHARACTER
1141 003630 104004 DATCHK ;CHK DATA
1142 003632 005337 001246 DEC CTRA ;DECREMENT CHARACTER COUNT
1143 003636 001352 BNE DATAA
1144 003640 005726 TST (6)+ ;POP STACK
1145 003642 104012 SCOPE
1146
1147 003644 012737 003654 000024 PFAIL: MOV #PWRUP,24
1148 003652 000000 HALT
1149
1150 003654 000005 PWRUP: RESET
1151 003656 004737 002144 JSR PC,DCACHE ;DISABLE CACHE. ;:++C
1152 003662 012706 001076 MOV #SPBOT,%6
1153 003666 104003 ERROR
1154 003670 013700 001140 RESTART:MOV PRGNUM,%0 ;GET PROGRAM NUMBER
1155 003674 006300 ASL %0
1156 003676 012737 003644 000024 MOV #PFAIL,24 ;RELOAD POWER FAIL VECTOR
1157 003704 004737 014772 JSR 7,STPARB ;RELOAD LINE PARAMETERS
1158 003710 000170 003714 JMP @RSTART(0) ;GO RESTART SELECTED PROGRAM
1159
1160 003714 003752 RSTART: PRG0A ;PROGRAM 0 RESTART ADDRESS
1161 003716 014444 PRG1A ;PROGRAM 1 RESTART ADDRESS
1162 003720 014510 PRG2A ;PROGRAM 2 RESTART ADDRESS
1163 003722 014604 PRG3A ;PROGRAM 3 RESTART ADDRESS
1164 003724 014640 PRG4A ;PROGRAM 4 RESTART ADDRESS
1165
1166 ;PRG0 - INPUT-OUTPUT LOGIC TESTS
1167 003726 012737 003756 001142 PRG0: MOV #AT0,KSTART
1168 003734 104000 TYPE ;TYPE TITLE AND INSTRUCTIONS
1169 003736 015311 POTIT
1170 003740 000000 HALT
1171 003742 004537 003002 JSR 5,LINSEL ;GO GET LINE # FROM USER
1172 003746 004737 001272 JSR 7,SETSR
1173 003752 000137 001646 PRG0A: JMP GETRDY ;GET STARTED.
1174 177777 TX=-1
```

```
1175 ;*****
1176 003756 000000 ATO: 0 ;TEST NUMBER 0 *
1177 003760 004010 ; AT1 ;ADDRESS OF NEXT TEST *
1178 003762 001750 ; 1000. ;TEST ITERATION COUNT *
1179 003764 003766 ; AAA ;SCOPE ENTRY POINT *
1180 ;*****
1181 ;TEST ABILITY TO REFERENCE RECEIVER CSR WITHOUT TRAPPING
1182 003766 012737 004002 000004 AAA: MOV #AAE,MACHER ;SET UP MACHINE ERROR TRAP.
1183 003774 005777 175100 ; TST @RXCSR ;REFERENCE RXCSR
1184 004000 104012 AAB: SCOPE ;OK IF NO TRAP. SCOPE
1185 004002 022626 AAE: POPSP2
1186 004004 104003 ; ERROR ;TRAPPED WHEN REFERENCING RXCSR.
1187 004006 000774 ; BR AAB
1188 ;*****
1189 004010 000001 AT1: 1 ;TEST NUMBER 1 *
1190 004012 004042 ; AT2 ;ADDRESS OF NEXT TEST *
1191 004014 001750 ; 1000. ;TEST ITERATION COUNT *
1192 004016 004020 ; ABA ;SCOPE ENTRY POINT *
1193 ;*****
1194 ;TEST ABILITY TO REFERENCE RECEIVER BUFFER WITHOUT TRAPPING
1195 004020 012737 004034 000004 ABA: MOV #ABE,MACHER ;SET UP MACHINE ERROR TRAP.
1196 004026 005777 175050 ; TST @RXBUF ;REFERENCE RXBUF
1197 004032 104012 ABB: SCOPE ;OK IF NO TRAP SCOPE
1198 004034 022626 ABE: POPSP2
1199 004036 104003 ; ERROR ;TRAPPED WHEN REFERENCING RXBUF
1200 004040 000774 ; BR ABB
1201
1202 ;*****
1203 004042 000002 AT2: 2 ;TEST NUMBER 2 *
1204 004044 004074 ; AT3 ;ADDRESS OF NEXT TEST *
1205 004046 001750 ; 1000. ;TEST ITERATION COUNT *
1206 004050 004052 ; ACA ;SCOPE ENTRY POINT *
1207 000000 ; TX-TX+1
1208 ;*****
1209 ;TEST ABILITY TO REFERENCE TRANSMITTER CSR WITHOUT TRAPPING.
1210 004052 012737 004066 000004 ACA: MOV #ACE,MACHER ;SET UP MACHINE ERROR TRAP.
1211 004060 005777 175020 ; TST @TXCSR ;REFERENCE TXCSR
1212 004064 104012 ACB: SCOPE ;SCOPE
1213 004066 022626 ACE: POPSP2
1214 004070 104003 ; ERROR ;TRAPPED WHEN REFERENCING TXCSR
1215 004072 000774 ; BR ACB
1216
1217 ;*****
1218 004074 000003 AT3: 3 ;TEST NUMBER 3 *
1219 004076 004126 ; AT4 ;ADDRESS OF NEXT TEST *
1220 004100 001750 ; 1000. ;TEST ITERATION COUNT *
1221 004102 004104 ; ADA ;SCOPE ENTRY POINT *
1222 ;*****
1223 ;TEST ABILITY TO REFERENCE TRANSMITTER BUFFER WITHOUT TRAPPING
1224 004104 012737 004120 000004 ADA: MOV #ADE,MACHER ;SET UP MACHINE ERROR TRAP.
1225 004112 005777 174770 ; TST @TXBUF ;REFERENCE TX BUF.
1226 004116 104012 ADB: SCOPE ;SCOPE
1227 004120 022626 ADE: POPSP2
1228 004122 104003 ; ERROR ;TRAPPED WHEN REFERENCING TXBUF
1229 004124 000774 ; BR ADB
1230
```

```
1231 :*****
1232 004126 000004 AT4: 4 ;TEST NUMBER 4 *
1233 004130 004226 AT5 ;ADDRESS OF NEXT TEST *
1234 004132 000144 100. ;TEST ITERATION COUNT *
1235 004134 004136 AEA ;SCOPE ENTRY POINT *
1236 :*****
1237 :TEST THAT TXCSR BIT0 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1238 004136 032777 000001 174740 AEA: BIT #BIT0,@TXCSR ;SEE IF TXCSR BIT0 IS CLEAR.
1239 004144 001402 BEQ AEB ;BRANCH IF BIT IS CLEAR.
1240 004146 104003 ERROR ;RESET DID NOT CLEAR TXCSR BIT0
1241 004150 000421 BR AED
1242 004152 052777 000001 174724 AEB: BIS #BIT0,@TXCSR ;SET TXCSR BIT0.
1243 004160 032777 000001 174716 BIT #BIT0,@TXCSR ;SEE IF BIT IS SET.
1244 004166 001702 BNE AEC ;BRANCH IF BIT IS SET.
1245 004170 104003 ERROR ;TXCSR BIT0 FAILED TO SET.
1246 004172 000410 BR AED
1247 004174 042777 000001 174702 AEC: BIC #BIT0,@TXCSR ;CLEAR TXCSR BIT0
1248 004202 032777 000001 174674 BIT #BIT0,@TXCSR ;SEE IF BIT IS CLEAR.
1249 004210 001401 BEQ AED
1250 004212 104003 ERROR ;TXCSR BIT0 FAILED TO CLEAR.
1251 004214 052777 000001 174662 AED: BIS #BIT0,@TXCSR ;SET TXCSR BIT0.
1252 004222 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1253 004224 104012 SCOPE ;SCOPE
1254 :*****
1255 004226 000005 AT5: 5 ;TEST NUMBER 5 *
1256 004230 004324 AT6 ;ADDRESS OF NEXT TEST *
1257 004232 000144 100. ;TEST ITERATION COUNT *
1258 004234 004236 AFA ;SCOPE ENTRY POINT *
1259 :*****
1260 :TEST THAT TXCSR BIT1 (CLEAR TO SEND) CAN BE SET, AND CLEARED
1261 004236 042777 000001 174634 AFA: BIC #BIT0,@TXCSR ;CLEAR DATA TERMINAL READY
1262 004244 032777 000002 174632 BIT #BIT1,@TXCSR ;SEE IF TXCSR BIT1 IS CLEAR.
1263 004252 001402 BEQ AFB ;BRANCH IF BIT IS CLEAR.
1264 004254 104003 ERROR ;TXCSR BIT1 IS NOT CLEAR.
1265 004256 000421 BR AFD ;EXIT TEST
1266 004260 052777 000001 174612 AFB: BIS #BIT0,@TXCSR ;SET DATA TERM. RDY. (SETS CTS VIA JUMPER)
1267 004266 032777 000002 174610 BIT #BIT1,@TXCSR ;IS CLEAR TO SEND SET?
1268 004274 001002 BNE AFC ;BRANCH IF SET
1269 004276 104003 ERROR ;CTS NOT BEING SET VIA DTR
1270 004300 000410 BR AFD ;EXIT TEST
1271 004302 042777 000001 174570 AFC: BIC #BIT0,@TXCSR ;CLEAR DATA TERM. RDY.
1272 004310 032777 000002 174566 BIT #BIT1,@TXCSR ;IS CTS CLEAR?
1273 004316 001401 BEQ AFD
1274 004320 104003 ERROR ;CTS FAILED TO CLEAR VIA DTR
1275 004322 104012 AFD: SCOPE ;SCOPE
1276 :*****
1277 :*****
1278 004324 000006 AT6: 6 ;TEST NUMBER 6 *
1279 004326 004424 AT7 ;ADDRESS OF NEXT TEST *
1280 004330 000144 100. ;TEST ITERATION COUNT *
1281 004332 004334 AGA ;SCOPE ENTRY POINT *
1282 :*****
1283 :TEST THAT TXCSR BIT2 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1284 004334 032777 000004 174542 AGA: BIT #BIT2,@TXCSR ;SEE IF TXCSR BIT2 IS CLEAR.
1285 004342 001402 BEQ AGB ;BRANCH IF BIT IS CLEAR.
1286 004344 104003 ERROR ;RESET DID NOT CLEAR TXCSR BIT2
```



```
1287 004346 000421          BR      AGD
1288 004350 052777 000004 174526 AGB:  BIS    #BIT2,@TXCSR ;SET TXCSR BIT2.
1289 004356 032777 000004 174520      BIT    #BIT2,@TXCSR ;SEE IF BIT IS SET.
1290 004364 001002          BNE    AGC          ;BRANCH IF BIT IS SET.
1291 004366 104003          ERROR
1292 004370 000410          BR      AGD          ;TXCSR BIT2 FAILED TO SET.
1293 004372 042777 000004 174504 AGC:  BIC    #BIT2,@TXCSR ;CLEAR TXCSR BIT2
1294 004400 032777 000004 174476      BIT    #BIT2,@TXCSR ;SEE IF BIT IS CLEAR.
1295 004406 001401          BEQ    AGD
1296 004410 104003          ERROR
1297 004412 052777 000004 174464 AGD:  BIS    #BIT2,@TXCSR ;TXCSR BIT2 FAILED TO CLEAR.
1298 004420 104011          SRESET ;SET TXCSR BIT2.
1299 004422 104012          SCOPE ;ISSUE RESET TO CLEAR BIT.
1300                                     ;SCOPE
1301 004424 000007          AT7:   7          ;*****
1302 004426 004524          AT10  ;TEST NUMBER 7 *
1303 004430 000144          100.  ;ADDRESS OF NEXT TEST *
1304 004432 004434          AHA    ;TEST ITERATION COUNT *
1305                                     ;SCOPE ENTRY POINT *
1306                                     ;*****
1307 004434 032777 000010 174442 AHA:  BIT    #BIT3,@TXCSR ;TEST THAT TXCSR BIT3 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1308 004442 001402          BEQ    AHB          ;SEE IF TXCSR BIT3 IS CLEAR.
1309 004444 104003          ERROR ;BRANCH IF BIT IS CLEAR.
1310 004446 000421          BR      AHD          ;RESET DID NOT CLEAR TXCSR BIT3
1311 004450 052777 000010 174426 AHB:  BIS    #BIT3,@TXCSR ;SET TXCSR BIT3.
1312 004456 032777 000010 174420      BIT    #BIT3,@TXCSR ;SEE IF BIT IS SET.
1313 004464 001002          BNE    AHC          ;BRANCH IF BIT IS SET.
1314 004466 104003          ERROR ;TXCSR BIT3 FAILED TO SET.
1315 004470 000410          BR      AHD
1316 004472 042777 000010 174404 AHC:  BIC    #BIT3,@TXCSR ;CLEAR TXCSR BIT3
1317 004500 032777 000010 174376      BIT    #BIT3,@TXCSR ;SEE IF BIT IS CLEAR.
1318 004506 001401          BEQ    AHD
1319 004510 104003          ERROR ;TXCSR BIT3 FAILED TO CLEAR.
1320 004512 052777 000010 174364 AHD:  BIS    #BIT3,@TXCSR ;SET TXCSR BIT3.
1321 004520 104011          SRESET ;ISSUE RESET TO CLEAR BIT.
1322 004522 104012          SCOPE ;SCOPE
```

```
1323
1324 :*****
1325 004524 000010 AT10: 10 ;TEST NUMBER 10 *
1326 004526 004624 AT11 ;ADDRESS OF NEXT TEST *
1327 004530 000144 100. ;TEST ITERATION COUNT *
1328 004532 004534 AIA ;SCOPE ENTRY POINT *
1329 :*****
1330 :TEST THAT TXCSR BIT4 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1331 004534 032777 000020 174342 AIA: BIT #BIT4,@TXCSR ;SEE IF TXCSR BIT4 IS CLEAR.
1332 004542 001402 BEQ AIB ;BRANCH IF BIT IS CLEAR.
1333 004544 104003 ERROR ;RESET DID NOT CLEAR TXCSR BIT4
1334 004546 000421 BR AID
1335 004550 052777 000020 174326 AIB: BIS #BIT4,@TXCSR ;SET TXCSR BIT4.
1336 004556 032777 000020 174320 BIT #BIT4,@TXCSR ;SEE IF BIT IS SET.
1337 004564 001002 BNE AIC ;BRANCH IF BIT IS SET.
1338 004566 104003 ERROR ;TXCSR BIT4 FAILED TO SET.
1339 004570 000410 BR AID
1340 004572 042777 000020 174304 AIC: BIC #BIT4,@TXCSR ;CLEAR TXCSR BIT4
1341 004600 032777 000020 174276 BIT #BIT4,@TXCSR ;SEE IF BIT IS CLEAR.
1342 004606 001401 BEQ AID
1343 004610 104003 ERROR ;TXCSR BIT4 FAILED TO CLEAR.
1344 004612 052777 000020 174264 AID: BIS #BIT4,@TXCSR ;SET TXCSR BIT4.
1345 004620 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1346 004622 104012 SCOPE ;SCOPE
1347 :*****
1348 004624 000011 AT11: 11 ;TEST NUMBER 11 *
1349 004626 004732 AT12 ;ADDRESS OF NEXT TEST *
1350 004630 000144 100. ;TEST ITERATION COUNT *
1351 004632 004634 AJA ;SCOPE ENTRY POINT *
1352 :*****
1353 :TEST THAT TXCSR BIT6 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1354 004634 012737 000340 177776 AJA: MOV #PRTY7,PSW ;SET PRIORITY 7.
1355 004642 032777 000100 174234 BIT #BIT6,@TXCSR ;SEE IF TXCSR BIT6 IS CLEAR.
1356 004650 001402 BEQ AJB ;BRANCH IF BIT IS CLEAR.
1357 004652 104003 ERROR ;RESET DID NOT CLEAR TXCSR BIT6
1358 004654 000421 BR AJD
1359 004656 052777 000100 174220 AJB: BIS #BIT6,@TXCSR ;SET TXCSR BIT6.
1360 004664 032777 000100 174212 BIT #BIT6,@TXCSR ;SEE IF BIT IS SET.
1361 004672 001002 BNE AJC ;BRANCH IF BIT IS SET.
1362 004674 104003 ERROR ;TXCSR BIT6 FAILED TO SET.
1363 004676 000410 BR AJD
1364 004700 042777 000100 174176 AJC: BIC #BIT6,@TXCSR ;CLEAR TXCSR BIT6
1365 004706 032777 000100 174170 BIT #BIT6,@TXCSR ;SEE IF BIT IS CLEAR.
1366 004714 001401 BEQ AJD
1367 004716 104003 ERROR ;TXCSR BIT6 FAILED TO CLEAR.
1368 004720 052777 000100 174156 AJD: BIS #BIT6,@TXCSR ;SET TXCSR BIT6.
1369 004726 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1370 004730 104012 SCOPE ;SCOPE
1371
```

```

1372
1373
1374 004732 000012
1375 004734 004754
1376 004736 000144
1377 004740 004742
1378
1379
1380
1381 004742 105777 174136
1382 004746 100401
1383 004750 104003
1384 004752 104012
1385
1386 004754 000013
1387 004756 005054
1388 004760 000144
1389 004762 004764
1390
1391
1392 004764 032777 000400 174112
1393 004772 001402
1394 004774 104003
1395 004776 000421
1396 005000 052777 000400 174076
1397 005006 032777 000400 174070
1398 005014 001002
1399 005016 104003
1400 005020 000410
1401 005022 042777 000400 174054
1402 005030 032777 000400 174046
1403 005036 001401
1404 005040 104003
1405 005042 052777 000400 174034
1406 005050 104011
1407 005052 104012
1408
1409
1410
1411 005054 000014
1412 005056 005100
1413 005060 000144
1414 005062 005064
1415
1416
1417 005064 032777 100000 174012
1418 005072 001401
1419 005074 104003
1420 005076 104012
1421

```

