

DMP-11
DMR-11

M8203 STATIC DIAG.#1
CZDMRDO

AH-E232D-MC
FICHE 1 OF 2

NOV 1980
COPYRIGHT © 79-80
MADE IN USA



DMP-11
DMR-11

M8203 STATIC DIAG.#1
CZDMRDO

AH-E232D-MC
FICHE 2 OF 2

NOV 1980
COPYRIGHT © 79-80
MADE IN USA



.REM @

IDENTIFICATION

PRODUCT CODE: AC-E231D-MC
PRODUCT NAME: CZDMRD0 M8203 STATIC DIAG #1
PRODUCT DATE: AUGUST 1980
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHJR: DAVID HOFFMAN

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 RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979, 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

CONTENTS

- 1.0 INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
 - 4.1 DIAGNOSTIC SUPERVISOR
 - 4.2 EXECUTION TIME
 - 4.3 XXDP+
 - 4.4 ACT/SLIDE
 - 4.5 APT
 - 4.6 MEMORY MANAGEMENT
 - 4.7 MEMORY PARITY OPTION
 - 4.8 ERROR LOGGING
- 5.0 PROGRAM LOAD MEDIA
- 6.0 OPERATING INSTRUCTIONS
 - 6.1 LOADING AND STARTING PROCEDURES
 - 6.1.1 LOADING PROCEDURES
 - 6.1.2 STARTING PROCEDURES
 - 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
 - 6.2 INITIAL DIALOGUE
 - 6.3 PROGRAM OPTIONS
 - 6.3.1 START COMMAND
 - 6.3.1.1 TESTS SWITCH
 - 6.3.1.2 PASS SWITCH
 - 6.3.1.3 FLAGS SWITCH
 - 6.3.1.4 END OF PASS SWITCH
 - 6.3.1.5 EFFECT OF START COMMAND
 - 6.3.2 RESTART COMMAND
 - 6.3.2.1 TESTS, PASS, AND FLAG SWITCHES
 - 6.3.2.2 UNITS SWITCH
 - 6.3.2.3 EFFECT OF RESTART COMMAND
 - 6.3.3 CONTINUE COMMAND
 - 6.3.3.1 PASS SWITCH
 - 6.3.3.2 FLAGS SWITCH
 - 6.3.3.3 EFFECT OF CONTINUE COMMAND
 - 6.3.4 PROCEED COMMAND
 - 6.3.4.1 FLAGS SWITCH
 - 6.3.4.2 EFFECT OF PROCEED COMMAND
 - 6.3.5 ADD COMMAND
 - 6.3.5.1 UNITS SWITCH
 - 6.3.5.2 EFFECT OF ADD COMMAND
 - 6.3.6 DROP COMMAND
 - 6.3.6.1 UNITS SWITCH
 - 6.3.6.2 EFFECT OF DROP COMMAND
 - 6.3.7 PRINT COMMAND
 - 6.3.7.1 EFFECT OF PRINT COMMAND

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

6.3.8 DISPLAY COMMAND
6.3.8.1 UNITS SWITCH
6.3.8.2 EFFECT OF DISPLAY COMMAND
6.3.9 FLAGS COMMAND
6.3.9.1 EFFECT OF FLAGS COMMAND
6.3.10 ZFLAGS COMMAND
6.3.10.1 EFFECT OF ZFLAGS COMMAND
6.3.11 CONTROL CHARACTERS
6.3.12 HARDWARE PARAMETERS
6.3.13 SOFTWARE PARAMETERS
6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

7.0 DEVICE INFORMATION TABLES

8.0 TEST DESCRIPTIONS

8.1 DATA PATTERNS USED

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

1.0 INTRODUCTION

THE M8203 IS A SINGLE-LINE SYNCHRONOUS LINE UNIT MODULE WHICH SUPPORTS BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS TO PERFORM DIAGNOSTIC TESTING OF ALL M8203 LOGIC IN A RELATIVELY STATIC MANNER. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: LINE UNIT REGISTER ADDRESSING, USYRT ADDRESSING, STATIC BIT INTERACTION AND READ/WRITE LOGIC TESTS, BASIC TRANSMITTER AND RECEIVER SEQUENCING AND DATA BUFFERING AND STATIC OPERATIONS IN CHARACTER AND BIT-STUFFING MODES. IN ADDITION DATA MESSAGES WILL BE SENT AT SPEEDS OF 2400 BAUD TO 1 MEGABAUD, WITH LOOPBACK IN THE USYRT, ON THE LINE UNIT AT TTL LEVEL, OR THROUGH AN EXTERNAL TEST CONNECTOR WITH A SPECIFIC MODEM INTERFACE SELECTED.

THE STATIC LOGIC TESTS WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO 'LOCK' ONTO INTERMITTENT ERRORS. IN ADDITION TESTS WILL BE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

THIS PROGRAM WILL BE IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN WILL CONFORM TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM WILL BE COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8203 STATIC LOGIC TESTS:

PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70
16K MEMORY
CONSOLE TERMINAL
DMC-11 OR KMC-11 MICROPROCESSOR

M8203 LINE UNIT AND BC08S-1 CABLE AND BERG CONNECTORS

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM OPERATES THE MICROPROCESSOR EXTENSIVELY IN ORDER TO TEST THE LINE INIT. FOR THIS REASON, THE MICROPROCESSOR DIAGNOSTIC AND SUBSYSTEM FUNCTIONAL TESTS SHOULD BE RUN FIRST, AND ANY FAULTS FOUND IN THE MICROPROCESSOR MODULE SHOULD BE REPAIRED, PRIOR TO RUNNING THE M8203 STATIC LOGIC TESTS.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE MAXIMUM TIME REQUIRED TO RUN THE M8203 STATIC LOGIC TESTS IS ABOUT 45 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE
DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT
THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR
RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE
ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM
ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE
ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST,
FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE
DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY
THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE
ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD
MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR
WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC
PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+,
WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS-C>)
- C) ENTER STA<CR>
- D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
- E) GET END OF PASS MESSAGES OR ERROR MESSAGES
- F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

DRS LOADED
DIAG. RUN-TIME SERVICES
CZDMR-D-0
M8203 STATIC LOGIC TESTS - PART 1 OF 2
UNIT IS M8203
CR>

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/EOP:<INCR>

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT	LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION '# UNITS?' TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM 'UNIT' REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION "'# UNITS?'" IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
<FLAG-LIST>/UNITS:<UNIT-LIST>

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT

THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>

6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

PRO(CEED)/FLAGS:<FLAG-LIST>

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

6.3.5 ADD COMMAND

ADD/UNITS:<UNIT-LIST>

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

DRO(P)/UNITS:<UNIT-LIST>

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST
START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT
STATISTICAL REPORTING) FLAG IS CLEARED.

6.3.8 DISPLAY COMMAND

DIS(PLAY)/UNITS:<UNIT-LIST>

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED
OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS
THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO
DESIGNATED.

6.3.9 FLAGS COMMAND

FLA(GS)

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

ZFL(AGS)

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES
A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR
DIALOGUES- HARD CORE QUESTIONS (SEE 6.2), HARDWARE DIALOGUE

(SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE
DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL 0 (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC
CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE
REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER 0 IS TYPED,
WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 4 QUESTIONS WILL BE ASKED ON A START COMMAND.
THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE
DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN
RESPONSE.

1. DEVICE CSR ADDRESS : (O) 160170?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE
ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776
(OCTAL), AND THE DEFAULT VALUE IS 160170.

2. DEVICE VECTOR ADDRESS : (O) 300 ?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS
DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE
DEFAULT VALUE IS 300.

3. DEVICE PRIORITY LEVEL : (O) 5 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF
THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7,
AND THE DEFAULT VALUE IS 5.

4. M8207 RUN SWITCH (E28 SW7) - TYPE 0 IF OFF, 1 IF ON : (O) 1 ?

THIS TELLS THE PROGRAM IF THE RUN SWITCH ON THE MICROPROCESSOR
IS SET OR NOT. IF IT IS SET, RUN IS NOT INHIBITED, AND TESTS
REQUIRING THE RUN STATE CAN BE EXECUTED. THE ALLOWABLE VALUES
ARE 0 AND 1, AND THE DEFAULT VALUE IS 1 (RUN IS NOT INHIBITED).

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 OF THE
STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY
THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "N UNITS?" IS ANSWERED (WITH THE
NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES.
ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A
ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER

571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627

QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

UNITS (D) ? 16
UNIT 0
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 7
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15.

628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644

SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

7.0 DEVICE INFORMATION TABLES

```
*****
* MAINTENANCE REGISTER - BSEL1
*****
RUN      = BIT7
MCLR     = BIT6
STEPLU   = BIT4
LULOOK   = BIT3
ROMO     = BIT2
ROMI     = BIT1
STEPMP   = BIT0

*****
* OBUS REG 10 - TRANSMITTER BUFFER
*****
TX7      = BIT7
TX6      = BIT6
TX5      = BIT5
TX4      = BIT4
TX3      = BIT3
TX2      = BIT2
TX1      = BIT1
TX0      = BIT0

*****
* OBUS REG 11
*****
OC       = BIT7
GOAH     = BIT3
ABORT    = BIT2
EOM      = BIT1
SOM      = BIT0

*****
* OBUS REG 12
*****
IC       = BIT7
BPOLL    = BIT6
LULP     = BIT5

*****
* OBUS REG 13
*****
POLL     = BIT7
DTR      = BIT6
SEIFR    = BIT5
HDX      = BIT4
MAINT1   = BIT3
MAINT2   = BIT2
SELSBY   = BIT1
*****
```

```
58      * OBUS REG 14
59      ;*****
60      TXEN      = BIT6
61      DISSI     = BIT5
62      RDAX      = BIT4
63      WAX       = BIT3
64      ENAX      = BIT2
65      AX2       = BIT1
66      AX1       = BIT0
67
68      ;*****
69      * OBUS REG 17
70      ;*****
71      CRC2      = BIT7
72      CRC1      = BIT6
73      IDLE      = BIT5
74      SECA      = BIT4
75      STRIP     = BIT3
76      RDALL     = BIT2
77      IERR      = BIT1
78      DDCMP     = BIT0
79
80      ;*****
81      * IBUS REG 10 - RECEIVER BUFFER
82      ;*****
83      RX7       = BIT7
84      RX6       = BIT6
85      RX5       = BIT5
86      RX4       = BIT4
87      RX3       = BIT3
88      RX2       = BIT2
89      RX1       = BIT1
90      RX0       = BIT0
91
92      ;*****
93      * IBUS REG 11
94      ;*****
95      OC        = BIT7
96      OACT      = BIT6
97      SW3       = BIT5
98      ORDY     = BIT4
99      SW2       = BIT3
100     SW1       = BIT2
101     SW0       = BIT1
102     UNRR      = BIT0
103
104     ;*****
105     * IBUS REG 12
106     ;*****
107     IC        = BIT7
108     IACT      = BIT6
109     LULP      = BIT5
110     IRDY     = BIT4
111     OVRR      = BIT3
112     RAB       = BIT2
113     EBLK     = BIT1
114     BCC       = BIT0
```

```
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
```

```
*****
* IBUS REG 13
*****
RING      = BIT7
DTR       = BIT6
RTS       = BIT5
HDX       = BIT4
MODR      = BIT3
CS        = BIT2
STBY      = BIT1
CARR      = BIT0

*****
* IBUS REG 14
*****
READY     = BIT7
TXEN      = BIT6
DISSI     = BIT5
RDAX      = BIT4
WAX       = BIT3
ENAX      = BIT2
AX2       = BIT1
AX1       = BIT0

*****
* IBUS REG 17
*****
SIGH      = BIT7
SIGO      = BIT6
TXDATA    = BIT5
OCOR      = BIT4
ICIR      = BIT3
TESTMD    = BIT2
MCLK      = BIT1
DDCMP     = BIT0

*****
* AX0-15 - USYRT REG 0 (READ ONLY)
*****
RX7       = BIT7
RX6       = BIT6
RX5       = BIT5
RX4       = BIT4
RX3       = BIT3
RX2       = BIT2
RX1       = BIT1
RX0       = BIT0

*****
* AX0-16 - USYRT REG 1 (READ ONLY)
*****
RERR      = BIT7
ASBC2     = BIT6
ASBC1     = BIT5
ASBC0     = BIT4
ROR       = BIT3
```

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228RABT = BIT2
REOM = BIT1
RSOM = BIT0;*****
* AX1-15 - USYRT REG 2;*****
TX7 = BIT7
TX6 = BIT6
TX5 = BIT5
TX4 = BIT4
TX3 = BIT3
TX2 = BIT2
TX1 = BIT1
TX0 = BIT0;*****
* AX1-16 - USYRT REG 3;*****
TERR = BIT7
TXGA = BIT3
TXAB = BIT2
TEOM = BIT1
TSOM = BIT0;*****
* AX2-15 - USYRT REG 4;*****
SYN7 = BIT7
SYN6 = BIT6
SYN5 = BIT5
SYN4 = BIT4
SYN3 = BIT3
SYN2 = BIT2
SYN1 = BIT1
SYN0 = BIT0
SYNCH = 226;*****
* AX2-16 - USYRT REG 5;*****
APA = BIT7
DDC = BIT6
STR = BIT5
SEC = BIT4
IDL = BIT3
CRCTY2 = BIT2
CRCTY1 = BIT1
CRCTY0 = BIT0;*****
* AX3-15 - USYRT REG 6;*****
I422 = BIT7
XYZ = BIT6
C32BCC = BIT5
V35 = BIT4

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248

INTGRL = BIT3
C32ENB = BIT2
OP = BIT1
TEST = BIT0
AX315U = I422!XYZ!C32BCC!V35!INTGRL!OP

* AX3-16 - USYRT REG 7

TXLEN2 = BIT7
TXLEN1 = BIT6
TXLENO = BIT5
RXLEN2 = BIT2
RXLEN1 = BIT1
RXLENO = BIT0

8.0 TEST DESCRIPTIONS

TEST 1 - MICROPROCESSOR CSR ADDRESSING TEST (SELO)
*
* THIS TEST ADDRESSES THE FIRST MICROPROCESSOR CSR (SELO), TO MAKE SURE
* THAT A NON-EXISTENT MEMORY TIME-OUT TRAP DOES NOT OCCUR WHILE
* ATTEMPTING TO ADDRESS THE MICROPROCESSOR.

TEST 2 - INBUS/OUTBUS REG 14 INITIALIZATION TEST
*
* MASTER CLEAR (MCLR) IS SET IN THE MICROPROCESSOR, IBUS REG 14 IS READ
* AND COMPARED TO 200.

TEST 3 - INBUS/OUTBUS REG 14 READ/WRITE BIT TEST
*
* WRITE, READ, AND COMPARE ALL WORDS OF DATA PATTERN A INTO REG 14,
* A BYTE AT A TIME. NON-R/W BITS ARE MASKED OFF TO 0 BEFORE WRITING AND
* READING.
* DATA PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177.

TEST 4 - REG 14 MASTER CLEAR TEST
* WRITE 377 INTO REG 14, ISSUE MASTER CLEAR, READ REG 14 AND COMPARE
* TO 200.

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

```
;*****
TEST 5 - REG 14 UNIBUS RESET (INIT) TEST
* WRITE 377 INTO REG 14, ISSUE UNIBUS RESET (INIT), READ REG 14 AND COMPARE
* TO 200.
;*****

;*****
TEST 6 - LINE UNIT FALSE SELECTION TEST
*
* FIRST, A MASTER CLEAR IS PERFORMED. THEN, THE PROGRAM SINGLE-STEPS THE
* MICROPROCESSOR THROUGH AN INSTRUCTION WHICH LOADS 041 (OCT) INTO THE MAR
* REGISTER (OBUS* ADRS 14). THEN, THE LINE UNIT REGISTER 14 IS READ AND CHECKED
* TO BE UNAFFECTED (STILL = 0). THIS TEST IS INTENDED TO DETECT A FALSE
* SELECTION OF THE LINE UNIT REGISTERS, WHEN THE LINE UNIT IS NOT BEING
* ACCESSED.
;*****

;*****
TEST 7 - INBUS REG MASTER CLEAR TEST
*
* FIRST, ALL READ/WRITE BITS OF REGS 10-17 ARE SET BY LOADING A
* DIFFERENT WORD OF PATTERN G INTO EACH REG. THEN,
* A MASTER CLEAR IS ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF
* PATTERN M, WHICH CONTAINS THE INITIALIZED STATES OF THE REGS. (UNPREDICTABLE
* BITS ARE MASKED OFF TO 0 BEFORE COMPARISON).
* PATTERN G = 000,000,240,120,177,000,000,001
* PATTERN M = 000,020,000,000,200,000,000,051
;*****

;*****
TEST 8 - REGISTER 10-17 ADDRESSING TEST
*
* FIRST, A MASTER CLEAR IS ISSUED. THEN,
* WRITE A DIFFERENT WORD OF DATA PATTERN B INTO EACH OF REGS 10-17,
* AND AFTER EACH WRITE, READ AND COMPARE ALL REGS TO EXPECTED VALUES.
* UNPREDICTABLE BITS ARE MASKED OFF TO 0 WHEN READ FOR COMPARISON.
* PATTERN B = 000,000,040,100,220,000,000,051
;*****

;*****
TEST 9 - REG 11 READ/WRITE BIT TEST
```

```
115
116      * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN C INTO REG 11 :
117      * DATA PATTERN C = 020,020,020.
118      ;*****
119
120
121
122
123
124      ;*****
125      TEST 10 - REG 12 READ/WRITE BIT TEST
126      *
127      * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN D INTO REG 12 :
128      * DATA PATTERN D = 000,040,000.
129      ;*****
130
131
132
133
134
135      ;*****
136      TEST 11 - REG 13 READ/WRITE BIT TEST
137      *
138      * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN E INTO REG 13 :
139      * DATA PATTERN E = 000,120,020,100,120,000.
140      ;*****
141
142
143
144
145
146      ;*****
147      TEST 12 - REG 17 READ/WRITE BIT TEST
148      *
149      * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN F INTO REG 17 :
150      * DATA PATTERN F = 050,051,050.
151      ;*****
152
153
154
155
156
157      ;*****
158      TEST 13 - MAINTENANCE CLOCK BIT TEST
159      *
160      * FIRST, A MASTER CLEAR IS ISSUED TO INIT ALL REGS. THEN, THE MICROPROCESSOR
161      * IS PLACED IN A LOOP ON AN INSTRUCTION, BY SETTING THE INSTRUCTION IN SEL6
162      * AND SETTING ROM1 AND RUN IN BSEL1. THE INSTRUCTION IS ONE WHICH REPETITIVELY
163      * READS LINE UNIT REG 17 INTO BSEL2. THE PDP-11 CAN THEN SCAN BSEL2 TO MONITOR
164      * THE MAINTENANCE CLOCK BIT, MCLK. THE FOLLOWING SEQUENCE IS THEN PERFORMED
165      * TO MONITOR MCLK :
166      * - THE PROGRAM REPEATEDLY CHECKS THE MCLK BIT FOR THE 1 STATE, AND IF IT IS
167      * NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC (DEPENDING ON THE PROCESSOR)
168      * AN ERROR IS REPORTED. (THE MAINTENANCE CLOCK HAS A PERIOD OF 41.6 MICRO-
169      * SEC).
170      * - THE PROGRAM NEXT REPEATEDLY CHECKS THE MCLK BIT FOR THE 0 STATE, AND IF
171      * IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC AN ERROR IS REPORTED.
```

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228

* - THE PROGRAM NEXT REPEATEDLY CHECKS MCLK BIT FOR THE 1 STATE AGAIN, AND
* IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC, AN ERROR IS REPORTED.
*
* IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE M8207 RUN SWITCH (E28 SW7)
* IS OFF, THE TEST WILL BE SKIPPED.
;*****

;*****
TEST 14 - EXTENDED REGISTER MASTER CLEAR TEST
*
* FIRST, ALL READ/WRITE BITS OF EXTENDED REGS AX0-AX3 ARE SET BY LOADING
* A DIFFERENT WORD OF PATTERN H INTO EACH REG. THEN, A MASTER CLEAR IS
* ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF PATTERN I, WHICH
* CONTAINS THE INITIALIZED STATES OF ALL THE EXTENDED REGS.
* PATTERN H = 000,000,377,017,377,377,375,377
* PATTERN I = 000,000,000,000,000,103,000,000
;*****

;*****
TEST 15 - EXTENDED REGISTER ADDRESSING TEST
*
* FIRST, ISSUE A MASTER CLEAR TO PUT REGS INTO INITIALIZED STATES SHOWN IN
* PATTERN I. THEN, WRITE A DIFFERENT WORD OF PATTERN J INTO EACH EXTENDED (AX)
* REG, AND AFTER EACH WRITE, READ AND COMPARE ALL EXTENDED REGS TO EXPECTED
* VALUES.
* PATTERN I = 000,000,000,000,000,103,000,000
* PATTERN J = 000,000,001,002,004,103,040,100
;*****

;*****
TEST 16 - REGS 15,16 / AX2-15,AX2-16 READ/WRITE BIT TEST
*
* USING REGS 15,16, THE INDIRECT REGS AX2-15,AX2-16 (USYRT REGS 4,5) ARE
* WRITTEN AND READ USING EACH WORD OF PATTERN K. AX2-15 IS COMPARED
* TO THE WORD WRITTEN, AND AX2-16 IS ALWAYS COMPARED TO 103. (AX2-16 IS NOT
* WRITEABLE).
* PATTERN K =
* FOR REG 15: 000,377,125,252,001,002,004,010,020,040,100,200,000,000,
* 000,000,000,000,000,000,000,376,375,373,367,357,337,277,177,
* 377,377,377,377,377,377,377,377
* FOR REG 16: 000,377,125,252,000,000,000,000,000,000,000,000,001,002,
* 004,010,020,040,100,200,377,377,377,377,377,377,377,377,
* 376,375,373,367,357,337,277,177.
;*****

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

```
*****
TEST 20 - REG 17 - AX2-16 READ/WRITE, MASTER CLEAR TEST
*
* THIS TEST CONSISTS OF 2 SUBTESTS. IN THE FIRST SUBTEST, EACH BYTE OF PAT 0
* IS WRITTEN INTO REG 17 AND AFTER EACH WRITE, AX2-16 IS READ AND COMPARED
* TO A BYTE OF PAT P.
*   PATTERN 0 = 000,041,004,010,020,040,100,101,200,201,300,111,301,375
*   PATTERN P = 000,113,200,040,020,010,001,104,007,105,007,144,107,157
* IN THE SECOND SUBTEST, REG 17 IS LOADED WITH 375, A MASTER CLEAR IS ISSUED,
* AND AX2-16 IS COMPARED TO ITS INITIALIZED STATE (103).
*****
```

```
*****
TEST 21 - TRANSMITTER BUFFER DATA TEST
* A MASTER CLEAR IS DONE FIRST, AND THEN A BYTE OF PATTERN N IS LOADED INTO
* REG 11 AND THE NEXT BYTE IS LOADED TWICE INTO REG 10. THE PROGRAM THEN WAITS
* AT LEAST 50 MICRO-SEC, AND THEN IT READS AND COMPARES AX1-15 TO THE BYTE WHICH
* WAS LOADED INTO REG 10, AND IT READS AND COMPARES AX1-16 TO THE BYTE WHICH WAS
* LOADED INTO REG 11. THIS PROCESS IS REPEATED (INCLUDING THE MASTER CLEAR)
* FOR EACH PAIR OF BYTES IN PATTERN N.
*   PATTERN N =
*   FOR REG 10: 000,125,252,377,000,000,000
*   FOR REG 11: 000,000,000,000,005,012,017
*****
```

```
*****
TEST 22 - TRANSMITTER BUFFER SEQUENCING TEST
*
* FIRST, A MASTER CLEAR IS DONE, AND THE PROGRAM CHECKS FOR ORDY=1, OCOR=0.
* THEN, 2 TSOM CHARS ARE LOADED INTO THE TX SILO, AND ALLOWED TO RIPPLE
* DOWN TO THE OUTPUT. THE PROGRAM CHECKS FOR ORDY=1, OCOR=1.
* NEXT, THE PROGRAM CYCLES THE STEPLU BIT UNTIL OCOR=0 AGAIN, AND CHECKS FOR
* THIS TO OCCUR WITHIN 3 CYCLES.
* THE SILO IS THEN FILLED WITH 64 BYTES OF A 256-BYTE BINARY COUNT PATTERN
* (000-377) AND THE PROGRAM CHECKS FOR ORDY=0 AFTER THE 64TH CHAR IS LOADED.
* THE PROGRAM CYCLES STEPLU FOR 8 CYCLES AND CHECKS THAT AFTER THE 8TH, ORDY=1.
* AX1-15 IS READ AND COMPARED TO EXPECTED DATA.
* THE REST OF THE BINARY COUNT DATA BYTES ARE LOADED, CYCLED 8 CLOCKS, READ AND
* COMPARED, A BYTE AT A TIME. UPON COMPLETION, THE SILO IS CHECKED TO BE EMPTY
* WITH ORDY=1, OCOR=0.
*****
```

```
*****
TEST 23 - TX MSG TIMING TEST, CHAR MODE, WITH CMC
*****
```

343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-16.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING SYNCHS ARE SENT.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE AFTER THE 3RD SYNCH COMPLETES.
;*****

;*****
TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC
*
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* BIT ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-CCITT-1.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN BIT MODE.
* SOM IS SET TWICE TO SEND 2 FLAG CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING FLAGS ARE SENT.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE.
;*****

;*****
TEST 25 - TX MSG TIMING TEST, CHAR MODE, WITH NO CRC
*
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND NO ERROR CHECKING.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING SYNCHS ARE SENT.
* THE TEST IS PERFORMED WITH TXEN (REG 14, BIT6) SET, AND THE PROGRAM CHECKS
* THAT THIS HOLDS RTS HIGH PAST THE END OF THE MESSAGE.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE.
;*****

;*****

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

TEST 26 - TX UNDERRUN SET AND CLEAR TEST - CHAR MODE

- * IN THIS TEST, A TX UNDERRUN ERROR IS FORCED IN EACH OF 2 SITUATIONS,
- * AND THEN CLEARED DIFFERENTLY IN EACH.
- * IN THE FIRST, A MESSAGE IS INITIATED, A 000 CHAR IS SENT, AND THE TX
- * BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG,
- * WHICH CAUSES UNRR TO SET IN REG 11. THEN, SOM IS SET TO CLEAR THE ERROR,
- * AND THIS IS VERIFIED.
- * IN THE SECOND SITUATION, A MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX
- * BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH
- * AGAIN CAUSES UNRR TO SET. THEN, A MASTER CLEAR IS DONE, AND THE UNRR BIT
- * IS CHECKED TO BE CLEARED.

:*****

:*****
TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC

- * THE LINE UNIT IS PLACED IN CHAR MODE (DDCMP) AND A MESSAGE IS INITIATED
- * WITH AN 8-BIT SYNCH AND A 5-BIT SYNCH CHAR. NEXT, A 000 CHAR IS SENT WITH
- * EACH OF THE FOLLOWING TX CHAR LENGTHS : 5 BITS, 6 BITS, 7 BITS, 8 BITS.
- * (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED). TWO
- * TERMINATING SYNCHS ARE SENT AFTER THE DATA.

:*****

:*****
TEST 28 - TRANSMIT CHAR LENGTH TIMING TEST - BIT MODE, CRC

- * THE LINE UNIT IS PLACED IN BIT MODE AND A MESSAGE IS INITIATED
- * WITH 2 FLAG CHARS. NEXT, 2 8-BIT 000 CHARS ARE SENT, FOLLOWED BY 000 CHARS
- * WITH EACH OF THE FOLLOWING TRANSMITTER CHAR LENGTHS:
- * 1 BIT, 2 BITS, 3 BITS, 4 BITS, 5 BITS, 6 BITS, 7 BITS, AND 8 BITS.
- * (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED).
- * TWO TERMINATING FLAGS ARE SENT AFTER THE DATA.

:*****

:*****
TEST 29 - TXDATA BIT TEST - CHAR MODE, CRC

- * THE LINE UNIT IS INITIALIZED AND A MSG IS INITIATED (USING STEPLU) WITH CRC-
- * 16 SELECTED IN CHAR MODE. TWO SYNCHS, 000,125,252,377,000, AND 2 TERMINATING
- * SYNCHS ARE THEN SENT. THE PROGRAM CHECKS EACH BIT OF THE TRANSMITTED
- * DATA CHARS, BY MONITORING TXDATA (REG 17) AS THE DATA IS CLOCKED OUT OF
- * THE USYRT TRANSMITTER.

:*****

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

TEST 30 - USYRT RECEIVER MSG TEST - CHAR MODE, CRC

*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-16
* SELECTED. TWO SYNCHS, 000,125,252,377,000, AND FOUR TERMINATING SYNCHS ARE
* SENT. THE PROGRAM MONITORS IACT, AND THE RCV'D CHARS AND CRC BYTES ARE READ
* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
* STILL SET AFTER THE MESSAGE.

TEST 31 - USYRT RECEIVER MSG TEST - BIT MODE, CRC

*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-
* CCITT-1. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS ARE THEN
* SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE READ
* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.

TEST 32 - USYRT RECEIVER MSG TEST - CHAR MODE, NO CRC

*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH NO
* ERROR DETECTION. TWO SYNCHS, 000,125,252,377,000, AND TWO SYNCHS ARE
* THEN SENT. THE PROGRAM MONITORS IACT, AND THE RECEIVED CHARS ARE READ FROM
* AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
* STILL = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT = 0.

TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC

*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH ERROR
* DETECTION INHIBITED. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS
* ARE THEN SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE
* READ FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR

514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570

* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
;*****

;*****
TEST 34 - SILO-DISABLED TRANSMITTER LOAD TEST
*
* THIS TEST DISABLES THE SILOS, LOADS A 125 CHARACTER INTO THE TX SILO, AND
* READS AX1-15 AND CHECKS THAT THE DATA DID NOT GET LOADED INTO THE USYRT TX
* BUFFER.
;*****

;*****
TEST 35 - SILO-DISABLED MESSAGE TEST - BIT MODE, NO CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU)
* WITH LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, WITH SILO
* DISABLE SET, AND WITH NO ERROR DETECTION. TWO FLAGS, 000,125,252, AND
* TERMINATING FLAGS ARE THEN SENT BY LOADING THE TRANSMITTED CHARS INTO
* REG AX1. THE PROGRAM MONITORS OACT, IACT, RSOM, REOM, ORDY, OCUR, ICIR,
* IRDY, AND THE RECEIVED CHARS ARE READ FROM AX0 AND COMPARED TO EXPECTED
* VALUES.
;*****

;*****
TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC
*
* FIRST, A MASTER CLEAR IS DONE AND THE PROGRAM CHECKS FOR ICIR = 1 AND IRDY
* = 0. THEN, 2 SOM CHARS ARE LOADED AND CLOCKED INTO THE USYRT, AND 64
* BYTES OF A 256-BYTE BINARY COUNT DATA PATTERN (000-377) ARE LOADED INTO
* THE TX SILO.
* THE LINE UNIT IS THEN CLOCKED UNTIL IRDY = 1, AND THE PROGRAM CHECKS FOR
* THIS TO OCCUR WITHIN 40-43 CYCLES. THE PROGRAM READS THE RCV SILO, CHECKS THE
* CHAR FOR 000, AND CHECKS FOR IRDY = 0 AGAIN.
* THE LINE UNIT IS THEN CLOCKED IN GROUPS OF 8 CYCLES, AND AFTER EACH, THE
* PROGRAM CHECKS FOR ICIR = 1, IRDY = 1, UNTIL THE 64TH GROUP, AFTER WHICH
* IT CHECKS FOR ICIR = 0, IRDY = 1. THE SECOND DATA CHAR IS READ FROM THE
* RECEIVER SILO AND COMPARED TO 001. THEN, THE PROGRAM CHECKS FOR ICIR = 1,
* IRDY = 1 AGAIN.
* THE REST OF THE BINARY COUNT DATA BYTES ARE CYCLED 8 CLOCKS AND READ AND
* COMPARED A BYTE AT A TIME.
;*****

571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627

TEST 37 - RECEIVER CHAR LENGTH TIMING TEST - CHAR MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN CHAR MODE, WITH NO ERROR DETECTION, AND A MSG IS
* INITIATED WITH 2 SYNCH CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOK
* SET, WHILE THE RECEIVER CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 5,6,7,8.
* FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THET USYRT RECEIVER
* FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV CHAR LENGTH.
* (FOR EXAMPLE A 5-BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED). A MASTER
* CLEAR IS THEN DONE TO TERMINATE THE OPERATION.

TEST 38 - RECEIVER CHAR LENGTH TIMING TEST - BIT MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN BIT MODE WITH NO ERROR DETECTION, AND A MESSAGE IS
* INITIATED WITH 2 FLAG CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOK
* SET, WHILE THE RCV CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 8,8,8,7,6,5,
* 4,3,2,1. FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THAT THE
* USYRT RECEIVER FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV
* CHAR LENGTH. (FOR EXAMPLE, A 5 BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED).
* A MASTER CLEAR IS THEN DONE TO TERMINATE THE OPERATION.

TEST 39 - TRANSMITTER UNDERRUN ERROR, IDLE MARKING, CHAR MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN CHAR MODE, AND THE IDLE BIT IS SET. THEN, A
* MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX BUFFER IS NOT SERVICED
* IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH CAUSES A TX UNDERRUN
* ERROR. THEN, THE RECEIVER IS CLOCKED AND CHECKED FOR TWO 377 CHARS TO BE
* RECEIVED (LINE MARKING) BEFORE SHUTTING DOWN WITH A MASTER CLEAR.

TEST 40 - MSG TERMINATION WITH GA CHARS - BIT MODE, NO CRC
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN BIT MODE.
* 2 FLAG CHARACTERS ARE SENT, FOLLOWED BY
* THE FOLLOWING DATA CHARACTERS : 000, 125, 252, 377, 000. THEN THE LOOP
* MODE BIT (STRIP) IS SET AND 2 TERMINATING GO-AHEAD CHARACTERS ARE

628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684

* SENT. EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 5 DATA
* WORDS ARE READ AND COMPARED TO EXPECTED VALUES.
* ALSO, THE FIRST GA CHAR IS CHECKED BY SCANNING THE TXDATA BIT AS THE GA
* IS BEING TRANSMITTED (GA = 376 OCTAL).
* THE TEST ALSO CHECKS FOR SETTING OF RAB AND EBLK
* IN LOOP MODE.

TEST 41 - IDLE SYNCHS TEST - CHAR MODE

*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN CHAR MODE. 24(DEC) SYNCHS ARE SENT.
* EACH SYNCH IS TIMED AS IT IS RECEIVED, AND THE BITS ARE CHECKED
* FOR A VALID SYNCH CHAR FOR EACH OF THE 22 SYNCHS WHICH FOLLOW
* THE FIRST TWO (THESE PERFORM SYNCHRONIZATION, AND ARE NOT READ).
* WHILE THE LAST SYNCH IS BEING TRANSMITTED, OC IS SET, AND THE
* NEXT CHAR RCV'D AFTER THE SYNCH IS CHECKED TO BE 377 (LINE MARKING).
* THEN, A MASTER CLEAR IS ISSUED.
* THE SYNCH CHAR USED IS 226 (OCTAL).

TEST 42 - STRIP SYNCH TEST

*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN CHAR MODE AND WITH THE STRIP SYNCH
* BIT SET. THEN 24 (DEC) SYNCHS ARE SENT
* FOLLOWED BY THE FOLLOWING DATA CHARACTERS : 377, 000, 125, 252,
* AND THEN 2 TERMINATING SYNCHS.
* EACH OF THE 23 SYNCHS AFTER THE FIRST ARE CHECKED AT THE TRANSMITTER OUTPUT,
* BY SCANNING THE TXDATA BIT.
* EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 4 DATA WORDS
* ARE READ AND COMPARED TO EXPECTED VALUES.
* FINALLY, THE LINE UNIT IS CLOCKED FOR SEVERAL CHAR TIMES, AND A CHECK
* IS MADE FOR OACT = 0 (TEOM SHOULD CAUSE TX ENABLE TO DROP).
* THE ABOVE TEST IS REPEATED FOR EACH OF THE FOLLOWING SYNCH CHAR DATA
* PATTERNS : 226,000,125,252,376,177.

8.1 DATA PATTERNS USED

***** DATA PATTERN A *****

```
685          PATA:
686              .BYTE 125
687              .BYTE 252
688              .BYTE 000
689              .BYTE 377
690              .BYTE 001
691              .BYTE 002
692              .BYTE 004
693              .BYTE 010
694              .BYTE 020
695              .BYTE 040
696              .BYTE 100
697              .BYTE 200
698              .BYTE 376
699              .BYTE 375
700              .BYTE 373
701              .BYTE 367
702              .BYTE 357
703              .BYTE 337
704              .BYTE 277
705              .BYTE 177
706
707          ***** DATA PATTERN B *****
708          PATB:
709              .BYTE 000
710              .BYTE 000
711              .BYTE 040
712              .BYTE 100
713              .BYTE 220
714              .BYTE 000
715              .BYTE 000
716              .BYTE 051
717
718          ***** DATA PATTERN C *****
719          PATC:
720              .BYTE 020
721              .BYTE 020
722              .BYTE 020
723
724          ***** DATA PATTERN D *****
725          PATD:
726              .BYTE 000
727              .BYTE 040
728              .BYTE 000
729
730          ***** DATA PATTERN E *****
731          PATE:
732              .BYTE 000
733              .BYTE 120
734              .BYTE 020
735              .BYTE 100
736              .BYTE 120
737              .BYTE 000
738
739          ***** DATA PATTERN F *****
740          PATF:
741              .BYTE 050
```

```
742          .BYTE 051
743          .BYTE 050
744
745          ***** DATA PATTERN G *****
746          PATG:
747              .BYTE 000
748              .BYTE 000
749              .BYTE 240
750              .BYTE 120
751              .BYTE 177
752              .BYTE 000
753              .BYTE 000
754              .BYTE 001
755
756          ***** DATA PATTERN H *****
757          PATH:
758              .BYTE 000
759              .BYTE 000
760              .BYTE 377
761              .BYTE 017
762              .BYTE 377
763              .BYTE 377
764              .BYTE 375
765              .BYTE 377
766
767          ***** DATA PATTERN I *****
768          PATI:
769              .BYTE 000
770              .BYTE 000
771              .BYTE 000
772              .BYTE 000
773              .BYTE 000
774              .BYTE 103
775              .BYTE 000
776              .BYTE 000
777
778          ***** DATA PATTERN J *****
779          PATJ:
780              .BYTE 000
781              .BYTE 000
782              .BYTE 010
783              .BYTE 002
784              .BYTE 004
785              .BYTE 103
786              .BYTE 001
787              .BYTE 100
788
789          ***** DATA PATTERN K *****
790          PATK:
791              .BYTE 000
792              .BYTE 000
793              .BYTE 377
794              .BYTE 377
795              .BYTE 125
796              .BYTE 125
797              .BYTE 252
798              .BYTE 252
799              .BYTE 001
```

799	.BYTE	000
800	.BYTE	002
801	.BYTE	000
802	.BYTE	004
803	.BYTE	000
804	.BYTE	010
805	.BYTE	000
806	.BYTE	020
807	.BYTE	000
808	.BYTE	040
809	.BYTE	000
810	.BYTE	100
811	.BYTE	000
812	.BYTE	200
813	.BYTE	000
814	.BYTE	000
815	.BYTE	001
816	.BYTE	000
817	.BYTE	002
818	.BYTE	000
819	.BYTE	004
820	.BYTE	000
821	.BYTE	010
822	.BYTE	000
823	.BYTE	020
824	.BYTE	000
825	.BYTE	040
826	.BYTE	000
827	.BYTE	100
828	.BYTE	000
829	.BYTE	200
830	.BYTE	376
831	.BYTE	377
832	.BYTE	375
833	.BYTE	377
834	.BYTE	373
835	.BYTE	377
836	.BYTE	367
837	.BYTE	377
838	.BYTE	357
839	.BYTE	377
840	.BYTE	337
841	.BYTE	377
842	.BYTE	277
843	.BYTE	377
844	.BYTE	177
845	.BYTE	377
846	.BYTE	377
847	.BYTE	376
848	.BYTE	377
849	.BYTE	375
850	.BYTE	377
851	.BYTE	373
852	.BYTE	377
853	.BYTE	367
854	.BYTE	377
855	.BYTE	357

856	.BYTE	377
857	.BYTE	337
858	.BYTE	377
859	.BYTE	277
860	.BYTE	377
861	.BYTE	177
862		
863	***** DATA PATTERN L *****	
864	PATL:	
865	.BYTE	000
866	.BYTE	000
867	.BYTE	377
868	.BYTE	377
869	.BYTE	000
870	.BYTE	000
871		
872	***** DATA PATTERN M *****	
873	PATM:	
874	.BYTE	000
875	.BYTE	020
876	.BYTE	000
877	.BYTE	000
878	.BYTE	200
879	.BYTE	000
880	.BYTE	000
881	.BYTE	051
882		
883	***** DATA PATTERN N *****	
884	PATN:	
885	.BYTE	000
886	.BYTE	000
887	.BYTE	000
888	.BYTE	125
889	.BYTE	000
890	.BYTE	252
891	.BYTE	000
892	.BYTE	377
893	.BYTE	005
894	.BYTE	000
895	.BYTE	012
896	.BYTE	000
897	.BYTE	017
898	.BYTE	000
899		
900	***** DATA PATTERN O *****	
901	PATO:	
902	.BYTE	000
903	.BYTE	041
904	.BYTE	004
905	.BYTE	010
906	.BYTE	020
907	.BYTE	040
908	.BYTE	100
909	.BYTE	101
910	.BYTE	200
911	.BYTE	201
912	.BYTE	300

913	.BYTE	111
914	.BYTE	301
915	.BYTE	375

***** DATA PATTERN P *****

PATP:

.BYTE	000
.BYTE	113
.BYTE	200
.BYTE	040
.BYTE	020
.BYTE	010
.BYTE	001
.BYTE	104
.BYTE	007
.BYTE	105
.BYTE	007
.BYTE	144
.BYTE	107
.BYTE	157

***** DATA PATTERN U *****

PATU:

.BYTE	000
.BYTE	000
.BYTE	001
.BYTE	000
.BYTE	013
.BYTE	000
.BYTE	011
.BYTE	000
.BYTE	021
.BYTE	000
.BYTE	101
.BYTE	000
.BYTE	301
.BYTE	000
.BYTE	000
.BYTE	001
.BYTE	000
.BYTE	002
.BYTE	000
.BYTE	004
.BYTE	000
.BYTE	040
.BYTE	000
.BYTE	100
.BYTE	000
.BYTE	200
.BYTE	000
.BYTE	346
.BYTE	000
.BYTE	345
.BYTE	000
.BYTE	343
.BYTE	000
.BYTE	307
.BYTE	000

916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969

970	.BYTE	247
971	.BYTE	000
972	.BYTE	147

***** DATA PATTERN V *****

PATV:	.BYTE	000
	.BYTE	000
	.BYTE	333
	.BYTE	000
	.BYTE	331
	.BYTE	000
	.BYTE	323
	.BYTE	000
	.BYTE	313
	.BYTE	000
	.BYTE	233
	.BYTE	000
	.BYTE	133
	.BYTE	000
	.BYTE	000
	.BYTE	001
	.BYTE	000
	.BYTE	002
	.BYTE	000
	.BYTE	004
	.BYTE	000
	.BYTE	040
	.BYTE	000
	.BYTE	100
	.BYTE	000
	.BYTE	200
	.BYTE	000
	.BYTE	346
	.BYTE	000
	.BYTE	345
	.BYTE	000
	.BYTE	343
	.BYTE	000
	.BYTE	307
	.BYTE	000
	.BYTE	247
	.BYTE	000
	.BYTE	147

973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES AN "IRDY NOT SET" ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE PC OF THE CALL TO THE SUBROUTINE REPORTING IT, THE FAILING REGISTER NAME, AND DEVICE REGISTER CONTENTS :

CZDMR DVC FTL ERR 00017 ON UNIT 00 TST 034 SUB 000 PC: 006210
IRDY NOT SET
PC OF SUBR CALL: 030044
DEVICE CSR ADDRESS : 160170

FAILING REG: INBUS/OUTBUS REG 12

LINE UNIT INBUS REGS:
REG10 REG11 REG12 REG13
000 120 000 257
REG14 REG15 REG16 REG17
024 377 377 035

LINE UNIT EXTENDED REGS:
AX0-15 AX0-16 AX1-15 AX1-16
000 000 000 000
AX2-15 AX2-16 AX3-15 AX3-16
000 000 000 000

FOR OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE, AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

IF EXTENDED ERROR INFORMATION HAD BEEN INHIBITED USING THE IXE FLAG PRIOR TO RUNNING THE TEST, THE ABOVE ERROR WOULD HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :

CZDMR DVC FTL ERR 00017 ON UNIT 00 TST 034 SUB 000 PC:006210
IRDY NOT SET
PC OF SUBR CALL: 030044
DEVICE CSR ADDRESS : 160170

FAILING REG: INBUS/OUTBUS REG 12

58
59
60
61
62
63
64
65

a

```
1          .TITLE CZDMRD M8203 STATIC DIAG #1
10         .=2000
11
12
13
14
15
16
17         .MCALL SVC
18 002000   SVC                      ; INITIALIZE SUPERVISOR MACROS
19
20
21
22
23
24 002000   BGNMOD LU1MOD
25
26
27         000001 $LSTIN= 1
28         000001 $LSTTAG= 1
29         000001 SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
30         000001 SVCTST= 1     ; LIST TEST TAGS, SHIFTED RIGHT
31         000001 SVCSUB= 1     ; LIST SUBTEST TAGS, SHIFTED RIGHT
32         000001 SVCGBL= 1     ; LIST GLOBAL TAGS, SHIFTED RIGHT
33         000001 SVCTAG= 1     ; LIST OTHER TAGS, SHIFTED RIGHT
34
35         ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
36         ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
37         ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
38         ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
39
40
```

.SBTTL PROGRAM HEADER

:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--

POINTER BGNAU,BGNDU

:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
: IF ANY OPTIONAL POINTERS ARE TO BE USED IN THE 'HEADER', CHANGE
: 'POINTER' TO CONTAIN THE CORRECT ARGUMENTS. IF ALL OPTIONAL
: POINTERS ARE TO BE USED, CHANGE 'POINTER' TO BE 'POINTER ALL'.
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

HEADER CZDMR,D,0,45.,0

LSNAME::

.ASCII /C/
.ASCII /Z/
.ASCII /D/
.ASCII /M/
.ASCII /R/
.BYTE 0
.BYTE 0
.BYTE 0

LSREV::

.ASCII /D/

LSDEPO::

.ASCII /O/

LSUNIT::

.WORD 0

LSTIML::

.WORD 45.

LSHPCP::

.WORD LSHARD

LSSPCP::

.WORD 0

LSHPTP::

.WORD LSHW

LSSPTP::

.WORD 0

LSLADP::

.WORD LSLAST

LSSTA::

.WORD 0

LSCO::

.WORD 0

LSDTYP::

.WORD 0

LSAPT::

.WORD 0

LSDTP::

.WORD LSDISPATCH

LSPRIO::

.WORD 0

LSENV1::

.WORD 0

LSEXP1::

1
2
3
4
5
6
7 002000
8
10
11
12
13
14
16
17 002000
C02000
002000 103
002001 132
002002 104
002003 115
002004 122
002005 000
002006 000
002007 000
002010
002010 104
002011
002011 060
002012
002012 000000
002014
002014 000055
002016
002016 035554
002020
002020 000000
002022
002022 002252
002024
002024 000000
002026
002026 036234
002030
002030 000000
002032
002032 000000
002034
002034 000000
002036
002036 000000
002040
002040 002124
002042
002042 000000
002044
002044 000000
002046

002046 000000
002050
002050 003
002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003462
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 021632
002072
002072 021550
002074
002074 000000
002076
002076 003470
002100
002100 104035
002102
002102 000000
002104
002104 021116
002106
002106 021546
002110
002110 021466
002112
002112 021110
002114
002114 000000
002116
002116 000000
002120
002120 000000

LSMREV:: .WORD 0
 .BYTE CSREVISION
 .BYTE CSREDIT
LSEF:: .WORD 0
 .WORD 0
LSSPC:: .WORD 0
LSDEVP:: .WORD LSDVTYP
LSREPP:: .WORD 0
LSEXP4:: .WORD 0
LSEXP5:: .WORD 0
LSAUT:: .WORD LSAU
LSDUT:: .WORD LSDU
LSLUN:: .WORD 0
LSDESP:: .WORD LSDESC
LSLOAD:: EMT ESLOAD
LSETP:: .WORD 0
LSICP:: .WORD LSINIT
LSCCP:: .WORD LSCLEAN
LSACP:: .WORD LSAUTO
LSPRT:: .WORD LSPROT
LSTEST:: .WORD 0
LSDLY:: .WORD 0
LSHIME:: .WORD 0

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CHANGE THE 'HEADER' TO CONTAIN THE PROPER ARGUMENTS.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

.EVEN

18
20
21
22
24
25
26
27
28
29
30
31

.SBTTL DISPATCH TABLE

```

: // THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
: // IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
:

```

DISPATCH 42

```

        .WORD      42
LSDISPATCH::
        .WORD      T1
        .WORD      T2
        .WORD      T3
        .WORD      T4
        .WORD      T5
        .WORD      T6
        .WORD      T7
        .WORD      T8
        .WORD      T9
        .WORD      T10
        .WORD      T11
        .WORD      T12
        .WORD      T13
        .WORD      T14
        .WORD      T15
        .WORD      T16
        .WORD      T17
        .WORD      T18
        .WORD      T19
        .WORD      T20
        .WORD      T21
        .WORD      T22
        .WORD      T23
        .WORD      T24
        .WORD      T25
        .WORD      T26
        .WORD      T27
        .WORD      T28
        .WORD      T29
        .WORD      T30
        .WORD      T31
        .WORD      T32
        .WORD      T33
        .WORD      T34
        .WORD      T35
        .WORD      T36
        .WORD      T37
        .WORD      T38
        .WORD      T39
        .WORD      T40
        .WORD      T41
        .WORD      T42

```

[illegible]

CZDMRD M8203 STATIC DIAG #1
DISPATCH TABLE

MACRO V03.01 7-AUG-80 11:01:24 PAGE 9-1

SEQ 45

16
17
18
19
20

```
1      .SBTTL  DEFAULT HARDWARE P-TABLE
2
3      ;////////////////////////////////////
4      ;// THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
5      ;// THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
6      ;// IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
7      ;////////////////////////////////////
8
9      002250      BGNHW  DFPTBL
10     002250      000013
11     002252
12     002252
13
14     002252      000007      .WORD      7      ;MICROPROCESSOR TYPE = M8207
15     002254      160170      .WORD      160170    ;M8207 CSR UNIBUS ADDRESS
16     002256      000300      .WORD      300      ;M8207 INTERRUPT VECTOR
17     002260      005000      .WORD      5000     ;M8207 INTERRUPT PRIORITY LEVEL = 5
18     002262      000003      .WORD      3        ;LINE UNIT = M8203
19     002264      000000      .WORD      000      ;M8203 REG 11 (E121 SW10,9 , E134 SW9,10)
20     002266      000000      .WORD      000      ;M8203 REG 15 (E134 SW1-8)
21     002270      000000      .WORD      000      ;M8203 REG 16 (E121 SW1-8)
22     002272      000000      .WORD      0        ;H3254&H3255 USED
23     002274      000004      .WORD      4        ;BAUD RATE = 56 K
24     002276      000001      .WORD      1        ;M8207 RUN SWITCH (E28 SW7) IS ON
25
26     002300      ENDPHW
27
28     002300
```

.WORD L10000-LSHW/2
LSHW::
DFPTBL::

L10000:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

.SBTTL SOFTWARE P-TABLE

:/
:/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
:/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
:/

002300
002300 000000
002302
002302

BGNSW SFPTBL

.WORD L10001-LSSW/2
LSSW::
SFPTBL::

002302
002302

ENDSW

L10001:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 002302

.SBTTL GLOBAL EQUATES SECTION

```
:/
:/ THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
:/ ARE USED IN MORE THAN ONE TEST.
:/
```

EQUALS

: BIT DEFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

: EVENT FLAG DEFINITIONS

: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	: START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED

000035
000034EF.NEW== 29.
EF.PWR== 28.; A NEW PASS HAS BEEN STARTED
; A POWER-FAIL/POWER-UP OCCURRED

: PRIORITY LEVEL DEFINITIONS

000340
000300
000240
000200
000140
000100
000040
000000PRI07== 340
PRI06== 300
PRI05== 240
PRI04== 200
PRI03== 140
PRI02== 100
PRI01== 40
PRI00== 0

: OPERATOR FLAG BITS

000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000EVL== 4
LOT== 10
ADR== 20
IDU== 40
ISR== 100
UAM== 200
BOE== 400
PNT== 1000
PRI== 2000
IXE== 4000
IBE== 10000
IER== 20000
LOE== 40000
HOE== 100000: *****
: * PROGRAM EVENT FLAG DEFINITIONS
: *****: *****
: * MAINTENANCE REGISTER - BSEL1
: *****000200
000100
000020
000010
000004
000002
000001RUN = BIT7
MCLR = BIT6
STEPLU = BIT4
LULOOB = BIT3
ROMO = BIT2
ROMI = BIT1
STEPMP = BIT0: *****
: * OBUS REG 10 - TRANSMITTER BUFFER
: *****21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

47      000200      TX7      = BIT7
48      000100      TX6      = BIT6
49      000040      TX5      = BIT5
50      000020      TX4      = BIT4
51      000010      TX3      = BIT3
52      000004      TX2      = BIT2
53      000002      TX1      = BIT1
54      000001      TX0      = BIT0
55
56      ;*****
57      ;* OBUS REG 11
58      ;*****
59      000200      OC        = BIT7
60      000010      GOAH      = BIT3
61      000004      ABORT     = BIT2
62      000002      EOM       = BIT1
63      000001      SOM       = BIT0
64
65      ;*****
66      ;* OBUS REG 12
67      ;*****
68      000200      IC        = BIT7
69      000100      BPOLL     = BIT6
70      000040      LULP      = BIT5
71
72      ;*****
73      ;* OBUS REG 13
74      ;*****
75      000200      POLL      = BIT7
76      000100      DTR       = BIT6
77      000040      SELFR     = BIT5
78      000020      HDX       = BIT4
79      000010      MAINT1    = BIT3
80      000004      MAINT2    = BIT2
81      000002      SELSBY    = BIT1
82
83      ;*****
84      ;* OBUS REG 14
85      ;*****
86      000100      TXEN      = BIT6
87      000040      DISSI     = BIT5
88      000020      RDAX      = BIT4
89      000010      WAX       = BIT3
90      000004      ENAX      = BIT2
91      000002      AX2       = BIT1
92      000001      AX1       = BIT0
93
94      ;*****
95      ;* OBUS REG 17
96      ;*****
97      000200      CRC2      = BIT7
98      000100      CRC1      = BIT6
99      000040      IDLE      = BIT5
100     000020      SECA      = BIT4
101     000010      STRIP     = BIT3
102     000004      RDALL     = BIT2
103     000002      IERR      = BIT1
```

```
104          000001          DDCMP   = BIT0
105
106          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
107          : * IBUS REG 10 - RECEIVER BUFFER
108          : ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
109          000200          RX7     = BIT7
110          000100          RX6     = BIT6
111          000040          RX5     = BIT5
112          000020          RX4     = BIT4
113          000010          RX3     = BIT3
114          000004          RX2     = BIT2
115          000002          RX1     = BIT1
116          000001          RX0     = BIT0
117
118          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
119          : * IBUS REG 11
120          : ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
121          000200          OC       = BIT7
122          000100          OACT    = BIT6
123          000040          SW3     = BIT5
124          000020          ORDY    = BIT4
125          000010          SW2     = BIT3
126          000004          SW1     = BIT2
127          000002          SW0     = BIT1
128          000001          UNRR    = BIT0
129
130          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
131          : * IBUS REG 12
132          : ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
133          000200          IC       = BIT7
134          000100          IACT    = BIT6
135          000040          LULP    = BIT5
136          000020          IRDY    = BIT4
137          000010          OVR    = BIT3
138          000004          RAB     = BIT2
139          000002          EBLK    = BIT1
140          000001          BCC     = BIT0
141
142          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
143          : * IBUS REG 13
144          : ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
145          000200          RING     = BIT7
146          000100          DTR     = BIT6
147          000040          RTS     = BIT5
148          000020          HDX     = BIT4
149          000010          MODR    = BIT3
150          000004          CS      = BIT2
151          000002          STBY    = BIT1
152          000001          CARR    = BIT0
153
154          ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
155          : * IBUS REG 14
156          : ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
157          000200          READY    = BIT7
158          000100          TXEN     = BIT6
159          000040          DISSI    = BIT5
160          000020          RDAX     = BIT4
```

```
161      000010      WAX      = BIT3
162      000004      ENAX     = BIT2
163      000002      AX2      = BIT1
164      000001      AX1      = BIT0
165
166      ;*****
167      ;* IBUS REG 17
168      ;*****
169      000200      SIGH      = BIT7
170      000100      SIGQ     = BIT6
171      000040      TXDATA   = BIT5
172      000020      CCOR     = BIT4
173      000010      ICIR     = BIT3
174      000004      TESTMD   = BIT2
175      000002      MCLK     = BIT1
176      000001      DDCMP    = BIT0
177
178      ;*****
179      ;* AX0-15 - USYRT REG 0 (READ ONLY)
180      ;*****
181      000200      RX7       = BIT7
182      000100      RX6       = BIT6
183      000040      RX5       = BIT5
184      000020      RX4       = BIT4
185      000010      RX3       = BIT3
186      000004      RX2       = BIT2
187      000002      RX1       = BIT1
188      000001      RX0       = BIT0
189
190      ;*****
191      ;* AX0-16 - USYRT REG 1 (READ ONLY)
192      ;*****
193      000200      RERR      = BIT7
194      000100      ASBC2     = BIT6
195      000040      ASBC1     = BIT5
196      000020      ASBC0     = BIT4
197      000010      ROR       = BIT3
198      000004      RABT      = BIT2
199      000002      REOM      = BIT1
200      000001      RSOM      = BIT0
201
202      ;*****
203      ;* AX1-15 - USYRT REG 2
204      ;*****
205      000200      TX7       = BIT7
206      000100      TX6       = BIT6
207      000040      TX5       = BIT5
208      000020      TX4       = BIT4
209      000010      TX3       = BIT3
210      000004      TX2       = BIT2
211      000002      TX1       = BIT1
212      000001      TX0       = BIT0
213
214      ;*****
215      ;* AX1-16 - USYRT REG 3
216      ;*****
217      000200      TERR      = BIT7
```

```
218      000010      TXGA      = BIT3
219      000004      TXAB      = BIT2
220      000002      TEOM      = BIT1
221      000001      TSOM      = BIT0
222
223      ;*****
224      ;* AX2-15 - USYRT REG 4
225      ;*****
226      000200      SYN7      = BIT7
227      000100      SYN6      = BIT6
228      000040      SYN5      = BIT5
229      000020      SYN4      = BIT4
230      000010      SYN3      = BIT3
231      000004      SYN2      = BIT2
232      000002      SYN1      = BIT1
233      000001      SYN0      = BIT0
234      000226      SYNCH     = 226
235
236      ;*****
237      ;* AX2-16 - USYRT REG 5
238      ;*****
239      000200      APA       = BIT7
240      000100      DDC       = BIT6
241      000040      STR       = BIT5
242      000020      SEC       = BIT4
243      000010      IDL       = BIT3
244      000004      CRCTY2    = BIT2
245      000002      CRCTY1    = BIT1
246      000001      CRCTY0    = BIT0
247
248      ;*****
249      ;* AX3-15 - USYRT REG 6
250      ;*****
251      000200      I422      = BIT7
252      000100      XYZ       = BIT6
253      000040      C32BCC    = BIT5
254      000020      V35       = BIT4
255      000010      INTGRL    = BIT3
256      000004      C32ENB    = BIT2
257      000002      OP        = BIT1
258      000001      TEST      = BIT0
259      000372      AX315U    = I422!XYZ!C32BCC!V35!INTGRL!OP
260
261      ;*****
262      ;* AX3-16 - USYRT REG 7
263      ;*****
264      000200      TXLEN2     = BIT7
265      000100      TXLEN1     = BIT6
266      000040      TXLEN0     = BIT5
267      000004      RXLEN2     = BIT2
268      000002      RXLEN1     = BIT1
269      000001      RXLEN0     = BIT0
270
271
272
273
274
```

```
275 ;*****
276 ;* TX CONTROL BITS DEFINED ON WORD BASIS
277 ;*****
278 004000 TXGOA = BIT11
279 002000 TXABT = BIT10
280 001000 TXEOM = BIT9
281 000400 TXSOM = BIT8
282
283
284
285
286
287 ;*****
288 ;* RCV CONTROL BITS DEFINED ON WORD BASIS
289 ;*****
290 004000 RXOVR = BIT11
291 002000 RXABT = BIT10
292 001000 RXEBL = BIT9
293 000400 RXBCC = BIT8
294
295
296
297
298 ;*****
299 ;* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)
300 ;*****
301 002302 LUR10 = LUREG+0 ;LINE UNIT IBUS REG 10
302 002304 LUR11 = LUREG+2 ;LINE UNIT IBUS REG 11
303 002306 LUR12 = LUREG+4 ;LINE UNIT IBUS REG 12
304 002310 LUR13 = LUREG+6 ;LINE UNIT IBUS REG 13
305 002312 LUR14 = LUREG+10 ;LINE UNIT IBUS REG 14
306 002314 LUR15 = LUREG+12 ;LINE UNIT IBUS REG 15
307 002316 LUR16 = LUREG+14 ;LINE UNIT IBUS REG 16
308 002320 LUR17 = LUREG+16 ;LINE UNIT IBUS REG 17
309 002322 AX0.15 = LUREG+20 ;USYRT REG 0
310 002324 AX0.16 = LUREG+22 ;USYRT REG 1
311 002326 AX1.15 = LUREG+24 ;USYRT REG 2
312 002330 AX1.16 = LUREG+26 ;USYRT REG 3
313 002332 AX2.15 = LUREG+30 ;USYRT REG 4
314 002334 AX2.16 = LUREG+32 ;USYRT REG 5
315 002336 AX3.15 = LUREG+34 ;USYRT REG 6
316 002340 AX3.16 = LUREG+36 ;USYRT REG 7
317
318
319
320
321
322 100000 CHPCHK = BIT15
323
324 100000 BCCCHK = BIT15
325 100000 CRCCHK = BIT15
326
327
328
329
330
331 ;*****
```

```
332      ;* MICROINSTRUCTION DEFINITIONS
333      ;*****
334      021000      MVI0X  = 021000      ;MOVE IBUS TO OBUS*
335      122000      MVIX0  = 122000      ;MOVE IBUS* TO OBUS
336      121000      MVIXOX = 121000      ;MOVE IBUS* TO OBUS*
337
338
339
340      ;***** ERROR1 BIT FLAG DEFINITIONS *****
341      000001      RRDYTO = BIT0
342      000002      WRDYTO = BIT1
343
344
345
346
347
```

```
1      .SBTTL  GLOBAL DATA SECTION
2
3      ;////////////////////////////////////
4      ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
5      ;/      IN MORE THAN ONE TEST.
6      ;////////////////////////////////////
7
8      ;*****
9      ;* STORAGE FOR DEVICE REGISTERS
10     ;*****
11     002302  LUREG:  .BLKW  16.
12
13     ;*****
14     ;* MISCELLANEOUS STORAGE
15     ;*****
16     002342  000000  SCRACH:  .WORD  0      ;GEN'L PURPOSE SCRATCH WORD
17     002344  000000  LOGDEV:  .WORD  0      ;LOGICAL DEVICE NUMBER
18     002346  000000  PSTACK:  .WORD  0      ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
19     002350  000000  PRIOR:   .WORD  0      ;CPU PRIORITY FOR PRINTOUT
20     002352  000000  SUBRPC:  .WORD  0      ;PC OF SUBR CALL FOR ERROR REPORTS
21     002354  000000  INTFLG:  .WORD  0      ;INTERRUPT RECEIVED FLAGS
22                                     ; BIT 0 FOR TX, BIT 1 FOR RCV
23     002356  000000  ERRFLG:  .WORD  0      ;SUBROUTINE ERROR FLAG
24     002360  000000  TIMFLG:  .WORD  0      ;EVENT TIME-OUT FLAG
25     002362  000000  RETADR:  .WORD  0      ;SUBR ERROR RETURN ADDRESS
26     002364  000000  REDBYT:  .WORD  0      ;LO BYTE CONTAINS BYTE READ FROM LU REG
27     002366  000000  WRIBYT:  .WORD  0      ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
28     002370  000000  RAX15:   .WORD  0      ;LO BYTE CONTAINS BYTE READ FROM REG 15
29     002372  000000  RAX16:   .WORD  0      ;LO BYTE CONTAINS BYTE READ FROM REG 16
30     002374  000000  WAX15:   .WORD  0      ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
31     002376  000000  WAX16:   .WORD  0      ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
32     002400  000000  REGNUM:  .WORD  0      ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
33     002402  000000  AXNUM:   .WORD  0      ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
34     002404  000000  GOODAT:  .WORD  0      ;STORAGE FOR EXPECTED DATA
35     002406  000000  BADDAT:  .WORD  0      ;STORAGE FOR ACTUAL DATA
36     002410  000000  LOADAT:  .WORD  0      ;CONTAINS TEST DATA LOADED INTO REG
37     002412  000000  FRSTIM:  .WORD  0      ;FLAG=0 IF PROGRAM JUST LOADED
38     002414  000000  SAVE4:   .WORD  0      ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
39     002416  000000  SAVE6:   .WORD  0      ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
40     002420  000000  ERROR1:  .WORD  0      ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
41     002422  000000  TXWORD:  .WORD  0      ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
42     002424  000000  RXWORD:  .WORD  0      ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
43     002426  000000  DISILO:  .WORD  0      ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
44     002430  000000  CHPTYP:  .WORD  0      ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
45     002432  000000  SAVLEN:  .WORD  0      ;SAVED TX AND RCV CHAR LENGTHS
46     002434  000000  DEVMAP:  .WORD  0      ;BIT MAP OF ACTIVE DEVICES
47     002436  000000  DEVPTR:  .WORD  0      ;DEVICE MAP BIT POINTER
48     002440  000000  UNIT:    .WORD  0      ;CONTAINS UNIT NO. (1 TO N)
49     002442  000000  TSTNUM:  .WORD  0      ;CONTAINS TEST NUMBER FOR SOME TESTS
50     002444  000000  STARES:  .WORD  0      ;FLAG=0 IF FIRST PASS AFTER STA OR RES
51
52     ;***** CURRENT DEVICE PARAMETERS *****
53     002446  160170  MPCSR:   .WORD  160170  ;POINTER TO MICROPROCESSOR CSR'S
54     002450  160171  BSEL1:   .WORD  160171  ;POINTER TO BSEL1
55     002452  160172  BSEL2:   .WORD  160172  ;POINTER TO BSEL2
56     002454  BSEL4:
57     002454  160174  SEL4:    .WORD  160174  ;POINTER TO SEL4
```

58	002456	160176	SEL6:	.WORD	160176	; POINTER TO SEL6
59	002460	000300	MPIVEC:	.WORD	300	; MICROPROCESSOR INPUT INTERRUPT VECTOR
60	002462	000304	MPOVEC:	.WORD	304	; MICROPROCESSOR OUTPUT INTERRUPT VECTOR
61	002464	000240	MPRIOR:	.WORD	240	; MICROPROCESSOR DEVICE PRIORITY
62	002466	000000	LUSW11:	.WORD	0	; LINE UNIT SWITCH PACK #1
63	002470	000000	LUSW12:	.WORD	0	; LINE UNIT SWITCH PACK #2
64	002472	000000	LUSW13:	.WORD	0	; LINE UNIT SWITCH PACK #3
65	002474	000000	TSTCON:	.WORD	0	; TEST CONNECTOR INDICATOR
66	002476	000000	RUNINH:	.WORD	0	; RUN SWITCH INDICATOR
67						
68			;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****			
69	002500	000	FDDAT:	.BYTE	0	
70	002501	000		.BYTE	0	
71	002502	000		.BYTE	0	
72	002503	000		.BYTE	0	
73	002504	000		.BYTE	0	
74	002505	000		.BYTE	0	
75	002506	000		.BYTE	0	
76	002507	000		.BYTE	0	
77						
78			;***** GEN'L PURPOSE SCRATCH STORAGE *****			
79	002510	000000	REG0:	.WORD	0	
80	002512	000000	REG1:	.WORD	0	
81	002514	000000	REG2:	.WORD	0	
82	002516	000000	REG3:	.WORD	0	
83	002520	000000	REG4:	.WORD	0	
84	002522	000000	REG5:	.WORD	0	
85	002524	000000	REG6:	.WORD	0	
86	002526	000000	REG7:	.WORD	0	
87						
88			;***** SCRATCH STORAGE FOR MESSAGE REPORTING *****			
89	002530	000000	TMP0:	.WORD	0	
90	002532	000000	TMP1:	.WORD	0	
91	002534	000000	TMP2:	.WORD	0	
92	002536	000000	TMP3:	.WORD	0	
93	002540	000000	TMP4:	.WORD	0	
94	002542	000000	TMP5:	.WORD	0	
95	002544	000000	TMP6:	.WORD	0	
96	002546	000000	TMP7:	.WORD	0	
97						
98			;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****			
99	002550		UPBITS:			
100	002550	000		.BYTE	000	; MASK FOR REG 10
101	002551	056		.BYTE	056	; MASK FOR REG 11
102	002552	000		.BYTE	000	; MASK FOR REG 12
103	002553	257		.BYTE	257	; MASK FOR REG 13
104	002554	100		.BYTE	100	; MASK FOR REG 14
105	002555	377		.BYTE	377	; MASK FOR REG 15
106	002556	377		.BYTE	377	; MASK FOR REG 16
107	002557	306		.BYTE	306	; MASK FOR REG 17
108						
109	002560	200	R14NRW:	.BYTE	200	; REG 14 NON-R/W BITS
110						
111			;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****			
112	002561		ANBITS:			
113	002561	377		.BYTE	377	; MASK FOR AX0-15
114	002562	377		.BYTE	377	; MASK FOR AX0-16

115	002563	000	.BYTE	000	;MASK FOR AX1-15
116	002564	360	.BYTE	360	;MASK FOR AX1-16
117	002565	000	.BYTE	000	;MASK FOR AX2-15
118	002566	000	.BYTE	000	;MASK FOR AX2-16
119	002567	004	.BYTE	004	;MASK FOR AX3-15
120	002570	030	.BYTE	030	;MASK FOR AX3-16
121					
122					
123	002571		;***** DATA PATTERN A *****		
124	002571	125	PATA:		
125	002572	252	.BYTE	125	
126	002573	000	.BYTE	252	
127	002574	377	.BYTE	000	
128	002575	001	.BYTE	377	
129	002576	002	.BYTE	001	
130	002577	004	.BYTE	002	
131	002600	010	.BYTE	004	
132	002601	020	.BYTE	010	
133	002602	040	.BYTE	020	
134	002603	100	.BYTE	040	
135	002604	200	.BYTE	100	
136	002605	376	.BYTE	200	
137	002606	375	.BYTE	376	
138	002607	373	.BYTE	375	
139	002610	367	.BYTE	373	
140	002611	357	.BYTE	367	
141	002612	337	.BYTE	357	
142	002613	277	.BYTE	337	
143	002614	177	.BYTE	277	
144			.BYTE	177	
145					
146	002615		;***** DATA PATTERN B *****		
147	002615	000	PATB:		
148	002616	000	.BYTE	000	
149	002617	040	.BYTE	000	
150	002620	100	.BYTE	040	
151	002621	220	.BYTE	100	
152	002622	000	.BYTE	220	
153	002623	000	.BYTE	000	
154	002624	051	.BYTE	000	
155			.BYTE	051	
156					
157	002625		;***** DATA PATTERN C *****		
158	002625	020	PATC:		
159	002626	020	.BYTE	020	
160	002627	020	.BYTE	020	
161			.BYTE	020	
162					
163	002630		;***** DATA PATTERN D *****		
164	002630	000	PATD:		
165	002631	040	.BYTE	000	
166	002632	000	.BYTE	040	
167			.BYTE	000	
168					
169	002633		;***** DATA PATTERN E *****		
170	002633	000	PATE:		
171	002634	120	.BYTE	000	
			.BYTE	120	

172	002635	020	.BYTE	020
173	002636	100	.BYTE	100
174	002637	120	.BYTE	120
175	002640	000	.BYTE	000
176				
177			;***** DATA PATTERN F *****	
178	002641		PATF:	
179	002641	050	.BYTE	050
180	002642	051	.BYTE	051
181	002643	050	.BYTE	050
182				
183			;***** DATA PATTERN G *****	
184	002644		PATG:	
185	002644	000	.BYTE	000
186	002645	000	.BYTE	000
187	002646	240	.BYTE	240
188	002647	120	.BYTE	120
189	002650	177	.BYTE	177
190	002651	000	.BYTE	000
191	002652	000	.BYTE	000
192	002653	001	.BYTE	001
193				
194			;***** DATA PATTERN H *****	
195	002654		PATH:	
196	002654	000	.BYTE	000
197	002655	000	.BYTE	000
198	002656	377	.BYTE	377
199	002657	017	.BYTE	017
200	002660	377	.BYTE	377
201	002661	377	.BYTE	377
202	002662	375	.BYTE	375
203	002663	377	.BYTE	377
204				
205			;***** DATA PATTERN I *****	
206	002664		PATI:	
207	002664	000	.BYTE	000
208	002665	000	.BYTE	000
209	002666	000	.BYTE	000
210	002667	000	.BYTE	000
211	002670	000	.BYTE	000
212	002671	103	.BYTE	103
213	002672	000	.BYTE	000
214	002673	000	.BYTE	000
215				
216			;***** DATA PATTERN J *****	
217	002674		PATJ:	
218	002674	000	.BYTE	000
219	002675	000	.BYTE	000
220	002676	010	.BYTE	010
221	002677	002	.BYTE	002
222	002700	004	.BYTE	004
223	002701	103	.BYTE	103
224	002702	001	.BYTE	001
225	002703	100	.BYTE	100
226				
227			;***** DATA PATTERN K *****	
228	002704	000	PATK: .BYTE 000	

229	002705	000	.BYTE	000
230	002706	377	.BYTE	377
231	002707	377	.BYTE	377
232	002710	125	.BYTE	125
233	002711	125	.BYTE	125
234	002712	252	.BYTE	252
235	002713	252	.BYTE	252
236	002714	001	.BYTE	001
237	002715	000	.BYTE	000
238	002716	002	.BYTE	002
239	002717	000	.BYTE	000
240	002720	004	.BYTE	004
241	002721	000	.BYTE	000
242	002722	010	.BYTE	010
243	002723	000	.BYTE	000
244	002724	020	.BYTE	020
245	002725	000	.BYTE	000
246	002726	040	.BYTE	040
247	002727	000	.BYTE	000
248	002730	100	.BYTE	100
249	002731	000	.BYTE	000
250	002732	200	.BYTE	200
251	002733	000	.BYTE	000
252	002734	000	.BYTE	000
253	002735	001	.BYTE	001
254	002736	000	.BYTE	000
255	002737	002	.BYTE	002
256	002740	000	.BYTE	000
257	002741	004	.BYTE	004
258	002742	000	.BYTE	000
259	002743	010	.BYTE	010
260	002744	000	.BYTE	000
261	002745	020	.BYTE	020
262	002746	000	.BYTE	000
263	002747	040	.BYTE	040
264	002750	000	.BYTE	000
265	002751	100	.BYTE	100
266	002752	000	.BYTE	000
267	002753	200	.BYTE	200
268	002754	376	.BYTE	376
269	002755	377	.BYTE	377
270	002756	375	.BYTE	375
271	002757	377	.BYTE	377
272	002760	373	.BYTE	373
273	002761	377	.BYTE	377
274	002762	367	.BYTE	367
275	002763	377	.BYTE	377
276	002764	357	.BYTE	357
277	002765	377	.BYTE	377
278	002766	337	.BYTE	337
279	002767	377	.BYTE	377
280	002770	277	.BYTE	277
281	002771	377	.BYTE	377
282	002772	177	.BYTE	177
283	002773	377	.BYTE	377
284	002774	377	.BYTE	377
285	002775	376	.BYTE	376

286	002776	377	.BYTE	377
287	002777	375	.BYTE	375
288	003000	377	.BYTE	377
289	003001	373	.BYTE	373
290	003002	377	.BYTE	377
291	003003	367	.BYTE	367
292	003004	377	.BYTE	377
293	003005	357	.BYTE	357
294	003006	377	.BYTE	377
295	003007	337	.BYTE	337
296	003010	377	.BYTE	377
297	003011	277	.BYTE	277
298	003012	377	.BYTE	377
299	003013	177	.BYTE	177

***** DATA PATTERN L *****
PATL:

302	003014		.BYTE	000
303	003014	000	.BYTE	000
304	003015	000	.BYTE	377
305	003016	377	.BYTE	377
306	003017	377	.BYTE	000
307	003020	000	.BYTE	000
308	003021	000	.BYTE	000

***** DATA PATTERN M *****
PATM:

311	003022		.BYTE	000
312	003022	000	.BYTE	020
313	003023	020	.BYTE	000
314	003024	000	.BYTE	000
315	003025	000	.BYTE	200
316	003026	200	.BYTE	000
317	003027	000	.BYTE	000
318	003030	000	.BYTE	051
319	003031	051	.BYTE	

***** DATA PATTERN N *****
PATN:

322	003032		.BYTE	000
323	003032	000	.BYTE	000
324	003033	000	.BYTE	000
325	003034	000	.BYTE	125
326	003035	125	.BYTE	000
327	003036	000	.BYTE	252
328	003037	252	.BYTE	000
329	003040	000	.BYTE	377
330	003041	377	.BYTE	005
331	003042	005	.BYTE	000
332	003043	000	.BYTE	012
333	003044	012	.BYTE	000
334	003045	000	.BYTE	017
335	003046	017	.BYTE	000
336	003047	000	.BYTE	

***** DATA PATTERN O *****
PATO:

339	003050		.BYTE	000
340	003050	000	.BYTE	041
341	003051	041	.BYTE	004
342	003052	004	.BYTE	

343	003053	010	.BYTE	010
344	003054	020	.BYTE	020
345	003055	040	.BYTE	040
346	003056	100	.BYTE	100
347	003057	101	.BYTE	101
348	003060	200	.BYTE	200
349	003061	201	.BYTE	201
350	003062	300	.BYTE	300
351	003063	111	.BYTE	111
352	003064	301	.BYTE	301
353	003065	375	.BYTE	375

:***** DATA PATTERN P *****
PATP:

356	003066		.BYTE	000
357	003066	000	.BYTE	113
358	003067	113	.BYTE	200
359	003070	200	.BYTE	040
360	003071	040	.BYTE	020
361	003072	020	.BYTE	010
362	003073	010	.BYTE	001
363	003074	001	.BYTE	104
364	003075	104	.BYTE	007
365	003076	007	.BYTE	105
366	003077	105	.BYTE	007
367	003100	007	.BYTE	144
368	003101	144	.BYTE	107
369	003102	107	.BYTE	157
370	003103	157		

:***** DATA PATTERN U *****
PATU:

373	003104	000	.BYTE	000
374	003105	000	.BYTE	000
375	003106	001	.BYTE	001
376	003107	000	.BYTE	000
377	003110	013	.BYTE	013
378	003111	000	.BYTE	000
379	003112	011	.BYTE	011
380	003113	000	.BYTE	000
381	003114	021	.BYTE	021
382	003115	000	.BYTE	000
383	003116	101	.BYTE	101
384	003117	000	.BYTE	000
385	003120	301	.BYTE	301
386	003121	000	.BYTE	000
387	003122	000	.BYTE	000
388	003123	001	.BYTE	001
389	003124	000	.BYTE	000
390	003125	002	.BYTE	002
391	003126	000	.BYTE	000
392	003127	004	.BYTE	004
393	003130	000	.BYTE	000
394	003131	040	.BYTE	040
395	003132	000	.BYTE	000
396	003133	100	.BYTE	100
397	003134	000	.BYTE	000
398	003135	200	.BYTE	200
399	003136	000	.BYTE	000

400	003137	346	.BYTE	346
401	003140	000	.BYTE	000
402	003141	345	.BYTE	345
403	003142	000	.BYTE	000
404	003143	343	.BYTE	343
405	003144	000	.BYTE	000
406	003145	307	.BYTE	307
407	003146	000	.BYTE	000
408	003147	247	.BYTE	247
409	003150	000	.BYTE	000
410	003151	147	.BYTE	147

411

412

:***** PATTERN V *****

413	003152	000	PATV: .BYTE	000
414	003153	000	.BYTE	000
415	003154	333	.BYTE	333
416	003155	000	.BYTE	000
417	003156	331	.BYTE	331
418	003157	000	.BYTE	000
419	003160	323	.BYTE	323
420	003161	000	.BYTE	000
421	003162	313	.BYTE	313
422	003163	000	.BYTE	000
423	003164	233	.BYTE	233
424	003165	000	.BYTE	000
425	003166	133	.BYTE	133
426	003167	000	.BYTE	000
427	003170	000	.BYTE	000
428	003171	001	.BYTE	001
429	003172	000	.BYTE	000
430	003173	002	.BYTE	002
431	003174	000	.BYTE	000
432	003175	004	.BYTE	004
433	003176	000	.BYTE	000
434	003177	040	.BYTE	040
435	003200	000	.BYTE	000
436	003201	100	.BYTE	100
437	003202	000	.BYTE	000
438	003203	200	.BYTE	200
439	003204	000	.BYTE	000
440	003205	346	.BYTE	346
441	003206	000	.BYTE	000
442	003207	345	.BYTE	345
443	003210	000	.BYTE	000
444	003211	343	.BYTE	343
445	003212	000	.BYTE	000
446	003213	307	.BYTE	307
447	003214	000	.BYTE	000
448	003215	247	.BYTE	247
449	003216	000	.BYTE	000
450	003217	147	.BYTE	147

451

452 003220

ENDPAT:
.EVEN

453

454

455

456

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495

003220 000400
003222 000400
003224 000000
003226 000125
003230 000252
003232 000377
003234 000000
003236 001000
003240 001000
003242 001000
003244 001000

003246 000377
003250 000000
003252 000125
003254 000252
003256 001000
003260 001000

003262

;*** TEST MESSAGES TO BE TRANSMITTED ***

MSG1: TXSOM
TXSOM
000
125
252
377
000
TXEOM
TXEOM
TXEOM
TXEOM

MSG4: 377
000
125
252
TXEOM
TXEOM

;*** RECEIVED DATA BUFFER (64. WORDS) ***
RCVBUF: .BLKW 64.

```

1      .SBTTL  GLOBAL TEXT SECTION
2
3      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4      ;      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5      ;      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
6      ;      MORE THAN ONE TEST.
7      ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8
9      ;*****
10     ;  NAMES OF DEVICES SUPPORTED BY PROGRAM
11     ;*****
12     003462      DEVTYP  <M8203>
13
14     003462      LSDVTYP::
15     003462      .ASCIZ  /M8203/
16     003465      115      070      062
17     060      063      000
18
19
20
21
22
23
24
25     ;*****
26     ;  TITLE OF PROGRAM
27     ;*****
28     003470      DESCRIPT      <M8203 STATIC LOGIC TESTS - PART 1 OF 2>
29     003470      LSDESC::
30     003470      115      070      062      .ASCIZ  /M8203 STATIC LOGIC
31     003473      060      063      040
32     003476      123      124      101
33     003501      124      111      103
34     003504      040      114      117
35     003507      107      111      103
36     003512      040      124      105
37     003515      123      124      123
38     003520      040      055      040
39     003523      120      101      122
40     003526      124      040      061
41     003531      040      117      106
42     003534      040      062      000
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

.SBTTL GLOBAL SUBROUTINES

://///////////////////////
:/ THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
://///////////////////////:*****
:* STPCLK - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
:* EXECUTE AN INSTRUCTION WHICH IS PASSED IN THE WORD FOLLOWING THE CALL.
:*****

STPCLK:

BISB #ROMO!ROMI,@BSEL1 ;SET ROMO, ROMI BITS IN BSEL1
MOV @ (SP),@SEL6 ;PUT INSTRUCTION INTO SEL6
BISB #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
BICB #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
ADD #2,(SP) ;FIX UP RETURN PC
RTS PC ;RETURN:*****
:* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOK
:*****

MSTCLR:

MOV R1,-(SP) ;SAVE R1
MOV REGNUM,-(SP) ;SAVE LU REG NO.
MOVB #MCLR,@BSEL1 ;SET MASTER CLEAR BIT
BICB #RUN!MCLR,@BSEL1 ;CLEAR RUN AND MCLR BITS
MOV #20.,R1 ;INITIALIZE STALL COUNTER
2\$: NOP ;STALL IN LOOP FOR SEVERAL MICRO-SEC
DEC R1
BNE 2\$
BISB #LULOOK,@BSEL1 ;SET LU LOOP
MOV #13,REGNUM ;SET LU REG NO. = 13
CLR WRIBYT
JSR PC,WRITLU ;CLEAR REG 13
MOV (SP)+,REGNUM ;RESTORE LU REG NO.
MOV (SP)+,R1 ;RESTORE R1
CLR SAVLEN ;CLEAR CHAR LENGTH FROM SETUP
RTS PC ;RETURN:*****
:* READLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR
:* TO EXECUTE AN INSTRUCTION WHICH READS THE LINE UNIT REG WHOSE
:* NUMBER IS PASSED IN REGNUM, INTO REDBYT.
:*****

READLU:

```
58 003672 010146          MOV      R1,-(SP)      ;SAVE R1
59 003674 013701 002400   MOV      REGNUM,R1    ;GET LINE UNIT REG NUMBER
60 003700 006301          ASL      R1            ;SHIFT INTO SOURCE BITS 4-7
61 003702 006301          ASL      R1
62 003704 006301          ASL      R1
63 003706 006301          ASL      R1
64 003710 052701 000004   BIS      #4,R1        ;SET DESTINATION = BSEL4
65 003714 052701 021000   BIS      #MVIOX,R1    ;SET REST OF MOVE INSTRUCTION
66 003720 010137 003730   MOV      R1,2$        ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
67 003724 004737 003540   JSR      PC,STPCLK     ;EXECUTE MOVE INSTRUCTION
68 003730 000000          .WORD    0            ;INSTRUCTION GOES HERE
69 003732 117737 176516 002364 2$: MOVB   @BSEL4,REDBYT ;GET LU REG CONTENTS INTO REDBYT
70 003740 105037 002365   CLRB   REDBYT+1      ;CLR HI BYTE OF STORAGE
71 003744 012601          MOV      (SP)+,R1      ;RESTORE R1
72 003746 000207          RTS      PC            ;RETURN
```

```
*****
;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
;* EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
;* INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM,
*****
```

WRITLU:

```
83 003750
84 003750 010146          MOV      R1,-(SP)      ;SAVE R1
85 003752 013701 002400   MOV      REGNUM,R1    ;GET LINE UNIT REG NUMBER
86 003756 052701 000100   BIS      #100,R1      ;SET SOURCE = BSEL4
87 003762 052701 122000   BIS      #MVIOX,R1    ;SET REST OF MOVE INSTRUCTION
88 003766 010137 004010   MOV      R1,2$        ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
89 003772 105037 002367   CLRB   WRIBYT+1      ;CLR HI BYTE OF STORAGE
90 003776 113777 002366 176450 2$: MOVB   WRIBYT,@BSEL4 ;LOAD BYTE INTO BSEL4
91 004004 004737 003540   JSR      PC,STPCLK     ;EXECUTE MOVE INSTRUCTION
92 004010 000000          .WORD    0
93 004012 012601          MOV      (SP)+,R1      ;RESTORE R1
94 004014 000207          RTS      PC            ;RETURN
```

```
*****
;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
;* REGISTER STORAGE TABLE (LUREG:).
*****
```

GETREG:

```
104 004016 010146          MOV      R1,-(SP)      ;SAVE R1
105 004020 013746 002400   MOV      REGNUM,-(SP) ;SAVE CURRENT REG NO.
106 004024 012701 002302   MOV      #LUR10,R1   ;INIT POINTER TO REG STORAGE TABLE
107 004030 012737 000010 002400 3$: MOV      #10,REGNUM ;INIT LU REG NO. TO 10
108 004036 004737 003672   JSR      PC,READLU    ;READ A LINE UNIT REG
109 004042 113721 002364   MOVB   REDBYT,(R1)+   ;PUT BYTE READ INTO TABLE
110 004046 105021          CLRB   (R1)+         ;CLEAR UPPER BYTE OF TABLE ENTRY
111 004050 005237 002400   INC      REGNUM        ;INCREMENT REG NO.
112 004054 023727 002400 000020 CMP      REGNUM,#20 ;SEE IF ALL REGS READ YET
113 004062 002765          BLT      3$           ;BR IF NOT
114 004064 012637 002400   MOV      (SP)+,REGNUM ;RESTORE CURRENT REG NO.
```

115 004070 012601
116 004072 000207

MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

117
118
119
120
121
122
123
124
125
126
127
128
129

* LOOPIN - THIS SUBROUTINE PLACES THE MICROPROCESSOR IN A LOOP ON AN
* INSTRUCTION, BY MOVING THE INSTRUCTION FROM THE WORD FOLLOWING THE CALL
* INTO SEL6, AND SETTING RUN AND ROMI IN BSEL1. THE SUBROUTINE RETURNS
* WITH THE MICROPROCESSOR STUCK IN THE LOOP, AND IF IT IS DESIRED TO
* TERMINATE THE LOOP, THE PDP-11 PROGRAM MUST CLEAR THE RUN BIT IN
* BSEL1, OR CALL SUBROUTINE MSTCLR TO DO THIS.

130 004074
131 004074 152777 000006 176346
132 004102 017677 000000 176346
133 004110 152777 000206 176332
134 004116 062716 000002
135 004122 000207

LOOPIN:
BISB #ROMI!ROMI,@BSEL1 ;SET ROMO, ROMI BITS IN BSEL1
MOV @ (SP),@SEL6 ;PUT MICROINSTRUCTION INTO SEL6
BISB #RUN!ROMI!ROMI,@BSEL1 ;SET RUN, ROMO, ROMI IN BSEL1
ADD #2,(SP) ;FIX UP RETURN PC
RTS PC ;RETURN WITH MICROPROCESSOR STUCK IN SINGLE
; INSTRUCTION LOOP

136
137
138
139
140
141
142
143
144
145
146
147

* READAX - THIS SUBROUTINE READS THE USYRT REG PAIR WHOSE NUMBER (0-3)
* IS PASSED IN BITS 1,2 OF AXNUM ON ENTRY, AND RETURNS THE BYTES READ IN
* RAX15 AND RAX16. IF THE LINE UNIT DOES NOT RESPOND WITH READY IN REG 14,
* RRDYTO BIT IS SET IN ERROR1 ON RETURN.

148 004124 010146
149 004126 013746 002400
150 004132 042737 000001 002420
151 004140 012737 000014 002400
152 004146 113737 002402 002366
153 004154 006237 002366
154 004160 152737 000024 002366
155 004166 053737 002426 002366
156 004174 004737 003750
157 004200 005001
158 004202 004737 003672 6\$:
159 004206 132737 000200 002364
160 004214 001006
161 004216 005201
162 004220 001370
163 004222 052737 000001 002420
164 004230 000424
165 004232 012737 000015 002400 9\$:
166 004240 004737 003672
167 004244 113737 002364 002370
168 004252 105037 002371
169 004256 012737 000016 002400
170 004264 004737 003672
171 004270 113737 002364 002372

READAX: MOV R1,-(SP) ;SAVE R1
MOV REGNUM,-(SP) ;STORE CURRENT REG NO.
BIC #RRDYTO,ERROR1 ;CLEAR ERROR BIT
MOV #14,REGNUM ;SET LU REG NO. = 14
MOVB AXNUM,WRIBYT ;SET UP AX REG NO. BITS
ASR WRIBYT
BISB #RDAX!ENAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
BIS DISILO,WRIBYT ;SET PROPER STATE OF DISSI BIT
JSR PC,WRITLU ;SET RDAX AND ENAX IN REG 14
CLR R1 ;INIT TIMER
JSR PC,READLU ;READ REG 14
BITB #READY,REDBYT ;SEE IF READY BIT SET IN REG 14 YET
BNE 9\$;BR IF READY SET
INC R1 ;INCR TIMER
BNE 6\$;BR IF TIMER DIDN'T TIME OUT YET
BIS #RRDYTO,ERROR1 ;SET ERROR FLAG FOR TIME OUT ON READ RDY
BR 12\$;BR TO RETURN
9\$: MOV #15,REGNUM ;SET REG NO. = 15
JSR PC,READLU ;READ REG 15
MOVB REDBYT,RAX15 ;STORE REG AX-15
CLRB RAX15+1 ;CLR HI BYTE OF STORAGE
MOV #16,REGNUM ;SET REG NO. = 16
JSR PC,READLU ;READ REG 16
MOVB REDBYT,RAX16 ;STORE REG AX-16

172 004276 105037 002373
173 004302 012637 002400
174 004306 012601
175 004310 000207

12\$: CLR B RAX16+1 ;CLR HI BYTE OF STORAGE
MOV (SP)+,REGNUM ;RESTORE CURRENT REG NO.
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

176
177
178
179
180
181
182
183
184
185

* WRITAX - THIS SUBROUTINE WRITES THE USYRT REG PAIR WHOSE NUMBER (0-3) IS
* PASSED IN BITS 1,2 OF AXNUM ON ENTRY, WITH THE DATA FROM WAX15 AND
* WAX16. IF LINE UNIT DOES NOT RESPOND WITH READY IN REG 14, WRDYTO BIT
* IS SET IN ERROR1 ON RETURN.

186 004312 010146
187 004314 013746 002400
188 004320 042737 000002 002420
189 004326 012737 000014 002400
190 004334 113737 002402 002366
191 004342 006237 002366
192 004346 053737 002426 002366
193 004354 004737 003750
194 004360 012737 000015 002400
195 004366 105037 002375
196 004372 113737 002374 002366
197 004400 004737 003750
198 004404 005237 002400
199 004410 105037 002377
200 004414 113737 002376 002366
201 004422 004737 003750
202 004426 012737 000014 002400
203 004434 113737 002402 002366
204 004442 006237 002366
205 004446 152737 000014 002366
206 004454 053737 002426 002366
207 004462 004737 003750
208 004466 005001
209 004470 004737 003672
210 004474 132737 000200 002364
211 004502 001005
212 004504 005201
213 004506 001370
214 004510 052737 000002 002420
215 004516 012637 002400
216 004522 012601
217 004524 000207

WRITAX: MOV R1,-(SP) ;SAVE R1
MOV REGNUM,-(SP) ;SAVE CURRENT REG NO.
BIC #WRDYTO,ERROR1 ;CLEAR ERROR BIT
MOV #14,REGNUM ;SET LU REG NO. = 14
MOVB AXNUM,WRIBYT ;SET AX REG NO. BITS
ASR WRIBYT
BIS DISILO,WRIBYT ;SET PROPER STATE OF DISSI BIT
JSR PC,WRITLU ;SET AX NO. BITS IN REG 14
MOV #15,REGNUM ;SET REG NO. = 15
CLRB WAX15+1 ;CLR HI BYTE OF STORAGE
MOVB WAX15,WRIBYT ;SET UP BYTE TO WRITE INTO REG 15
JSR PC,WRITLU ;WRITE BYTE INTO REG 15
INC REGNUM ;SET REG NO. = 16
CLRB WAX16+1 ;CLR HI BYTE OF STORAGE
MOVB WAX16,WRIBYT ;SET UP BYTE TO WRITE INTO REG 16
JSR PC,WRITLU ;WRITE BYTE INTO REG 16
MOV #14,REGNUM ;SET REG NO. = 14
MOVB AXNUM,WRIBYT ;SET AX REG NO. BITS
ASR WRIBYT
BISB #ENAX!WAX,WRIBYT ;SET UP BITS TO LOAD INTO REG 14
BIS DISILO,WRIBYT ;SET PROPER STATE OF DISSI BIT
JSR PC,WRITLU ;SET ENAX AND WAX IN REG 14
CLR R1 ;INIT PROGRAM TIMER
JSR PC,READLU ;READ REG 14
BITB #READY,REDBYT ;SEE IF READY BIT SET IN REG 14 YET
BNE 9\$;BR IF READY SET
INC R1 ;INCR TIMER
BNE 6\$;BR IF TIMER DIDN'T TIME OUT YET
BIS #WRDYTO,ERROR1 ;SET ERROR FLAG BIT FOR TIME OUT ON WRITE RDY
9\$: MOV (SP)+,REGNUM ;RESTORE CURRENT REG NO.
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

218
219
220
221
222
223
224
225
226

* GETALL - THIS SUBROUTINE READS THE LINE UNIT REGS 10-17 AND THE EXTENDED
* REGISTERS AX0-AX3 INTO REGISTER STORAGE TABLE (LUREG:).

227 004526 010146
228 004530 013746 002402

GETALL: MOV R1,-(SP) ;SAVE R1
MOV AXNUM,-(SP) ;SAVE CURRENT AX REG BYTE NO.

```

229 004534 012737 014435 002530      MOV    #DH5,TMP0      ;SET AX LO BYTE NO.
230 004542 032737 000001 002402      BIT    #BIT0,AXNUM    ;SEE IF LO OR HI BYTE
231 004550 001403                BEQ     1$                ;BR IF LO BYTE
232 004552 012737 014440 002530      MOV    #DH6,TMP0      ;SET AX HI BYTE NO.
233 004560 004737 004016 1$:        JSR     PC,GETREG        ;READ AND STORE REGS 10-17
234 004564 142777 000010 175656      BICB   #LULoop,@BSEL1 ;CLEAR LULoop
235 004572 012701 002322                MOV    #AX0.15,R1      ;INIT POINTER TO REG STORAGE TABLE
236 004576 005037 002402                CLR     AXNUM          ;INIT AX REG BYTE NO. TO 0
237 004602 004737 004124 3$:        JSR     PC,READAX        ;READ 2 AX REG BYTES
238 004606 113721 002370                MOVB   RAX15,(R1)+      ;PUT LO BYTE READ INTO TABLE
239 004612 105021                CLRB    (R1)+              ;CLEAR UPPER BYTE OF TABLE ENTRY
240 004614 113721 002372                MOVB   RAX16,(R1)+      ;PUT HI BYTE READ INTO TABLE
241 004620 105021                CLRB    (R1)+              ;CLEAR UPPER BYTE OF TABLE ENTRY
242 004622 062737 000002 002402      ADD     #2,AXNUM        ;INCR AX REG BYTE NO.
243 004630 023727 002402 000010      CMP     AXNUM,#10      ;SEE IF ALL REGS READ YET
244 004636 002761                BLT     3$                ;BR IF NOT
245 004640 012637 002402                MOV     (SP)+,AXNUM    ;RESTORE CURRENT AX REG BYTE NO.
246 004644 012601                MOV     (SP)+,R1      ;RESTORE R1
247 004646 013737 002402 002532      MOV     AXNUM,TMP1
248 004654 006237 002532                ASR     TMP1
249 004660 000207                RTS      PC                ;GET EXTENDED REG NO. FOR PRINTOUT
                                         ;RETURN

```

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

004752

004752

004754

004756

004760

278 004762

279 004764

280 004772

281 004774

104455

000007

012374

015622

000451

132737

001407

004737

002400

002352

002352

001006

016637

162737

000011

003672

000001

000020

000020

002364

002364

004526

004526

```

*****
* OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)
* AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
* AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE
* CALL.
* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN
* RETADR.
*****

```

```

*****
OSIRDY: MOV    REGNUM,-(SP)      ;SAVE LU REG NO.
        MOV    SUBRPC,-(SP)
        TST    SUBRPC          ;SEE IF THIS IS A NESTED CALL
        BNE    1$              ;BR IF YES
        MOV    4(SP),SUBRPC     ;GET PC OF SUBROUTINE CALL
        SUB    #4,SUBRPC
        MOV    #11,REGNUM      ;SET REG NO. TO 11
        JSR    PC,READLU       ;READ REG 11
        BIT    #BIT0,@4(SP)    ;GET EXPECTED STATE OF ORDY
        BEQ    3$              ;BR IF EXPECTED ORDY = 0
        BITB   #ORDY,REDBYT    ;SEE IF ORDY = 1
        BNE    9$              ;BR IF ORDY = 1
        JSR    PC,GETALL       ;GET REGS FOR PRINTOUT
        ;REPORT ORDY NOT SET
        ERRDF  7,EM7,ERR4

```

```

TRAP    C$ERDF
.WORD   7
.WORD   EM7
.WORD   ERR4

```

```

        BR     16$              ;TAKE ERROR RETURN
        BITB   #ORDY,REDBYT    ;SEE IF ORDY = 0
        BEQ    9$              ;BR IF ORDY = 0
        JSR    PC,GETALL       ;GET REGS FOR PRINTOUT

```

```
282 ;REPORT ORDY NOT CLEARED
283 005000 ERRDF 8,EM8,ERR4
      005000 104455
      005002 000010
      005004 012411
      005006 015622
284 005010 000436
285 005012 012737 000017 002400 9$: BR 16$ ;TAKE ERROR RETURN
286 005020 004737 003372 PC,READLU ;SET REG NO. = 17
287 005024 132776 000002 000004 BITB #BIT1,24(SP) ;READ LU REG 17
288 005032 001413 BEQ 12$ ;GET EXPECTED STATE OF OCOR
289 005034 132737 000020 002364 BITB #OCOR,REDBYT ;BR IF EXPECTED OCOR = 0
290 005042 001031 BNE 20$ ;SEE IF OCOR = 1
291 005044 004737 004526 JSR PC,GETALL ;BR IF OCOR = 1
292 ;REPORT OCOR NOT SET ;GET REGS FOR PRINTOUT
293 005050 ERRDF 9,EM9,ERR4
      005050 104455
      005052 000011
      005054 012432
      005056 015622
294 005060 000412
295 005062 132737 000020 002364 12$: BR 16$ ;TAKE ERROR RETURN
296 005070 001416 BEQ 20$ ;SEE IF OCOR = 0
297 005072 004737 004526 JSR PC,GETALL ;BR IF OCOR = 0
298 ;REPORT OCOR NOT CLEARED ;GET REGS FOR PRINTOUT
299 005076 ERRDF 10,EM10,ERR4
      005076 104455
      005100 000012
      005102 012447
      005104 015622
300 005106 016637 000002 002400 16$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.
301 005114 013706 002346 PSTACK,SP ;RESTORE STACK POINTER TO BASE LEVEL
302 005120 013746 002362 MOV RETADR,-(SP) ;FIX ERROR RETURN PC
303 005124 000407 BR 23$
304 005126 062766 000002 000004 20$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
305 005134 012637 002352 MOV (SP)+,SUBRPC
306 005140 012637 002400 MOV (SP)+,REGNUM ;RESTORE LU REG NO.
307 005144 000207 23$: RTS PC ;RETURN
308
309
310
311
312
313
314
315
316 005146 010146
317 005150 012701 000310
318 005154 005301
319 005156 001376
320 005160 012601
321 005162 000207
322
323
324
325
326
```

```
*****
* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
*****
WAIT50: MOV R1,-(SP) ;SAVE R1
        MOV #200.,R1 ;INIT COUNTER
3$: DEC R1 ;DECREMENT COUNTER
    BNE 3$ ;BR IF NOT DONE YET
    MOV (SP)+,R1 ;RESTORE R1
    RTS PC ;RETURN
```

```
327
328
329
330 005164 000240
331 005166 000240
332 005170 000240
333 005172 000207
334
335
336
337
338
339
340
341
342
343 005174 013746 002400
344 005200 042737 170000 002422
345 005206 012737 000011 002400
346 005214 113737 002423 002366
347 005222 004737 003750
348 005226 012737 000010 002400
349 005234 113737 002422 002366
350 005242 004737 003750
351 005246 012637 002400
352 005252 000207
353
354
355
356
357
358
359
360
361
362
363
364 005254 010146
365 005256 017601 000002
366 005262 001426
367 005264 100006
368 005266 042701 100000
369 005272 005737 002430
370 005276 001401
371 005300 005301
372 005302 152777 000010 175140 2$:
373 005310 152777 000020 175132 3$:
374 005316 004737 005164
375 005322 142777 000020 175120
376 005330 004737 005164
377 005334 005301
378 005336 001364
379 005340 062766 000002 000002 6$:
380 005346 012601
381 005350 000207
382
383
```

```
*****
; * STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
*****
STALL:  NOP
        NOP
        NOP
        RTS      PC

*****
; * LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
; *      IN BITS 0-11 OF TXWORD.
*****
LDTXSI: MOV      REGNUM, -(SP)      ;SAVE LU REG NO.
        BIC      #170000, TXWORD  ;CLEAR UNUSED BITS
        MOV      #11, REGNUM      ;SET REG NO. = 11
        MOVB     TXWORD+1, WRIBYT ;SET DATA TO BE WRITTEN INTO REG 11
        JSR      PC, WRITLU       ;LOAD DATA INTO REG 11
        MOV      #10, REGNUM      ;SET REG NO. = 10
        MOVB     TXWORD, WRIBYT   ;SET DATA TO BE WRITTEN INTO REG 10
        JSR      PC, WRITLU       ;LOAD DATA INTO REG 10
        MOV      (SP)+, REGNUM    ;RESTORE LU REG NO.
        RTS      PC              ;RETURN

*****
; * STPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES PASSED
; *      IN BITS 0-14 OF THE WORD FOLLOWING THE CALL.
; *      IF BIT 15 = 1, A CHECK IS MADE TO DETERMINE IF THE USYRT CHIP TYPE
; *      REQUIRES DECREMENTING THE NO. OF CYCLES BY 1.
*****
STPLU:  MOV      R1, -(SP)        ;SAVE R1
        MOV      @2(SP), R1      ;GET DESIRED NO. OF CYCLES
        BEQ      6$,             ;IF DESIRED CYCLES = 0, RETURN
        BPL      2$,             ;BR IF CHIP TYPE CHECK NOT NECESSARY
        BIC      #BIT15, R1      ;CLEAR FLAG BIT
        TST      CHPTYP          ;SEE IF SIG USYRT
        BEQ      2$,             ;BR IF YES
        DEC      R1              ;DECREMENT CYCLE COUNT
        BISB     #LULOOP, @BSEL1 ;SET LU LOOP BIT
        BISB     #STEPLU, @BSEL1 ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
        JSR      PC, STALL       ;STALL
        BICB     #STEPLU, @BSEL1 ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
        JSR      PC, STALL       ;STALL
        DEC      R1              ;DECREMENT CYCLE COUNTER
        BNE      3$,             ;BR IF NOT DONE YET
        ADD      #2, 2(SP)        ;FIX UP RETURN PC
        MOV      (SP)+, R1       ;RESTORE R1
        RTS      PC              ;RETURN
```

```
384
385
386
387
388
389
390
391
392 005352 013746 002400
393 005356 013746 002352
394 005362 005737 002352
395 005366 001006
396 005370 016637 000004 002352
397 005376 162737 000004 002352
398 005404 012737 000011 002400
399 005412 004737 003672
400 005416 032776 000001 000004
401 005424 001413
402 005426 132737 000100 002364
403 005434 001031
404 005436 004737 004526
405
406 005442
    005442 104455
    005444 000013
    005446 012470
    005450 015622
407 005452 000412
408 005454 132737 000100 002364
409 005462 001416
410 005464 004737 004526
411
412 005470
    005470 104455
    005472 000014
    005474 012505
    005476 015622
413 005500 016637 000002 002400
414 005506 013706 002346
415 005512 013746 002362
416 005516 000407
417 005520 062766 000002 000004
418 005526 012637 002352
419 005532 012637 002400
420 005536 000207
421
422
423
424
425
426
427
428
429
430
431
432
```

```
*****
* OACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF OACT (REG 11) AND
* REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE
* WORD FOLLOWING THE CALL.
*****
OACTIV: MOV     REGNUM,-(SP)      ;SAVE LU REG NO.
        MOV     SUBRPC,-(SP)
        TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
        BNE     1$             ;BR IF YES
        MOV     4(SP),SUBRPC    ;GET PC OF SUBROUTINE CALL
        SUB     #4,SUBRPC       ;SET REG NO. = 11
1$:      MOV     #11,REGNUM      ;READ REG 11
        JSR     PC,READLU       ;GET EXPECTED STATE OF OACT
        BIT     #BIT0,24(SP)    ;BR IF EXPECTED OACT = 0
        BEQ     3$             ;SEE IF OACT = 1
        BITB    #OACT,REDBYT    ;BR IF OACT = 1
        BNE     9$             ;GET REGS FOR PRINTOUT
        JSR     PC,GETALL
        ;REPORT OACT NOT SET
        ERRDF   11,EM11,ERR4

        BR     6$             ;TAKE ERROR RETURN
        BITB    #OACT,REDBYT    ;SEE IF OACT = 0
        BEQ     9$             ;BR IF OACT = 0
        JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
        ;REPORT OACT NOT CLEARED
        ERRDF   12,EM12,ERR4

        MOV     2(SP),REGNUM    ;RESTORE LU REG NO.
        MOV     PSTACK,SP      ;RESTORE PROGRAM STACK TO BASE LEVEL
        MOV     RETADR,-(SP)    ;FIX UP ERROR RETURN PC
        BR     12$
        ADD     #2,4(SP)        ;FIX UP ERROR-FREE RETURN PC
        MOV     (SP)+,SUBRPC    ;RESTORE LU REG NO.
        MOV     (SP)+,REGNUM    ;RETURN
        RTS     PC

*****
* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY DOING A
* MASTER CLEAR, LOADING AX2-15 AND REG 17 WITH THE DATA PASSED IN THE 2
* WORDS FOLLOWING THE CALL, LOADING 2 SOM CHARS INTO THE TX SILO, AND
* CLOCKING THE LINE UNIT UNTIL THE FIRST SYNCH OR FLAG HAS BEEN SERIALIZED
* IN THE USYRT. THE PROGRAM MONITORS ORDY,OCOR, AND OACT FOR VALID STATES,
* THROUGHOUT THE PROCESS.
*****
```

433
434
435
436 005540 010146
437 005542 013746 002400
438 005546 013746 002402
439 005552 016637 000006 002352
440 005560 162737 000004 002352
441 005566 004737 003576
442 005572 004737 004662
443 005576 000001
444 005600 004737 005352
445 005604 000000
446 005606 012737 000004 002402
447 005614 117637 000006 002374
448 005622 012737 000400 002422
449 005630 113737 002374 002422
450 005636 005037 002376
451 005642 004737 004312
452 005646 012737 000017 002400
453 005654 062766 000002 000006
454 005662 117637 000006 002366
455 005670 004737 003750
456 005674 004737 005174
457 005700 004737 005174
458 005704 004737 005146
459 005710 004737 004662
460 005714 000003
461 005716 004737 005352
462 005722 000000
463 005724 005001
464 005726 012737 000011 002400
465 005734 152777 000010 174506 6\$:
466 005742 152777 000020 174500
467 005750 004737 005164
468 005754 004737 003672
469 005760 132737 000100 002364
470 005766 001014
471 005770 142777 000020 174452
472 005776 004737 005164
473 006002 005201
474 006004 020127 000003
475 006010 002751
476 006012 004737 005352
477 006016 000001
478 006020 012737 000017 002400 9\$:
479 006026 005037 002430
480 006032 004737 003672
481 006036 132737 000020 002364
482 006044 001403
483 006046 012737 000001 002430
484 006054 142777 000020 174366 12\$:
485 006062 004737 005164
486 006066 004737 004662
487 006072 000001
488 006074 062766 000002 000006
489 006102 012637 002402

IF THE SUBROUTINE DETECTS AN ERROR, A RETURN IS MADE TO THE TEST, AT THE ADDRESS CONTAINED IN RETADR.

INITRN: MOV R1,-(SP) ;SAVE R1
MOV REGNUM,-(SP) ;SAVE LU REG NO.
MOV AXNUM,-(SP) ;SAVE AX BYTE NO.
MOV 6(SP),SUBRPC
SUB #4,SUBRPC ;GET PC OF SUBR CALL
JSR PC,MSTCLR ;ISSUE A MASTER CLEAR
JSR PC,OSIRDY ;CHECK ORDY=1, OCOR=0
1
JSR PC,OACTIV ;CHK OACT=0
0
MOV #4,AXNUM ;SET AX BYTE NO. = 4 FOR AX2
MOVB @6(SP),WAX15 ;SET DATA BYTE TO LOAD INTO AX2-15
MOV #TXSOM,TXWORD ;SET TSOM BIT
MOVB WAX15,TXWORD ;SET SYNCH CHAR
CLR WAX16
JSR PC,WRITAX ;LOAD AX2
MOV #17,REGNUM ;SET REG NO. = 17
ADD #2,6(SP) ;INCR POINTER TO NEXT DATA BYTE
MOVB @6(SP),WRIBYT ;SET DATA BYTE TO LOAD INTO REG 17
JSR PC,WRITLU ;LOAD REG 17
JSR PC,LDTXSI ;LOAD THE SILO WITH SOM CHAR
JSR PC,LDTXSI ;LOAD ANOTHER SOM INTO SILO
JSR PC,WAIT50 ;WAIT FOR DATA TO RIPPLE
JSR PC,OSIRDY ;CHK ORDY=1, OCOR=1
3
JSR PC,OACTIV ;CHK FOR OACT = 0
0
CLR R1 ;INIT CYCLE COUNTER
MOV #11,REGNUM ;SET LU REG NO. = 11
BISB #LULOOP,@BSEL1 ;SET LINE UNIT LOOP BIT
BISB #STEPLU,@BSEL1 ;SET CLOCK BIT
JSR PC,STALL ;STALL FOR MICRO-SEC
JSR PC,READLU ;READ REG 11
BITB #OACT,REDBYT ;SEE IF OACT = 1 YET
BNE 9\$;BR IF OACT = 1
BICB #STEPLU,@BSEL1 ;CLEAR CLOCK BIT
JSR PC,STALL ;STALL FOR A MICRO-SEC
INC R1 ;INCR CYCLE COUNT
CMP R1,#3 ;SEE IF 3 CYCLES DONE YET
BLT 6\$;BR IF NOT
JSR PC,OACTIV ;CHK FOR OACT = 1
1
MOV #17,REGNUM ;SET REG NO. = 17
CLR CHPTYP ;CLEAR USYRT CHIP INDICATOR
JSR PC,READLU ;READ REG 17
BITB #OCOR,REDBYT ;CHK FOR OCOR CLEARED YET
BEQ 12\$;BR IF YES - IT IS SIG CHIP
MOV #1,CHPTYP ;SET INDICATOR FOR OTHER CHIP TYPE
BICB #STEPLU,@BSEL1 ;CLEAR CLOCK BIT
JSR PC,STALL ;STALL FOR MICRO-SEC
JSR PC,OSIRDY ;CHK FOR ORDY = 1, OCOR = 0
1
ADD #2,6(SP) ;FIX UP RETURN PC
MOV (SP)+,AXNUM ;RESTORE AX BYTE NO.

490 006106 012637 002400
491 006112 012601
492 006114 005037 002352
493 006120 000207

MOV (SP)+,REGNUM ;RESTORE LU REG NO.
MOV (SP)+,R1 ;RESTORE R1
CLR SUBRPC ;CLEAR SUBR CALL PC
RTS PC ;RETURN

494
495
496
497
498
499

500
501
502
503
504
505
506
507
508
509

* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHARACTER, BY LOADING
* THE TX SILO WITH DATA PASSED IN BITS 0-11 OF THE WORD FOLLOWING THE CALL
* AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES PASSED IN BITS 0-14
* OF THE SECOND WORD FOLLOWING THE CALL. IF BIT 15 = 1, A CHK IS MADE TO
* DETERMINE IF THE USYRT CHIP TYPE REQUIRES DECREMENTING THE NO. OF CYCLES
* BY 1. THE PROGRAM CHECKS FOR VALID STATES OF ORDY,
* OCOR, AND OACT THROUGHOUT THE PROCESS.
* IF AN ERROR IS DETECTED, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
* CONTAINED IN RETADR.

510 006122 010146
511 006124 010246
512 006126 016637 000004 002352
513 006134 162737 000004 002352
514 006142 017637 000004 002422
515 006150 004737 005174
516 006154 004737 005146
517 006160 062766 000002 000004
518 006166 005001
519 006170 017602 000004
520 006174 005702
521 006176 100006
522 006200 042702 100000
523 006204 005737 002430
524 006210 001401
525 006212 005302
526 006214 004737 005352
527 006220 000001
528 006222 020102
529 006224 001410
530 006226 004737 004662
531 006232 000003
532 006234 004737 005254
533 006240 000001
534 006242 005201
535 006244 000763
536 006246 004737 004662
537 006252 000001
538 006254 062766 000002 000004
539 006262 005037 002352
540 006266 012602
541 006270 012601
542 006272 000207

TXCHAR: MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
MOV 4(SP),SUBRPC
SUB #4,SUBRPC ;GET PC OF SUBR CALL
MOV @4(SP),TXWORD ;GET DATA TO BE TRANSMITTED
JSR PC,LDTXSI ;LOAD THE TX SILO WITH THE DATA
JSR PC,WAIT50 ;WAIT FOR DATA TO RIPPLE DOWN SILO
ADD #2,4(SP) ;INCR POINTER
CLR R1 ;INIT CYCLE COUNT
MOV @4(SP),R2 ;GET DESIRED NO. OF CYCLES
TST R2 ;SEE IF CHIP TYPE CHK SHOULD BE MADE
BPL 9\$;BR IF NOT
BIC #BIT15,R2 ;CLEAR FLAG BIT
TST CHPTYP ;SEE IF SIG USYRT
BEQ 9\$;BR IF YES
DEC R2 ;DECREMENT NO. OF CYCLES
9\$: JSR PC,OACTIV ;CHK OACT = 1
1
CMP R1,R2 ;SEE IF REQUIRED CYCLES DONE YET
BEQ 12\$;BR IF YES
JSR PC,OSIRDY ;CHK ORDY=1, OCOR=1
3
JSR PC,STPLU ;STEP LU ONE CYCLE
1
INC R1 ;INCR CYCLE COUNT
BR 9\$
12\$: JSR PC,OSIRDY ;CHK ORDY=1, OCOR=0
1
ADD #2,4(SP) ;FIX UP RETURN PC
CLR SUBRPC ;CLEAR SUBR CALL PC
MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

543
544
545
546

```
547
548
549
550
551
552 006274 010146
553 006276 013746 002400
554 006302 016637 000004 002352
555 006310 162737 000004 002352
556 006316 012737 000011 002400
557 006324 012737 000200 002366
558 006332 004737 003750
559 006336 004737 005146
560 006342 004737 004662
561 006346 000001
562 006350 004737 005352
563 006354 000000
564 006356 012737 000013 002400
565 006364 004737 003672
566 006370 032737 000040 002364
567 006376 001406
568 006400 004737 004526
569
570 006404
    006404 104455
    006406 000101
    006410 014244
    006412 015622
571 006414 005037 002352
572 006420 012637 002400
573 006424 012601
574 006426 000207
575
576
577
578
579
580
581
582
583
584
585
586
587
588 006430 013746 002400
589 006434 013746 002352
590 006440 005737 002352
591 006444 001006
592 006446 016637 000004 002352
593 006454 162737 000004 002352
594 006462 012737 000012 002400
595 006470 004737 003672
596 006474 032776 000002 000004
597 006502 001413
598 006504 132737 000020 002364
599 006512 001022

;*****
;* ENDTRN - THIS SUBROUTINE CLEARS THE TRANSMITTER BY SETTING OC. THE PROGRAM
;*   WAITS FOR 50 US, AND CHECKS FOR ORDY=1, OCOR=0, OACT=0, RTS=0.
;*****
ENDTRN: MOV     R1, -(SP)           ;SAVE R1
        MOV     REGNUM, -(SP)      ;SAVE LU REG NO.
        MOV     4(SP), SUBRPC
        SUB     #4, SUBRPC         ;GET PC OF SUBROUTINE CALL
        MOV     #11, REGNUM        ;SET LU REG NO. = 11
        MOV     #OC, WRIBYT        ;SET OC IN DATA
        JSR     PC, WRITLU         ;SET OC IN REG 11
        JSR     PC, WAIT50         ;STALL FOR >50 US.
        JSR     PC, OSIRDY         ;CHK ORDY=1, OCOR=0
        JSR     PC, OACTIV         ;CHK OACT = 0
        MOV     #13, REGNUM        ;SET REG NO. = 13
        JSR     PC, READLU         ;READ REG 13
        BIT     #RTS, REDBYT       ;CHK FOR RTS = 0
        BEQ     3$                ;BR IF RTS = 0
        JSR     PC, GETAIL         ;GET REGS FOR PRINTOUT
;REPORT RTS NOT CLEARED
ERRDF 65, EM65, ERR4

TRAP  C$ERDF
.WORD 65
.WORD EM65
.WORD ERR4

3$: CLR     SUBRPC                ;CLEAR SUBR CALL PC
    MOV     (SP)+, REGNUM         ;RESTORE LU REG NO.
    MOV     (SP)+, R1             ;RESTORE R1
    RTS     PC                    ;RETURN

;*****
;* ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
;*   AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
;*   AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE
;*   CALL.
;*   IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
;*   IN RETADR.
;*****
ISIRDY: MOV     REGNUM, -(SP)      ;SAVE LU REG NO.
        MOV     SUBRPC, -(SP)
        TST     SUBRPC            ;SEE IF THIS IS A NESTED CALL
        BNE     1$               ;BR IF YES
        MOV     4(SP), SUBRPC
        SUB     #4, SUBRPC        ;GET PC OF SUBR CALL
        MOV     #12, REGNUM       ;SET REG NO. TO 12
        JSR     PC, READLU        ;READ REG 12
        BIT     #BIT1, 24(SP)     ;GET EXPECTED STATE OF IRDY
        BEQ     3$               ;BR IF EXPECTED IRDY = 0
        BITB    #IRDY, REDBYT     ;SEE IF IRDY = 1
        BNE     9$               ;BR IF IRDY = 1
        1$:
        MOV     #12, REGNUM
        JSR     PC, READLU
        BIT     #BIT1, 24(SP)
        BEQ     3$
        BITB    #IRDY, REDBYT
        BNE     9$
```

```
600 006514 004737 004526          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
601          ;REPORT IRDY NOT SET
602          ERRDF 17,EM17,ERR4
        006520 104455
        006522 000021
        006524 012643
        006526 015622
        006530 000451
603 006532 132737 000020 002364 3$: BR    16$          ;TAKE ERROR EXIT
604          BITB  #IRDY,REDBYT      ;SEE IF IRDY = 0
605          BEQ   9$                ;BR IF IRDY = 0
606          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
607          ;REPORT IRDY NOT SET
608          ERRDF 18,EM18,ERR4
        006546 104455
        006550 000022
        006552 012660
        006554 015622
        006556 000436
609 006560 012737 000017 002400 9$: MOV    #17,REGNUM      ;SET REG NO. = 17
610          JSR    PC,READLU        ;READ REG 17
611          BITB  #BIT0,24(SP)      ;GET EXPECTED STATE OF ICIR
612          BEQ   12$                ;BR IF EXPECTED ICIR = 0
613          BITB  #ICIR,REDBYT      ;SEE IF ICIR = 1
614          BNE   20$                ;BR IF ICIR = 1
615          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
616          ;REPORT ICIR NOT SET
617          ERRDF 19,EM19,ERR4
        006616 104455
        006620 000023
        006622 012701
        006624 015622
        006626 000412
619 006630 132737 000010 002364 12$: BR    16$          ;TAKE ERROR RETURN
620          BITB  #ICIR,REDBYT      ;SEE IF ICIR = 0
621          BEQ   20$                ;BR IF ICIR = 0
622          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
623          ;REPORT ICIR NOT SET
624          ERRDF 20,EM20,ERR4
        006644 104455
        006646 000024
        006650 012716
        006652 015622
        006654 016637 000002 002400 16$: MOV    2(SP),REGNUM ;RESTORE LU REG NO.
626          MOV    PSTACK,SP        ;RESTORE STACK POINTER TO BASE LEVEL
627          MOV    RETADR,-(SP)      ;FIX ERROR RETURN PC
628          BR     23$
629          ADD    #2,4(SP)          ;FIX UP ERROR-FREE RETURN PC
630          MOV    (SP)+,SUBRPC
631          MOV    (SP)+,REGNUM      ;RESTORE LU REG NO.
632          RTS    PC                ;RETURN
633
634
635
636
637
638
639
640
```

* IACTIV - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF IACT (REG 12) AND
* REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE

```
641      ;*      WORD FOLLOWING THE CALL.
642      ;*      IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
643      ;*      RETADR.
644      ;*****
645 006714 013746 002400 IACTIV: MOV     REGNUM,-(SP)    ;SAVE LU REG NO.
646 006720 013746 002352      MOV     SUBRPC,-(SP)
647 006724 005737 002352      TST     SUBRPC
648 006730 001006      BNE     1$
649 006732 016637 000004 002352      MOV     4(SP),SUBRPC
650 006740 162737 000004 002352      SUB     #4,SUBRPC
651 006746 012737 000012 002400 1$:      MOV     #12,REGNUM
652 006754 004737 003672      JSR     PC,READLU
653 006760 032776 000001 000004      BIT     #BIT0,24(SP)
654 006766 001413      BEQ     3$
655 006770 132737 000100 002364      BITB    #IACT,REDBYT
656 006776 001031      BNE     9$
657 007000 004737 004526      JSR     PC,GETALL
658      ;REPORT IACT NOT SET
659 007004      ERRDF 21,EM21,ERR4
660      TRAP C$ERDF
661      .WORD 21
662      .WORD EM21
663      .WORD ERR4
664 007014 000412      BR      6$
665 007016 132737 000100 002364 3$:      BITB    #IACT,REDBYT
666 007024 001416      BEQ     9$
667 007026 004737 004526      JSR     PC,GETALL
668      ;REPORT IACT NOT CLEARED
669      ERRDF 22,EM22,ERR4
670      TRAP C$ERDF
671      .WORD 22
672      .WORD EM22
673      .WORD ERR4
674 007032 000412      BR      6$
675 007034 104455
676 007036 000026
677 007040 012754
678 007042 016637 000002 002400 6$:      MOV     2(SP),REGNUM
679 007050 013706 002346      MOV     PSTACK,SP
680 007054 013746 002362      MOV     RETADR,-(SP)
681 007060 000407      BR      12$
682 007062 062766 000002 000004 9$:      ADD     #2,4(SP)
683 007070 012637 002352      MOV     (SP)+,SUBRPC
684 007074 012637 002400      MOV     (SP)+,REGNUM
685 007100 000207      RTS     PC
686      ;RESTORE LU REG NO.
687      ;RESTORE PROGRAM STACK TO BASE LEVEL
688      ;FIX UP ERROR RETURN PC
689      ;FIX UP ERROR-FREE RETURN PC
690      ;RESTORE LU REG NO.
691      ;RETURN
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

Line	Address	Hex	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464
------	---------	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

```
731
732
733
734
735
736
737 007354 013746 002400
738 007360 012737 000012 002400
739 007366 004737 003672
740 007372 113737 002364 002425
741 007400 042737 170000 002424
742 007406 012737 000010 002400
743 007414 004737 003672
744 007420 113737 002364 002424
745 007426 012637 002400
746 007432 000207
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761 007434 010146
762 007436 010346
763 007440 013746 002400
764 007444 016637 000006 002352
765 007452 162737 000004 002352
766 007460 012737 000012 002400
767 007466 005001
768 007470 017603 000006
769 007474 062703 000003
770 007500 005776 000006
771 007504 001414
772 007506 004737 006714
773 007512 000000
774 007514 004737 006430
775 007520 000001
776 007522 004737 007102
777 007526 000000
778 007530 004737 005254
779 007534 000001
780 007536 004737 005146
781 007542 005201
782 007544 004737 003672
783 007550 132737 000020 002364
784 007556 001005
785 007560 020103
786 007562 002762
787 007564 004737 006430

*****
* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
* SILO ENTRY IN BITS 0-11 OF RXWORD.
*****
RDRXSI: MOV REGNUM, -(SP) ;SAVE LU REG NO.
MOV #12, REGNUM ;SET REG NO. = 12
JSR PC, READLU ;READ LU REG 12
MOVB REDBYT, RXWORD+1 ;GET HI BITS OF SILO ENTRY
BIC #170000, RXWORD ;CLEAR UNUSED BITS
MOV #10, REGNUM ;SET REG NO. = 10
JSR PC, READLU ;READ REG 10
MOVB REDBYT, RXWORD ;GET LOW BITS OF SILO ENTRY
MOV (SP)+, REGNUM ;RESTORE LU REG NO.
RTS PC ;RETURN

*****
* RCVIST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE, AND MONITORS
* STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR IACT = 0, IRDY = 0,
* ICIR = 1, AND RSOM = 0. THEN, THE LINE UNIT IS CLOCKED USING
* STEPLU UNTIL IRDY = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3
* CYCLES AFTER THE NO. OF CYCLES PASSED IN THE WORD FOLLOWING THE CALL.
* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
* CONTAINED IN RETADR.
*****
RCVIST: MOV R1, -(SP) ;SAVE R1
MOV R3, -(SP) ;SAVE R3
MOV REGNUM, -(SP) ;SAVE LU REG NO.
MOV 6(SP), SUBRPC
SUB #4, SUBRPC ;GET PC OF SUBROUTINE CALL
MOV #12, REGNUM ;SET LU REG NO. = 12
CLR R1 ;INIT CYCLE COUNT TO 0
MOV @6(SP), R3 ;GET CYCLE COUNT LIMIT
ADD #3, R3
TST @6(SP) ;SEE IF DESIRED CYCLES = 0
BEQ 8$ ;BR IF YES
JSR PC, IACTIV ;CHK FOR IACT = 0
0 ;
JSR PC, ISIRDY ;CHK FOR ICIR = 1, IRDY = 0
1 ;
JSR PC, RSEOM ;CHK RSOM = 0, REOM = 0 IN AX0-16
0 ;
6$: JSR PC, STPLU ;CLOCK LU FOR 1 CYCLE
1 ;
8$: JSR PC, WAIT50 ;ALLOW SILO DATA TO RIPPLE
INC R1 ;INCREMENT CYCLE COUNT
JSR PC, READLU ;READ REG 12
BITB #IRDY, REDBYT ;SEE IF IRDY = 1 YET
BNE 9$ ;BR IF IRDY = 1
CMP R1, R3 ;SEE IF LIMIT EXCEEDED
BLT 6$ ;BR IF NOT YET
JSR PC, ISIRDY ;CHK FOR ICIR = 1, IRDY = 1
```

```
788 007570 000003          3
789 007572 020176 000006    9$: CMP    R1,@6(SP)      ;SEE IF LESS THAN REQUIRED CYCLES
790 007576 002003          BGE    12$              ;BR IF NOT
791 007600 004737 006430    JSR    PC,ISIRDY        ;CHK FOR ICIR = 1, IRDY = 0
792 007604 000001          1
793 007606 004737 006714    12$: JSR    PC,IACTIV      ;CHK FOR IACT = 1
794 007612 000001          1
795 007614 004737 006430    JSR    PC,ISIRDY        ;CHK FOR ICIR = 1, IRDY = 1
796 007620 000003          3
797 007622 062766 000002 000006 ADD    #2,6(SP)      ;FIX UP RETURN PC
798 007630 012637 002400    MOV    (SP)+,REGNUM      ;RESTORE LU REG NO.
799 007634 012603          MOV    (SP)+,R3          ;RESTORE R3
800 007636 012601          MOV    (SP)+,R1          ;RESTORE R1
801 007640 005037 002352    CLR    SUBRPC          ;CLEAR SUBR CALL PC
802 007644 000207          RTS    PC                ;RETURN
803
804
805
806
807
808
809
810
811
812
813
```

```
*****
* STPERR - THIS SUBROUTINE LOADS THE CONTENTS OF THE FIRST WORD FOLLOWING THE
* CALL INTO REG 17, AND SETS THE IERR BIT, AND CLOCKS THE LINE UNIT
* FOR THE NO. OF CYCLES PASSED IN THE 2ND WORD FOLLOWING THE CALL. THEN,
* IT RESTORES REG 17 TO ITS ORIGINAL CONTENTS, CLEARING THE IERR BIT.
*****
```

```
814 007646 013746 002400 STPERR: MOV    REGNUM,-(SP)      ;SAVE LU REG NO.
815 007652 012737 000017 002400 MOV    #17,REGNUM      ;SET LU REG NO. = 17
816 007660 017637 000002 002366 MOV    @2(SP),WRIBYT
817 007666 152737 000002 002366 BISB    #IERR,WRIBYT
818 007674 004737 003750 JSR    PC,WRITLU      ;SET IERR BIT IN REG 17
819 007700 062766 000002 000002 ADD    #2,2(SP)      ;INCREMENT SUBR ARGUMENT POINTER
820 007706 017637 000002 007720 MOV    @2(SP),3$      ;GET DESIRED NO. OF CYCLES
821 007714 004737 005254 JSR    PC,STPLU        ;CLOCK LU FOR DESIRED NO. OF CYCLES
822 007720 000000 3$: .WORD 0 ;NO. OF CYCLES GOES HERE
823 007722 142737 000002 002366 BICB    #IERR,WRIBYT
824 007730 004737 003750 JSR    PC,WRITLU      ;CLEAR IERR BIT IN REG 17
825 007734 062766 000002 000002 ADD    #2,2(SP)      ;FIX UP RETURN PC
826 007742 012637 002400 MOV    (SP)+,REGNUM      ;RESTORE LU REG NO.
827 007746 000207 RTS    PC                ;RETURN
828
829
830
831
832
833
```

```
*****
* CKDATA - THIS SUBROUTINE READS THE RCV SILO AND COMPARES THE SILO ENTRY
* TO BITS 0-11 OF THE FIRST WORD FOLLOWING THE CALL. IF THERE IS A
* MISMATCH, THE ERROR IS REPORTED AND A RETURN IS MADE TO THE TEST AT THE
* ADDRESS CONTAINED IN RETADR. IF BIT 15 = 0 IN THE FIRST WORD
* FOLLOWING THE CALL, THE SUBROUTINE WILL NOT CHECK THE BCC BIT (SILO
* BIT 8). IF THERE ARE NO ERRORS, THE LINE UNIT IS Clocked FOR THE
* NUMBER OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
*****
```