```

:*****
AT12: 12 ;TEST NUMBER 12 *
      AT13 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AKA ;SCOPE ENTRY POINT *
:*****
;TEST THAT TXCSR BIT 7 (READY BIT) IS SET UPON ENTERING ROUTINE AND
;THAT IT CAN BE READ RELIABLY.
AKA: TSTB @TXCSR ;SEE IF TXCSR BIT 7 IS SET.
      BMI AKB ;BRANCH IF SET.
      ERROR ;TXCSR BIT 7 NOT SET.
AKB: SCOPE ;SCOPE
:*****
AT13: 13 ;TEST NUMBER 13 *
      AT14 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ALA ;SCOPE ENTRY POINT *
:*****
;TEST THAT TXCSR BIT8 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
ALA: BIT #BIT8,@TXCSR ;SEE IF TXCSR BIT8 IS CLEAR.
      BEQ ALB ;BRANCH IF BIT IS CLEAR.
      ERROR ;RESET DID NOT CLEAR TXCSR BIT8
      BR ALD
ALB: BIS #BIT8,@TXCSR ;SET TXCSR BIT8.
      BIT #BIT8,@TXCSR ;SEE IF BIT IS SET.
      BNE ALC ;BRANCH IF BIT IS SET.
      ERROR ;TXCSR BIT8 FAILED TO SET.
      BR ALD
ALC: BIC #BIT8,@TXCSR ;CLEAR TXCSR BIT8
      BIT #BIT8,@TXCSR ;SEE IF BIT IS CLEAR.
      BEQ ALD
      ERROR ;TXCSR BIT8 FAILED TO CLEAR.
ALD: BIS #BIT8,@TXCSR ;SET TXCSR BIT8.
      SRESET ;ISSUE RESET TO CLEAR BIT.
      SCOPE ;SCOPE
:*****
AT14: 14 ;TEST NUMBER 14 *
      AT15 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AMA ;SCOPE ENTRY POINT *
:*****
;TEST THAT TXCSR BIT15 IS CLEAR AND CAN BE READ RELIABLY.
AMA: BIT #BIT15,@TXCSR ;SEE IF TXCSR BIT15 IS CLEAR.
      BEQ AMB ;BRANCH IF BIT IS CLEAR.
      ERROR ;TXCSR BIT15 IS NOT CLEAR.
AMB: SCOPE ;SCOPE

```

```

1422
1423 005100 000015
1424 005102 005172
1425 005104 000144
1426 005106 005110
1427
1428
1429 005110 052777 000001 173762
1430 005116 032777 000001 173754
1431 005124 001002
1432 005126 104003
1433 005130 000417
1434 005132 104011
1435 005134 032777 000001 173736
1436 005142 001002
1437 005144 104003
1438 005146 000410
1439 005150 042777 000001 173722
1440 005156 032777 000001 173714
1441 005164 001401
1442 005166 104003
1443 005170 104012
1444
1445 005172 000016
1446 005174 005272
1447 005176 000144
1448 005200 005202
1449
1450
1451
1452 005202 032777 000002 173670
1453 005210 001402
1454 005212 104003
1455 005214 000421
1456 005216 052777 000002 173654
1457 005224 032777 000002 173646
1458 005232 001002
1459 005234 104003
1460 005236 000410
1461 005240 042777 000002 173632
1462 005246 032777 000002 173624
1463 005254 001401
1464 005256 104003
1465 005260 052777 000002 173612
1466 005266 104011
1467 005270 104012
1468
1469
1470 005272 000017
1471 005274 005316
1472 005276 000144
1473 005300 005302
1474
1475
1476 005302 032777 000004 173570
1477 005310 001401

```

```

:*****
AT15: 15 ;TEST NUMBER 15 *
      AT16 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ANA ;SCOPE ENTRY POINT *
:*****
;TEST THAT RXCSR BIT 0 (DATA TERMINAL READY) CAN BE SET, NOT CLEARED BY RESET, AND CLEAR
ANA: BIS #BIT0,@RXCSR ;SET RXCSR BIT 0.
      BIT #BIT0,@RXCSR ;SEE IF BIT IS SET.
      BNE ANB ;BRANCH IF BIT IS SET.
      ERROR
      BR AND
ANB: SRESET ;ISSUE RESET.
      BIT #BIT0,@RXCSR ;SEE IF BIT IS STILL SET.
      BNE ANC ;BRANCH IF BIT SET.
      ERROR ;RESET CLEARED RXCSR BIT 0.
      BR AND
ANC: BIC #BIT0,@RXCSR ;CLEAR RXCSR BIT 0.
      BIT #BIT0,@RXCSR ;SEE IF BIT IS CLEAR.
      BEQ AND ;BRANCH IF BIT IS CLEAR.
      ERROR ;RXCSR BIT 0 FAILED TO CLEAR.
AND: SCOPE ;SCOPE
:*****
AT16: 16 ;TEST NUMBER 16 *
      AT17 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ANW ;SCOPE ENTRY POINT *
:*****
;TEST THAT RXCSR BIT 1 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT
ANW: BIT #BIT1,@RXCSR ;SEE IF BIT 1 IS CLEAR
      BEQ ANX ;BRANCH IF CLEAR
      ERROR ;RESET DID NOT CLEAR RXCSR BIT 1
      BR ANZ
ANX: BIS #BIT1,@RXCSR ;SET RXCSR BIT1
      BIT #BIT1,@RXCSR ;SEE IF BIT IS SET
      BNE ANY ;BRANCH IF SET
      ERROR ;RXCSR BIT 1 FAILED TO SET
      BR ANZ
ANY: BIC #BIT1,@RXCSR ;CLEAR RXCSR BIT 1
      BIT #BIT1,@RXCSR ;SEE IF BIT IS CLEAR
      BEQ ANZ
      ERROR ;RXCSR BIT 1 FAILED TO CLEAR
ANZ: BIS #BIT1,@RXCSR ;SET RXCSR BIT 1
      SRESET ;ISSUE RESET TO CLEAR BIT
      SCOPE ;SCOPE
:*****
AT17: 17 ;TEST NUMBER 17 *
      AT20 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      APA ;SCOPE ENTRY POINT *
:*****
;TEST THAT RXCSR BIT2 IS CLEAR AND CAN BE READ RELIABLY.
APA: BIT #BIT2,@RXCSR ;SEE IF RXCSR BIT2 IS CLEAR.
      BEQ APB ;BRANCH IF BIT IS CLEAR.

```

```
1478 005312 104003          ERROR          ;RXCSR BIT2 IS NOT CLEAR.
1479 005314 104012          APB: SCOPE          ;SCOPE
1480          ;*****
1481 005316 000020          AT20: 20           ;TEST NUMBER 20          *
1482 005320 005416          AT21           ;ADDRESS OF NEXT TEST   *
1483 005322 000144          100.          ;TEST ITERATION COUNT   *
1484 005324 005326          AQA           ;SCOPE ENTRY POINT     *
1485          ;*****
1486          ;TEST THAT RXCSR BIT3 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1487 005326 032777 000010 173544 AQA: BIT #BIT3,@RXCSR ;SEE IF RXCSR BIT3 IS CLEAR.
1488 005334 001402          BEQ AQB        ;BRANCH IF BIT IS CLEAR.
1489 005336 104003          ERROR        ;RESET DID NOT CLEAR RXCSR BIT3
1490 005340 000421          BR AQB        ;
1491 005342 052777 000010 173530 AQB: BIS #BIT3,@RXCSR ;SET RXCSR BIT3.
1492 005350 032777 000010 173522 AQB: BIT #BIT3,@RXCSR ;SEE IF BIT IS SET.
1493 005356 001002          BNE AQC        ;BRANCH IF BIT IS SET.
1494 005360 104003          ERROR        ;RXCSR BIT3 FAILED TO SET.
1495 005362 000410          BR AQC        ;
1496 005364 042777 000010 173506 AQC: BIC #BIT3,@RXCSR ;CLEAR RXCSR BIT3
1497 005372 032777 000010 173500 AQC: BIT #BIT3,@RXCSR ;SEE IF BIT IS CLEAR.
1498 005400 001401          BEQ AQB        ;
1499 005402 104003          ERROR        ;RXCSR BIT3 FAILED TO CLEAR.
1500 005404 052777 000010 173466 AQB: BIS #BIT3,@RXCSR ;SET RXCSR BIT3.
1501 005412 104011          SRESET       ;ISSUE RESET TO CLEAR BIT.
1502 005414 104012          SCOPE        ;SCOPE
1503          ;*****
1504 005416 000021          AT21: 21           ;TEST NUMBER 21          *
1505 005420 005516          AT22           ;ADDRESS OF NEXT TEST   *
1506 005422 000144          100.          ;TEST ITERATION COUNT   *
1507 005424 005426          ARA           ;SCOPE ENTRY POINT     *
1508          ;*****
1509          ;TEST THAT RXCSR BIT4 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1510 005426 032777 000020 173444 ARA: BIT #BIT4,@RXCSR ;SEE IF RXCSR BIT4 IS CLEAR.
1511 005434 001402          BEQ ARB        ;BRANCH IF BIT IS CLEAR.
1512 005436 104003          ERROR        ;RESET DID NOT CLEAR RXCSR BIT4
1513 005440 000421          BR ARB        ;
1514 005442 052777 000020 173430 ARB: BIS #BIT4,@RXCSR ;SET RXCSR BIT4.
1515 005450 032777 000020 173422 ARB: BIT #BIT4,@RXCSR ;SEE IF BIT IS SET.
1516 005456 001002          BNE ARC        ;BRANCH IF BIT IS SET.
1517 005460 104003          ERROR        ;RXCSR BIT4 FAILED TO SET.
1518 005462 000410          BR ARC        ;
1519 005464 042777 000020 173406 ARC: BIC #BIT4,@RXCSR ;CLEAR RXCSR BIT4
1520 005472 032777 000020 173400 ARC: BIT #BIT4,@RXCSR ;SEE IF BIT IS CLEAR.
1521 005500 001401          BEQ ARD        ;
1522 005502 104003          ERROR        ;RXCSR BIT4 FAILED TO CLEAR.
1523 005504 052777 000020 173366 ARD: BIS #BIT4,@RXCSR ;SET RXCSR BIT4.
1524 005512 104011          SRESET       ;ISSUE RESET TO CLEAR BIT.
1525 005514 104012          SCOPE        ;SCOPE
1526          ;*****
1527          ;*****
1528 005516 000022          AT22: 22           ;TEST NUMBER 22          *
1529 005520 005542          AT23           ;ADDRESS OF NEXT TEST   *
1530 005522 000144          100.          ;TEST ITERATION COUNT   *
1531 005524 005526          ARBA          ;SCOPE ENTRY POINT     *
1532          ;*****
1533          ;TEST THAT PARITY INDICATOR (BIT5 RXCSR) IS CLEAR
```

```

1534                                     ;AND CAN BE READ RELIABLY.
1535
1536 005526 032777 000040 173344 ARBA: BIT #BIT5,@RXCSR ;SEE IF PARITY INDICATOR IS CLEAR
1537 005534 001401 BEQ ARBB ;BRANCH IF CLEAR
1538 005536 104003 ERROR ;IS NOT CLEAR
1539 005540 104012 ARBB: SCOPE ;SCOPE
1540
1541
1542 005542 000023 AT23: 23 ;TEST NUMBER 23 *
1543 005544 005650 AT24 ;ADDRESS OF NEXT TEST *
1544 005546 000144 100. ;TEST ITERATION COUNT *
1545 005550 005552 ASA ;SCOPE ENTRY POINT *
1546
1547 ;*****
1548 005552 012737 000340 177776 ASA: MOV #PRTY7,PSW ;SET PRIORITY 7.
1549 005560 032777 000100 173312 BIT #BIT6,@RXCSR ;SEE IF RXCSR BIT6 IS CLEAR.
1550 005566 001402 BEQ ASB ;BRANCH IF BIT IS CLEAR.
1551 005570 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT6
1552 005572 000421 BR ASD
1553 005574 052777 000100 173276 ASB: BIS #BIT6,@RXCSR ;SET RXCSR BIT6.
1554 005602 032777 000100 173270 BIT #BIT6,@RXCSR ;SEE IF BIT IS SET.
1555 005610 001002 BNE ASC ;BRANCH IF BIT IS SET.
1556 005612 104003 ERROR ;RXCSR BIT6 FAILED TO SET.
1557 005614 000410 BR ASD
1558 005616 042777 000100 173254 ASC: BIC #BIT6,@RXCSR ;CLEAR RXCSR BIT6
1559 005624 032777 000100 173246 BIT #BIT6,@RXCSR ;SEE IF BIT IS CLEAR.
1560 005632 001401 BEQ ASD
1561 005634 104003 ERROR ;RXCSR BIT6 FAILED TO CLEAR.
1562 005636 052777 000100 173234 ASD: BIS #BIT6,@RXCSR ;SET RXCSR BIT6.
1563 005644 104011 SRESET ;ISSUE RESET TO CLEAR BIT.
1564 005646 104012 SCOPE ;SCOPE
1565
1566 005650 000024 AT24: 24 ;TEST NUMBER IS 24 *
1567 005652 005674 AT25 ;ADDRESS OF NEXT TEST *
1568 005654 000144 100. ;TEST ITERATION COUNT *
1569 005656 005660 ATA ;SCOPE ENTRY POINT *
1570
1571 ;*****
1572 005660 032777 000200 173212 ATA: BIT #BIT7,@RXCSR ;SEE IF RXCSR BIT7 IS CLEAR.
1573 005666 001401 BEQ ATB ;BRANCH IF BIT IS CLEAR.
1574 005670 104003 ERROR ;RXCSR BIT7 IS NOT CLEAR.
1575 005672 104012 ATB: SCOPE ;SCOPE
1576
1577
1578 005674 000025 AT25: 25 ;TEST NUMBER 25 *
1579 005676 005774 AT26 ;ADDRESS OF NEXT TEST *
1580 005700 000144 100. ;TEST ITERATION COUNT *
1581 005702 005704 AUA ;SCOPE ENTRY POINT *
1582
1583 ;*****
1584 005704 032777 000400 173166 AUA: BIT #BIT8,@RXCSR ;SEE IF RXCSR BIT8 IS CLEAR.
1585 005712 001402 BEQ AUB ;BRANCH IF BIT IS CLEAR.
1586 005714 104003 ERROR ;RESET DID NOT CLEAR RXCSR BIT8
1587 005716 000421 BR AUD
1588 005720 052777 000400 173152 AUB: BIS #BIT8,@RXCSR ;SET RXCSR BIT8.
1589 005726 032777 000400 173144 BIT #BIT8,@RXCSR ;SEE IF BIT IS SET.

```

```

1590 005734 001002          BNE      AUC          ;BRANCH IF BIT IS SET.
1591 005736 104003          ERROR
1592 005740 000410          BR       AUD          ;RXCSR BIT8 FAILED TO SET.
1593 005742 042777 000400 173130 AUC:    BIC      #BIT8,@RXCSR ;CLEAR RXCSR BIT8
1594 005750 032777 000400 173122          BIT      #BIT8,@RXCSR ;SEE IF BIT IS CLEAR.
1595 005756 001401          BEQ     AUD
1596 005760 104003          ERROR          ;RXCSR BIT8 FAILED TO CLEAR.
1597 005762 052777 000400 173110 AUD:    BIS      #BIT8,@RXCSR ;SET RXCSR BIT8.
1598 005770 104011          SRESET          ;ISSUE RESET TO CLEAR BIT.
1599 005772 104012          SCOPE          ;SCOPE
1600
1601 005774 000026          AT26:   26          ;TEST NUMBER 26 *
1602 005776 006074          AT27          ;ADDRESS OF NEXT TEST *
1603 006000 000144          100.         ;TEST ITERATION COUNT *
1604 006002 006004          AVA          ;SCOPE ENTRY POINT *
1605
1606          ;*****
1606          ;TEST THAT RXCSR BIT9 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1607 006004 032777 001000 173066 AVA:    BIT      #BIT9,@RXCSR ;SEE IF RXCSR BIT9 IS CLEAR.
1608 006012 001402          BEQ     AVB          ;BRANCH IF BIT IS CLEAR.
1609 006014 104003          ERROR          ;RESET DID NOT CLEAR RXCSR BIT9
1610 006016 000421          BR       AVD
1611 006020 052777 001000 173052 AVB:    BIS      #BIT9,@RXCSR ;SET RXCSR BIT9.
1612 006026 032777 001000 173044          BIT      #BIT9,@RXCSR ;SEE IF BIT IS SET.
1613 006034 001002          BNE     AVC          ;BRANCH IF BIT IS SET.
1614 006036 104003          ERROR          ;RXCSR BIT9 FAILED TO SET.
1615 006040 000410          BR       AVD
1616 006042 042777 001000 173030 AVC:    BIC      #BIT9,@RXCSR ;CLEAR RXCSR BIT9
1617 006050 032777 001000 173022          BIT      #BIT9,@RXCSR ;SEE IF BIT IS CLEAR.
1618 006056 001401          BEQ     AVD
1619 006060 104003          ERROR          ;RXCSR BIT9 FAILED TO CLEAR.
1620 006062 052777 001000 173010 AVD:    BIS      #BIT9,@RXCSR ;SET RXCSR BIT9.
1621 006070 104011          SRESET          ;ISSUE RESET TO CLEAR BIT.
1622 006072 104012          SCOPE          ;SCOPE
1623
1624 006074 000027          AT27:   27          ;TEST NUMBER 27 *
1625 006076 006174          AT30          ;ADDRESS OF NEXT TEST *
1626 006100 000144          100.         ;TEST ITERATION COUNT *
1627 006102 006104          AWA          ;SCOPE ENTRY POINT *
1628
1629          ;*****
1629          ;TEST THAT RXCSR BIT10 CAN BE SET, CLEARED, AND THAT RESET CLEARS IT.
1630 006104 032777 002000 172766 AWA:    BIT      #BIT10,@RXCSR ;SEE IF RXCSR BIT10 IS CLEAR.
1631 006112 001402          BEQ     AWB          ;BRANCH IF BIT IS CLEAR.
1632 006114 104003          ERROR          ;RESET DID NOT CLEAR RXCSR BIT10
1633 006116 000421          BR       AWD
1634 006120 052777 002000 172752 AWB:    BIS      #BIT10,@RXCSR ;SET RXCSR BIT10.
1635 006126 032777 002000 172744          BIT      #BIT10,@RXCSR ;SEE IF BIT IS SET.
1636 006134 001002          BNE     AWC          ;BRANCH IF BIT IS SET.
1637 006136 104003          ERROR          ;RXCSR BIT10 FAILED TO SET.
1638 006140 000410          BR       AWD
1639 006142 042777 002000 172730 AWC:    BIC      #BIT10,@RXCSR ;CLEAR RXCSR BIT10
1640 006150 032777 002000 172722          BIT      #BIT10,@RXCSR ;SEE IF BIT IS CLEAR.
1641 006156 001401          BEQ     AWD
1642 006160 104003          ERROR          ;RXCSR BIT10 FAILED TO CLEAR.
1643 006162 052777 002000 172710 AWD:    BIS      #BIT10,@RXCSR ;SET RXCSR BIT10.
1644 006170 104011          SRESET          ;ISSUE RESET TO CLEAR BIT.
1645 006172 104012          SCOPE          ;SCOPE

```