```
842 007750 010146 CKDATA: MOV    R1,-(SP)      ;SAVE R1
843 007752 013746 002400 MOV    REGNUM,-(SP)      ;SAVE LU REG NO.
844 007756 016637 000004 002352 MOV    4(SP),SUBRPC
```

```
845 007764 162737 000004 002352 SUB #4,SUBRPC ;GET PC OF SUBR CALL
846 007772 017601 000004 MOV @4(SP),R1 ;GET EXPECTED SILO ENTRY
847 007776 042701 170000 BIC #170000,R1 ;CLEAR UNUSED BITS FOR COMPARE
848 010002 004737 007354 JSR PC,RDRXSI ;READ RCV SILO
849 010006 023727 002432 000347 CMP SAVLEN,#TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
850 010014 001005 BNE 4$ ;BR IF CHAR LENGTH NOT = 7
851 010016 042701 000200 BIC #BIT7,R1 ;MASK OFF BIT 8TH BIT
852 010022 042737 000200 002424 BIC #BIT7,RXWORD
853 010030 120137 002424 4$: CMPB R1,RXWORD ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
854 010034 001445 BEQ 6$ ;BR IF MATCH
855 010036 005037 002404 CLR GOODAT
856 010042 110137 002404 MOVB R1,GOODAT ;GET EXPECTED DATA
857 010046 005037 002406 CLR BADDAT
858 010052 113737 002424 002406 MOVB RXWORD,BADDAT ;GET ACTUAL DATA
859 010060 012737 000011 002400 MOV #11,REGNUM ;SET REG NO. = 11
860 010066 004737 003672 JSR PC,READLU ;READ REG 11
861 010072 132737 000001 002364 BITB #UNRR,REDBYT ;SEE IF TX UNDERRUN ERROR
862 010100 001410 BEQ 5$ ;BR IF NOT
863 010102 004737 004526 JSR PC,GETALL ;GET REGS FOR PRINTOUT
864 ;REPORT TX UNDERRUN ERROR
865 010106 ERRDF 54,EM54,ERR4
      010106 104455 TRAP C$ERDF
      010110 000066 .WORD 54
      010112 014206 .WORD EM54
      010114 015622 .WORD ERR4
866 010116 000137 010600 JMP 36$ ;TAKE ERROR EXIT
867 010122 012737 000010 002400 5$: MOV #10,REGNUM ;SET REG NO. = 10
868 010130 004737 004526 JSR PC,GETALL ;GET REGS FOR PRINTOUT
869 ;REPORT RCV'D DATA MISCOMPARE
870 010134 ERRDF 34,EM34,ERR8
      010134 104455 TRAP C$ERDF
      010136 000042 .WORD 34
      010140 013320 .WORD EM34
      010142 020122 .WORD ERR8
871 010144 000137 010600 JMP 36$ ;TAKE ERROR EXIT
872 010150 000301 6$: SWAB R1
873 010152 012737 000012 002400 MOV #12,REGNUM ;SET LU REG NO. FOR ERROR REPORTS
874 010160 120137 002425 CMPB R1,RXWORD+1 ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
875 010164 001002 BNE 7$ ;BR IF MISMATCH
876 010166 000137 010554 JMP 22$ ;CONTINUE
877 010172 005037 002404 7$: CLR GOODAT
878 010176 110137 002404 MOVB R1,GOODAT ;SET EXPECTED DATA
879 010202 005037 002406 CLR BADDAT
880 010206 113737 002425 002406 MOVB RXWORD+1,BADDAT ;SET ACTUAL DATA
881 010214 032776 100000 000004 BIT #BCCCHK,@4(SP) ;SEE IF BCC SHOULD BE IGNORED
882 010222 001433 BEQ 10$ ;BR IF YES
883 010224 132701 000001 BITB #BCC,R1 ;SEE IF EXPECTED BIT = 1
884 010230 001014 BNE 8$ ;BR IF YES
885 010232 132737 000001 002425 BITB #BCC,RXWORD+1 ;SEE IF ACTUAL BIT = 0
886 010240 001424 BEQ 10$ ;BR IF YES
887 010242 004737 004526 JSR PC,GETALL ;GET REGS FOR PRINTOUT
888 ;REPORT BCC NOT CLEARED
889 010246 ERRDF 35,EM35,ERR8
      010246 104455 TRAP C$ERDF
      010250 000043 .WORD 35
      010252 013346 .WORD EM35
      010254 020122 .WORD ERR8
```

```
890 010256 000137 010600      JMP      36$      ;TAKE ERROR EXIT
891 010262 132737 000001 002425 8$: BITB      #BCC,RXWORD+1 ;SEE IF ACTUAL BIT = 1
892 010270 001010          BNE      10$      ;BR IF YES
893 010272 004737 004526      JSR      PC,GETALL ;GET REGS FOR PRINTOUT
894          ;REPORT BCC NOT SET
895          ERRDF      36,EM36,ERR8

      010276 104455          TRAP      C$ERDF
      010300 000044          .WORD      36
      010302 013366          .WORD      EM36
      010304 020122          .WORD      ERR8

896 010306 000137 010600      JMP      36$      ;TAKE ERROR EXIT
897 010312          10$:      BITB      #EBLK,R1      ;SEE IF EXPECTED BIT = 1
898 010312 132701 000002      BNE      12$      ;BR IF YES
899 010316 001014          BITB      #EBLK,RXWORD+1 ;SEE IF ACTUAL BIT = 0
900 010320 132737 000002 002425 BEQ      14$      ;BR IF YES
901 010326 001424          JSR      PC,GETALL ;GET REGS FOR PRINTOUT
902 010330 004737 004526      ;REPORT EBLK NOT CLEARED
903          ERRDF      37,EM37,ERR8
904          TRAP      C$ERDF
      010334 104455          .WORD      37
      010336 000045          .WORD      EM37
      010340 013402          .WORD      ERR8
      010342 020122

905 010344 000137 010600      JMP      36$      ;TAKE ERROR EXIT
906 010350 132737 000002 002425 12$: BITB      #EBLK,RXWORD+1 ;SEE IF ACTUAL BIT = 1
907 010356 001010          BNE      14$      ;BR IF YES
908 010360 004737 004526      JSR      PC,GETALL ;GET REGS FOR PRINTOUT
909          ;REPORT EBLK NOT SET
910          ERRDF      38,EM38,ERR8
      010364 104455          TRAP      C$ERDF
      010366 000046          .WORD      38
      010370 013423          .WORD      EM38
      010372 020122          .WORD      ERR8

911 010374 000137 010600      JMP      36$      ;TAKE ERROR EXIT
912 010400          14$:      BITB      #RAB,R1      ;SEE IF EXPECTED BIT = 1
913 010400 132701 000004      BNE      16$      ;BR IF YES
914 010404 001014          BITB      #RAB,RXWORD+1 ;SEE IF ACTUAL BIT = 0
915 010406 132737 000004 002425 BEQ      18$      ;BR IF YES
916 010414 001424          JSR      PC,GETALL ;GET REGS FOR PRINTOUT
917 010416 004737 004526      ;REPORT RAB NOT CLEARED
918          ERRDF      39,EM39,ERR8
919          TRAP      C$ERDF
      010422 104455          .WORD      39
      010424 000047          .WORD      EM39
      010426 013440          .WORD      ERR8
      010430 020122

920 010432 000137 010600      JMP      36$      ;TAKE ERROR EXIT
921 010436 132737 000004 002425 16$: BITB      #RAB,RXWORD+1 ;SEE IF ACTUAL BIT = 1
922 010444 001010          BNE      18$      ;BR IF YES
923 010446 004737 004526      JSR      PC,GETALL ;GET REGS FOR PRINTOUT
924          ;REPORT RAB NOT SET
925          ERRDF      40,EM40,ERR8
      010452 104455          TRAP      C$ERDF
      010454 000050          .WORD      40
      010456 013460          .WORD      EM40
      010460 020122          .WORD      ERR8

926 010462 000137 010600      JMP      36$      ;TAKE ERROR EXIT
```

```
927 010466      18$:
928 010466 132701 000010      BITB    #OVRR,R1      ;SEE IF EXPECTED BIT = 1
929 010472 001014      BNE      20$      ;BR IF YES
930 010474 132737 000010 002425  BITB    #OVRR,RXWORD+1 ;SEE IF ACTUAL BIT = 0
931 010502 001424      BEQ      22$      ;BR IF YES
932 010504 004737 004526      JSR      PC,GETALL ;GET REGS FOR PRINTOUT
933      ;REPORT OVRR NOT CLEARED
934 010510      ERRDF 41,EM41,ERR8
      010510 104455
      010512 000051
      010514 013474
      010516 020122
      TRAP C$ERDF
      .WORD 41
      .WORD EM41
      .WORD ERR8
935 010520 000137 010600      JMP      36$      ;TAKE ERROR EXIT
936 010524 132737 000010 002425 20$:  BITB    #OVRR,RXWORD+1 ;SEE IF ACTUAL BIT = 1
937 010532 001010      BNE      22$      ;BR IF YES
938 010534 004737 004526      JSR      PC,GETALL ;GET REGS FOR PRINTOUT
939      ;REPORT OVRR NOT SET
940 010540      ERRDF 42,EM42,ERR8
      010540 104455
      010542 000052
      010544 013515
      010546 020122
      TRAP C$ERDF
      .WORD 42
      .WORD EM42
      .WORD ERR8
941 010550 000137 010600      JMP      36$      ;TAKE ERROR EXIT
942 010554      22$:
943 010554 062766 000002 000004  ADD      #2,4(SP)      ;INCR SUBROUTINE ARGUMENT POINTER
944 010562 017637 000004 010574  MOV      @4(SP),24$      ;GET DESIRED CYCLE COUNT
945 010570 004737 005254      JSR      PC,STPLJ      ;CLOCK LU FOR DESIRED CYCLES
946 010574 000000      24$:  .WORD    0
947 010576 000407      BR      38$      ;TAKE ERROR-FREE EXIT
948 010600 011637 002400      36$:  MOV      (SP),REGNUM      ;RESTORE LU REG NO.
949 010604 013706 002346      MOV      PSTACK,SP      ;RESTORE PROGRAM STACK TO BASE LEVEL
950 010610 013746 002362      MOV      RETADR,-(SP)      ;FIX UP ERROR RETURN PC
951 010614 000406      BR      40$
952 010616 062766 000002 000004 38$:  ADD      #2,4(SP)      ;FIX UP ERROR-FREE RETURN PC
953 010624 012637 002400      MOV      (SP)+,REGNUM      ;RESTORE LU REG NO.
954 010630 012601      MOV      (SP)+,R1      ;RESTORE R1
955 010632 005037 002352      40$:  CLR      SUBRPC      ;CLEAR SUBROUTINE PC
956 010636 000207      RTS      PC      ;RETURN
957
958
959
960
961
962
963
964
965
966
*****
; * SETUP - THIS SUBROUTINE LOADS THE FIRST WORD AFTER THE CALL INTO AX2-15
; * (SYNCH CHAR), LOADS THE SECOND WORD AFTER THE CALL INTO REG 17
; * LOADS THE THIRD WORD INTO AX3-15, AND LOADS THE FOURTH INTO AX3-16.
*****
967 010640 013746 002402      SETUP:  MOV      AXNUM,-(SP)      ;SAVE AX BYTE NO.
968 010644 013746 002400      MOV      REGNUM,-(SP)      ;SAVE LU REG NO.
969 010650 012737 000004 002402      MOV      #4,AXNUM      ;SET AX BYTE NO. FOR AX2
970 010656 017637 000004 002374      MOV      @4(SP),WAX15
971 010664 005037 002376      CLR      WAX16
972 010670 004737 004312      JSR      PC,WRITAX      ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
973 010674 012737 000017 002400      MOV      #17,REGNUM      ;SET LU REG NO. = 17
974 010702 062766 000002 000004      ADD      #2,4(SP)      ;INCREMENT ARGUMENT POINTER
975 010710 017637 000004 002366      MOV      @4(SP),WRIBYT
```

976	010716	004737	003750		JSR	PC,WRITLU	;LOAD REG 17
977	010722	012737	000006	002402	MOV	#6,AXNUM	;SET AX BYTE NO. FOR AX3
978	010730	062766	000002	000004	ADD	#2,4(SP)	;INCREMENT ARGUMENT POINTER
979	010736	017637	000004	002374	MOV	@4(SP),WAX15	
980	010744	062766	000002	000004	ADD	#2,4(SP)	;INCR ARGUMENT POINTER
981	010752	017637	000004	002376	MOV	@4(SP),WAX16	
982	010760	013737	002376	002432	MOV	WAX16,SAVLEN	;STORE TX AND RCV CHAR LENGTH
983	010766	004737	004312		JSR	PC,WRITAX	;LOAD AX3-15, AX3-16
984	010772	062766	000002	000004	ADD	#2,4(SP)	;FIX RETURN PC
985	011000	012637	002400		MOV	(SP)+,REGNUM	;RESTORE LU REG NO.
986	011004	012637	002402		MOV	(SP)+,AXNUM	;RESTORE AX BYTE NO.
987	011010	005037	002352		CLR	SUBRPC	;CLEAR SUBROUTINE PC STORAGE
988	011014	000207			RTS	PC	;RETURN

989
990
991992
993

994

995

996

997

998

999 011016 010146

1000 011020 010246

1001 011022 017601 000004

1002 011026 062766 000002 000004

1003 011034 017602 000004 000004

1004 011040 062766 000002 000004

1005 011046 012137 002422

1006 011052 004737 005174

1007 011056 005302

1008 011060 001372

1009 011062 004737 005146

1010 011066 012602

1011 011070 012601

1012 011072 000207

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025 011074 010146

1026 011076 016637 000002 002352

1027 011104 162737 000004 002352

1028 011112 004737 007354

1029 011116 004737 005146

1030 011122 005001

1031 011124 020176 000002

1032 011130 001410

; * LODMSG - THIS SUBROUTINE LOADS THE NO. OF WORDS PASSED IN THE SECOND WORD
; * FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
; * WORD FOLLOWING THE CALL, INTO THE TRANSMITTER SILO.

LODMSG: MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
MOV @4(SP),R1 ;GET MSG POINTER INTO R1
ADD #2,4(SP) ;INCR ARG POINTER
MOV @4(SP),R2 ;GET WORD COUNT INTO R2
ADD #2,4(SP) ;FIX UP RETURN PC
6\$: MOV (R1)+,TXWORD ;GET NEXT MSG WORD
JSR PC,LDTXSI ;LOAD A WORD INTO TX SILO
DEC R2 ;DECR COUNT
BNE 6\$;BR IF NOT DONE YET
JSR PC,WAIT50 ;WAIT FOR SILO TO RIPPLE
MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

; * RXCHAR - THIS SUBROUTINE READS THE RCV SILO AND CLOCKS THE LINE UNIT.
; * FIRST, IT READS THE CHAR FROM THE RCV SILO, AND CHECKS FOR ICIR
; * = 1, IRDY = 0. IT THEN CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES
; * PASSED IN THE WORD FOLLOWING THE CALL, AND THEN CHECKS FOR ICIR
; * = 1, IRDY = 1.

RXCHAR: MOV R1,-(SP) ;SAVE R1
MOV 2(SP),SUBRPC ;GET PC OF SUBR CALL
SUB #4,SUBRPC ;READ RCV SILO
JSR PC,RDRXSI ;ALLOW SILO TO RIPPLE
JSR PC,WAIT50 ;INIT CYCLE COUNT
CLR R1 ;SEE IF REQUIRED CYCLES DONE YET
9\$: CMP R1,@2(SP) ;BR IF YES
BEQ 12\$

```
1033 011132 004737 006430      JSR    PC,ISIRDY      ;CHK ICIR = 1, IRDY = 0
1034 011136 000001              1
1035 011140 004737 005254      JSR    PC,STPLU       ;CLK LU 1 CYCLE
1036 011144 000001              1
1037 011146 005201              INC    R1                ;INCR CYCLE COUNT
1038 011150 000765              BR     9$
1039 011152 004737 006430      12$: JSR    PC,ISIRDY      ;CHK ICIR = 1 IRDY = 1
1040 011156 000003              3
1041 011160 062766 000002 000002 ADD    #2,2(SP)      ;FIX RETURN PC
1042 011166 005037 002352      CLR    SUBRPC        ;CLEAR SUBR CALL PC
1043 011172 012601              MOV    (SP)+,R1      ;RESTORE R1
1044 011174 000207              RTS     PC            ;RETURN
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057 011176 013746 002400      CKTBIT: MOV    REGNUM,-(SP)    ;SAVE LU REG NO.
1058 011202 010146              MOV    R1,-(SP)        ;SAVE R1
1059 011204 016637 000004 002352 MOV    4(SP),SUBRPC    ;GET PC OF SUBROUTINE CALL
1060 011212 162737 000004 002352 SUB     #4,SUBRPC
1061 011220 012701 000001      MOV    #BIT0,R1        ;INIT BIT POINTER
1062 011224 012737 000017 002400 MOV    #17,REGNUM      ;SET REG NO. = 17
1063 011232 004737 005254      4$: JSR    PC,STPLU       ;CLOCK LINE UNIT 1 CYCLE
1064 011236 000001              1
1065 011240 004737 003672      JSR    PC,READLU      ;READ REG 17
1066 011244 030176 000004      BIT     R1,#4(SP)      ;SEE IF EXPECTED BIT = 1
1067 011250 001013              BNE    9$            ;BR IF YES
1068 011252 132737 000040 002364 BITB    #TXDATA,REDBYT ;SEE IF ACTUAL BIT = 0
1069 011260 001422              BEQ     12$          ;BR IF YES
1070 011262 004737 004526      JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
1071
1072 011266              ;REPORT TXDATA NOT CLEARED
1073 011266 104455              ERRDF    32,EM32,ERR4
1074 011270 000040
1075 011272 013246
1076 011274 015622
1073 011276 000420
1074 011300 132737 000040 002364 9$: BR     20$          ;TAKE ERROR RETURN
1075 011306 001007              BITB    #TXDATA,REDBYT ;SEE IF ACTUAL BIT = 1
1076 011310 004737 004526      BNE    12$          ;BR IF YES
1077
1078 011314              ;REPORT TXDATA BIT NOT SET
1079 011314 104455              ERRDF    33,EM33,ERR4
1080 011316 000041
1081 011320 013275
1082 011322 015622
1079 011324 000405
1080 011326 006301
1081 011330 020127 000400      12$: BR     20$          ;TAKE ERROR EXIT
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
```

1082 011334 001336 BNE 4\$;BR IF NO
1083 011336 000405 BR 22\$
1084 011340 013706 002346 20\$: MOV PSTACK,SP ;RESTORE PROGRAM STACK POINTER TO BASE LEVEL
1085 011344 013746 002362 MOV RETADR,-(SP) ;FIX UP RETURN PC
1086 011350 000406 BR 40\$
1087 011352 062766 000002 000004 22\$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
1088 011360 012601 MOV (SP)+,R1 ;RESTORE R1
1089 011362 012637 002400 MOV (SP)+,REGNUM ;RESTORE REG NO.
1090 011366 005037 002352 40\$: CLR SUBRPC ;CLEAR SUBR CALL PC
1091 011372 000207 RTS PC ;RETURN

1092
1093
1094
1095
1096
1097

;* LDMSG1 - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH MSG1, AND LOADS
;* THE DATA CHARS INTO THE RCV MSG BUFFER (RCVBUF:), AS EXPECTED DATA
;* FOR LATER COMPARISON.

1101
1102 011374 010146 LDMSG1: MOV R1,-(SP) ;SAVE R1
1103 011376 010246 MOV R2,-(SP) ;SAVE R2
1104 011400 004737 011016 JSR PC,LODMSG ;LOAD MSG1 INTO TX SILO
1105 011404 003220 MSG1
1106 011406 000011 9.
1107 011410 012701 003224 MOV #MSG1+4,R1 ;GET POINTER TO MSG1
1108 011414 012702 003262 MOV #RCVBUF,R2 ;GET POINTER TO MSG BUF
1109 011420 012122 3\$: MOV (R1)+,(R2)+ ;LOAD A CHAR INTO MSG BUF
1110 011422 020127 003236 CMP R1,#MSG1+14. ;SEE IF DID LAST DATA CHAR YET
1111 011426 103774 BLO 3\$;BR IF NOT
1112 011430 052762 100400 177776 BIS #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
1113 011436 012726 000160 MOV #160,(SP)+ ;LOAD HI CRC BYTE
1114 011442 012726 000034 MOV #034,(SP)+ ;LOAD LO CRC BYTE
1115 011446 012602 MOV (SP)+,R2 ;RESTORE R2
1116 011450 012601 MOV (SP)+,R1 ;RESTORE R1
1117 011452 000207 RTS PC ;RETURN

1118
1119
1120
1121
1122

```
1      .SBTTL  GLOBAL ERROR REPORT SECTION
2
3      :///////////////////////////////////////////////////
4      ://      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
5      ://      THAT ARE USED IN MORE THAN ONE TEST.
6      :///////////////////////////////////////////////////
7
8
9 011454      045      124      045  FMT1:  .ASCIIZ  /%T%06%N/
011457      117      066      045
011462      116      000
10 011464      045      116      045  FMT2:  .ASCIIZ  /%N%AFAILING REG: /
011467      101      106      101
011472      111      114      111
011475      116      107      040
011500      122      105      107
011503      072      040      000
11 011506      045      101      105  FMT3:  .ASCIIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%N/
011511      130      120      105
011514      103      124      105
011517      104      072      040
011522      045      117      063
011525      045      123      065
011530      045      101      101
011533      103      124      125
011536      101      114      072
011541      040      045      117
011544      063      045      116
011547      000
12 011550      045      116      045  FMT4:  .ASCIIZ  /%N%T%N%T%N/
011553      124      045      116
011556      045      124      045
011561      116      000
13 011563      045      117      063  FMT5:  .ASCIIZ  /%03%S5%03%S5%03%S5%03%N/
011566      045      123      065
011571      045      117      063
011574      045      123      065
011577      045      117      063
011602      045      123      065
011605      045      117      063
011610      045      116      000
14 011613      045      123      064  FMT6:  .ASCIIZ  /%S4%03%S5%03%S5%03%S5%03%N/
011616      045      117      063
011621      045      123      065
011624      045      117      063
011627      045      123      065
011632      045      117      063
011635      045      123      065
011640      045      117      063
011643      045      116      000
15 011646      045      124      045  FMT7:  .ASCIIZ  /%T%02%N/
011651      117      062      045
011654      116      000
16 011656      045      101      105  FMT8:  .ASCIIZ  /%AEXTENDED REG AX%01%A-%T%N/
011661      130      124      105
011664      116      104      105
011667      104      040      122
```

	011672	105	107	040	
	011675	101	130	045	
	011700	117	061	045	
	011703	101	055	045	
	011706	124	045	116	
	011711	000			
17	011712	045	124	045	FMT9: .ASCIZ /%T%N/
	011715	116	000		
18	011717	045	101	120	FMT10: .ASCIZ /%APC OF SUBR CALL: %06%N/
	011722	103	040	117	
	011725	106	040	123	
	011730	125	102	122	
	011733	040	103	101	
	011736	114	114	072	
	011741	040	045	117	
	011744	066	045	116	
	011747	000			
19	011750	045	101	122	FMT11: .ASCIZ /%AREG %02%A LOADED WITH: %03%N/
	011753	105	107	040	
	011756	045	117	062	
	011761	045	101	040	
	011764	114	117	101	
	011767	104	105	104	
	011772	040	127	111	
	011775	124	110	072	
	012000	040	045	117	
	012003	063	045	116	
	012006	000			
20	012007	045	116	045	FMT19: .ASCIZ /%N%ATEST %D2%A NOT RUN%N/
	012012	101	124	105	
	012015	123	124	040	
	012020	045	104	062	
	012023	045	101	040	
	012026	116	117	124	
	012031	040	122	125	
	012034	116	045	116	
	012037	000			
21	012040	045	116	045	FMT24: .ASCIZ /%N%APLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON%N/
	012043	101	120	114	
	012046	105	101	123	
	012051	105	040	111	
	012054	116	123	125	
	012057	122	105	040	
	012062	115	070	062	
	012065	060	067	040	
	012070	122	125	116	
	012073	040	123	127	
	012076	111	124	103	
	012101	110	040	050	
	012104	105	062	070	
	012107	040	123	127	
	012112	067	051	040	
	012115	111	123	040	
	012120	117	116	045	
	012123	116	000		

24					
25	012125	103	123	122	EM1: .ASCIZ /CSR ADDRESS TIME-OUT (SELO)/
	012130	040	101	104	
	012133	104	122	105	
	012136	123	123	040	
	012141	124	111	115	
	012144	105	055	117	
	012147	125	124	040	
	012152	050	123	105	
	012155	114	060	051	
	012160	000			
26	012161	122	105	107	EM2: .ASCIZ /REG NOT INITIALIZED BY MST CLR/
	012164	040	116	117	
	012167	124	040	111	
	012172	116	111	124	
	012175	111	101	114	
	012200	111	132	105	
	012203	104	040	102	
	012206	131	040	115	
	012211	123	124	040	
	012214	103	114	122	
	012217	000			
27	012220	122	105	107	EM3: .ASCIZ /REG MISCOMPARE/
	012223	040	115	111	
	012226	123	103	117	
	012231	115	120	101	
	012234	122	105	000	
28	012237	122	105	107	EM4: .ASCIZ /REG NOT INITIALIZED BY UNIBUS RESET (INIT)/
	012242	040	116	117	
	012245	124	040	111	
	012250	116	111	124	
	012253	111	101	114	
	012256	111	132	105	
	012261	104	040	102	
	012264	131	040	125	
	012267	116	111	102	
	012272	125	123	040	
	012275	122	105	123	
	012300	105	124	040	
	012303	050	111	116	
	012306	111	124	051	
	012311	000			
29	012312	115	101	111	EM5: .ASCIZ /MAINT CLK BIT STUCK AT 0/
	012315	116	124	040	
	012320	103	114	113	
	012323	040	102	111	
	012326	124	040	123	
	012331	124	125	103	
	012334	113	040	101	
	012337	124	040	060	
	012342	000			
30	012343	115	101	111	EM6: .ASCIZ /MAINT CLK BIT STUCK AT 1/
	012346	116	124	040	
	012351	103	114	113	
	012354	040	102	111	
	012357	124	040	123	
	012362	124	125	103	

	012365	113	040	101	
	012370	124	040	061	
	012373	000			
31	012374	117	122	104	EM7: .ASCIZ /ORDY NOT SET/
	012377	131	040	116	
	012402	117	124	040	
	012405	123	105	124	
	012410	000			
32	012411	117	122	104	EM8: .ASCIZ /ORDY NOT CLEARED/
	012414	131	040	116	
	012417	117	124	040	
	012422	103	114	105	
	012425	101	122	105	
	012430	104	000		
33	012432	117	103	117	EM9: .ASCIZ /OCOR NOT SET/
	012435	122	040	116	
	012440	117	124	040	
	012443	123	105	124	
	012446	000			
34	012447	117	103	117	EM10: .ASCIZ /OCOR NOT CLEARED/
	012452	122	040	116	
	012455	117	124	040	
	012460	103	114	105	
	012463	101	122	105	
	012466	104	000		
35	012470	117	101	103	EM11: .ASCIZ /OACT NOT SET/
	012473	124	040	116	
	012476	117	124	040	
	012501	123	105	124	
	012504	000			
36	012505	117	101	103	EM12: .ASCIZ /OACT NOT CLEARED/
	012510	124	040	116	
	012513	117	124	040	
	012516	103	114	105	
	012521	101	122	105	
	012524	104	000		
37	012526	125	116	122	EM13: .ASCIZ /UNRR NOT CLEARED BY SOM/
	012531	122	040	116	
	012534	117	124	040	
	012537	103	114	105	
	012542	101	122	105	
	012545	104	040	102	
	012550	131	040	123	
	012553	117	115	000	
38	012556	125	116	122	EM14: .ASCIZ /UNRR NOT SET/
	012561	122	040	116	
	012564	117	124	040	
	012567	123	105	124	
	012572	000			
39	012573	125	116	122	EM15: .ASCIZ /UNRR NOT CLEARED BY OC/
	012576	122	040	116	
	012601	117	124	040	
	012604	103	114	105	
	012607	101	122	105	
	012612	104	040	102	
	012615	131	040	117	
	012620	103	000		

40	012622	125	116	122	EM16: .ASCIZ /UNRR NOT CLEARED/
	012625	122	040	116	
	012630	117	124	040	
	012633	103	114	105	
	012636	101	122	105	
	012641	104	000		
41	012643	111	122	104	EM17: .ASCIZ /IRDY NOT SET/
	012646	131	040	116	
	012651	117	124	040	
	012654	123	105	124	
	012657	000			
42	012660	111	122	104	EM18: .ASCIZ /IRDY NOT CLEARED/
	012663	131	040	116	
	012666	117	124	040	
	012671	103	114	105	
	012674	101	122	105	
	012677	104	000		
43	012701	111	103	111	EM19: .ASCIZ /ICIR NOT SET/
	012704	122	040	116	
	012707	117	124	040	
	012712	123	105	124	
	012715	000			
44	012716	111	103	111	EM20: .ASCIZ /ICIR NOT CLEARED/
	012721	122	040	116	
	012724	117	124	040	
	012727	103	114	105	
	012732	101	122	105	
	012735	104	000		
45	012737	111	101	103	EM21: .ASCIZ /IACT NOT SET/
	012742	124	040	116	
	012745	117	124	040	
	012750	123	105	124	
	012753	000			
46	012754	111	101	103	EM22: .ASCIZ /IACT NOT CLEARED/
	012757	124	040	116	
	012762	117	124	040	
	012765	103	114	105	
	012770	101	122	105	
	012773	104	000		
47	012775	104	123	123	EM23: .ASCIZ /DSSI NOT CLEARED/
	013000	111	040	116	
	013003	117	124	040	
	013006	103	114	105	
	013011	101	122	105	
	013014	104	000		
48	013016	104	123	123	EM24: .ASCIZ /DSSI NOT SET/
	013021	111	040	116	
	013024	117	124	040	
	013027	123	105	124	
	013032	000			
49	013033	104	123	123	EM25: .ASCIZ /DSSI NOT CLEARED BY MST CLR/
	013036	111	040	116	
	013041	117	124	040	
	013044	103	114	105	
	013047	101	122	105	
	013052	104	040	102	
	013055	131	040	115	

	013060	123	124	040	
	013063	103	114	122	
	013066	000			
50	013067	111	116	103	EM26: .ASCIIZ /INCORRECT DATA CHAR RCV'D/
	013072	117	122	122	
	013075	105	103	124	
	013100	040	104	101	
	013103	124	101	040	
	013106	103	110	101	
	013111	122	040	122	
	013114	103	126	047	
	013117	104	000		
51	013121	111	116	103	EM27: .ASCIIZ /INCORRECT CRC BYTE RCV'D/
	013124	117	122	122	
	013127	105	103	124	
	013132	040	103	122	
	013135	103	040	102	
	013140	131	124	105	
	013143	040	122	103	
	013146	126	047	104	
	013151	000			
52	013152	122	123	117	EM28: .ASCIIZ /RSOM NOT CLEARED/
	013155	115	040	116	
	013160	117	124	040	
	013163	103	114	105	
	013166	101	122	105	
	013171	104	000		
53	013173	122	123	117	EM29: .ASCIIZ /RSOM NOT SET/
	013176	115	040	116	
	013201	117	124	040	
	013204	123	105	124	
	013207	000			
54	013210	122	105	117	EM30: .ASCIIZ /REOM NOT CLEARED/
	013213	115	040	116	
	013216	117	124	040	
	013221	103	114	105	
	013224	101	122	105	
	013227	104	000		
55	013231	122	105	117	EM31: .ASCIIZ /REOM NOT SET/
	013234	115	040	116	
	013237	117	124	040	
	013242	123	105	124	
	013245	000			
56	013246	124	130	104	EM32: .ASCIIZ /TXDATA BIT NOT CLEARED/
	013251	101	124	101	
	013254	040	102	111	
	013257	124	040	116	
	013262	117	124	040	
	013265	103	114	105	
	013270	101	122	105	
	013273	104	000		
57	013275	124	130	104	EM33: .ASCIIZ /TXDATA BIT NOT SET/
	013300	101	124	101	
	013303	040	102	111	
	013306	124	040	116	
	013311	117	124	040	
	013314	123	105	124	

58	013317	000			
	013320	122	103	126	EM34: .ASCIZ /RCV'D DATA MISCOMPARE/
	013323	047	104	040	
	013326	104	101	124	
	013331	101	040	115	
	013334	111	123	103	
	013337	117	115	120	
	013342	101	122	105	
	013345	000			
59	013346	102	103	103	EM35: .ASCIZ /BCL NOT CLEARED/
	013351	040	116	117	
	013354	124	040	103	
	013357	114	105	101	
	013362	122	105	104	
	013365	000			
60	013366	102	103	103	EM36: .ASCIZ /BCC NOT SET/
	013371	040	116	117	
	013374	124	040	123	
	013377	105	124	000	
61	013402	105	102	114	EM37: .ASCIZ /EBLK NOT CLEARED/
	013405	113	040	116	
	013410	117	124	040	
	013413	103	114	105	
	013416	101	122	105	
	013421	104	000		
62	013423	105	102	114	EM38: .ASCIZ /EBLK NOT SET/
	013426	113	040	116	
	013431	117	124	040	
	013434	123	105	124	
	013437	000			
63	013440	122	101	102	EM39: .ASCIZ /RAB NOT CLEARED/
	013443	040	116	117	
	013446	124	040	103	
	013451	114	105	101	
	013454	122	105	104	
	013457	000			
64	013460	122	101	102	EM40: .ASCIZ /RAB NOT SET/
	013463	040	116	117	
	013466	124	040	123	
	013471	105	124	000	
65	013474	117	126	122	EM41: .ASCIZ /OVRR NOT CLEARED/
	013477	122	040	116	
	013502	117	124	040	
	013505	103	114	105	
	013510	101	122	105	
	013513	104	000		
66	013515	117	126	122	EM42: .ASCIZ /OVRR NOT SET/
	013520	122	040	116	
	013523	117	124	040	
	013526	123	105	124	
	013531	000			
67	013532	123	127	040	EM43: .ASCIZ /SW PACK #1 INCORRECT/
	013535	120	101	103	
	013540	113	040	043	
	013543	061	040	111	
	013546	116	103	117	
	013551	122	122	105	

68	013554	103	124	000	
	013557	123	127	040	EM44: .ASCIIZ /SW PACK #2 INCORRECT/
	013562	120	101	103	
	013565	113	040	043	
	013570	062	040	111	
	013573	116	103	117	
	013576	122	122	105	
	013601	103	124	000	
69	013604	123	127	040	EM45: .ASCIIZ /SW PACK #3 INCORRECT/
	013607	120	101	103	
	013612	113	040	043	
	013615	063	040	111	
	013620	116	103	117	
	013623	122	122	105	
	013626	103	124	000	
70	013631	122	103	126	EM46: .ASCIIZ /RCV SILO NOT CLEARED BY IC/
	013634	040	123	111	
	013637	114	117	040	
	013642	116	117	124	
	013645	040	103	114	
	013650	105	101	122	
	013653	105	104	040	
	013656	102	131	040	
	013661	111	103	000	
71	013664	101	123	123	EM47: .ASCIIZ /ASSEMB BIT COUNT INCORRECT/
	013667	105	115	102	
	013672	040	102	111	
	013675	124	040	103	
	013700	117	125	116	
	013703	124	040	111	
	013706	116	103	117	
	013711	122	122	105	
	013714	103	124	000	
72	013717	117	104	104	EM48: .ASCIIZ /ODD VRC PARITY BIT NOT SET/
	013722	040	126	122	
	013725	103	040	120	
	013730	101	122	111	
	013733	124	131	040	
	013736	102	111	124	
	013741	040	116	117	
	013744	124	040	123	
	013747	105	124	000	
73	013752	117	104	104	EM49: .ASCIIZ /ODD VRC PARITY BIT NOT CLEARED/
	013755	040	126	122	
	013760	103	040	120	
	013763	101	122	111	
	013766	124	131	040	
	013771	102	111	124	
	013774	040	116	117	
	013777	124	040	103	
	014002	114	105	101	
	014005	122	105	104	
	014010	000			
74	014011	105	126	105	EM50: .ASCIIZ /EVEN VRC PARITY BIT NOT SET/
	014014	116	040	126	
	014017	122	103	040	
	014022	120	101	122	

	014025	111	124	131	
	014030	040	102	111	
	014033	124	040	116	
	014036	117	124	040	
	014041	123	105	124	
	014044	000			
75	014045	105	126	105	EM51: .ASCIIZ /EVEN VRC PARITY BIT NOT CLEARED/
	014050	116	040	126	
	014053	122	103	040	
	014056	120	101	122	
	014061	111	124	131	
	014064	040	102	111	
	014067	124	040	116	
	014072	117	124	040	
	014075	103	114	105	
	014100	101	122	105	
	014103	104	000		
76	014105	122	105	101	EM52: .ASCIIZ /READY NOT SET AFTER AX REG WRITE/
	014110	104	131	040	
	014113	116	117	124	
	014116	040	123	105	
	014121	124	040	101	
	014124	106	124	105	
	014127	122	040	101	
	014132	130	040	122	
	014135	105	107	040	
	014140	127	122	111	
	014143	124	105	000	
77	014146	122	105	101	EM53: .ASCIIZ /READY NOT SET AFTER AX REG READ/
	014151	104	131	040	
	014154	116	117	124	
	014157	040	123	105	
	014162	124	040	101	
	014165	106	124	105	
	014170	122	040:	101	
	014173	130	040	122	
	014176	105	107	040	
	014201	122	105	101	
	014204	104	000		
78	014206	124	130	040	EM54: .ASCIIZ /TX UNDERRUN ERROR/
	014211	125	116	104	
	014214	105	122	122	
	014217	125	116	040	
	014222	105	122	122	
	014225	117	122	000	
79	014230	122	124	123	EM60: .ASCIIZ /RTS NOT SET/
	014233	040	116	117	
	014236	124	040	123	
	014241	105	124	000	
80	014244	122	124	123	EM65: .ASCIIZ /RTS NOT CLEARED/
	014247	040	116	117	
	014252	124	040	103	
	014255	114	105	101	
	014260	122	105	104	
	014263	000			

81
82

83					
84	014264	111	116	102	DH1: .ASCIZ &INBUS/OUTBUS REG 8
	014267	125	123	057	
	014272	117	125	124	
	014275	102	125	123	
	014300	040	122	105	
	014303	107	040	000	
85	014306	114	111	116	DH2: .ASCIZ /LINE UNIT INBUS REGS :/
	014311	105	040	125	
	014314	116	111	124	
	014317	040	111	116	
	014322	102	125	123	
	014325	040	122	105	
	014330	107	123	040	
	014333	072	000		
86	014335	122	105	107	DH3: .ASCIZ /REG10 REG11 REG12 REG13/
	014340	061	060	040	
	014343	040	040	122	
	014346	105	107	061	
	014351	061	040	040	
	014354	040	122	105	
	014357	107	061	062	
	014362	040	040	040	
	014365	122	105	107	
	014370	061	063	000	
87	014373	040	040	040	DH4: .ASCIZ / REG14 REG15 REG16 REG17/
	014376	040	122	105	
	014401	107	061	064	
	014404	040	040	040	
	014407	122	105	107	
	014412	061	065	040	
	014415	040	040	122	
	014420	105	107	061	
	014423	066	040	040	
	014426	040	122	105	
	014431	107	061	067	
	014434	000			
88	014435	061	065	000	DH5: .ASCIZ /15/
89	014440	061	066	000	DH6: .ASCIZ /16/
90	014443	114	111	116	DH7: .ASCIZ /LINE UNIT EXTENDED REGS :/
	014446	105	040	125	
	014451	116	111	124	
	014454	040	105	130	
	014457	124	105	116	
	014462	104	105	104	
	014465	040	122	105	
	014470	107	123	040	
	014473	072	000		
91	014475	101	130	060	DH8: .ASCIZ /AX0-15 AX0-16 AX1-15 AX1-16/
	014500	055	061	065	
	014503	040	040	101	
	014506	130	060	055	
	014511	061	066	040	
	014514	040	101	130	
	014517	061	055	061	
	014522	065	040	040	
	014525	101	130	061	

	014530	055	361	066	
	014533	000			
92	014534	040	040	040	DH9: .ASCIZ / AX2-15 AX2-16 AX3-15 AX3-16/
	014537	040	101	130	
	014542	062	055	061	
	014545	065	040	040	
	014550	101	130	062	
	014553	055	061	066	
	014556	040	040	101	
	014561	130	063	055	
	014564	061	065	040	
	014567	040	101	130	
	014572	063	055	061	
	014575	066	000		

93
94
95
96
97
98
99

.EVEN

100	014600			BGNMSG	ERR1	
	014600			PRINTB	#FMT1,#ADDRES,MPCSR	
101	014600					
	014600	013746	002446			
	014604	012746	035620			
	014610	012746	011454			
	014614	012746	000003			
	014620	010600				
	014622	104414				
	014624	062706	000010			
102	014630			ENDMSG		
	014630					
	014630	104423				

ERR1::

MOV MPCSR,-(SP)
MOV #ADDRES,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP

L10002:

TRAP C\$MSG

103						
104						
105						
106	014632			BGNMSG	ERR2	
	014632			PRINTB	#FMT1,#ADDRES,MPCSR	
107	014632					
	014632	013746	002446			
	014636	012746	035620			
	014642	012746	011454			
	014646	012746	000003			
	014652	010600				
	014654	104414				
	014656	062706	000010			
108	014662			PRINTB	#FMT2	
	014662	012746	011464			
	014666	012746	000001			
	014672	010600				
	014674	104414				
	014676	062706	000004			
109	014702			PRINTB	#FMT7,#DH1,REGNUM	
	014702	013746	002400			
	014706	012746	014264			
	014712	012746	011646			

ERR2::

MOV MPCSR,-(SP)
MOV #ADDRES,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP

MOV #FMT2,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

MOV REGNUM,-(SP)
MOV #DH1,-(SP)
MOV #FMT7,-(SP)

```
014716 012746 000003
014722 010600
014724 104414
014726 062706 000010
110 014732 PRINTB #FMT3,GOODAT,BADDAT
014732 013746 002406
014736 013746 002404
014742 012746 011506
014746 012746 000003
014752 010600
014754 104414
014756 062706 000010
111 014762 PRINTX #FMT4,#DH2,#DH3
014762 012746 014335
014766 012746 014306
014772 012746 011550
014776 012746 000003
015002 010600
015004 104415
015006 062706 000010
112 015012 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
015012 013746 002310
015016 013746 002306
015022 013746 002304
015026 013746 002302
015032 012746 011563
015036 012746 000005
015042 010600
015044 104415
015046 062706 000014
113 015052 PRINTX #FMT9,#DH4
015052 012746 014373
015056 012746 011712
015062 012746 000002
015066 010600
015070 104415
015072 062706 000006
114 015076 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
015076 013746 002320
015102 013746 002316
015106 013746 002314
015112 013746 002312
015116 012746 011613
015122 012746 000005
015126 010600
015130 104415
015132 062706 000014
115 015136 ENDMSG
015136
015136 104423
116
117
118
119
120
121 015140 BGNMSG ERR3
015140
```

```
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP

MOV BADDAT,-(SP)
MOV GOODAT,-(SP)
MOV #FMT3,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP

MOV #DH3,-(SP)
MOV #DH2,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #10,SP

MOV LUR13,-(SP)
MOV LUR12,-(SP)
MOV LUR11,-(SP)
MOV LUR10,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP

MOV #DH4,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #6,SP

MOV LUR17,-(SP)
MOV LUR16,-(SP)
MOV LUR15,-(SP)
MOV LUR14,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP
```

L10003:

TRAP C\$MSG

ERR3::

122	015140		PRINTB	#FMT1,#ADDRES,MPCSR		MOV	MPCSR,-(SP)
	015140	013746				MOV	#ADDRES,-(SP)
	015144	012746				MOV	#FMT1,-(SP)
	015150	012746				MOV	#3,-(SP)
	015154	012746				MOV	SP,R0
	015160	010600				TRAP	C\$PNTB
	015162	104414				ADD	#10,SP
	015164	062706					
123	015170		PRINTB	#FMT2		MOV	#FMT2,-(SP)
	015170	012746				MOV	#1,-(SP)
	015174	012746				MOV	SP,R0
	015200	010600				TRAP	C\$PNTB
	015202	104414				ADD	#4,SP
	015204	062706					
124	015210		PRINTB	#FMT8,TMP1,TMP0		MOV	TMP0,-(SP)
	015210	013746				MOV	TMP1,-(SP)
	015214	013746				MOV	#FMT8,-(SP)
	015220	012746				MOV	#3,-(SP)
	015224	012746				MOV	SP,R0
	015230	010600				TRAP	C\$PNTB
	015232	104414				ADD	#10,SP
	015234	062706					
125	015240		PRINTB	#FMT3,GOODAT,BADDAT		MOV	BADDAT,-(SP)
	015240	013746				MOV	GOODAT,-(SP)
	015244	013746				MOV	#FMT3,-(SP)
	015250	012746				MOV	#3,-(SP)
	015254	012746				MOV	SP,R0
	015260	010600				TRAP	C\$PNTB
	015262	104414				ADD	#10,SP
	015264	062706					
126	015270		PRINTX	#FMT4,#DH2,#DH3		MOV	#DH3,-(SP)
	015270	012746				MOV	#DH2,-(SP)
	015274	012746				MOV	#FMT4,-(SP)
	015300	012746				MOV	#3,-(SP)
	015304	012746				MOV	SP,R0
	015310	010600				TRAP	C\$PNTX
	015312	104415				ADD	#10,SP
	015314	062706					
127	015320		PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13		MOV	LUR13,-(SP)
	015320	013746				MOV	LUR12,-(SP)
	015324	013746				MOV	LUR11,-(SP)
	015330	013746				MOV	LUR10,-(SP)
	015334	013746				MOV	#FMT5,-(SP)
	015340	012746				MOV	#5,-(SP)
	015344	012746				MOV	SP,R0
	015350	010600				TRAP	C\$PNTX
	015352	104415				ADD	#14,SP
	015354	062706					
128	015360		PRINTX	#FMT9,#DH4		MOV	#DH4,-(SP)
	015360	012746				MOV	#FMT9,-(SP)
	015364	012746				MOV	#2,-(SP)
	015370	012746				MOV	SP,R0
	015374	010600				TRAP	C\$PNTX
	015376	104415				ADD	#6,SP
	015400	062706					
129	015404		PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17		MOV	LUR17,-(SP)
	015404	013746					

015410 013746 002316
015414 013746 002314
015420 013746 002312
015424 012746 011613
015430 012746 000005
015434 010600
015436 104415
015440 062706 000014

130 015444
015444 012746 014475
015450 012746 014443
015454 012746 011550
015460 012746 000003
015464 010600
015466 104415
015470 062706 000010

131 015474
015474 013746 002330
015500 013746 002326
015504 013746 002324
015510 013746 002322
015514 012746 011563
015520 012746 000005
015524 010600
015526 104415
015530 062706 000014

132 015534
015534 012746 014534
015540 012746 011712
015544 012746 000002
015550 010600
015552 104415
015554 062706 000006

133 015560
015560 013746 002340
015564 013746 002336
015570 013746 002334
015574 013746 002332
015600 012746 011613
015604 012746 000005
015610 010600
015612 104415
015614 062706 000014

134 015620
015620
015620 104423

135
136
137
138
139
140 015622
015622
141 015622
015622 013746 002352
015626 012746 011717
015632 012746 000002

PRINTX #FMT4,#DH7,#DH8

PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16

PRINTX #FMT9,#DH9

PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16

ENDMSG

BGNMSG ERR4

PRINTB #FMT10,SUBRPC

MOV LUR16,-(SP)
MOV LUR15,-(SP)
MOV LUR14,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP

MOV #DH8,-(SP)
MOV #DH7,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #10,SP

MOV AX1.16,-(SP)
MOV AX1.15,-(SP)
MOV AX0.16,-(SP)
MOV AX0.15,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP

MOV #DH9,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #6,SP

MOV AX3.16,-(SP)
MOV AX3.15,-(SP)
MOV AX2.16,-(SP)
MOV AX2.15,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP

L10004:

TRAP C\$MSG

ERR4::

MOV SUBRPC,-(SP)
MOV #FMT10,-(SP)
MOV #2,-(SP)

	015636	010600			MOV	SP,R0
	015640	104414			TRAP	C\$PNTB
	015642	062706	000006		ADD	#6,SP
142	015646			PRINTB	#FMT1,#ADDRES,MPCSR	
	015646	013746	002446		MOV	MPCSR,-(SP)
	015652	012746	035620		MOV	#ADDRES,-(SP)
	015656	012746	011454		MOV	#FMT1,-(SP)
	015662	012746	000003		MOV	#3,-(SP)
	015666	010600			MOV	SP,R0
	015670	104414			TRAP	C\$PNTB
	015672	062706	000010		ADD	#10,SP
143	015676			PRINTB	#FMT2	
	015676	012746	011464		MOV	#FMT2,-(SP)
	015702	012746	000001		MOV	#1,-(SP)
	015706	010600			MOV	SP,R0
	015710	104414			TRAP	C\$PNTB
	015712	062706	000004		ADD	#4,SP
144	015716			PRINTB	#FMT7,#DH1,REGNUM	
	015716	013746	002400		MOV	REGNUM,-(SP)
	015722	012746	014264		MOV	#DH1,-(SP)
	015726	012746	011646		MOV	#FMT7,-(SP)
	015732	012746	000003		MOV	#3,-(SP)
	015736	010600			MOV	SP,R0
	015740	104414			TRAP	C\$PNTB
	015742	062706	000010		ADD	#10,SP
145	015746			PRINTX	#FMT4,#DH2,#DH3	
	015746	012746	014335		MOV	#DH3,-(SP)
	015752	012746	014306		MOV	#DH2,-(SP)
	015756	012746	011550		MOV	#FMT4,-(SP)
	015762	012746	000003		MOV	#3,-(SP)
	015766	010600			MOV	SP,R0
	015770	104415			TRAP	C\$PNTX
	015772	062706	000010		ADD	#10,SP
146	015776			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	015776	013746	002310		MOV	LUR13,-(SP)
	016002	013746	002306		MOV	LUR12,-(SP)
	016006	013746	002304		MOV	LUR11,-(SP)
	016012	013746	002302		MOV	LUR10,-(SP)
	016016	012746	011563		MOV	#FMT5,-(SP)
	016022	012746	000005		MOV	#5,-(SP)
	016026	010600			MOV	SP,R0
	016030	104415			TRAP	C\$PNTX
	016032	062706	000014		ADD	#14,SP
147	016036			PRINTX	#FMT9,#DH4	
	016036	012746	014373		MOV	#DH4,-(SP)
	016042	012746	011712		MOV	#FMT9,-(SP)
	016046	012746	000002		MOV	#2,-(SP)
	016052	010600			MOV	SP,R0
	016054	104415			TRAP	C\$PNTX
	016056	062706	000006		ADD	#6,SP
148	016062			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	016062	013746	002320		MOV	LUR17,-(SP)
	016066	013746	002316		MOV	LUR16,-(SP)
	016072	013746	002314		MOV	LUR15,-(SP)
	016076	013746	002312		MOV	LUR14,-(SP)
	016102	012746	011613		MOV	#FMT6,-(SP)
	016106	012746	000005		MOV	#5,-(SP)

149	016112	010600				MOV	SP,R0
	016114	104415				TRAP	C\$PNTX
	016116	062706	000014			ADD	#14,SP
	016122			PRINTX	#FMT4,#DH7,#DH8		
	016122	012746	014475			MOV	#DH8,-(SP)
	016126	012746	014443			MOV	#DH7,-(SP)
	016132	012746	011550			MOV	#FMT4,-(SP)
	016136	012746	000003			MOV	#3,-(SP)
	016142	010600				MOV	SP,R0
	016144	104415				TRAP	C\$PNTX
	016146	062706	000010			ADD	#10,SP
150	016152			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16		
	016152	013746	002330			MOV	AX1.16,-(SP)
	016156	013746	002326			MOV	AX1.15,-(SP)
	016162	013746	002324			MOV	AX0.16,-(SP)
	016166	013746	002322			MOV	AX0.15,-(SP)
	016172	012746	011563			MOV	#FMT5,-(SP)
	016176	012746	000005			MOV	#5,-(SP)
	016202	010600				MOV	SP,R0
	016204	104415				TRAP	C\$PNTX
	016206	062706	000014			ADD	#14,SP
151	016212			PRINTX	#FMT9,#DH9		
	016212	012746	014534			MOV	#DH9,-(SP)
	016216	012746	011712			MOV	#FMT9,-(SP)
	016222	012746	000002			MOV	#2,-(SP)
	016226	010600				MOV	SP,R0
	016230	104415				TRAP	C\$PNTX
	016232	062706	000006			ADD	#6,SP
152	016236			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16		
	016236	013746	002340			MOV	AX3.16,-(SP)
	016242	013746	002336			MOV	AX3.15,-(SP)
	016246	013746	002334			MOV	AX2.16,-(SP)
	016252	013746	002332			MOV	AX2.15,-(SP)
	016256	012746	011613			MOV	#FMT6,-(SP)
	016262	012746	000005			MOV	#5,-(SP)
	016266	010600				MOV	SP,R0
	016270	104415				TRAP	C\$PNTX
	016272	062706	000014			ADD	#14,SP
153	016276			ENDMSG			
	016276					L10005:	
	016276	104423				TRAP	C\$MSG
154							
155							
156							
157							
158							
159	016300			BGNMSG	ERR5		
	016300					ERR5::	
160	016300			PRINTB	#FMT1,#ADDRES,MPCSR		
	016300	013746	002446			MOV	MPCSR,-(SP)
	016304	012746	035620			MOV	#ADDRES,-(SP)
	016310	012746	011454			MOV	#FMT1,-(SP)
	016314	012746	000003			MOV	#3,-(SP)
	016320	010600				MOV	SP,R0
	016322	104414				TRAP	C\$PNTB
	016324	062706	000010			ADD	#10,SP
161	016330			PRINTB	#FMT11,REGNUM,LOADAT		

	016330	013746	002410		MOV	LOADAT,-(SP)
	016334	013746	002400		MOV	REGNUM,-(SP)
	016340	012746	011750		MOV	#FMT11,-(SP)
	016344	012746	000003		MOV	#3,-(SP)
	016350	010600			MOV	SP,R0
	016352	104414			TRAP	C\$PNTB
	016354	062706	000010		ADD	#10,SP
162	016360			PRINTB	#FMT2	
	016360	012746	011464		MOV	#FMT2,-(SP)
	016364	012746	000001		MOV	#1,-(SP)
	016370	010600			MOV	SP,R0
	016372	104414			TRAP	C\$PNTB
	016374	062706	000004		ADD	#4,SP
163	016400			PRINTB	#FMT8,TMP1,TMP0	
	016400	013746	002530		MOV	TMP0,-(SP)
	016404	013746	002532		MOV	TMP1,-(SP)
	016410	012746	011656		MOV	#FMT8,-(SP)
	016414	012746	000003		MOV	#3,-(SP)
	016420	010600			MOV	SP,R0
	016422	104414			TRAP	C\$PNTB
	016424	062706	000010		ADD	#10,SP
164	016430			PRINTB	#FMT3,GOODAT,BADDAT	
	016430	013746	002406		MOV	BADDAT,-(SP)
	016434	013746	002404		MOV	GOODAT,-(SP)
	016440	012746	011506		MOV	#FMT3,-(SP)
	016444	012746	000003		MOV	#3,-(SP)
	016450	010600			MOV	SP,R0
	016452	104414			TRAP	C\$PNTB
	016454	062706	000010		ADD	#10,SP
165	016460			PRINTX	#FMT4,#DH2,#DH3	
	016460	012746	014335		MOV	#DH3,-(SP)
	016464	012746	014306		MOV	#DH2,-(SP)
	016470	012746	011550		MOV	#FMT4,-(SP)
	016474	012746	000003		MOV	#3,-(SP)
	016500	010600			MOV	SP,R0
	016502	104415			TRAP	C\$PNTX
	016504	062706	000010		ADD	#10,SP
166	016510			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	016510	013746	002310		MOV	LUR13,-(SP)
	016514	013746	002306		MOV	LUR12,-(SP)
	016520	013746	002304		MOV	LUR11,-(SP)
	016524	013746	002302		MOV	LUR10,-(SP)
	016530	012746	011563		MOV	#FMT5,-(SP)
	016534	012746	000005		MOV	#5,-(SP)
	016540	010600			MOV	SP,R0
	016542	104415			TRAP	C\$PNTX
	016544	062706	000014		ADD	#14,SP
167	016550			PRINTX	#FMT9,#DH4	
	016550	012746	014373		MOV	#DH4,-(SP)
	016554	012746	011712		MOV	#FMT9,-(SP)
	016560	012746	000002		MOV	#2,-(SP)
	016564	010600			MOV	SP,R0
	016566	104415			TRAP	C\$PNTX
	016570	062706	000006		ADD	#6,SP
168	016574			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	016574	013746	002320		MOV	LUR17,-(SP)
	016600	013746	002316		MOV	LUR16,-(SP)

016604	013746	002314		MOV	LUR15,-(SP)
016610	013746	002312		MOV	LUR14,-(SP)
016614	012746	011613		MOV	#FMT6,-(SP)
016620	012746	000005		MOV	#5,-(SP)
016624	010600			MOV	SP,R0
016626	104415			TRAP	C\$PNTX
016630	062706	000014		ADD	#14,SP
169 016634			PRINTX #FMT4,#DH7,#DH8		
016634	012746	014475		MOV	#DH8,-(SP)
016640	012746	014443		MOV	#DH7,-(SP)
016644	012746	011550		MOV	#FMT4,-(SP)
016650	012746	000003		MOV	#3,-(SP)
016654	010600			MOV	SP,R0
016656	104415			TRAP	C\$PNTX
016660	062706	000010		ADD	#10,SP
170 016664			PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16		
016664	013746	002330		MOV	AX1.16,-(SP)
016670	013746	002326		MOV	AX1.15,-(SP)
016674	013746	002324		MOV	AX0.16,-(SP)
016700	013746	002322		MOV	AX0.15,-(SP)
016704	012746	011563		MOV	#FMT5,-(SP)
016710	012746	000005		MOV	#5,-(SP)
016714	010600			MOV	SP,R0
016716	104415			TRAP	C\$PNTX
016720	062706	000014		ADD	#14,SP
171 016724			PRINTX #FMT9,#DH9		
016724	012746	014534		MOV	#DH9,-(SP)
016730	012746	011712		MOV	#FMT9,-(SP)
016734	012746	000002		MOV	#2,-(SP)
016740	010600			MOV	SP,R0
016742	104415			TRAP	C\$PNTX
016744	062706	000006		ADD	#6,SP
172 016750			PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16		
016750	013746	002340		MOV	AX3.16,-(SP)
016754	013746	002336		MOV	AX3.15,-(SP)
016760	013746	002334		MOV	AX2.16,-(SP)
016764	013746	002332		MOV	AX2.15,-(SP)
016770	012746	011613		MOV	#FMT6,-(SP)
016774	012746	000005		MOV	#5,-(SP)
017000	010600			MOV	SP,R0
017002	104415			TRAP	C\$PNTX
017004	062706	000014		ADD	#14,SP
173 017010			ENDMSG		
017010					
017010	104423				
174					
175					
176					
177					
178					
179 017012			BGNMSG ERR6		
017012					
180 017012			PRINTB #FMT10,SUBRPC		
017012	013746	002352		MOV	SUBRPC,-(SP)
017016	012746	011717		MOV	#FMT10,-(SP)
017022	012746	000002		MOV	#2,-(SP)
017026	010600			MOV	SP,R0

L10006:

TRAP C\$MSG

ERR6::

```
017030 104414
017032 062706 000006
181 017036 PRINTB #FMT1,#ADDRES,MPCSR
017036 013746 002446
017042 012746 035620
017046 012746 011454
017052 012746 000003
017056 010600
017060 104414
017062 062706 000010
182 017066 PRINTB #FMT2
017066 012746 011464
017072 012746 000001
017076 010600
017100 104414
017102 062706 000004
183 017106 PRINTB #FMT8,TMP1,TMP0
017106 013746 002530
017112 013746 002532
017116 012746 011656
017122 012746 000003
017126 010600
017130 104414
017132 062706 000010
184 017136 PRINTX #FMT4,#DH2,#DH3
017136 012746 014335
017142 012746 014306
017146 012746 011550
017152 012746 000003
017156 010600
017160 104415
017162 062706 000010
185 017166 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
017166 013746 002310
017172 013746 002306
017176 013746 002304
017202 013746 002302
017206 012746 011563
017212 012746 000005
017216 010600
017220 104415
017222 062706 000014
186 017226 PRINTX #FMT9,#DH4
017226 012746 014373
017232 012746 011712
017236 012746 000002
017242 010600
017244 104415
017246 062706 000006
187 017252 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
017252 013746 002320
017256 013746 002316
017262 013746 002314
017266 013746 002312
017272 012746 011613
017276 012746 000005
017302 010600
```

```
TRAP C$PNTB
ADD #6,SP
MOV MPCSR,-(SP)
MOV #ADDRES,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
MOV #FMT2,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
MOV TMP0,-(SP)
MOV TMP1,-(SP)
MOV #FMT8,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
MOV #DH3,-(SP)
MOV #DH2,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #10,SP
MOV LUR13,-(SP)
MOV LUR12,-(SP)
MOV LUR11,-(SP)
MOV LUR10,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP
MOV #DH4,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #6,SP
MOV LUR17,-(SP)
MOV LUR16,-(SP)
MOV LUR15,-(SP)
MOV LUR14,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
```

017304 104415
017306 062706 000014
188 017312
017312 012746 014475
017316 012746 014443
017322 012746 011550
017326 012746 000003
017332 010600
017334 104415
017336 062706 000010
189 017342
017342 013746 002330
017346 013746 002326
017352 013746 002324
017356 013746 002322
017362 012746 011563
017366 012746 000005
017372 010600
017374 104415
017376 062706 000014
190 017402
017402 012746 014534
017406 012746 011712
017412 012746 000002
017416 010600
017420 104415
017422 062706 000006
191 017426
017426 013746 002340
017432 013746 002336
017436 013746 002334
017442 013746 002332
017446 012746 011613
017452 012746 000005
017456 010600
017460 104415
017462 062706 000014
192 017466
017466
017466 104423
193
194
195
196
197
198 017470
017470
199 017470
017470 013746 002446
017474 012746 035620
017500 012746 011454
017504 012746 000003
017510 010600
017512 104414
017514 062706 000010
200 017520
017520 012746 011464

PRINTX #FMT4,#DH7,#DH8

PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16

PRINTX #FMT9,#DH9

PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16

ENDMSG

BGNMSG ERR7

PRINTB #FMT1,#ADDRES,MPCSR

PRINTB #FMT2

TRAP C\$PNTX
ADD #14,SPMOV #DH8,-(SP)
MOV #DH7,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #10,SPMOV AX1.16,-(SP)
MOV AX1.15,-(SP)
MOV AX0.16,-(SP)
MOV AX0.15,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SPMOV #DH9,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #6,SPMOV AX3.16,-(SP)
MOV AX3.15,-(SP)
MOV AX2.16,-(SP)
MOV AX2.15,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP

L10007:

TRAP C\$MSG

ERR7::

MOV MPCSR,-(SP)
MOV #ADDRES,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #FMT2,-(SP)

```
017524 012746 000001
017530 010600
017532 104414
017534 062706 000004
201 017540 PRINTB #FMT7,#DH1,REGNUM
017540 013746 002400
017544 012746 014264
017550 012746 011646
017554 012746 000003
017560 010600
017562 104414
017564 062706 000010
202 017570 PRINTX #FMT4,#DH2,#DH3
017570 012746 014335
017574 012746 014306
017600 012746 011550
017604 012746 000003
017610 010600
017612 104415
017614 062706 000010
203 017620 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
017620 013746 002310
017624 013746 002306
017630 013746 002304
017634 013746 002302
017640 012746 011563
017644 012746 000005
017650 010600
017652 104415
017654 062706 000014
204 017660 PRINTX #FMT9,#DH4
017660 012746 014373
017664 012746 011712
017670 012746 000002
017674 010600
017676 104415
017700 062706 000006
205 017704 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
017704 013746 002320
017710 013746 002316
017714 013746 002314
017720 013746 002312
017724 012746 011613
017730 012746 000005
017734 010600
017736 104415
017740 062706 000014
206 017744 PRINTX #FMT4,#DH7,#DH8
017744 012746 014475
017750 012746 014443
017754 012746 011550
017760 012746 000003
017764 010600
017766 104415
017770 062706 000010
207 017774 PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
017774 013746 002330
```

```
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP

MOV REGNUM,-(SP)
MOV #DH1,-(SP)
MOV #FMT7,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP

MOV #DH3,-(SP)
MOV #DH2,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #10,SP

MOV LUR13,-(SP)
MOV LUR12,-(SP)
MOV LUR11,-(SP)
MOV LUR10,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP

MOV #DH4,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #6,SP

MOV LUR17,-(SP)
MOV LUR16,-(SP)
MOV LUR15,-(SP)
MOV LUR14,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP

MOV #DH8,-(SP)
MOV #DH7,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #10,SP

MOV AX1.16,-(SP)
```

020000	013746	002326		MOV	AX1.15,-(SP)
020004	013746	002324		MOV	AX0.16,-(SP)
020010	013746	002322		MOV	AX0.15,-(SP)
020014	012746	011563		MOV	#FMT5,-(SP)
020020	012746	000005		MOV	#5,-(SP)
020024	010600			MOV	SP,R0
020026	104415			TRAP	C\$PNTX
020030	062706	000014		ADD	#14,SP
208 020034			PRINTX #FMT9,#DH9		
020034	012746	014534		MOV	#DH9,-(SP)
020040	012746	011712		MOV	#FMT9,-(SP)
020044	012746	000002		MOV	#2,-(SP)
020050	010600			MOV	SP,R0
020052	104415			TRAP	C\$PNTX
020054	062706	000006		ADD	#6,SP
209 020060			PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16		
020060	013746	002340		MOV	AX3.16,-(SP)
020064	013746	002336		MOV	AX3.15,-(SP)
020070	013746	002334		MOV	AX2.16,-(SP)
020074	013746	002332		MOV	AX2.15,-(SP)
020100	012746	011613		MOV	#FMT6,-(SP)
020104	012746	000005		MOV	#5,-(SP)
020110	010600			MOV	SP,R0
020112	104415			TRAP	C\$PNTX
020114	062706	000014		ADD	#14,SP
210 020120			ENDMSG		
020120				L10010:	
020120	104423			TRAP	C\$MSG
211					
212					
213					
214					
215					
216 020122			BGNMSG ERR8		
020122				ERR8::	
217 020122			PRINTB #FMT10,SUBRPC		
020122	013746	002352		MOV	SUBRPC,-(SP)
020126	012746	011717		MOV	#FMT10,-(SP)
020132	012746	000002		MOV	#2,-(SP)
020136	010600			MOV	SP,R0
020140	104414			TRAP	C\$PNTB
020142	062706	000006		ADD	#6,SP
218 020146			PRINTB #FMT1,#ADDRES,MPCSR		
020146	013746	002446		MOV	MPCSR,-(SP)
020152	012746	035620		MOV	#ADDRES,-(SP)
020156	012746	011454		MOV	#FMT1,-(SP)
020162	012746	000003		MOV	#3,-(SP)
020166	010600			MOV	SP,R0
020170	104414			TRAP	C\$PNTB
020172	062706	000010		ADD	#10,SP
219 020176			PRINTB #FMT2		
020176	012746	011464		MOV	#FMT2,-(SP)
020202	012746	000001		MOV	#1,-(SP)
020206	010600			MOV	SP,R0
020210	104414			TRAP	C\$PNTB
020212	062706	000004		ADD	#4,SP
220 020216			PRINTB #FMT7,#DH1,REGNUM		

	020216	013746	002400		MOV	REGNUM,-(SP)
	020222	012746	014264		MOV	#DH1,-(SP)
	020226	012746	011646		MOV	#FMT7,-(SP)
	020232	012746	000003		MOV	#3,-(SP)
	020236	010600			MOV	SP,R0
	020240	104414			TRAP	C\$PNTB
	020242	062706	000010		ADD	#10,SP
221	020246			PRINTB	#FMT3,GOODAT,BADDAT	
	020246	013746	002406		MOV	BADDAT,-(SP)
	020252	013746	002404		MOV	GOODAT,-(SP)
	020256	012746	011506		MOV	#FMT3,-(SP)
	020262	012746	000003		MOV	#3,-(SP)
	020266	010600			MOV	SP,R0
	020270	104414			TRAP	C\$PNTB
	020272	062706	000010		ADD	#10,SP
222	020276			PRINTX	#FMT4,#DH2,#DH3	
	020276	012746	014335		MOV	#DH3,-(SP)
	020302	012746	014306		MOV	#DH2,-(SP)
	020306	012746	011550		MOV	#FMT4,-(SP)
	020312	012746	000003		MOV	#3,-(SP)
	020316	010600			MOV	SP,R0
	020320	104415			TRAP	C\$PNTX
	020322	062706	000010		ADD	#10,SP
223	020326			PRINTX	#FMT5,LUR10,LJR11,LUR12,LUR13	
	020326	013746	002310		MOV	LUR13,-(SP)
	020332	013746	002306		MOV	LUR12,-(SP)
	020336	013746	002304		MOV	LUR11,-(SP)
	020342	013746	002302		MOV	LUR10,-(SP)
	020346	012746	011563		MOV	#FMT5,-(SP)
	020352	012746	000005		MOV	#5,-(SP)
	020356	010600			MOV	SP,R0
	020360	104415			TRAP	C\$PNTX
	020362	062706	000014		ADD	#14,SP
224	020366			PRINTX	#FMT9,#DH4	
	020366	012746	014373		MOV	#DH4,-(SP)
	020372	012746	011712		MOV	#FMT9,-(SP)
	020376	012746	000002		MOV	#2,-(SP)
	020402	010600			MOV	SP,R0
	020404	104415			TRAP	C\$PNTX
	020406	062706	000006		ADD	#6,SP
225	020412			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	020412	013746	002320		MOV	LUR17,-(SP)
	020416	013746	002316		MOV	LUR16,-(SP)
	020422	013746	002314		MOV	LUR15,-(SP)
	020426	013746	002312		MOV	LUR14,-(SP)
	020432	012746	011613		MOV	#FMT6,-(SP)
	020436	012746	000005		MOV	#5,-(SP)
	020442	010600			MOV	SP,R0
	020444	104415			TRAP	C\$PNTX
	020446	062706	000014		ADD	#14,SP
226	020452			PRINTX	#FMT4,#DH7,#DH8	
	020452	012746	014475		MOV	#DH8,-(SP)
	020456	012746	014443		MOV	#DH7,-(SP)
	020462	012746	011550		MOV	#FMT4,-(SP)
	020466	012746	000003		MOV	#3,-(SP)
	020472	010600			MOV	SP,R0
	020474	104415			TRAP	C\$PNTX

227 020476 062706 000010
020502 013746 002330
020506 013746 002326
020512 013746 002324
020516 013746 002322
020522 012746 011563
020526 012746 000005
020532 010600
020534 104415
020536 062706 000014
228 020542 012746 014534
020546 012746 011712
020552 012746 000002
020556 010600
020560 104415
020562 062706 000006
229 020566 013746 002340
020572 013746 002336
020576 013746 002334
020602 013746 002332
020606 012746 011613
020612 012746 000005
020616 010600
020620 104415
020622 062706 000014
230 020626 104423
231
232
233
234
235
236 020630 013746 002446
020634 012746 035620
020640 012746 011454
020644 012746 000003
020650 010600
020652 104414
020654 062706 000010
238 020660 012746 011464
020664 012746 000001
020670 010600
020672 104414
020674 062706 000004
239 020700 013746 002400
020704 012746 014264
020710 012746 011646
020714 012746 000003

PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16

PRINTX #FMT9,#DH9

PRINTX #FMT6,AX2.15,AX2.16,AX3.15,AX3.16

ENDMSG

BGNMSG ERR9

PRINTB #FMT1,#ADDRES,MPCSR

PRINTB #FMT2

PRINTB #FMT7,#DH1,REGNUM

ADD #10,SP
MOV AX1.16,-(SP)
MOV AX1.15,-(SP)
MOV AX0.16,-(SP)
MOV AX0.15,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP
MOV #DH9,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #6,SP
MOV AX3.16,-(SP)
MOV AX3.15,-(SP)
MOV AX2.16,-(SP)
MOV AX2.15,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP
L10011: TRAP C\$MSG
ERR9::
MOV MPCSR,-(SP)
MOV #ADDRES,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP
MOV #FMT2,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP
MOV REGNUM,-(SP)
MOV #DH1,-(SP)
MOV #FMT7,-(SP)
MOV #3,-(SP)

```
020720 010600
020722 104414
020724 062706 000010
240 020730 PRINTX #FMT4,#DH2,#DH3
020730 012746 014335
020734 012746 014306
020740 012746 011550
020744 012746 000003
020750 010600
020752 104415
020754 062706 000010
241 020760 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
020760 013746 002310
020764 013746 002306
020770 013746 002304
020774 013746 002302
021000 012746 011563
021004 012746 000005
021010 010600
021012 104415
021014 062706 000014
242 021020 PRINTX #FMT9,#DH4
021020 012746 014373
021024 012746 011712
021030 012746 000002
021034 010600
021036 104415
021040 062706 000006
243 021044 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
021044 013746 002320
021050 013746 002316
021054 013746 002314
021060 013746 002312
021064 012746 011613
021070 012746 000005
021074 010600
021076 104415
021100 062706 000014
244 021104 ENDMMSG
021104 104423
245
246
247
248
249
```

```
MOV SP,R0
TRAP C$PNTB
ADD #10,SP

MOV #DH3,-(SP)
MOV #DH2,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #10,SP

MOV LUR13,-(SP)
MOV LUR12,-(SP)
MOV LUR11,-(SP)
MOV LUR10,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP

MOV #DH4,-(SP)
MOV #FMT9,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #6,SP

MOV LUR17,-(SP)
MOV LUR16,-(SP)
MOV LUR15,-(SP)
MOV LUR14,-(SP)
MOV #FMT6,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #14,SP

L10012: TRAP C$MSG
```

.SBTTL REPORT CODING SECTION

:/ THE REPORT CODING SECTION CONTAINS THE
:/ 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
:/

BGNRPT

LSRPT::

ENDRPT

L10013: TRAP LSRPT

1
2
3
4
5
6
7
8
9 021106
021106
10
11 021106
021106
021106 104425
12
13
14
15
16

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

.SBTTL LOAD DEVICE PROTECTION TABLE

:/
:/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
:/ PROTECTED FROM TESTING, IF DESIRED.
:/

BGNPROT

L\$PROT::

.WORD -1 ;DON'T CHK CSR ADRS
.WORD -1 ;DON'T CHK MASSBUS UNIT NO.
.WORD -1 ;DON'T CHK DRIVE NO.
ENDPROT

```
1          .SBTTL  INITIALIZE SECTION
2
3          ;////////////////////////////////////
4          ;// THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5          ;// AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
6          ;////////////////////////////////////
7
8 021116          BGNINIT
9 021116
10 021116 010637 002346          MOV     SP,PSTACK          ;SAVE BASE-LEVEL STACK POINTER
11 021122 005037 002352          CLR     SUBRPC          ;CLEAR SUBR CALL PC
12 021126 005037 002426          CLR     DISILO          ;CLEAR CURRENT STATE OF DISSI
13 021132 005037 002430          CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
14 021136 005037 002420          CLR     ERROR1          ;CLEAR ERROR FLAG
15 021142 005037 002432          CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
16 021146 005737 002412          TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
17 021152 001007              BNE     6$          ;BR IF NOT
18 021154 013737 000004 002414          MOV     @#4,SAVE4          ;SAVE ERROR TRAP VECTOR
19 021162 013737 000006 002416          MOV     @#6,SAVE6
20 021170 000406              BR      9$
21 021172 013737 002414 000004 6$:          MOV     SAVE4,@#4          ;RESTORE ERROR TRAP VECTOR
22 021200 013737 002416 000006          MOV     SAVE6,@#6
23 021206 012737 000001 002412 9$:          MOV     #1,FRSTIM          ;MARK FLAG FOR NEXT TIME THROUGH
24          ;SEE IF PROGRAM JUST STARTED, BR IF YES
25 021214          READEF  #EF.START
26 021214 012700 000040          MOV     #EF.START,RO
27 021220 104447          TRAP     C$REFG
28 021222          BCOMPLETE          STARST
29 021222 103415          BCS     STARST
30          ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
31 021224          READEF  #EF.RESTART
32 021224 012700 000037          MOV     #EF.RESTART,RO
33 021230 104447          TRAP     C$REFG
34 021232          BCOMPLETE          STARST
35 021232 103411          BCS     STARST
36          ;SEE IF THIS IS A NEW PASS, BR IF YES
37 021234          READEF  #EF.NEW
38 021234 012700 000035          MOV     #EF.NEW,RO
39 021240 104447          TRAP     C$REFG
40 021242          BCOMPLETE          NEWST
41 021242 103411          BCS     NEWST
42          ;SEE IF PROGRAM WAS JUST CONTINUED
43 021244          READEF  #EF.CONTINUE
44 021244 012700 000036          MOV     #EF.CONTINUE,RO
45 021250 104447          TRAP     C$REFG
46 021252          BCOMPLETE          ENDIT
47 021252 103504          BCS     ENDIT
48 021254 000414          BR      GETPRM
49 021256          STARST:          CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
50 021256 005037 002444          ;CLEAR DEVICE MAP
51 021256          CLR     DEVMAP
52 021262 005037 002434          NEWST:          MOV     #-1,LOGDEV          ;RESET LOGICAL DEVICE TO -1
53 021266 012737 177777 002344          INC     STARES          ;INCR NO. OF PASSES SINCE STA OR RES
54 021274 005237 002444          MOV     #BIT0,DEVPTN          ;INIT DEVICE MAP BIT POINTER
55 021300 012737 000001 002436
```

```
45 ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
46 ; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
47 GETPRM:
48 021306 005237 002344 002012 INC LOGDEV ; INCREMENT LOGICAL DEVICE NUMBER
49 021312 023737 002344 002012 CMP LOGDEV,L$UNIT ; SEE IF MAXIMUM UNIT NO. EXCEEDED
50 021320 002362 BGE NEWST ; BR IF YES
51 021322 013700 002344 GPHARD LOGDEV,R1 ; GET P-TABLE POINTER INTO R1
    021322 013700 002344 MOV LOGDEV,R0
    021326 104442 TRAP CS$GPHRD
    021330 010001 MOV R0,R1
52 021332 BCOMPLETE 10$ ; BR IF DEVICE AVAILABLE
    021332 103403 BCS 10$
53 021334 006337 002436 ASL DEVPTR ; SHIFT DEVICE POINTER
54 021340 000762 BR GETPRM ; SKIP THIS DEVICE
55 021342 053737 002436 002434 10$: BIS DEVPTR,DEVMAP ; SET BIT FOR THIS DEVICE
56 021350 006337 002436 ASL DEVPTR ; SHIFT BIT POINTER
57 021354 062701 000002 ADD #2,R1 ; INCREMENT R1 PAST MICROPROCESSOR TYPE
58 021360 011137 002446 MOV (R1),MPCSR ; STORE POINTER TO MICROPROCESSOR CSR'S
59 021364 011137 002450 MOV (R1),BSEL1
60 021370 005237 002450 INC BSEL1 ; GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
61 021374 013737 002450 002452 MOV BSEL1,BSEL2
62 021402 005237 002452 INC BSEL2 ; GET POINTER TO BSEL2
63 021406 011137 002454 MOV (R1),SEL4
64 021412 062737 000004 002454 ADD #4,SEL4 ; GET POINTER TO SEL4
65 021420 012137 002456 MOV (R1)+,SEL6
66 021424 062737 000006 002456 ADD #6,SEL6 ; STORE POINTER TO SEL6
67 021432 011137 002460 MOV (R1),MPIVEC ; GET MICROPROCESSOR INPUT INTRPT VECTOR
68 021436 012137 002462 MOV (R1)+,MPOVEC
69 021442 062737 000004 002462 ADD #4,MPOVEC ; GET MICROPROCESSOR OUTPUT INTRPT VECTOR
70 021450 012137 002464 MOV (R1)+,MPRIOR ; GET MICROPROCESSOR DEVICE PRIORITY
71 021454 062701 000014 ADD #14,R1 ; POINT R1 TO RUN SWITCH INDICATOR
72 021460 011137 002476 MOV (R1),RUNINH ; GET STATE OF MICROPROCESSOR RUN SWITCH
73 ENDIT:
74 021464 ENDINIT
    021464 L10015:
    021464 104411 TRAP CS$INIT
75
76
77
78
79
80
```

```
1      .SBTTL  AUTO DROP UNIT SECTION
2
3      ;////////////////////////////////////
4      ;// THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
5      ;// WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
6      ;////////////////////////////////////
7      021466      BGNAUTO
8
9      021466      L$AUTO::
10     021466      ;ESTABLISH CPU PRIORITY = 7
11     021472      SETPRI  #PRI07
12     021474      012700 000340      MOV      #6$,@#4      ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
13     021472      104441      TRAP      C$SPRI
14     021474      012737 021516 000004      MOV      #PRI07,@#6
15     021502      012737 000340 000006      TST      @MPCSR      ;ADDRESS SELO
16     021510      005777 160732      BR      9$      ;TAKE THIS BR IF DEVICE RESPONDS
17     021514      000405      ;COME HERE IF DEVICE CSR IS NON-EXISTENT
18     021516      062706 000004      6$:      ADD      #4,SP      ;CLEAN UP THE STACK POINTER
19     021522      013700 002344      DODU      LOGDEV      ;DROP THIS UNIT FROM TESTING
20     021522      104451      MOV      LOGDEV,R0
21     021526      013737 002414 000004      TRAP      C$DODU
22     021530      013737 002416 000006      9$:      MOV      SAVE4,@#4      ;RESTORE ERROR TRAP VECTOR
23     021544      021544      MOV      SAVE6,@#6
24     021544      104461      ENDAUTO
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

.SBTTL CLEANUP CODING SECTION

:/
:/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
:/ AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
:/

021546
021546

BGNCLN

L\$CLEAN::

021546
021546
021546

ENDCLN

L10017: TRAP C\$CLEAN

104412

```
1      .SBTTL  DROP UNIT SECTION
2
3      ;////////////////////////////////////
4      ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5      ;/ TO NO LONGER BE TESTED.
6      ;////////////////////////////////////
7
8 021550      BGNDU
9 021550
10 021550      ;ISSUE UNIBUS RESET TO CLEAN UP
10 021550 104433      BRESET
11
12 021552      ;PRINT 'UNIT XX DROPPED'
12 021552 013746 002344      PRINTF #FMT27,LOGDEV
12 021556 012746 021600
12 021562 012746 000002
12 021566 010600
12 021570 104417
12 021572 062706 000006
13 021576      ENDDU
13 021576
13 021576 104453
14
15 021600      045      116      045  FMT27: .ASCIZ  /%N%AUNIT %D2%A DROPPED%N/
15 021603      101      125      116
15 021606      111      124      040
15 021611      045      104      062
15 021614      045      101      040
15 021617      104      122      117
15 021622      120      120      105
15 021625      104      045      116
15 021630      000
16
17
18
19
20
21
```

LSDU::

TRAP	C\$RESET
MOV	LOGDEV,-(SP)
MOV	#FMT27,-(SP)
MOV	#2,-(SP)
MOV	SP,R0
TRAP	C\$PNTF
ADD	#6,SP

L10020:

TRAP	C\$DU
------	-------

.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

.SBTTL ADD UNIT SECTION

:/
:/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
:/ TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
:/ 'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
:/

021632
021632
021632
021632 104452

BGNAU
ENDAU

LSAU::
L10021: TRAP CSAU

.SBTTL HARDWARE TESTS

```
*****
.SBTTL      TEST 1 - MICROPROCESSOR CSR ADDRESSING TEST (SELO)
*
* THIS TEST ADDRESSES THE FIRST MICROPROCESSOR CSR (SELO), TO MAKE SURE
* THAT A NON-EXISTENT MEMORY TIME-OUT TRAP DOES NOT OCCUR WHILE
* ATTEMPTING TO ADDRESS THE MICROPROCESSOR.
*****
BGNTEST
```

T1::

```
;ESTABLISH CPU PRIORITY = 7
SETPRI #PRI07
```

```
MOV #PRI07,R0
TRAP C$SPRI
;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR

MOV #6$,a#4
MOV #PRI07,a#6
TST @MPCSR
BR 9$
;ADDRESS SELO
;TAKE THIS BR IF DEVICE RESPONDS

;COME HERE IF DEVICE CSR IS NON-EXISTENT
6$: ADD #4,SP
;CLEAN UP THE STACK POINTER
;REPORT CSR ADDRESS TIME-OUT
ERRDF 1,EM1,ERR1
```

```
TRAP C$ERDF
.WORD 1
.WORD EM1
.WORD ERR1
```

```
9$: MOV SAVE4,a#4
MOV SAVE6,a#6
;RESTORE ERROR TRAP VECTOR

ENDTEST
```

L10022:

TRAP C\$ETST

```
*****
.SBTTL      TEST 2 - INBUS/OUTBUS REG 14 INITIALIZATION TEST
*
* MASTER CLEAR (MCLR) IS SET IN THE MICROPROCESSOR, IBUS REG 14 IS READ
* AND COMPARED TO 200.
*****
BGNTEST
```

T2::

```
MOV #14,REGNUM
JSR PC,MSTCLR
JSR PC,READLU
CMPB REDBYT,PATM+4
BEQ 6$
CLR GOODAT
MOVB PATM+4,GOODAT
MOV REDBYT,BADDAT
;SET LU REG NO. = 14
;ISSUE MASTER CLEAR
;READ REG 14
;CHK FOR INITIALIZED STATE
;BR IF YES
;SET EXPECTED REG CONTENTS - 000
;SET EXPECTED DATA
;SET ACTUAL REG CONTENTS
```

```

48 021764 004737 004016      ;REPORT JSR PC,GETREG      ;GET REGS FOR PRINTOUT
49                               ;REPORT REG NOT CLEARED BY MASTER CLEAR
50 021770                               ERRDF 2,EM2,ERR2
    021770 104455
    021772 000002
    021774 012161
    021776 014632
51 022000
52 022000
    022000
    022000 104401
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67 022002
    022002
68 022002 004737 003576
69 022006 012737 000014 002400
70 022014 012701 002571
71 022020
72 022020
    022020 104404
73 022022 111137 002366
74 022026 143737 002560 002366
75 022034 004737 003750
76 022040 004737 003672
77 022044 143737 002560 002364
78 022052 123737 002364 002366
79 022060 001414
80 022062 013737 002366 002404
81 022070 013737 002364 002406
82 022076 004737 004016
83
84 022102
    022102 104455
    022104 000003
    022106 012220
    022110 014632
85 022112
86 022112
    022112
    022112 104405
87 022114 005201
88 022116 020127 002615
89 022122 103736
90 022124

;*****
;SBTTL TEST 3 - INBUS/OUTBUS REG 14 READ/WRITE BIT TEST
;
; * WRITE, READ, AND COMPARE ALL WORDS OF DATA PATTERN A INTO REG 14,
; * A BYTE AT A TIME. NON-R/W BITS ARE MASKED OFF TO 0 BEFORE WRITING AND
; * READING.
; * DATA PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
; *                   375,373,367,357,337,277,177.
;*****
BGNST
T3::
    JSR PC,MSTCLR      ;ISSUE MASTER CLEAR
    MOV #14,REGNUM     ;SET LU REG NO. = 14
    MOV #PATA,R1       ;GET POINTER TO DATA PAT IN R1
3$:
    BGNSEG
                                TRAP C$BSEG
    MOVB (R1),WRIBYT    ;GET A BYTE OF PAT A
    BICB R14NRW,WRIBYT ;MASK OFF NON-READ/WRITE BITS
    JSR PC,WRITLU      ;WRITE DATA BYTE INTO REG 14
    JSR PC,READLU      ;READ DATA BYTE FROM REG 14
    BICB R14NRW,REDBYT ;MASK OFF NON-READ/WRITE BITS
    CMPB REDBYT,WRIBYT ;COMPARE BYTE READ TO BYTE WRITTEN
    BEQ 6$             ;BR IF BYTES MATCH
    MOV WRIBYT,GOODAT   ;SET EXPECTED REG CONTENTS
    MOV REDBYT,BADDAT   ;SET ACTUAL REG CONTENTS
    JSR PC,GETREG      ;GET REGS FOR PRINTOUT
;REPORT LINE UNIT REG MISCOMPARE
ERRDF 3,EM3,ERR2
                                TRAP C$ERDF
                                .WORD 3
                                .WORD EM3
                                .WORD ERR2
6$:
    ENDSEG
                                10000$:
                                TRAP C$ESEG
    INC R1
    CMP R1,#PATB
    BLO 3$
;INCREMENT DATA PATTERN POINTER
;SEE IF ALL WORDS OF PATTERN A USED YET
;BR IF NOT DONE YET
FNDTST

```

```

022124
022124 104401
91
92
93
94
95
96
97
98
99
100
101 022126
102 022126 004737 003576
103 022132 012737 000014 002400
104 022140 112737 000377 002366
105 022146 004737 003750
106 022152 004737 003576
107 022156 004737 003672
108 022162 123737 002364 003026
109 022170 001416
110 022172 005037 002404
111 022176 113737 003026 002404
112 022204 013737 002364 002406
113 022212 004737 004016
114
115 022216
116 022216 104455
117 022220 000002
118 022222 012161
119 022224 014632
120
121
122
123
124
125
126
127
128 022230
129 022230 004737 003576
130 022234 012737 000014 002400
131 022242 112737 000377 002366
132 022250 004737 003750
133 022254
134 022254 104433
135 022256 142777 000200 160164
136 022264 012701 000024
137 022270 000240

;*****
;SBTTL TEST 4 - REG 14 MASTER CLEAR TEST
;* WRITE 377 INTO REG 14, ISSUE MASTER CLEAR, READ REG 14 AND COMPARE
;* TO 200.
;*****
BGNSTST
T4::
JSR PC,MSTCLR ;PERFORM MASTER CLEAR
MOV #14,REGNUM ;SET LU REG NO. = 14
MOVB #377,WRIBYT ;SET DATA BYTE = 377
JSR PC,WRITLU ;WRITE 377 INTO REG 14
JSR PC,MSTCLR ;ISSUE MASTER CLEAR
JSR PC,READLU ;READ REG 14
CMPB REDBYT,PATM+4 ;CHK FOR INIT'D STATE
BEQ 6$ ;BR IF REG GOT CLEARED
CLR GOODAT
MOVB PATM+4,GOODAT ;SET EXPECTED DATA
MOV REDBYT,BADDAT ;SET ACTUAL DATA
JSR PC,GETREG ;GET REGS FOR PRINTOUT
;REPORT REG NOT CLEARED BY MASTER CLEAR
ERRDF 2,EM2,ERR2
6$:
ENDTST
L10024: TRAP C$SETST

;*****
;SBTTL TEST 5 - REG 14 UNIBUS RESET (INIT) TEST
;* WRITE 377 INTO REG 14, ISSUE UNIBUS RESET (INIT), READ REG 14 AND COMPARE
;* TO 200.
;*****
BGNSTST
T5::
JSR PC,MSTCLR ;PERFORM MASTER CLEAR
MOV #14,REGNUM ;SET LU REG NO. = 14
MOVB #377,WRIBYT ;SET DATA BYTE = 377
JSR PC,WRITLU ;WRITE 377 INTO REG 14
BRESET ;ISSUE UNIBUS RESET (INIT)
TRAP C$RESET
BICB #RUN,@BSEL1 ;CLEAR RUN BIT
MOV #20.,R1 ;INIT LOOP COUNTER
2$:
NOP

```

137 022272 005301
 138 022274 001375
 139 022276 004737 003672
 140 022302 142737 000100 002364
 141 022310 123737 002364 003026
 142 022316 001416
 143 022320 005037 002404
 144 022324 113737 003026 002404
 145 022332 013737 002364 002406
 146 022340 004737 004016
 147
 148 022344
 022344 104455
 022346 000004
 022350 012237
 022352 014632
 149 022354
 150 022354
 022354
 022354 104401
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166 022356
 022356
 167 022356 004737 003576
 168 022362 012737 000014 002400
 169 022370 013701 002400
 170 022374 052701 000100
 171 022400 052701 121000
 172 022404 010137 022422
 173 022410 112777 000041 160036
 174 022416 004737 003540
 175 022422 000000
 176 022424 004737 003672
 177 022430 123737 002364 003026
 178 022436 001416
 179 022440 005037 002404
 180 022444 113737 003026 002404
 181 022452 013737 002364 002406
 182 022460 004737 004016
 183
 184 022464
 022464 104455
 022466 000003

DEC R1 ;DECR COUNTER
 BNE 2\$;BR TO STALL
 JSR PC,READLU ;READ REG 14
 BICB #TXEN,REDBYT ;CLEAR UNPREDICTABLE BIT
 CMPB REDBYT,PATM+4 ;CHK FOR INIT'D STATE
 BEQ 6\$;BR IF REG GOT CLEARED
 CLR GOODAT
 MOVB PATM+4,GOODAT ;SET EXPECTED DATA
 MOV REDBYT,BADDAT ;SET ACTUAL DATA
 JSR PC,GETREG ;GET REGS FOR PRINTOUT
 ;REPORT REG NOT CLEARED BY UNIBUS RESET (INIT)
 ERRDF 4,EM4,ERR2

TRAP C\$ERDF
 .WORD 4
 .WORD EM4
 .WORD ERR2

6\$:
 ENDTST

L10026:
 TRAP C\$ETST

 .SBTTL TEST 6 - LINE UNIT FALSE SELECTION TEST
 *
 * FIRST, A MASTER CLEAR IS PERFORMED. THEN, THE PROGRAM SINGLE-STEPS THE
 * MICROPROCESSOR THROUGH AN INSTRUCTION WHICH LOADS 041 (OCT) INTO THE MAR
 * REGISTER (OBUS* ADRS 14). THEN, THE LINE UNIT REGISTER 14 IS READ AND CHECKED
 * TO BE UNAFFECTED (STILL = 0). THIS TEST IS INTENDED TO DETECT A FALSE
 * SELECTION OF THE LINE UNIT REGISTERS, WHEN THE LINE UNIT IS NOT BEING
 * ACCESSED.

BGNTST

T6::

JSR PC,MSTCLR ;ISSUE A MASTER CLEAR
 MOV #14,REGNUM ;SET LU REG NO. = 14
 MOV REGNUM,R1 ;SET DESTINATION = OBUS* REG 14
 BIS #100,R1 ;SET SOURCE = BSEL4
 BIS #MVIXOX,R1 ;SET REST OF MOVE INSTRUCTION
 MOV R1,2\$;SET INSTRUCTION AS SUBROUTINE ARGUMENT
 MOVB #041,BSEL4 ;SET DATA BYTE = 041
 JSR PC,STPCLK ;EXECUTE MOVE INSTRUCTION
 .WORD 0 ;INSTRUCTION GOES HERE
 JSR PC,READLU ;READ LU REG 14
 CMPB REDBYT,PATM+4 ;CHECK FOR LU REG 14 UNCHANGED
 BEQ 4\$;BR IF LU REG 14 UNCHANGED
 CLR GOODAT ;SET EXPECTED DATA
 MOVB PATM+4,GOODAT
 MOV REDBYT,BADDAT ;SET ACTUAL DATA
 JSR PC,GETREG ;GET REGS FOR PRINTOUT
 ;REPORT REGISTER MISCOMPARE
 ERRDF 3,EM3,ERR2

TRAP C\$ERDF
 .WORD 3

185	022470	012220					
186	022472	014632				.WORD	EM3
	022474		4\$:			.WORD	ERR2
	022474		ENDTST				
	022474	104401			L10027:	TRAP	C\$ETST

187
188
189
190
191
192

```

*****
:SBTTL      TEST 7 - INBUS REG MASTER CLEAR TEST
:
:* FIRST, ALL READ/WRITE BITS OF REGS 10-17 ARE SET BY LOADING A
:* DIFFERENT WORD OF PATTERN G INTO EACH REG. THEN,
:* A MASTER CLEAR IS ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF
:* PATTERN M, WHICH CONTAINS THE INITIALIZED STATES OF THE REGS. (UNPREDICTABLE
:* BITS ARE MASKED OFF TO 0 BEFORE COMPARISON).
:* PATTERN G = 000,000,240,120,177,000,000,001
:* PATTERN M = 000,020,000,000,200,000,000,051
*****
:BGNTST

```

203	022476						
204	022476	004737	003576				
205	022502	012737	000010	002400			
206	022510	012701	002644				
207	022514	112137	002366		2\$:		
208	022520	004737	003750				
209	022524	005237	002400				
210	022530	020127	002654				
211	022534	103767					
212	022536	004737	003576				
213	022542	012737	000010	002400			
214	022550	012702	003022				
215	022554	012701	002550				
216	022560				3\$:		
217	022560	004737	003672				
218	022564	142137	002364				
219	022570	123712	002364				
220	022574	001417					
221	022576	005037	002404				
222	022602	111237	002404				
223	022606	013737	002364	002406			
224	022614	004737	004016				
225							
226	022620						
	022620	104455					
	022622	000002					
	022624	012161					
	022626	014632					
227	022630						
	022630	104410					
	022632	000016					
228	022634	005237	002400		6\$:		
229	022640	005202					
230	022642	020227	003032				

```

T7::
JSR PC,MSTCLR      ;ISSUE MASTER CLEAR
MOV #10,REGNUM     ;INIT REG NO. TO 10
MOV #PATG,R1       ;INIT DATA PATTERN POINTER
MOVB (R1)+,WRIBYT  ;SET DATA PATTERN BYTE TO BE WRITTEN
JSR PC,WRITLU      ;WRITE BYTE INTO REG
INC REGNUM          ;INCREMENT REG NO. FOR WRITING
CMP R1,#PATH       ;SEE IF ALL BYTES WRITTEN YET
BLO 2$             ;BR IF NOT DONE WRITING YET
JSR PC,MSTCLR      ;ISSUE MASTER CLEAR
MOV #10,REGNUM     ;INIT LU REG NO. TO 10
MOV #PATM,R2       ;INIT DATA PATTERN POINTER
MOV #UPBITS,R1     ;INIT POINTER TO UNPREDICTABLE BITS

JSR PC,READLU      ;READ A LINE UNIT REG
BICB (R1)+,REDBYT  ;MASK OUT UNPREDICTABLE BITS FOR THIS REG
CMPB REDBYT,(R2)   ;COMPARE MASKED DATA TO EXPECTED
BEQ 6$             ;BR IF DATA READ IS OK
CLR GOODAT         ;SET EXPECTED DATA
MOVB (R2),GOODAT   ;SET ACTUAL DATA
MOV REDBYT,BADDAT  ;GET REGS FOR PRINTOUT
JSR PC,GETREG      ;GET REGS FOR PRINTOUT

;REPORT REG NOT CLEARED BY MASTER CLEAR
ERRDF 2,EM2,ERR2

```

TRAP C\$ERDF
 .WORD 2
 .WORD EM2
 .WORD ERR2

ESCAPE TST

TRAP C\$ESCAPE
 .WORD L10030-

```

6$: INC REGNUM      ;INCREMENT REG NO.
     INC R2         ;INCR DATA PATTERN POINTER
     CMP R2,#PATN   ;SEE IF ALL DONE YET

```

231 022646 103744 BLO 3\$;BR IF NOT DONE YET
 232 022650
 022650
 022650 104401 L10030: TRAP C\$ETST

233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246

 SBTTL TEST 8 - REGISTER 10-17 ADDRESSING TEST
 *
 * FIRST, A MASTER CLEAR IS ISSUED. THEN,
 * WRITE A DIFFERENT WORD OF DATA PATTERN B INTO EACH OF REGS 10-17,
 * AND AFTER EACH WRITE, READ AND COMPARE ALL REGS TO EXPECTED VALUES.
 * UNPREDICTABLE BITS ARE MASKED OFF TO 0 WHEN READ FOR COMPAPISON.
 * PATTERN B = 000,000,040,100,220,000,000,051

 BGNST

247 022652

248 022652 004737 003576 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
 249 022656 012701 002500 MOV #REDDAT,R1 ;INIT POINTER TO EXPECTED DATA AREA
 250 022662 012702 003022 MOV #PATM,R2 ;GET POINTER TO PATTERN M
 251 022666 012703 000010 MOV #8,R3 ;SET COUNTER
 252 022672 112221 3\$: MOVB (R2)+,(R1)+ ;LOAD BYTE OF PATRN INTO EXPECTED DATA AREA
 253 022674 005303 DEC R3 ;DECR COUNTER
 254 022676 001375 BNE 3\$;BR IF NOT DONE LOADING YET
 255 022700 005001 CLR R1 ;INIT DATA PATTERN INDEX FOR WRITING
 256 022702 010137 002400 6\$: MOV R1,REGNUM
 257 022706 062737 000010 002400 ADD #10,REGNUM ;GET REG NO. FOR WRITING
 258 022714 116137 002615 002366 MOVB PATB(R1),WRIBYT ;SET DATA BYTE TO BE WRITTEN
 259 022722 113761 002366 002500 MOVB WRIBYT,REDDAT(R1) ;SET EXPECTED DATA FOR READ
 260 022730 146161 002550 002500 BICB UPBITS(R1),REDDAT(R1) ;MASK OUT UNPREDICTABLE BITS
 261 022736 004737 003750 JSR PC,WRITLU ;WRITE DATA BYTE INTO REG
 262 022742 005003 CLR R3 ;INIT DATA PAT INDEX FOR READS
 263 022744 010337 002400 9\$: MOV R3,REGNUM
 264 022750 062737 000010 002400 ADD #10,REGNUM ;GET REG NO. FOR READING
 265 022756 004737 003672 JSR PC,READLU ;READ A LINE UNIT REG
 266 022762 146337 002550 002364 BICB UPBITS(R3),REDBYT ;MASK OUT UNPREDICTABLE BITS
 267 022770 023727 002400 000011 CMP REGNUM,#11 ;SEE IF READING REG 11
 268 022776 001006 BNE 10\$;BR IF NOT
 269 023000 142737 000020 002364 BICB #ORDY,REDBYT ;MASK ORDY BIT IN ACTUAL BYTE
 270 023006 142763 000020 002500 BICB #ORDY,REDDAT(R3) ;MASK ORDY BIT IN EXPECTED BYTE
 271 023014 123763 002364 002500 10\$: CMPB REDBYT,REDDAT(R3) ;COMPARE BYTE READ TO EXPECTED
 272 023022 001420 BEQ 12\$;BR IF DATA MATCHES
 273 023024 005037 002404 CLR GOODAT
 274 023030 116337 002500 002404 MOVB REDDAT(R3),GOODAT ;SET EXPECTED DATA
 275 023036 013737 002364 002406 MOV REDBYT,BADDAT ;SET ACTUAL DATA
 276 023044 004737 004016 JSR PC,GETREG ;READ AND STORE REGS 10-17 FOR PRINTOUT
 277 ;REPORT REG MISCOMPARE
 278 ERRDF 3,EM3,ERR2

TRAP C\$ERDF
 .WORD 3
 .WORD EM3
 .WORD ERR2

278 023050
 023050 104455
 023052 000003
 023054 012220
 023056 014632
 279 023060
 023060 104410

ESCAPE TST

TRAP C\$ESCAPE

```

280 023062 000022
281 023064 005203 000010 12$: INC R3 ;INCREMENT DATA PATTERN INDEX FOR READING
282 023072 002724 000010 CMP R3,#10 ;SEE IF ALL REGS READ YET
283 023074 005201 BLT 9$ ;BR IF NOT
284 023076 020127 000010 INC R1 ;INCREMENT DATA PAT INDEX FOR WRITING
285 023102 002677 CMP R1,#10 ;SEE IF ALL REGS WRITTEN YET
286 023104 002677 BLT 6$ ;BR IF NOT
287 023104 104401 ENDTST
288
289
290
291
292
293
294
295
296
297
298 023106 BGNTST
299 023106 004737 003576 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
300 023112 012737 000011 002400 MOV #11,REGNUM ;SET LU REG NO. = 11
301 023120 012701 002625 MOV #PATC,R1 ;GET POINTER TO DATA PAT IN R1
302 023124 3$: BGNSEG
303 023124 104404 TRAP C$BSEG
304 023126 111137 002366 MOVB (R1),WRIBYT ;GET A BYTE OF PAT C
305 023132 004737 003750 JSR PC,WRITLU ;WRITE DATA BYTE INTO REG 11
306 023136 004737 003672 JSR PC,READLU ;READ DATA BYTE FROM REG 11
307 023142 143737 002551 002364 BICB UPBITS+1,REDBYT ;MASK OUT UNPREDICTABLE BITS
308 023150 123737 002364 002366 CMPB REDBYT,WRIBYT ;COMPARE BYTE READ TO BYTE WRITTEN
309 023156 001414 BEQ 6$ ;BR IF BYTES MATCH
310 023160 013737 002366 002404 MOV WRIBYT,GOODAT ;SET EXPECTED REG CONTENTS
311 023166 013737 002364 002406 MOV REDBYT,BADDAT ;SET ACTUAL REG CONTENTS
312 023174 004737 004016 JSR PC,GETREG ;GET REGS FOR PRINTOUT
313 ;REPORT LINE UNIT REG MISCOMPARE
314 023200 ERRDF 3,EM3,ERR2
315 023200 104455 TRAP C$ERDF
316 023202 000003 .WORD 3
317 023204 012220 .WORD EM3
318 023206 014632 .WORD ERR2
319 023210 6$: ENDSEG
320 023210 10000$: TRAP C$ESEG
321 023210 104405 INC R1 ;INCREMENT DATA PATTERN POINTER
322 023212 005201 002630 CMP R1,#PATD ;SEE IF ALL WORDS OF PATTERN C USED YET
323 023214 020127 002630 BLO 3$ ;BR IF NOT DONE YET
324 023220 103741 ENDTST
325 023222 104401 L10032: TRAP C$SETST
326 023222 104401
327
328
329
330
331
332
333

```

```

324
325
326
327
328
329
330
331
332 023224
    023224
333 023224 004737 003576
334 023230 012737 000012 002400
335 023236 012701 002630
336 023242
337 023242
    023242 104404
338 023244 111137 002366
339 023250 004737 003750
340 023254 004737 003672
341 023260 143737 002552 002364
342 023266 123737 002364 002366
343 023274 001414
344 023276 013737 002366 002404
345 023304 013737 002364 002406
346 023312 004737 004016
347
348 023316
    023316 104455
    023320 000003
    023322 012220
    023324 014632
349 023326
350 023326
    023326 104405
351 023330 005201
352 023332 020127 002633
353 023336 103741
354 023340
    023340
    023340 104401
355
356
357
358
359
360
361
362
363
364
365
366 023342
    023342
367 023342 004737 003576
368 023346 012737 000013 002400
369 023354 012701 002633

;*****
;SBTTL      TEST 10 - REG 12 READ/WRITE BIT TEST
;
;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN D INTO REG 12 :
;* DATA PATTERN D = 000,040,000.
;*****
BGNTST
                                T10::
                                JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR
                                MOV    #12,REGNUM      ;SET LU REG NO. = 12
                                MOV    #PATD,R1        ;GET POINTER TO DATA PAT IN R1
3$:
    BGNSEG
                                TRAP    C$BSEG
                                MOVB   (R1),WRIBYT    ;GET A BYTE OF PAT D
                                JSR     PC,WRITLU      ;WRITE DATA BYTE INTO REG 12
                                JSR     PC,READLU      ;READ DATA BYTE FROM REG 12
                                BICB   UPBITS+2,REDBYT ;MASK OUT UNPREDICTABLE BITS
                                CMPB   REDBYT,WRIBYT  ;COMPARE BYTE READ TO BYTE WRITTEN
                                BEQ     6$             ;BR IF BYTES MATCH
                                MOV     WRIBYT,GOODAT  ;SET EXPECTED REG CONTENTS
                                MOV     REDBYT,BADDAT  ;SET ACTUAL REG CONTENTS
                                JSR     PC,GETREG      ;GET REGS FOR PRINTOUT
                                ;REPORT LINE UNIT REG MISCOMPARE
                                ERRDF  3,EM3,ERR2
                                TRAP    C$ERDF
                                .WORD   3
                                .WORD   EM3
                                .WORD   ERR2
6$:
    ENDSEG
                                10000$:
                                TRAP    C$ESEG
                                INC     R1             ;INCREMENT DATA PATTERN POINTER
                                CMP     R1,#PATE      ;SEE IF ALL WORDS OF PATTERN D USED YET
                                BLO     3$            ;BR IF NOT DONE YET
                                100033:
                                TRAP    C$ETST
ENDTST

;*****
;SBTTL      TEST 11 - REG 13 READ/WRITE BIT TEST
;
;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN E INTO REG 13 :
;* DATA PATTERN E = 000,120,020,100,120,000.
;*****
BGNTST
                                T11::
                                JSR     PC,MSTCLR      ;ISSUE MASTER CLEAR
                                MOV     #13,REGNUM      ;SET LU REG NO. = 13
                                MOV     #PATE,R1        ;GET POINTER TO DATA PAT IN R1

```

```

370 023360
371 023360
372 023362 104404 002366
373 023366 004737 003750
374 023372 004737 003672
375 023376 143737 002553 002364
376 023404 123737 002364 002366
377 023412 001414
378 023414 013737 002366 002404
379 023422 013737 002364 002406
380 023430 004737 004016
381
382 023434
383 023434 104455
384 023436 000003
385 023440 012220
386 023442 014632
387 023444
388 023444
389 023444 104405
390 023446 005201
391 023450 020127 002641
392 023454 103741
393 023456
394 023456
395 023456 104401
396
397
398
399
400 023460
401 023460 004737 003576
402 023464 012737 000017 002400
403 023472 012701 002641
404 023476
405 023476
406 023476 104404
407 023500 111137 002366
408 023504 004737 003750
409 023510 004737 003672
410 023514 143737 002557 002364
411 023522 123737 002364 002366
412 023530 001414
413 023532 013737 002366 002404
414 023540 013737 002364 002406
415 023546 004737 004016

3$:
BGNSEG
MOV (R1),WRIBYT ;GET A BYTE OF PAT E
JSR PC,WRITLU ;WRITE DATA BYTE INTO REG 13
JSR PC,READLU ;READ DATA BYTE FROM REG 13
BICB UPBITS+3,REDBYT ;MASK OUT UNPREDICTABLE BITS
CMPB REDBYT,WRIBYT ;COMPARE BYTE READ TO BYTE WRITTEN
BEQ 6$ ;BR IF BYTES MATCH
MOV WRIBYT,GOODAT ;SET EXPECTED REG CONTENTS
MOV REDBYT,BADDAT ;SET ACTUAL REG CONTENTS
JSR PC,GETREG ;GET REGS FOR PRINTOUT
;REPORT LINE UNIT REG MISCOMPARE
ERRDF 3,EM3,ERR2

6$:
ENDSEG

10000$:
TRAP C$BSEG

INC R1 ;INCREMENT DATA PATTERN POINTER
CMP R1,#PATF ;SEE IF ALL WORDS OF PATTERN E USED YET
BLO 3$ ;BR IF NOT DONE YET

ENDTST

L10034:
TRAP C$SETST

;*****
;SBTTL TEST 12 - REG 17 READ/WRITE BIT TEST
;*
;* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN F INTO REG 17 :
;* DATA PATTERN F = 050,051,050.
;*****
BGNTST

T12::
JSR PC,MSTCLR ;ISSUE MASTER CLEAR
MOV #17,REGNUM ;SET LU REG NO. = 17
MOV #PATF,R1 ;GET POINTER TO DATA PAT IN R1

3$:
BGNSEG
MOV (R1),WRIBYT ;GET A BYTE OF PAT F
JSR PC,WRITLU ;WRITE DATA BYTE INTO REG 17
JSR PC,READLU ;READ DATA BYTE FROM REG 17
BICB UPBITS+7,REDBYT ;MASK OUT UNPREDICTABLE BITS
CMPB REDBYT,WRIBYT ;COMPARE BYTE READ TO BYTE WRITTEN
BEQ 6$ ;BR IF BYTES MATCH
MOV WRIBYT,GOODAT ;SET EXPECTED REG CONTENTS
MOV REDBYT,BADDAT ;SET ACTUAL REG CONTENTS
JSR PC,GETREG ;GET REGS FOR PRINTOUT
;REPORT LINE UNIT REG MISCOMPARE

```

```

416 023552          ERRDF 3,EM3,ERR2
    023552 104455
    023554 000003
    023556 012220
    023560 014632
417 023562          6$:
418 023562          ENDSEG
    023562          10000$:
419 023564 104405          TRAP C$ESEG
    023564 005201
420 023566 020127 002644
421 023572 103741
422 023574          : INC R1 ;INCREMENT DATA PATTERN POINTER
    023574 104401          CMP R1,#PATG ;SEE IF ALL WORDS OF PATTERN F USED YET
    023574          BLO 3$ ;BR IF NOT DONE YET
    023574          : ENDTST
    023574          L10035:
    023574          TRAP C$ETST
  
```

423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448

```

*****
:SBTTL      TEST 13 - MAINTENANCE CLOCK BIT TEST
:
: * FIRST, A MASTER CLEAR IS ISSUED TO INIT ALL REGS. THEN, THE MICROPROCESSOR
: * IS PLACED IN A LOOP ON AN INSTRUCTION, BY SETTING THE INSTRUCTION IN SEL6
: * AND SETTING ROMI AND RUN IN BSEL1. THE INSTRUCTION IS ONE WHICH REPETITIVELY
: * READS LINE UNIT REG 17 INTO BSEL2. THE PDP-11 CAN THEN SCAN BSEL2 TO MONITOR
: * THE MAINTENANCE CLOCK BIT, MCLK. THE FOLLOWING SEQUENCE IS THEN PERFORMED
: * TO MONITOR MCLK :
: * - THE PROGRAM REPEATEDLY CHECKS THE MCLK BIT FOR THE 1 STATE, AND IF IT IS
: *   NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC (DEPENDING ON THE PROCESSOR)
: *   AN ERROR IS REPORTED. (THE MAINTENANCE CLOCK HAS A PERIOD OF 41.6 MICRO-
: *   SEC).
: * - THE PROGRAM NEXT REPEATEDLY CHECKS THE MCLK BIT FOR THE 0 STATE, AND IF
: *   IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC AN ERROR IS REPORTED.
: * - THE PROGRAM NEXT REPEATEDLY CHECKS MCLK BIT FOR THE 1 STATE AGAIN, AND
: *   IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MILLI-SEC, AN ERROR IS REPORTED.
: *
: * IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE M8207 RUN SWITCH (E28 SW7)
: * IS OFF, THE TEST WILL BE SKIPPED.
*****
:BGNTST
  
```

```

449 023576          T13::
    023576
450 023576 004737 003576          JSR PC,MSTCLR ;PERFORM MASTER CLEAR
451 023602 012737 000015 002442          MOV #13,,TSTNUM ;SET TEST NO.
452 023610 005737 002476          TST RUNINH ;SEE IF RUN SWITCH IS SET
453 023614 001020          BNE 1$ ;BR IF YES, TO RUN TEST
454 023616 023727 002444 000001          CMP STARES,#1 ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
455 023624 001012          BNE 40$ ;BR IF NOT, TO SKIP PRINTING
456 023626          PRINTF #FMT19,TSTNUM ;PRINT MSG TO SAY TEST NOT RUN
    023626 013746 002442          MOV TSTNUM,-(SP)
    023632 012746 012007          MOV #FMT19,-(SP)
    023636 012746 000002          MOV #2,-(SP)
    023642 010600          MOV SP,R0
    023644 104417          TRAP C$PNTF
    023646 062706 000006          ADD #6,SP
457 023652 000137 024220          40$: JMP A1 ;GO TO SKIP TEST
  
```

```

458 023656 012737 000017 002400 1$: MOV #17,REGNUM ;SET REG NO. = 17
459 023664 012701 000360 MOV #360,R1 ;SET INSTRUCTION SOURCE = INBUS REG 17
460 023670 052701 000002 BIS #2,R1 ;SET INSTRUCTION DESTINATION = BSEL2
461 023674 052701 021000 BIS #MVI0X,R1 ;SET REST OF MOVE INSTRUCTION
462 023700 010137 023710 MOV R1,2$ ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
463 023704 004737 004074 JSR PC,LOOPIN ;GET MICROPROCESSOR LOOPING ON MOVE INSTRUCTION
464 023710 000000 2$: .WORD 0 ;INSTRUCTION GOES HERE
465
466 -----
467 ; WAIT FOR MCLK BIT TO BE SET TO 1
468 -----
469 023712 005037 002510 CLR REG0 ;INIT PROGRAM TIMER
470 023716 117737 156530 002364 3$: MOVB @BSEL2,REDBYT ;GET REG 17 INTO REDBYT
471 023724 132737 000002 002364 BITB #MCLK,REDBYT ;SEE IF MCLK BIT = 1 YET
472 023732 001031 BNE 6$ ;BR IF MCLK = 1
473 023734 005237 002510 INC REG0 ;INCREMENT TIMER
474 023740 001366 BNE 3$ ;BR IF PROGRAM TIMER DID NOT TIME OUT YET
475 ; (TIME OUT = SEVERAL HUNDRED MILLI-SEC)
476 023742 012737 000002 002404 MOV #MCLK,GOODAT ;SET EXPECTED REG CONTENTS
477 023750 013737 002364 002406 MOV REDBYT,BADDAT ;SET ACTUAL REG CONTENTS
478 023756 004737 004016 JSR PC,GETREG ;GET REGS FOR PRINTOUT
479 ;REPORT MCLK BIT STUCK AT 0
480 023762 ERRDF 5,EM5,ERR2
481 023762 104455 TRAP C$ERDF
482 023764 000005 .WORD 5
483 023766 012312 .WORD EM5
484 023770 014632 .WORD ERR2
485 ;TYPE 'PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON'
486 PRINTF #FMT24
487
488 023772 012746 012040 MOV #FMT24,-(SP)
489 023776 012746 000001 MOV #1,-(SP)
490 024002 010600 MOV SP,R0
491 024004 104417 TRAP C$PNTF
492 024006 062706 000004 ADD #4,SP
493 024012 000137 024220 JMP A1 ;ESCAPE TO END OF TEST
494
495 -----
496 ; WAIT FOR MCLK BIT TO BE CLEARED TO 0
497 -----
498 6$: CLR REG0 ;INIT PROGRAM TIMER
499 024016 005037 002510 8$: MOVB @BSEL2,REDBYT ;GET REG 17 INTO REDBYT
500 024022 117737 156424 002364 BITB #MCLK,REDBYT ;SEE IF MCLK BIT = 0 YET
501 024030 132737 000002 002364 BEQ 10$ ;BR IF MCLK = 0
502 024036 001430 INC REG0 ;INCREMENT TIMER
503 024040 005237 002510 BNE 8$ ;BR IF TIMER DID NOT TIME OUT YET
504 024044 001366 CLR GOODAT ;SET EXPECTED REG CONTENTS
505 024046 005037 002404 MOV REDBYT,BADDAT ;SET ACTUAL REG CONTENTS
506 024052 013737 002364 002406 JSR PC,GETREG ;GET REGS FOR PRINTOUT
507 024060 004737 004016 ;REPORT MCLK BIT STUCK AT 1
508 024064 ERRDF 6,EM6,ERR2
509 024064 104455 TRAP C$ERDF
510 024066 000006 .WORD 6
511 024070 012343 .WORD EM6
512 024072 014632 .WORD ERR2
513 ;TYPE 'PLEASE INSURE M8207 RUN SWITCH (E28 SW7) IS ON'
514 PRINTF #FMT24

```

```

024074 012746 012040
024100 012746 000001
024104 010600
024106 104417
024110 062706 000004
502 024114 000137 024220      JMP      A1          ;ESCAPE TO END OF TEST
503
504
505      -----
506      ; WAIT FOR MCLK BIT TO BE SET TO 1 AGAIN
507      -----
507 024120      10$:
508 024120      005037 002510      CLR      REGO          ;INIT PROGRAM TIMER
509 024124      117737 156322 002364 12$:      MOVB     @BSEL2,REDBYT ;GET REG 17 INTO REDBYT
510 024132      132737 000002 002364      BITB     #MCLK,REDBYT ;SEE IF MCLK BIT = 1 YET
511 024140      001027      BNE     A1          ;BE IF MCLK = 1
512 024142      005237 002510      INC      REGO          ;INCREMENT TIMER
513 024146      001366      BNE     12$        ;BR IF TIMER DID NOT TIME OUT YET
514 024150      012737 000002 002404      MOV      #MCLK,GOODAT ;SET EXPECTED REG CONTENTS
515 024156      013737 002364 002406      MOV      REDBYT,BADDAT ;SET ACTUAL REG CONTENTS
516 024164      004737 004016      JSR      PC,GETREG ;GET REGS FOR PRINTOUT
517      ;REPORT MCLK BIT STUCK AT 0
518 024170      ERRDF 5,EM5,ERR2
519 024170      104455
520 024172      000005
521 024174      012312
522 024176      014632
523      ;TYPE 'PLEASE INSURE MB207 RUN SWITCH (E28 SW7) IS ON'
524      PRINTF #FMT24
525
526 024200      012746 012040
527 024204      012746 000001
528 024210      010600
529 024212      104417
530 024214      062706 000004
531 024220
532 024220      004737 003576
533 024224
534 024224      104401
535
536
537
538
539 024226
540 024226      004737 003576
541 024232      005037 002402

      MOV      #FMT24,-(SP)
      MOV      #1,-(SP)
      MOV      SP,R0
      TRAP     C$PNTF
      ADD      #4,SP

      TRAP     C$ERDF
      .WORD    5
      .WORD    EM5
      .WORD    ERR2

      MOV      #FMT24,-(SP)
      MOV      #1,-(SP)
      MOV      SP,R0
      TRAP     C$PNTF
      ADD      #4,SP

      JSR      PC,MSTCLR ;ISSUE MASTER CLEAR
      ENDTST

      L10036:
      TRAP     C$ETST

      *****
      SBTTL     TEST 14 - EXTENDED REGISTER MASTER CLEAR TEST
      *
      * FIRST, ALL READ/WRITE BITS OF EXTENDED REGS AX0-AX3 ARE SET BY LOADING
      * A DIFFERENT WORD OF PATTERN H INTO EACH REG. THEN, A MASTER CLEAR IS
      * ISSUED AND EACH REG IS READ AND COMPARED TO A WORD OF PATTERN I, WHICH
      * CONTAINS THE INITIALIZED STATES OF ALL THE EXTENDED REGS.
      * PATTERN H = 000,000,377,017,377,377,375,377
      * PATTERN I = 000,000,000,000,000,103,000,000
      *****
      BGNTST

      T14::
      JSR      PC,MSTCLR ;ISSUE AN INITIAL MASTER CLEAR
      CLR      AXNUM     ;INIT AX REG BY'E NO. TO 0
  
```

542	024236	012701	002654			MOV	#PATH,R1	;INIT DATA PATTERN POINTER	
543	024242	112137	002374	1\$:		MOVB	(R1)+,WAX15	;SET BITS TO LOAD INTO LO BYTE	
544	024246	112137	002376			MOVB	(R1)+,WAX16	;SET BITS TO LOAD INTO HI BYTE	
545	024252	004737	004312			JSR	PC,WRITAX	;WRITE EXTENDED REG	
546	024256	032737	000002	002420		BIT	#WRDYTO,ERROR1	;SEE IF READY FAILED TO SET	
547	024264	001413				BEQ	8\$	;BR IF NOT	
548	024266	012737	000014	002400		MOV	#14,REGNUM	;SET LU REG NO = 14	
549	024274	004737	004016			JSR	PC,GETREG	;GET REGS FOR PRINTOUT	
550						;REPORT READY NOT SET AFTER AX REG WRITE			
551	024300					ERRDF	52,EM52,ERR9		
	024300	104455						TRAP	C\$ERDF
	024302	000064						.WORD	52
	024304	014105						.WORD	EM52
	024306	020630						.WORD	ERR9
552	024310					ESCAPE TST			
	024310	104410						TRAP	C\$ESCAPE
	024312	000244						.WORD	L10037-.
553	024314	062737	000002	002402	8\$:	ADD	#2,AXNUM	;INCR REG BYTE NO.	
554	024322	020127	002664			CMP	R1,#PATI	;SEE IF ALL REGS WRITTEN YET	
555	024326	103745				BLO	1\$	;BR IF NOT DONE WRITING YET	
556	024330	004737	003576			JSR	PC,MSTCLR	;ISSUE MASTER CLEAR	
557	024334	005037	002402			CLR	AXNUM	;INIT EXTENDED REG BYTE NO. TO 0	
558	024340	012701	002664			MOV	#PATI,R1	;INIT DATA PAT POINTER FOR READS	
559	024344	004737	004124		2\$:	JSR	PC,READAX	;READ AN EXTENDED REG	
560	024350	032737	000001	002420		BIT	#RRDYTO,ERROR1	;SEE IF READY FAILED TO SET	
561	024356	001413				BEQ	10\$	;BR IF NOT	
562	024360	012737	000014	002400		MOV	#14,REGNUM	;SET LU REG NO. = 14	
563	024366	004737	004016			JSR	PC,GETREG	;GET REGS FOR PRINTOUT	
564						;REPORT READY NOT SET AFTER AX REG READ			
565	024372					ERRDF	53,EM53,ERR9		
	024372	104455						TRAP	C\$ERDF
	024374	000065						.WORD	53
	024376	014146						.WORD	EM53
	024400	020630						.WORD	ERR9
566	024402					ESCAPE TST			
	024402	104410						TRAP	C\$ESCAPE
	024404	000152						.WORD	L10037-.
567	024406	023727	002402	000006	10\$:	CMP	AXNUM,#6	;SEE IF AX3-15	
568	024414	001003				BNE	3\$	;BR IF NOT	
569	024416	142737	000372	002370		BICB	#AX315U,RAX15	;MASK OFF UNPREDICTABLE BITS	
570	024424	123711	002370		3\$:	CMPB	RAX15,(R1)	;COMPARE LO BYTE TO EXPECTED VALUE	
571	024430	001417				BEQ	4\$	;BR IF DATA MATCHES	
572	024432	005037	002404			CLR	GOODAT		
573	024436	111137	002404			MOVB	(R1),GOODAT	;GET EXPECTED DATA BYTE	
574	024442	013737	002370	002406		MOV	RAX15,BADDAT	;GET ACTUAL DATA BYTE	
575	024450	004737	004526			JSR	PC,GETALL	;GET REGS FOR PRINTOUT	
576						;REPORT REG NOT INITIALIZED BY MASTER CLEAR			
577	024454					ERRDF	2,EM2,ERR3		
	024454	104455						TRAP	C\$ERDF
	024456	000002						.WORD	2
	024460	012161						.WORD	EM2
	024462	015140						.WORD	ERR3
578	024464					ESCAPE TST			
	024464	104410						TRAP	C\$ESCAPE
	024466	000070						.WORD	L10037-.
579	024470	005237	002402		4\$:	INC	AXNUM	;INCREMENT AX BYTE NO.	
580	024474	005201				INC	R1	;INCREMENT PAT POINTER	

```

581 024476 123711 002372      CMPB    RAX16,(R1)      ;COMPARE HI BYTE TO EXPECTED VALUE
582 024502 001417              BEQ      6$              ;BR IF DATA MATCHES
583 024504 005037 002404      CLR      GOODAT
584 024510 111137 002404      MOVB    (R1),GOODAT      ;GET EXPECTED DATA BYTE
585 024514 013737 002372 002406  MOV    RAX16,BADAT      ;GET ACTUAL DATA BYTE
586 024522 004737 004526      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
587                      ;REPORT REG NOT INITIALIZED BY MASTER CLEAR
588                      ERRDF  2,EM2,ERR3
589 024526 104455              TRAP     C$ERDF
590 024530 000002              .WORD   2
591 024532 012161              .WORD   EM2
592 024534 015140              .WORD   ERR3
593 024536              ESCAPE  TST
594 024536 104410              TRAP     C$ESCAPE
595 024540 000016              .WORD   L10037-.
596 024542 005237 002402      6$:    INC     AXNUM      ;INCR AX BYTE NO.
597 024546 005201              INC     R1              ;INCR PAT POINTER
598 024550 020127 002674      CMP     R1,#PATJ      ;SEE IF ALL REGS READ YET
599 024554 103673              BLO     2$              ;BR IF NOT DONE READING YET
600                      ENDTST
601                      L10037:
602                      TRAP     C$SETST
603 024556 104401
604
605
606
607
608
609
610 024560
611 024560 004737 003576      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
612 024564 012701 002664      MOV     #PAT1,R1      ;INIT POINTER TO PAT 1
613 024570 012702 002500      MOV     #REDDAT,R2      ;INIT POINTER TO EXPECTED DATA AREA
614 024574 112122              3$:    MOVB    (R1)+,(R2)+      ;MOVE PAT 1 INTO REDDAT TABLE
615 024576 020127 002674      CMP     R1,#PATJ
616 024602 103774              BLO     3$
617 024604 005001              CLR     R1              ;INIT INDEX FOR WRITING
618 024606 010137 002402      6$:    MOV     R1,AXNUM      ;GET AX BYTE NO. FOR WRITING
619 024612 116137 002674 002374  MOVB    PATJ(R1),WAX15      ;SET LO AX BYTE TO BE WRITTEN
620 024620 116137 002675 002376  MOVB    PATJ+1(R1),WAX16      ;SET HI AX BYTE TO BE WRITTEN
621 024626 113761 002374 002500  MOVB    WAX15,REDDAT(R1) ;SET EXPECTED LO BYTE IN TABLE
622 024634 113761 002376 002501  MOVB    WAX16,REDDAT+1(R1) ;SET EXPECTED HI BYTE IN TABLE
623 024642 004737 004312      JSR      PC,WRITAX      ;WRITE DATA BYTES INTO EXTENDED REG
624 024646 005003              CLR     R3              ;INIT INDEX FOR READING
625 024650 010337 002402      9$:    MOV     R3,AXNUM      ;GET AX BYTE NO. FOR READING
626 024654 004737 004124      JSR      PC,READAX      ;READ 2 AX BYTES
627 024660 023727 002402 000006  CMP     AXNUM,#6      ;SEE IF AX3-15
628 024666 001003              BNE     10$      ;BR IF NOT

```

```

*****
;SBTTL      TEST 15 - EXTENDED REGISTER ADDRESSING TEST
;
; * FIRST, ISSUE A MASTER CLEAR TO PUT REGS INTO INITIALIZED STATES SHOWN IN
; * PATTERN I. THEN, WRITE A DIFFERENT WORD OF PATTERN J INTO EACH EXTENDED (AX)
; * REG, AND AFTER EACH WRITE, READ AND COMPARE ALL EXTENDED REGS TO EXPECTED
; * VALUES.
; * PATTERN I = 000,000,000,000,000,103,000,000
; * PATTERN J = 000,000,010,002,004,103,001,100
*****
BGNTST