```
1646 :*****
1647 006174 000030 AT30: 30 ;TEST NUMBER 30 *
1648 006176 006220 ; AT31 ;ADDRESS OF NEXT TEST *
1649 006200 000144 100. ;TEST ITERATION COUNT *
1650 006202 006204 AXA ;SCOPE ENTRY POINT *
1651 :*****
1652 :TEST THAT RXCSR BIT12 IS CLEAR AND CAN BE READ RELIABLY.
1653 006204 032777 010000 172666 AXA: BIT #BIT12,@RXCSR ;SEE IF RXCSR BIT12 IS CLEAR.
1654 006212 001401 BEQ AXB ;BRANCH IF BIT IS CLEAR.
1655 006214 104003 ERROR ;RXCSR BIT12 IS NOT CLEAR.
1656 006216 104012 AXB: SCOPE ;SCOPE
1657
1658 :*****
1659 006220 000031 AT31: 31 ;TEST NUMBER 31 *
1660 006222 006244 ; AT32 ;ADDRESS OF NEXT TEST *
1661 006224 000144 100. ;TEST ITERATION COUNT *
1662 006226 006230 AYA ;SCOPE ENTRY POINT *
1663 :*****
1664 :TEST THAT RXCSR BIT13 IS CLEAR AND CAN BE READ RELIABLY.
1665 006230 032777 020000 172642 AYA: BIT #BIT13,@RXCSR ;SEE IF RXCSR BIT13 IS CLEAR.
1666 006236 001401 BEQ AYB ;BRANCH IF BIT IS CLEAR.
1667 006240 104003 ERROR ;RXCSR BIT13 IS NOT CLEAR.
1668 006242 104012 AYB: SCOPE ;SCOPE
1669
1670 :*****
1671 :TEST THAT RXCSR BIT14 IS CLEAR AND CAN BE READ RELIABLY.
1672 006244 000032 AT32: 32 ;TEST NUMBER 32 *
1673 006246 006270 ; AT33 ;ADDRESS OF NEXT TEST *
1674 006250 000144 100. ;TEST ITERATION COUNT *
1675 006252 006254 AZA ;SCOPE ENTRY POINT *
1676 :*****
1677 :TEST THAT RXCSR BIT14 IS CLEAR AND CAN BE READ RELIABLY.
1678 006254 032777 040000 172616 AZA: BIT #BIT14,@RXCSR ;SEE IF RXCSR BIT14 IS CLEAR.
1679 006262 001401 BEQ AZB ;BRANCH IF BIT IS CLEAR.
1680 006264 104003 ERROR ;RXCSR BIT14 IS NOT CLEAR.
1681 006266 104012 AZB: SCOPE ;SCOPE
1682 :*****
1683 006270 000033 AT33: 33 ;TEST NUMBER 33 *
1684 006272 006314 ; AT34 ;ADDRESS OF NEXT TEST *
1685 006274 000144 100. ;TEST ITERATION COUNT *
1686 006276 006300 AAAA ;SCOPE ENTRY POINT *
1687 :*****
1688 :TEST THAT RXCSR BIT15 IS CLEAR AND CAN BE READ RELIABLY.
1689 006300 032777 100000 172572 AAAA: BIT #BIT15,@RXCSR ;SEE IF RXCSR BIT15 IS CLEAR.
1690 006306 001401 BEQ AAAB ;BRANCH IF BIT IS CLEAR.
1691 006310 104003 ERROR ;RXCSR BIT15 IS NOT CLEAR.
1692 006312 104012 AAAB: SCOPE ;SCOPE
```

```
1693
1694
1695
1696
1697
1698
1699
1700 006314 000034
1701 006316 006372
1702 006320 000144
1703 006322 006324
1704
1705
1706
1707
1708 006324 052777 000001 172546
1709 006332 032777 000004 172540
1710 006340 001002
1711 006342 104003
1712 006344 000410
1713 006346 042777 000001 172524
1714 006354 032777 000004 172516
1715 006362 001401
1716 006364 104003
1717 006366 104011
1718 006370 104012
1719
1720 006372 000035
1721 006374 006532
1722 006376 000144
1723 006400 006402
1724
1725
1726
1727
1728 006402 042777 000001 172470
1729 006410 017737 172464 001260
1730 006416 032777 040000 172454
1731 006424 001402
1732 006426 104003
1733 006430 000436
1734 006432 005277 172442
1735 006436 000004
1736
1737
1738 006440 017737 172434 001260
1739 006446 032737 040000 001260
1740 006454 001002
1741 006456 104003
1742 006460 000422
1743 006462 032777 040000 172410
1744
1745 006470 001402
1746 006472 104003
1747 006474 000414
1748

;ALL PREVIOUS TESTS MUST HAVE BEEN RUN SUCCESSFULLY PRIOR
;   RUNNING THE FOLLOWING TESTS. ALSO, THE JUMPER CONNECTOR
;MUST BE INSERTED IN THE DC11 CABLE. TO THE MODEM. COMMENTS
;REFER TO OPERATION WITH JUMPER INSERTED.
:
:*****
AT34: 34 ;TEST NUMBER 34
      AT35 ;ADDRESS OF NEXT TEST
      100. ;TEST ITERATION COUNT
      AFBA ;SCOPE ENTRY POINT
:*****
;TEST THAT CARRIER DETECT SETS AND CLEARS WHEN DATA TERMINAL
;READY SETS AND CLEARS.
:
AFBA: BIS #BIT0,@RXCSR ;SET DATA TERMINAL READY
      BIT #BIT2,@RXCSR ;TEST CARRIER DETECT
      BNE AFBB ;SHOULD BE SET
      ERROR ;WASN'T
      BR AFBC
AFBB: BIC #BIT0,@RXCSR ;CLEAR DATA TERMINAL READY
      BIT #BIT2,@RXCSR ;TEST CARRIER DETECT
      BEQ AFBC
      ERROR ;WAS SET, ERROR
AFBC: SRESET
      SCOPE
:*****
AT35: 35 ;TEST NUMBER 35
      AT36 ;ADDRESS OF NEXT TEST
      100. ;TEST ITERATION COUNT
      AGBA ;SCOPE ENTRY POINT
:*****
;TEST THAT CARRIER TRANSITION (BIT 14) SETS WHEN CARRIER DETECT
;CHANGES STATE, AND IS CLEARED WHEN RXCSR IS READ.
:
AGBA: BIC #BIT0,@RXCSR ;CLEAR DATA TERMINAL READY
      MOV @RXCSR,RXCSRT ;READ RXCSR
      BIT #BIT14,@RXCSR ;TEST CARRIER TRANSITION
      BEQ AGBB ;WAS CLEAR GO TO AGBB
      ERROR ;WASN'T CLEAR
      BR AGBE ;GO TO SCOPE
AGBB: INC @RXCSR ;SETTING DATA TERMINAL READY
      IOT ;CAUSES CARRIER DETECT TO SET
      ;WHICH CAUSES CARRIER TRANSITION
      ;TO SET.
      MOV @RXCSR,RXCSRT ;MOVE RXCSR TO TEMPORARY LOCATION
      BIT #BIT14,RXCSRT ;TEST CARRIER TRANSITION
      BNE AGBC ;SHOULD BE SET GO TO AGBC
      ERROR ;WAS CLEAR
      BR AGBE ;GO TO SCOPE
AGBC: BIT #BIT14,@RXCSR ;CARRIER TRANSITION BIT SHOULD
      ;HAVE BEEN CLEARED
      BEQ AGBD ;IT WAS GO TO AGBD
      ERROR ;IT WASN'T
      BR AGBE ;GO TO SCOPE
```

```
1749 006476 042777 000001 172374 AGBD: BIC #BIT0,@RXCSR ;CLEARING DATA TERMINAL READY
1750 ;CAUSES CARRIER DETECT TO CLEAR
1751 ;BUT CARRIER TRANSITION
1752 ;WILL NOT SET
1753 006504 017737 172370 001260 MOV @RXCSR,RXCSRT ;MOV RXCSR TO TEMPORARY LOCATION
1754 006512 032737 040000 001260 BIT #BIT14,RXCSRT ;TEST CARRIER TRANSITION
1755 006520 001402 BEQ AGBE ;SHOULD BE CLEAR
1756 006522 104003 ERROR ;IT WASN'T
1757 006524 000400 BR AGBE
1758 006526 104011 AGBE: SRESET ;ISSUE RESET
1759 006530 104012 SCOPE ;SCOPE
1760 ;*****
1761
1762 006532 000036 AT36: 36 ;TEST NUMBER 36
1763 006534 006646 AT37 ;ADDRESS OF NEXT TEST
1764 006536 000144 100. ;TEST ITERATION COUNT
1765 006540 006542 AMBA ;SCOPE ENTRY POINT
1766 ;*****
1767 ;TEST THAT CARRIER TRANSITION SETTING CAUSES ERROR (BIT 15 RXCSR) TO
1768 ;SET AND THAT READING RXCSR CLEARS ERROR.
1769
1770 006542 042777 000001 172330 AMBA: BIC #BIT0,@RXCSR ;CLEAR DATA TERMINAL READY
1771 006550 005277 172324 INC @RXCSR ;SET DATA TERMINAL READY
1772 006554 000004 IOT
1773 006556 017737 172316 001260 MOV @RXCSR,+RXCSRT ;MOVE RXCSR TO TEMPORARY LOCATION
1774 006564 032737 100000 001260 BIT #BIT15,RXCSRT ;TEST ERROR BIT
1775 006572 001002 BNE AMBB ;ERROR BIT SHOULD BE SET
1776 006574 104003 ERROR
1777 006576 000421 BR AMBD
1778 006600 032777 100000 172272 AMBB: BIT #BIT15,@RXCSR ;TEST ERROR BIT
1779 006606 001402 BEQ AMBC ;SHOULD BE CLEAR
1780 006610 104003 ERROR
1781 006612 000413 BR AMBD
1782 006614 042777 000001 172256 AMBC: BIC #BIT0,@RXCSR ;CLEAR DATA TERMINAL READY
1783 006622 017737 172252 001260 MOV @RXCSR,RXCSRT ;MOV RXCSR TO TEMPORARY LOCATION
1784 006630 032737 100000 001260 BIT #BIT15,RXCSRT ;TEST ERROR BIT
1785 006636 001401 BEQ AMBD ;SHOULD BE CLEAR
1786 006640 104003 ERROR
1787 006642 104011 AMBD: SRESET ;ISSUE RESET
1788 006644 104012 SCOPE ;SCOPE
1789
1790
1791 ;*****
1792 006646 000037 AT37: 37 ;TEST NUMBER 37
1793 006650 006742 AT40 ;ADDRESS OF NEXT TEST
1794 006652 000144 100. ;TEST ITERATION COUNT
1795 006654 006656 AJBA ;SCOPE ENTRY POINT
1796 ;*****
1797 ;TEST THAT CLEAR TO SEND (BIT1) SET/CLEARS WHEN DATA TERMINAL
1798 ;READY SETS/CLEARS.
1799
1800 006656 042777 000001 172214 AJBA: BIC #BIT0,@RXCSR ;CLEAR DATA TERMINAL READY
1801 006664 032777 000004 172212 BIT #BIT2,@TXCSR ;TEST CLEAR TO SEND
1802 006672 001400 BEQ AJBB
1803
1804
```

```
1805 006674 052777 000001 172176 AJBB: BIS #BIT0,@RXCSR ;SET DATA TERMINAL READY
1806 006702 032777 000002 172174 BIT #BIT1,@TXCSR ;TEST CLEAR TO SEND
1807 006710 001002 BNE AJBC ;BRANCH IF SET
1808 006712 104003 ERROR ;CLEAR TO SEND SHOULD BE SET
1809 006714 000410 BR AJBD
1810 006716 042777 000001 172154 AJBC: BIC #BIT0,@RXCSR ;CLEAR DATA TERMINAL READY
1811 006724 032777 000002 172152 BIT #BIT1,@TXCSR ;TEST CLEAR TO SEND
1812 006732 001401 BEQ AJBD
1813 006734 104003 ERROR ;CLEAR TO SEND SHOULD BE CLEAR
1814 006736 104011 AJBD: SRESET ;ISSUE RESET
1815 006740 104012 SCOPE ;SCOPE
1816 *****
1817 006742 000040 AT40: 40 ;TEST NUMBER 40
1818 006744 007060 AT41 ;ADDRESS OF NEXT TEST
1819 006746 000144 100. ;TEST ITERATION COUNT
1820 006750 006752 AKBA ;SCOPE ENTRY POINT
1821 *****
1822 ;TEST THAT RING (BIT 13 RXCSR) SETS WHEN REQUEST TO
1823 ;SEND SETS AND THEN CLEARS; AND RING CLEARS WHEN RXCSR IS READ.
1824 ;AND THAT RESET CLEARS RING.
1825
1826 006752 042777 000001 172124 AKBA: BIC #BIT0,@TXCSR ;CLEAR REQUEST TO SEND
1827 006760 052777 000001 172116 AKBB: BIS #BIT0,@TXCSR ;SET REQUEST TO SEND
1828 006766 042777 000001 172110 BIC #BIT0,@TXCSR
1829 006774 032777 020000 172076 BIT #BIT13,@RXCSR ;TEST RING
1830 007002 001002 BNE AKBC
1831 007004 104003 ERROR ;RING SHOULD BE SET
1832 007006 000422 BR AKBE
1833 007010 032777 020000 172062 AKBC: BIT #BIT13,@RXCSR ;TEST RING
1834 007016 001402 BEQ AKBD
1835 007020 104003 ERROR ;RING SHOULD BE CLEAR
1836 007022 000414 BR AKBE
1837 007024 052777 000001 172052 AKBD: BIS #BIT0,@TXCSR ;SET
1838 007032 042777 000001 172044 BIC #BIT0,@TXCSR ;RING
1839 007040 000005 RESET
1840 007042 032777 020000 172030 BIT #BIT13,@RXCSR ;TEST RING
1841 007050 001401 BEQ AKBE ;BRANCH IF CLEAR
1842 007052 104003 ERROR ;RING SHOULD BE CLEAR AFTER RESET
1843 ;BUT WAS SET
1844 007054 104011 AKBE: SRESET ;ISSUE RESET
1845 007056 104012 SCOPE ;SCOPE
1846 *****
1847 007060 000041 AT41: 41 ;TEST NUMBER 41
1848 007062 007166 AT42 ;ADDRESS OF NEXT TEST
1849 007064 000144 100. ;TEST ITERATION COUNT
1850 007066 007070 AOBA ;SCOPE ENTRY POINT
1851 *****
1852 ;TEST THAT ERROR (BIT 15 RXCSR) SETS WHEN RING SETS.
1853
1854 007070 042777 000001 172006 AOBA: BIC #BIT0,@TXCSR ;SET REQUEST TO SEND
1855 007076 032777 100000 171774 BIT #BIT15,@RXCSR ;TEST ERROR BIT
1856 007104 001402 BEQ AOBB
1857 007106 104003 ERROR
1858 007110 000424 BR AOBD
1859 007112 052777 000001 171764 AOBB: BIS #BIT0,@TXCSR ;SET REQUEST TO SEND
1860 007120 042777 000001 171756 BIC #BIT0,@TXCSR ;CLEAR REQUEST TO SEND
```



1917 007354 000006  
1918 007356 004537 003300  
1919 007362 001256  
1920 007364 015476  
1921 007366 000006  
1922 007370 104015  
1923 007372 015446  
1924 007374 104012

6  
JSR %5,0ACNV  
TXCSRT  
ATXWAS  
6  
ERROR1  
ATXCSR  
ABAB: SCOPE

;  
;#OF DIGITS TO CONVERT.  
;GO TO OCTAL TO ASCII CONVERT.  
;SOURCE ADDR.  
;DESTINATION ADDR.  
;#OF DIGITS TO CONVERT.  
;RESET FAILED TO CLEAR ALL BITS EXCEPT  
;BIT 7 - SEE PRINTOUT  
;SCOPE

```

1925
1926 007376 000044
1927 007400 007552
1928 007402 000144
1929 007404 007406
1930
1931
1932
1933 007406 012737 000340 177776
1934 007414 042777 000001 171456
1935 007422 012777 177775 171450
1936 007430 052777 000030 171446
1937 007436 005077 171444
1938 007442 105777 171436
1939 007446 100375
1940 007450 012777 000001 171430
1941 007456 105777 171422
1942 007462 100375
1943 007464 104011
1944 007466 017737 171406 001260
1945 007474 022737 000005 001260
1946 007502 001417
1947 007504 012737 000005 001264
1948 007512 004537 003300
1949 007516 001264
1950 007520 015520
1951 007522 000006
1952 007524 004537 003300
1953 007530 001260
1954 007532 015535
1955 007534 000006
1956 007536 104015
1957 007540 015505
1958 007542 042777 000001 171330
1959 007550 104012
1960
1961 007552 000045
1962 007554 007602
1963 007556 000144
1964 007560 007562
1965
1966
1967 007562 005077 171320
1968 007566 105777 171312
1969 007572 100001
1970 007574 104003
1971 007576 104011
1972 007600 104012

;*****
AT44: 44 ;TEST NUMBER 44 *
      AT45 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ACAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RESET CLEARS ALL RXCSR BITS EXCEPT BIT 0 (DATA TERMINAL READY)
;RING, AND THE BREAK BIT.
ACAA: MOV #PRTY7,PSW ;SET PRIORITY 7
      BIC #BIT0,@RXCSR ;CLEAR DATA TERM.READY
      MOV #177775,@RXCSR ;SET ALL POSSIBLE BITS IN RXCSR
      BIS #30,@TXCSR ;SET MAINT BIT
      CLR @TXBUF ;TRANSMIT A CHAR
      TSTB @TXCSR ;WAIT FOR
      BPL -4 ;TRANSMITTER TO FINISH
      MOV #1,@TXBUF ;TRANSMIT ANOTHER CHAR.
      TSTB @TXCSR ;WAIT FOR
      BPL -4 ;TRANSMITTER TO FINISH
      SRESET ;ISSUE RESET TO CLEAR BITS.
      MOV @RXCSR,RXCSRT ;MOVE RXCSR CONTENTS TO RXCSRT
      CMP #5,RXCSRT ;SEE IF ONLY BIT 0 IS SET
      BEQ ACAB ;BRANCH IF ONLY BIT 0 IS SET.
      MOV #5,TEMP
      JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
      TEMP ;SOURCE ADDR.
      ARXSB ;DESTINATION ADDR.
      6 ;#OF DIGITS TO CONVERT.
      JSR %5,OACNV ;GO TO OCTAL TO ASCII CONVERT.
      RXCSRT ;SOURCE ADDR.
      ARXWAS ;DESTINATION ADDR.
      6 ;#OF DIGITS TO CONVERT.
      ERROR1 ;RESET FAILED TO CLEAR ALL BITS EXCEPT
      ARXCSR ;BIT 0. SEE ERROR PRINTOUT.
ACAB: BIC #BIT0,@RXCSR ;CLEAR DATA TERM. READY
      SCOPE ;SCOPE
;*****
AT45: 45 ;TEST NUMBER 45 *
      AT46 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ADAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT LOADING TXBUF (TRANSMITTER BUFFER) CLEARS TXCSR BIT 7 (READY)
ADAA: CLR @TXBUF ;LOAD TXBUF
      TSTB @TXCSR ;TEST TXCSR BIT 7 (READY BIT)
      BPL ADAR ;BRANCH IF BIT NOT SET.
      ERROR ;ERROR. LOADING TXBUF FAILED TO CLEAR READY.
ADAB: SRESET ;ISSUE RESET TO SET READY.
      SCOPE ;SCOPE.

```