```

T15::

```

629 024670 142737 000372 002370      BICB    #AX315U,RAX15    ;MASK OFF UNPREDICTABLE BITS
630 024676 123763 002370 002500 10$:  CMPB    RAX15,REDDAT(R3) ;COMPARE LO BYTE READ TO EXPECTED
631 024704 001420                BEQ      12$          ;BR IF LO BYTE MATCHES EXPECTED
632 024706 005037 002404          CLR      GOODAT
633 024712 116337 002500 002404      MOVB    REDDAT(R3),GOODAT ;GET EXPECTED LO BYTE
634 024720 013737 002370 002406      MOV     RAX15,BADDAT    ;GET ACTUAL DATA
635 024726 004737 004526          JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
636                                ;REPORT REG MISCOMPARE
637                                ERRDF    3,EM3,ERR3

                                TRAP      C$ERDF
                                .WORD     3
                                .WORD     EM3
                                .WORD     ERR3

638 024732 104455                ESCAPE  TST
                                TRAP      C$ESCAPE
                                .WORD     L10040-.

                                TRAP      C$ERDF
                                .WORD     3
                                .WORD     EM3
                                .WORD     ERR3

639 024732 000003
640 024734 012220
641 024736 015140
642 024740 104410
643 024742 000102
644 024744 005237 002402 002501 12$:  INC      AXNUM          ;INCR AX BYTE NO.
645 024746 123763 002372 002501      CMPB    RAX16,REDDAT+1(R3) ;COMPARE HI BYTE READ TO EXPECTED
646 024752 001420                BEQ      15$          ;BR IF HI BYTE MATCHES EXPECTED
647 024754 005037 002404          CLR      GOODAT
648 024756 116337 002501 002404      MOVB    REDDAT+1(R3),GOODAT ;GET EXPECTED HI BYTE
649 024758 013737 002372 002406      MOV     RAX16,BADDAT    ;GET ACTUAL DATA
650 025002 004737 004526          JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
651                                ;REPORT REG MISCOMPARE
652                                ERRDF    3,EM3,ERR3

                                TRAP      C$ERDF
                                .WORD     3
                                .WORD     EM3
                                .WORD     ERR3

653 025006 104455                ESCAPE  TST
                                TRAP      C$ESCAPE
                                .WORD     L10040-.

654 025006 000003
655 025010 012220
656 025012 015140
657 025014 104410
658 025016 000026
659 025018 000002 000002 15$:  ADD      #2,R3          ;INCR INDEX FOR READS
660 025020 062703 000002          CMP      R3,#10        ;SEE IF ALL AX BYTES READ YET
661 025022 020327 000010          BLT      9$           ;BR IF NOT DONE READING YET
662 025024 002706 000002          ADD      #2,R1          ;INCR INDEX FOR WRITES
663 025026 062701 000002          CMP      R1,#10        ;SEE IF ALL AX BYTES WRITTEN YET
664 025028 020127 000010          BLT      6$           ;BR IF NOT DONE WRITING YET
665 025030 002660
666 025032
667 025034
668 025036
669 025038
670 025040
671 025042
672 025044
673 025046
674 025048
675 025050
676 025052
677 025054
678 025056
679 025058
680 025060
681 025062
682 025064
683 025066
684 025068
685 025070
686 025072
687 025074
688 025076
689 025078
690 025080
691 025082
692 025084
693 025086
694 025088
695 025090
696 025092
697 025094
698 025096
699 025098
700 025100
701 025102
702 025104
703 025106
704 025108
705 025110
706 025112
707 025114
708 025116
709 025118
710 025120
711 025122
712 025124
713 025126
714 025128
715 025130
716 025132
717 025134
718 025136
719 025138
720 025140
721 025142
722 025144
723 025146
724 025148
725 025150
726 025152
727 025154
728 025156
729 025158
730 025160
731 025162
732 025164
733 025166
734 025168
735 025170
736 025172
737 025174
738 025176
739 025178
740 025180
741 025182
742 025184
743 025186
744 025188
745 025190
746 025192
747 025194
748 025196
749 025198
750 025200
751 025202
752 025204
753 025206
754 025208
755 025210
756 025212
757 025214
758 025216
759 025218
760 025220
761 025222
762 025224
763 025226
764 025228
765 025230
766 025232
767 025234
768 025236
769 025238
770 025240
771 025242
772 025244
773 025246
774 025248
775 025250
776 025252
777 025254
778 025256
779 025258
780 025260
781 025262
782 025264
783 025266
784 025268
785 025270
786 025272
787 025274
788 025276
789 025278
790 025280
791 025282
792 025284
793 025286
794 025288
795 025290
796 025292
797 025294
798 025296
799 025298
800 025300
801 025302
802 025304
803 025306
804 025308
805 025310
806 025312
807 025314
808 025316
809 025318
810 025320
811 025322
812 025324
813 025326
814 025328
815 025330
816 025332
817 025334
818 025336
819 025338
820 025340
821 025342
822 025344
823 025346
824 025348
825 025350
826 025352
827 025354
828 025356
829 025358
830 025360
831 025362
832 025364
833 025366
834 025368
835 025370
836 025372
837 025374
838 025376
839 025378
840 025380
841 025382
842 025384
843 025386
844 025388
845 025390
846 025392
847 025394
848 025396
849 025398
850 025400
851 025402
852 025404
853 025406
854 025408
855 025410
856 025412
857 025414
858 025416
859 025418
860 025420
861 025422
862 025424
863 025426
864 025428
865 025430
866 025432
867 025434
868 025436
869 025438
870 025440
871 025442
872 025444
873 025446
874 025448
875 025450
876 025452
877 025454
878 025456
879 025458
880 025460
881 025462
882 025464
883 025466
884 025468
885 025470
886 025472
887 025474
888 025476
889 025478
890 025480
891 025482
892 025484
893 025486
894 025488
895 025490
896 025492
897 025494
898 025496
899 025498
900 025500
901 025502
902 025504
903 025506
904 025508
905 025510
906 025512
907 025514
908 025516
909 025518
910 025520
911 025522
912 025524
913 025526
914 025528
915 025530
916 025532
917 025534
918 025536
919 025538
920 025540
921 025542
922 025544
923 025546
924 025548
925 025550
926 025552
927 025554
928 025556
929 025558
930 025560
931 025562
932 025564
933 025566
934 025568
935 025570
936 025572
937 025574
938 025576
939 025578
940 025580
941 025582
942 025584
943 025586
944 025588
945 025590
946 025592
947 025594
948 025596
949 025598
950 025600
951 025602
952 025604
953 025606
954 025608
955 025610
956 025612
957 025614
958 025616
959 025618
960 025620
961 025622
962 025624
963 025626
964 025628
965 025630
966 025632
967 025634
968 025636
969 025638
970 025640
971 025642
972 025644
973 025646
974 025648
975 025650
976 025652
977 025654
978 025656
979 025658
980 025660
981 025662
982 025664
983 025666
984 025668
985 025670
986 025672
987 025674
988 025676
989 025678
990 025680
991 025682
992 025684
993 025686
994 025688
995 025690
996 025692
997 025694
998 025696
999 025698
1000 025700
1001 025702
1002 025704
1003 025706
1004 025708
1005 025710
1006 025712
1007 025714
1008 025716
1009 025718
1010 025720
1011 025722
1012 025724
1013 025726
1014 025728
1015 025730
1016 025732
1017 025734
1018 025736
1019 025738
1020 025740
1021 025742
1022 025744
1023 025746
1024 025748
1025 025750
1026 025752
1027 025754
1028 025756
1029 025758
1030 025760
1031 025762
1032 025764
1033 025766
1034 025768
1035 025770
1036 025772
1037 025774
1038 025776
1039 025778
1040 025780
1041 025782
1042 025784
1043 025786
1044 025788
1045 025790
1046 025792
1047 025794
1048 025796
1049 025798
1050 025800
1051 025802
1052 025804
1053 025806
1054 025808
1055 025810
1056 025812
1057 025814
1058 025816
1059 025818
1060 025820
1061 025822
1062 025824
1063 025826
1064 025828
1065 025830
1066 025832
1067 025834
1068 025836
1069 025838
1070 025840
1071 025842
1072 025844
1073 025846
1074 025848
1075 025850
1076 025852
1077 025854
1078 025856
1079 025858
1080 025860
1081 025862
1082 025864
1083 025866
1084 025868
1085 025870
1086 025872
1087 025874
1088 025876
1089 025878
1090 025880
1091 025882
1092 025884
1093 025886
1094 025888
1095 025890
1096 025892
1097 025894
1098 025896
1099 025898
1100 025900
1101 025902
1102 025904
1103 025906
1104 025908
1105 025910
1106 025912
1107 025914
1108 025916
1109 025918
1110 025920
1111 025922
1112 025924
1113 025926
1114 025928
1115 025930
1116 025932
1117 025934
1118 025936
1119 025938
1120 025940
1121 025942
1122 025944
1123 025946
1124 025948
1125 025950
1126 025952
1127 025954
1128 025956
1129 025958
1130 025960
1131 025962
1132 025964
1133 025966
1134 025968
1135 025970
1136 025972
1137 025974
1138 025976
1139 025978
1140 025980
1141 025982
1142 025984
1143 025986
1144 025988
1145 025990
1146 025992
1147 025994
1148 025996
1149 025998
1150 026000
1151 026002
1152 026004
1153 026006
1154 026008
1155 026010
1156 026012
1157 026014
1158 026016
1159 026018
1160 026020
1161 026022
1162 026024
1163 026026
1164 026028
1165 026030
1166 026032
1167 026034
1168 026036
1169 026038
1170 026040
1171 026042
1172 026044
1173 026046
1174 026048
1175 026050
1176 026052
1177 026054
1178 026056
1179 026058
1180 026060
1181 026062
1182 026064
1183 026066
1184 026068
1185 026070
1186 026072
1187 026074
1188 026076
1189 026078
1190 026080
1191 026082
1192 026084
1193 026086
1194 026088
1195 026090
1196 026092
1197 026094
1198 026096
1199 026098
1200 026100
1201 026102
1202 026104
1203 026106
1204 026108
1205 026110
1206 026112
1207 026114
1208 026116
1209 026118
1210 026120
1211 026122
1212 026124
1213 026126
1214 026128
1215 026130
1216 026132
1217 026134
1218 026136
1219 026138
1220 026140
1221 026142
1222 026144
1223 026146
1224 026148
1225 026150
1226 026152
1227 026154
1228 026156
1229 026158
1230 026160
1231 026162
1232 026164
1233 026166
1234 026168
1235 026170
1236 026172
1237 026174
1238 026176
1239 026178
1240 026180
1241 026182
1242 026184
1243 026186
1244 026188
1245 026190
1246 026192
1247 026194
1248 026196
1249 026198
1250 026200
1251 026202
1252 026204
1253 026206
1254 026208
1255 026210
1256 026212
1257 026214
1258 026216
1259 026218
1260 026220
1261 026222
1262 026224
1263 026226
1264 026228
1265 026230
1266 026232
1267 026234
1268 026236
1269 026238
1270 026240
1271 026242
1272 026244
1273 026246
1274 026248
1275 026250
1276 026252
1277 026254
1278 026256
1279 026258
1280 026260
1281 026262
1282 026264
1283 026266
1284 026268
1285 026270
1286 026272
1287 026274
1288 026276
1289 026278
1290 026280
1291 026282
1292 026284
1293 026286
1294 026288
1295 026290
1296 026292
1297 026294
1298 026296
1299 026298
1300 026300
1301 026302
1302 026304
1303 026306
1304 026308
1305 026310
1306 026312
1307 026314
1308 026316
1309 026318
1310 026320
1311 026322
1312 026324
1313 026326
1314 026328
1315 026330
1316 026332
1317 026334
1318 026336
1319 026338
1320 026340
1321 026342
1322 026344
1323 026346
1324 026348
1325 026350
1326 026352
1327 026354
1328 026356
1329 026358
1330 026360
1331 026362
1332 026364
1333 026366
1334 026368
1335 026370
1336 026372
1337 026374
1338 026376
1339 026378
1340 026380
1341 026382
1342 026384
1343 026386
1344 026388
1345 026390
1346 026392
1347 026394
1348 026396
1349 026398
1350 026400
1351 026402
1352 026404
1353 026406
1354 026408
1355 026410
1356 026412
1357 026414
1358 026416
1359 026418
1360 026420
1361 026422
1362 026424
1363 026426
1364 026428
1365 026430
1366 026432
1367 026434
1368 026436
1369 026438
1370 026440
1371 026442
1372 026444
1373 026446
1374 026448
1375 026450
1376 026452
1377 026454
1378 026456
1379 026458
1380 026460
1381 026462
1382 026464
1383 026466
1384 026468
1385 026470
1386 026472
1387 026474
1388 026476
1389 026478
1390 026480
1391 026482
1392 026484
1393 026486
1394 026488
1395 026490
1396 026492
1397 026494
1398 026496
1399 026498
1400 026500
1401 026502
1402 026504
1403 026506
1404 026508
1405 026510
1406 026512
1407 026514
1408 026516
1409 026518
1410 026520
1411 026522
1412 026524
1413 026526
1414 026528
1415 026530
1416 026532
1417 026534
1418 026536
1419 026538
1420 026540
1421 026542
1422 026544
1423 026546
1424 026548
1425 026550
1426 026552
1427 026554
1428 026556
1429 026558
1430 026560
1431 026562
1432 026564
1433 026566
1434 026568
1435 026570
1436 026572
1437 026574
1438 026576
1439 026578
1440 026580
1441 026582
1442 026584
1443 026586
1444 026588
1445 026590
1446 026592
1447 026594
1448 026596
1449 026598
1450 026600
1451 026602
1452 026604
1453 026606
1454 026608
1455 026610
1456 026612
1457 026614
1458 026616
1459 026618
1460 026620
1461 026622
1462 026624
1463 026626
1464 026628
1465 026630
1466 026632
1467 026634
1468 026636
1469 026638
1470 026640
1471 026642
1472 026644
1473 026646
1474 026648
1475 026650
1476 026652
1477 026654
1478 026656
1479 026658
1480 026660
1481 026662
1482 026664
1483 026666
1484 026668
1485 026670
1486 026672
1487 026674
1488 026676
1489 026678
1490 026680
1491 026682
1492 026684
1493 026686
1494 026688
1495 026690
1496 026692
1497 026694
1498 026696
1499 026698
1500 026700
1501 026702
1502 026704
1503 026706
1504 026708
1505 026710
1506 026712
1507 026714
1508 026716
1509 026718
1510 026720
1511 026722
1512 026724
1513 026726
1514 026728
1515 026730
1516 026732
1517 026734
1518 026736
1519 026738
1520 026740
1521 026742
1522 026744
1523 026746
1524 026748
1525 026750
1526 026752
1527 026754
1528 026756
1529 026758
1530 026760
1531 026762
1532 026764
1533 026766
1534 026768
1535 026770
1536 026772
1537 026774
1538 026776
1539 026778
1540 026780
1541 026782
1542 026784
1543 026786
1544 026788
1545 026790
1546 026792
1547 026794
1548 026796
1549 026798
1550 026800
1551 026802
1552 026804
1553 026806
1554 026808
1555 026810
1556 026812
1557 026814
1558 026816
1559 026818
1560 026820
1561 026822
1562 026824
1563 026826
1564 026828
1565 026830
1566 026832
1567 026834
1568 026836
1569 026838
1570 026840
1571 026842
1572 026844
1573 026846
1574 026848
1575 026850
1576 026852
1577 026854
1578 026856
1579 026858
1580 026860
1581 026862
1582 026864
1583 026866
1584 026868
1585 026870
1586 026872
1587 026874
1588 026876
1589 026878
1590 026880
1591 026882
1592 026884
1593 026886
1594 026888
1595 026890
1596 026892
1597 026894
1598 026896
1599 026898
1600 0
```

```

672      ;*      FOR REG 16: 000,377,125,252,000,000,000,000,000,000,000,001,002,
673      ;*      004,010,020,040,100,200,377,377,377,377,377,377,377,
674      ;*      376,375,373,367,357,337,277,177.
675      ;*****
676 025050      BGNSTST
677 025050      004737      003576      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
678 025054      012701      002704      MOV      #PATL,R1      ;INIT DATA PATTERN POINTER
679 025060      3$:
680 025060      BGNSEG
681 025062      104404      000004      002402      MOV      #4,AXNUM      ;SET BYTE NO. = 4
682 025070      111137      002374      MOV      (R1),WAX15      ;SET DATA TO WRITE INTO LO BYTE
683 025074      116137      000001      002376      MOV      1(R1),WAX16      ;SET DATA TO WRITE INTO HI BYTE
684 025102      143737      002565      002374      BICB      ANBITS+4,WAX15      ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
685 025110      143737      002566      002376      BICB      ANBITS+5,WAX16      ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
686 025116      004737      004312      JSR      PC,WRITAX      ;LOAD DATA INTO AX2-15,AX2-16
687 025122      004737      004124      JSR      PC,READAX      ;READ AX2-15 AND AX2-16
688 025126      123737      002370      002374      CMPB      RAX15,WAX15      ;COMPARE LO BYTE DATA READ
689 025134      001416      BEQ      6$      ;BR IF DATA MATCHES
690 025136      013737      002374      002404      MOV      WAX15,GOODAT      ;SET EXPECTED DATA
691 025144      013737      002370      002406      MOV      RAX15,BADDAT      ;SET ACTUAL DATA
692 025152      004737      004526      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
693      ;REPORT REG MISCOMPARE
694 025156      ERRDF      3,EM3,ERR3
695 025156      104455      TRAP      C$ERDF
696 025160      000003      .WORD      3
697 025162      012220      .WORD      EM3
698 025164      015140      .WORD      ERR3
699 025166      ESCAPE      SEG
700 025166      104410      TRAP      C$ESCAPE
701 025170      000052      .WORD      10000$-.
702 025172      005237      002402      6$:      INC      AXNUM      ;SET AX BYTE NO. = 5
703 025176      123737      002372      002671      CMPB      RAX16,PATL+5      ;COMPARE HI BYTE DATA READ
704 025204      001416      BEQ      9$      ;BR IF DATA MATCHES
705 025206      005037      002404      CLR      GOODAT
706 025212      113737      002671      002404      MOV      PATL+5,GOODAT      ;SET EXPECTED DATA
707 025220      013737      002372      002406      MOV      RAX16,BADDAT      ;SET ACTUAL DATA
708 025226      004737      004526      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
709      ;REPORT REG MISCOMPARE
710 025232      ERRDF      3,EM3,ERR3
711 025232      104455      TRAP      C$ERDF
712 025234      000003      .WORD      3
713 025236      012220      .WORD      EM3
714 025240      015140      .WORD      ERR3
715 025242      9$:
716 025242      ENDSEG
717 025242      10000$:
718 025242      104405      TRAP      C$ESEG
719 025244      062701      000002      ADD      #2,R1      ;INCR PATTERN POINTER
720 025250      020127      003014      CMP      R1,#PATL      ;SEE IF ALL DATA WRITTEN YET
721 025254      103701      BLO      3$      ;BR IF NOT DONE YET
722      ENDTST
723 025256      104401      L10041:
724 025256      TRAP      C$ETST

```

713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756

025260
 025260
 025260 004737 003576
 025264 012701 003014
 025270
 025270
 025270 104404
 025272 012737 000000 002402
 025300 111137 002374
 025304 110137 000001 002376
 025312 143737 002561 002374
 025320 143737 002562 002376
 025326 004737 004312
 025332 004737 004124
 025336 123737 002370 002374
 025344 001416
 025346 013737 002374 002404
 025354 013737 002370 002406
 025362 004737 004526
 025366
 025366 104455
 025370 000003
 025372 012220
 025374 015140
 025376
 025376 104410
 025400 000046
 025402 005237 002402
 025406 123737 002372 002376
 025414 001414
 025416 013737 002376 002404
 025424 013737 002372 002406
 025432 004737 004526
 025436
 025436 104455
 025440 000003
 025442 012220
 025444 015140
 025446
 025446

```

*****
SBTTL      TEST 17 - AX0-15,AX0-16 READ/WRITE BIT TEST
*
* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
* ARE PERFORMED IN REGS AX0-15,AX0-16 USING EACH WORD OF PATTERN L.
* ANY BITS IN AX0-15,AX0-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
* IN THE EXPECTED VALUE BEFORE COMPARISON.
* PATTERN L =
*   FOR REG 15: 000,377,000
*   FOR REG 16: 000,377,000.
*****
BGNSTST
T17::
      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
      MOV      #PATL,R1      ;INIT DATA PATTERN POINTER
3$:
      BGNSEG
                                TRAP      C$BSEG
      MOV      #0,AXNUM      ;SET BYTE NO. = 0
      MOVB     (R1),WAX15     ;SET DATA TO WRITE INTO LO BYTE
      MOVB     1(R1),WAX16    ;SET DATA TO WRITE INTO HI BYTE
      BICB     ANBITS+0,WAX15 ;MASK OFF NON-READ/WRITE BITS IN LO BYTE
      BICB     ANBITS+1,WAX16 ;MASK OFF NON-READ/WRITE BITS IN HI BYTE
      JSR      PC,WRITAX      ;LOAD DATA INTO AX0-15,AX0-16
      JSR      PC,READAX      ;READ AX0-15 AND AX0-16
      CMPB     RAX15,WAX15    ;COMPARE LO BYTE DATA READ
      BEQ      6$             ;BR IF DATA MATCHES
      MOV      WAX15,GOODAT    ;SET EXPECTED DATA
      MOV      RAX15,BADDAT    ;SET ACTUAL DATA
      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
;REPORT REG MISCOMPARE
      ERDF      3,EM3,ERR3
                                TRAP      C$ERDF
                                .WORD     3
                                .WORD     EM3
                                .WORD     ERR3
      ESCAPE  SEG
                                TRAP      C$ESCAPE
                                .WORD     10000$.
6$:
      INC      AXNUM          ;SET AX BYTE NO. = 1
      CMPB     RAX16,WAX16    ;COMPARE HI BYTE DATA READ
      BEQ      9$             ;BR IF DATA MATCHES
      MOV      WAX16,GOODAT    ;SET EXPECTED DATA
      MOV      RAX16,BADDAT    ;SET ACTUAL DATA
      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
;REPORT REG MISCOMPARE
      ERDF      3,EM3,ERR3
                                TRAP      C$ERDF
                                .WORD     3
                                .WORD     EM3
                                .WORD     ERR3
9$:
      ENDSEG
10000$:

```

757	025446	104405	000002	ADD	#2,R1	;INCR PATTERN POINTER	TRAP	C\$ESEG
758	025450	062701	003022	CMP	R1,#PATM	;SEE IF ALL DATA WRITTEN YET		
759	025454	020127		BLO	3\$	;BR IF NOT DONE YET		
760	025460	103703						
	025462			ENDTST				
	025462						L10042:	
	025462	104401					TRAP	C\$ETST

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

```

*****
:SBTTL      TEST 18 - AX1-15,AX1-16 READ/WRITE BIT TEST
:
:* IN THIS TEST, A MASTER CLEAR IS DONE, AND THEN A WRITE, READ, AND COMPARE
:* ARE PERFORMED IN REGS AX1-15,AX1-16 USING EACH WORD OF PATTERN K.
:* ANY BITS IN AX1-15,AX1-16 WHICH ARE NOT READ/WRITE ARE MASKED OFF (TO 0)
:* IN THE EXPECTED VALUE BEFORE COMPARISON.
*****

```

BGNTST

T18::

775	025464	004737	003576	JSR	PC,MSTCLR	;ISSUE MASTER CLEAR		
776	025470	012701	002704	MOV	#PATK,R1	;INIT DATA PATTERN POINTER		
777	025474			3\$:				
778	025474			BGNSEG				
779	025474	104404					TRAP	C\$BSEG
779	025476	012737	000002	MOV	#2,AXNUM	;SET BYTE NO. = 2		
780	025504	111137	002374	MOVB	(R1),WAX15	;SET DATA TO WRITE INTO LO BYTE		
781	025510	116137	000001	MOVB	1(R1),WAX16	;SET DATA TO WRITE INTO HI BYTE		
782	025516	143737	002563	BICB	ANBITS+2,WAX15	;MASK OFF NON-READ/WRITE BITS IN LO BYTE		
783	025524	143737	002564	BICB	ANBITS+3,WAX16	;MASK OFF NON-READ/WRITE BITS IN HI BYTE		
784	025532	004737	004312	JSR	PC,WRITAX	;LOAD DATA INTO AX1-15,AX1-16		
785	025536	004737	004124	JSR	PC,READAX	;READ AX1-15 AND AX1-16		
786	025542	123737	002370	CMPB	RAX15,WAX15	;COMPARE LO BYTE DATA READ		
787	025550	001416		BEQ	6\$	;BR IF DATA MATCHES		
788	025552	013737	002374	MOV	WAX15,GOODAT	;SET EXPECTED DATA		
789	025560	013737	002370	MOV	RAX15,BADDAT	;SET ACTUAL DATA		
790	025566	004737	004526	JSR	PC,GETALL	;GET REGS FOR PRINTOUT		
791				;REPORT REG MISCOMPARE				
792	025572			ERRDF	3,EM3,ERR3			

TRAP	C\$ERDF
.WORD	3
.WORD	EM3
.WORD	ERR3

ESCAPE SEG

TRAP	C\$ESCAPE
.WORD	10000\$-

794	025606	005237	002402	6\$:	INC	AXNUM	;SET AX BYTE NO. = 3	
795	025612	123737	002372	CMPB	RAX16,WAX16	;COMPARE HI BYTE DATA READ		
796	025620	001414		BEQ	9\$	;BR IF DATA MATCHES		
797	025622	013737	002376	MOV	WAX16,GOODAT	;SET EXPECTED DATA		
798	025630	013737	002372	MOV	RAX16,BADDAT	;SET ACTUAL DATA		
799	025636	004737	004526	JSR	PC,GETALL	;GET REGS FOR PRINTOUT		
800				;REPORT REG MISCOMPARE				
801	025642			ERRDF	3,EM3,ERR3			

TRAP	C\$ERDF
------	---------

Address	Instruction	Comments	Segment	Flags
025644	000003			
025646	012220			
025650	015140			
802 025652				
803 025652				
025652				
025652	104405			
804 025654	062701	000002		
805 025660	020127	003014		
806 025664	103703			
807 025666				
025666				
025666	104401			
808				
809				
810				
811				
812				
813				
814				
815				
816				
817				
818				
819				
820				
821				
822				
823				
824				
825				
826				
827				
828				
829				
830				
831				
832 025670				
025670				
833 025670	004737	003576		
834 025674	142777	000010	154546	
835 025702	012702	003152		
836 025706	012701	003104		
837 025712				
838 025712				
025712				
025712	104404			
839 025714	012737	000006	002402	
840 025722	111237	002374		
841 025726	116237	000001	002376	
842 025734	004737	004312		
843 025740	004737	004124		
844 025744	132737	000001	002374	
845 025752	001003			
846 025754	142737	000372	002370	
847 025762	142737	000040	002370	
848 025770	123711	002370		
849 025774	001417			

```

850 025776 005037 002404          CLR      GOODAT
851 026002 111137 002404          MOV      (R1),GOODAT      ;SET EXPECTED DATA
852 026006 013737 002370 002406  MOV      RAX15,BADDAT      ;SET ACTUAL DATA
853 026014 004737 004526          JSR      PC,GETALL      ;GET REG; FOR PRINTOUT
854          ;REPORT REG MISCOMPARE
855          ERRDF 3,EM3,ERR3

      026020 104455
      026022 000003
      026024 012220
      026026 015140
856 026030          ESCAPE SEG
      026030 104410
      026032 000052
857 026034 005237 002402 000001 6$: INC      AXNUM      ;SET AX BYTE NO. = 7
858 026040 123761 002372          CMPB     RAX16,1(R1)      ;COMPARE HI BYTE DATA READ
859 026046 001416          BEQ      9$      ;BR IF DATA MATCHES
860 026050 005037 002404          CLR      GOODAT
861 026054 116137 000001 002404  MOV      1(R1),GOODAT      ;SET EXPECTED DATA
862 026062 013737 002372 002406  MOV      RAX16,BADDAT      ;SET ACTUAL DATA
863 026070 004737 004526          JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
864          ;REPORT REG MISCOMPARE
865          ERRDF 3,EM3,ERR3

      026074 104455
      026076 000003
      026100 012220
      026102 015140
866 026104          9$:
867 026104          ENDSEG
      026104          10000$:
      026104 104405          TRAP     C$ESEG
868 026106 062702 000002          ADD      #2,R2      ;INCR PATTERN V POINTER
869 026112 062701 000002          ADD      #2,R1      ;INCR PATTERN U POINTER
870 026116 020127 003152          CMP      R1,#PATV      ;SEE IF ALL DATA WRITTEN YET
871 026122 103673          BLO      3$      ;BR IF NOT DONE YET
872 026124          ENDTST
      026124          L10044:
      026124 104401          TRAP     C$ETST

873
874
875
876
877
878
879          ;*****
880          .SBTTL      TEST 20 - REG 17 - AX2-16 READ/WRITE, MASTER CLEAR TEST
881          ;*
882          ;* THIS TEST CONSISTS OF 2 SUBTESTS. IN THE FIRST SUBTEST, EACH BYTE OF PAT 0
883          ;* IS WRITTEN INTO REG 17 AND AFTER EACH WRITE, AX2-16 IS READ AND COMPARED
884          ;* TO A BYTE OF PAT P.
885          ;* PATTERN 0 = 000,041,004,010,020,040,100,101,200,201,300,111,301,375
886          ;* PATTERN P = 000,113,200,040,020,010,001,104,007,105,007,144,107,157
887          ;* IN THE SECOND SUBTEST, REG 17 IS LOADED WITH 375, A MASTER CLEAR IS ISSUED,
888          ;* AND AX2-16 IS COMPARED TO ITS INITIALIZED STATE (103).
889          ;*****
890          BGNST
891          JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
          MOV      #PATO,R1      ;INIT PAT 0 POINTER
  
```

```

892 026136 012702 003066      MOV    #PATP,R2      ;INIT PAT P POINTER
893 026142 012737 000017 002400  MOV    #17,REGNUM    ;SET LU REG NO. = 17
894 026150 012737 000005 002402  MOV    #5,AXNUM      ;SET AX BYTE NO. = 5 FOR AX2-16
895                                     -----
896                                     ; WRITE REG 17, READ AND COMPARE AX2-16
897                                     -----
898 026156      BGNSUB
      026156
      026156 104402      T20.1:      TRAP    C$BSUB
899 026160      3$:
900 026160      BGNSEG
      026160 104404      TRAP    C$BSEG
901 026162 111137 002366      MOVB    (R1),WRIBYT    ;SET BYTE TO BE WRITTEN
902 026166 004737 003750      JSR      PC,WRITLU    ;WRITE DATA BYTE INTO REG 17
903 026172 004737 004124      JSR      PC,READAX    ;READ AX2
904 026176 123712 002372      CMPB    RAX16,(R2)    ;COMPARE AX2-16 TO EXPECTED DATA
905 026202 001421      BEQ      6$      ;BR IF DATA MATCHES
906 026204 005037 002410      CLR      LOADAT
907 026210 111137 002410      MOVB    (R1),LOADAT    ;SET DATA WHICH WAS WRITTEN
908 026214 005037 002404      CLR      GOODAT
909 026220 111237 002404      MOVB    (R2),GOODAT    ;SET EXPECTED DATA READ
910 026224 013737 002372 002406  MOV    RAX16,BADDAT    ;SET ACTUAL DATA READ
911 026232 004737 004526      JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
912      ;REPORT REG MISCOMPARE
913 026236      ERRDF 3,EM3,ERR5
      026236 104455      TRAP    C$ERDF
      026240 000003      .WORD    3
      026242 012220      .WORD    EM3
      026244 016300      .WORD    ERR5
914 026246      6$:
915 026246      ENDSEG
      026246      10000$:      TRAP    C$ESEG
      026246 104405
916 026250 005201      INC      R1      ;INCR PAT O POINTER
917 026252 005202      INC      R2      ;INCR PAT P POINTER
918 026254 020127 003066      CMP      R1,#PATP    ;SEE IF ALL BYTES LOADED YET
919 026260 103737      BLO      3$      ;BR IF NOT
920 026262      ENDSUB
      026262      L10046:      TRAP    C$ESUB
      026262 104403
921      -----
922      ; LOAD REG 17, DO MASTER CLEAR, READ AND COMPARE AX2-16
923      -----
924 026264      BGNSUB
      026264
      026264 104402      T20.2:      TRAP    C$BSUB
925 026266 112737 000375 002366  MOVB    #375,WRIBYT    ;SET DATA TO BE LOADED
926 026274 004737 003750      JSR      PC,WRITLU    ;LOAD DATA INTO REG 17
927 026300 004737 003576      JSR      PC,MSTCLR    ;PERFORM MASTER CLEAR
928 026304 004737 004124      JSR      PC,READAX    ;READ AX2-15,AX2-16
929 026310 123737 002372 002671  CMPB    RAX16,PAT1+5    ;SEE IF AX2-16 WAS INIT'D CORRECTLY
930 026316 001416      BEQ      6$      ;BR IF YES
931 026320 005037 002404      CLR      GOODAT
932 026324 113737 002671 002404  MOVB    PAT1+5,GOODAT    ;SET EXPECTED DATA
933 026332 013737 002372 002406  MOV    RAX16,BADDAT    ;SET ACTUAL DATA
934 026340 004737 004526      JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
935      ;REPORT REG NOT INITIALIZED BY MASTER CLEAR

```

936	026344		ERRDF	2,EM2,ERR3		
	026344	104455			TRAP	C\$ERDF
	026346	000002			.WORD	2
	026350	012161			.WORD	EM2
	026352	015140			.WORD	ERR3
937	026354		6\$:			
938	026354		ENDSUB			
	026354				L10047:	
	026354	104403			TRAP	C\$ESUB
939	026356		ENDTST			
	026356				L10045:	
	026356	104401			TRAP	C\$ETST

```

*****
:SBTTL      TEST 21 - TRANSMITTER BUFFER DATA TEST
:* A MASTER CLEAR IS DONE FIRST, AND THEN A BYTE OF PATTERN N IS LOADED INTO
:* REG 11 AND THE NEXT BYTE IS LOADED TWICE INTO REG 10. THE PROGRAM THEN WAITS
:* AT LEAST 50 MICRO-SEC, AND THEN IT READS AND COMPARES AX1-15 TO THE BYTE
:* WHICH WAS LOADED INTO REG 10, AND IT READS AND COMPARES AX1-16 TO THE BYTE
:* WHICH WAS LOADED INTO REG 11. THIS PROCESS IS REPEATED (INCLUDING THE MASTER
:* CLEAR) FOR EACH PAIR OF BYTES IN PATTERN N.
:* PATTERN N =
:*   FOR REG 10: 000,125,252,377,000,000,000
:*   FOR REG 11: 000,000,000,000,005,012,017
*****
:BGNTST

```

957	026360						
	026360						T21::
958	026360	012701	003032	MOV	#PATN,R1	;INIT PATTERN POINTER	
959	026364	012737	026626	MOV	#A2,RETADR	;SET SUBROUTINE ERROR RETURN ADDRESS	
960	026372		002362	BGNSUB			
	026372						T21.1:
	026372	104402				TRAP	C\$BSUB
961	026374	004737	003576	3\$:	JSR	PC,MSTCLR	;ISSUE MASTER CLEAR
962	026400	004737	004662		JSR	PC,OSIRDY	;CHECK ORDY AND OCOR FOR EXPECTED STATES
963	026404	000001			1		
964	026406	012737	000011	MOV	#11,REGNUM	;SET LU REG NO. = 11	
965	026414	111137	002366	MOVB	(R1),WRIBYT	;SET DATA BYTE TO BE WRITTEN	
966	026420	004737	003750	JSR	PC,WRITLU	;WRITE BYTE INTO REG 11	
967	026424	012737	000010	MOV	#10,REGNUM	;SET LU REG NO. = 10	
968	026432	116137	000001	MOVB	1(R1),WRIBYT	;SET DATA BYTE TO BE WRITTEN	
969	026440	004737	003750	JSR	PC,WRITLU	;WRITE BYTE INTO REG 10	
970	026444	004737	003750	JSR	PC,WRITLU	;WRITE IT AGAIN (SO 2 ENTRIES ARE IN SILO)	
971	026450	004737	005146	JSR	PC,WAIT50	;WAIT FOR SILO DATA TO RIPPLE	
972	026454	004737	004662	JSR	PC,OSIRDY	;CHECK ORDY AND OCOR FOR EXPECTED STATES	
973	026460	000003		3			
974	026462	012737	000002	MOV	#2,AXNUM	;SET BYTE NO. FOR AX1-15	
975	026470	004737	004124	JSR	PC,READAX	;READ AX1-15, AX1-16	
976	026474	123761	002370	CMPB	RAX15,1(R1)	;COMPARE AX1-15 TO EXPECTED	
977	026502	001420		BEQ	6\$	;BR IF MATCH	
978	026504	005037	002404	CLR	GOODAT		
979	026510	116137	000001	MOVB	1(R1),GOODAT	;SET EXPECTED DATA	
980	026516	013737	002370	MOV	RAX15,BADDAT	;SET ACTUAL DATA	
981	026524	004737	004526	JSR	PC,GETALL	;GET REGS FOR PRINTOUT	

```

982                                     ;REPORT REG MISCOMPARE
983 026530                             ERRDF 3,EM3,ERR3
    026530 104455
    026532 000003
    026534 012220
    026536 015140
984 026540                             ESCAPE SUB
    026540 104410
    026542 000064
985 026544 005237 002402
986 026550 123711 002372
987 026554 001417
988 026556 005037 002404
989 026562 111137 002404
990 026566 013737 002372 002406
991 026574 004737 004526
992                                     ;REPORT REG MISCOMPARE
993 026600                             ERRDF 3,EM3,ERR3
    026600 104455
    026602 000003
    026604 012220
    026606 015140
994 026610                             ESCAPE SUB
    026610 104410
    026612 000014
995 026614 062701 000002
996 026620 020127 003050
997 026624 103663
998 026626
999 026626
    026626
    026626 104403
1000 026630
    026630
    026630 104401
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022 026632

                                     ;REPORT REG MISCOMPARE
                                     ERRDF 3,EM3,ERR3

                                     ESCAPE SUB

6$:  INC  AXNUM          ;INCR AX BYTE NO.
     CMPB RAX16,(R1)    ;COMPARE AX1-16 TO EXPECTED
     BEQ  9$            ;BR IF MATCH
     CLR  GOODAT
     MOVB (R1),GOODAT   ;SET EXPECTED DATA
     MOV  RAX16,BADDAT  ;SET ACTUAL DATA
     JSR  PC,GETALL     ;GET REGS FOR PRINTOUT
                                     ;REPORT REG MISCOMPARE
                                     ERRDF 3,EM3,ERR3

                                     ESCAPE SUB

9$:  ADD  #2,R1          ;INCR DATA PATTERN POINTER
     CMP  R1,#PATO      ;SEE IF ALL DATA BYTES WRITTEN YET
     BLO  3$            ;BR IF NOT DONE YET

A2:  ENDSUB

                                     L10051:
                                     TRAP  C$ESUB

                                     L10050:
                                     TRAP  C$ETST

;*****
;SBTTL  TEST 22 - TRANSMITTER BUFFER SEQUENCING TEST
;
; * FIRST, A MASTER CLEAR IS DONE, AND THE PROGRAM CHECKS FOR ORDY=1, OCOR=0.
; * THEN, 2 TSOM CHARS ARE LOADED INTO THE TX SILO, AND ALLOWED TO RIPPLE
; * DOWN TO THE OUTPUT. THE PROGRAM CHECKS FOR ORDY=1, OCOR=1.
; * NEXT, THE PROGRAM CYCLES THE STEPLU BIT UNTIL OCOR=0 AGAIN, AND CHECKS FOR
; * THIS TO OCCUR WITHIN 3 CYCLES.
; * THE SILO IS THEN FILLED WITH 64 BYTES OF A 256-BYTE BINARY COUNT PATTERN
; * (000-377) AND THE PROGRAM CHECKS FOR ORDY=0 AFTER THE 64TH CHAR IS LOADED.
; * THE PROGRAM CYCLES STEPLU FOR 8 CYCLES AND CHECKS THAT AFTER THE 8TH, ORDY=1.
; * AX1-15 IS READ AND COMPARED TO EXPECTED DATA.
; * THE REST OF THE BINARY COUNT DATA BYTES ARE LOADED, CYCLED 8 CLOCKS, READ AND
; * COMPARED, A BYTE AT A TIME. UPON COMPLETION, THE SILO IS CHECKED TO BE EMPTY
; * WITH ORDY=1, OCOR=0.
;*****
BGNTST

```

T22::

```

1023 026632 012737 027306 002362      MOV    #A3,RETADR      ;SET SUBR ERROR RETURN ADDR
1024                                     -----
1025                                     ; SET MASTER CLEAR, CHECK FOR ORDY=1, OCOR=0
1026                                     -----
1027 026640 004737 003576      JSR     PC,MSTCLR      ;ISSUE MASTER CLEAR
1028 026644 004737 004662      JSR     PC,OSIRDY      ;CHK ORDY=1, OCOR=0
1029 026650 000001      1
1030                                     -----
1031                                     ; LOAD 2 SOM CHARS, ALLOW SILO TO RIPPLE, CHK ORDY=1, OCOR=1
1032                                     -----
1033 026652 012737 000400 002422      MOV     #TXSOM,TXWORD    ;SET DATA TO WRITE INTO SILO
1034 026660 004737 005174      JSR     PC,LDTXSI      ;LOAD THE SILO WITH SOM
1035 026664 004737 005174      JSR     PC,LDTXSI      ;LOAD ANOTHER SOM
1036 026670 004737 005146      JSR     PC,WAIT50      ;WAIT FOR DATA TO RIPPLE
1037 026674 004737 004662      JSR     PC,OSIRDY      ;CHK ORDY=1, OCOR=1
1038 026700 000003      3
1039                                     -----
1040                                     ; CLOCK LINE UNIT, CHK FOR OCOR = 0 WITHIN 3 CYCLES
1041                                     -----
1042 026702 005001      CLR     R1          ;INIT CYCLE COUNTER TO 0
1043 026704 012737 000017 002400      MOV     #17,REGNUM      ;SET REG NO. = 17
1044 026712 004737 005254      3$: JSR     PC,STPLU      ;STEP LU 1 CYCLE
1045 026716 000001      1
1046 026720 004737 003672      JSR     PC,READLU      ;READ REG 17
1047 026724 132737 000020 002364      BITB    #OCOR,REDBYT    ;SEE IF OCOR = 0 YET
1048 026732 001404      BEQ     6$          ;BR IF OCOR = 0
1049 026734 005201      INC     R1          ;INCR CYCLE COUNT
1050 026736 020127 000003      CMP     R1,#3      ;SEE IF 3 CYCLES DONE YET
1051 026742 002763      BLT     3$          ;BR IF NO
1052 026744 004737 004662      6$: JSR     PC,OSIRDY      ;CHK ORDY=1, OCOR=0
1053 026750 000001      1
1054                                     -----
1055                                     ; LOAD 64 BINARY COUNT CHARS INTO SILO, CHK ORDY=0
1056                                     -----
1057 026752 005003      CLR     R3          ;INIT PATTERN FOR WRITING
1058 026754 010337 002422      8$: MOV     R3,TXWORD      ;SET DATA TO BE WRITTEN
1059 026760 004737 005174      JSR     PC,LDTXSI      ;LOAD DATA CHAR INTO TX SILO
1060 026764 004737 005146      JSR     PC,WAIT50      ;WAIT FOR DATA TO RIPPLE IN SILO
1061 026770 020327 000077      CMP     R3,#63.      ;SEE IF 64TH CHAR JUST LOADED
1062 026774 001004      BNE     9$          ;BR IF NO
1063 026776 012737 000002 027020      MOV     #2,14$      ;SET UP TO CHK ORDY=0,OCOR=1
1064 027004 000403      BR      12$
1065 027006 012737 000003 027020      9$: MOV     #3,14$      ;SET UP TO CHK ORDY=1,OCOR=1
1066 027014 004737 004662      12$: JSR     PC,OSIRDY      ;CHK ORDY, OCOR
1067 027020 000000      14$: .WORD    0
1068 027022 005203      INC     R3          ;INCR PATTERN FOR WRITES
1069 027024 020327 000100      CMP     R3,#64.      ;SEE IF 64 CHARS LOADED YET
1070 027030 002751      BLT     8$          ;BR IF NO
1071                                     -----
1072                                     ; CLOCK LINE UNIT, CHECK ORDY = 1 WITHIN 8 CYCLES
1073                                     -----
1074 027032 012737 000011 002400      MOV     #11,REGNUM      ;SET REG NO. = 11
1075 027040 005001      CLR     R1          ;INIT CYCLE COUNT
1076 027042 004737 005254      16$: JSR     PC,STPLU      ;CLOCK LU FOR 1 CYCLE
1077 027046 000001      1
1078 027050 004737 003672      JSR     PC,READLU      ;READ REG 11

```

```

1079 027054 132737 000020 002364      BITB      #ORDY,REDBYT      ;SEE IF ORDY = 1 YET
1080 027062 001004                      BNE        19$          ;BR IF YES
1081 027064 005201                      INC        R1          ;INCR CYCLE COUNT
1082 027066 020127 000010      CMP        R1,#8.        ;SEE IF 8 CYCLES YET
1083 027072 002763                      BLT        16$          ;BR IF NOT YET
1084 027074 004737 004662      19$:      JSR        PC,OSIRDY      ;CHK ORDY = 1, OCOR = 1
1085 027100 000003                      3
1086
1087      ;-----
1088      ; READ AND COMPARE FIRST CHARACTER IN AX1-15
1089      ;-----
1089 027102 005004                      CLR        R4          ;INIT PATTERN FOR READING
1090 027104 012737 000002 002402      MOV        #2,AXNUM      ;SET AX BYTE NO. FOR AX1-15
1091 027112 004737 004124      JSR        PC,READAX      ;READ AX1-15
1092 027116 123704 002370      CMPB       RAX15,R4      ;COMPARE AX1-15 TO EXPECTED
1093 027122 001415                      BEQ        20$          ;BR IF MATCH
1094 027124 010437 002404      MOV        R4,GOODAT      ;SET EXPECTED DATA
1095 027130 013737 002370 002406      MOV        RAX15,BADDAT    ;SET ACTUAL DATA
1096 027136 004737 004526      JSR        PC,GETALL      ;GET REGS FOR PRINTOUT
1097      ;REPORT REG MISCMPARE
1098      ERRDF      3,EM3,ERR3
1098 027142 104455                      TRAP       C$ERDF
1098 027142 000003                      .WORD     3
1098 027146 012220                      .WORD     EM3
1098 027150 015140                      .WORD     ERR3
1099      ESCAPE      TST
1099 027152 104410                      TRAP       C$ESCAPE
1099 027154 000132                      .WORD     L10052-.
1100 027156
1101      20$:
1102      ;-----
1103      ; LOAD AND COMPARE REST OF CHARS, MONITOR ORDY, OCOR
1104      ;-----
1104 027156 020327 000377      24$:      CMP        R3,#255.      ;SEE IF ALL CHARS LOADED YET
1105 027162 003010                      BGT        26$          ;BR IF YES
1106 027164 010337 002422      MOV        R3,TXWORD      ;SET DATA TO BE WRITTEN
1107 027170 004737 005174      JSR        PC,LDTXSI      ;LOAD DATA CHAR INTO TX SILO
1108 027174 005203                      INC        R3          ;INCR DATA TO BE WRITTEN
1109 027176 004737 004662      JSR        PC,OSIRDY      ;CHK ORDY=0, OCOR=1
1110 027202 000002                      2
1111 027204 005204                      26$:      INC        R4          ;INCR PAT FOR READING
1112 027206 004737 005254      JSR        PC,STPLU        ;CLOCK LINE UNIT FOR 8 CYCLES
1113 027212 000010                      8.
1114 027214 004737 004124      JSR        PC,READAX      ;READ AX1-15
1115 027220 123704 002370      CMPB       RAX15,R4      ;COMPARE AX1-15 TO EXPECTED
1116 027224 001415                      BEQ        27$          ;BR IF MATCH
1117 027226 010437 002404      MOV        R4,GOODAT      ;SET EXPECTED DATA
1118 027232 013737 002370 002406      MOV        RAX15,BADDAT    ;SET ACTUAL DATA
1119 027240 004737 004526      JSR        PC,GETALL      ;GET REGS FOR PRINTOUT
1120      ;REPORT REG MISCMPARE
1121      ERRDF      3,EM3,ERR3
1121 027244 104455                      TRAP       C$ERDF
1121 027244 000003                      .WORD     3
1121 027250 012220                      .WORD     EM3
1121 027252 015140                      .WORD     ERR3
1122      ESCAPE      TST
1122 027254 104410                      TRAP       C$ESCAPE
1122 027256 000030                      .WORD     L10052-.
1123 027260 020427 000377      27$:      CMP        R4,#255.      ;SEE IF WE READ LAST CHAR YET

```

1124	027264	001004		BNE	29\$	;BR IF NOT
1125	027266	004737	004662	JSR	PC,OSIRDY	;CHK ORDY=1, OCOR=0
1126	027272	000001		1		
1127	027274	000404		BR	A3	
1128	027276	004737	004662	29\$: JSR	PC,OSIRDY	;CHK ORDY=1, OCOR=1
1129	027302	000003		3		
1130	027304	000724		BR	24\$	;CONTINUE
1131	027306			A3:		
1132	027306			ENDTST		
	027306	104401				L10052: TRAP C\$ETST

1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151

```

*****
:SBTTL      TEST 23 - TX MSG TIMING TEST, CHAR MODE, WITH CRC
:
:* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
:* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
:* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
:* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-16.
:* THE FOLLOWING STEPS ARE DONE:
:* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
:* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
:* THEN 2 TERMINATING SYNCHS ARE SENT.
:* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
:* CLEARED STATE AFTER THE 3RD SYNCH COMPLETES.
*****
:BGNTST
  
```

1152	027310						
	027310						T23::
1153	027310	012737	027422	002362	MOV	#A4,RETADR	;SET TEST EXIT ADDRESS FOR ERRORS
1154	027316	004737	005540		JSR	PC,INITRN	;DO MASTER CLR, !OAD 2 SOM'2
1155	027322	000226			SYNCH		
1156	027324	000011			STRIP!DDCMP		
1157	027326	004737	006122		JSR	PC,TXCHAR	;LOAD A 000 CHAR, TX FIRST SYNCH (226)
1158	027332	000000			000		
1159	027334	100010			CHPCHK!8.		
1160	027336	004737	006122		JSR	PC,TXCHAR	;LOAD 2ND 000 CHAR, TX 2ND SYNCH
1161	027342	000000			000		
1162	027344	000010			8.		
1163	027346	004737	006122		JSR	PC,TXCHAR	;LOAD EOM CHAR, TX FIRST 000 CHAR
1164	027352	001000			TXEOM		
1165	027354	000010			8.		
1166	027356	004737	006122		JSR	PC,TXCHAR	;LOAD EOM CHAR, TX 2ND 000 CHAR
1167	027362	001000			TXEOM		
1168	027364	000010			8.		
1169	027366	004737	006122		JSR	PC,TXCHAR	;LOAD EOM, TX CRC-16 CHAR
1170	027372	001000			TXEOM		
1171	027374	000020			16.		
1172	027376	004737	006122		JSR	PC,TXCHAR	;LOAD EOM, TX FIRST TERMINATING SYNCH
1173	027402	001000			TXEOM		
1174	027404	000010			8.		
1175	027406	004737	006122		JSR	PC,TXCHAR	;LOAD EOM, TX 2ND TERMINATING SYNCH
1176	027412	001000			TXEOM		
1177	027414	000010			8.		

```

1178 027416 004737 006274      JSR      PC,ENDTRN      ;SET OC, MONITOR OCOR
1179 027422                                     A4:
1180 027422 004737 003576      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
1181 027426                                     ENDTST
      027426                                     L10053:
      027426 104401                                     TRAP      C$ETST
  
```

1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200

```

*****
SBTTL      TEST 24 - TX MSG TIMING TEST, BIT MODE, WITH CRC
*
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOOP)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* BIT ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND CRC-CCITT-1.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN BIT MODE.
* SOM IS SET TWICE TO SEND 2 FLAG CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING FLAGS ARE SENT.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE.
*****
BGNST
  
```

```

1201 027430
      027430
1202 027430 012737 027532 002362      MOV      #A5,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
1203 027436 004737 005540      JSR      PC,INTRN      ;DO MASTER CLR, LOAD 2 SOM'2
1204 027442 000000      000
1205 027444 000000      000
1206 027446 004737 006122      JSR      PC,TXCHAR      ;LOAD A 000 CHAR, TX FIRST FLAG
1207 027452 000000      000
1208 027454 100010      CHPCHK!8.
1209 027456 004737 006122      JSR      PC,TXCHAR      ;LOAD 2ND 000 CHAR, TX 2ND FLAG
1210 027462 000000      000
1211 027464 000010      8.
1212 027466 004737 006122      JSR      PC,TXCHAR      ;LOAD EOM CHAR, TX FIRST 000 CHAR
1213 027472 001000      TXEOM
1214 027474 000010      8.
1215 027476 004737 006122      JSR      PC,TXCHAR      ;LOAD EOM, TX 2ND 000 CHAR AND CRC-CCITT-1 CHAR
1216 027502 001000      TXEOM
1217 027504 000030      24.
1218 027506 004737 006122      JSR      PC,TXCHAR      ;LOAD EOM, TX FIRST TERMINATING FLAG
1219 027512 001000      TXEOM
1220 027514 000010      8.
1221 027516 004737 006122      JSR      PC,TXCHAR      ;LOAD EOM, TX 2ND TERMINATING FLAG
1222 027522 001000      TXEOM
1223 027524 000010      8.
1224 027526 004737 006274      JSR      PC,ENDTRN      ;SET OC, MONITOR OCOR
1225 027532                                     A5:
1226 027532 004737 003576      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
1227 027536                                     ENDTST
      027536                                     L10054:
      027536 104401                                     TRAP      C$ETST
  
```

1228
 1229

1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248

```

*****
SBTTL      TEST 25 - TX MSG TIMING TEST, CHAR MODE, WITH NO CRC
*
* IN THIS TEST, AN ENTIRE MESSAGE IS TRANSMITTED (USING STEPLU AND LULOO)
* AND THE PROGRAM MONITORS THE OCCURRENCE OF USYRT TX BUFFER EMPTY FLAGS
* (BY SCANNING ORDY AND OCOR) AND OACT, AT EACH STEP. THE TEST IS DONE IN
* CHARACTER ORIENTED PROTOCOL MODE, USING 8-BIT CHARS AND NO ERROR CHECKING.
* THE FOLLOWING STEPS ARE DONE:
* A MASTER CLEAR IS DONE, AND THE LINE UNIT IS PLACED IN CHAR MODE.
* SOM IS SET TWICE TO SEND 2 SYNCH CHARS. THEN, 2 000 CHARS ARE SENT, AND
* THEN 2 TERMINATING SYNCHS ARE SENT.
* THE TEST IS PERFORMED WITH TXEN (REG14, BIT6) SET, AND THE PROGRAM CHECKS
* THAT THIS HOLDS RTS HIGH PAST THE END OF THE MESSAGE.
* THE TRANSMITTER IS THEN DISABLED, USING OC, AND OACT IS MONITORED FOR THE
* CLEARED STATE.
*****
BGNTST
  
```

1249 027540
 027540
 1250 027540 012737 027726 002362
 1251 027546 004737 005540
 1252 027552 000226
 1253 027554 000311
 1254 027556 012737 000014 002400
 1255 027564 012737 000100 002366
 1256 027572 004737 003750
 1257 027576 004737 006122
 1258 027602 000000
 1259 027604 100010
 1260 027606 004737 006122
 1261 027612 000000
 1262 027614 000010
 1263 027616 004737 006122
 1264 027622 001000
 1265 027624 000010
 1266 027626 004737 006122
 1267 027632 001000
 1268 027634 000010
 1269 027636 004737 006122
 1270 027642 001000
 1271 027644 000010
 1272 027646 004737 006122
 1273 027652 001000
 1274 027654 000010
 1275 027656 004737 005254
 1276 027662 000030
 1277 027664 012737 000013 002400
 1278 027672 004737 003672
 1279 027676 032737 000040 002364
 1280 027704 001006
 1281 027706 004737 004526
 1282
 1283 027712
 027712 104455
 027714 000074

```

T25::
MOV      #A6,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
JSR      PC,INITRN       ;DO MASTER CLR, LOAD 2 SOM'2
SYNCH
CRC2!CRC1!STRIP!DDCMP
MOV      #14,REGNUM      ;SET REG NO. = 14
MOV      #TXEN,WRIBYT    ;SET TXEN BIT
JSR      PC,WRITLU       ;LOAD TXEN BIT IN REG 14
JSR      PC,TXCHAR       ;LOAD A 000 CHAR, TX FIRST SYNCH (226)
000
CHPCHK!8.
JSR      PC,TXCHAR       ;LOAD 2ND 000 CHAR, TX 2ND SYNCH
000
8.
JSR      PC,TXCHAR       ;LOAD EOM CHAR, TX FIRST 000 CHAR
TXEOM
8.
JSR      PC,TXCHAR       ;LOAD EOM CHAR, TX 2ND 000 CHAR
TXEOM
8.
JSR      PC,TXCHAR       ;LOAD EOM, TX FIRST TERMINATING SYNCH
TXEOM
8.
JSR      PC,TXCHAR       ;LOAD EOM, TX 2ND TERMINATING SYNCH
TXEOM
8.
JSR      PC,STPLU        ;CLK PAST END OF MSG
24.
MOV      #13,REGNUM      ;SET REG NO. = 13
JSR      PC,READLU       ;READ REG 13
BIT      #RTS,REDBYT     ;CHK FOR RTS STILL SET
BNE      3$              ;BR IF RTS SET
JSR      PC,GETALL       ;GET REGS FOR PRINTOUT
;REPORT RTS NOT SET
ERRDF    60,EM60,ERR7
  
```

TRAP CSERDF
 .WORD 60

.WORD EM60
 .WORD ERR7

027716 014230
 027720 017470
 1284 027722 004737 006274
 1285 027726
 1286 027726 004737 003576
 1287 027732
 027732
 027732 104401

3\$: JSR PC,ENDTRN ;SET OC, MONITOR OCOR
 A6: JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
 ENDTST

L10055: TRAP C\$ETST

1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306

 SBTTL TEST 26 - TX UNDERRUN SET AND CLEAR TEST - CHAR MODE
 *
 * IN THIS TEST, A TX UNDERRUN ERROR IS FORCED IN EACH OF 2 SITUATIONS,
 * AND THEN CLEARED DIFFERENTLY IN EACH.
 * IN THE FIRST, A MESSAGE IS INITIATED, A 000 CHAR IS SENT, AND THE TX
 * BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG,
 * WHICH CAUSES UNRR TO SET IN REG 11. THEN, SOM IS SET TO CLEAR THE ERROR,
 * AND THIS IS VERIFIED.
 * IN THE SECOND SITUATION, A MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX
 * BUFFER IS NOT SERVICED IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH
 * AGAIN CAUSES UNRR TO SET. THEN, A MASTER CLEAR IS DONE, AND THE UNRR BIT
 * IS CHECKED TO BE CLEARED.

1307 027734
 027734

BGNTST

T26::

 CAUSE TX UNDERRUN, CHK UNRR = 1; SET SOM, CHK UNRR = 0

1308
 1309
 1310
 1311 027734 012737 030272 002362
 1312 027742 004737 005540
 1313 027746 000226
 1314 027750 000011
 1315 027752 004737 006122
 1316 027756 000000
 1317 027760 100010
 1318 027762 004737 005254
 1319 027766 000016
 1320 027770 012737 000011 002400
 1321 027776 004737 003672
 1322 030002 132737 000001 002364
 1323 030010 001410
 1324 030012 004737 004526
 1325

MOV #A7,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
 JSR PC,INITRN ;DO MASTER CLEAR, LOAD 2 SOM'S
 SYNCH
 STRIP!DDCMP
 JSR PC,IXCHAR ;LOAD A 000 CHAR, TX FIRST SYNCH
 000
 CHPCBK!8.
 JSR PC,STPLU ;CLOCK THE TRANSMITTER UNTIL 7 BITS OF
 14. ; THE 000 CHAR HAVE BEEN TRANSMITTED
 MOV #11,REGNUM ;SET LU REG NO. = 11
 JSR PC,READLU ;READ REG 11
 BITB #UNRR,REDBYT ;CHK FOR UNRR = 0
 BEQ 6\$;BR IF UNRR = 0
 JSR PC,GETALL ;GET REGS FOR PRINTOUT
 ;REPORT UNRR NOT CLEARED
 ERRDF 16,EM16,ERR7

TRAP C\$ERDF
 .WORD 16
 .WORD EM16
 .WORD ERR7

1326 030016
 030016 104455
 030020 000020
 030022 012622
 030024 017470

1327 030026 000137 030272
 1328 030032 004737 005254
 1329 030036 000003
 1330 030040 004737 003672
 1331 030044 132737 000001 002364

6\$: JMP A7 ;SKIP TO END OF TEST
 JSR PC,STPLU ;CLOCK LAST BIT OF 000 CHAR
 3
 JSR PC,READLU ;READ REG 11
 BITB #UNRR,REDBYT ;CHK FOR UNRR = 1

```

1332 030052 001010          BNE      9$          ;BR IF UNRR = 1
1333 030054 004737 004526    JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
1334                      ;REPORT UNRR NOT SET
1335 030060          ERRDF    14,EM14,ERR7

                                TRAP      C$ERDF
                                .WORD     14
                                .WORD     EM14
                                .WORD     ERR7

1336 030070 000137 030272    JMP      A7          ;SKIP TO END OF TEST
1337 030074 012737 000400 002422 9$:    MOV     #TXSOM, TXWORD ;SET SOM CHAR TO BE WRITTEN
1338 030102 004737 005174    JSR      PC,LDTXSI    ;LOAD SOM CHAR INTO TX SILO
1339 030106 004737 005146    JSR      PC,WAIT50    ;WAIT FOR SILO DATA TO RIPPLE
1340 030112 004737 005254    JSR      PC,STPLU     ;CLOCK LU FOR 2 CYCLES
1341 030116 000002          2
1342 030120 004737 003672    JSR      PC,READLU    ;READ REG 11
1343 030124 132737 000001 002364    BITB     #UNRR, REDBYT ;CHK FOR UNRR = 0
1344 030132 001410          BEQ      12$          ;BR IF UNRR = 0
1345 030134 004737 004526    JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
1346                      ;REPORT UNRR NOT CLEARED BY SOM
1347 030140          ERRDF    13,EM13,ERR7

                                TRAP      C$ERDF
                                .WORD     13
                                .WORD     EM13
                                .WORD     ERR7

1348 030150 000137 030272    JMP      A7          ;SKIP TO END OF TEST
1349 030154          12$:
1350                      ;-----
1351                      ;CAUSE TX UNDERRUN, CHK UNRR = 1; DO MASTER CLR, CHK UNRR - 0
1352                      ;-----
1353 030154 004737 005540    JSR      PC,INITRN    ;DO MASTER CLEAR, LOAD 2 SOM'S
1354 030160 000226          SYNCH
1355 030162 000051          IDLE.STRIP!DDCMP
1356 030164 004737 006122    JSR      PC, TXCHAR    ;LOAD A 000 CHAR, TX FIRST SYNCH
1357 030170 000000          000
1358 030172 100010          CHPCHK.8.
1359 030174 004737 005254    JSR      PC,STPLU     ;STEP THE LU UNTIL 000 HAS BEN TRANSMITTED
1360 030200 000021          17.
1361 030202 004737 003672    JSR      PC,READLU    ;READ REG 11
1362 030206 132737 000001 002364    BITB     #UNRR, REDBYT ;CHK FOR UNR = 1
1363 030214 001010          BNE      16$          ;BR IF UNRR = 1
1364 030216 004737 004526    JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
1365                      ;REPORT UNRR NOT SET
1366 030222          ERRDF    14,EM14,ERR7

                                TRAP      C$ERDF
                                .WORD     14
                                .WORD     EM14
                                .WORD     ERR7

1367 030232 000137 030272    JMP      A7          ;SKIP TO END OF TEST
1368 030236 004737 003576 16$:    JSR      PC,MSTCLR    ;ISSUE MASTER CLEAR
1369 030242 004737 003672    JSR      PC,READLU    ;READ REG 11
1370 030246 132737 000001 002364    BITB     #UNRR, REDBYT ;CHK FOR UNRR = 0
1371 030254 001406          BEQ      A7          ;BR IF UNRR = 0
1372 030256 004737 004526    JSR      PC,GETALL    ;GET REGS FOR PRINTOUT
1373                      ;REPORT UNRR NOT CLEARED BY OC
1374 030262          ERRDF    15,EM15,ERR7

                                TRAP      C$ERDF
                                .WORD     15
  
```

.WORD EM15
 .WORD ERR7

L10056: TRAP C\$ETST

1375 030266 012573
 1376 030270 017470
 1377 030272 004737 003576
 1378 030276
 1379 030276 104401
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391 030300

A7: JSR PC,MSTCLR ;ISSUE CLEAN-UP MASTER CLEAR
 ENDTST

 SBTTL TEST 27 - TRANSMIT CHAR LENGTH TIMING TEST - CHAR MODE, CRC

* THE LINE UNIT IS PLACED IN CHAR MODE (DDCMP) AND A MESSAGE IS INITIATED
 * WITH AN 8-BIT SYNCH AND A 5-BIT SYNCH CHAR. NEXT, A 000 CHAR IS SENT WITH
 * EACH OF THE FOLLOWING TX CHAR LENGTHS : 5 BITS, 6 BITS, 7 BITS, 8 BITS.
 * (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED). TWO
 * TERMINATING SYNCHS ARE SENT AFTER THE DATA.

BGNTST

T27::

1392 030300 012737 030506 002362
 1393 030306 004737 005540
 1394 030312 000000
 1395 030314 000041
 1396 030316 012737 000006 002402
 1397 030324 012737 000000 002374
 1398 030332 012737 000240 002376
 1399 030340 004737 004312
 1400 030344 004737 006122
 1401 030350 000000
 1402 030352 100010
 1403 030354 004737 006122
 1404 030360 000000
 1405 030362 000005
 1406 030364 012737 000300 002376
 1407 030372 004737 004312
 1408 030376 004737 006122
 1409 030402 000000
 1410 030404 000005
 1411 030406 012737 000340 002376
 1412 030414 004737 004312
 1413 030420 004737 006122
 1414 030424 000000
 1415 030426 000006
 1416 030430 012737 000000 002376
 1417 030436 004737 004312
 1418 030442 004737 006122
 1419 030446 001000
 1420 030450 000007
 1421 030452 004737 006122
 1422 030456 001000
 1423 030460 000010
 1424 030462 004737 006122
 1425 030466 001000
 1426 030470 000020

MOV #A8,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
 JSR PC,INITRN ;DO MASTER CLR, LOAD 2 SOM'S
 000
 IDLE!DDCMP
 MOV #6,AXNUM ;SET BYTE NO. = 6 FOR AX3
 MOV #000,WAX15 ;SET DATA FOR AX3-15 = 0
 MOV #TXLEN2!TXLENO,WAX16 ;SET TX LENGTH = 5 FOR AX3-16
 JSR PC,WRITAX ;LOAD AX3-15,AX3-16
 JSR PC,TXCHAR ;LOAD 5-BIT 000 CHAR, TX 8-BIT SYNCH
 000
 CHPCMK!8.
 JSR PC,TXCHAR ;LOAD 6-BIT 000 CHAR, TX 5-BIT SYNCH
 000
 5
 MOV #TXLEN2!TXLEN1,WAX16
 JSR PC,WRITAX ;SET TX CHAR LENGTH = 6
 JSR PC,TXCHAR ;LOAD 7-BIT 000 CHAR, TX 5-BIT 000 CHAR
 000
 5
 MOV #TXLEN2!TXLEN1!TXLENO,WAX16
 JSR PC,WRITAX ;SET TX CHAR LENGTH = 7
 JSR PC,TXCHAR ;LOAD 8-BIT 000 CHAR, TX 6-BIT 000 CHAR
 000
 6
 MOV #000,WAX16
 JSR PC,WRITAX ;SET TX CHAR LENGTH = 8
 JSR PC,TXCHAR ;LOAD EOM, TX 7-BIT 000 CHAR
 TXEOM
 7
 JSR PC,TXCHAR ;LOAD EOM, TX 8-BIT 000 CHAR
 TXEOM
 8.
 JSR PC,TXCHAR ;LOAD EOM, TX CRC-16 CHAR
 TXEOM
 16.

```

1427 030472 004737 006122      JSR      PC,TXCHAR      ;LOAD EOM, TX FIRST TERMINATING SYNCH
1428 030476 001000              TXEOM
1429 030500 000010              8.
1430 030502 004737 006274      JSR      PC,ENDTRN      ;CLEAR TRANSMITTER
1431 030506                      A8:
1432 030506                      ENDTST
1433 030506                      L10057:
1434 030506                      TRAP      C$ETST
1435 104401

```

1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447

```

*****
:SBTTL      TEST 28 - TRANSMIT CHAR LENGTH TIMING TEST - BIT MODE, CRC
:
:  * THE LINE UNIT IS PLACED IN BIT MODE AND A MESSAGE IS INITIATED
:  * WITH 2 FLAG CHARS. NEXT, 2 8-BIT 000 CHARS ARE SENT, FOLLOWED BY 000 CHARS
:  * WITH EACH OF THE FOLLOWING TRANSMITTER CHAR LENGTHS:
:  *   1 BIT, 2 BITS, 3 BITS, 4 BITS, 5 BITS, 6 BITS, 7 BITS, AND 8 BITS.
:  * (FOR EXAMPLE, A 5-BIT CHAR REQUIRES 5 CLOCK CYCLES TO BE TRANSMITTED).
:  * TWO TERMINATING FLAGS ARE SENT AFTER THE DATA.
:*****
:BGNTST

```

```

1448 030510
1449 030510 012737 031036 002362      MOV      #A9,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
1450 030516 004737 005540              JSR      PC,INITRN      ;DO MASTER CLR, LOAD 2 SOM'S
1451 030522 000000              000
1452 030524 000000              000
1453 030526 004737 006122      JSR      PC,TXCHAR      ;LOAD FIRST 8-BIT 000 CHAR, TX FIRST FLAG
1454 030532 000000              000
1455 030534 100010              CHPK.8.
1456 030536 004737 006122      JSR      PC,TXCHAR      ;LOAD 2ND 8-BIT 000 CHAR, TX 2ND FLAG
1457 030542 000000              000
1458 030544 000010              8.
1459 030546 004737 006122      JSR      PC,TXCHAR      ;LOAD 1-BIT 000 CHAR, TX FIRST 8-BIT 000 CHAR
1460 030552 000000              000
1461 030554 000010              8.
1462 030556 012737 000006 002402      MOV      #6,AXNUM      ;SET BYTE NO. = 6 FOR AX3
1463 030564 012737 000000 002374      MOV      #000,WAX15      ;SET DATA FOR AX3-15 = 0
1464 030572 012737 000040 002376      MOV      #TXLEN0,WAX16      ;SET TX CHAR LENGTH = 1 FOR AX3-16
1465 030600 004737 004312              JSR      PC,WRITAX      ;LOAD AX3-15,AX3-16
1466 030604 004737 006122      JSR      PC,TXCHAR      ;LOAD 2-BIT 000 CHAR, TX 2ND 8-BIT 000 CHAR
1467 030610 000000              000
1468 030612 000010              8.
1469 030614 012737 000100 002376      MOV      #TXLEN1,WAX16      ;SET TX CHAR LENGTH = 2
1470 030622 004737 004312              JSR      PC,WRITAX      ;LOAD 3-BIT 000 CHAR, TX 1-BIT 000 CHAR
1471 030626 004737 006122      JSR      PC,TXCHAR
1472 030632 000000              000
1473 030634 000001              1
1474 030636 012737 000140 002376      MOV      #TXLEN1!TXLEN0,WAX16      ;SET TX CHAR LENGTH = 3
1475 030644 004737 004312              JSR      PC,WRITAX      ;LOAD 4-BIT 000 CHAR, TX 2-BIT 000 CHAR
1476 030650 004737 006122      JSR      PC,TXCHAR
1477 030654 000000              000
1478 030656 000002              2
1479 030660 012737 000200 002376      MOV      #TXLEN2,WAX16      ;SET TX CHAR LENGTH = 4
1480 030666 004737 004312      JSR      PC,WRITAX

```

```

1481 030672 004737 006122 JSR PC,TXCHAR ;LOAD 5-BIT 000 CHAR, TX 3-BIT 000 CHAR
1482 030676 000000 000
1483 030700 000003 3
1484 030702 012737 000240 002376 MOV #TXLEN2!TXLEN0,WAX16
1485 030710 004737 004312 JSR PC,WRITAX ;SET TX CHAR LENGTH = 5
1486 030714 004737 006122 JSR PC,TXCHAR ;LOAD 6-BIT 000 CHAR, TX 4-BIT 000 CHAR
1487 030720 000000 000
1488 030722 000004 4
1489 030724 012737 000300 002376 MOV #TXLEN2!TXLEN1,WAX16
1490 030732 004737 004312 JSR PC,WRITAX ;SET TX CHAR LENGTH = 6
1491 030736 004737 006122 JSR PC,TXCHAR ;LOAD 7-BIT 000 CHAR, TX 5-BIT 000 CHAR
1492 030742 000000 000
1493 030744 000005 5
1494 030746 012737 000340 002376 MOV #TXLEN2!TXLEN1!TXLEN0,WAX16
1495 030754 004737 004312 JSR PC,WRITAX ;SET TX CHAR LENGTH = 7
1496 030760 004737 006122 JSR PC,TXCHAR ;LOAD 8-BIT 000 CHAR, TX 6-BIT 000 CHAR
1497 030764 000000 000
1498 030766 000006 6
1499 030770 012737 000000 002376 MOV #000,WAX16
1500 030776 004737 004312 JSR PC,WRITAX ;SET TX CHAR LENGTH = 8
1501 031002 004737 006122 JSR PC,TXCHAR ;LOAD EOM, TX 7-BIT 000 CHAR
1502 031006 001000 TXEOM
1503 031010 000007 7
1504 031012 004737 006122 JSR PC,TXCHAR ;LOAD EOM, TX 8-BIT 000 CHAR, CRC-CCITT-1 CHAR
1505 031016 001000 TXEOM
1506 031020 000031 25.
1507 031022 004737 006122 JSR PC,TXCHAR ;LOAD EOM, TX FIRST TERMINATING FLAG
1508 031026 001000 TXEOM
1509 031030 000010 8.
1510 031032 004737 006274 JSR PC,ENDTRN ;CLEAR TRANSMITTER
1511 031036 A9:
1512 031036 ENDTST
031036 L10060:
031036 TRAP CSETST
104401

```

1513
1514
1515
1516
1517

1518
1519
1520
1521
1522
1523
1524
1525
1526

1527 031040
031040

```

1528 031040 004737 005540 JSR PC,INITRN ;DO MASTER CLR, LOAD 2 SOM'S
1529 031044 000226 SYNCH
1530 031046 000011 STRIP!DDCMP
1531 031050 012701 003224 MOV #MSG1+4,R1 ;GET POINTER TO DATA
1532 031054 010103 MOV R1,R3
1533 031056 012137 002422 MOV (R1)+,TXWORD
1534 031062 004737 005174 JSR PC,LDTXSI ;LOAD A DATA CHAR INTO TX SILO

```

```

*****
:SBTTL TEST 29 - TXDATA BIT TEST - CHAR MODE, CRC
:
: * THE LINE UNIT IS INITIALIZED AND A MSG IS INITIATED (USING STEPLU) WITH CRC-
: * 16 SELECTED IN CHAR MODE. TWO SYNCHS, 000,125,252,377,000, AND 2 TERMINATING
: * SYNCHS ARE THEN SENT. THE PROGRAM CHECKS EACH BIT OF THE TRANSMITTED
: * DATA CHARS, BY MONITORING TXDATA (REG 17) AS THE DATA IS CLOCKED OUT OF
: * THE USYRT TRANSMITTER.
:*****
BGNTST

```

T29::

```

1535 031066 020127 003242      CMP      R1,#MSG1+18.      ;SEE IF ALL CHARS LOADED YET
1536 031072 103771              BLO      3$              ;BR IF NOT YET
1537 031074 004737 005146      JSR      PC,WAIT50      ;WAIT FOR SILO TO RIPPLE
1538 031100 004737 005254      JSR      PC,STPLU      ;CLOCK LU UNTIL SYNCHS ARE TX'D
1539 031104 100020              CHPCHK!16.
1540 031106 011337 031116      6$:      MOV      (R3),8$      ;GET EXPECTED DATA CHAR
1541 031112 004737 011176      JSR      PC,CKTBIT      ;CHECK TXDATA FOR CHAR BITS
1542 031116 000000              8$:      .WORD      0      ;EXPECTED CHAR GOES HERE
1543 031120 062703 000002      ADD      #2,R3      ;INCR PATTERN POINTER
1544 031124 020327 003236      CMP      R3,#MSG1+14.      ;SEE IF ALL CHARS CHECKED YET
1545 031130 103766              BLO      6$              ;BR IF NOT YET
1546 031132 004737 003576      15$:     JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
1547 031136              ENDTST
      031136
      031136 104401
      L'0061:
      TRAP      CSETST

1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563 031140
      031140
1564 031140 012737 031502 002362      MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
1565 031146 004737 003576              JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
1566 031152 004737 010640              JSR      PC,SETUP      ;PROGRAM THE USYRT
1567 031156 000226              SYNCH
1568 031160 000013              STRIP! IERR!DDCMP
1569 031162 000000              000
1570 031164 000000              000
1571 031166 012737 000012 002400      MOV      #12,REGNUM      ;SET LU REG NO. = 12
1572 031174 005037 002402              CLR      AXNUM      ;SET AX BYTE NO. = 0 FOR AX0
1573 031200 112737 000040 002366      MOVB     #LULP,WRIBYT
1574 031206 004737 003750              JSR      PC,WRITLU      ;SET LULP IN REG 12
1575 031212 012701 003220              MOV      #MSG1,R1      ;GET POINTER TO MSG DATA TABLE
1576 031216 012137 002422      3$:      MOV      (R1)+,TXWORD      ;GET CHAR TO BE LOADED
1577 031222 004737 005174              JSR      PC,LDTXSI      ;LOAD CHAR INTO TX SILO
1578 031226 020127 003246      CMP      R1,#MSG1+22.      ;SEE IF ALL MSG CHARS LOADED YET
1579 031232 103771              BLO      3$              ;BR IF NOT YET
1580 031234 004737 005146      JSR      PC,WAIT50      ;ALLOW DATA TO RIPPLE IN SILO
1581 031240 004737 005254      JSR      PC,STPLU      ;CLOCK LU FOR 40 CYCLES (UNTIL FIRST
1582 031244 000050              40.      ; DATA CHAR IS ABOUT TO BE RECEIVED)
1583 031246 004737 006714      JSR      PC,IACTIV      ;CHK IACT = 0
1584 031252 000000              0
1585 031254 004737 005254      JSR      PC,STPLU      ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
1586 031260 000006              6
1587 031262 012701 003224      10$:     MOV      #MSG1+4,R1
1588 031266 004737 006714              JSR      PC,IACTIV      ;CHK IACT = 1

```

```

*****
.SBTTL      TEST 30 - USYRT RECEIVER MSG TEST - CHAR MODE, CRC
*
* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-16
* SELECTED. TWO SYNCHS, 000,125,252,377,000, AND FOUR TERMINATING SYNCHS ARE
* SENT. THE PROGRAM MONITORS IACT, AND THE RCV'D CHARS AND CRC BYTES ARE READ
* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
* STILL SET AFTER THE MESSAGE.
*****
BGNTST

```

T30::

```

1589 031272 000001          1
1590 031274 004737 004124   JSR    PC,READAX      ;READ AX0
1591 031300 023721 002370   CMP    RAX15,(R1)+    ;COMPARE RCV'D CHAR TO EXPECTED
1592 031304 001415          BEQ    12$              ;BR IF RCV'D DATA OK
1593 031306 016137 177776 002404  MOV    -2(R1),GOODAT    ;GET EXPECTED DATA
1594 031314 013737 002370 002406  MOV    RAX15,BADDAT    ;GET ACTUAL DATA
1595 031322 004737 004526          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
1596          ;REPORT INCORRECT DATA CHAR RCV'D
1597          ERRDF 26,EM26,ERR3
                                TRAP    C$ERDF
                                .WORD 26
                                .WORD EM26
                                .WORD ERR3
1598 031326 104455
1599 031330 000032
1600 031332 013067
1601 031334 015140
1598 031336 000461
1599 031340 004737 005254   12$:  BR    24$
1600 031344 000010          JSR    PC,STPLU      ;CLOCK LU 8 CYCLES
1601 031346 020127 003236   8.    JSR
1602 031352 103745          CMP    R1,#MSG1+14.    ;SEE IF CHECKING HI CRC BYTE YET
1603 031354 004737 006714   BLO    10$              ;BR IF NOT YET
1604 031360 000001          JSR    PC,IACTIV    ;CHK IACT = 1
1605 031362 004737 004124   1      JSR    PC,READAX      ;READ AX0
1606 031366 123727 002370 000160  CMPB   RAX15,#160      ;CMP RCV'D CHAR TO EXPECTED HI CRC BYTE
1607 031374 001415          BEQ    16$              ;BR IF HI CRC BYTE RCV'D OK
1608 031376 012737 000160 002404  MOV    #160,GOODAT    ;GET EXPECTED DATA
1609 031404 013737 002370 002406  MOV    RAX15,BADDAT    ;SET ACTUAL DATA
1610 031412 004737 004526          JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
1611          ;REPORT INCORRECT CRC BYTE RCV'D
1612          ERRDF 27,EM27,ERR3
                                TRAP    C$ERDF
                                .WORD 27
                                .WORD EM27
                                .WORD ERR3
1613 031416 104455
1614 031420 000033
1615 031422 013121
1616 031424 015140
1613 031426 000425
1614 031430 004737 005254   16$:  BR    24$
1615 031434 000010          JSR    PC,STPLU      ;CLOCK LU FOR 8 CYCLES
1616 031436 004737 006714   8.    JSR
1617 031442 000001          JSR    PC,IACTIV    ;CHK IACT = 1
1618 031444 012737 000034 002404  1      MOV    #034,GOODAT    ;GET EXPECTED LO CRC BYTE
1619 031452 004737 004124          JSR    PC,READAX      ;READ AX0
1620 031456 123727 002370 000034  CMPB   RAX15,#034      ;CMP RCV'D CHAR TO EXPECTED LO CRC BYTE
1621 031464 001347          BNE    14$              ;BR IF LO CRC INCORRECT
1622 031466 004737 005254   JSR    PC,STPLU      ;CLOCK LU 8 CYCLES
1623 031472 000010   8.    JSR
1624 031474 004737 006714   JSR    PC,IACTIV    ;CHK IACT STILL = 1
1625 031500 000001   1      JSR
1626 031502 004737 003576   24$:  JSR    PC,MSTCLR      ;ISSUE CLEAN-UP MASTER CLEAR
1627 031506          ENDTST
                                L10062:
                                TRAP    C$ETST
1628 031506 104401
1629
1630
1631
1632
1633          ;*****
1634          ;SBTTL      TEST 31 - USYRT RECEIVER MSG TEST - BIT MODE, CRC
1635          ;*

```

```

1636      ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
1637      ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH CRC-
1638      ;* CCITT-1. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS ARE THEN
1639      ;* SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE READ
1640      ;* FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
1641      ;* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
1642      ;*****
1643      BGNTST
1644      031510      012737      032044      002362      MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
1645      031510      004737      003576      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
1646      031522      004737      010640      JSR      PC,SETUP      ;PROGRAM THE USYRT
1647      031526      000000      000      000
1648      031530      000002      000      IERR
1649      031532      000000      000
1650      031534      000000      000
1651      031536      012737      000012      002400      MOV      #12,REGNUM      ;SET LU REG NO. = 12
1652      031544      005037      002402      CLR      AXNUM      ;SET AX BYTE NO. = 0 FOR AX0
1653      031550      112737      000040      002366      MOVB     #LULP,WRIBYT
1654      031556      004737      003750      JSR      PC,WRITLU      ;SET LULP IN REG 12
1655      031562      012701      003220      MOV      #MSG1,R1      ;GET POINTER TO MSG DATA TABLE
1656      031566      012137      002422      3$:      MOV      (R1)+,TXWORD      ;GET CHAR TO BE LOADED
1657      031572      004737      005174      JSR      PC,LDTXSI      ;LOAD CHAR INTO TX SILO
1658      031576      020127      003242      CMP      R1,#MSG1+18.      ;SEE IF ALL MSG CHARS LOADED YET
1659      031602      103771      000      BLO      3$      ;BR IF NOT YET
1660      031604      004737      005146      JSR      PC,WAIT50      ;ALLOW DATA TO RIPPLE IN SILO
1661      031610      004737      005254      JSR      PC,STPLU      ;CLOCK LU FOR 50 CYCLES (UNTIL FIRST
1662      031614      000062      000      50.      ; DATA CHAR IS ABOUT TO BE RECEIVED)
1663      031616      004737      006714      JSR      PC,IACTIV      ;CHK IACT = 0
1664      031622      000000      000
1665      031624      004737      007102      JSR      PC,RSEOM      ;CHK RSOM = 0, REOM = 0
1666      031630      000000      000
1667      031632      004737      005254      JSR      PC,STPLU      ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
1668      031636      000006      000
1669      031640      012701      003224      MOV      #MSG1+4,R1
1670      031644      020127      003224      5$:      CMP      R1,#MSG1+4      ;SEE IF 1ST CHAR RCV'D
1671      031650      001007      000      BNE      6$      ;BR IF NO
1672      031652      004737      006714      JSR      PC,IACTIV      ;CHK IACT = 1
1673      031656      000001      000
1674      031660      004737      007102      JSR      PC,RSEOM      ;CHK RSOM = 1, REOM = 0
1675      031664      000001      000
1676      031666      000420      000      BR      9$
1677      031670      020127      003234      6$:      CMP      R1,#MSG1+12.      ;SEE IF LAST CHAR RCV'D
1678      031674      001007      000      BNE      8$      ;BR IF NO
1679      031676      004737      006714      JSR      PC,IACTIV      ;CHK FOR IACT = 0
1680      031702      000000      000
1681      031704      004737      007102      JSR      PC,RSEOM      ;CHK RSOM = 0, REOM = 0
1682      031710      000000      000
1683      031712      000406      000      BR      9$
1684      031714      004737      006714      8$:      JSR      PC,IACTIV      ;CHK FOR IACT = 1
1685      031720      000001      000
1686      031722      004737      007102      JSR      PC,RSEOM      ;CHK RSOM = 0, REOM = 0
1687      031726      000000      000
1688      031730      004737      004124      9$:      JSR      PC,READAX      ;READ AX0
1689      031734      023721      002370      CMP      RAX15,(R1)+      ;COMPARE RCV'D CHAR TO EXPECTED
1690      031740      001415      000      BEQ      12$      ;BR IF RCV'D DATA OK
1691      031742      016137      177776      002404      MOV      -2(R1),GOODAT      ;GET EXPECTED DATA

```

```

1692 031750 013737 002370 002406      MOV    RAX15,BALDAT ;GET ACTUAL DATA
1693 031756 004737 004526      JSR     PC,GETALL  ;GET REGS FOR PRINTOUT
1694                                     ;REPORT INCORRECT DATA CHAR RCV'D
1695                                     ERRDF  26,EM26,ERR3
1695 031762 104455                                     TRAP    C$ERDF
1695 031762 000032                                     .WORD  26
1695 031764 013067                                     .WORD  EM26
1695 031766 015140                                     .WORD  ERR3
1695 031770 000424
1696 031772 004737 005254      12$:  BR      24$
1697 031774 000010      JSR     PC,STPLU ;CLOCK LU 8 CYCLES
1698 032000 000010      8.
1699 032002 020127 003236      CMP     R1,#MSG1+14. ;SEE IF ALL DATA CHARS CHECKED YET
1700 032006 103716      BLO     5$ ;BR IF NOT YET
1701 032010 004737 006714      JSR     PC,IACTIV ;CHK IACT = 0
1702 032014 000000      0
1703 032016 004737 007102      JSR     PC,RSEOM ;CHK RSOM = 0, REOM = 0
1704 032022 000000      0
1705 032024 012737 000200 002366      MOV     #IC,WRIBYT
1706 032032 004737 003750      JSR     PC,WRITLU ;SET IC (INPUT CLEAR) IN REG 12
1707 032036 004737 006714      JSR     PC,IACTIV ;CHK IACT = 0
1708 032042 000000      0
1709 032044 004737 003576      24$:  JSR     PC,MSTCLR ;ISSUE CLEAN-UP MASTER CLEAR
1710 032050      ENDTST
1710 032050      L10063:
1710 032050 104401      TRAP    C$ETST
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726 032052      ;*****
1726 032052      ;SBTTL TEST 32 - USYRT RECEIVER MSG TEST - CHAR MODE, NO CRC
1727 032052 012737 032314 002362      ;*
1728 032060 004737 003576      ;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
1729 032064 004737 010640      ;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH NO
1730 032070 000226      ;* ERROR DETECTION. TWO SYNCHS, 000,125,252,377,000, AND TWO SYNCHS ARE
1731 032072 000313      ;* THEN SENT. THE PROGRAM MONITORS IACT, AND THE RECEIVED CHARS ARE READ FROM
1732 032074 000000      ;* AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR IACT
1733 032076 000000      ;* STILL = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT = 0.
1734 032100 012737 000012 002400      ;*****
1735 032106 005037 002402      BGNTST
1736 032112 112737 000040 002366      T32::
1737 032120 004737 003750      MOV     #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1738 032124 012701 003220      JSR     PC,MSTCLR ;ISSUE MASTER CLEAR
1739 032130 012137 002422      JSR     PC,SETUP ;PROGRAM THE USYRT
1740 032134 004737 005174      SYNCH
1741 032140 020127 003242      CRC2!CRC1!STRIP!IERR!DDCMP
1741 032140 020127 003242      000
1741 032140 020127 003242      000
1741 032140 020127 003242      MOV     #12,REGNUM ;SET LU REG NO. = 12
1741 032140 020127 003242      CLR     AXNUM ;SET AX BYTE NO. = 0 FOR AX0
1741 032140 020127 003242      MOV     #LULP,WRIBYT
1741 032140 020127 003242      JSR     PC,WRITLU ;SET LULP IN REG 12
1741 032140 020127 003242      MOV     #MSG1,R1 ;GET POINTER TO MSG DATA TABLE
1741 032140 020127 003242      3$:  MOV     (R1)+,TXWORD ;GET CHAR TO BE LOADED
1741 032140 020127 003242      JSR     PC,LDTXSI ;LOAD CHAR INTO TX SILO
1741 032140 020127 003242      CMP     R1,#MSG1+18. ;SEE IF ALL MSG CHARS LOADED YET

```

```

1742 032144 103771          BLO      3$          ;BR IF NOT YET
1743 032146 004737 005146    JSR      PC,WAIT50    ;ALLOW DATA TO RIPPLE IN SILO
1744 032152 004737 005254    JSR      PC,STPLU     ;CLOCK LU FOR 24 CYCLES (UNTIL FIRST
1745 032156 000030          24.          ; DATA CHAR IS ABOUT TO BE RECEIVED)
1746 032160 004737 006714    JSR      PC,IACTIV    ;CHK IACT = 0
1747 032164 000000          0            ;
1748 032166 004737 005254    JSR      PC,STPLU     ;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
1749 032172 000006          6            ;
1750 032174 012701 003224    MOV      #MSG1+4,R1
1751 032200 004737 006714    10$: JSR      PC,IACTIV    ;CHK IACT = 1
1752 032204 000001          1            ;
1753 032206 004737 004124    JSR      PC,READAX    ;READ AX0
1754 032212 023721 002370    CMP      RAX15,(R1)+  ;COMPARE RCV'D CHAR TO EXPECTED
1755 032216 001415          BEQ      12$          ;BR IF RCV'D DATA OK
1756 032220 016137 177776 002404    MOV      -2(R1),GOODAT ;GET EXPECTED DATA
1757 032226 013737 002370 002406    MOV      RAX15,BADDAT ;GET ACTUAL DATA
1758 032234 004737 004526    JSR      PC,GETALL   ;GET REGS FOR PRINTOUT
1759          ;REPORT INCORRECT DATA CHAR RCV'D
1760 032240          ERRDF 26,EM26,ERR3
1761 032240 104455          TRAP      C$ERDF
1762 032242 000032          .WORD    26
1763 032244 013067          .WORD    EM26
1764 032246 015140          .WORD    ERR3
1765 032250 000421          BR      24$
1766 032252 004737 005254    12$: JSR      PC,STPLU     ;CLOCK LU 8 CYCLES
1767 032256 000010          8.
1768 032260 020127 003236    CMP      R1,#MSG1+14. ;SEE IF ALL 5 DATA CHARS RCV'D YET
1769 032264 103745          BLO      10$          ;BR IF NOT YET
1770 032266 004737 006714    JSR      PC,IACTIV    ;CHK FOR IACT STILL = 1
1771 032272 000001          1
1772 032274 012737 000200 002366    MOV      #IC,WRIBYT
1773 032302 004737 003750    JSR      PC,WRITLU     ;SET IC (INPUT CLEAR) IN REG 12
1774 032306 004737 006714    JSR      PC,IACTIV    ;CHK IACT = 0
1775 032312 000000          0
1776 032314 004737 003576    24$: JSR      PC,MSTCLR   ;ISSUE CLEAN-UP MASTER CLEAR
1777 032320          ENDTST
1778 032320          L10064:
1779 032320 104401          TRAP      C$ETST
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789 032322          ;*****
1790 032322          ;SBTTL      TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC
1791 032330 012737 032656 002362    MOV      #24$,RETADR
1792 032330 004737 003576    JSR      PC,MSTCLR   ;ISSUE MASTER CLEAR

```

```

;*****
;SBTTL      TEST 33 - USYRT RECEIVER MSG TEST - BIT MODE, NO CRC
;
;* THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU) WITH
;* LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, AND WITH EPROR
;* DETECTION INHIBITED. TWO FLAGS, 000,125,252,377,000, AND TWO TERMINATING FLAGS
;* ARE THEN SENT. THE PROGRAM MONITORS IACT, RSOM, AND THE RCV'D CHARS ARE
;* READ FROM AX0-15 AND COMPARED TO EXPECTED VALUES. THE PROGRAM THEN CHECKS FOR
;* IACT = 0, SETS IC TO CLEAR THE RECEIVER, AND CHECKS FOR IACT STILL = 0.
;*****
;BGTST

```

T33::

1792	032334	004737	010640		JSR	PC,SETUP	;PROGRAM THE USYRT
1793	032340	000000			000		
1794	032342	000302			CRC2!CRC1!IERR		
1795	032344	000000			000		
1796	032346	000000			000		
1797	032350	012737	000012	002400	MOV	#12,REGNUM	;SET LU REG NO. = 12
1798	032356	005037	002402		CLR	AXNUM	;SET AX BYTE NO. = 0 FOR AX0
1799	032362	112737	000040	002366	MOVB	#LULP,WRIBYT	
1800	032370	004737	003750		JSR	PC,WRITLU	;SET LULP IN REG 12
1801	032374	012701	003220		MOV	#MSG1,R1	;GET POINTER TO MSG DATA TABLE
1802	032400	012137	002422	3\$:	MOV	(R1)+,TXWORD	;GET CHAR TO BE LOADED
1803	032404	004737	005174		JSR	PC,LDIXSI	;LOAD CHAR INTO TX SILO
1804	032410	020127	003242		CMP	R1,#MSG1+18.	;SEE IF ALL MSG CHARS LOADED YET
1805	032414	103771			BLO	3\$	;BR IF NOT YET
1806	032416	004737	005146		JSR	PC,WAIT50	;ALLOW DATA TO RIPPLE IN SILO
1807	032422	004737	005254		JSR	PC,STPLU	;CLOCK LU FOR 33 CYCLES (UNTIL FIRST
1808	032426	000041			33.		; DATA CHAR IS ABOUT TO BE RECEIVED)
1809	032430	004737	006714		JSR	PC,IACIV	;CHK IACT = 0
1810	032434	000000			0		
1811	032436	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 0, REOM = 0
1812	032442	000000			0		
1813	032444	004737	005254		JSR	PC,STPLU	;CLOCK LU UNTIL 1ST DATA CHAR IS RCV'D
1814	032450	000006			6		
1815	032452	012701	003224	5\$:	MOV	#MSG1+4,R1	
1816	032456	020127	003224		CMP	R1,#MSG1+4	;SEE IF 1ST CHAR RCV'D
1817	032462	001007			BNE	6\$	;BR IF NO
1818	032464	004737	006714		JSR	PC,IACIV	;CHK IACT = 1
1819	032470	000001			1		
1820	032472	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 1, REOM = 0
1821	032476	000001			1		
1822	032500	000420			BR	9\$	
1823	032502	020127	003234	6\$:	CMP	R1,#MSG1+12.	;SEE IF LAST CHAR RCV'D
1824	032506	001007			BNE	8\$	;BR IF NO
1825	032510	004737	006714		JSR	PC,IACIV	;CHK FOR IACT = 0
1826	032514	000000			0		
1827	032516	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 0, REOM = 0
1828	032522	000000			0		
1829	032524	000406			BR	9\$	
1830	032526	004737	006714	8\$:	JSR	PC,IACIV	;CHK FOR IACT = 1
1831	032532	000001			1		
1832	032534	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 0, REOM = 0
1833	032540	000000			0		
1834	032542	004737	004124	9\$:	JSR	PC,READAX	;READ AX0
1835	032546	023721	002370		CMP	RAX15,(R1)+	;COMPARE RCV'D CHAR TO EXPECTED
1836	032552	001415			BEQ	12\$	;BR IF RCV'D DATA OK
1837	032554	016137	177776	002404	MOV	-2(R1),GOODAT	;GET EXPECTED DATA
1838	032562	013737	002370	002406	MOV	RAX15,BADAT	;GET ACTUAL DATA
1839	032570	004737	004526		JSR	PC,GETALL	;GET REGS FOR PRINTOUT
1840							
1841	032574				;REPORT	INCORRECT DATA CHAR RCV'D	
	032574	104455			ERRDF	26,EM26,ERR3	
	032576	000032					
	032600	013067					
	032602	015140					
1842	032604	000424			BR	24\$	
1843	032606	004737	005254	12\$:	JSR	PC,STPLU	;CLOCK LU 8 CYCLES
1844	032612	000010			8.		

TRAP	C\$ERDF
.WORD	26
.WORD	EM26
.WORD	ERR3

```

1845 032614 020127 003236      CMP      R1,#MSG1+14.      ;SEE IF ALL DATA CHARS CHECKED YET
1846 032620 103716      BLO      5$              ;BR IF NOT YET
1847 032622 004737 006714      JSR      PC,IACTIV      ;CHK IACT = 0
1848 032626 000000      0
1849 032630 004737 007102      JSR      PC,RSEOM      ;CHK RSOM = 0, REOM = 0
1850 032634 000000      0
1851 032636 012737 000200 002366      MOV      #IC,WRIBYT
1852 032644 004737 003750      JSR      PC,WRITLU      ;SET IC (INPUT CLEAR) IN REG 12
1853 032650 004737 006714      JSR      PC,IACTIV      ;CHK IACT = 0
1854 032654 000000      0
1855 032656 004737 003576      JSR      PC,MSTCLR      ;ISSUE CLEAN-UP MASTER CLEAR
1856 032662      24$:      ENDTST
      032662      L10065:
      032662 104401      TRAP      C$ETST

```

1857

1858

1859

1860

1861

1862

1863

1864

1865

1866

1867

1868

1869 032664

032664

1870 032664 004737 003576

1871 032670 012737 000014 002400

1872 032676 012737 000040 002366

1873 032704 004737 003750

1874 032710 012737 000040 002426

1875 032716 012737 000125 002422

1876 032724 004737 005174

1877 032730 012737 000002 002402

1878 032736 004737 004124

1879 032742 123727 002370 000000

1880 032750 001414

1881 032752 012737 000000 002404

1882 032760 013737 002370 002406

1883 032766 004737 004526

1884

1885 032772

032772 104455

032774 000003

032776 012220

033000 015140

1886 033002 005037 002426

1887 033006 004737 003576

1888 033012

033012

033012 104401

1889

1890

1891

1892

```

*****
;SBTTL      TEST 34 - SILO-DISABLED TRANSMITTER LOAD TEST
;
;* THIS TEST DISABLES THE SILOS, LOADS A 125 CHARACTER INTO THE TX SILO, AND
;* READS AX1-15 AND CHECKS THAT THE DATA DID NOT GET LOADED INTO THE USYRT TX
;* BUFFER.
*****
BGNTST

```

T34::

```

      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
      MOV      #14,REGNUM      ;SET REG NO. = 14
      MOV      #DISSI,WRIBYT      ;SET DISSI BIT
      JSR      PC,WRITLU
      MOV      #DISSI,DISILO      ;SET DISABLE SILO FLAG
      MOV      #125,TXWORD      ;LOAD 125 INTO TX SILO
      JSR      PC,LDTXSI
      MOV      #2,AXNUM      ;SET REG NO. FOR AX1
      JSR      PC,READAX      ;READ AX1-15, AX1-16
      CMPB     RAX15,#000      ;CHECK FOR AX1-15 UNCHANGED
      BEQ      3$              ;BR IF UNCHANGED
      MOV      #000,GOODAT      ;GET EXPECTED DATA
      MOV      RAX15,BADDAT      ;GET ACTUAL DATA
      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
;REPORT REG MISCOMPARE
ERRDF 3,EM3,ERR3

```

```

TRAP      C$ERDF
.WORD     3
.WORD     EM3
.WORD     EPR3

```

```

3$:      CLR      DISILO      ;CLEAR DISABLE SILO FLAG
      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
ENDTST

```

```

L10066:      TRAP      C$ETST

```

1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904

1905 033014

033014

1906 033014 012737 033714 002362

1907 033022 004737 005540

1908 033026 000000

1909 033030 000000

1910 033032 004737 003576

1911 033036 004737 010640

1912 033042 000000

1913 033044 000302

1914 033046 000000

1915 033050 000000

1916 033052 012737 000014 002400

1917 033060 012737 000140 002366

1918 033066 004737 003750

1919 033072 012737 000140 002426

1920 033100 012737 000012 002400

1921 033106 112737 000040 002366

1922 033114 004737 003750

1923 033120 012701 003220

1924 033124 004737 004662

1925 033130 000001

1926 033132 012737 000002 002402

1927 033140 112137 002374

1928 033144 112137 002376

1929 033150 004737 004312

1930 033154 004737 004662

1931 033160 000000

1932 033162 004737 005352

1933 033166 000000

1934 033170 004737 005254

1935 033174 000003

1936 033176 004737 005352

1937 033202 000001

1938 033204 012703 000004

1939 033210 112137 002374

1940 033214 112137 002376

1941 033220 020327 000001

1942 033224 001006

1943 033226 005737 002430

1944 033232 001403

1945 033234 112737 000002 002376

1946 033242 004737 004662

1947 033246 000001

1948 033250 004737 004312

```

*****
:SBTTL      TEST 35 - SILO-DISABLED MESSAGE TEST - BIT MODE, NO CRC
:
: * THE LINE UNIT IS INITIALIZED AND A MESSAGE IS INITIATED (USING STEPLU)
: * WITH LULP (REG 12) SET TO LOOP THE DATA INTERNALLY IN THE USYRT, WITH SILO
: * DISABLE SET, AND WITH NO ERROR DETECTION. TWO FLAGS, 000,125,252, AND
: * TERMINATING FLAGS ARE THEN SENT BY LOADING THE TRANSMITTED CHARS INTO
: * REG AX1. THE PROGRAM MONITORS OACT, IACT, RSOM, REOM, ORDY, OCOR, ICIR,
: * IRDY, AND THE RECEIVED CHARS ARE READ FROM AX0 AND COMPARED TO EXPECTED
: * VALUES.
:*****
RGNTST

```

```

T35::
MOV      #18$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
JSR      PC,INITRN        ;FIND OUT WHICH USYRT CHIP
0
0
JSR      PC,MSTCLR        ;ISSUE MASTER CLEAR
JSR      PC,SETUP         ;PROGRAM THE USYRT
000
CRC2!CRC1!IERR
000
000
MOV      #14,REGNUM       ;SET REG NO. = 14
MOV      #TXEN!DISSI,WRIBYT
JSR      PC,WRITLU        ;SET TXEN AND DISSI IN REG 14
MOV      #TXEN!DISSI,DISILO ;SET DISABLE SILO FLAG
MOV      #12,REGNUM       ;SET LU REG NO. = 12
MOVB     #LULP,WRIBYT
JSR      PC,WRITLU        ;SET LULP IN REG 12
MOV      #MSG1,R1         ;GET POINTER TO MSG
JSR      PC,OSIRDY        ;CHK ORDY = 1
1
MOV      #2,AXNUM         ;SET AX BYTE NO. FOR AX1
MOVB     (R1)+,WAX15      ;GET A CHAR
MOVB     (R1)+,WAX16
JSR      PC,WRITAX        ;LOAD CHAR INTO USYRT TX BUFFER
JSR      PC,OSIRDY        ;CHK ORDY = 0
0
JSR      PC,OACTIV        ;CHK OACT = 0
0
JSR      PC,STPLU         ;CLOCK LU FOR 3 CYCLES
3
JSR      PC,OACTIV        ;CHK OACT = 1
1
MOV      #4,R3            ;INIT COUNTER
MOVB     (R1)+,WAX15      ;GET ANOTHER CHAR
MOVB     (R1)+,WAX16
CMP      R3,#1            ;SEE IF LOADING LAST DATA CHAR YET
BNE      5$              ;BR IF NOT
TST      CHPTYP           ;SEE IF SIG USYRT
BEQ      5$              ;BR IF YES
MOVB     #TEOM,WAX16      ;SET TEOM WITH LAST DATA CHAR
JSR      PC,OSIRDY        ;CHK ORDY = 1
1
JSR      PC,WRITAX        ;LOAD ANOTHER CHAR INTO USYRT TX BUFFER

```

4\$:

5\$:

1949	033254	004737	004662		JSR	PC,OSIRDY	;CHK ORDY = 0	
1950	033260	000000			0			
1951	033262	004737	006714		JSR	PC,IACTIV	;CHK IACT = 0	
1952	033266	000000			0			
1953	033270	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 0, REOM = 0	
1954	033274	000000			0			
1955	033276	004737	005254		JSR	PC,STPLU	;CLOCK LU FOR 8 CYCLES	
1956	033302	000010			8.			
1957	033304	004737	005352		JSR	PC,OACTIV	;CHK OACT = 1	
1958	033310	000001			1			
1959	033312	004737	004662		JSR	PC,OSIRDY	;CHK ORDY = 1	
1960	033316	000001			1			
1961	033320	005303			DEC	R3	;DECR COUNTER	
1962	033322	001332			BNE	4\$	;BR IF NOT DONE YET	
1963	033324	004737	006430		JSR	PC,ISIRDY	;CHK IRDY = 0	
1964	033330	000000			0			
1965	033332	005737	002430		TST	CHPTYP	;SEE IF SIG USYRT	
1966	033336	001007			BNE	11\$	;BR IF NOT	
1967	033340	105037	002374		CLRB	WAX15		
1968	033344	112737	000002	002376	MOVB	#TEOM,WAX16	;LOAD EOM CHAR	
1969	033352	004737	004312		JSR	PC,WRITAX	;LOAD ANOTHER CHAR INTO USYRT TX BUFFER	
1970	033356	004737	005254		JSR	PC,STPLU	;CLOCK LU FOR 3 CYCLES	
1971	033362	000003			3			
1972	033364	004737	006430		JSR	PC,ISIRDY	;CHK IRDY = 1	
1973	033370	000002			2			
1974	033372	004737	005352		JSR	PC,OACTIV	;CHK OACT = 1	
1975	033376	000001			1			
1976	033400	004737	006714		JSR	PC,IACTIV	;CHK IACT = 1	
1977	033404	000001			1			
1978	033406	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 1, REOM = 0	
1979	033412	000001			1			
1980	033414	004737	006430		JSR	PC,ISIRDY	;CHK IRDY = 0	
1981	033420	000000			0			
1982	033422	012737	000000	002402	MOV	#0,AXNUM	;SET AX BYTE NO. FOR AX0	
1983	033430	123727	002370	000000	CMPB	RAX15,#000	;COMPARE RCV'D CHAR TO 000	
1984	033436	001415			BEQ	9\$	;BR IF MATCH	
1985	033440	012737	000000	002404	MOV	#0,GOODAT	;SET EXPECTED DATA	
1986	033446	013737	002370	002406	MOV	RAX15,BADDAT	;SET ACTUAL DATA	
1987	033454	004737	004526		JSR	PC,GETALL	;GET REGS FOR PRINTOUT	
1988								
1989	033460				;REPORT	INCORRECT DATA CHAR RCV'D		
	033460	104455			ERRDF	26,EM26,ERR3		
	033462	000032						
	033464	013067						
	033466	015140						
1990	033470	000511			BR	18\$		
1991	033472	004737	005254		JSR	PC,STPLU	;CLOCK LU FOR 8 CYCLES	
1992	033476	000010			8.			
1993	033500	004737	006430		JSR	PC,ISIRDY	;CHK IRDY - 1	
1994	033504	000002			2			
1995	033506	004737	005352		JSR	PC,OACTIV	;CHK OACT = 1	
1996	033512	000001			1			
1997	033514	004737	006714		JSR	PC,IACTIV	;CHK IACT - 1	
1998	033520	000001			1			
1999	033522	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 0, REOM 0	
2000	033526	000000			0			
2001	033530	004737	006430		JSR	PC,ISIRDY	;CHK IRDY - 0	

TRAP CSERDF
 .WORD 26
 .WORD EM26
 .WORD ERR3

2002	033534	000000			0		
2003	033536	123727	002370	000125	CMPB	RAX15,#125	;COMPARE 2ND RCV'D CHAR TO 125
2004	033544	001404			BEQ	12\$	;BR IF MATCH
2005	033546	012737	000125	002404	MOV	#125,GOODAT	;SET EXPECTED DATA
2006	033554	000734			BR	6\$	;BR TO REPORT ERROR
2007	033556	004737	005254		JSR	PC,STPLU	;CLOCK LU FOR 8 CYCLES
2008	033562	000010			8.		
2009	033564	004737	006430		JSR	PC,ISIRDY	;CHK IRDY = 1
2010	033570	000002			2		
2011	033572	004737	005352		JSR	PC,OACTIV	;CHK OACT = 1
2012	033576	000001			1		
2013	033600	004737	006714		JSR	PC,IActiv	;CHK IACT = 0
2014	033604	000000			0		
2015	033606	004737	007102		JSR	PC,RSEOM	;CHK RSOM = 0, REOM = 1
2016	033612	000002			2		
2017	033614	004737	006430		JSR	PC,ISIRDY	;CHK IRDY = 0
2018	033620	000000			0		
2019	033622	123727	002370	000252	CMPB	RAX15,#252	;COMPARE 3RD RCV'D CHAR TO 252
2020	033630	001404			BEQ	14\$	;BR IF MATCH
2021	033632	012737	000252	002404	MOV	#252,GOODAT	;SET EXPECTED DATA
2022	033640	000702			BR	6\$	;BR TO REPORT ERROR
2023	033642	012737	000014	002400	14\$: MOV	#14,REGNUM	;SET REG NO. = 14
2024	033650	012737	000040	002366	MOV	#DISSI,WRIBYT	
2025	033656	004737	003750		JSR	PC,WRITLU	;CLEAR TX ENABLE
2026	033662	012737	000011	002400	MOV	#11,REGNUM	;SET REG NO. = 11
2027	033670	012737	000200	002366	MOV	#OC,WRIBYT	
2028	033676	004737	003750		JSR	PC,WRITLU	;SET OC TO SHUT DOWN TRANSMITTER
2029	033702	004737	005146		JSR	PC,WAIT50	;WAIT FOR SHUTDOWN
2030	033706	004737	005352		JSR	PC,OACTIV	;CHK FOR OACT = 0
2031	033712	000000			0		
2032	033714	005037	002426		18\$: CLR	DISILO	;CLEAR DISABLE SILO FLAG
2033	033720	004737	003576		JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
2034	033724				ENDTST		
	033724						L10067:
	033724	104401					TRAP C\$ETST

2035
 2036
 2037
 2038
 2039
 2040
 2041
 2042
 2043
 2044
 2045
 2046
 2047
 2048
 2049
 2050
 2051
 2052
 2053
 2054
 2055
 2056

```

*****
.SBTTL      TEST 36 - RECEIVER BUFFER TEST - CHAR MODE, CRC
;
; * FIRST, A MASTER CLEAR IS DONE AND THE PROGRAM CHECKS FOR ICIR = 1 AND IRDY
; * = 0. THEN, 2 SOM CHARS ARE LOADED AND CLOCKED INTO THE USYRT, AND 64
; * BYTES OF A 256-BYTE BINARY COUNT DATA PATTERN (000-377) ARE LOADED INTO
; * THE TX SILO.
; * THE LINE UNIT IS THEN CLOCKED UNTIL IRDY = 1, AND THE PROGRAM CHECKS FOR
; * THIS TO OCCUR WITHIN 40-43 CYCLES. THE PROGRAM READS THE RCV SILO, CHECKS THE
; * CHAR FOR 000, AND CHECKS FOR IRDY = 0 AGAIN.
; * THE LINE UNIT IS THEN CLOCKED IN GROUPS OF 8 CYCLES, AND AFTER EACH, THE
; * PROGRAM CHECKS FOR ICIR = 1, IRDY = 1, UNTIL THE 64TH GROUP, AFTER WHICH
; * IT CHECKS FOR ICIR = 0, IRDY = 1. THE SECOND DATA CHAR IS READ FROM THE
; * RECEIVER SILO AND COMPARED TO 001. THEN, THE PROGRAM CHECKS FOR ICIR = 1,
; * IRDY = 1 AGAIN.
; * THE REST OF THE BINARY COUNT DATA BYTES ARE CYCLED 8 CLOCKS AND READ AND
; * COMPARED A BYTE AT A TIME.

```

```

2057
2058 033726      ;*****
033726      BGNST
2059 033726 012737 034202 002362      MOV    #A10,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
2060      ;-----
2061      ; DO MASTER CLR, CHK FOR ICIR = 1, IRDY = 0
2062      ;-----
2063 033734 004737 003576      JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR
2064 033740 004737 006430      JSR    PC,ISIRDY      ;CHK ICIR = 1, IRDY = 0
2065 033744 000001      1
2066      ;-----
2067      ; LOAD AND CLOCK 2 SOM'S, LOAD 64 BYTES OF BINARY COUNT PATTERN INTO TX SILO,
2068      ; CLOCK LINE UNIT, CHK FOR IRDY = 1 WITHIN 40-43 CYCLES
2069      ;-----
2070 033746 004737 005540      JSR    PC,INITRN      ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
2071 033752 000226      SYNCH
2072 033754 000011      STRIP!DDCMP
2073 033756 005003      CLR    R3      ;INIT BINARY COUNT DATA FOR WRITING
2074 033760 010337 002422 2$: MOV    R3,TXWORD
2075 033764 004737 005174      JSR    PC,LDTXSI      ;LOAD A DATA BYTE INTO TX SILO
2076 033770 005203      INC    R3      ;INCR DATA
2077 033772 020327 000100      CMP    R3,#64.      ;SEE IF 64 BYTES LOADED YET
2078 033776 002770      BLT    2$      ;BR IF NOT YET
2079 034000 004737 007434      JSR    PC,RCV1ST      ;RECEIVE AND TIME FIRST CHARACTER
2080 034004 000050      40.
2081      ;-----
2082      ; READ RCV SILO, COMPARE FIRST CHAR TO 000, CHK FOR ICIR = 1, IRDY = 0
2083      ;-----
2084 034006 005004      9$: CLR    R4      ;INIT PATTERN FOR READING
2085 034010 004737 007750      JSR    PC,CKDATA      ;READ RCV SILO, COMPARE DATA
2086 034014 000000      0      ;EXPECTED DATA = 000
2087 034016 000000      0      ;DON'T CLOCK LINE UNIT
2088 034020 005204      16$: INC    R4      ;INCR DATA FOR READING
2089 034022 004737 006430      JSR    PC,ISIRDY      ;CHK FOR ICIR = 1, IRDY = 0
2090 034026 000001      1
2091      ;-----
2092      ; CLOCK 63 CHARS INTO RCV SILO, CHK ICIR = 1, IRDY = 1
2093      ;-----
2094 034030 004737 005254      18$: JSR    PC,STPLU      ;CLOCK LU FOR 8 CYCLES
2095 034034 000010      8.
2096 034036 010337 002422      MOV    R3,TXWORD
2097 034042 004737 005174      JSR    PC,LDTXSI      ;LOAD ANOTHER WORD INTO TX SILO
2098 034046 005203      INC    R3      ;INCR PATTERN FOR WRITING
2099 034050 004737 006430      JSR    PC,ISIRDY      ;CHK ICIR = 1, IRDY = 1
2100 034054 000003      3
2101 034056 020327 000177      CMP    R3,#127.      ;SEE IF 63 MORE CHARS CLOCKED YET
2102 034062 002762      BLT    18$      ;BR IF NOT YET
2103      ;-----
2104      ; CLOCK 1 MORE CHAR INTO RCV SILO, CHK ICIR = 0, IRDY = 1
2105      ;-----
2106 034064 004737 005254      JSR    PC,STPLU      ;CLOCK LU FOR 8 CYCLES
2107 034070 000010      8.
2108 034072 004737 006430      JSR    PC,ISIRDY      ;CHK ICIR = 0, IRDY = 1
2109 034076 000002      2
2110      ;-----
2111      ; READ, COMPARE, CLOCK REST OF DATA CHARS
2112      ;-----

```

2113	034100	010437	034110	20\$:	MOV	R4,21\$	;SET EXPECTED DATA
2114	034104	004737	007750		JSR	PC,CKDATA	;READ AND COMPARE DATA
2115	034110	000000		21\$:	0		;EXPECTED SILO ENTRY GOES HERE
2116	034112	000000			0		;DON'T CLOCK LINE UNIT
2117	034114	005204		22\$:	INC	R4	;INCR DATA PATTERN FOR READS
2118	034116	020427	000400		CMP	R4,#400	;SEE IF ALL DONE READING YET
2119	034122	001427			BEQ	32\$	;BR IF DONE READING
2120	034124	004737	006430		JSR	PC,ISIRDY	;CHK ICIR = 1, IRDY = 1
2121	034130	000003			3		
2122	034132	004737	005254		JSR	PC,STPLU	;CLOCK LU FOR 8 CYCLES
2123	034136	000010			8.		
2124	034140	020327	000377		CMP	R3,#377	;SEE IF ALL CHARS LOADED INTO TX SILO YET
2125	034144	003007			BGT	24\$	;BR IF YES
2126	034146	010337	002422		MOV	R3,TXWORD	
2127	034152	004737	005174		JSR	PC,LDTXSI	;LOAD ANOTHER CHAR INTO TX SILO
2128	034156	005203			INC	R3	;INCR DATA PATTERN FOR WRITING
2129	034160	000137	034100		JMP	20\$	
2130	034164	012737	001000	002422	24\$:	MOV	#TXEOM,TXWORD
2131	034172	004737	005174		JSR	PC,LDTXSI	;LOAD EOM INTO TX SILO
2132	034176	000137	034100		JMP	20\$	
2133	034202			32\$:			
2134	034202	004737	003576	A10:	JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
2135	034206			ENDTST			
	034206	104401					

L10070:

TRAP C\$ETST

2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153

```

*****
;SBTIL      TEST 37 - RECEIVER CHAR LENGTH TIMING TEST - CHAR MODE, NO CRC
;
;* TX LINE UNIT IS PLACED IN CHAR MODE, WITH NO ERROR DETECTION, AND A MSG IS
;* INITIATED WITH 2 SYNCH CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
;* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
;* SET, WHILE THE RECEIVER CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 5,6,7,8.
;* FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THE USYRT RECEIVER
;* FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV CHAR LENGTH.
;* (FOR EXAMPLE A 5-BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED). A MASTER
;* CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
*****
BGNTST

```

2154	034210	012737	034402	002362	MOV	#24\$,REIADR	;SET TEST EXIT ADRS FOR ERRORS
2155	034216	004737	005540		JSR	PC,INITRN	;DO MASTER CLR, LOAD 2 SOM'S
2156	034222	000000			000		
2157	034224	000341			CRC2!CRC1!IDLE!DDCMP		
2158	034226	012701	000017		MOV	#15.,R1	;INIT COUNTER
2159	034232	005037	002422		CLR	TXWORD	
2160	034236	004737	005174	3\$:	JSR	PC,LDTXSI	;LOAD A 000 CHAR INTO TX SILO
2161	034242	005301			DEC	R1	;DECR COUNTER
2162	034244	001374			BNE	3\$	;BR IF NOT DONE LOADING YET
2163	034246	004737	005146		JSR	PC,WAIT50	;WAIT FOR SILO TO RIPPLE
2164	034252	012737	000006	002402	MOV	#6,AXNUM	;SET BYTE NO. = 6 FOR AX3
2165	034260	012737	000000	002374	MOV	#000,WAX15	;SET DATA FOR AX3-15 = 0
2166	034266	012737	000005	002376	MOV	#RXLEN2!RXLENO,WAX16	;SET RCV LEN = 5

2167	034274	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2168	034300	004737	005254		JSR	PC,STPLU	;CLK LU UNTIL TX'ING 1ST DATA CHAR
2169	034304	100012			CHPCHK!10.		
2170	034306	012737	000006	002376	MOV	#RXLEN2!RXLEN1,WAX16	;SET RCV LEN = 6
2171	034314	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2172	034320	004737	007434		JSR	PC,RCV1ST	;CLOCK 5-BIT DATA CHAR
2173	034324	000005			5		
2174	034326	012737	000007	002376	MOV	#RXLEN2!RXLEN1!RXLENO,WAX16	;SET RCV LEN = 7
2175	034334	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2176	034340	004737	011074		JSR	PC,RXCHAR	;RCV 5-BIT DATA CHAR, CLK 6-BIT
2177	034344	000006			6		
2178	034346	012737	000000	002376	MOV	#0,WAX16	;SET RCV LEN = 8
2179	034354	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2180	034360	004737	011074		JSR	PC,RXCHAR	;RCV 6-BIT DATA CHAR, CLK 7-BIT
2181	034364	000007			7		
2182	034366	004737	011074		JSR	PC,RXCHAR	;RCV 7-BIT DATA CHAR, CLK 8-BIT
2183	034372	000010			8.		
2184	034374	004737	011074		JSR	PC,RXCHAR	;RCV 8-BIT DATA CHAR
2185	034400	000010			8.		
2186	034402	004737	003576		JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
2187	034406				24\$:		
	034406				ENDTST		
	034406						L10071:
	104401						TRAP C\$ETST

2188
 2189
 2190
 2191
 2192
 2193
 2194
 2195
 2196
 2197
 2198
 2199
 2200
 2201
 2202
 2203
 2204

```

*****
SBTTL      TEST 38 - RECEIVER CHAR LENGTH TIMING TEST - BIT MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN BIT MODE WITH NO ERROR DETECTION, AND A MESSAGE IS
* INITIATED WITH 2 FLAG CHARS. NEXT, FIFTEEN 000 CHARS ARE LOADED INTO THE
* TRANSMITTER SILO. THE LINE UNIT IS THEN CLOCKED USING STEPLU WITH LULOOP
* SET. WHILE THE RCV CHAR LENGTH IS SET TO THE FOLLOWING VALUES : 8,8,8,7,6,5,
* 4,3,2,1. FOR EACH RCV CHAR LENGTH, THE PROGRAM CHECKS TO MAKE SURE THAT THE
* USYRT RECEIVER FLAGS OCCUR THE PROPER NO. OF CYCLES APART, FOR EACH RCV
* CHAR LENGTH. (FOR EXAMPLE, A 5 BIT CHAR TAKES 5 CLOCK CYCLES TO BE RECEIVED).
* A MASTER CLEAR IS THEN DONE TO TERMINATE THE OPERATION.
*****
BGNTST
  
```

2205	034410						
	034410						T38::
2206	034410	012737	034702	002362	MOV	#24\$,RETADR	;SET TEST EXIT ADRS FOR ERRORS
2207	034416	004737	005540		JSR	PC,INITRN	;DO MASTER CLR, LOAD 2 SOM'S
2208	034422	000000			000		
2209	034424	000300			CRC2!CRC1		
2210	034426	012701	000017		MOV	#15.,R1	;INIT COUNTER
2211	034432	005037	002422		CLR	TXWORD	
2212	034436	004737	005174		JSR	PC,LDTXSI	;LOAD A 000 CHAR INTO TX SILO
2213	034442	005301			DEC	R1	;DECR COUNTER
2214	034444	001374			BNE	3\$	;BR IF NOT DONE LOADING YET
2215	034446	004737	005146		JSR	PC,WAIT50	;WAIT FOR SILO TO RIPPLE
2216	034452	012737	000006	002402	MOV	#6,AXNUM	;SET BYTE NO. = 6 FOR AX3
2217	034460	012737	000000	002374	MOV	#000,WAX15	;SET DATA FOR AX3-15 = 0
2218	034466	004737	007434		JSR	PC,RCV1ST	;CLOCK FIRST 8-BIT DATA CHAR
2219	034472	000040			32.		
2220	034474	012737	000000	002376	MOV	#0,WAX16	;SET RCV LEN = 8

2221	034502	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2222	034506	004737	011074		JSR	PC,RXCHAR	;RCV FIRST 8-BIT DATA CHAR, CLK SECOND 8-BIT
2223	034512	000010			8.		
2224	034514	012737	000007	002376	MOV	#RXLEN2!RXLEN1!RXLENO,WAX16	;SET RCV LEN = 7
2225	034522	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2226	034526	004737	011074		JSR	PC,RXCHAR	;RCV SECOND 8-BIT DATA CHAR, CLK 3RD 8-BIT
2227	034532	000010			8.		
2228	034534	012737	000006	002376	MOV	#RXLEN2!RXLEN1,WAX16	;SET RCV LEN = 6
2229	034542	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2230	034546	004737	011074		JSR	PC,RXCHAR	;RCV 3RD 8-BIT DATA CHAR, CLK 7-BIT
2231	034552	000007			7		
2232	034554	012737	000005	002376	MOV	#RXLEN2!RXLENO,WAX16	;SET RCV LEN = 5
2233	034562	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2234	034566	004737	011074		JSR	PC,RXCHAR	;RCV 7-BIT DATA CHAR, CLK 6-BIT
2235	034572	000006			6		
2236	034574	012737	000004	002376	MOV	#RXLEN2,WAX16	;SET RCV LEN = 4
2237	034602	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2238	034606	004737	011074		JSR	PC,RXCHAR	;RCV 6-BIT DATA CHAR, CLK 5-BIT
2239	034612	000005			5		
2240	034614	012737	000003	002376	MOV	#RXLEN1!RXLENO,WAX16	;SET RCV LEN = 3
2241	034622	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2242	034626	004737	011074		JSR	PC,RXCHAR	;RCV 5-BIT DATA CHAR, CLK 4-BIT
2243	034632	000004			4		
2244	034634	012737	000002	002376	MOV	#RXLEN1,WAX16	;SET RCV LEN = 2
2245	034642	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2246	034646	004737	011074		JSR	PC,RXCHAR	;RCV 4-BIT DATA CHAR, CLK 3-BIT
2247	034652	000003			3		
2248	034654	012737	000001	002376	MOV	#RXLENO,WAX16	;SET RCV LEN = 1
2249	034662	004737	004312		JSR	PC,WRITAX	;LOAD AX3
2250	034666	004737	011074		JSR	PC,RXCHAR	;RCV 3-BIT DATA CHAR, CLK 2-BIT
2251	034672	000002			2		
2252	034674	004737	011074		JSR	PC,RXCHAR	;RCV 2-BIT DATA CHAR, CLK 1-BIT
2253	034700	000001			1		
2254	034702	004737	003576		24\$: JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
2255	034706				ENDTST		
	034706						
	034706	104401					

L10072:

TRAP C\$ETST

2256
 2257
 2258
 2259
 2260
 2261
 2262
 2263
 2264
 2265
 2266
 2267
 2268
 2269
 2270 034710
 034710
 2271 034710 012737 034772 002362
 2272 034716 004737 005540
 2273 034722 000226
 2274 034724 000341

 :SBTTL TEST 39 - TRANSMITTER UNDERRUN ERROR, IDLE MARKING, CHAR MODE,NO CRC
 :
 :* THE LINE UNIT IS PLACED IN CHAR MODE, AND THE IDLE BIT IS SET. THEN, A
 :* MSG IS INITIATED, A 000 CHAR IS SENT, AND THE TX BUFFER IS NOT SERVICED
 :* IN RESPONSE TO THE USYRT TX BUFFER EMPTY FLAG, WHICH CAUSES A TX UNDERRUN
 :* ERROR. THEN, THE RECEIVER IS CLOCKED AND CHECKED FOR TWO 377 CHARS TO BE
 :* RECEIVED (LINE MARKING) BEFORE SHUTTING DOWN WITH A MASTER CLEAR.
 :*****
 :BGNTST

T39::

MOV #24\$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
 SYNCH
 CRC2!CRC1!IDLE!DDCMP

```

2275 034726 012737 000000 002422      MOV      #000,TXWORD
2276 034734 004737 005174              JSR      PC,LDTXSI      ;LOAD 000 CHAR INTO TX SILO
2277 034740 004737 005254              JSR      PC,STPLU      ;CLOCK LINE UNIT UNTIL LINE GOES MARKING
2278 034744 000063                    51.
2279 034746 004737 007354              JSR      PC,RDRXSI      ;READ 000 CHAR
2280 034752 004737 007750              JSR      PC,CKDATA      ;READ AND CHECK FOR MARK CHAR (377)
2281 034756 000377                    377
2282 034760 000000                    000
2283 034762 004737 007750              JSR      PC,CKDATA      ;READ AND CHECK FOR ANOTHER MARK CHAR
2284 034766 000377                    377
2285 034770 000000                    000
2286 034772 004737 003576              JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
2287 034776                                24$:
                                ENDIST
                                L10073:
                                TRAP      CSETST

2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308 035000
2309 035000 012737 035134 002362      MOV      #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
2310 035006 004737 005540              JSR      PC,INTRN      ;MST CLR, LOAD 2 SOM'S
2311 035012 000000                    000
2312 035014 000310                    CRC2!CRC1!STRIP
2313 035016 004737 011016              JSR      PC,LODMSG      ;LOAD DATA CHARS INTO TX SILO
2314 035022 003224                    MSG1+4
2315 035024 000005                    5
2316 035026 012737 005000 002422      MOV      #TXGOA!TXEOM,TXWORD
2317 035034 004737 005174              JSR      PC,LDTXSI      ;LOAD A GA CHAR INTO TX SILO
2318 035040 004737 005174              JSR      PC,LDTXSI      ;LOAD ANOTHER GA
2319 035044 004737 005146              JSR      PC,WAIT50      ;ALLOW SILO TO RIPPLE
2320 035050 004737 005254              JSR      PC,STPLU      ;CLOCK LU UNTIL GA CHAR IS TX'ING
2321 035054 100071                    CHPCHK!57.
2322 035056 004737 011176              JSR      PC,CKTBIT      ;SCAN TXDATA BIT FOR GA CHAR
2323 035062 000376                    376
2324 035064 004737 007750              JSR      PC,CKDATA      ;RCV 000 CHAR, CLK 125
2325 035070 000000                    000
2326 035072 000000                    0
2327 035074 004737 007750              JSR      PC,CKDATA      ;RCV 125 CHAR, CLK 252
2328 035100 000125                    125

```

```

*****
SBTTL      TEST 40 - MSG TERMINATION WITH GA CHARS - BIT MODE,NO CRC
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN BIT MODE.
* 2 FLAG CHARACTERS ARE SENT, FOLLOWED BY
* THE FOLLOWING DATA CHARACTERS : 000, 125, 252, 377, 000. THEN THE LOOP
* MODE BIT (STRIP) IS SET AND 2 TERMINATING GO-AHEAD CHARACTERS ARE
* SENT. EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 5 DATA
* WORDS ARE READ AND COMPARED TO EXPECTED VALUES.
* ALSO, THE FIRST GA CHAR IS CHECKED BY SCANNING THE TXDATA BIT AS THE GA
* IS BEING TRANSMITTED (GA = 376 OCTAL).
* THE TEST ALSO CHECKS FOR SETTING OF RAB AND EBLK
* IN LOOP MODE.
*****
BGNTST

```

T40::

```

2329 035102 000000      0
2330 035104 004737 007750 JSR    PC,CKDATA      ;RCV 252 CHAR, CLK 377
2331 035110 000252      252
2332 035112 000000      0
2333 035114 004737 007750 JSR    PC,CKDATA      ;RCV 377 CHAR
2334 035120 000377      377
2335 035122 000010      8.
2336 035124 004737 007750 JSR    PC,CKDATA      ;RCV 000 CHAR, CHK RAB = 1, EBLK = 1
2337 035130 003000      3000
2338 035132 000010      8.
2339 035134 004737 003576 24$: JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR
2340 035140      ENDTST
      035140
      035140 104401
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359 035142
      035142
2360 035142 012737 035302 002362 MOV    #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
2361 035150 004737 005540 JSR    PC,INITRN      ;MST CLR, LOAD 2 SOM'S
2362 035154 000226      SYNCH
2363 035156 000341      CRC2!CRC1!IDLE!DDCMP
2364 035160 012701 000026 MOV    #22.,R1          ;INIT COUNTER
2365 035164 012737 000226 002422 MOV    #SYNCH,TXWORD
2366 035172 004737 005174 6$: JSR    PC,LDTXSI      ;LOAD AN SOM INTO TX SILO
2367 035176 005301      DEC    R1          ;DECR COUNTER
2368 035200 001374      BNE    6$          ;BR IF MORE TO LOAD
2369 035202 004737 005146 JSR    PC,WAIT50      ;ALLOW SILO TO RIPPLE
2370 035206 004737 007434 JSR    PC,RCV1ST      ;CLK LU UNTIL 3RD SYNCH RCV'D (RCVR IS ACTIVE)
2371 035212 000030      24.
2372 035214 012701 000025 MOV    #21.,R1          ;INIT COUNTER
2373 035220 004737 007750 8$: JSR    PC,CKDATA      ;READ A SYNCH, CLK NEXT ONE
2374 035224 000226      SYNCH
2375 035226 000010      8.
2376 035230 005301      DEC    R1          ;DECR COUNTER
2377 035232 001372      BNE    8$          ;BR IF NOT ALL CHECKED
2378 035234 004737 007750 JSR    PC,CKDATA      ;CHECK LAST SYNCH
2379 035240 000226      SYNCH
2380 035242 000004      4
2381 035244 012737 000011 002400 MOV    #11,REGNUM      ;SET REG NO. 11
2382 035252 112737 000200 002366 MOVB   #0C,WRIBYT

```

L10074: TRAP C\$ETST

```

*****
;SBTTL      TEST 41 - IDLE SYNCHS TEST - CHAR MODE
;
;* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
;* INITIATED IN CHAR MODE. 24(DEC) SYNCHS ARE SENT.
;* EACH SYNCH IS TIMED AS IT IS RECEIVED, AND THE BITS ARE CHECKED
;* FOR A VALID SYNCH CHAR FOR EACH OF THE 22 SYNCHS WHICH FOLLOW
;* THE FIRST TWO (THESE PERFORM SYNCHRONIZATION, AND ARE NOT READ).
;* WHILE THE LAST SYNCH IS BEING TRANSMITTED, OC IS SET, AND THE
;* NEXT CHAR RCV'D AFTER THE SYNCH IS CHECKED TO BE 377 (LINE MARKING).
;* THEN, A MASTER CLEAR IS ISSUED.
;* THE SYNCH CHAR USED IS 226 (OCTAL).
*****
;BGNST

```

T41::

2383	035260	004737	003750	JSR	PC,WRITLU	;SET OC IN REG 11
2384	035264	004737	005254	JSR	PC,STPLU	;FINISH CLOCKING CHAR, AND THEN SOME
2385	035270	000014		12.		
2386	035272	004737	007750	JSR	PC,CKDATA	;RCV A MARK CHAR, CHK IT
2387	035276	000377		377		
2388	035300	000000		0		
2389	035302	004737	003576	24\$: JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
2390	035306			ENDTST		
	035306					L10075:
	035306	104401				TRAP C\$ETST

2391
 2392
 2393
 2394
 2395
 2396
 2397
 2398
 2399
 2400
 2401
 2402
 2403
 2404
 2405
 2406
 2407
 2408
 2409
 2410
 2411
 2412

```

*****
:SBTTL      TEST 42 - STRIP SYNCH TEST
:
:* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
:* INITIATED IN CHAR MODE AND WITH THE STRIP SYNCH
:* BIT SET. THEN 24 (DEC) SYNCHS ARE SENT
:* FOLLOWED BY THE FOLLOWING DATA CHARACTERS : 377, 000, 125, 252,
:* AND 2 TERMINATING SYNCHS.
:* EACH OF THE 23 SYNCHS AFTER THE FIRST ARE CHECKED AT THE TRANSMITTER OUTPUT,
:* BY SCANNING THE TXDATA BIT.
:* EACH USYRT RCV FLAG IS TIMED AS IT IS RECEIVED, AND THE 4 DATA WORDS
:* ARE READ AND COMPARED TO EXPECTED VALUES.
:* FINALLY, THE LINE UNIT IS CLOCKED FOR SEVERAL CHAR TIMES, AND A CHECK
:* IS MADE FOR OACT = 0 (TEOM SHOULD CAUSE TX ENABLE TO DROP).
:* THE ABOVE TEST IS REPEATED FOR EACH OF THE FOLLOWING SYNCH CHAR DATA
:* PATTERNS : 226,000,125,252,376,177.
*****
:BGNTST