```
1973 :*****
1974 007602 000046 AT46: 46 ;TEST NUMBER 46 *
1975 007604 007634 AT47 ;ADDRESS OF NEXT TEST *
1976 007606 000012 10. ;TEST ITERATION COUNT *
1977 007610 007612 AEAA ;SCOPE ENTRY POINT *
1978 :*****
1979 ;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 1000 MSECS AFTER
1980 ;LOADING TXBUF WITH TRANSMIT SPEED SET TO 00 (TXCSR BITS 3 AND 4)
1981 007612 005077 171270 AEAA: CLR @TXBUF ;LOAD TXBUF
1982 007616 104016 DELAY ;DELAY 1000 MSECS APPROX.
1983 007620 001750 1000.
1984 007622 105777 171256 TSTB @TXCSR ;SEE IF READY BIT IS SET
1985 007626 100401 BMI AEAB ;BRANCH IF READY IS SET
1986 007630 104003 ERROR ;READY NOT SET 200 MSECS AFTER BUFFER
1987 ;LOAD. TX SPEED = 00.
1988 007632 104012 AEAB: SCOPE ;SCOPE
1989 :*****
1990 :*****
1991 007634 000047 AT47: 47 ;TEST NUMBER 47 *
1992 007636 007674 AT50 ;ADDRESS OF NEXT TEST *
1993 007640 000012 10. ;TEST ITERATION COUNT *
1994 007642 007644 AFAA ;SCOPE ENTRY POINT *
1995 :*****
1996 ;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 500 MSECS AFTER
1997 ;LOADING TXBUF, WITH TRANSMIT SPEED SET TO 01 (TXCSR BITS 3 AND 4).
1998 007644 052777 000010 171232 AFAA: BIS #10,@TXCSR ;SET TX SPEED TO 01.
1999 007652 005077 171230 CLR @TXBUF ;LOAD TXBUF
2000 007656 104016 DELAY ;DELAY 500 MSECS
2001 007660 000764 500.
2002 007662 105777 171216 TSTB @TXCSR ;SEE IF READY BIT IS SET
2003 007666 100401 BMI AFAB ;BRANCH IF READY IS SET
2004 007670 104003 ERROR ;READY NOT SET 200 MSECS AFTER BUFFER
2005 ;LOAD. TX SPEED = 01.
2006 007672 104012 AFAB: SCOPE ;SCOPE.
2007 :*****
2008 007674 000050 AT50: 50 ;TEST NUMBER 50 *
2009 007676 007734 AT51 ;ADDRESS OF NEXT TEST *
2010 007700 000012 10. ;TEST ITERATION COUNT *
2011 007702 007704 AGAA ;SCOPE ENTRY POINT *
2012 :*****
2013 ;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 400 MSECS AFTER
2014 ;LOADING TXBUF, WITH TRANSMIT SPEED SET TO 10 (TXCSR BITS 3 AND 4).
2015 007704 052777 000020 171172 AGAA: BIS #20,@TXCSR ;SET TX SPEED TO 10.
2016 007712 005077 171170 CLR @TXBUF ;LOAD TXBUF
2017 007716 104016 DELAY ;DELAY 400 MSECS
2018 007720 000620 400.
2019 007722 105777 171156 TSTB @TXCSR ;SEE IF READY BIT IS SET
2020 007726 100401 BMI AGAB ;BRANCH IF READY BIT IS SET
2021 007730 104003 ERROR ;READY NOT SET 200 MSECS AFTER BUFFER
2022 ;LOAD. TX SPEED = 10.
2023 007732 104012 AGAB: SCOPE ;SCOPE
```

```
2024
2025 007734 000051
2026 007736 007774
2027 007740 000012
2028 007742 007744
2029
2030
2031
2032 007744 052777 000030 171132
2033 007752 005077 171130
2034 007756 104016
2035 007760 000372
2036 007762 105777 171116
2037 007766 100401
2038 007770 104003
2039
2040 007772 104012
2041
2042
2043 007774 000052
2044 007776 010212
2045 010000 000144
2046 010002 010004
2047
2048
2049
2050 010004 005037 001246
2051 010010 005037 001250
2052 010014 005037 001252
2053 010020 005037 001254
2054 010024 042777 000030 171052
2055 010032 004737 010146
2056 010036 063737 010210 001246
2057 010044 052777 000010 171032
2058 010052 004737 010146
2059 010056 063737 010210 001250
2060 010064 042777 000030 171012
2061 010072 052777 000020 171004
2062 010100 004737 010146
2063 010104 063737 010210 001252
2064 010112 052777 000030 170764
2065 010120 004737 010146
2066 010124 063737 010210 001254
2067 010132 004737 014334
2068 010136 000402
2069 010140 104015
2070 010142 015544
2071 010144 104012
2072 010146 005037 010210
2073 010152 105777 170726
2074 010156 100375
2075 010160 104016
2076 010162 000024
2077 010164 005077 170716
2078 010170 104016
2079 010172 000001

:*****
AT51: 51 ;TEST NUMBER 51 *
      AT52 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AHAA ;SCOPE ENTRY POINT *
:*****
;TEST THAT READY BIT (TXCSR BIT 7) BECOMES SET NO LATER THAN 250 MSECS AFTER
;LOADING TXBUF, WITH TRANSMIT SPEED SET TO 11 (TXCSR BITS 3 AND 4).
AHAA: BIS #30,@TXCSR ;SET TX SPEED TO 30.
      CLR @TXBUF ;LOAD TXBUF
      DELAY ;DELAY 250 MSECS.
      250.
      TSTB @TXCSR ;SEE IF READY BIT IS SET.
      BMI AHAB ;BRANCH IF READY BIT IS SET.
      ERROR ;READY NOT SET 200 MSECS AFTER
           ;BUFFER LOAD. TX SPEED = 11.
AHAB: SCOPE ;SCOPE
:*****
AT52: 52 ;TEST NUMBER 52 *
      AT53 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AIAA ;SCOPE ENTRY POINT *
:*****
;TEST THAT TRANSMIT SPEEDS ARE ARRANGED IN ASCENDING ORDER BY CHECKING THAT TIME
;TO READY BIT (TXCSR BIT 7) DECREASES AS A HIGHER SPEED IS SELECTED.
AIAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD
      CLR CTRB ;(USED TO COUNT ELAPSED TIME.)
      CLR CTRC
      CLR CTRD
      BIC #30,@TXCSR ;SELECT TX SPEED 0
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRA ;ADD ELAPSED TIME TO CTRA.
      BIS #10,@TXCSR ;SELECT TX SPEED 1
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRB ;ADD ELAPSED TIME TO CTRB.
      BIC #30,@TXCSR ;SELECT TX SPEED 2
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRC ;ADD ELAPSED TIME TO CTRC.
      BIS #30,@TXCSR ;SELECT TX SPEED 3
      JSR %7,AIAS ;OUTPUT CHAR AND TIME.
      ADD AIAST,CTRD ;ADD ELAPSED TIME TO CTRD.
      JSR %7,CMP ;CHECK THAT CTRA THROUGH CTRD CONTAIN
           ;DESCENDING VALUES
      BR AIAF ;TRANSMIT SPEEDS NOT ARRANGED IN
           ;ASCENDING ORDER.
AIAF: SCOPE ;SCOPE
AIAS: CLR AIAST ;CLEAR ELAPSED TIME COUNTER.
      TSTB @TXCSR ;WAIT FOR TX READY.
      BPL .-4
      DELAY ;WAIT 20 MSECS.
      20.
AIASA: CLR @TXBUF ;LOAD TXBUF.
        DELAY ;DELAY 1 MSEC.
        1
```

```

2080 010174 005237 010210          INC      AIAST      ;INCREMENT ELAPSED TIME COUNTER.
2081 010200 105777 170700          TSTB     @TXCSR    ;READY SET?
2082 010204 100371                   BPL      AIASA     ;BRANCH IF READY NOT SET.
2083 010206 000207                   RTS      %7        ;EXIT.
2084 010210 000000          AIAST:  OPEN
2085                                     ;*****
2086 010212 000053          AT53:   53          ;TEST NUMBER 53 *
2087 010214 010376          AT54   AT54          ;ADDRESS OF NEXT TEST *
2088 010216 000144          100.    ;TEST ITERATION COUNT *
2089 010220 010222          AJAA    AJAA          ;SCOPE ENTRY POINT *
2090                                     ;*****
2091                                     ;TEST FOR CORRECT OPERATION OF STOP CODE BIT (TXCSR BIT 8) BY CHECKING THAT TIME.
2092                                     ;REQUIRED TO COMPLETE TRANSMISSION OF 2 CONSECUTIVE CHARACTERS WITH STOP BIT
2093                                     ;SET TO 0 IS LONGER THAN TIME REQUIRED WITH STOP CODE BIT SET TO A 1.
2094 010222 005037 001246          AJAA:   CLR      CTRA      ;CLEAR CTRA AND CTRB
2095 010226 005037 001250          CLR      CTRB      ;(ELAPSED TIME COUNTERS).
2096 010232 042777 000400 170644          BIC      #BIT8,@TXCSR ;SET STOP CODE TO 0 (2 STOP CODES)
2097 010240 004737 010310          JSR     %7,AJAS    ;OUTPUT CHAR AND TIME
2098 010244 063737 010374 001246          ADD     AJAST,CTRA  ;ADD ELAPSED TIME TO CTRA
2099 010252 052777 000400 170624          BIS     #BIT8,@TXCSR ;SET STOP CODE TO 1 (1 STOP CODE)
2100 010260 004737 010310          JSR     %7,AJAS    ;OUTPUT CHARACTER AND TIME.
2101 010264 063737 010374 001250          ADD     AJAST,CTRB  ;ADD ELAPSED TIME TO CTRB
2102 010272 023737 001246 001250          CMP     CTRA,CTRB  ;SEE IF CTRA IS GREATER THAN CTRB
2103 010300 101002                   BHI     AJAB       ;BRANCH IF CTRA IS GREATER.
2104 010302 104015                   ERROR1  ;ERROR. ELAPSED TIME FOR 2 STOP CODE
2105 010304 015606                   ESTPCD  ;OPERATION NOT GREATER THAN FOR 1 STOP
2106                                     ;CODE.
2107 010306 104012          AJAB:   SCOPE
2108 010310 005037 010374          AJAS:   CLR      AJAST   ;CLEAR ELAPSED TIME COUNTER AJAST
2109 010314 105777 170564          TSTB     @TXCSR    ;WAIT FOR TX READY.
2110 010320 100375                   BPL     -.4
2111 010322 104016                   DELAY   ;WAIT 20 MSECS.
2112 010324 000024                   20.
2113 010326 005077 170554          AJASA:  CLR      @TXBUF  ;LOAD TXBUF
2114 010332 104016                   DELAY   ;DELAY 1 MSEC
2115 010334 000001                   1
2116 010336 005237 010374          INC     AJAST      ;INCREMENT ELAPSED TIME COUNTER
2117 010342 105777 170536          TSTB     @TXCSR    ;READY SET?
2118 010346 100371                   BPL     AIASA     ;BRANCH IF READY NOT SET.
2119 010350 005077 170532          CLR     @TXBUF    ;LOAD TXBUF.
2120 010354 104016          AJASB:  DELAY   ;DELAY 1 MSEC.
2121 010356 000001                   1
2122 010360 005237 010374          INC     AJAST      ;INCR ELAPSED TIME COUNTER.
2123 010364 105777 170514          TSTB     @TXCSR    ;READY SET?
2124 010370 100371                   BPL     AJASB     ;BRANCH IF READY NOT SET.
2125 010372 000207                   RTS     %7        ;EXIT
2126 010374 000000          AIAST:  OPEN      ;ELAPSED TIME COUNTER.

```

```
2127
2128
2129 010376 000054
2130 010400 010622
2131 010402 000144
2132 010404 010406
2133
2134
2135
2136
2137 010406 005037 001246
2138 010412 005037 001250
2139 010413 005037 001252
2140 010422 005037 001254
2141 010426 042777 003000 170444
2142 010434 004737 010556
2143 010440 063737 010620 001246
2144 010446 042777 003000 170424
2145 010454 052777 001000 170416
2146 010462 004737 010556
2147 010466 063737 010620 001250
2148 010474 042777 003000 170376
2149 010502 052777 002000 170370
2150 010510 004737 010556
2151 010514 063737 010620 001252
2152 010522 052777 003000 170350
2153 010530 004737 010556
2154 010534 063737 010620 001254
2155 010542 004737 014334
2156 010546 000402
2157 010550 104015
2158 010552 015657
2159 010554 104012
2160 010556 005037 010620
2161 010562 105777 170316
2162 010566 100375
2163 010570 104016
2164 010572 000024
2165 010574 005077 170306
2166 010600 104016
2167 010602 000001
2168 010604 005237 010620
2169 010610 105777 170270
2170 010614 100371
2171 010616 000207
2172 010620 000000

:*****
AT54: 54 ;TEST NUMBER 54 *
      AT55 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AKAA ;SCOPE ENTRY POINT *
:*****
:TEST FOR CORRECT OPERATION OF CHARACTER LENGTH SELECTION (RXCSR BITS 9 AND 10)
:BY CHECKING THAT TIME REQUIRED FOR OUTPUTTING A CHARACTER IS LONGEST FOR
:8 BIT CODE THAN FOR 7 BIT CODE ETC.
AKAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD.
      CLR CTRB ;(ELAPSED TIME COUNTERS).
      CLR CTRC
      CLR CTRD
      BIC #3000,@RXCSR ;SET CHAR LENGTH TO 00 (8 BIT CODE).
      JSR %7,AKAS ;OUTPUT CHARACTER AND TIME.
      ADD AKAST,CTRA ;ADD ELAPSED TIME TO CTRA
      BIC #3000,@RXCSR ;SET CHAR LENGTH TO 01 (7 BIT CODE).
      BIS #1000,@RXCSR
      JSR %7,AKAS ;OUTPUT CHARACTER AND TIME.
      ADD AKAST,CTRB ;ADD ELAPSED TIME TO CTRB.
      BIC #3000,@RXCSR ;SET CHAR LENGTH TO 10 (6 BIT CODE)
      BIS #2000,@RXCSR
      JSR %7,AKAS ;OUTPUT CHARACTER AND TIME.
      ADD AKAST,CTRC ;ADD ELAPSED TIME TO CTRC
      BIS #3000,@RXCSR ;SET CHAR LENGTH TO 11 (5 BIT CODE)
      JSR %7,AKAS ;OUTPUT CHARACTER AND TIME
      ADD AKAST,CTRD ;ADD ELAPSED TIME TO CTRD
      JSR %7,CMPT ;CHECK THAT CTRA THROUGH CTRD
      BR AKAB ;DESCENDING VALUES.
      ERROR1 ;TX CHARACTER LENGTH NOT ARRANGED
      ETCLGT ;IN DESCENDING ORDER.
      AKAB: SCOPE ;SCOPE
      AKAS: CLR AKAST ;CLEAR ELAPSED TIME COUNTER AKAST
           TSTB @TXCSR ;WAIT FOR TX READY.
           BPL -4
           DELAY 20 ;WAIT 20 MSECS.
      AKASA: CLR @TXBUF ;LOAD TXBUF
            DELAY 1 MSEC ;DELAY 1 MSEC
           INC AKAST ;INCREMENT ELAPSED TIME COUNTER
           TSTB @TXCSR ;READY SET?
           BPL AKASA ;BRANCH IF READY NOT SET
           RTS %7 ;EXIT
      AKAST: OPEN ;ELAPSED TIME COUNTER
```

```
2173
2174
2175 010622 000055
2176 010624 010676
2177 010626 000144
2178 010630 010632
2179
2180
2181
2182
2183 010632 052777 000004 170244
2184 010640 005077 170242
2185 010644 104016
2186 010646 000310
2187 010650 105777 170224
2188 010654 100402
2189 010656 104003
2190 010660 000405
2191 010662 104011
2192 010664 105777 170210
2193 010670 100001
2194 010672 104003
2195 010674 104012
2196
2197 010676 000056
2198 010700 010744
2199 010702 000144
2200 010704 010706
2201
2202
2203
2204 010706 052777 000004 170170
2205 010714 005077 170166
2206 010720 105777 170154
2207 010724 100375
2208 010726 005777 170150
2209 010732 105777 170142
2210 010736 100001
2211 010740 104003
2212 010742 104012

;*****
AT55: 55 ;TEST NUMBER 55 *
      AT56 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ALAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT OUTPUTTING A CHARACTER WITH THE MAINTENANCE BIT SET (TXCSR BIT 2)
;RESULTS IN DONE BIT SETTING (RXCSR BIT 7) NO LATER THAN 200 MSECS, AND
;THAT RESET INSTRUCTION CLEARS THE DONE BIT
ALAA: BIS #BIT2,@TXCSR ;SET MAINTENANCE (TXCSR BIT 2)
      CLR @TXBUF ;LOAD TXBUF
      DELAY 200. ;WAIT 200 MSECS.
      TSTB @RXCSR ;SEE IF DONE BIT IS SET
      BMI ALAB ;BRANCH IF DONE BIT IS SET
      ERROR ;DONE BIT FAILED TO SET
      BR ALAC
ALAB: SRESET ;ISSUE RESET TO CLEAR DONE BIT
      TSTB @RXCSR ;SEE IF DONE BIT IS CLEARED
      BPL ALAC ;BRANCH IF DONE BIT IS CLEARED
      ERROR ;RESET FAILED TO CLEAR DONE BIT
ALAC: SCOPE ;SCOPE
;*****
AT56: 56 ;TEST NUMBER 56 *
      AT57 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AMAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT DONE BIT (RXCSR BIT 7) IS CLEARED BY READING RXBUF.
;DONE SET BY OUTPUTTING CHARACTER WITH MAINTENANCE BIT SET (TXCSR BIT 2)
AMAA: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT (TXCSR BIT 2)
      CLR @TXBUF ;LOAD TXBUF
AMAB: TSTB @RXCSR ;WAIT FOR DONE BIT TO SET.
      BPL AMAB
      TST @RXBUF ;READ RXBUF TO CLEAR DONE BIT
      TSTB @RXCSR ;SEE IF DONE BIT IS CLEAR
      BPL AMAC ;BRANCH IF DONE BIT IS CLEAR
      ERROR ;READING RXBUF FAILED TO CLEAR DONE BIT
AMAC: SCOPE ;SCOPE
```

```
2213
2214
2215 010744 000057
2216 010746 011032
2217 010750 000012
2218 010752 010754
2219
2220
2221
2222
2223 010754 042777 000030 170122
2224 010762 052777 000014 170114
2225 010770 042777 000030 170102
2226 010776 052777 000010 170074
2227 011004 005077 170076
2228 011010 104016
2229 011012 000764
2230 011014 105777 170160
2231 011020 100401
2232 011022 104003
2233 011024 005777 170052
2234 011030 104012
2235
2236
2237 011032 000060
2238 011034 011120
2239 011036 000012
2240 011040 011042
2241
2242
2243
2244
2245 011042 042777 000030 170034
2246 011050 052777 000024 170026
2247 011056 042777 000030 170014
2248 011064 052777 000020 170006
2249 011072 005077 170010
2250 011076 104016
2251 011100 000620
2252 011102 105777 167772
2253 011106 100401
2254 011110 104003
2255 011112 005777 167764
2256 011116 104012
2257
2258 011120 000061
2259 011122 011172
2260 011124 000012
2261 011126 011130
2262
2263
2264
2265
2266 011130 052777 000034 167746
2267 011136 052777 000030 167734
2268 011144 005077 167736
```

```
*****
AT57: 57 ;TEST NUMBER 57 *
      AT60 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      ANAA ;SCOPE ENTRY POINT *
*****
;TEST THAT DONE BIT (RXCSR BIT 7) SETS NO LATER THAN 500 MSECS AFTER OUTPUTTING
;CHARACTER WITH MAINTENANCE BIT SET AND RECEIVE SPEED SET TO 01 (TRANSMIT
;SPEED ALSO SET TO 01
ANAA: BIC #30,@TXCSR ;SET MAINTENANCE BIT AND SET
      BIS #14,@TXCSR ;TX SPEED=01
      BIC #30,@RXCSR ;SET RX SPEED =01
      BIS #10,@RXCSR
      CLR @TXBUF ;LOAD TXBUF
      DELAY 500. ;DELAY 500 MSECS.
      TSTB @RXCSR ;SEE IF DONE BIT IS SET.
      BMI ANAB ;BRANCH IF DONE IS SET.
      ERROR ;DONE FAILED TO SET WITH RX SPEED-01.
ANAB: TST @RXBUF ;CLEAR DONE BIT IF SET.
      SCOPE ;SCOPE
*****
AT60: 60 ;TEST NUMBER 60 *
      AT61 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AOAA ;SCOPE ENTRY POINT *
*****
;TEST THAT DONE BIT (RXCSR BIT 7) SETS NO LATER THAN 400 MSECS AFTER OUTPUTTING
;CHARACTER WITH MAINTENANCE BIT SET AND RECEIVE SPEED SET TO 10 (TRANSMIT
;SPEED ALSO SET TO 10).
AOAA: BIC #30,@TXCSR ;SET MAINTENANCE BIT AND SET
      BIS #24,@TXCSR ;TX SPEED=10.
      BIC #30,@RXCSR ;SET RX SPEED-10.
      BIS #20,@RXCSR
      CLR @TXBUF ;LOAD TXBUF
      DELAY 400. ;DELAY 400 MSECS
      TSTB @RXCSR ;SEE IF DONE BIT IS SET.
      BMI AOAB ;BRANCH IF DONE BIT IS SET.
      ERROR ;DONE FAILED TO SET WITH RX SPEED-10.
AOAB: TST @RXBUF ;CLEAR DONE BIT IF SET
      SCOPE ;SCOPE
*****
AT61: 61 ;TEST NUMBER 61 *
      AT62 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      APAA ;SCOPE ENTRY POINT *
*****
;TEST THAT DONE BIT (RXCSR BIT 7) SETS NO LATER THAN 250 MSECS AFTER OUTPUTTING
;CHARACTER WITH MAINTENANCE BIT SET AND RECEIVE SPEED SET 11 (TRANSMIT SPEED
;ALSO SET TO 11).
APAA: BIS #34,@TXCSR ;SET MAINT BIT AND TX SPEED-11
      BIS #30,@RXCSR ;SET RX SPEED-11
      CLR @TXBUF ;LOAD TXBUF
```

```
2269 011150 104016          DELAY          ;DELAY 250 MSECS.
2270 011152 000372          250.
2271 011154 105777 167720  TSTB @RXCSR   ;SEE IF DONE BIT IS SET.
2272 011160 100401          BMI APAB      ;BRANCH IF DONE BIT IS SET.
2273 011162 104003          ERROR        ;DONE FAILED TO SET WITH RX SPEED-11
2274 011164 005777 167712  APAB: TST @RXBUF ;CLEAR DONE BIT IF SET.
2275 011170 104012          SCOPE       ;SCOPE
2276
2277
2278
2279 011172 000062          AT62: 62      ;TEST NUMBER 62 *
2280 011174 011474          AT63        ;ADDRESS OF NEXT TEST *
2281 011176 000144          100.       ;TEST ITERATION COUNT *
2282 011200 011202          AQAA       ;SCOPE ENTRY POINT *
2283
2284
2285
2286
2287 011202 005037 001246  AQAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD
2288 011206 005037 001250  CLR CTRB ;(ELAPSED TIME COUNTERS)
2289 011212 005037 001252  CLR CTRC
2290 011216 005037 001254  CLR CTRD
2291 011222 042777 000030 167654  BIC #30,@TXCSR ;SELECT TX SPEED 00
2292 011230 052777 000004 167646  BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
2293 011236 042777 000030 167634  BIC #30,@RXCSR ;SELECT RX SPEED 00
2294 011244 004737 011424  JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
2295 011250 063737 011472 001246  ADD AQAST,CTRA ;ADD ELAPSED TIME TO CTRA
2296 011256 042777 000030 167620  BIC #30,@TXCSR ;SELECT TX SPEED 01
2297 011264 052777 000010 167612  BIS #10,@TXCSR
2298 011272 042777 000030 167600  BIC #30,@RXCSR ;SELECT RX SPEED 01.
2299 011300 052777 000010 167572  BIS #10,@RXCSR
2300 011306 004737 011424  JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
2301 011312 063737 011472 001250  ADD AQAST,CTRB ;ADD ELAPSED TIME TO CTRB
2302 011320 042777 000030 167556  BIC #30,@TXCSR ;SELECT TX SPEED 10
2303 011326 052777 000020 167550  BIS #20,@TXCSR
2304 011334 042777 000030 167536  BIC #30,@RXCSR ;SELECT RX SPEED 10
2305 011342 052777 000020 167530  BIS #20,@RXCSR
2306 011350 004737 011424  JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT.
2307 011354 063737 011472 001252  ADD AQAST,CTRC ;ADD ELAPSED TIME TO CTRC.
2308 011362 052777 000030 167514  BIS #30,@TXCSR ;SELECT TX SPEED 11
2309 011370 052777 000030 167502  BIS #30,@RXCSR ;SELECT RX SPEED 11
2310 011376 004737 011424  JSR %7,AQAS ;OUTPUT CHARACTER AND TIME DONE BIT
2311 011402 063737 011472 001254  ADD AQAST,CTRD ;ADD ELAPSED TIME TO CTRD.
2312 011410 004737 014334  JSR %7,CMPT ;CHECK THAT CTRA THROUGH CTRD CONTAIN
2313 011414 000402          BR AQAB      ;DESCENDING VALUES.
2314 011416 104015          ERROR1     ;RECEIVE SPEEDS NOT ARRANGED IN
2315 011420 015730          ERXTIM    ;ASCENDING ORDER.
2316 011422 104012          AQAB: SCOPE ;SCOPE
2317 011424 005037 011472  AQAS: CLR AQAST ;CLEAR ELAPSED TIME COUNTER AQAST
2318 011430 105777 167450  TSTB @TXCSR ;WAIT FOR TX READY.
2319 011434 100375          BPL -.4
2320 011436 104016          DELAY     ;WAIT 20 MSECS.
2321 011440 000024          20.
2322 011442 005777 167434  TST @RXBUF   ;CLEAR DONE BIT IF SET
2323 011446 005077 167434  CLR @TXBUF   ;LOAD TXBUF
2324 011452 104016          AQASA: DELAY ;DELAY 1 MSEC
```



2325	011454	000001		1			
2326	011456	005237	011472	INC	AQAST	:	INCREMENT ELAPSED TIME COUNTER
2327	011462	105777	167412	TSTB	@RXCSR	:	DONE SET?
2328	011466	100371		BPL	AQASA	:	BRANCH IF DONE NOT SET
2329	011470	000207		RTS	%7	:	EXIT
2330	011472	000000		AQAST:	OPEN	:	ELAPSED TIME COUNTER

```
2331
2332 :*****
2333 011474 000063 AT63: 63 ;TEST NUMBER 63 *
2334 011476 011732 ;ADDRESS OF NEXT TEST *
2335 011500 000144 100. ;TEST ITERATION COUNT *
2336 011502 011504 ARAA ;SCOPE ENTRY POINT *
2337 :*****
2338 ;TEST FOR CORRECT OPERATION OF CHARACTER LENGTH SELECTION DURING RECEIVE
2339 ;(RXCSR BITS 9 AND 10) BY CHECKING THAT TIME REQUIRED TO RECEIVE A CHARACTER
2340 ;IS LONGEST FOR 8 BIT CODE THAN FOR 7 BIT CODE ETC.
2341 011504 005037 001246 ARAA: CLR CTRA ;CLEAR CTRA THROUGH CTRD
2342 011510 005037 001250 CLR CTRB ;(ELAPSED TIME COUNTERS)
2343 011514 005037 001252 CLR CTRC
2344 011520 005037 001254 CLR CTRD
2345 011524 042777 003000 167346 BIC #3000,@RXCSR ;SET CHAR LENGTH TO 00 (8 BIT CODE)
2346 011532 004737 011654 JSR %7,ARAS ;OUTPUT CHAR AND TIME DONE BIT.
2347 011536 063737 011730 001246 ADD ARAST,CTRA ;ADD ELAPSED TIME TO CTRA
2348 011544 042777 003000 167326 BIC #3000,@RXCSR ;SET CHAR LENGTH TO 01 (7 BIT CODE)
2349 011552 052777 001000 167320 BIS #1000,@RXCSR
2350 011560 004737 011654 JSR %7,ARAS ;OUTPUT CHAR AND TIME DONE BIT
2351 011564 063737 011730 001250 ADD ARAST,CTRB ;ADD ELAPSED TIME TO CTRB
2352 011572 042777 003000 167300 BIC #3000,@RXCSR ;SET CHAR LENGTH TO 10 (6 BIT CODE)
2353 011600 052777 002000 167272 BIS #2000,@RXCSR
2354 011606 004737 011654 JSR %7,ARAS ;OUTPUT CHAR AND TIME DONE BIT
2355 011612 063737 011730 001252 ADD ARAST,CTRC ;ADD ELAPSED TIME TO CTRC
2356 011620 052777 003000 167252 BIS #3000,@RXCSR ;SET CHAR LENGTH TO 11 (5 BIT CODE)
2357 011626 004737 011654 JSR %7,ARAS ;OUTPUT CHAR AND TIME DONE BIT
2358 011632 063737 011730 001254 ADD ARAST, CTRD ;ADD ELAPSED TIME TO CTRD
2359 011640 004737 014334 JSR %7,CMPT ;CHECK THAT CTRA THROUGH CTRD
2360 011644 000402 BR ARAB ;CONTAIN DESCENDING VALUES
2361 011646 104015 ERROR1 ;RECEIVE CHARACTER LENGTHS NOT ARRANGED
2362 011650 015772 ERCLGT ;IN DESCENDING ORDER
2363 011652 104012 ARAA: SCOPE ;SCOPE
2364 011654 005037 011730 ARAA: CLR ARAST ;CLEAR ELAPSED TIME COUNTER ARAST
2365 011660 105777 167220 ARAA: TSTB @TXCSR ;WAIT FOR TX READY.
2366 011664 100375 BPL .-4
2367 011666 104016 DELAY ;WAIT 20 MSECS.
2368 011670 000024 20.
2369 011672 005777 167204 TST @RXBUF ;CLEAR DONE BIT IF SET
2370 011676 052777 000004 167200 BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
2371 011704 005077 167176 CLR @TXBUF ;LOAD TXBUF
2372
2373 011710 104016 ARAA: DELAY ;DELAY 1 MSEC.
2374 011712 000001 1
2375 011714 005237 011730 INC ARAST ;INCREMENT ELAPSED TIME COUNTER
2376 011720 105777 167154 TSTB @RXCSR ;SEE IF DONE BIT IS SET.
2377 011724 100371 BPL ARAA ;BRANCH IF NOT SET
2378 011726 000207 RTS %7 ;EXIT
2379 011730 000000 ARAA: OPEN ;ELAPSED TIME COUNTER
```

```
2380
2381
2382 011732 000064      :*****
2383 011734 012046      AT64: 64          ;TEST NUMBER 64          *
2384 011736 000144      AT65          ;ADDRESS OF NEXT TEST   *
2385 011740 011742      100.         ;TEST ITERATION COUNT  *
2386                                     ASAA          ;SCOPE ENTRY POINT     *
2387 :*****
2388 011742 004737 012026 :TEST CORRECT OPERATION OF DATA OVERRUN BIT (RXCSR BIT 12)
2389 011746 004737 012026 ASAA: JSR %7,ASAS ;OUTPUT CHARACTER AND WAIT 200 MSECS
2390 011752 017737 167122 001260 JSR %7,ASAS ;OUTPUT CHARACTER AND WAIT 200 MSECS
2391 011760 032737 010000 001260 MOV @RXCSR,RXCRT ;SAVE RXCSR CONTENTS
2392 011766 001002 BIT #BIT12,RXCRT ;SEE IF DATA OVERRUN BIT WAS SET
2393 011770 104003 BNE ASAB ;BRANCH IF BIT WAS SET
2394 011772 000412 BR ASAD
2395 011774 005737 001260 ASAB: TST RXCSRT ;SEE IF ERROR BIT WAS SET (RXCSR BIT 15)
2396 012000 100402 BMI ASAC
2397 012002 104003 ERROR ;ERROR BIT FAILED TO SET
2398                                     ;WHEN DATA OVERRUN SET
2399 012004 000405 BR ASAD
2400 012006 032777 010000 167064 ASAC: BIT #BIT12,@RXCSR ;SEE IF DATA OVERRUN WAS
2401                                     ;CLEARED WHEN RXCSR WAS READ
2402 012014 001401 BEQ ASAD ;BRANCH IF CLEAR
2403 012016 104003 ERROR ;READING RXCSR FAILED
2404                                     ;TO CLEAR DATA OVERRUN
2405 012020 005777 167056 ASAD: TST @RXBUF ;CLEAR DONE BIT (RXCSR BIT 7)
2406 012024 104012 SCOPE ;SCOPE
2407 012026 052777 000004 167050 ASAS: BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
2408 012034 005077 167046 CLR @TXBUF ;LOAD TXBUF
2409 012040 104016 DELAY ;DELAY 200 MSECS
2410 012042 000310 200.
2411 012044 000207 RTS %7 ;EXIT
2412
2413 :*****
2414 012046 000065      AT65: 65          ;TEST NUMBER 65          *
2415 012050 012120      AT66          ;ADDRESS OF NEXT TEST   *
2416 012052 000012      10.         ;TEST ITERATION COUNT  *
2417 012054 012066      ATAA          ;SCOPE ENTRY POINT     *
2418 :*****
2419 :TEST THAT TRANSMITTER IS ABLE TO INTERRUPT. IF THE INTERRUPT IS SERVICED,
2420 :IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2421 012056 004737 002632 JSR 7,OVRLAY ;GO TO OVER LAY ROUTINE
2422 012062 104007 STTXV ;SET TX INTERRUPT SERVICE
2423 012064 012114 ATAC ;TO ATAC
2424 012066 042777 000100 167010 ATAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPT
2425 012074 005037 177776 CLR PSW ;SET PROCESSOR PRIORITY TO 0
2426 012100 052777 000104 166776 BIS #104,@TXCSR ;ENABLE TX INTERRUPT
2427 012106 000240 NOP
2428 012110 104003 ERROR ;READY DID NOT CAUSE AN INTERRUPT
2429 012112 104012 ATAB: SCOPE ;SCOPE
2430 012114 022626 ATAC: POPSP2 ;HERE IF INTERRUPT IS SERVICED. POP
2431 012116 000775 BR ATAB ;THE STOCK TWICE
2432
```

```
2433
2434
2435 012120 000066
2436 012122 012176
2437 012124 001750
2438 012126 012134
2439
2440
2441
2442 012130 104007
2443 012132 012166
2444 012134 013737 001116 177776
2445 012142 042777 000100 166734
2446 012150 052777 000104 166726
2447 012156 000240
2448
2449
2450 012160 042777 000100 166716
2451 012166 104012
2452 012170 022626
2453 012172 104003
2454 012174 000774
2455
2456
2457 012176 000067
2458 012200 012262
2459 012202 000012
2460 012204 012212
2461
2462
2463
2464 012206 104007
2465 012210 012250
2466 012212 042777 000100 166664
2467 012220 013737 001116 177776
2468 012226 162737 000040 177776
2469 012234 052777 000104 166642
2470 012242 000240
2471 012244 104003
2472 012246 000401
2473 012250 022626
2474 012252 042777 000100 166624
2475 012260 104012
2476

:*****
AT66: 66 ;TEST NUMBER 66 *
      AT67 ;ADDRESS OF NEXT TEST *
      1000. ;TEST ITERATION COUNT *
      AUAA ;SCOPE ENTRY POINT *
:*****
:TEST THAT READY DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS
:AT THE SAME PRIORITY AS THE TRANSMITTER INTERRUPT REQUEST LEVEL
      STTXV ;SET TX INTERRUPT SERVICE TO
      AUAC
AUAA: MOV TXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS TX PRIORITY
      BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
      BIS #104,@TXCSR ;ENABLE TX INTERRUPTS
      NOP
AUAB: BIC #BIT6,@TXCSR ;OK IF NO INTERRUPT OCCURS. DISABLE INTERRUPTS
AUAC: SCOPE ;SCOPE
      POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
      ERROR ;TX INTERRUPTED WITH PROCESSOR AT SAME
      BR AUAC ;PRIORITY AS THE TRANSMITTER
:*****
AT67: 67 ;TEST NUMBER 67 *
      AT70 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AVAA ;SCOPE ENTRY POINT *
:*****
:TEST THAT TRANSMITTER INTERRUPTS WHEN PROCESSOR IS AT PRIORITY ONE LEVEL
:LOWER THAN THE TRANSMITTER INTERRUPT PRIORITY.
      STTXV ;SET TX INTERRUPT SERVICE TO AVAB
      AVAB
AVAA: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
      MOV TXLVL,PSW ;SET PROCESSOR PRIORITY TO ONE LEVEL
      SUB #40,PSW ;LOWER THAN TX PRIORITY
      BIS #104,@TXCSR ;ENABLE TX INTERRUPTS
      NOP
      ERROR ;TX FAILED TO INTERRUPT
      BR AVAC
AVAB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
AVAC: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
      SCOPE ;SCOPE
```

```
2477
2478
2479 012262 000070
2480 012264 012360
2481 012266 000144
2482 012270 012272
2483
2484
2485
2486 012272 104007
2487 012274 012332
2488 012276 042777 000100 166600
2489 012304 005037 177776
2490 012310 052777 000104 166566
2491 012316 000240
2492 012320 104003
2493 012322 042777 000100 166554
2494 012330 104012
2495 012332 012777 012352 166554
2496 012340 012716 012346
2497 012344 000002
2498 012346 000240
2499 012350 000764
2500 012352 022626
2501 012354 104003
2502 012356 000761
2503
2504
2505 012360 000071
2506 012362 012436
2507 012364 000012
2508 012366 012404
2509
2510
2511
2512 012370 004737 002632
2513 012374 104006
2514 012376 012432
2515 012400 004737 014314
2516 012404 042777 000100 166466
2517 012412 005037 177776
2518 012416 052777 000100 166454
2519 012424 000240
2520 012426 104003
2521 012430 000401
2522 012432 022626
2523 012434 104012
2524