```

2413	035310					
	035310					T42::
2414	035310	012737	035530	002362	MOV	#24\$,RETADR ;SET TEST EXIT ADRS FOR ERRORS
2415	035316	012701	035536		MOV	#SYNPA1,R1 ;GET POINTER TO DATA
2416	035322	011137	035332		2\$: MOV	(R1),3\$;GET A SYNCH PATTERN
2417	035326	004737	005540		JSR	PC,INITRN ;MST CLR, LOAD 2 SOM'S
2418	035332	000000			3\$: .WORD	0 ;SYNCH PATTERN GOES HERE
2419	035334	000311				
2420	035336	012737	000400	002422	CRC2!CRC1!STRIP!DDCMP	
2421	035344	012702	000026		MOV	#TXSOM,TXWORD
2422	035350	004737	005174		6\$: MOV	#22.,R2 ;LOAD 22 SOM'S INTO TX SILO
2423	035354	005302			JSR	PC,LDTXSI
2424	035356	001374			DEC	R2
2425	035360	004737	011016		BNE	6\$
2426	035364	003246			JSR	PC,LODMSG ;LOAD DATA CHARS INTO TX SILO
2427	035366	000006			MSG4	
2428	035370	012737	001000	002422	6	
2429	035376	004737	005174		MOV	#TXEOM,TXWORD
2430	035402	004737	005174		JSR	PC,LDTXSI ;LOAD A TEOM
2431	035406	004737	005146		JSR	PC,LDTXSI ;LOAD ANOTHER TEOM
2432	035412	011137	035434		JSR	PC,WAIT50 ;ALLOW SILO TO RIPPLE
2433	035416	012702	000027		MOV	(R1),16\$;GET CURRENT SYNCH PATTERN
2434	035422	004737	005254		MOV	#23.,R2 ;INIT COUNTER
2435	035426	100010			JSR	PC,STPLU ;CLOCK OUT FIRST SYNCH
2436	035430	004737	011176		CHPCHK!8.	
				14\$: JSR	PC,CKTBIT ;CHECK TX'D SYNCH	

CZDMRD M8203 STATIC DIAG #1
TEST 42 - STRIP SYNCH TEST

MACRO V03.01 7-AUG-80 11:01:24 PAGE 24-50

SEQ 171

```

2437 035434 000000      16$: .WORD 0      ;SYNCH PATTERN GOES HERE
2438 035436 005302      DEC R2      ;DECR COUNTER
2439 035440 001373      BNE 14$     ;BR IF NOT DONE CHECKING LAST 23 SYNCHS
2440 035442 004737 007434 JSR PC,RCV1ST ;CLOCK UNTIL 000 CHAR RCV'D
2441 035446 000010      8.
2442 035450 004737 007750 JSR PC,CKDATA ;RCV 377, CLOCK 000
2443 035454 000377      377
2444 035456 000010      8.
2445 035460 004737 007750 JSR PC,CKDATA ;RCV 000, CLK 125
2446 035464 000000      000
2447 035466 000010      8.
2448 035470 004737 007750 JSR PC,CKDATA ;RCV 125, CLK 252
2449 035474 000125      125
2450 035476 000010      8.
2451 035500 004737 007750 JSR PC,CKDATA ;RCV 252, CLK END OF MSG
2452 035504 000252      252
2453 035506 000040      32.
2454 035510 004737 005352 JSR PC,OACTIV ;CHK FOR OACT = 0
2455 035514 000000      0
2456 035516 062701 000002 ADD #2,R1    ;INIT SYNCH PATTERN POINTER
2457 035522 020127 035552 CMP R1,#SYNPAT+12. ;SEE IF ALL PATTERNS CHECKED YET
2458 035526 103675      BLO 2$      ;BR IF NOT YET
2459 035530 004737 003576 JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
2460 035534      END-ST
      035534
      035534 104401
      L10076: TRAP CSETST

2461
2462 035536 000226      SYNPAT: 226
2463 035540 000000      000
2464 035542 000125      125
2465 035544 000252      252
2466 035546 000376      376
2467 035550 000177      177
2468
2469
2470
2471
2472

```

```

1      .SBTTL  HARDWARE PARAMETER CODING SECTION
2
3
4
5      :///////////////////////////////////////////////////
6      :// THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7      :// THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8      :// MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
9      :// INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
10     :// MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11     :// WITH THE OPERATOR.
12     :///////////////////////////////////////////////////
13
14     035552      BGNHRD
15     035552      000022
16     035554      GPRMA  ADDRES,2,0,160000,177776,YES
17     035554      GPRMA  VECTOR,4,0,0,674,YES
18     035574      GPRMD  PRIRTY,6,0,7000,4,7,YES
19     035606      GPRMD  ISRUN,24,0,7,0,7,YES
20
21     035620      ENDRD
22
23     035620      104      105      126  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
24     035623      111      103      105
25     035626      040      103      123
26     035631      122      040      101
27     035634      104      104      122
28     035637      105      123      123
29     035642      040      072      040
30     035645      000
31     035646      104      105      126  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
32     035651      111      103      105
33     035654      040      126      105
34     035657      103      124      117
35     035662      122      040      101
  
```

	035665	104	104	122	
	035670	105	123	123	
	035673	040	072	040	
	035676	000			
25	035677	104	105	126	PRIITY: .ASCIZ /DEVICE PRIORITY LEVEL : /
	035702	111	103	105	
	035705	040	120	122	
	035710	111	117	122	
	035713	111	124	131	
	035716	040	114	105	
	035721	126	105	114	
	035724	040	072	040	
	035727	000			
26	035730	115	070	062	ISRUN: .ASCIZ /M8207 RUN SWITCH (E28 SW7) - TYPE 0 IF OFF, 1 IF ON : /
	035733	060	067	040	
	035736	122	125	116	
	035741	040	123	127	
	035744	11	124	103	
	035747	110	040	050	
	035752	105	062	070	
	035755	040	123	127	
	035760	067	051	040	
	035763	055	040	124	
	035766	131	120	105	
	035771	040	060	040	
	035774	111	106	040	
	035777	117	106	106	
	036002	054	040	061	
	036005	040	111	106	
	036010	040	117	116	
	036013	040	072	040	
	036016	000			
27					.EVEN
28					
29					
30					
31					
32					
33					

```

1      .SBTTL  SOFTWARE PARAMETER CODING SECTION
2
3
4      ;////////////////////////////////////
5      ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
6      ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7      ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8      ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
9      ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10     ;/ WITH THE OPERATOR.
11     ;////////////////////////////////////
12
13     036020      BGNSFT
14     036020      000000
15     036022
16     036022      ENDSFT
17     036022
18     .EVEN
19
20
21
22
23
24
25
26
27     ;***** PATCH AREA FOR DEBUG *****
28     PATCH:
29     .=.+200
30     NOP
31     NOP
32     NOP
33     ;*****
34
35
36
37     036230      ENDMOD
38
39     036230      LASTAD
40
41     036230      000000
42     036232      000000
43     036234
44
45     .EVEN
46     .WORD      0
47     .WORD      0
48
49     L$LAST::
50
51     .END

```

ABORT = 000004	BIT7 = 000200 G	C\$OPEN= 000034	EM23 012775	EVL = 000004 G
ADDRES 035620	BIT8 = 000400 G	C\$PNTB= 000014	EM24 013016	E\$END = 002100
ADR = 000020 G	BIT9 = 001000 G	C\$PNTF= 000017	EM25 013033	E\$LOAD= 000035
ANBITS 002561	BOE = 000400 G	C\$PNTS= 000016	EM26 013067	FMT1 011454
APA = 000200	BPOLL = 000100	C\$PNTX= 000015	EM27 013121	FMT10 011717
ASBC0 = 000020	BSEL1 002450	C\$QIO = 000377	EM28 013152	FMT11 011750
ASBC1 = 000040	BSEL2 002452	C\$RDBU= 000007	EM29 013173	FMT19 012007
ASBC2 = 000100	BSEL4 002454	C\$REFG= 000047	EM3 012220	FMT2 011464
ASSEMB= 000010	CARR = 000001	C\$RESE= 000033	EM30 013210	FMT24 012040
AXNUM 002402	CHPCHK= 100000	C\$REVI= 000003	EM31 013231	FMT27 021600
AX0.15= 002322	CHPTYP 002430	C\$RFLA= 000021	EM32 013246	FMT3 011506
AX0.16= 002324	CKDATA 007750	C\$RPT = 000025	EM33 013275	FMT4 011550
AX1 = 000001	CKTBIT 011176	C\$SEFG= 000046	EM34 013320	FMT5 011563
AX1.15= 002326	CRCCHK= 100000	C\$SPRI= 000041	EM35 013346	FMT6 011613
AX1.16= 002330	CRCTY0= 000001	C\$SVEC= 000037	EM36 013366	FMT7 011646
AX2 = 000002	CRCTY1= 000002	C\$TPRI= 000013	EM37 013402	FMT8 011656
AX2.15= 002332	CRCTY2= 000004	C32BCC= 000040	EM38 013423	FMT9 011712
AX2.16= 002334	CRC1 = 000100	C32ENB= 000004	EM39 013440	FRSTIM 002412
AX3.15= 002336	CRC2 = 000200	DDC = 000100	EM4 012237	F\$AU = 000015
AX3.16= 002340	CS = 000004	DDCMP = 000001	EM40 013460	F\$AUTO= 000020
AX315U= 000372	C\$AU = 000052	DEVMAP 002434	EM41 013474	F\$BGN = 000040
A1 024220	C\$AUTO= 000061	DEVPTR 002436	EM42 013515	F\$CLEA= 000007
A10 034202	C\$BRK = 000022	DFPTBL 002252 G	EM43 013532	F\$DU = 000016
A2 026626	C\$BSEG= 000004	DH1 014264	EM44 013557	F\$END = 000041
A3 027306	C\$BSUB= 000002	DH2 014306	EM45 013604	F\$HARD= 000004
A4 027422	C\$CEFG= 000045	DH3 014335	EM46 013631	F\$HW = 000013
A5 027532	C\$CLCK= 000062	DH4 014373	EM47 013664	F\$INIT= 000006
A6 027726	C\$CLEA= 000012	DH5 014435	EM48 013717	F\$JMP = 000050
A7 030272	C\$CLOS= 000035	DH6 014440	EM49 013752	F\$MO2 = 000000
A8 030506	C\$CLP1= 000006	DH7 014443	EM5 012312	F\$MSG = 000011
A9 031036	C\$CVEC= 000036	DH8 014475	EM50 014011	F\$PROT= 000021
BADDAT 002406	C\$DCLN= 000044	DH9 014534	EM51 014045	F\$PWR = 000017
BCC = 000001	C\$DODU= 000051	DIAGMC= 000000	EM52 014105	F\$RPT = 000012
BCCCHK= 100000	C\$DRPT= 000024	DISILO 002426	EM53 014146	F\$SEG = 000003
BIT0 = 000001 G	C\$DU = 000053	DISSI = 000040	EM54 014206	F\$SOFT= 000005
BIT00 = 000001 G	C\$EDIT= 000003	DTR = 000100	EM6 012343	F\$SRV = 000010
BIT01 = 000002 G	C\$ERDF= 000055	EBLK = 000002	EM60 014230	F\$SUB = 000002
BIT02 = 000004 G	C\$ERHR= 000056	EF.CON= 000036 G	EM65 014244	F\$SW = 000014
BIT03 = 000010 G	C\$ERPO= 000060	EF.NEW= 000035 G	EM7 012374	F\$TEST= 000001
BIT04 = 000020 G	C\$ERSF= 000054	EF.PWR= 000034 G	EM8 012411	GETALL 004526
BIT05 = 000040 G	C\$ERSO= 000057	EF.RES= 000037 G	EM9 012432	GETPRM 021306
BIT06 = 000100 G	C\$ESCA= 000010	EF.STA= 000040 G	ENAX = 000004	GETREG 004016
BIT07 = 000200 G	C\$ESEG= 000005	EM1 012125	ENDIT 021464	GOAH = 000010
BIT08 = 000400 G	C\$ESUB= 000003	EM10 012447	ENDPAT 003220	GOODAT 002404
BIT09 = 001000 G	C\$ETST= 000001	EM11 012470	ENDTRN 006274	G\$CNT0= 000200
BIT1 = 000002 G	C\$EXIT= 000032	EM12 012505	EOM = 000002	G\$DELM= 000372
BIT10 = 002000 G	C\$GETB= 000026	EM13 012526	ERRFLG 002356	G\$DISP= 000003
BIT11 = 004000 G	C\$GETW= 000027	EM14 012556	ERROR1 002420	G\$EXCP= 000400
BIT12 = 010000 G	C\$GMAN= 000043	EM15 012573	ERR1 014600 G	G\$HILI= 000002
BIT13 = 020000 G	C\$GPHR= 000042	EM16 012622	ERR2 014632 G	G\$LOLI= 000001
BIT14 = 040000 G	C\$GPLO= 000030	EM17 012643	ERR3 015140 G	G\$NO = 000000
BIT15 = 100000 G	C\$GPRI= 000040	EM18 012660	ERR4 015622 G	G\$OFFS= 000400
BIT2 = 000004 G	C\$INIT= 000011	EM19 012701	ERR5 016300 G	G\$OFFSI= 000376
BIT3 = 000010 G	C\$INLP= 000020	EM2 012161	ERR6 017012 G	G\$PRMA= 000001
BIT4 = 000020 G	C\$MANI= 000050	EM20 012716	ERR7 017470 G	G\$PRMD= 000002
BIT5 = 000040 G	C\$MEM = 000031	EM21 012737	ERR8 020122 G	G\$PRML= 000000
BIT6 = 000100 G	C\$MSG = 000023	EM22 012754	ERR9 020630 G	G\$RADA= 000140

GSRADB= 000000	LUREG 002302	LSSOFT 036022 G	L10061 031136	PATH 002654
GSRADD= 000040	LUR10 = 002302	LSSPC 002056 G	L10062 031506	PATI 002664
GSRADL= 000120	LUR11 = 002304	LSSPCP 002020 G	L10063 032050	PATJ 002674
GSRADO= 000020	LUR12 = 002306	LSSPTP 002024 G	L10064 032320	PATK 002704
GSXFER= 000004	LUR13 = 002310	LSSSTA 002030 G	L10065 032662	PATL 003014
G\$YES = 000010	LUR14 = 002312	LSSW 002302 G	L10066 033012	PATM 003022
HDX = 000020	LUR15 = 002314	LSTEST 002114 G	L10067 033724	PATN 003032
HELP = 000001	LUR16 = 002316	LSTIML 002014 G	L10070 034206	PATO 003050
HOE = 100000 G	LUR17 = 002320	L\$UNIT 002012 G	L10071 034406	PATP 003066
IACT = 000100	LUSW11 002466	L10000 002300	L10072 034706	PATU 003104
IACTIV 006714	LUSW12 002470	L10001 002302	L10073 034776	PATV 003152
IBE = 010000 G	LUSW13 002472	L10002 014630	L10074 035140	PNT = 001000 G
IC = 000200	LU1MOD 002000 G	L10003 015136	L10075 035306	POLL = 000200
ICIR = 000010	LSACP 002110 G	L10004 015620	L10076 035534	PRI = 002000 G
IDL = 000010	LSAPT 002036 G	L10005 016276	L10077 035620	PRIOR 002350
IDLE = 000040	LSAU 021632 G	L10006 017010	L10100 036022	PRIPTY 035677
IDU = 000040 G	LSAUT 002070 G	L10007 017466	MAINT1= 000010	PRI00 = 000000 G
IER = 020000 G	LSAUTO 021466 G	L10010 020120	MAINT2= 000004	PRI01 = 000040 G
IERR = 000002	LSACP 002106 G	L10011 020626	MCLK = 000002	PRI02 = 000100 G
INITRN 005540	LSCCP 002106 G	L10012 021104	MCLR = 000100	PRI03 = 000140 G
INTFLG 002354	LSCLEA 021546 G	L10013 021106	MODR = 000010	PRI04 = 000200 G
INTGRL= 000010	LSCD 002032 G	L10015 021464	MPCSR 002446	PRI05 = 000240 G
IRDY = 000020	LSDEPO 002011 G	L10016 021544	MPIVEC 002460	PRI06 = 000300 G
ISIRDY 006430	LSDESC 003470 G	L10017 021546	MPOVEC 002462	PRI07 = 000340 G
ISR = 000100 G	LSDESP 002076 G	L10020 021576	MPRIOR 002464	PSTACK 002346
ISRUN 035730	LSDEVP 002060 G	L10021 021632	MSG1 003220	RAB = 000004
IXE = 004000 G	LSDISP 002124 G	L10022 021632	MSG4 003246	RABT = 000004
ISAU = 000041	LSDLY 002116 G	L10023 021714	MSTCLR 003576	RAX15 002370
ISAUTO= 000041	LSDTP 002040 G	L10024 022000	MVIOX = 021000	RAX16 002372
ISCLN = 000041	LSDTYP 002034 G	L10024 022124	MVIXO = 122000	RCVBUF 003262
ISDU = 000041	LSDU 021550 G	L10025 022226	MVIXOX= 121000	RCV1ST 007434
ISHRD = 000041	LSDUT 002072 G	L10026 022354	NEWST 021266	RDALL = 000004
ISINIT= 000041	LSDVTY 003462 G	L10027 022474	OACT = 000100	RDAX = 000020
ISMOD = 000041	LSEF 002052 G	L10030 022650	OACTIV 005352	RDRXSI 007354
ISMSG = 000041	LSENV1 002044 G	L10031 023104	OC = 000200	READAX 004124
ISPROT= 000040	LSETP 002102 G	L10032 023222	OCOR = 000020	READLU 003672
ISPTAB= 000041	LSEXP1 002046 G	L10033 023340	OP = 000002	READY = 000200
ISPR = 000041	LSEXP4 002064 G	L10034 023456	ORDY = 000020	REDBYT 002364
ISRPT = 000041	LSEXP5 002066 G	L10035 023574	OSIRDY 004662	REDDAT 002500
ISSEG = 000041	LSHARD 035554 G	L10036 024224	OVRR = 000010	REGNUM 002400
ISSETU= 000041	LSHIME 002120 G	L10037 024556	OSAPTS= 000000	REG0 002510
ISSFT = 000041	LSHPCP 002016 G	L10040 025046	OSAU = 000001	REG1 002512
ISSRV = 000041	LSHPTP 002022 G	L10041 025256	OSBGNR= 000000	REG2 002514
ISSUB = 000041	LSHW 002252 G	L10042 025462	OSBGNS= 000000	REG3 002516
ISTST = 000041	LSICP 002104 G	L10043 025666	OSDU = 000001	REG4 002520
I422 = 000200	LSINIT 021116 G	L10044 026124	OSERRT= 000000	REG5 002522
JSJMP = 000167	LSLADP 002026 G	L10045 026356	OSGNSW= 000000	REG6 002524
LDMSG1 011374	LSLAST 036234 G	L10046 026262	OSPOIN= 000001	REG7 002526
LDTXSI 005174	LSLOAD 002100 G	L10047 026354	OSSETU= 000000	REOM = 000002
LOADAT 002410	LSLUN 002074 G	L10050 026630	PATA 002571	RERR = 000200
LODM5G 011016	LSMREV 002050 G	L10051 026626	PATB 002615	RETADR 002362
LOE = 040000 G	LSNAME 002000 G	L10052 027306	PATC 002625	RING = 000200
LOGDEV 002344	LSPRIO 002042 G	L10053 027426	PATCH 03602	ROMI = 000002
LOCPIN 004074	LSPROT 021110 G	L10054 027536	PATD 002630	ROMO = 000004
LOT = 000010 G	LSRPT 002112 G	L10055 027732	PATE 002633	ROR = 000010
LULOOP= 000010	LSREPP 002062 G	L10056 030276	PATF 002641	RRDYTO= 000001
LILP = 000040	LSREV 002010 G	L10057 030506	PATG 002644	RSEOM 007102
	LSRPT 021106 G	L10060 031036		

RSOM = 000001	STPCLK 003540	TXABT = 002000	T\$TEMP= 000000	T29 031040 G
RTS = 000040	STPERR 007646	TXCHAR 006122	T\$TEST= 000052	T3 022002 G
RL = 000200	STPLU 005254	TXDATA= 000040	T\$TSTM= 177777	T30 031140 G
RUNINH 002476	STR = 000040	TXEN = 000100	T\$TSTS= 000001	T31 031510 G
RXABT = 002000	STRIP = 000010	TXEOM = 001000	T\$SAU = 010021	T32 032052 G
RXBCC = 000400	SUBRPC 002352	TXGA = 000010	T\$SAUT= 010016	T33 032322 G
RXCHAR 011074	SVCGBL= 000000	TXGOA = 004000	T\$SCLE= 010017	T34 032664 G
RXEBL = 001000	SVCINS= 000001	TXLEN0= 000040	T\$SDU = 010020	T35 033014 G
RXLEN0= 000001	SVCSUB= 000001	TXLEN1= 000100	T\$SHAR= 010077	T36 033726 G
RXLEN1= 000002	SVCTAG= 000001	TXLEN2= 000200	T\$SHW = 010000	T37 034210 G
RXLEN2= 000004	SVCTST= 000001	TXSOM = 000400	T\$SINI= 010015	T38 034410 G
RXOVR = 004000	SW0 = 000002	TXWORD 002422	T\$MSG= 010012	T39 034710 G
RXWORD 002424	SW1 = 000004	TX0 = 000001	T\$PRO= 010014	T4 022126 G
RX0 = 000001	SW2 = 000010	TX1 = 000002	T\$RPT= 010013	T40 035000 G
RX1 = 000002	SW3 = 000040	TX2 = 000004	T\$SEG= 010000	T41 035142 G
RX2 = 000004	SYNCH = 000226	TX3 = 000010	T\$SOF= 010100	T42 035310 G
RX3 = 000010	SYNPAT 035536	TX4 = 000020	T\$SUB= 010051	T5 022230 G
RX4 = 000020	SYN0 = 000001	TX5 = 000040	T\$SW = 010001	T6 022554 G
RX5 = 000040	SYN1 = 000002	TX6 = 000100	T\$TES= 010076	T7 022476 G
RX6 = 000100	SYN2 = 000004	TX7 = 000200	T1 021634 G	T8 022652 G
RX7 = 000200	SYN3 = 000010	T\$ARGC= 000001	T10 023224 G	T9 023106 G
R14NRW 002560	SYN4 = 000020	T\$CODE= 012032	T11 023342 G	UAM = 000200 G
SAVE4 002414	SYN5 = 000040	T\$ERRN= 000032	T12 023460 G	UNIT 002440
SAVE6 002416	SYN6 = 000100	T\$EXCP= 000000	T13 023576 G	UNRR = 000001
SAVLEN 002432	SYN7 = 000200	T\$FLAG= 000040	T14 024226 G	UPBITS 002550
SCRACH 002342	S\$LSYM= 010000	T\$GMAN= 000000	T15 024560 G	VECTOR 035646
SEC = 000020	TEOM = 000002	T\$HILI= 000007	T16 025050 G	V35 = 000020
SECA = 000020	TERR = 000200	T\$LAST= 000001	T17 025260 G	WAIT50 005146
SEFR = 000040	TEST = 000001	T\$LOLI= 000000	T18 025464 G	WAX = 000010
SELSBY= 000002	TESTMD= 000004	T\$LSYM= 010000	T19 025670 G	WAX15 002374
SEL4 002454	TIMFLG 002360	T\$LTNO= 000052	T2 021716 G	WAX16 002376
SEL6 002456	TMP0 002530	T\$NEST= 177777	T20 026126 G	WRDYTO= 000002
SETUP 010640	TMP1 002532	T\$NS0 = 000000	T20.1 026156	WRIBYT 002366
SFPTBL 002302 G	TMP2 002534	T\$NS1 = 000005	T20.2 026264	WRITAX 004312
SIG0 = 000100	TMP3 002536	T\$NS2 = 000002	T21 026360 G	WRITLU 003750
SIGR = 000200	TMP4 002540	T\$NS3 = 000003	T21.1 026372	XYZ = 000100
SOM = 000001	TMP5 002542	T\$PTNU= 000000	T22 026632 G	X\$ALWA= 000000
STALL 005164	TMP6 002544	T\$SAVL= 177777	T23 027310 G	X\$FALS= 000040
STARES 002444	TMP7 002546	T\$SEGL= 177777	T24 027430 G	X\$OFFS= 000400
STARST 021256	TSOM = 000001	T\$SEK0= 010000	T25 027540 G	X\$TRUE= 000020
STBY = 000002	TSTCON 002474	T\$SUBN= 000000	T26 027734 G	\$LSTIN= 000001
STEPLU= 000020	TSTNUM 002442	T\$TAGL= 177777	T27 030300 G	\$LSTIA= 000001
STEPMP= 000001	TXAB = 000004	T\$TAGN= 010101	T28 030510 G	

. ABS. 036234 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 21669 WORDS (85 PAGES)

DYNAMIC MEMORY AVAILABLE FOR 69 PAGES

CZDMRD.BIC,CZDMRD.SEO/C/N:TOC=SVCS4R.MLB,CZDMRD.P11

[illegible]

	12-199	12-211	12-220	12-232	12-245	12-257	12-268	12-342	15-287	15-596	15-707			
BIT10	12-20#	12-279	12-291											
BIT11	12-20#	12-278	12-290											
BIT12	12-20#													
BIT13	12-20#													
BIT14	12-20#													
BIT15	12-20#	12-322	12-324	12-325	15-368	15-522								
BIT2	12-20#	12-40	12-52	12-61	12-80	12-90	12-102	12-114	12-126	12-138	12-150	12-162	12-174	12-186
	12-198	12-210	12-219	12-231	12-244	12-256	12-267							
BIT3	12-20#	12-39	12-51	12-60	12-79	12-89	12-101	12-113	12-125	12-137	12-149	12-161	12-173	12-185
	12-197	12-209	12-218	12-230	12-243	12-255								
BIT4	12-20#	12-38	12-50	12-78	12-88	12-100	12-112	12-124	12-136	12-148	12-160	12-172	12-184	12-196
	12-208	12-229	12-242	12-254										
BIT5	12-20#	12-49	12-70	12-77	12-87	12-99	12-111	12-123	12-135	12-147	12-159	12-171	12-183	12-195
	12-207	12-228	12-241	12-253	12-266									
BIT6	12-20#	12-37	12-48	12-69	12-76	12-86	12-98	12-110	12-122	12-134	12-146	12-158	12-170	12-182
	12-194	12-206	12-227	12-240	12-252	12-265								
BIT7	12-20#	12-36	12-47	12-59	12-68	12-75	12-97	12-109	12-121	12-133	12-145	12-157	12-169	12-181
	12-193	12-205	12-217	12-226	12-239	12-251	12-264	15-851	15-852					
BIT8	12-20#	12-281	12-293											
BIT9	12-20#	12-280	12-292											
BOE	12-20#													
B POLL	12-69#													
BSEL1	13-54#	15-16*	15-18*	15-19*	15-33*	15-34*	15-39*	15-131*	15-133*	15-234*	15-372*	15-373*	15-375*	15-465*
	15-466*	15-471*	15-484*	19-59*	19-60*	19-61	24-134*	24-834*						
BSEL2	13-55#	19-61*	19-62*	24-470	24-490	24-509								
BSEL4	13-56#	15-69	15-90*	24-173*										
C\$AU	7-18#	23-10												
C\$AUTO	7-18#	20-19												
C\$BRK	7-18#													
C\$BSEG	7-18#	24-72	24-303	24-337	24-371	24-405	24-680	24-731	24-778	24-838	24-900			
C\$BSUB	7-18#	24-898	24-924	24-960										
C\$CEFG	7-18#													
C\$CLCK	7-18#													
C\$CLEA	7-18#	21-11												
C\$CLOS	7-18#													
C\$CLP1	7-18#													
C\$LVEC	7-18#													
C\$DCLN	7-18#													
C\$DODU	7-18#	20-16												
C\$DRPT	7-18#													
C\$DU	7-18#	22-13												
C\$EDIT	7-18#	8-17												
C\$ERDF	7-18#	15-277	15-283	15-293	15-299	15-406	15-412	15-570	15-602	15-608	15-618	15-624	15-659	15-665
	15-699	15-705	15-713	15-719	15-865	15-870	15-889	15-895	15-904	15-910	15-919	15-925	15-934	15-940
	15-:72	15-:78	24-24	24-50	24-84	24-115	24-148	24-184	24-226	24-278	24-314	24-348	24-382	24-416
	24-480	24-499	24-518	24-551	24-565	24-577	24-588	24-637	24-647	24-694	24-704	24-745	24-754	24-792
	24-801	24-855	24-865	24-913	24-936	24-983	24-993	24-:98	24-:21	24-<83	24-=26	24-=35	24-=47	24-=66
	24-=74	24-?97	24-a12	24-a95	24-A60	24-B41	24-B85	24-C89						
C\$ERHR	7-18#													
C\$ERRO	7-18#													
C\$ERSF	7-18#													
C\$ERSO	7-18#													
C\$ESCA	7-18#	24-227	24-279	24-552	24-566	24-578	24-589	24-638	24-648	24-695	24-746	24-793	24-856	24-984
	24-994	24-:99	24-:22											
C\$ESEG	7-18#	24-86	24-316	24-350	24-384	24-418	24-706	24-756	24-803	24-867	24-915			
C\$ESUB	7-18#	24-920	24-938	24-999										

CSETST	7-18# 24-594 24-?12 24-H60	24-27 24-655 24-?47	24-52 24-710 24-@27	24-90 24-760 24-A10	24-117 24-807 24-A73	24-150 24-872 24-B56	24-186 24-939 24-B88	24-232 24-:00 24-D34	24-286 24-;32 24-E35	24-320 24-;81 24-E87	24-354 24-<27 24-F55	24-388 24-<87 24-F87	24-422 24-=76 24-G40	24-523 24->32 24-G90
CSEXIT	7-18#													
CSGETB	7-18#													
CSGETW	7-18#													
CSGMAN	7-18#													
CSGPHR	7-18#	19-51												
CSGPLO	7-18#													
CSGPRI	7-18#													
CSINIT	7-18#	19-74												
CSINLP	7-18#													
CSMANI	7-18#													
CSMEM	7-18#													
CSMSG	7-18#	16-102	16-115	16-134	16-153	16-173	16-192	16-210	16-230	16-244				
CSOPEN	7-18#													
CSPNTB	7-18# 16-160 16-219	16-101 16-161 16-220	16-107 16-162 16-221	16-108 16-163 16-237	16-109 16-164 16-238	16-110 16-180 16-239	16-122 16-181 16-239	16-123 16-182 16-239	16-124 16-183 16-239	16-125 16-199 16-239	16-141 16-200 16-239	16-142 16-201 16-239	16-143 16-217 16-239	16-144 16-218 16-239
CSPNTF	7-18#	22-12	24-456	24-482	24-501	24-520								
CSPNTS	7-18#													
CSPNTX	7-18# 16-146 16-172 16-207 16-243	16-111 16-147 16-184 16-208	16-112 16-148 16-185 16-209	16-113 16-149 16-186 16-222	16-114 16-150 16-187 16-223	16-126 16-151 16-188 16-224	16-127 16-152 16-189 16-225	16-128 16-165 16-190 16-226	16-129 16-166 16-191 16-227	16-130 16-167 16-202 16-228	16-131 16-168 16-203 16-229	16-132 16-169 16-204 16-240	16-133 16-170 16-205 16-241	16-145 16-171 16-206 16-242
CSQIO	7-18#													
CSRDBU	7-18#													
CSREFG	7-18#	19-25	19-28	19-31	19-34									
CSRESE	7-18#	7-18#	22-10	24-133										
CSREVI	7-18#	8-17												
CSRFLA	7-18#													
CSRPT	7-18#	17-11												
CSSEFG	7-18#													
CSSPRI	7-18#	20-9	24-16											
CSSEVC	7-18#													
CSIPRI	7-18#													
C32BCC	12-253#	12-259	24-847											
C32ENB	12-256#													
CARR	12-152#													
CHPCHK	12-322#	24-:59	24-<08	24-<59	24-=17	24-=58	24->02	24->55	24-?39	24-E69	24-G21	24-H35		
CHPTYP	13-44#	15-369	15-479*	15-483*	15-523	19-13*	24-C43	24-C65						
CKDATA	15-842# 24-H45	24-D85 24-H48	24-E14 24-H51	24-F80 24-H36	24-F83	24-G24	24-G27	24-G30	24-G33	24-G36	24-G73	24-G78	24-G86	24-H42
CKTBIT	15-:57#	24-?41	24-G22	24-H36										
CRC1	12-98#	24-<53	24-A31	24-A94	24-C13	24-E57	24-F09	24-F74	24-G12	24-G63	24-H19			
CRC2	12-97#	24-<53	24-A31	24-A94	24-C13	24-E57	24-F09	24-F74	24-G12	24-G63	24-H19			
CRCCHK	12-325#	15-;12												
CRCTYO	12-246#													
CRCTY1	12-245#													
CRCTY2	12-244#													
CS	12-150#													
DDC	12-240#													
DDCMP	12-104# 24-H19	12-176#	24-;56	24-<53	24-=14	24-=55	24-=95	24-?30	24-?68	24-A31	24-D72	24-E57	24-F74	24-G63
DEVMAP	13-46#	19-40*	19-55*											

[illegible]

[illegible]

	16-243													
FMT7	16-15#	16-109	16-144	16-201	16-220	16-239								
FMT8	16-16#	16-124	16-163	16-183										
FMT9	16-17#	16-113	16-128	16-132	16-147	16-151	16-167	16-171	16-186	16-190	16-204	16-208	16-224	16-228
	16-242													
FRSTIM	13-37#	19-16	19-23*											
GSCNTO	7-18#													
GSDLM	7-18#													
GSDISP	7-18#													
GSEXCP	7-18#													
GSHILI	7-18#													
GSLOLI	7-18#													
GSNO	7-18#													
GSOFFS	7-18#	25-16	25-17	25-18	25-19									
GSOFSI	7-18#	25-16	25-17	25-18	25-19									
GSPRMA	7-18#	25-16	25-17											
GSPRMD	7-18#	25-18	25-19											
GSPRML	7-18#													
GSRADA	7-18#													
GSRADB	7-18#													
GSRADD	7-18#													
GSRADL	7-18#													
GSRADO	7-18#	25-16	25-17	25-18	25-19									
GSXFER	7-18#													
GSYES	7-18#	25-16	25-17	25-18	25-19									
GETALL	15-227#	15-275	15-281	15-291	15-297	15-404	15-410	15-568	15-600	15-606	15-616	15-622	15-657	15-663
	15-697	15-703	15-711	15-717	15-863	15-868	15-887	15-893	15-902	15-908	15-917	15-923	15-932	15-938
	15-:70	15-:76	24-575	24-586	24-635	24-645	24-692	24-702	24-743	24-752	24-790	24-799	24-853	24-863
	24-911	24-934	24-981	24-991	24-:96	24-:19	24-<81	24-=24	24-=33	24-=45	24-=64	24-=72	24-?95	24-210
	24-293	24-A58	24-B39	24-B83	24-C87									
GETPRM	19-36	19-47#	19-54											
GETREG	15-104#	15-233	24-48	24-82	24-113	24-146	24-182	24-224	24-276	24-312	24-346	24-380	24-414	24-478
	24-497	24-516	24-549	24-563										
GOAH	12-60#													
GOODAT	13-34#	15-855*	15-856*	15-877*	15-878*	16-110	16-125	16-164	16-221	24-45*	24-46*	24-80*	24-110*	24-111*
	24-143*	24-144*	24-179*	24-180*	24-221*	24-222*	24-273*	24-274*	24-310*	24-344*	24-378*	24-412*	24-476*	24-495*
	24-514*	24-572*	24-573*	24-583*	24-584*	24-632*	24-633*	24-642*	24-643*	24-690*	24-699*	24-700*	24-741*	24-750*
	24-788*	24-797*	24-850*	24-851*	24-860*	24-861*	24-908*	24-909*	24-931*	24-932*	24-978*	24-979*	24-988*	24-989*
	24-:94*	24-:17*	24-:93*	24-208*	24-218*	24-291*	24-A56*	24-B37*	24-B81*	24-C85*	24-D05*	24-D21*		
HDX	12-78#	12-148#												
HELP	7-4#	8-9	8-19	9-10	14-24									
HOE	12-20#													
ISAU	7-18#	23-9#	23-10#											
ISAUTO	7-18#	20-7#	20-19#											
ISCLN	7-18#	21-8#	21-11#											
ISDU	7-18#	22-8#	22-13#											
ISHRD	25-14#	25-21#												
ISINIT	7-18#	19-8#	19-74#											
ISMOD	7-18#	7-24	7-24#	26-37	26-37#									
ISMSG	7-18#	16-100#	16-102#	16-106#	16-115#	16-121#	16-134#	16-140#	16-153#	16-159#	16-173#	16-179#	16-192#	16-198#
	16-210#	16-216#	16-230#	16-236#	16-244#									
ISPROT	7-18#	18-8#												
ISPTAB	7-18#													
ISPWR	7-18#													
ISRPT	7-18#	17-9#	17-11#											
ISSEG	7-18#	24-14	24-39	24-67	24-72#	24-86#	24-101	24-128	24-166	24-203	24-247	24-298	24-303#	24-316#
	24-332	24-337#	24-350#	24-366	24-371#	24-384#	24-400	24-405#	24-418#	24-449	24-539	24-610	24-676	24-680#

	24-695	24-706#	24-727	24-731#	24-746	24-756#	24-774	24-778#	24-793	24-803#	24-832	24-838#	24-856	24-867#
	24-889	24-898	24-900#	24-915#	24-924	24-957	24-960	24-:22	24-:52	24-<01	24-<49	24-=07	24-=91	24->48
	24-?27	24-?63	24-@43	24-A26	24-A89	24-B69	24-C05	24-D58	24-E53	24-F05	24-F70	24-G08	24-G59	24-H13
ISSETU	7-18#													
ISSFT	26-13#	26-16#												
ISSRV	7-18#													
ISSUB	7-18#	24-14	24-39	24-67	24-101	24-128	24-166	24-203	24-247	24-298	24-332	24-366	24-400	24-449
	24-539	24-610	24-676	24-727	24-774	24-832	24-889	24-898	24-898#	24-920	24-920#	24-920#	24-924	24-924#
	24-938	24-938#	24-938#	24-957	24-960	24-960#	24-984	24-994	24-999	24-999#	24-999#	24-:22	24-:52	24-<01
	24-<49	24-=07	24-=91	24->48	24-?27	24-?63	24-@43	24-A26	24-A89	24-B69	24-C05	24-D58	24-E53	24-F05
	24-F70	24-G08	24-G59	24-H13										
ISTST	7-18#	24-14	24-14#	24-27	24-27#	24-27#	24-39	24-39#	24-52	24-52#	24-52#	24-67	24-67#	24-90
	24-90#	24-90#	24-101	24-101#	24-117	24-117#	24-117#	24-128	24-128#	24-150	24-150#	24-150#	24-166	24-166#
	24-186	24-186#	24-186#	24-203	24-203#	24-227	24-232	24-232#	24-232#	24-247	24-247#	24-279	24-286	24-286#
	24-286#	24-298	24-298#	24-320	24-320#	24-320#	24-332	24-332#	24-354	24-354#	24-354#	24-366	24-366#	24-388
	24-388#	24-388#	24-400	24-400#	24-422	24-422#	24-422#	24-449	24-449#	24-523	24-523#	24-523#	24-539	24-539#
	24-552	24-566	24-578	24-589	24-594	24-594#	24-594#	24-610	24-610#	24-638	24-648	24-655	24-655#	24-655#
	24-676	24-676#	24-710	24-710#	24-710#	24-727	24-727#	24-760	24-760#	24-760#	24-774	24-774#	24-807	24-807#
	24-807#	24-832	24-832#	24-872	24-872#	24-872#	24-889	24-889#	24-898	24-924	24-939	24-939#	24-939#	24-957
	24-957#	24-960	24-:00	24-:00#	24-:00#	24-:22	24-:22#	24-:22#	24-:99	24-:22	24-:32	24-:32#	24-:32#	24-:52
	24-:81	24-:81#	24-:81#	24-<01	24-<01#	24-<27	24-<27#	24-<27#	24-<49	24-<49#	24-<87	24-<87#	24-<87#	24-=07
	24-=07#	24-=76	24-=76#	24-=76#	24-=91	24-=91#	24->32	24->32#	24->32#	24->48	24->48#	24-?12	24-?12#	24-?12#
	24-?27	24-?27#	24-?47	24-?47#	24-?47#	24-?63	24-?63#	24-@27	24-@27#	24-@27#	24-@27#	24-@43	24-@43#	24-A10
	24-A10#	24-A26	24-A26#	24-A73	24-A73#	24-A73#	24-A89	24-A89#	24-B56	24-B56#	24-B56#	24-B69	24-B69#	24-B88
	24-B88#	24-B88#	24-C05	24-C05#	24-D34	24-D34#	24-D34#	24-D58	24-D58#	24-E35	24-E35#	24-E35#	24-E53	24-E53#
	24-E87	24-E87#	24-E87#	24-F05	24-F05#	24-F55	24-F55#	24-F55#	24-F70	24-F70#	24-F87	24-F87#	24-F87#	24-G08
	24-G08#	24-G40	24-G40#	24-G40#	24-G59	24-G59#	24-G90	24-G90#	24-G90#	24-G90#	24-H13	24-H13#	24-H60	24-H60#
1422	12-251#	12-259												
IAC	12-134#	15-655	15-661											
IACIV	15-645#	15-772	15-793	24-?83	24-?88	24-@03	24-@16	24-@24	24-@63	24-@72	24-@79	24-@84	24-A01	24-A07
	24-A46	24-A51	24-A66	24-A70	24-B09	24-B18	24-B25	24-B30	24-B47	24-B53	24-C51	24-C76	24-C97	24-D13
IBE	12-20#													
IC	12-68#	12-133#	24-A05	24-A68	24-B51									
ICIR	12-173#	15-614	15-620											
IDL	12-243#													
IDLE	12-99#	24-=55	24-=95	24-E57	24-F74	24-G63								
IDU	12-20#													
IER	12-20#													
IERR	12-103#	15-817	15-823	24-?68	24-@48	24-A31	24-A94	24-C13						
INITRN	15-436#	24-:54	24-<03	24-<51	24-=12	24-=53	24-=93	24->50	24-?28	24-C07	24-D70	24-E55	24-F07	24-F72
	24-G10	24-G61	24-H17											
INTFLG	13-21#													
INTGRL	12-255#	12-259												
IRDY	12-136#	15-598	15-604	15-783										
ISIRDY	15-588#	15-774	15-787	15-791	15-795	15-:33	15-:39	24-C63	24-C72	24-C80	24-C93	24-D01	24-D09	24-D17
	24-D64	24-D89	24-D99	24-E08	24-E20									
ISR	12-20#													
ISRUN	25-19	25-26#												
IXE	12-20#													
JSJMP	7-18#													
LSACP	8-17#													
LSAPT	8-17#													
LSAU	8-17	23-9#												
LSAUT	8-17#													
LSAUTO	8-17	20-7#												
LSCCP	8-17#													
LSCLEA	8-17	21-8#												

L\$CO	8-17#		
L\$DEPO	8-17#		
L\$DESC	8-17	14-17#	
L\$DESP	8-17#		
L\$DEVP	8-17#		
L\$DISP	8-17	9-8#	
L\$DLY	8-17#		
L\$DTP	8-17#		
L\$DTYP	8-17#		
L\$DU	8-17	22-8#	
L\$DUT	8-17#		
L\$DVTY	8-17	14-12#	
L\$EF	8-17#		
L\$ENVI	8-17#		
L\$ETP	8-17#		
L\$EXP1	8-17#		
L\$EXP4	8-17#		
L\$EXP5	8-17#		
L\$HARD	8-17	25-14	25-14#
L\$HIME	8-17#		
L\$HPCP	8-17#		
L\$HPTP	8-17#		
L\$HW	8-17	10-9	10-9#
L\$ICP	8-17#		
L\$INIT	8-17	19-8#	
L\$LADP	8-17#		
L\$LAST	8-17	26-39#	
L\$LOAD	8-17#		
L\$LUN	8-17#		
L\$MREV	8-17#		
L\$NAME	8-17#		
L\$PRIO	8-17#		
L\$PROT	8-17	18-8#	
L\$PRT	8-17#		
L\$REPP	8-17#		
L\$REV	8-17#		
L\$RPT	17-9#		
L\$SOFT	26-13	26-13#	
L\$SPC	8-17#		
L\$SPCP	8-17#		
L\$SPTP	8-17#		
L\$STA	8-17#		
L\$SW	11-8	11-8#	
L\$TEST	8-17#		
L\$TIML	8-17#		
L\$UNIT	8-17#	19-49	
L10000	10-9	10-23#	
L10001	11-8	11-11#	
L10002	16-102#		
L10003	16-115#		
L10004	16-134#		
L10005	16-153#		
L10006	16-173#		
L10007	16-192#		
L10010	16-210#		
L10011	16-230#		
L10012	16-244#		

[illegible]

LOADAT	13-36#	16-161	24-906*	24-907*															
LODMSG	15-999#	15-;04	24-G13	24-H25															
LOE	12-20#																		
LOGDEV	13-17#	19-42*	19-48*	19-49	19-51	20-16	22-12												
LOOPIN	15-130#	24-463																	
LOT	12-20#																		
LU1MOD	7-24#																		
LULoop	12-39#	15-39	15-234	15-372	15-465	24-834													
LULP	12-70#	12-135#	24-?73	24-a53	24-A36	24-A99	24-C21												
LUR10	12-301#	15-106	16-112	16-127	16-146	16-166	16-185	16-203	16-223	16-241									
LUR11	12-302#	16-112	16-127	16-146	16-166	16-185	16-203	16-223	16-241										
LUR12	12-303#	16-112	16-127	16-146	16-166	16-185	16-203	16-223	16-241										
LUR13	12-304#	16-112	16-127	16-146	16-166	16-185	16-203	16-223	16-241										
LUR14	12-305#	16-114	16-129	16-148	16-168	16-187	16-205	16-225	16-243										
LUR15	12-306#	16-114	16-129	16-148	16-168	16-187	16-205	16-225	16-243										
LUR16	12-307#	16-114	16-129	16-148	16-168	16-187	16-205	16-225	16-243										
LUR17	12-308#	16-114	16-129	16-148	16-168	16-187	16-205	16-225	16-243										
LUREG	12-301	12-302	12-303	12-304	12-305	12-306	12-307	12-308	12-309	12-310	12-311	12-312	12-313	12-314					
	12-315	12-316	13-11#																
LUSW11	13-62#																		
LUSW12	13-63#																		
LUSW13	13-64#																		
MA' -1	12-79#																		
MAINT2	12-80#																		
MCLK	12-175#	24-471	24-476	24-491	24-510	24-514													
MCLR	12-37#	15-33	15-34																
MODR	12-149#																		
MPCSR	13-53#	16-101	16-107	16-122	16-142	16-160	16-181	16-199	16-218	16-237	19-58*	20-12	24-19						
MPIVEC	13-59#	19-67*																	
MPOVEC	13-60#	19-68*	19-69*																
MPRIOR	13-61#	19-70*																	
MSG1	13-461#	15-;05	15-;07	15-;10	24-?31	24-?35	24-?44	24-?75	24-?78	24-?87	24-a01	24-a55	24-a58	24-a69					
	24-a70	24-a77	24-a99	24-A38	24-A41	24-A50	24-A64	24-B01	24-B04	24-B15	24-B16	24-B23	24-B45	24-C23					
	24-G14																		
MSG4	13-473#	24-H26																	
MSTCLR	15-30#	15-441	24-41	24-68	24-102	24-f06	24-129	24-167	24-204	24-212	24-248	24-299	24-333	24-367					
	24-401	24-450	24-522	24-540	24-556	24-611	24-677	24-728	24-775	24-833	24-890	24-927	24-961	24-;27					
	24-;80	24-<26	24-<86	24-=68	24-=75	24-?46	24-?65	24-a26	24-a45	24-A09	24-A28	24-A72	24-A91	24-B55					
	24-B70	24-B87	24-C10	24-D33	24-D63	24-E34	24-F86	24-F54	24-F86	24-G39	24-G89	24-H59							
MVIOX	12-334#	15-65	24-461																
MVIXO	12-335#	15-87																	
MVIXOX	12-336#	24-171																	
NEWT	19-32	19-41#	19-50																
OSAPTS	7-18#	8-17																	
OSAU	7-18#	8-7#	8-17																
OSBGNR	7-18#	8-17																	
OSBGNS	7-18#	8-17																	
OSDU	7-18#	8-7#	8-17																
OSERRT	7-18#	8-17																	
OSGNSW	7-18#	8-17																	
OSPOIN	7-18#	8-7	8-7#	8-7#	8-17														
OSSETU	7-18#	8-17	26-39																
OACT	12-122#	15-402	15-408	15-469															
OACTIV	15-392#	15-444	15-461	15-476	15-526	15-562	24-C32	24-C36	24-C57	24-C74	24-C95	24-D11	24-D30	24-H54					
OC	12-59#	12-121#	15-557	24-D27	24-G82														
OCOR	12-172#	15-289	15-295	15-481	24-:47														
OP	12-257#	12-259																	

ORDY	12-124#	15-273	15-279	24-269	24-270	24-:79								
OSIRDY	15-263#	15-442	15-459	15-486	15-530	15-536	15-560	24-962	24-972	24-:28	24-:37	24-:52	24-:66	24-:84
	24-:09	24-:25	24-:28	24-C24	24-C30	24-C46	24-C49	24-C59						
OVR	12-137#	15-928	15-930	15-936										
PATA	13-123#	24-70												
PATB	13-146#	24-88	24-258											
PATC	13-157#	24-301												
PATCH	26-28#													
PATD	13-163#	24-318	24-335											
PATE	13-169#	24-352	24-369											
PATF	13-178#	24-386	24-403											
PATG	13-184#	24-206	24-420											
PATH	13-195#	24-210	24-542											
PATI	13-206#	24-554	24-558	24-612	24-677	24-700	24-929	24-932						
PATJ	13-217#	24-592	24-615	24-619	24-620									
PATK	13-228#	24-678	24-776											
PATL	13-302#	24-708	24-729	24-805										
PATM	13-311#	24-43	24-46	24-108	24-111	24-141	24-144	24-177	24-180	24-214	24-250	24-758		
PATN	13-322#	24-230	24-958											
PATO	13-339#	24-891	24-996											
PATP	13-356#	24-892	24-918											
PATU	13-373#	24-836												
PATV	13-413#	24-835	24-870											
PNT	12-20#													
POLL	12-75#													
PRI	12-20#													
PRI00	12-20#													
PRI01	12-20#													
PRI02	12-20#													
PRI03	12-20#													
PRI04	12-20#													
PRI05	12-20#													
PRI06	12-20#													
PRI07	12-20#	20-9	20-11	24-16	24-18									
PRIOR	13-19#													
PRIRTY	25-18	25-25#												
PSTACK	13-18#	15-301	15-414	15-626	15-667	15-721	15-949	15-:84	19-10*					
R14NRW	13-109#	24-74	24-77											
RAB	12-138#	15-913	15-915	15-921										
RABT	12-198#													
RAX15	13-28#	15-167*	15-168*	15-238	24-569*	24-570	24-574	24-629*	24-630	24-634	24-688	24-691	24-739	24-742
	24-786	24-789	24-846*	24-847*	24-848	24-852	24-976	24-980	24-:92	24-:95	24-:15	24-:18	24-?91	24-?94
	24-206	24-209	24-220	24-289	24-292	24-A54	24-A57	24-B35	24-B38	24-B79	24-B87	24-C83	24-C86	24-D03
	24-D19													
RAX16	13-29#	15-171*	15-172*	15-240	15-695	15-701	15-709	15-715	24-581	24-585	24-640	24-644	24-697	24-701
	24-748	24-751	24-795	24-798	24-858	24-862	24-904	24-910	24-929	24-933	24-986	24-990		
RCV1ST	15-761#	24-D79	24-E72	24-F18	24-G70	24-H40								
RCVBUF	13-485#	15-:08												
RDALL	12-102#													
RDAX	12-88#	12-160#	15-154											
RDRXSI	15-737#	15-848	15-:28	24-F79										
READAX	15-148#	15-237	15-692	24-559	24-626	24-687	24-738	24-785	24-843	24-903	24-928	24-975	24-:91	24-:14
	24-?90	24-205	24-219	24-288	24-A53	24-B34	24-B78							
READLU	15-57#	15-108	15-158	15-166	15-170	15-209	15-270	15-286	15-399	15-468	15-480	15-565	15-595	15-611
	15-652	15-739	15-743	15-782	15-860	15-:65	24-42	24-76	24-107	24-139	24-176	24-217	24-265	24-306
	24-340	24-374	24-408	24-:46	24-:78	24-<78	24-=21	24-=30	24-=42	24-=61	24-=69			
READY	12-157#	15-159	15-210											

[illegible]

SSL SYM	7-18#	10-23#	11-11#	16-102#	16-115#	16-134#	16-153#	16-173#	16-192#	16-210#	16-230#	16-244#	17-11#	19-74#
	20-19#	21-11#	22-13#	23-10#	24-27#	24-52#	24-72	24-72	24-72#	24-90#	24-117#	24-150#	24-186#	24-232#
	24-286#	24-303	24-303	24-303#	24-320#	24-337	24-337	24-337#	24-354#	24-371	24-371	24-371#	24-388#	24-405
	24-405	24-405#	24-422#	24-523#	24-594#	24-655#	24-680	24-680	24-680#	24-710#	24-731	24-731	24-731#	24-760#
	24-778	24-778	24-778#	24-807#	24-838	24-838	24-838#	24-872#	24-900	24-900	24-900#	24-920#	24-938#	24-939#
	24-999#	24-:00#	24-:32#	24-:81#	24-<27#	24-<87#	24-=:76#	24->32#	24-?12#	24-?47#	24-@27#	24-A10#	24-A73#	24-B56#
	24-B88#	24-D34#	24-E35#	24-E87#	24-F55#	24-F87#	24-G40#	24-G90#	24-H60#	25-21#	26-16#			
SAVE4	13-38#	19-18*	19-21	20-17	24-25									
SAVE6	13-39#	19-19*	19-22	20-18	24-26									
SAVLEN	13-45#	15-45*	15-849	15-982*	19-15*									
SCRACH	13-16#													
SEC	12-242#													
SECA	12-100#													
SEL4	13-57#	19-63*	19-64*											
SEL6	13-58#	15-17*	15-132*	19-65*	19-66*									
SELEFR	12-77#													
SELSBY	12-81#													
SETUP	15-967#	24-?66	24-@46	24-A29	24-A92	24-C11								
SFPTBL	11-8#													
SIGQ	12-170#													
SIGR	12-169#													
SOM	12-63#													
STALL	15-330#	15-374	15-376	15-467	15-472	15-485								
STARES	13-50#	19-38*	19-43*	24-454										
STARST	19-26	19-29	19-37#											
STBY	12-151#													
STEPLU	12-38#	15-373	15-375	15-466	15-471	15-484								
STEPMP	12-42#	15-18	15-19											
STPCLK	15-15#	15-67	15-91	24-174										
STPERR	15-814#													
STPLU	15-364#	15-532	15-778	15-821	15-945	15-:35	15-:63	24-:44	24-:76	24-:12	24-<75	24-=:18	24-=:28	24-=:40
	24-=:59	24-?38	24-?81	24-?85	24-?99	24-@14	24-@22	24-@61	24-@67	24-@97	24-A44	24-A48	24-A62	24-B07
	24-B13	24-B43	24-C34	24-C55	24-C70	24-C91	24-D07	24-D94	24-E06	24-E22	24-E68	24-F77	24-G20	24-G84
	24-H34													
STR	12-241#													
STRIP	12-101#	24-=:56	24-<53	24-=:14	24-=:55	24-?30	24-?68	24-A31	24-D72	24-G12	24-H19			
SUBRPC	13-20#	15-264	15-265	15-267*	15-268*	15-305*	15-393	15-394	15-396*	15-397*	15-418*	15-439*	15-440*	15-492*
	15-512*	15-513*	15-539*	15-554*	15-555*	15-571*	15-589	15-590	15-592*	15-593*	15-630*	15-646	15-647	15-649*
	15-650*	15-671*	15-686	15-687	15-689*	15-690*	15-725*	15-764*	15-765*	15-801*	15-844*	15-845*	15-955*	15-987*
	15-:26*	15-:27*	15-:42*	15-:59*	15-:60*	15-:90*	16-141	16-180	16-217	19-11*				
SVCGBL	7-18#	7-24	7-24	7-24	7-32#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	9-8	10-9	10-9	11-8	11-8	14-12	14-17	16-100	16-106	16-121	16-140	16-159	16-179
	16-198	16-216	16-236	17-9	18-8	19-8	20-7	21-8	22-8	23-9	25-14	26-13	26-39	26-39
	26-39	26-39#												
SVCINS	7-18#	7-29#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17									

[illegible]

[illegible]

[illegible]

[illegible]

TSSHAR	25-14	25-14#	25-21											
TSSHW	10-9	10-9#	10-23											
TSSINI	19-8#	19-74												
TSSMSG	16-100#	16-102	16-106#	16-115	16-121#	16-134	16-140#	16-153	16-159#	16-173	16-179#	16-192	16-198#	16-210
	16-216#	16-230	16-236#	16-244										
TSSPRO	18-8#													
TSSRPT	17-9#	17-11												
TSSSEG	24-72	24-72#	24-86	24-86#	24-303	24-303#	24-316	24-316#	24-337	24-337#	24-350	24-350#	24-371	24-371#
	24-384	24-384#	24-405	24-405#	24-418	24-418#	24-680	24-680#	24-695	24-706	24-706#	24-731	24-731#	24-746
	24-756	24-756#	24-778	24-778#	24-793	24-803	24-803#	24-838	24-838#	24-856	24-867	24-867#	24-900	24-900#
	24-915	24-915#												
TSSSOF	26-13	26-13#	26-16											
TSSSUB	24-898#	24-920	24-924#	24-938	24-960#	24-984	24-994	24-999						
TSSSW	11-8	11-8#	11-11											
TSTES	24-14#	24-27	24-39#	24-52	24-67#	24-90	24-101#	24-117	24-128#	24-150	24-166#	24-186	24-203#	24-227
	24-232	24-247#	24-279	24-286	24-298#	24-320	24-332#	24-354	24-366#	24-388	24-400#	24-422	24-449#	24-523
	24-539#	24-552	24-566	24-578	24-589	24-594	24-60#	24-638	24-648	24-655	24-676#	24-710	24-727#	24-760
	24-774#	24-807	24-832#	24-872	24-889#	24-939	24-957#	24-:00	24-:2	24-:99	24-:22	24-:32	24-:52#	24-:81
	24-<01#	24-<27	24-<49#	24-<87	24-:=07#	24-:=76	24-:=91#	24->32	24->48#	24-?12	24-?27#	24-?47	24-?63#	24-?27
	24-243#	24-A10	24-A26#	24-A73	24-A89#	24-B56	24-B69#	24-B88	24-C05#	24-D34	24-D58#	24-E35	24-E53#	24-E87
	24-F05#	24-F55	24-F70#	24-F87	24-G08#	24-G40	24-G59#	24-G90	24-H13#	24-H60				
TSARGC	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	16-101	16-101	16-101	16-101	16-101#	16-101#	16-101#	16-101#	16-107	16-107	16-107
	16-107#	16-107#	16-107#	16-108	16-108	16-108#	16-109	16-109	16-109	16-109	16-109#	16-109#	16-109#	16-110
	16-110	16-110	16-110	16-110#	16-110#	16-110#	16-111	16-111	16-111	16-111	16-111#	16-111#	16-111#	16-112
	16-112	16-112	16-112	16-112	16-112	16-112#	16-112#	16-112#	16-112#	16-112#	16-112#	16-113	16-113	16-113#
	16-113#	16-114	16-114	16-114	16-114	16-114	16-114	16-114#	16-114#	16-114#	16-114#	16-114#	16-114#	16-122
	16-122	16-122	16-122#	16-122#	16-122#	16-123	16-123	16-123#	16-124	16-124	16-124	16-124	16-124#	16-124#
	16-124#	16-125	16-125	16-125	16-125	16-125#	16-125#	16-125#	16-126	16-126	16-126	16-126	16-126#	16-126#
	16-126#	16-127	16-127	16-127	16-127	16-127	16-127	16-127#	16-127#	16-127#	16-127#	16-127#	16-127#	16-128
	16-128	16-128#	16-128#	16-129	16-129	16-129	16-129	16-129	16-129	16-129#	16-129#	16-129#	16-129#	16-129#
	16-130	16-130	16-130	16-130	16-130#	16-130#	16-130#	16-131	16-131	16-131	16-131	16-131	16-131	16-131#
	16-131#	16-131#	16-131#	16-131#	16-132	16-132	16-132	16-132#	16-132#	16-133	16-133	16-133	16-133	16-133
	16-133	16-133#	16-133#	16-133#	16-133#	16-133#	16-141	16-141	16-141	16-141#	16-141#	16-142	16-142	16-142
	16-142	16-142#	16-142#	16-142#	16-143	16-143	16-143#	16-144	16-144	16-144	16-144	16-144#	16-144#	16-144#
	16-145	16-145	16-145	16-145	16-145#	16-145#	16-145#	16-146	16-146	16-146	16-146	16-146	16-146	16-146#
	16-146#	16-146#	16-146#	16-146#	16-147	16-147	16-147	16-147#	16-147#	16-148	16-148	16-148	16-148	16-148
	16-148	16-148#	16-148#	16-148#	16-148#	16-148#	16-149	16-149	16-149	16-149	16-149#	16-149#	16-149#	16-150
	16-150	16-150	16-150	16-150	16-150	16-150#	16-150#	16-150#	16-150#	16-150#	16-151	16-151	16-151	16-151#
	16-151#	16-152	16-152	16-152	16-152	16-152	16-152	16-152#	16-152#	16-152#	16-152#	16-152#	16-152#	16-160
	16-160	16-160	16-160#	16-160#	16-160#	16-161	16-161	16-161	16-161	16-161#	16-161#	16-161#	16-162	16-162
	16-162#	16-163	16-163	16-163	16-163	16-163#	16-163#	16-163#	16-164	16-164	16-164	16-164	16-164#	16-164#
	16-164#	16-165	16-165	16-165	16-165	16-165#	16-165#	16-165#	16-166	16-166	16-166	16-166	16-166	16-166
	16-166#	16-166#	16-166#	16-166#	16-166#	16-167	16-167	16-167	16-167#	16-167#	16-167#	16-168	16-168	16-168
	16-168	16-168	16-168#	16-168#	16-168#	16-168#	16-169	16-169	16-169	16-169	16-169	16-169#	16-169#	16-169#
	16-170	16-170	16-170	16-170	16-170	16-170	16-170#	16-170#	16-170#	16-170#	16-170#	16-171	16-171	16-171
	16-171#	16-171#	16-172	16-172	16-172	16-172	16-172	16-172#	16-172#	16-172#	16-172#	16-172#	16-172#	16-180
	16-180	16-180	16-180#	16-180#	16-181	16-181	16-181	16-181	16-181#	16-181#	16-181#	16-182	16-182	16-182#
	16-183	16-183	16-183	16-183	16-183#	16-183#	16-183#	16-184	16-184	16-184	16-184	16-184#	16-184#	16-184#
	16-185	16-185	16-185	16-185	16-185	16-185	16-185#	16-185#	16-185#	16-185#	16-185#	16-186	16-186	16-186
	16-186#	16-186#	16-187	16-187	16-187	16-187	16-187	16-187	16-187#	16-187#	16-187#	16-187#	16-187#	16-188
	16-188	16-188	16-188	16-188#	16-188#	16-188#	16-189	16-189	16-189	16-189	16-189	16-189	16-189#	16-189#
	16-189#	16-189#	16-189#	16-190	16-190	16-190	16-190#	16-190#	16-191	16-191	16-191	16-191	16-191	16-191
	16-191#	16-191#	16-191#	16-191#	16-191#	16-191#	16-199	16-199	16-199	16-199#	16-199#	16-199#	16-199#	16-200
	16-200#	16-201	16-201	16-201	16-201	16-201#	16-201#	16-201#	16-202	16-202	16-202	16-202	16-202#	16-202#
	16-202#	16-203	16-203	16-203	16-203	16-203	16-203	16-203#	16-203#	16-203#	16-203#	16-203#	16-204	16-204
	16-204	16-204#	16-204#	16-205	16-205	16-205	16-205	16-205	16-205	16-205#	16-205#	16-205#	16-205#	16-205#

CROSS REFERENCE TABLE (CREF V01-05)