;*****
AT70: 70 ;TEST NUMBER 70 *
      AT71 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AWAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT TRANSMITTER DOES NOT REINTERRUPT AFTER THE INITIAL INTERRUPT HAS
;OCCURRED AND HAS BEEN SERVICED.
AWAA: STTXV ;SET TX INTERRUPT SERVICE TO AWAC
      AWAC
      BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
      CLR PSW ;SET PROCESSOR PRIORITY TO 0
      BIS #104,@TXCSR ;ENABLE TX INTERRUPTS
      NOP
      ERROR ;TRANSMITTER FAILED TO INTERRUPT
AWAB: BIC #BIT6,@TXCSR ;DISABLE TX INTERRUPTS
      SCOPE ;SCOPE
AWAC: MOV #AWAE,@TXVTR ;HERE IF INTERRUPT OCCURS. CHANGE EXIT
      MOV #AWAD,@%6 ;POINTER TO AWAD AND EXIT INTERRUPT
      RTI
AWAD: NOP ;OK IF NO INTERRUPT REOCCURS.
      BR AWAB
AWAE: POPSP2 ;HERE IF INTERRUPT REOCCURS
      ERROR ;TX REINTERRUPTED AFTER RTI
      BR AWAB
;*****
AT71: 71 ;TEST NUMBER 71 *
      AT72 ;ADDRESS OF NEXT TEST *
      10. ;TEST ITERATION COUNT *
      AXAA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RECEIVER DONE BIT IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
;SERVICED IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
JSR 7,OVRLAY ;GO TO OVERLAY ROUTINE
STRXV ;SET RX INTERRUPT SERVICE TO AXAB
AXAB
AXAA: JSR %7,STRXD ;SET RX DONE BIT
      BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
      CLR PSW ;SET PROCESSOR PRIORITY TO 0
      BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
      NOP
      ERROR ;RX FAILED TO INTERRUPT
AXAB: BR AXAC
AXAC: POPSP2 ;HERE IF INTERRUPT OCCURS
      SCOPE ;SCOPE
```

```

2525
2526 ;*****
2527 012436 000072 AT72: 72 ;TEST NUMBER 72 *
2528 012440 012520 ;ADDRESS OF NEXT TEST *
2529 012442 001750 1000. ;TEST ITERATION COUNT *
2530 012444 012456 AYAA ;SCOPE ENTRY POINT *
2531 ;*****
2532 ;TEST THAT RECEIVER DONE BIT DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR
2533 ;IS AT THE SAME PRIORITY LEVEL AS THE RECEIVER INTERRUPT REQUEST LEVEL
2534 012446 104006 STRXV ;SET RX INTERRUPT SERVICE TO AYAC
2535 012450 012512 AYAC
2536 012452 004737 014314 JSR %7,STRXD ;SET RX DONE BIT
2537 012456 042777 000100 166414 AYAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2538 012464 013737 001112 177776 MOV RXLVL,PSW ;SET PROCESSOR PRIORITY SAME AS RECEIVER'S
2539 012472 052777 000100 166400 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2540 012500 000240 NOP
2541 012502 042777 000100 166370 AYAB: BIC #BIT6,@RXCSR ;OK IF NO INTERRUPT. DISABLE RX INTERRUPTS
2542 012510 104012 SCOPE ;SCOPE
2543 012512 022626 AYAC: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
2544 012514 104003 ERROR ;RX INTERRUPTED WITH PROCESOR AT SAME
2545 012516 000771 BR AYAB ;PRIORITY AS THE RECEIVER
2546
2547 ;*****
2548 012520 000073 AT73: 73 ;TEST NUMBER 73 *
2549 012522 012610 AT74 ;ADDRESS OF NEXT TEST *
2550 012524 000012 10. ;TEST ITERATION COUNT *
2551 012526 012540 AZAA ;SCOPE ENTRY POINT *
2552 ;*****
2553 ;TEST THAT RECEIVER DONE BIT CAUSES INTERRUPT WHEN PROCESSOR IS AT PRIORITY
2554 ;ONE LEVEL LOWER THAN THE RECEIVER'S INTERRUPT REQUEST LEVEL
2555 012530 104006 STRXV ;SET RX INTERRUPT TO AZAB
2556 012532 012576 AZAB
2557 012534 004737 014314 JSR %7,STRXD ;SET RX DONE BIT
2558 012540 042777 000100 166332 AZAA: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2559 012546 013737 001112 177776 MOV RXLVL,PSW ;SET PROCESSOR PRIORITY ONE LEVEL
2560 012554 162737 000040 177776 SUB #40,PSW ;LOWER THAN RECEIVER'S PRIORITY
2561 012562 052777 000100 166310 BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
2562 012570 000240 NOP
2563 012572 104003 ERROR ;RX FAILED TO INTERRUPT WITH PROCESSOR AT
2564 012574 000401 BR AZAC ;PRIORITY ONE LEVEL LOWER THAN RECEIVER'S
2565
2566 012576 022626 AZAB: POPSP2 ;HERE IF INTERRUPT OCCURS
2567 012600 042777 000100 166272 AZAC: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
2568 012606 104012 SCOPE ;SCOPE

```

```
2569
2570
2571 012610 000074
2572 012612 012706
2573 012614 000144
2574 012616 012624
2575
2576
2577
2578 012620 004737 014314
2579 012624 104006
2580 012626 012660
2581 012630 042777 000100 166242
2582 012636 052777 000100 166234
2583 012644 000240
2584 012646 104003
2585 012650 042777 000100 166222
2586 012656 104012
2587 012660 012777 012700 166222
2588 012666 012716 012674
2589 012672 000002
2590 012674 000240
2591 012676 000764
2592 012700 022626
2593 012702 104003
2594 012704 000761
2595
2596
2597 012706 000075
2598 012710 012762
2599 012712 000144
2600 012714 012716
2601
2602
2603 012716 004737 014314
2604 012722 005077 166160
2605 012726 104016
2606 012730 000024
2607 012732 017737 166142 001260
2608 012740 105777 166134
2609 012744 100001
2610 012746 104003
2611 012750 104016
2612 012752 000310
2613 012754 005777 166122
2614 012760 104012
2615

;*****
AT74: 74 ;TEST NUMBER 74 *
      AT75 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      AABA ;SCOPE ENTRY POINT *
;*****
;TEST THAT RECEIVER DOES NOT INTERRUPT AFTER THE INITIAL INTERRUPT HAS
;OCCURED AND DONE BIT HAS NOT BEEN CLEARED
AABA: JSR %7,STRXD ;SET RX DONE BIT
      STRXV ;SET RX INTERRUPT SERVICE TO AABC
      AABC
      BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
      BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
      NOP
      ERROR ;RX FAILED TO INTERRUPT
AABB: BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
      SCOPE ;SCOPE
AABC: MOV #AABE,@RXVTR ;HERE IF INTERRUPT OCCURS. CHANGE SERVICE TO
      MOV #AABD,@%6 ;AABE, SET EXIT POINTER TO AABD
      RTI ;EXIT INTERRUPT SERVICE
AABD: NOP ;OK IF NO INTERRUPT REOCCURS
      BR AABB
AABE: POPSP2 ;HERE IF INTERRUPT REOCCURS
      ERROR ;RX REINTERRUPTED AFTER RTI
      BR AABB
;*****
AT75: 75 ;TEST NUMBER 75 *
      AT76 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ABBA ;SCOPE ENTRY POINT *
;*****
;TEST THAT DATA OVERRUN (RXCSR BIT 12) CLEARS THE DONE BIT (RXCSR BIT 7)
ABBA: JSR %7,STRXD ;SET RX DONE BIT
      CLR @TXBUF ;LOAD TXBUF
      DELAY ;WAIT 20 MSECS.
      20.
      MOV @RXCSR,RXCSRT ;SAVE CONTENT OF RXCSR
      TSTB @RXCSR ;SEE IF DONE BIT IS CLEAR
      BPL ABBB ;BRANCH IF DONE BIT IS CLEAR
      ERROR
ABBB: DELAY ;WAIT FOR RX DONE TO SET.
      200.
      TST @RXBUF ;CLEAR DONE BIT IF SET
      SCOPE ;SCOPE
```

```
2616
2617
2618 012762 000076
2619 012764 013040
2620 012766 000144
2621 012770 012776
2622
2623
2624 012772 104006
2625 012774 013034
2626 012776 004737 014314
2627 013002 004737 014314
2628 013006 042777 000100 166064
2629 013014 005037 177776
2630 013020 052777 000100 166052
2631 013026 000240
2632 013030 104003
2633 013032 000401
2634 013034 022626
2635 013036 104012
2636
2637
2638
2639
2640 013040 000077
2641 013042 013222
2642 013044 000144
2643 013046 013062
2644
2645
2646 013050 004537 003126
2647 013054 052777 000004 166022
2648 013062 112737 000144 001246
2649 013070 112737 000010 001247
2650 013076 004537 003232
2651 013102 105777 165776
2652 013106 100375
2653 013110 010177 165772
2654 013114 105777 165760
2655 013120 100375
2656 013122 017737 165754 001232
2657 013130 005000
2658 013132 006037 001232
2659 013136 103001
2660 013140 005100
2661 013142 105337 001247
2662 013146 001371
2663 013150 032777 000040 165722
2664 013156 001403
2665 013160 005700
2666 013162 001403
2667 013164 000412
2668 013166 005700
2669 013170 001410
2670 013172 104003
2671 013174 004537 003300

:*****
AT76: 76 ;TEST NUMBER 76 *
      AT77 ;ADDRESS OF NEXT TEST *
      100. ;TEST ITERATION COUNT *
      ACBA ;SCOPE ENTRY POINT *
:*****
:TEST THAT ERROR BIT (RXCSR BIT 15) IS ABLE TO CAUSE AN INTERRUPT
STRXV ;SET RX INTERRUPT SERVICE TO ACBB.
ACBB: ACBB
ACBA: JSR %7,STRXD ;SET RX DONE BIT
      JSR %7,STRXD ;SET RX DATA OFLOW
      BIC #BIT6,@RXCSR ;DISABLE RX INTERRUPTS
      CLR PSW ;SET PROCESSOR PRIORITY TO 0
      BIS #BIT6,@RXCSR ;ENABLE RX INTERRUPTS
      NOP
      ERROR ;RX ERROR BIT FAILED TO CAUSE INTERRUPT
      BR ACBC
ACBB: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STOCK TWICE
ACBC: SCOPE ;SCOPE

:*****
AT77: 77 ;TEST NUMBER 77
      AT100 ;ADDRESS OF NEXT TEST
      100. ;TEST ITERATION COUNT
      ANBB ;SCOPE ENTRY POINT
:*****
:TEST THAT PARITY INDICATOR OPERATES CORRECT.
ANBA: JSR 5,INBIN ;INITIALIZE BINARY COUNT PATTERN
      BIS #BIT2,@TXCSR ;SET MAINTENANCE BIT
ANBB: MOVB #100.,CTRA ;GET CHARACTER COUNT
ANBC: MOVB #8.,CTRA+1 ;GET CHARACTER BIT COUNT
      JSR 5,GTBINP ;GET A CHARACTER (IN R1)
      TSTB @TXCSR ;WAIT FOR
      BPL -4 ;TRANSMITTER READY FLAG
      MOV %1,@TXBUF ;LOAD TRANSMITTER BUFFER
      TSTB @RXCSR ;WAIT FOR
      BPL -4 ;RECEIVER READY FLAG
      MOV @RXBUF,CRBUFA ;GET RECEIVED CHARACTER
      CLR %0 ;CLEAR WORKING REGISTER
ANBD: ROR CRBUFA ;LOOK AT CHARACTER BITS
      BCC .+4 ;AND COMPLEMENT R0 WHEN
      COM %0 ;A 1 IS RECEIVED
      DECB CTRA+1 ;IF R0=1'S, ODD#1'S RECEIVED
      BNE ANBD ;IF R0=0'S, EVEN #1'S RECEIVED
      BIT #BIT5,@RXCSR ;TEST PARITY INDICATOR
      BEQ ANBE ;BRANCH IF INDICATES EVEN
      TST %0 ;TEST RECEIVED PARITY (IN R0)
      BEQ ANBF ;ERROR BRANCH
      BR ANBG ;OK BRANCH
ANBE: TST %0 ;TEST RECEIVED PARITY (IN R0)
      BEQ ANBG ;OK BRANCH
ANBF: ERROR ;TYPE PC
      JSR 5,OACNV ;GO TO OCTAL
```



```
2672 013200 001232          CRBUFA          ;TO ASCII
2673 013202 016424          AWAS          ;ROUTINE AND
2674 013204 000003          3            ;CONVERT DATA
2675 013206 104015          ERROR1       ;TYPE
2676 013210 016424          AWAS          ;DATA
2677 013212 105337 001246  ANBG:  DECB      CTRA      ;DECREMENT CHARACTER COUNT
2678 013216 001324          BNE          ANBC
2679 013220 104012          SCOPE
2680 000077          X=77
2681 000000          Y=0
2682
2683 013222 000100  AT100: 100          ;ROUTINE #100 *
2684 013224 013240          AT101       ;ADDRESS OF NEXT TEST *
2685 013226 000003          3.         ;ITERATION COUNT *
2686 013230 013232          DAT0        ;SCOPE ENTRY POINT *
2687 000100          X=X+1
2688
2689 013232 004537 003532  DAT0:  JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2690 013236 000000          0          ;SEE NOTE 0 FOR DATA TEST PARAMETERS
2691 000001          Y=Y+1
2692
2693 013240 000101  AT101: 101          ;ROUTINE #101 *
2694 013242 013256          AT102       ;ADDRESS OF NEXT TEST *
2695 013244 000003          3.         ;ITERATION COUNT *
2696 013246 013250          DAT1        ;SCOPE ENTRY POINT *
2697 000101          X=X+1
2698
2699 013250 004537 003532  DAT1:  JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2700 013254 000001          1          ;SEE NOTE 1 FOR DATA TEST PARAMETERS
2701 000002          Y=Y+1
2702
2703 013256 000102  AT102: 102          ;ROUTINE #102 *
2704 013260 013274          AT103       ;ADDRESS OF NEXT TEST *
2705 013262 000003          3.         ;ITERATION COUNT *
2706 013264 013266          DAT2        ;SCOPE ENTRY POINT *
2707 000102          X=X+1
2708
2709 013266 004537 003532  DAT2:  JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2710 013272 000002          2          ;SEE NOTE 2 FOR DATA TEST PARAMETERS
2711 000003          Y=Y+1
2712
2713 013274 000103  AT103: 103          ;ROUTINE #103 *
2714 013276 013312          AT104       ;ADDRESS OF NEXT TEST *
2715 013300 000003          3.         ;ITERATION COUNT *
2716 013302 013304          DAT3        ;SCOPE ENTRY POINT *
2717 000103          X=X+1
2718
2719 013304 004537 003532  DAT3:  JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2720 013310 000003          3          ;SEE NOTE 3 FOR DATA TEST PARAMETERS
2721 000004          Y=Y+1
2722
2723 013312 000104  AT104: 104          ;ROUTINE #104 *
2724 013314 013330          AT105       ;ADDRESS OF NEXT TEST *
2725 013316 000003          3.         ;ITERATION COUNT *
2726 013320 013322          DAT4        ;SCOPE ENTRY POINT *
2727 000104          X=X+1
```

```

2728
2729 013322 004537 003532
2730 013326 000004
2731 000005
2732
2733 013330 000105
2734 013332 013346
2735 013334 000003
2736 013336 013340
2737 000105
2738
2739 013340 004537 003532
2740 013344 000005
2741 000006
2742
2743 013346 000106
2744 013350 013364
2745 013352 000003
2746 013354 013356
2747 000106
2748
2749 013356 004537 003532
2750 013362 000006
2751 000007
2752
2753 013364 000107
2754 013366 013402
2755 013370 000003
2756 013372 013374
2757 000107
2758
2759 013374 004537 003532
2760 013400 000007
2761 000010
2762
2763 013402 000110
2764 013404 013420
2765 013406 000003
2766 013410 013412
2767 000110
2768
2769 013412 004537 003532
2770 013416 000010
2771 000011
2772
2773 013420 000111
2774 013422 013436
2775 013424 000003
2776 013426 013430
2777 000111
2778
2779 013430 004537 003532
2780 013434 000011
2781 000012
2782
2783 013436 000112

```

```

;*****
DAT4: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      4 ;SEE NOTE 4 FOR DATA TEST PARAMETERS
      Y=Y+1
;*****
AT105: 105 ;ROUTINE #105 *
      AT106 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT5 ;SCOPE ENTRY POINT *
      X=X+1
;*****
DAT5: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      5 ;SEE NOTE 5 FOR DATA TEST PARAMETERS
      Y=Y+1
;*****
AT106: 106 ;ROUTINE #106 *
      AT107 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT6 ;SCOPE ENTRY POINT *
      X=X+1
;*****
DAT6: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      6 ;SEE NOTE 6 FOR DATA TEST PARAMETERS
      Y=Y+1
;*****
AT107: 107 ;ROUTINE #107 *
      AT110 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT7 ;SCOPE ENTRY POINT *
      X=X+1
;*****
DAT7: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      7 ;SEE NOTE 7 FOR DATA TEST PARAMETERS
      Y=Y+1
;*****
AT110: 110 ;ROUTINE #110 *
      AT111 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT10 ;SCOPE ENTRY POINT *
      X=X+1
;*****
DAT10: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      10 ;SEE NOTE 10 FOR DATA TEST PARAMETERS
      Y=Y+1
;*****
AT111: 111 ;ROUTINE #111 *
      AT112 ;ADDRESS OF NEXT TEST *
      3. ;ITERATION COUNT *
      DAT11 ;SCOPE ENTRY POINT *
      X=X+1
;*****
DAT11: JSR 5,DATTST ;LOAD PARAMETERS & RUN TEST
      11 ;SEE NOTE 11 FOR DATA TEST PARAMETERS
      Y=Y+1
;*****
AT112: 112 ;ROUTINE #112 *

```

```

2784 013440 013454          AT113          ;ADDRESS OF NEXT TEST          *
2785 013442 000003          3.          ;ITERATION COUNT              *
2786 013444 013446          DAT12         ;SCOPE ENTRY POINT            *
2787          000112          X=X+1
2788          ;*****
2789 013446 004537 003532 DAT12: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2790 013452 000012          12          ;SEE NOTE 12 FOR DATA TEST PARAMETERS
2791          000013          Y=Y+1
2792          ;*****
2793 013454 000113          AT113: 113          ;ROUTINE #113                  *
2794 013456 013472          AT114         ;ADDRESS OF NEXT TEST          *
2795 013460 000003          3.          ;ITERATION COUNT              *
2796 013462 013464          DAT13         ;SCOPE ENTRY POINT            *
2797          000113          X=X+1
2798          ;*****
2799 013464 004537 003532 DAT13: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2800 013470 000013          13          ;SEE NOTE 13 FOR DATA TEST PARAMETERS
2801          000014          Y=Y+1
2802          ;*****
2803 013472 000114          AT114: 114          ;ROUTINE #114                  *
2804 013474 013510          AT115         ;ADDRESS OF NEXT TEST          *
2805 013476 000003          3.          ;ITERATION COUNT              *
2806 013500 013502          DAT14         ;SCOPE ENTRY POINT            *
2807          000114          X=X+1
2808          ;*****
2809 013502 004537 003532 DAT14: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2810 013506 000014          14          ;SEE NOTE 14 FOR DATA TEST PARAMETERS
2811          000015          Y=Y+1
2812          ;*****
2813 013510 000115          AT115: 115          ;ROUTINE #115                  *
2814 013512 013526          AT116         ;ADDRESS OF NEXT TEST          *
2815 013514 000003          3.          ;ITERATION COUNT              *
2816 013516 013520          DAT15         ;SCOPE ENTRY POINT            *
2817          000115          X=X+1
2818          ;*****
2819 013520 004537 003532 DAT15: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2820 013524 000015          15          ;SEE NOTE 15 FOR DATA TEST PARAMETERS
2821          000016          Y=Y+1
2822          ;*****
2823 013526 000116          AT116: 116          ;ROUTINE #116                  *
2824 013530 013544          AT117         ;ADDRESS OF NEXT TEST          *
2825 013532 000003          3.          ;ITERATION COUNT              *
2826 013534 013536          DAT16         ;SCOPE ENTRY POINT            *
2827          000116          X=X+1
2828          ;*****
2829 013536 004537 003532 DAT16: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST
2830 013542 000016          16          ;SEE NOTE 16 FOR DATA TEST PARAMETERS
2831          000017          Y=Y+1
2832          ;*****
2833 013544 000117          AT117: 117          ;ROUTINE #117                  *
2834 013546 013562          AT120         ;ADDRESS OF NEXT TEST          *
2835 013550 000003          3.          ;ITERATION COUNT              *
2836 013552 013554          DAT17         ;SCOPE ENTRY POINT            *
2837          000117          X=X+1
2838          ;*****
2839 013554 004537 003532 DAT17: JSR      5,DATTST ;LOAD PARAMETERS & RUN TEST

```













```
3069
3070
3071      :*****
3072      :PRG1 - TRANSMITTER SCOPE LOOP
3073      :*****
3073 014422 104000      PRG1:  TYPE                ;TYPE PROGRAM TITLE.
3074 014424 016043      P1TIT
3075 014426 004537 003002      JSR      5,LINSEL      ;GO GET LINE # FROM USER
3076 014432 004737 014740      JSR      %7,SETPAR    ;GO SET PARAMETERS.
3077 014436 104000      TYPE                ;TYPE SELECT CHAR AND DELAY.
3078 014440 016275      SELCAD
3079 014442 000000      HALT                ;WAIT FOR USER.
3080 014444 113737 177570 014462 PRG1A: MOVB      SR,PRG1B      ;DELAY COUNT TO PRG1B.
3081 014452 113777 177571 164426      MOVB      SR+1,@TXBUF  ;LOAD TXBUF.
3082 014460 104016      DELAY                ;DELAY # OF MSECS. SET AT SR.
3083 014462 000000      PRG1B: OPEN
3084 014464 000767      BR      PRG1A        ;REPEAT.
3085
3086      :*****
3087      :PRG2 - RECEIVER SCOPE LOOP.
3088      :*****
3088 014466 104000      PRG2:  TYPE                ;TYPE PROGRAM TITLE.
3089 014470 016103      P2TIT
3090 014472 004537 003002      JSR      5,LINSEL      ;GO GET LINE # FROM USER
3091 014476 004737 014740      JSR      %7,SETPAR    ;GO SET PARAMETERS.
3092 014502 104000      TYPE                ;TYPE SELECT CHAR AND DELAY.
3093 014504 016275      SELCAD
3094 014506 000000      HALT                ;WAIT FOR USER.
3095 014510 004737 014772      PRG2A: JSR      7,STPARB      ;RELOAD PARAMETERS
3096 014514 052777 000004 164362      BIS      #BIT2,@TXCSR ;SET MAINTENANCE BIT.
3097 014522 113737 177570 014540      MOVB      SR,PRG2B      ;DELAY COUNT TO PRG2B.
3098 014530 113777 177571 164350      MOVB      SR+1,@TXBUF  ;LOAD TXBUF.
3099 014536 104016      DELAY                ;DELAY # OF MSECS. SET IN SR.
3100 014540 000000      PRG2B: OPEN
3101 014542 017700 164334      MOV      @RXBUF,%0    ;RXBUF CONTENTS TO R0.
3102 014546 000005      RESET                ;DISPLAY CONTENTS OF RXBUF (IN R0),
3103 014550 000005      RESET                ;BY ISSUING 5 RESET INSTRUCTIONS
3104 014552 000005      RESET
3105 014554 000005      RESET
3106 014556 000005      RESET
3107 014560 000753      BR      PRG2A
3108
3109      :*****
3110      :PRG3 - SINGLE CHARACTER MAINTENANCE MODE DATA TEST.
3111      :*****
3111 014562 104000      PRG3:  TYPE                ;TYPE PROGRAM TITLE.
3112 014564 016604      P3TIT
3113 014566 004537 003002      JSR      5,LINSEL      ;GO GET LINE # FROM USER
3114 014572 004737 014740      JSR      %7,SETPAR    ;SET PARAMETERS.
3115 014576 104000      TYPE                ;TYPE: SELECT CHARACTER.
3116 014600 016750      SELCAR
3117 014602 000000      HALT
3118 014604 113737 177570 001232 PRG3A: MOVB      SR,CBUBA      ;MOVE DATA CHAR TO CBUBA.
3119 014612 004737 014656      JSR      %7,MOUTIN    ;GO OUTPUT, RECEIVE, AND CHECK DATA.
3120 014616 000772      BR      PRG3A
3121
3122      :*****
3123      :PRG4 - SPECIAL BINARY COUNT MAINTENANCE MODE DATA TEST.
3124      :*****
3124 014620 104000      PRG4:  TYPE                ;TYPE PROGRAM TITLE.
```

```
3125 014622 016654          P4TIT
3126 014624 004537 003002   JSR      5,LINSEL      ;GO GET LINE # FROM USER
3127 014630 004737 014740   JSR      %7,SETPAR    ;SET PARAMETERS.
3128 014634 004737 003126   JSR      %7,INBIN     ;INITIALIZE BINARY COUNT.
3129 014640 004737 003232   PRG4A: JSR      %7,GTBINP ;GET BINARY CHARACTER.
3130 014644 110137 001232   MOV      %1,CRBUFA    ;SAVE AT CRBUFA.
3131 014650 004737 014656   JSR      %7,MOUTIN    ;GO OUTPUT, RECEIVE, AND CHECK DATA.
3132 014654 000771          BR       PRG4A        ;REPEAT.
3133          ;SUBROUTINE TO OUTPUT, RECEIVE, AND CHECK DATA WITH MAINTENANCE BIT SET.
3134 014656 032737 000400 177570 MOUTIN: BIT      #BIT8,SR ;SEE IF BIT 8 IS SET.
3135 014664 001001          BNE      .+4         ;BRANCH IF SET.
3136 014666 104002          STALL          ;SET. DO A RANDOM STALL.
3137 014670 105777 164210   TSTB    @TXCSR       ;WAIT FOR TX READY.
3138 014674 100375          BPL      .-4         ;
3139 014676 052777 000004 164200   BIS      #BIT2,@TXCSR ;SET MAINTENANCE BIT.
3140 014704 013777 001232 164174   MOV      CRBUFA,@TXBUF ;LOAD TXBUF.
3141 014712 043737 001234 001232   BIC      CARMSK,CRBUFA ;MASK OFF NON-EXPECTED BITS.
3142 014720 105777 164154          TSTB    @RXCSR       ;WAIT FOR RECEIVER DONE BIT.
3143 014724 100375          BPL      .-4         ;
3144 014726 017737 164150 001230   MOV      @RXBUF,CRBUF ;MOVE CHAR IN RX BUFFER TO CRBUF.
3145 014734 104004          DATCHK          ;COMPARE EXPECTED AND RECEIVED DATA
3146 014736 000207          RTS      %7         ;EXIT.
```

```
3147  
3148 ;SUBROUTINE TO SET STOF CODE,SPEED, AND CHARACTER LENGTH PARAMETERS SET  
3149 ;IN SR INTO TXCSR AND RXCSR.  
3150 014740 104000 SETPAR: TYPE ;TYPE: SELECT PARAMETERS.  
3151 014742 016140 SELPAR  
3152 014744 000000 HALT ;WAIT FOR USER.  
3153 014746 013737 177570 001266 MOV SR,SRT ;SR TO SRT.  
3154 014754 004537 003300 JSR 5,OACNV  
3155 014760 001266 SRT  
3156 014762 016745 APARM  
3157 014764 000002 2  
3158 014766 104000 TYPL  
3159 014770 016727 PARMTS  
3160 014772 032737 000020 001266 STPARB: BIT #BIT4,SRT ;SEE IF SR BIT 4 IS SET.  
3161 015000 001403 BEQ .+10 ;BRANCH IF NOT SET.  
3162 015002 052777 000400 164074 BIS #BIT8,@TXCSR ;SET. SET STOP CODE TO A 1.  
3163 015010 032737 000010 001266 BIT #BIT3,SRT ;SEE IF SR BIT 3 IS SET.  
3164 015016 001406 BEQ .+16 ;BRANCH IF NOT SET.  
3165 015020 052777 000020 164056 BIS #BIT4,@TXCSR ;SET BIT4 IN TXCSR AND RXCSR  
3166 015026 052777 000020 164044 BIS #BIT4,@RXCSR ;(MSB OF SPEED SELECT BITS.)  
3167 015034 032737 000004 001266 BIT #BIT2,SRT ;SEE IF SR BIT 2 IS SET.  
3168 015042 001406 BEQ .+16 ;BRANCH IF NOT SET.  
3169 015044 052777 000010 164032 BIS #BIT3,@TXCSR ;SET BIT3 IN TXCSR AND RXCSR  
3170 015052 052777 000010 164020 BIS #BIT3,@RXCSR ;(LSB OF SPEED SELECT BITS).  
3171 015060 012737 177400 001234 MOV #177400,CARMSK ;SET CHARACTER MASK TO 8 BITS.  
3172 015066 032737 000002 001266 BIT #BIT1,SRT ;SEE IF SR BIT 1 IS SET.  
3173 015074 001421 BEQ STPARA ;BRANCH IF NOT SET.  
3174 015076 012737 177700 001234 MOV #177700,CARMSK ;CHANGE CHAR MASK TO 6 BITS.  
3175 015104 052777 002000 163766 BIS #BIT10,@RXCSR ;SET RXCSR BIT 10(MSB OF CHAR LENGTH BITS).  
3176 015112 032737 000001 001266 BIT #BIT0,SRT ;SEE IF SR BIT0 IS SET.  
3177 015120 001406 BEQ .+16 ;BRANCH IF NOT SET.  
3178 015122 012737 177740 001234 MOV #177740,CARMSK ;CHANGE CHAR MASK TO 5 BITS.  
3179 015130 052777 001000 163742 BIS #BIT9,@RXCSR ;SET RXCSR BIT9 (LSB OF CHAR LENGTH BITS).  
3180 015136 000207 RTS %7 ;EXIT.  
3181 015140 032737 000001 001266 STPARA: BIT #BIT0,SRT ;SEE IF SR BIT0 IS SET.  
3182 015146 001773 BEQ STPARA-2 ;BRANCH IF NOT SET.  
3183 015150 012737 177600 001234 MOV #177600,CARMSK ;CHANGE CHAR MASK TO 7 BITS.  
3184 015156 000764 BR STPARA-10
```

			:VECTOR ASSIGNMENT TABLE			
			VECTAB:		:LINE 0 VECTOR	
3185			300		:LINE 1 VECTOR	
3186			310		:LINE 2 "	
3187	015160	000300	320		" 3 "	
3188	015162	000310	330		" 4 "	
3189	015164	000320	340		" 5 "	
3190	015166	000330	350		" 6 "	
3191	015170	000340	360		" 7 "	
3192	015172	000350	370		" 10 "	
3193	015174	000360	400		" 11 "	
3194	015176	000370	410		" 12 "	
3195	015200	000400	420		" 13 "	
3196	015202	000410	430		" 14 "	
3197	015204	000420	440		" 15 "	
3198	015206	000430	450		" 16 "	
3199	015210	000440	460		" 17 "	
3200	015212	000450	470		" 20 "	
3201	015214	000460	500		" 21 "	
3202	015216	000470	510		" 22 "	
3203	015220	000500	520		" 23 "	
3204	015222	000510	530		" 24 "	
3205	015224	000520	540		" 25 "	
3206	015226	000530	550		" 26 "	
3207	015230	000540	560		" 27 "	
3208	015232	000550	570		" 30 "	
3209	015234	000560	600		" 31 "	
3210	015236	000570	610		" 32 "	
3211	015240	000600	620		" 33 "	
3212	015242	000610	630		" 34 "	
3213	015244	000620	640		" 35 "	
3214	015246	000630	650		" 36 "	
3215	015250	000640	660		" 37 "	
3216	015252	000650	670			
3217	015254	000660				
3218	015256	000670				
3219						

```
3220
3221          ;ASCII MESSAGES
3222 015260 050045 040          EMO: .ASCII '%P '
3223 015263 040 020040 052040 APNUMB: .ASCII ' T '
3224 015270 040
3225 015271 040 020040 050040 ATNUMB: .ASCII ' PC '
3226 015276 020103
3227 015300 020040 020040 020040 APC: .ASCII ' @'
3228 015306 020040 100
3229 015311 045 050045 043522 POTIT: .ASCII '%%PRGO - INPUT-OUTPUT LOGIC TESTS. '
3230 015316 020060 020055 047111
3231 015324 052520 026524 052517
3232 015332 050124 052125 046040
3233 015340 043517 041511 052040
3234 015346 051505 051524 020056
3235 015354 044504 041523 047117          .ASCII 'DISCONNECT DC11 FROM MODEM.%'
3236 015362 042516 052103 042040
3237 015370 030503 020061 051106
3238 015376 046517 046440 042117
3239 015404 046505 022456
3240 015410 047101 020104 047503          .ASCII 'AND CONNECT JUMPER TO CABLE.%@'
3241 015416 047116 041505 020124
3242 015424 052512 050115 051105
3243 015432 052040 020117 040503
3244 015440 046102 027105 040045
3245 015446 054124 051503 020122 ATXCSR: .ASCII 'TXCSR S/B: '
3246 015454 027523 035102 040
3247 015461 040 020040 020040 ATXSB: .ASCII ' WAS: '
3248 015466 020040 053440 051501
3249 015474 020072
3250 015476 020040 020040 020040 ATXWAS: .ASCII ' @'
3251 015504 100
3252 015505 122 041530 051123 ARXCSR: .ASCII 'RXCSR S/B: '
3253 015512 051440 041057 020072
3254 015520 020040 020040 020040 ARXSB: .ASCII ' WAS: '
3255 015526 020040 040527 035123
3256 015534 040
3257 015535 040 020040 020040 ARXWAS: .ASCII ' @'
3258 015542 040040
3259 015544 054124 051440 042520 ETXTIM: .ASCII 'TX SPEEDS NOT IN ASCENDING ORDER.@'
3260 015552 042105 020123 047516
3261 015560 020124 047111 040440
3262 015566 041523 047105 044504
3263 015574 043516 047440 042122
3264 015602 051105 040056
3265 015606 044524 042515 043040 ESTPCD: .ASCII 'TIME FOR 2 STOP CODE OP LESS THAN FOR 1.@'
3266 015614 051117 031040 051440
3267 015622 047524 020120 047503
3268 015630 042504 047440 020120
3269 015636 042514 051523 052040
3270 015644 040510 020116 047506
3271 015652 020122 027061 100
3272 015657 124 020130 044103 ETCLGT: .ASCII 'TX CHAR LENGTHS NOT IN DESCENDING ORDER.@'
3273 015664 051101 046040 047105
3274 015672 052107 051510 047040
3275 015700 052117 044440 020116
```

3276	015706	042504	041523	047105	
3277	015714	044504	043516	047440	
3278	015722	042122	051105	040056	
3279	015730	054122	051440	042520	ERXTIM: .ASCII 'RX SPEEDS NOT IN ASCENDING ORDER.@'
3280	015736	042105	020123	047516	
3281	015744	020124	047111	040440	
3282	015752	041523	047105	044504	
3283	015760	043516	047440	042122	
3284	015766	051105	040056		
3285	015772	054122	041440	040510	ERCLGT: .ASCII 'RX CHAR LENGTHS NOT IN DESCENDING ORDER.@'
3286	016000	020122	042514	043516	
3287	016006	044124	020123	047516	
3288	016014	020124	047111	042040	
3289	016022	051505	042503	042116	
3290	016030	047111	020107	051117	
3291	016036	042504	027122	100	
3292	016043	045	050045	043522	P1TIT: .ASCII '%PRG1 - TRANSMITTER SCOPE LOOP@'
3293	016050	020061	020055	051124	
3294	016056	047101	046523	052111	
3295	016064	042524	020122	041523	
3296	016072	050117	020105	047514	
3297	016100	050117	100		
3298	016103	045	050045	043522	P2TIT: .ASCII '%PRG2 - RECEIVER SCOPE LOOP@'
3299	016110	020062	020055	042522	
3300	016116	042503	053111	051105	
3301	016124	051440	047503	042520	
3302	016132	046040	047517	040120	
3303	016140	051445	052105	050040	SELPAR: .ASCII '%SET PARAMETERS IN SR AS FOLLOWS:'
3304	016146	051101	046501	052105	
3305	016154	051105	020123	047111	
3306	016162	051440	020122	051501	
3307	016170	043040	046117	047514	
3308	016176	051527	072		
3309	016201	045	051123	020064	.ASCII '%SR4 - STOP CODE%SR3 AND 2 - SPEED'
3310	016206	020075	052123	050117	
3311	016214	041440	042117	022505	
3312	016222	051123	020063	047101	
3313	016230	020104	020062	020075	
3314	016236	050123	042505	104	
3315	016243	045	051123	020061	.ASCII '%SR1 AND 0 = CHAR LENGTH%@'
3316	016250	047101	020104	020060	
3317	016256	020075	044103	051101	
3318	016264	046040	047105	052107	
3319	016272	022510	100		
3320	016275	045	042523	020124	SELCAD: .ASCII '%SET TEST CHAR CODE IN SR15-SR8, SET DELAY TIME IN SR7-SR0.@'
3321	016302	042524	052123	041440	
3322	016310	040510	020122	047503	
3323	016316	042504	044440	020116	
3324	016324	051123	032461	051455	
3325	016332	034122	020054	042523	
3326	016340	020124	042504	040514	
3327	016346	020131	044524	042515	
3328	016354	044440	020116	051123	
3329	016362	026467	051123	027060	
3330	016370	100			
3331	016371	040	042040	052101	ERDAT: .ASCII ' DATA ERR S/B: '

3332	016376	020101	051105	020122		
3333	016404	051440	041057	020072		
3334	016412	020040	020040	053440	AASB: .ASCII	' WAS: '
3335	016420	051501	020072			
3336	016424	020040	040040		AWAS: .ASCII	' @'
3337	016430	037445	100		AINPRG: .ASCII	'%?@'
3338	016433	045	042523	020124	ASETSR: .ASCII	'%SET DESIRED SR OPTIONS. NORMAL OPERATION '
3339	016440	042504	044523	042522		
3340	016446	020104	051123	047440		
3341	016454	052120	047511	051516		
3342	016462	020056	047516	046522		
3343	016470	046101	047440	042520		
3344	016476	040522	044524	047117		
3345	016504	040				
3346	016505	111	020123	044527	.ASCII	'IS WITH SR = 000000@'
3347	016512	044124	051440	020122		
3348	016520	020075	030060	030060		
3349	016526	030060	100			
3350	016531	045	047111	047503	AINCRT: .ASCII	'%INCORRECT ROUTINE SELECTED.@'
3351	016536	051122	041505	020124		
3352	016544	047522	052125	047111		
3353	016552	020105	042523	042514		
3354	016560	052103	042105	040056		
3355	016565	050045	047522	051107	APGEN: .ASCII	'%PROGRAM END.@'
3356	016574	046501	042440	042116		
3357	016602	040056				
3358	016604	022445	051120	031507	P3TIT: .ASCII	'%%PRG3-SINGLE CHAR MAINT MODE DATA TEST@'
3359	016612	051455	047111	046107		
3360	016620	020105	044103	051101		
3361	016626	046440	044501	052116		
3362	016634	046440	042117	020105		
3363	016642	040504	040524	052040		
3364	016650	051505	040124			
3365	016654	022445	051120	032107	P4TIT: .ASCII	'%%PRG4-SPEC BIN COUNT MAINT MODE DATA TEST@'
3366	016662	051455	042520	020103		
3367	016670	044502	020116	047503		
3368	016676	047125	020124	040515		
3369	016704	047111	020124	047515		
3370	016712	042504	042040	052101		
3371	016720	020101	042524	052123		
3372	016726	100				
3373	016727	045	040520	040522	PARMTS: .ASCII	'%PARAMETERS = '
3374	016734	042515	042524	051522		
3375	016742	036440	040			
3376	016745	040	040040		ARM: .ASCII	' @'
3377	016750	051445	052105	052040	LCAR: .ASCII	'%SET TEST CHAR CODE IN SR7-SR0.@'
3378	016756	051505	020124	044101		
3379	016764	051101	041440	042117		
3380	016772	020105	047111	051440		
3381	017000	033522	051455	030122		
3382	017006	040056				
3383	017010	046045	040517	020104	LDLINE: .ASCII	'%LOAD LINE NO. (8) INTO SR 3-7@'
3384	017016	044514	042516	047040		
3385	017024	027117	024040	024470		
3386	017032	044440	052116	020117		
3387	017040	051123	031440	033455		

3388	017046	100					
3389	017047	040	046040	047111	ALINE:	.ASCII	' LINE NO.'
3390	017054	020105	047516	056			
3391	017061	040	020040	040527	SELINE:	.ASCII	' WAS SELECTED@'
3392	017066	020123	042523	042514			
3393	017074	052103	042105	100			
3394	017101	000001			DEND:	.END	



CROSS REFERENCE TABLE -- USER SYMBOLS

AAA	003766	1179	1182#				
AAAA	006300	1686	1689#				
AAAB	006312	1690	1692#				
AAB	004000	1184#	1187				
AABA	012624	2574	2579#				
AABB	012650	2585#	2591	2594			
AABC	012660	2580	2587#				
AABD	012674	2588	2590#				
AABE	012700	2587	2592#				
AAE	004002	1182	1185#				
AASB	016412	743	3334#				
ABA	004020	1192	1195#				
ABAA	007302	1904	1907#				
ABAB	007374	1911	1924#				
ABB	004032	1197#	1200				
ABBA	012716	2600	2603#				
ABBB	012750	2609	2611#				
ABE	004034	1195	1198#				
ACA	004052	1206	1210#				
ACAA	007406	1929	1933#				
ACAB	007542	1946	1958#				
ACB	004064	1212#	1215				
ACBA	012776	2621	2626#				
ACBB	013034	2625	2634#				
ACBC	013036	2633	2635#				
ACE	004066	1210	1213#				
ADA	004104	1221	1224#				
ADAA	007562	1964	1967#				
ADAB	007576	1969	1971#				
ADB	004116	1226#	1229				
ADE	004120	1224	1227#				
ADTENP	003512	1101	1120#				
AEA	004136	1235	1238#				
AEAA	007612	1977	1981#				
AEAB	007632	1985	1988#				
AEB	004152	1239	1242#				
AEC	004174	1244	1247#				
AED	004214	1241	1246	1249	1251#		
AFA	004236	1258	1261#				
AFAA	007644	1994	1998#				
AFAB	007672	2003	2006#				
AFB	004260	1263	1266#				
AFBA	006324	1703	1708#				
AFBB	006346	1710	1713#				
AFBC	006366	1712	1715	1717#			
AFC	004302	1268	1271#				
AFD	004322	1265	1270	1273	1275#		
AGA	004334	1281	1284#				
AGAA	007704	2011	2015#				
AGAB	007732	2020	2023#				
AGB	004350	1285	1288#				
AGBA	006402	1723	1728#				
AGBB	006432	1731	1734#				
AGBC	006462	1740	1743#				
AGBD	006476	1745	1749#				
AGBE	006526	1733	1742	1747	1755	1757	1758#

.MAIN. MACY11 30A(1052) 17-MAR-78 13:05 PAGE 75  
 CZDCAC.P11 17-MAR-78 13:04 CROSS REFERENCE TABLE -- USER SYMBOLS

AGC	004372	1290	1293#						
AGD	004412	1287	1292	1295	1297#				
AHA	004434	1304	1307#						
AHAA	007744	2028	2032#						
AHAB	007772	2037	2040#						
AHB	004450	1308	1311#						
AHC	004472	1313	1316#						
AHD	004512	1310	1315	1318	1320#				
AIA	004534	1328	1331#						
AIAA	010004	2046	2050#						
AIAF	010144	2068	2071#						
AIAS	010146	2055	2058	2062	2065	2072#			
AIASA	010170	2078#	2082						
AIAS	010210	2056	2059	2063	2066	2072*	2080*	2084#	
AIB	004550	1332	1335#						
AIC	004572	1337	1340#						
AID	004612	1334	1339	1342	1344#				
AINCRT	016531	718	3350#						
AINPRG	016430	3337#							
AJA	004634	1351	1354#						
AJAA	010222	2089	2094#						
AJAB	010306	2103	2107#						
AJAS	010310	2097	2100	2108#					
AJASA	010332	2114#	2118						
AJASB	010354	2120#	2124						
AJAST	010374	2098	2101	2108*	2116*	2122*	2126#		
AJB	004656	1356	1359#						
AJBA	006656	1795	1800#						
AJBB	006674	1802	1805#						
AJBC	006716	1807	1810#						
AJBD	006736	1809	1812	1814#					
AJC	004700	1361	1364#						
AJD	004720	1358	1363	1366	1368#				
AKA	004742	1377	1381#						
AKAA	010406	2132	2137#						
AKAB	010554	2156	2159#						
AKAS	010556	2142	2146	2150	2153	2160#			
AKASA	010600	2166#	2170						
AKAST	010620	2143	2147	2151	2154	2160*	2168*	2172#	
AKB	004752	1382	1384#						
AKBA	006752	1820	1826#						
AKBB	006760	1827#							
AKBC	007010	1830	1833#						
AKBD	007024	1834	1837#						
AKBE	007054	1832	1836	1841	1844#				
ALA	004764	1389	1392#						
ALAA	010632	2178	2183#						
ALAB	010662	2188	2191#						
ALAC	010674	2190	2193	2195#					
ALB	005000	1393	1396#						
ALBA	007176	1877	1882#						
ALBB	007220	1884	1887#						
ALBC	007246	1891	1894#						
ALBD	007266	1886	1893	1896	1898#				
ALC	005022	1398	1401#						
ALD	005042	1395	1400	1403	1405#				

ALINE	017047	1037	3389#					
AMA	005064	1414	1417#					
AMAA	010706	2200	2204#					
AMAB	010720	2206#	2207					
AMAC	010742	2210	2212#					
AMB	005076	1418	1420#					
AMBA	006542	1765	1770#					
AMBB	006600	1775	1778#					
AMBC	006614	1779	1782#					
AMBD	006642	1777	1787	1785		1787#		
ANA	005110	1426	1429#					
ANAA	010754	2218	2223#					
ANAB	011024	2231	2233#					
ANB	005132	1431	1434#					
ANBA	013050	2646#						
ANBB	013062	2643	2648#					
ANBC	013070	2649#	2678					
ANBD	013132	2658#	2662					
ANBE	013166	2664	2668#					
ANBF	013172	2666	2670#					
ANBG	013212	2667	2669	2677#				
ANC	005150	1436	1439#					
AND	005170	1433	1438	1441		1443#		
ANW	005202	1448	1452#					
ANX	005216	1453	1456#					
ANY	005240	1458	1461#					
ANZ	005260	1455	1460	1463		1465#		
AQAA	011042	2240	2245#					
AQAB	011112	2253	2255#					
AQBA	007070	1850	1854#					
AQBB	007112	1856	1859#					
AQBC	007142	1862	1865#					
AQBD	007162	1858	1864	1867		1869#		
APA	005302	1473	1476#					
APAA	011130	2261	2266#					
APAB	011164	2272	2274#					
APARM	016745	3156	3376#					
APB	005314	1477	1479#					
APBA	014152	3006	3015#					
APBB	014160	3016#	3027					
APC	015300	764	3227#					
APGEND	016566	722	3355#					
APNUMB	015263	768	3223#					
AQA	005326	1484	1487#					
AQAA	011202	2282	2287#					
AQAB	011422	2313	2316#					
AQAS	011424	2294	2300	2306		2310	2317#	
AQASA	011452	2324#	2328					
AQAST	011472	2295	2301	2307		2311	2317*	2326* 2330#
AQB	005342	1488	1491#					
AQBA	014246	3033	3038#					
AQC	005364	1493	1496#					
AQD	005404	1490	1495	1498		1500#		
ARA	005426	1507	1510#					
ARAA	011504	2336	2341#					
ARAB	011652	2360	2363#					

.MAIN. MACY11 30A(1052) 17-MAR-78 13:05 PAGE 77  
 CZDCAC.P11 17-MAR-78 13:04

SEQ 0075

## CROSS REFERENCE TABLE -- USER SYMBOLS

ARAS	011654	2346	2350	2354	2357	2364#		
ARASA	011710	2373#	2377					
ARAST	011730	2347	2351	2355	2358	2364*	2375*	2379#
ARB	005442	1511	1514#					
ARBA	005526	1531	1536#					
ARBB	005540	1537	1539#					
ARC	005464	1516	1519#					
ARD	005504	1513	1518	1521	1523#			
ARXCSR	015505	1957	3252#					
ARXSB	015520	1950	3254#					
ARXWAS	015535	1954	3257#					
ASA	005552	1545	1548#					
ASAA	011742	2385	2388#					
ASAB	011774	2392	2395#					
ASAC	012006	2396	2400#					
ASAD	012020	2394	2399	2402	2405#			
ASAS	012026	2388	2389	2407#				
ASB	005574	1550	1553#					
ASC	005616	1555	1558#					
ASD	005636	1552	1557	1560	1562#			
ASETSR	016433	714	3338#					
ATA	005660	1569	1572#					
ATAA	012066	2417	2424#					
ATAB	012112	2429#	2431					
ATAC	012114	2423	2430#					
ATB	005672	1573	1575#					
ATLAST=	177777	643#	3031					
ATNUMB	015271	772	3225#					
ATXCSR	015446	1923	3245#					
ATXSB	015461	1916	3247#					
ATXWAS	015476	1920	3250#					
ATO	003756	1167	1176#					
AT1	004010	1177	1189#					
AT10	004524	1302	1325#					
AT100	013222	2641	2683#					
AT101	013240	2684	2693#					
AT102	013256	2694	2703#					
AT103	013274	2704	2713#					
AT104	013312	2714	2723#					
AT105	013330	2724	2733#					
AT106	013346	2734	2743#					
AT107	013364	2744	2753#					
AT11	004624	1326	1348#					
AT110	013402	2754	2763#					
AT111	013420	2764	2773#					
AT112	013436	2774	2783#					
AT113	013454	2784	2793#					
AT114	013472	2794	2803#					
AT115	013510	2804	2813#					
AT116	013526	2814	2823#					
AT117	013544	2824	2833#					
AT12	004732	1349	1374#					
AT120	013562	2834	2843#					
AT121	013600	2844	2853#					
AT122	013616	2854	2863#					
AT123	013634	2864	2873#					

.MAIN. MACY11 30A(1052) 17-MAR-78 13:05 PAGE 78  
 CZDCAC.P11 17-MAR-78 13:04 CROSS REFERENCE TABLE -- USER SYMBOLS

AT124	013652	2874	2883#
AT125	013670	2884	2893#
AT126	013706	2894	2903#
AT127	013724	2904	2913#
AT13	004754	1375	1386#
AT130	013742	2914	2923#
AT131	013760	2924	2933#
AT132	013776	2934	2943#
AT133	014014	2944	2953#
AT134	014032	2954	2963#
AT135	014050	2964	2973#
AT136	014066	2974	2983#
AT137	014104	2984	2993#
AT14	005054	1387	1411#
AT140	014122	2994	3003#
AT141	014236	3004	3030#
AT15	005100	1412	1423#
AT16	005172	1424	1445#
AT17	005272	1446	1470#
AT2	004042	1190	1203#
AT20	005316	1471	1481#
AT21	005416	1482	1504#
AT22	005516	1505	1528#
AT23	005542	1529	1542#
AT24	005650	1543	1566#
AT25	005674	1567	1578#
AT26	005774	1579	1601#
AT27	006074	1602	1624#
AT3	004074	1204	1218#
AT30	006174	1625	1647#
AT31	006220	1648	1659#
AT32	006244	1660	1672#
AT33	006270	1673	1683#
AT34	006314	1684	1700#
AT35	006372	1701	1720#
AT36	006532	1721	1762#
AT37	006646	1763	1792#
AT4	004126	1219	1232#
AT40	006742	1793	1817#
AT41	007060	1818	1847#
AT42	007166	1848	1874#
AT43	007272	1875	1901#
AT44	007376	1902	1926#
AT45	007552	1927	1961#
AT46	007602	1962	1974#
AT47	007634	1975	1991#
AT5	004226	1233	1255#
AT50	007674	1992	2008#
AT51	007734	2009	2025#
AT52	007774	2026	2043#
AT53	010212	2044	2086#
AT54	010376	2087	2129#
AT55	010622	2130	2175#
AT56	010676	2176	2197#
AT57	010744	2198	2215#
AT6	004324	1256	1278#

CROSS REFERENCE TABLE -- USER SYMBOLS

AT60	011032	2216	2237#		
AT61	011120	2238	2258#		
AT62	011172	2259	2279#		
AT63	011474	2280	2333#		
AT64	011732	2334	2382#		
AT65	012046	2383	2414#		
AT66	012120	2415	2435#		
AT67	012176	2436	2457#		
AT7	004424	1279	1301#		
AT70	012262	2458	2479#		
AT71	012360	2480	2505#		
AT72	012436	2506	2527#		
AT73	012520	2528	2548#		
AT74	012610	2549	2571#		
AT75	012706	2572	2597#		
AT76	012762	2598	2618#		
AT77	013040	2619	2640#		
AUA	005704	1581	1584#		
AUAA	012134	2438	2444#		
AUAB	012160	2450#			
AUAC	012166	2443	2451#	2454	
AUB	005720	1585	1588#		
AUC	005742	1590	1593#		
AUD	005762	1587	1592	1595	1597#
AVA	006004	1604	1607#		
AVAA	012212	2460	2466#		
AVAB	012250	2465	2473#		
AVAC	012252	2472	2474#		
AVB	006020	1608	1611#		
AVC	006042	1613	1616#		
AVD	006062	1610	1615	1618	1620#
AWA	006104	1627	1630#		
AWAA	012272	2482	2486#		
AWAB	012322	2493#	2499	2502	
AWAC	012332	2487	2495#		
AWAD	012346	2496	2498#		
AWAE	012352	2495	2500#		
AWAS	016424	739	2673	2676	3336#
AWB	006120	1631	1634#		
AWC	006142	1636	1639#		
AWD	006162	1633	1638	1641	1643#
AXA	006204	1650	1653#		
AXAA	012404	2508	2516#		
AXAB	012432	2514	2522#		
AXAC	012434	2521	2523#		
AXB	006216	1654	1656#		
AYA	006230	1662	1665#		
AYAA	012456	2530	2537#		
AYAB	012502	2541#	2545		
AYAC	012512	2535	2543#		
AYB	006242	1666	1668#		
AZA	006254	1675	1678#		
AZAA	012540	2551	2558#		
AZAB	012576	2556	2566#		
AZAC	012600	2564	2567#		
AZB	006266	1679	1681#		















CROSS REFERENCE TABLE -- USER SYMBOLS

- 017101

2811#	2812	2821#	2822	2831#	2832	2841#	2842	2851#	2852	2861#	2862	2871#
2872	2881#	2882	2891#	2892	2901#	2902	2911#	2912	2921#	2922	2931#	2932
2941#	2942	2951#	2952	2961#	2962	2971#	2972	2981#	2982	2991#	2992	3001#
463#	464	466	468	470	472	474	478	480	482	484	486	488
490	492	494	496	498	500	502	504	506	508	510	512	514
516	518	520	522	524	526	528	530	532	534	536	538	540
542	544	546	548	550	552	554	556	558	560	562	564	566
568	570	572	574	576	578	580	582	584	586	588	590	645#
647#	724	948	1022	1056	1065	1133	1139	1939	1942	2074	2110	2162
2319	2366	2652	2655	2659	3017	3023	3042	3044	3135	3138	3143	3161
3164	3168	3177										

. ABS. 017101 000

ERRORS DETECTED: 0

DSKZ:CZDCAC,DSKZ:CZDCAC,SEQ-DSKZ:CZDCAC.SML,DSKZ:CZDCAC.P11

RUN-TIME: 10 15 1 SECONDS

RUN-TIME RATIO: 123/27-4.4

CORE USED: 32K (63 PAGES)

DOCUMENT PAGES: 84