	16-206	16-206	16-206	16-206	16-206#	16-206#	16-206#	16-207	16-207	16-207	16-207	16-207	16-207	16-207#
	16-207#	16-207#	16-207#	16-207#	16-208	16-208	16-208	16-208#	16-208#	16-209	16-209	16-209	16-209	16-209
	16-209	16-209#	16-209#	16-209#	16-209#	16-209#	16-209#	16-217	16-217	16-217#	16-217#	16-218	16-218	16-218
	16-218	16-218#	16-218#	16-218#	16-219	16-219	16-219#	16-220	16-220	16-220	16-220	16-220#	16-220#	16-220#
	16-221	16-221	16-221	16-221	16-221#	16-221#	16-221#	16-222	16-222	16-222	16-222	16-222#	16-222#	16-222#
	16-223	16-223	16-223	16-223	16-223	16-223	16-223#	16-223#	16-223#	16-223#	16-223#	16-224	16-224	16-224
	16-224#	16-224#	16-225	16-225	16-225	16-225	16-225	16-225	16-225#	16-225#	16-225#	16-225#	16-225#	16-226
	16-226	16-226	16-226	16-226#	16-226#	16-226#	16-227	16-227	16-227	16-227	16-227	16-227	16-227	16-227#
	16-227#	16-227#	16-227#	16-228	16-228	16-228	16-228#	16-228#	16-229	16-229	16-229	16-229	16-229	16-229
	16-229#	16-229#	16-229#	16-229#	16-229#	16-237	16-237	16-237	16-237	16-237#	16-237#	16-237#	16-238	16-238
	16-238#	16-239	16-239	16-239	16-239	16-239#	16-239#	16-239#	16-240	16-240	16-240	16-240	16-240#	16-240#
	16-240#	16-241	16-241	16-241	16-241	16-241	16-241	16-241#	16-241#	16-241#	16-241#	16-241#	16-242	16-242
	16-242	16-242#	16-242#	16-243	16-243	16-243	16-243	16-243	16-243	16-243#	16-243#	16-243#	16-243#	16-243#
	22-12	22-12	22-12	22-12#	22-12#	24-456	24-456	24-456	24-456#	24-456#	24-482	24-482	24-482#	24-501
	24-501	24-501#	24-520	24-520	24-520#									
TS CODE	25-16	25-16	25-16	25-16#	25-16#	25-16#	25-17	25-17	25-17	25-17#	25-17#	25-17#	25-18	25-18
TSERRN	25-18	25-18#	25-18#	25-18#	25-19	25-19	25-19	25-19#	25-19#	25-19#	25-19#	25-19#	25-19#	25-19#
	7-18#	15-277	15-277#	15-283	15-283#	15-293	15-293#	15-299	15-299#	15-406	15-406#	15-412	15-412#	15-570
	15-570#	15-602	15-602#	15-608	15-608#	15-618	15-618#	15-624	15-624#	15-659	15-659#	15-665	15-665#	15-699
	15-699#	15-705	15-705#	15-713	15-713#	15-719	15-719#	15-865	15-865#	15-870	15-870#	15-889	15-889#	15-895
	15-75#	15-904	15-904#	15-910	15-910#	15-919	15-919#	15-925	15-925#	15-934	15-934#	15-940	15-940#	15-972
	15-72#	15-78	15-78#	24-24	24-24#	24-50	24-50#	24-84	24-84#	24-115	24-115#	24-148	24-148#	24-184
	24-184#	24-226	24-226#	24-278	24-278#	24-314	24-314#	24-348	24-348#	24-382	24-382#	24-416	24-416#	24-480
	24-480#	24-499	24-499#	24-518	24-518#	24-551	24-551#	24-565	24-565#	24-577	24-577#	24-588	24-588#	24-637
	24-637#	24-647	24-647#	24-694	24-694#	24-704	24-704#	24-745	24-745#	24-754	24-754#	24-792	24-792#	24-801
	24-801#	24-855	24-855#	24-865	24-865#	24-913	24-913#	24-936	24-936#	24-983	24-983#	24-993	24-993#	24-998
	24-98#	24-21	24-21#	24-83	24-83#	24-26	24-26#	24-35	24-35#	24-47	24-47#	24-66	24-66#	24-74
	24-74#	24-97	24-97#	24-12	24-12#	24-95	24-95#	24-A60	24-A60#	24-B41	24-B41#	24-B85	24-B85#	24-C89
	24-C89#													
TSEXCP	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#						
TSFLAG	24-227	24-227#	24-227#	24-279	24-279#	24-279#	24-552	24-552#	24-552#	24-566	24-566#	24-566#	24-578	24-578#
	24-578#	24-589	24-589#	24-638	24-638#	24-638#	24-648	24-648#	24-648#	24-695	24-695#	24-695#	24-746	24-746#
	24-746#	24-746#	24-793	24-793#	24-793#	24-856	24-856#	24-856#	24-984	24-984#	24-984#	24-994	24-994#	24-994#
	24-99	24-99#	24-99#	24-22	24-22#	24-22#								
TSGMAN	7-18#													
TSNILI	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#						
TSLAST	7-18#	26-39#												
TSLOLI	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#						
TSLSYM	7-18	7-18#	10-23	11-11	16-102	16-115	16-134	16-153	16-173	16-192	16-210	16-230	16-244	17-11
	19-74	20-19	21-11	22-13	23-10	24-27	24-52	24-90	24-117	24-150	24-186	24-232	24-286	24-320
	24-354	24-388	24-422	24-523	24-594	24-655	24-710	24-760	24-807	24-872	24-920	24-938	24-939	24-999
	24-00	24-32	24-81	24-27	24-87	24-76	24-32	24-12	24-747	24-27	24-A10	24-A73	24-B56	24-888
	24-034	24-E35	24-E87	24-F55	24-F87	24-G40	24-G90	24-H60	25-21	26-16				
TSLTNO	26-39#													
TSNEST	7-18#	7-24	7-24	7-24#	10-9	10-9	10-9#	10-23	10-23	10-23	10-23#	11-8	11-8	11-8#
	11-11	11-11	11-11	11-11#	16-100	16-100	16-100#	16-102	16-102	16-102	16-102#	16-106	16-106	16-106#
	16-115	16-115	16-115	16-115#	16-121	16-121	16-121#	16-134	16-134	16-134	16-134#	16-140	16-140	16-140#
	16-153	16-153	16-153	16-153#	16-159	16-159	16-159#	16-173	16-173	16-173	16-173#	16-179	16-179	16-179#
	16-192	16-192	16-192	16-192#	16-198	16-198	16-198#	16-210	16-210	16-210	16-210#	16-216	16-216	16-216#
	16-230	16-230	16-230	16-230#	16-236	16-236	16-236#	16-244	16-244	16-244	16-244#	17-9	17-9	17-9#
	17-11	17-11	17-11	17-11#	18-8	18-8	18-8#	18-12	18-12	18-12	18-12#	19-8	19-8	19-8#
	19-74	19-74	19-74	19-74#	20-7	20-7	20-7#	20-19	20-19	20-19	20-19#	21-8	21-8	21-8#
	21-11	21-11	21-11	21-11#	22-8	22-8	22-8#	22-13	22-13	22-13	22-13#	23-9	23-9	23-9#
	23-10	23-10	23-10	23-10#	24-14	24-14	24-14#	24-27	24-27	24-27	24-27#	24-39	24-39	24-39#
	24-52	24-52	24-52	24-52#	24-67	24-67	24-67#	24-72	24-72	24-72	24-72#	24-86	24-86	24-86#
	24-90	24-90	24-90	24-90#	24-101	24-101	24-101#	24-117	24-117	24-117	24-117#	24-128	24-128	24-128#
	24-150	24-150	24-150	24-150#	24-166	24-166	24-166#	24-186	24-186	24-186	24-186#	24-203	24-203	24-203#

	24-232	24-232	24-232	24-232#	24-247	24-247	24-247#	24-286	24-286	24-286	24-286#	24-298	24-298	24-298#
	24-303	24-303	24-303#	24-316	24-316	24-316	24-316#	24-320	24-320	24-320	24-320#	24-332	24-332	24-332#
	24-337	24-337	24-337#	24-350	24-350	24-350	24-350#	24-354	24-354	24-354	24-354#	24-366	24-366	24-366#
	24-371	24-371	24-371#	24-384	24-384	24-384	24-384#	24-388	24-388	24-388	24-388#	24-400	24-400	24-400#
	24-405	24-405	24-405#	24-418	24-418	24-418	24-418#	24-422	24-422	24-422	24-422#	24-449	24-449	24-449#
	24-523	24-523	24-523	24-523#	24-539	24-539	24-539#	24-594	24-594	24-594	24-594#	24-610	24-610	24-610#
	24-655	24-655	24-655	24-655#	24-676	24-676	24-676#	24-680	24-680	24-680#	24-706	24-706	24-706	24-706#
	24-710	24-710	24-710	24-710#	24-727	24-727	24-727#	24-731	24-731	24-731#	24-756	24-756	24-756	24-756#
	24-760	24-760	24-760	24-760#	24-774	24-774	24-774#	24-778	24-778	24-778#	24-803	24-803	24-803	24-803#
	24-807	24-807	24-807	24-807#	24-832	24-832	24-832#	24-838	24-838	24-838#	24-867	24-867	24-867	24-867#
	24-872	24-872	24-872	24-872#	24-889	24-889	24-889#	24-898	24-898	24-898#	24-900	24-900	24-900#	24-915
	24-915	24-915	24-915#	24-920	24-920	24-920	24-920#	24-924	24-924	24-924#	24-938	24-938	24-938	24-938#
	24-939	24-939	24-939	24-939#	24-957	24-957	24-957#	24-960	24-960	24-960#	24-999	24-999	24-999	24-999#
	24-:00	24-:00	24-:00	24-:00#	24-:22	24-:22	24-:22#	24-:32	24-:32	24-:32	24-:32#	24-:52	24-:52	24-:52#
	24-:81	24-:81	24-:81	24-:81#	24-<01	24-<01	24-<01#	24-<27	24-<27	24-<27	24-<27#	24-<49	24-<49	24-<49#
	24-<87	24-<87	24-<87	24-<87#	24-=07	24-=07	24-=07#	24-=76	24-=76	24-=76	24-=76#	24-=91	24-=91	24-=91#
	24->32	24->32	24->32	24->32#	24->48	24->48	24->48#	24-?12	24-?12	24-?12	24-?12#	24-?27	24-?27	24-?27#
	24-?47	24-?47	24-?47	24-?47#	24-?63	24-?63	24-?63#	24-@27	24-@27	24-@27	24-@27#	24-@43	24-@43	24-@43#
	24-A10	24-A10	24-A10	24-A10#	24-A26	24-A26	24-A26#	24-A73	24-A73	24-A73	24-A73#	24-A89	24-A89	24-A89#
	24-B56	24-B56	24-B56	24-B56#	24-B69	24-B69	24-B69#	24-B88	24-B88	24-B88	24-B88#	24-C05	24-C05	24-C05#
	24-D34	24-D34	24-D34	24-D34#	24-D58	24-D58	24-D58#	24-E35	24-E35	24-E35	24-E35#	24-E53	24-E53	24-E53#
	24-E87	24-E87	24-E87	24-E87#	24-F05	24-F05	24-F05#	24-F55	24-F55	24-F55	24-F55#	24-F70	24-F70	24-F70#
	24-F87	24-F87	24-F87	24-F87#	24-G08	24-G08	24-G08#	24-G40	24-G40	24-G40	24-G40#	24-G59	24-G59	24-G59#
	24-G90	24-G90	24-G90	24-G90#	24-H13	24-H13	24-H13#	24-H60	24-H60	24-H60	24-H60#	25-14	25-14	25-14#
	25-21	25-21	25-21	25-21#	26-13	26-13	26-13#	26-16	26-16	26-16	26-16#	26-37	26-37	26-37
	26-37#													
TSNS0	7-24#	26-37												
TSNS1	10-9#	10-23	11-8#	11-11	16-100#	16-102	16-106#	16-115	16-121#	16-134	16-140#	16-153	16-159#	16-173
	16-179#	16-192	16-198#	16-210	16-216#	16-230	16-236#	16-244	17-9#	17-11	18-8#	18-12	19-8#	19-74
	20-7#	20-19	21-8#	21-11	22-8#	22-13	23-9#	23-10	24-14#	24-27	24-39#	24-52	24-67#	24-90
	24-101#	24-117	24-128#	24-150	24-166#	24-186	24-203#	24-232	24-247#	24-286	24-298#	24-320	24-332#	24-354
	24-366#	24-388	24-400#	24-422	24-449#	24-523	24-539#	24-594	24-610#	24-655	24-676#	24-710	24-727#	24-760
	24-774#	24-807	24-832#	24-872	24-889#	24-939	24-957#	24-:00	24-:22#	24-:32	24-:52#	24-:81	24-<01#	24-<27
	24-<49#	24-<87	24-=07#	24-=76	24-=91#	24->32	24->48#	24-?12	24-?27#	24-?47	24-?63#	24-@27	24-@43#	24-A10
	24-A26#	24-A73	24-A89#	24-B56	24-B69#	24-B88	24-C05#	24-D34	24-D58#	24-E35	24-E53#	24-E87	24-F05#	24-F55
	24-F70#	24-F87	24-G08#	24-G40	24-G59#	24-G90	24-H13#	24-H60	25-14#	25-21	26-13#	26-16		
TSNS2	24-72#	24-86	24-303#	24-316	24-337#	24-350	24-371#	24-384	24-405#	24-418	24-680#	24-706	24-731#	24-756
	24-778#	24-803	24-838#	24-867	24-898#	24-920	24-924#	24-938	24-960#	24-999				
TSNS3	24-900#	24-915												
TSPNU	7-18#													
TSSAVL	7-18#													
TSSEGL	7-18#	24-72	24-72	24-72#	24-86	24-86	24-86	24-86	24-86#	24-303	24-303	24-303#	24-316	24-316
	24-316	24-316	24-316#	24-337	24-337	24-337#	24-350	24-350	24-350	24-350	24-350#	24-371	24-371	24-371#
	24-384	24-384	24-384	24-384	24-384#	24-405	24-405	24-405#	24-418	24-418	24-418	24-418#	24-418#	24-680
	24-680	24-680#	24-695	24-706	24-706	24-706	24-706	24-706#	24-731	24-731	24-731#	24-746	24-756	24-756
	24-756	24-756	24-756#	24-778	24-778	24-778#	24-793	24-803	24-803	24-803	24-803	24-803#	24-838	24-838
	24-838#	24-856	24-867	24-867	24-867	24-867	24-867#	24-900	24-900	24-900#	24-915	24-915	24-915	24-915
	24-915#													
TSSEKO	24-72#	24-86	24-303#	24-316	24-337#	24-350	24-371#	24-384	24-405#	24-418	24-680#	24-695	24-706	24-731#
	24-746	24-756	24-778#	24-793	24-803	24-838#	24-856	24-867	24-900#	24-915				
TSSUBN	7-18#	24-14#	24-39#	24-67#	24-101#	24-128#	24-166#	24-203#	24-247#	24-298#	24-332#	24-366#	24-400#	24-449#
	24-539#	24-610#	24-676#	24-727#	24-774#	24-832#	24-889#	24-898	24-898	24-898#	24-924	24-924	24-924#	24-957#
	24-960	24-960	24-960#	24-:22#	24-:52#	24-<01#	24-<49#	24-=07#	24-=91#	24->48#	24-?27#	24-?63#	24-@43#	24-A26#
	24-A89#	24-B69#	24-C05#	24-D58#	24-E53#	24-F05#	24-F70#	24-G08#	24-G59#	24-H13#				
TSTAGL	7-18#													
TSTAGN	7-18#	10-9	10-9	10-9#	11-8	11-8	11-8#	16-100	16-100	16-100#	16-106	16-106	16-106#	16-121
	16-121	16-121#	16-140	16-140	16-140#	16-159	16-159	16-159#	16-179	16-179	16-179#	16-198	16-198	16-198#

	16-216	16-216	16-216#	16-236	16-236	16-236#	17-9	17-9	17-9#	18-8	18-8	18-8#	19-8	19-8
	19-8#	20-7	20-7	20-7#	21-8	21-8	21-8#	22-8	22-8	22-8#	23-9	23-9	23-9#	24-14
	24-14	24-14#	24-39	24-39	24-39#	24-67	24-67	24-67#	24-101	24-101	24-101#	24-128	24-128	24-128#
	24-166	24-166	24-166#	24-203	24-203	24-203#	24-247	24-247	24-247#	24-298	24-298	24-298#	24-332	24-332
	24-332#	24-366	24-366	24-366#	24-400	24-400	24-400#	24-449	24-449	24-449#	24-539	24-539	24-539#	24-610
	24-610	24-610#	24-676	24-676	24-676#	24-727	24-727	24-727#	24-774	24-774	24-774#	24-832	24-832	24-832#
	24-889	24-889	24-889#	24-898	24-898	24-898#	24-924	24-924	24-924#	24-957	24-957	24-957#	24-960	24-960
	24-960#	24-:22	24-:22	24-:22#	24-:52	24-:52	24-:52#	24-<01	24-<01	24-<01#	24-<49	24-<49	24-<49#	24-=07
	24-=07	24-=07#	24-=91	24-=91	24-=91#	24->48	24->48	24->48#	24-?27	24-?27	24-?27#	24-?63	24-?63	24-?63#
	24-@43	24-@43	24-@43#	24-A26	24-A26	24-A26#	24-A89	24-A89	24-A89#	24-B69	24-B69	24-B69#	24-C05	24-C05
	24-C05#	24-D58	24-D58	24-D58#	24-E53	24-E53	24-E53#	24-F05	24-F05	24-F05#	24-F70	24-F70	24-F70#	24-G08
	24-G08	24-G08#	24-G59	24-G59	24-G59#	24-H13	24-H13	24-H13#	25-14	25-14	25-14#	26-13	26-13	26-13#
TS TEMP	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#
	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#
	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#	9-8#
	9-8#	10-23	10-23#	11-11	11-11#	16-102	16-102#	16-115	16-115#	16-134	16-134#	16-153	16-153#	16-173
	16-173#	16-192	16-192#	16-210	16-210#	16-230	16-230#	16-244	16-244#	17-11	17-11#	18-12	18-12#	19-74
	19-74#	20-19	20-19#	21-11	21-11#	22-13	22-13#	23-10	23-10#	24-27	24-27#	24-52	24-52#	24-86
	24-86#	24-90	24-90#	24-117	24-117#	24-150	24-150#	24-186	24-186#	24-227	24-227#	24-232	24-232#	24-279
	24-279#	24-286	24-286#	24-316	24-316#	24-320	24-320#	24-350	24-350#	24-354	24-354#	24-384	24-384#	24-388
	24-388#	24-418	24-418#	24-422	24-422#	24-523	24-523#	24-552	24-552#	24-566	24-566#	24-578	24-578#	24-589
	24-589#	24-594	24-594#	24-638	24-638#	24-648	24-648#	24-655	24-655#	24-695	24-695#	24-695#	24-706	24-706#
	24-710	24-710#	24-746	24-746#	24-746#	24-756	24-756#	24-760	24-760#	24-793	24-793#	24-793#	24-803	24-803#
	24-807	24-807#	24-856	24-856#	24-856#	24-867	24-867#	24-872	24-872#	24-915	24-915#	24-920	24-920#	24-938
	24-938#	24-939	24-939#	24-984	24-984#	24-994	24-994#	24-999	24-999#	24-:00	24-:00#	24-:99	24-:99#	24-:22
	24-:22#	24-:32	24-:32#	24-:81	24-:81#	24-<27	24-<27#	24-<87	24-<87#	24-=76	24-=76#	24->32	24->32#	24-?12
	24-?12#	24-?47	24-?47#	24-@27	24-@27#	24-A10	24-A10#	24-A73	24-A73#	24-B56	24-B56#	24-B88	24-B88#	24-D34
	24-D34#	24-E35	24-E35#	24-E87	24-E87#	24-F55	24-F55#	24-F87	24-F87#	24-G40	24-G40#	24-G90	24-G90#	24-H60
	24-H60#	25-16	25-16	25-16	25-16#	25-16#	25-16#	25-17	25-17	25-17	25-17#	25-17#	25-17#	25-18
	25-18	25-18	25-18#	25-18#	25-18#	25-19	25-19	25-19	25-19#	25-19#	25-19#	25-21	25-21#	26-16
	26-16#	26-37	26-37#											
TS TEST	7-18#	24-14	24-14	24-14#	24-39	24-39	24-39#	24-67	24-67	24-67#	24-101	24-101	24-101#	24-128
	24-128	24-128#	24-166	24-166	24-166#	24-203	24-203	24-203#	24-247	24-247	24-247#	24-298	24-298	24-298#
	24-332	24-332	24-332#	24-366	24-366	24-366#	24-400	24-400	24-400#	24-449	24-449	24-449#	24-539	24-539
	24-539#	24-610	24-610	24-610#	24-676	24-676	24-676#	24-727	24-727	24-727#	24-774	24-774	24-774#	24-832
	24-832	24-832#	24-889	24-889	24-889#	24-898	24-898	24-924	24-924	24-957	24-957	24-957#	24-960	24-960
	24-:52	24-:52	24-:52#	24-<01	24-<01	24-<01#	24-<49	24-<49	24-<49#	24-=07	24-=07	24-=07#	24-=91	24-=91
	24-=91#	24->48	24->48	24->48#	24-?27	24-?27	24-?27#	24-?63	24-?63	24-?63#	24-@43	24-@43	24-@43#	24-A26
	24-A26	24-A26#	24-A89	24-A89	24-A89#	24-B69	24-B69	24-B69#	24-B69#	24-C05	24-C05	24-C05#	24-D58	24-D58
	24-E53	24-E53	24-E53#	24-F05	24-F05	24-F05#	24-F70	24-F70	24-F70#	24-F70#	24-G08	24-G08	24-G08#	24-G59
	24-G59#	24-H13	24-H13	24-H13#	26-39									
TS STM	7-18#	15-277	15-283	15-293	15-299	15-406	15-412	15-570	15-602	15-608	15-618	15-624	15-659	15-665
	15-699	15-705	15-713	15-719	15-865	15-870	15-889	15-895	15-904	15-910	15-919	15-925	15-934	15-940
	15-:72	15-:78	16-101	16-102	16-107	16-108	16-109	16-110	16-111	16-112	16-113	16-114	16-115	16-122
	16-123	16-124	16-125	16-126	16-127	16-128	16-129	16-130	16-131	16-132	16-133	16-134	16-141	16-142
	16-143	16-144	16-145	16-146	16-147	16-148	16-149	16-150	16-151	16-152	16-153	16-160	16-161	16-162
	16-163	16-164	16-165	16-166	16-167	16-168	16-169	16-170	16-171	16-172	16-173	16-180	16-181	16-182
	16-183	16-184	16-185	16-186	16-187	16-188	16-189	16-190	16-191	16-192	16-199	16-200	16-201	16-202
	16-203	16-204	16-205	16-206	16-207	16-208	16-209	16-210	16-217	16-218	16-219	16-220	16-221	16-222
	16-223	16-224	16-225	16-226	16-227	16-228	16-229	16-230	16-237	16-238	16-239	16-240	16-241	16-242
	16-243	16-244	17-11	19-25	19-28	19-31	19-34	19-51	19-74	20-9	20-16	20-19	21-11	22-10

	22-12	22-13	23-10	24-16	24-24	24-27	24-50	24-52	24-72	24-84	24-86	24-90	24-115	24-117
	24-133	24-148	24-150	24-184	24-186	24-226	24-227	24-232	24-278	24-279	24-286	24-303	24-314	24-316
	24-320	24-337	24-348	24-350	24-354	24-371	24-382	24-384	24-388	24-405	24-416	24-418	24-422	24-456
	24-480	24-482	24-499	24-501	24-518	24-520	24-523	24-551	24-552	24-565	24-566	24-577	24-578	24-588
	24-589	24-594	24-637	24-638	24-647	24-648	24-655	24-680	24-694	24-695	24-704	24-706	24-710	24-731
	24-745	24-746	24-754	24-756	24-760	24-778	24-792	24-793	24-801	24-803	24-807	24-838	24-855	24-856
	24-865	24-867	24-872	24-898	24-900	24-913	24-915	24-920	24-924	24-936	24-938	24-939	24-960	24-983
	24-984	24-993	24-994	24-999	24-:00	24-:98	24-:99	24-:21	24-:22	24-:32	24-:81	24-<21	24-<83	24-<87
	24-=26	24-=35	24-=47	24-=66	24-=74	24-=76	24->32	24-?12	24-?47	24-?97	24-a12	24-a27	24-a95	24-A10
	24-A60	24-A73	24-B41	24-B56	24-B85	24-B88	24-C89	24-D34	24-E35	24-E87	24-F55	24-F87	24-G40	24-G90
	24-H60													
T&TSTS	7-18#	24-14#	24-39#	24-67#	24-101#	24-128#	24-166#	24-203#	24-247#	24-298#	24-332#	24-366#	24-400#	24-449#
	24-539#	24-610#	24-676#	24-727#	24-774#	24-832#	24-889#	24-957#	24-:22#	24-:52#	24-<01#	24-<49#	24-=07#	24-=91#
	24->48#	24-?27#	24-?63#	24-a43#	24-A26#	24-A89#	24-B69#	24-C05#	24-D58#	24-E53#	24-F05#	24-F70#	24-G08#	24-G59#
	24-H13#													
T1	9-8	24-14#												
T10	9-8	24-332#												
T11	9-8	24-366#												
T12	9-8	24-400#												
T13	9-8	24-449#												
T14	9-8	24-539#												
T15	9-8	24-610#												
T16	9-8	24-676#												
T17	9-8	24-727#												
T18	9-8	24-774#												
T19	9-8	24-832#												
T2	9-8	24-39#												
T20	9-8	24-889#												
T20.1	24-898#													
T20.2	24-924#													
T21	9-8	24-957#												
T21.1	24-960#													
T22	9-8	24-:22#												
T23	9-8	24-:52#												
T24	9-8	24-<01#												
T25	9-8	24-<49#												
T26	9-8	24-=07#												
T27	9-8	24-=91#												
T28	9-8	24->48#												
T29	9-8	24-?27#												
T3	9-8	24-67#												
T30	9-8	24-?63#												
T31	9-8	24-a43#												
T32	9-8	24-A26#												
T33	9-8	24-A89#												
T34	9-8	24-B69#												
T35	9-8	24-C05#												
T36	9-8	24-D58#												
T37	9-8	24-E53#												
T38	9-8	24-F05#												
T39	9-8	24-F70#												
T4	9-8	24-101#												
T40	9-8	24-G08#												
T41	9-8	24-G59#												
T42	9-8	24-H13#												
T5	9-8	24-128#												
T6	9-8	24-166#												

T7	9-8	24-203#													
T8	9-8	24-247#													
T9	9-8	24-298#													
TEOM	12-220#	24-C45	24-C68												
TERR	12-217#														
TEST	12-258#	24-844													
TESTMD	12-174#														
TIMFLG	13-24#														
TMP0	13-89#	15-229*	15-232*	16-124	16-163	16-183									
TMP1	13-90#	15-247*	15-248*	16-124	16-163	16-183									
TMP2	13-91#														
TMP3	13-92#														
TMP4	13-93#														
TMP5	13-94#														
TMP6	13-95#														
TMP7	13-96#														
TSOM	12-221#														
TSTCON	13-65#														
TSTNUM	13-49#	24-451*	24-456												
TX0	12-54#	12-212#													
TX1	12-53#	12-211#													
TX2	12-52#	12-210#													
TX3	12-51#	12-209#													
TX4	12-50#	12-208#													
TX5	12-49#	12-207#													
TX6	12-48#	12-206#													
TX7	12-47#	12-205#													
TXAB	12-219#														
TXABT	12-279#														
TXCHAR	15-510#	24-:57	24-:60	24-:63	24-:66	24-:69	24-:72	24-:75	24-<06	24-<09	24-<12	24-<15	24-<18	24-<21	
	24-<57	24-<60	24-<63	24-<66	24-<69	24-<72	24-=15	24-=56	24->00	24->03	24->08	24->13	24->18	24->21	
	24->24	24->27	24->53	24->56	24->59	24->66	24->71	24->76	24->81	24->86	24->91	24->96	24-?01	24-?04	
	24-?07														
TXDATA	12-171#	15-:68	15-:74												
TXEN	12-86#	12-158#	24-140	24-<55	24-C17	24-C19									
TXEOM	12-280#	13-468	13-469	13-470	13-471	13-477	13-478	24-:64	24-:67	24-:70	24-:73	24-:76	24-<13	24-<16	
	24-<19	24-<22	24-<64	24-<67	24-<70	24-<73	24->19	24->22	24->25	24->28	24-?02	24-?05	24-?08	24-E30	
	24-G16	24-H28													
TXGA	12-218#														
TXGOA	12-278#	24-G16													
TXLENO	12-266#	15-849	24-=98	24->11	24->64	24->74	24->84	24->94							
TXLEN1	12-265#	15-849	24->06	24->11	24->69	24->74	24->89	24->94							
TXLEN2	12-264#	15-849	24-=98	24->06	24->11	24->79	24->84	24->89	24->94						
TXSOM	12-281#	13-461	13-462	15-448	24-:33	24-=37	24-H20								
TXWORD	13-41#	15-344*	15-346	15-349	15-448*	15-449*	15-514*	15-:05*	24-:33*	24-:58*	24-:06*	24-=37*	24-?33*	24-?76*	
	24-256*	24-A39*	24-B02*	24-B75*	24-D74*	24-D96*	24-E26*	24-E30*	24-E59*	24-F11*	24-F75*	24-G16*	24-G65*	24-H20*	
	24-H28*														
UAM	12-20#														
UNIT	13-48#														
UNRR	12-128#	15-861	24-=22	24-=31	24-=43	24-=62	24-=70								
UPBITS	13-99#	24-215	24-260	24-266	24-307	24-341	24-375	24-409							
V35	12-254#	12-259													
VECTOR	25-17	25-24#													
WAIT50	15-316#	15-458	15-516	15-559	15-780	15-:09	15-:29	24-971	24-:36	24-:60	24-=39	24-?37	24-?80	24-260	
	24-A43	24-B06	24-D29	24-E63	24-F15	24-G19	24-G69	24-H31							
WAX	12-89#	12-161#	15-205												
WAX15	13-30#	15-195*	15-196	15-447*	15-449	15-970*	15-979*	24-543*	24-619*	24-621	24-682*	24-684*	24-688	24-690	

[illegible]

[illegible]

ENDSFT	1-568#	7-18#	26-16											
ENDSRV	1-580#	7-18#												
ENDSUB	1-596#	7-18#	24-920	24-938	24-999									
ENDSW	1-614#	7-18#	11-11											
ENDTST	1-624#	7-18#	24-27	24-52	24-90	24-117	24-150	24-186	24-232	24-286	24-320	24-354	24-388	24-422
	24-523	24-594	24-655	24-710	24-760	24-807	24-872	24-939	24-:00	24-:32	24-:81	24-<27	24-<87	24-=76
	24->32	24-?12	24-?47	24-a27	24-A10	24-A73	24-B56	24-B88	24-D34	24-E35	24-E87	24-F55	24-F87	24-G40
	24-G90	24-H60												
EQUALS	1-642#	7-18#	12-20											
ERRDF	1-714#	7-18#	15-277	15-283	15-293	15-299	15-406	15-412	15-570	15-602	15-608	15-618	15-624	15-659
	15-665	15-699	15-705	15-713	15-719	15-865	15-870	15-889	15-895	15-904	15-910	15-919	15-925	15-934
	15-940	15-:72	15-:78	24-24	24-50	24-84	24-115	24-148	24-184	24-226	24-278	24-314	24-348	24-382
	24-416	24-480	24-499	24-518	24-551	24-565	24-577	24-588	24-637	24-647	24-694	24-704	24-745	24-754
	24-792	24-801	24-855	24-865	24-913	24-936	24-983	24-993	24-:98	24-:21	24-<83	24-=26	24-=35	24-=47
	24-=66	24-=74	24-?97	24-a12	24-a95	24-A60	24-B41	24-B85	24-C89					
ERRHRD	1-718#	7-18#												
ERROR	1-722#	7-18#												
ERRSF	1-726#	7-18#												
ERRSOF	1-730#	7-18#												
ERRTBL	1-734#	7-18#												
ESCAPE	1-744#	7-18#	24-227	24-279	24-552	24-566	24-578	24-589	24-638	24-648	24-695	24-746	24-793	24-856
	24-984	24-994	24-:99	24-:22										
EXIT	1-771#	7-18#												
FEQUAL	1-810#	7-18#												
GETBYT	1-824#	7-18#												
GETPRI	1-834#	7-18#												
GETWOR	1-829#	7-18#												
GMANIA	1-839#	7-18#												
GMANID	1-848#	7-18#												
GMANIL	1-859#	7-18#												
GPHARD	1-868#	7-18#	19-51											
GPRMA	1-874#	7-18#	25-16	25-17										
GPRMD	1-903#	7-18#	25-18	25-19										
GPRML	1-934#	7-18#												
HEADER	1-954#	7-18#	8-17											
INLOOP	1-962#	7-18#												
IOSETU	1-966#	7-18#												
IOSTAR	1-974#	7-18#												
KT11	1-982#	7-18#												
LASTAD	1-:47#	7-18#	26-39											
MSBYTE	1-D00#	7-18#	8-17	8-17	8-17	8-17#								
MSCHEC	1-E18#	7-18#												
MSCNTO	1-E82#	7-18#	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#				
MSCOUN	1-D66#	7-18#	16-101	16-101	16-101#	16-107	16-107	16-107#	16-108	16-108#	16-109	16-109	16-109#	16-110
	16-110	16-110#	16-111	16-111	16-111#	16-112	16-112	16-112#	16-112	16-112#	16-113	16-113#	16-114	16-114
	16-114	16-114	16-114#	16-122	16-122	16-122#	16-123	16-123#	16-124	16-124#	16-124	16-125	16-125	16-125#
	16-126	16-126	16-126#	16-127	16-127	16-127#	16-127	16-127#	16-128	16-128#	16-129	16-129	16-129	16-129
	16-129#	16-130	16-130	16-130#	16-131	16-131	16-131	16-131#	16-131	16-131#	16-132	16-132#	16-133	16-133
	16-133	16-133#	16-141	16-141#	16-142	16-142	16-142#	16-143	16-143#	16-144	16-144	16-144#	16-145	16-145
	16-145#	16-146	16-146	16-146	16-146	16-146#	16-147	16-147#	16-148	16-148	16-148	16-148	16-148#	16-149
	16-149	16-149#	16-150	16-150	16-150	16-150	16-150#	16-151	16-151#	16-152	16-152	16-152	16-152	16-152#
	16-160	16-160	16-160#	16-161	16-161	16-161#	16-162	16-162#	16-163	16-163	16-163#	16-164	16-164	16-164#
	16-165	16-165	16-165#	16-166	16-166	16-166	16-166	16-166#	16-167	16-167#	16-168	16-168	16-168	16-168
	16-168#	16-169	16-169	16-169#	16-170	16-170	16-170	16-170#	16-170	16-170#	16-171	16-171#	16-172	16-172
	16-172	16-172#	16-180	16-180#	16-181	16-181	16-181#	16-182	16-182#	16-183	16-183	16-183#	16-184	16-184
	16-184#	16-185	16-185	16-185	16-185	16-185#	16-186	16-186#	16-187	16-187	16-187	16-187	16-187#	16-188
	16-188	16-188#	16-189	16-189	16-189	16-189	16-189#	16-190	16-190#	16-191	16-191	16-191	16-191	16-191#

	16-199	16-199	16-199#	16-200	16-200#	16-201	16-201	16-201#	16-202	16-202	16-202#	16-203	16-203	16-203
	16-203	16-203#	16-204	16-204#	16-205	16-205	16-205	16-205	16-205#	16-206	16-206	16-206#	16-207	16-207
	16-207	16-207	16-207#	16-208	16-208#	16-209	16-209	16-209	16-209	16-209#	16-217	16-217#	16-218	16-218
	16-218#	16-219	16-219#	16-220	16-220#	16-221	16-221	16-221#	16-222	16-222	16-222#	16-223	16-223	16-223
	16-223	16-223	16-223#	16-224	16-224#	16-225	16-225	16-225	16-225	16-225#	16-226	16-226#	16-227	16-227
	16-227	16-227	16-227	16-227#	16-228	16-228#	16-229	16-229	16-229	16-229#	16-237	16-237	16-237#	16-237#
	16-238	16-238#	16-239	16-239	16-239#	16-240	16-240	16-240#	16-241	16-241	16-241	16-241#	16-241#	16-242
	16-242#	16-243	16-243	16-243	16-243	16-243#	22-12	22-12#	24-456	24-456#	24-482	24-482#	24-501	24-501#
	24-520	24-520#												
MSDATA	1-867#	7-18#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17#	8-17#	14-12
	14-12#	14-17	14-17#											
MSDECR	1-D29#	7-18#	10-23	10-23#	11-11	11-11#	16-102	16-102#	16-115	16-115#	16-134	16-134#	16-153	16-153#
	16-173	16-173#	16-192	16-192#	16-210	16-210#	16-230	16-230#	16-244	16-244#	17-11	17-11#	18-12	18-12#
	19-74	19-74#	20-19	20-19#	21-11	21-11#	22-13	22-13#	23-10	23-10#	24-27	24-27#	24-52	24-52#
	24-86	24-86	24-86#	24-86#	24-90	24-90#	24-117	24-117#	24-150	24-150#	24-186	24-186#	24-232	24-232#
	24-286	24-286#	24-316	24-316#	24-316	24-316#	24-320	24-320#	24-350	24-350	24-350#	24-350#	24-354	24-354#
	24-384	24-384	24-384#	24-384#	24-388	24-388#	24-418	24-418#	24-418	24-418#	24-422	24-422#	24-523	24-523#
	24-594	24-594#	24-655	24-655#	24-706	24-706#	24-706	24-706#	24-710	24-710#	24-756	24-756	24-756#	24-756#
	24-760	24-760#	24-803	24-803#	24-803	24-803#	24-807	24-807#	24-867	24-867	24-867#	24-867#	24-872	24-872#
	24-915	24-915	24-915#	24-915#	24-920	24-920#	24-938	24-938#	24-939	24-939#	24-999	24-999#	24-:00	24-:00#
	24-:32	24-:32#	24-:81	24-:81#	24-:27	24-:27#	24-:87	24-:87#	24-:76	24-:76#	24-:32	24-:32#	24-:12	24-:12#
	24-:47	24-:47#	24-:27	24-:27#	24-A10	24-A10#	24-A73	24-A73#	24-B56	24-B56#	24-B88	24-B88#	24-D34	24-D34#
	24-E35	24-E35#	24-E87	24-E87#	24-F55	24-F55#	24-F87	24-F87#	24-G40	24-G40#	24-G90	24-G90#	24-H60	24-H60#
	25-21	25-21#	26-16	26-16#	26-37	26-37#								
MSDEFA	1-E70#	7-18#	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#				
MSENDE	1-D74#	7-18#	10-23#	11-11#	16-102#	16-115#	16-134#	16-153#	16-173#	16-192#	16-210#	16-230#	16-244#	17-11#
	19-74#	20-19#	21-11#	22-13#	23-10#	24-27#	24-52#	24-86#	24-90#	24-117#	24-150#	24-186#	24-232#	24-286#
	24-316#	24-320#	24-350#	24-354#	24-384#	24-388#	24-418#	24-422#	24-523#	24-594#	24-655#	24-706#	24-710#	24-756#
	24-760#	24-803#	24-807#	24-867#	24-872#	24-915#	24-920#	24-938#	24-939#	24-999#	24-:00#	24-:32#	24-:81#	24-:27#
	24-:87#	24-:76#	24-:32#	24-:12#	24-:47#	24-:27#	24-A10#	24-A73#	24-B56#	24-B88#	24-D34#	24-E35#	24-E87#	24-F55#
	24-F87#	24-G40#	24-G90#	24-H60#	25-21#	26-16#	26-37#							
MSERRI	1-249#	7-18#	15-277	15-277#	15-283	15-283#	15-293	15-293#	15-299	15-299#	15-406	15-406#	15-412	15-412#
	15-570	15-570#	15-602	15-602#	15-608	15-608#	15-618	15-618#	15-624	15-624#	15-659	15-659#	15-665	15-665#
	15-699	15-699#	15-705	15-705#	15-713	15-713#	15-719	15-719#	15-865	15-865#	15-870	15-870#	15-889	15-889#
	15-895	15-895#	15-904	15-904#	15-910	15-910#	15-919	15-919#	15-925	15-925#	15-934	15-934#	15-940	15-940#
	15-:72	15-:72#	15-:78	15-:78#	24-24	24-24#	24-50	24-50#	24-84	24-84#	24-115	24-115#	24-148	24-148#
	24-184	24-184#	24-226	24-226#	24-278	24-278#	24-314	24-314#	24-348	24-348#	24-382	24-382#	24-416	24-416#
	24-480	24-480#	24-499	24-499#	24-518	24-518#	24-551	24-551#	24-565	24-565#	24-577	24-577#	24-588	24-588#
	24-637	24-637#	24-647	24-647#	24-694	24-694#	24-704	24-704#	24-745	24-745#	24-754	24-754#	24-792	24-792#
	24-801	24-801#	24-855	24-855#	24-865	24-865#	24-913	24-913#	24-936	24-936#	24-983	24-983#	24-993	24-993#
	24-:98	24-:98#	24-:21	24-:21#	24-:83	24-:83#	24-:26	24-:26#	24-:35	24-:35#	24-:47	24-:47#	24-:66	24-:66#
	24-:74	24-:74#	24-:97	24-:97#	24-:12	24-:12#	24-:95	24-:95#	24-A60	24-A60#	24-B41	24-B41#	24-B85	24-B85#
	24-C89	24-C89#												
MSESCA	1-D06#	7-18#	24-227	24-227#	24-279	24-279#	24-552	24-552#	24-566	24-566#	24-578	24-578#	24-589	24-589#
	24-638	24-638#	24-648	24-648#	24-695#	24-746#	24-793#	24-856#	24-984	24-984#	24-994	24-994#	24-:99	24-:99#
	24-:22	24-:22#												
MSESCS	1-D10#	7-18#	24-227#	24-279#	24-552#	24-566#	24-578#	24-589#	24-638#	24-648#	24-695	24-695#	24-746	24-746#
	24-793	24-793#	24-856	24-856#	24-984#	24-994#	24-:99#	24-:22#						
MSEXCP	1-E01#	7-18#	25-16	25-16	25-16#	25-17	25-17	25-17#	25-18	25-18	25-18#	25-19	25-19	25-19#
MSEXIT	1-D14#	7-18#												
MSEXSE	1-D22#	7-18#												
MSEXTJ	1-D18#	7-18#												
MSGEN	1-D38#	7-18#	7-24	7-24#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17

	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	9-8	9-8#	10-9
	10-9#	10-9#	10-23	10-23#	11-8	11-8	11-8#	11-11	11-11#	14-12	14-12#	14-17	14-17#
	16-100	16-100#	16-102	16-102#	16-106	16-106#	16-115	16-115#	16-121	16-121#	16-134	16-134#	16-140
	16-153	16-153#	16-159	16-159#	16-173	16-173#	16-179	16-179#	16-192	16-192#	16-198	16-198#	16-210
	16-216	16-216#	16-230	16-230#	16-236	16-236#	16-244	16-244#	17-9	17-9#	17-11	17-11#	18-8
	19-8	19-8#	19-74	19-74#	20-7	20-7#	20-19	20-19#	21-8	21-8#	21-11	21-11#	22-8
	22-13	22-13#	23-9	23-9#	23-10	23-10#	24-14	24-14#	24-27	24-27#	24-39	24-39#	24-52
	24-67	24-67#	24-86	24-86#	24-90	24-90#	24-101	24-101#	24-117	24-117#	24-128	24-128#	24-150
	24-166	24-166#	24-186	24-186#	24-203	24-203#	24-232	24-232#	24-247	24-247#	24-286	24-286#	24-298
	24-316	24-316#	24-320	24-320#	24-332	24-332#	24-350	24-350#	24-354	24-354#	24-366	24-366#	24-384
	24-388	24-388#	24-400	24-400#	24-418	24-418#	24-422	24-422#	24-449	24-449#	24-523	24-523#	24-539
	24-594	24-594#	24-610	24-610#	24-655	24-655#	24-676	24-676#	24-706	24-706#	24-710	24-710#	24-727
	24-756	24-756#	24-760	24-760#	24-774	24-774#	24-803	24-803#	24-807	24-807#	24-832	24-832#	24-867
	24-872	24-872#	24-889	24-889#	24-898	24-898#	24-915	24-915#	24-920	24-920#	24-924	24-924#	24-938
	24-939	24-939#	24-957	24-957#	24-960	24-960#	24-999	24-999#	24-:00	24-:00#	24-:22	24-:22#	24-:32
	24-:52	24-:52#	24-:81	24-:81#	24-:01	24-:01#	24-:27	24-:27#	24-:49	24-:49#	24-:87	24-:87#	24-:07
	24-:76	24-:76#	24-:91	24-:91#	24-:32	24-:32#	24-:48	24-:48#	24-:12	24-:12#	24-:27	24-:27#	24-:47
	24-:63	24-:63#	24-:27	24-:27#	24-:43	24-:43#	24-A10	24-A10#	24-A26	24-A26#	24-A73	24-A73#	24-A89
	24-B56	24-B56#	24-B69	24-B69#	24-B88	24-B88#	24-C05	24-C05#	24-D34	24-D34#	24-D58	24-D58#	24-E35
	24-E53	24-E53#	24-E87	24-E87#	24-F05	24-F05#	24-F55	24-F55#	24-F70	24-F70#	24-F87	24-F87#	24-G08
	24-G40	24-G40#	24-G59	24-G59#	24-G90	24-G90#	24-H13	24-H13#	24-H60	24-H60#	25-14	25-14#	25-21
	26-13	26-13#	26-16	26-16#	26-39	26-39#							
MSGENB	1-C38#	7-18#											
MSGETS	1-D35#	7-18#	10-23	10-23#	11-11	11-11#	16-102	16-102#	16-115	16-115#	16-134	16-134#	16-153
	16-173	16-173#	16-192	16-192#	16-210	16-210#	16-230	16-230#	16-244	16-244#	17-11	17-11#	18-12
	19-74	19-74#	20-19	20-19#	21-11	21-11#	22-13	22-13#	23-10	23-10#	24-27	24-27#	24-52
	24-86	24-86	24-86#	24-86#	24-90	24-90#	24-117	24-117#	24-150	24-150#	24-186	24-186#	24-232
	24-286	24-286#	24-316	24-316	24-316#	24-316#	24-320	24-320#	24-350	24-350	24-350#	24-350#	24-354
	24-384	24-384	24-384#	24-384#	24-388	24-388#	24-418	24-418	24-418#	24-418#	24-422	24-422#	24-523
	24-594	24-594#	24-655	24-655#	24-695	24-695#	24-706	24-706	24-706#	24-706#	24-710	24-710#	24-746
	24-756	24-756	24-756#	24-756#	24-760	24-760#	24-793	24-793#	24-803	24-803	24-803#	24-803#	24-807
	24-856	24-856#	24-867	24-867	24-867#	24-867#	24-872	24-872#	24-915	24-915	24-915#	24-915#	24-920
	24-938	24-938#	24-939	24-939#	24-999	24-999#	24-:00	24-:00#	24-:32	24-:32#	24-:81	24-:81#	24-:27
	24-:87	24-:87#	24-:76	24-:76#	24-:32	24-:32#	24-:12	24-:12#	24-:47	24-:47#	24-:27	24-:27#	24-A10
	24-A73	24-A73#	24-B56	24-B56#	24-B88	24-B88#	24-D34	24-D34#	24-E35	24-E35#	24-E87	24-E87#	24-F55
	24-F87	24-F87#	24-G40	24-G40#	24-G90	24-G90#	24-H60	24-H60#	25-21	25-21#	26-16	26-16#	26-37
MSGETT	1-B77#	7-18#	24-227#	24-279#	24-552#	24-566#	24-578#	24-589#	24-638#	24-648#	24-695	24-695#	24-746
	24-793	24-793#	24-856	24-856#	24-984#	24-994#	24-:99#	24-:22#					
MSGNGB	1-C02#	7-18#	7-24	7-24#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	9-8	9-8#	10-9
	10-9#	11-8	11-8	11-8#	14-12	14-12#	14-17	14-17#	16-100	16-100#	16-106	16-106#	16-121
	16-140	16-140#	16-159	16-159#	16-179	16-179#	16-198	16-198#	16-216	16-216#	16-236	16-236#	17-9
	18-8	18-8#	19-8	19-8#	20-7	20-7#	21-8	21-8#	22-8	22-8#	23-9	23-9#	25-14
	26-13	26-13#	26-39	26-39#									
MSGNIN	1-D49#	7-18#	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17	8-17
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#	8-17#
	8-17#	8-17#	8-17#										

[illegible]

	19-34#	19-34#	19-35	19-35#	19-51	19-51	19-51	19-51#	19-51#	19-51#	19-52	19-52#	19-74	19-74#
	20-9	20-9	20-9#	20-9#	20-16	20-16	20-16#	20-16#	20-19	20-19#	21-11	21-11#	22-10	22-10#
	22-12	22-12	22-12	22-12	22-12	22-12	22-12#	22-12#	22-12#	22-12#	22-12#	22-13	22-13#	23-10
	23-10#	24-16	24-16	24-16#	24-16#	24-24	24-24	24-24	24-24	24-24#	24-24#	24-24#	24-24#	24-24#
	24-27	24-27#	24-50	24-50	24-50	24-50	24-50#	24-50#	24-50#	24-50#	24-50#	24-52	24-52#	24-72
	24-72#	24-84	24-84	24-84	24-84	24-84#	24-84#	24-84#	24-84#	24-84#	24-86	24-86#	24-90	24-90#
	24-115	24-115	24-115	24-115	24-115#	24-115#	24-115#	24-115#	24-115#	24-117	24-117#	24-133	24-133#	24-148
	24-148	24-148	24-148	24-148#	24-148#	24-148#	24-148#	24-148#	24-150	24-150#	24-184	24-184	24-184	24-184
	24-184#	24-184#	24-184#	24-184#	24-184#	24-186	24-186#	24-226	24-226	24-226	24-226	24-226#	24-226#	24-226#
	24-226#	24-226#	24-227	24-227	24-227#	24-227#	24-232	24-232#	24-278	24-278	24-278	24-278	24-278#	24-278#
	24-278#	24-278#	24-278#	24-279	24-279#	24-279#	24-286	24-286#	24-303	24-303#	24-314	24-314	24-314	24-314
	24-314	24-314#	24-314#	24-314#	24-314#	24-316	24-316#	24-320	24-320#	24-337	24-337#	24-348	24-348	24-348
	24-348	24-348	24-348#	24-348#	24-348#	24-348#	24-350	24-350#	24-354	24-354#	24-371	24-371#	24-382	24-382
	24-382	24-382	24-382	24-382#	24-382#	24-382#	24-382#	24-384	24-384#	24-388	24-388#	24-405	24-405#	24-405#
	24-416	24-416	24-416	24-416	24-416#	24-416#	24-416#	24-416#	24-418	24-418#	24-422	24-422#	24-456	24-456
	24-456	24-456	24-456	24-456	24-456	24-456#	24-456#	24-456#	24-456#	24-480	24-480	24-480	24-480	24-480
	24-480#	24-480#	24-480#	24-480#	24-480#	24-482	24-482	24-482	24-482	24-482#	24-482#	24-482#	24-482#	24-482#
	24-499	24-499	24-499	24-499	24-499#	24-499#	24-499#	24-499#	24-499#	24-501	24-501	24-501	24-501	24-501
	24-501#	24-501#	24-501#	24-501#	24-518	24-518	24-518	24-518	24-518#	24-518#	24-518#	24-518#	24-518#	24-520
	24-520	24-520	24-520	24-520	24-520#	24-520#	24-520#	24-520#	24-523	24-523#	24-551	24-551	24-551	24-551
	24-551#	24-551#	24-551#	24-551#	24-551#	24-552	24-552	24-552#	24-552#	24-565	24-565	24-565	24-565	24-565#
	24-565#	24-565#	24-565#	24-565#	24-566	24-566	24-566#	24-566#	24-577	24-577	24-577	24-577	24-577#	24-577#
	24-577#	24-577#	24-577#	24-578	24-578	24-578#	24-578#	24-588	24-588	24-588	24-588	24-588#	24-588#	24-588#
	24-588#	24-588#	24-589	24-589	24-589#	24-589#	24-594	24-594#	24-637	24-637	24-637	24-637	24-637#	24-637#
	24-637#	24-637#	24-637#	24-638	24-638	24-638#	24-638#	24-647	24-647	24-647	24-647	24-647#	24-647#	24-647#
	24-647#	24-647#	24-648	24-648	24-648#	24-648#	24-655	24-655#	24-680	24-680#	24-694	24-694	24-694	24-694
	24-694#	24-694#	24-694#	24-694#	24-694#	24-695	24-695	24-695#	24-704	24-704	24-704	24-704	24-704	24-704#
	24-704#	24-704#	24-704#	24-704#	24-706	24-706#	24-710	24-710#	24-731	24-731#	24-745	24-745	24-745	24-745
	24-745#	24-745#	24-745#	24-745#	24-745#	24-746	24-746	24-746#	24-754	24-754	24-754	24-754	24-754	24-754#
	24-754#	24-754#	24-754#	24-754#	24-756	24-756#	24-760	24-760#	24-778	24-778#	24-792	24-792	24-792	24-792
	24-792#	24-792#	24-792#	24-792#	24-793	24-793	24-793#	24-793#	24-801	24-801	24-801	24-801	24-801	24-801#
	24-801#	24-801#	24-801#	24-801#	24-803	24-803#	24-807	24-807#	24-838	24-838#	24-855	24-855	24-855	24-855
	24-855#	24-855#	24-855#	24-855#	24-855#	24-856	24-856	24-856#	24-856#	24-865	24-865	24-865	24-865	24-865#
	24-865#	24-865#	24-865#	24-865#	24-867	24-867#	24-872	24-872#	24-898	24-898#	24-900	24-900#	24-913	24-913
	24-913	24-913	24-913#	24-913#	24-913#	24-913#	24-913#	24-915	24-915#	24-920	24-920#	24-924	24-924#	24-936
	24-936	24-936	24-936	24-936#	24-936#	24-936#	24-936#	24-936#	24-938	24-938#	24-939	24-939#	24-960	24-960#
	24-983	24-983	24-983	24-983	24-983#	24-983#	24-983#	24-983#	24-983#	24-984	24-984	24-984#	24-984#	24-993
	24-993	24-993	24-993	24-993#	24-993#	24-993#	24-993#	24-993#	24-994	24-994	24-994#	24-994#	24-999	24-999#
	24-:00	24-:00#	24-:98	24-:98	24-:98	24-:98	24-:98#	24-:98#	24-:98#	24-:98#	24-:98#	24-:99	24-:99	24-:99#
	24-:99#	24-:21	24-:21	24-:21	24-:21	24-:21#	24-:21#	24-:21#	24-:21#	24-:21#	24-:22	24-:22	24-:22#	24-:22#
	24-:32	24-:32#	24-:81	24-:81#	24-<27	24-<27#	24-<83	24-<83	24-<83	24-<83	24-<83#	24-<83#	24-<83#	24-<83#
	24-<83#	24-<87	24-<87#	24-=26	24-=26	24-=26	24-=26	24-=26#	24-=26#	24-=26#	24-=26#	24-=26#	24-=35	24-=35
	24-=35	24-=35	24-=35#	24-=35#	24-=35#	24-=35#	24-=35#	24-=47	24-=47	24-=47	24-=47	24-=47#	24-=47#	24-=47#
	24-=47#	24-=47#	24-=66	24-=66	24-=66	24-=66	24-=66#	24-=66#	24-=66#	24-=66#	24-=66#	24-=66#	24-=74	24-=74
	24-=74	24-=74#	24-=74#	24-=74#	24-=74#	24-=74#	24-=76	24-=76#	24->32	24->32#	24-?12	24-?12#	24-?47	24-?47#
	24-?97	24-?97	24-?97	24-?97	24-?97#	24-?97#	24-?97#	24-?97#	24-?97#	24-?97#	24-@12	24-@12	24-@12	24-@12#
	24-@12#	24-@12#	24-@12#	24-@12#	24-@27	24-@27#	24-@95	24-@95	24-@95	24-@95	24-@95#	24-@95#	24-@95#	24-@95#
	24-@95#	24-A10	24-A10#	24-A60	24-A60	24-A60	24-A60	24-A60#	24-A60#	24-A60#	24-A60#	24-A60#	24-A73	24-A73#
	24-B41	24-B41	24-B41	24-B41	24-B41#	24-B41#	24-B41#	24-B41#	24-B41#	24-B56	24-B56#	24-B85	24-B85	24-B85
	24-B85	24-B85#	24-B85#	24-B85#	24-B85#	24-B85#	24-B88	24-B88#	24-C89	24-C89	24-C89	24-C89	24-C89#	24-C89#
	24-C89#	24-C89#	24-C89#	24-D34	24-D34#	24-E35	24-E35#	24-E87	24-E87#	24-F55	24-F55#	24-F87	24-F87#	24-G40
	24-G40#	24-G90	24-G90#	24-H60	24-H60#	25-14	25-14#	25-16	25-16	25-16	25-16	25-16#	25-17	25-17
	25-17	25-17	25-17#	25-18	25-18	25-18	25-18	25-18	25-18#	25-19	25-19	25-19	25-19	25-19
	25-19#	25-21	25-21#	26-13	26-13#	26-16	26-16#	26-39	26-39	26-39	26-39#	26-39#	26-39#	26-39#
MSGNLS	1-C13#	7-18#	24-86	24-86#	24-316	24-316#	24-350	24-350#	24-384	24-384#	24-418	24-418#	24-706	24-706#
MSGNSU	24-756	24-756#	24-803	24-803#	24-867	24-867#	24-915	24-915#						
	1-B98#	7-18#	24-898	24-898#	24-924	24-924#	24-960	24-960#						

MSGNTA	1-B90#	7-18#	10-23	10-23#	11-11	11-11#	16-102	16-102#	16-115	16-115#	16-134	16-134#	16-153	16-153#
	16-173	16-173#	16-192	16-192#	16-210	16-210#	16-230	16-230#	16-244	16-244#	17-11	17-11#	19-74	19-74#
	20-19	20-19#	21-11	21-11#	22-13	22-13#	23-10	23-10#	24-27	24-27#	24-52	24-52#	24-90	24-90#
	24-117	24-117#	24-150	24-150#	24-186	24-186#	24-232	24-232#	24-286	24-286#	24-320	24-320#	24-354	24-354#
	24-388	24-388#	24-422	24-422#	24-523	24-523#	24-594	24-594#	24-655	24-655#	24-710	24-710#	24-760	24-760#
	24-807	24-807#	24-872	24-872#	24-920	24-920#	24-938	24-938#	24-939	24-939#	24-999	24-999#	24-:00	24-:00#
	24-:32	24-:32#	24-:81	24-:81#	24-<27	24-<27#	24-<87	24-<87#	24-=76	24-=76#	24->32	24->32#	24-?12	24-?12#
	24-?47	24-?47#	24-@2	24-@27#	24-A10	24-A10#	24-A73	24-A73#	24-B56	24-B56#	24-B88	24-B88#	24-D34	24-D34#
	24-E35	24-E35#	24-E87	24-E87#	24-F55	24-F55#	24-F87	24-F87#	24-G40	24-G40#	24-G90	24-G90#	24-H60	24-H60#
	25-21	25-21#	26-16	26-16#										
MSGNTE	1-B94#	7-18#	24-14	24-14#	24-39	24-39#	24-67	24-67#	24-101	24-101#	24-128	24-128#	24-166	24-166#
	24-203	24-203#	24-247	24-247#	24-298	24-298#	24-332	24-332#	24-366	24-366#	24-400	24-400#	24-449	24-449#
	24-539	24-539#	24-610	24-610#	24-676	24-676#	24-727	24-727#	24-774	24-774#	24-832	24-832#	24-889	24-889#
	24-957	24-957#	24-:22	24-:22#	24-:52	24-:52#	24-<01	24-<01#	24-<49	24-<49#	24-=07	24-=07#	24-=91	24-=91#
	24->48	24->48#	24-?27	24-?27#	24-?63	24-?63#	24-@43	24-@43#	24-A26	24-A26#	24-A89	24-A89#	24-B69	24-B69#
	24-C05	24-C05#	24-D58	24-D58#	24-E53	24-E53#	24-F05	24-F05#	24-F70	24-F70#	24-G08	24-G08#	24-G59	24-G59#
	24-H13	24-H13#												
MSHAPT	1-A39#	7-18#	8-17	8-17#										
MSHNAP	1-B24#	7-18#	8-17	8-17#										
MSINCR	1-D26#	7-18#	7-24	7-24#	10-9	10-9	10-9#	10-9#	11-8	11-8	11-8#	11-8#	15-277#	15-283#
	15-293#	15-299#	15-406#	15-412#	15-570#	15-602#	15-608#	15-618#	15-624#	15-659#	15-665#	15-699#	15-705#	15-713#
	15-719#	15-865#	15-870#	15-889#	15-895#	15-904#	15-910#	15-919#	15-925#	15-934#	15-940#	15-:72#	15-:78#	16-100
	16-100	16-100#	16-100#	16-101#	16-102#	16-106	16-106	16-106#	16-106#	16-107#	16-108#	16-109#	16-110#	16-111#
	16-112#	16-113#	16-114#	16-115#	16-121	16-121	16-121#	16-121#	16-122#	16-123#	16-124#	16-125#	16-126#	16-127#
	16-128#	16-129#	16-130#	16-131#	16-132#	16-133#	16-134#	16-140	16-140	16-140#	16-140#	16-141#	16-142#	16-143#
	16-144#	16-145#	16-146#	16-147#	16-148#	16-149#	16-150#	16-151#	16-152#	16-153#	16-159	16-159	16-159#	16-159#
	16-160#	16-161#	16-162#	16-163#	16-164#	16-165#	16-166#	16-167#	16-168#	16-169#	16-170#	16-171#	16-172#	16-173#
	16-179	16-179	16-179#	16-179#	16-180#	16-181#	16-182#	16-183#	16-184#	16-185#	16-186#	16-187#	16-188#	16-189#
	16-190#	16-191#	16-192#	16-198	16-198	16-198#	16-198#	16-199#	16-200#	16-201#	16-202#	16-203#	16-204#	16-205#
	16-206#	16-207#	16-208#	16-209#	16-210#	16-216	16-216	16-216#	16-216#	16-217#	16-218#	16-219#	16-220#	16-221#
	16-222#	16-223#	16-224#	16-225#	16-226#	16-227#	16-228#	16-229#	16-230#	16-236	16-236	16-236#	16-236#	16-237#
	16-238#	16-239#	16-240#	16-241#	16-242#	16-243#	16-244#	17-9	17-9	17-9#	17-9#	17-11#	18-8	18-8
	18-8#	18-8#	19-8	19-8	19-8#	19-8#	19-25#	19-28#	19-31#	19-34#	19-51#	19-74#	20-7	20-7
	20-7#	20-7#	20-9#	20-16#	20-19#	21-8	21-8	21-8#	21-8#	21-11#	22-8	22-8	22-8#	22-8#
	22-10#	22-12#	22-13#	23-9	23-9	23-9#	23-9#	23-10#	24-14	24-14	24-14	24-14#	24-14#	24-14#
	24-16#	24-24#	24-27#	24-39	24-39	24-39	24-39#	24-39#	24-39#	24-50#	24-52#	24-67	24-67	24-67
	24-67#	24-67#	24-67#	24-72	24-72	24-72	24-72#	24-72#	24-72#	24-72#	24-84#	24-86#	24-90#	24-101
	24-101	24-101	24-101#	24-101#	24-101#	24-115#	24-117#	24-128	24-128	24-128	24-128#	24-128#	24-128#	24-133#
	24-148#	24-150#	24-166	24-166	24-166	24-166#	24-166#	24-166#	24-184#	24-186#	24-203	24-203	24-203	24-203#
	24-203#	24-203#	24-226#	24-227#	24-232#	24-247	24-247	24-247	24-247#	24-247#	24-247#	24-278#	24-279#	24-286#
	24-298	24-298	24-298	24-298#	24-298#	24-298#	24-303	24-303	24-303	24-303#	24-303#	24-303#	24-303#	24-314#
	24-316#	24-320#	24-332	24-332	24-332	24-332#	24-332#	24-332#	24-332#	24-337	24-337	24-337	24-337#	24-337#
	24-337#	24-348#	24-350#	24-354#	24-366	24-366	24-366	24-366#	24-366#	24-366#	24-371	24-371	24-371	24-371#
	24-371#	24-371#	24-371#	24-382#	24-384#	24-368#	24-400	24-400	24-400	24-400#	24-400#	24-400#	24-405	24-405
	24-405	24-405#	24-405#	24-405#	24-405#	24-416#	24-418#	24-422#	24-449	24-449	24-449	24-449#	24-449#	24-449#
	24-456#	24-480#	24-482#	24-499#	24-501#	24-518#	24-520#	24-523#	24-539	24-539	24-539	24-539#	24-539#	24-539#
	24-551#	24-552#	24-565#	24-566#	24-577#	24-578#	24-588#	24-589#	24-594#	24-610	24-610	24-610	24-610#	24-610#
	24-610#	24-637#	24-638#	24-647#	24-648#	24-655#	24-676	24-676	24-676	24-676#	24-676#	24-676#	24-680	24-680
	24-680	24-680#	24-680#	24-680#	24-680#	24-694#	24-695#	24-704#	24-706#	24-710#	24-727	24-727	24-727	24-727#
	24-727#	24-727#	24-731	24-731	24-731	24-731#	24-731#	24-731#	24-731#	24-745#	24-746#	24-754#	24-756#	24-760#
	24-774	24-774	24-774	24-774#	24-774#	24-774#	24-778	24-778	24-778	24-778#	24-778#	24-778#	24-778#	24-792#
	24-793#	24-801#	24-803#	24-807#	24-832	24-832	24-832	24-832#	24-832#	24-832#	24-838	24-838	24-838	24-838#
	24-838#	24-838#	24-838#	24-855#	24-856#	24-865#	24-867#	24-872#	24-889	24-889	24-889	24-889#	24-889#	24-889#
	24-898	24-898	24-898	24-898#	24-898#	24-898#	24-900	24-900	24-900	24-900#	24-900#	24-900#	24-900#	24-913#
	24-915#	24-920#	24-924	24-924	24-924	24-924#	24-924#	24-924#	24-936#	24-938#	24-939#	24-957	24-957	24-957
	24-957#	24-957#	24-957#	24-960	24-960	24-960	24-960#	24-960#	24-960#	24-983#	24-984#	24-993#	24-994#	24-999#
	24-:00#	24-:22	24-:22	24-:22	24-:22#	24-:22#	24-:22#	24-:22#	24-:22#	24-:21#	24-:22#	24-:32#	24-:52	24-:52

[illegible]

[illegible]

	16-163#	16-163#	16-164	16-164	16-164	16-164	16-164#	16-164#	16-164#	16-164#	16-165	16-165	16-165	16-165
	16-165#	16-165#	16-165#	16-165#	16-166	16-166	16-166	16-166	16-166	16-166	16-166#	16-166#	16-166#	16-166#
	16-166#	16-166#	16-167	16-167	16-167	16-167#	16-167#	16-167#	16-167#	16-168	16-168	16-168	16-168	16-168
	16-168#	16-168#	16-168#	16-168#	16-168#	16-168#	16-169	16-169	16-169	16-169	16-169#	16-169#	16-169#	16-169#
	16-170	16-170	16-170	16-170	16-170	16-170	16-170#	16-170#	16-170#	16-170#	16-170#	16-170#	16-170#	16-171
	16-171	16-171#	16-171#	16-171#	16-172	16-172	16-172	16-172	16-172	16-172	16-172#	16-172#	16-172#	16-172#
	16-172#	16-172#	16-180	16-180	16-180	16-180#	16-180#	16-180#	16-180#	16-181	16-181	16-181	16-181	16-181#
	16-181#	16-181#	16-182	16-182	16-182#	16-182#	16-183	16-183	16-183	16-183	16-183#	16-183#	16-183#	16-183#
	16-184	16-184	16-184	16-184	16-184#	16-184#	16-184#	16-184#	16-184#	16-185	16-185	16-185	16-185	16-185
	16-185#	16-185#	16-185#	16-185#	16-185#	16-185#	16-186	16-186	16-186	16-186#	16-186#	16-186#	16-186#	16-187
	16-187	16-187	16-187	16-187	16-187#	16-187#	16-187#	16-187#	16-187#	16-187#	16-187#	16-188	16-188	16-188
	16-188#	16-188#	16-188#	16-188#	16-189	16-189	16-189	16-189	16-189	16-189	16-189#	16-189#	16-189#	16-189#
	16-189#	16-189#	16-190	16-190	16-190	16-190#	16-190#	16-190#	16-190#	16-191	16-191	16-191	16-191	16-191
	16-191#	16-191#	16-191#	16-191#	16-191#	16-191#	16-199	16-199	16-199	16-199	16-199#	16-199#	16-199#	16-199#
	16-200	16-200	16-200#	16-200#	16-201	16-201	16-201	16-201	16-201#	16-201#	16-201#	16-201#	16-201#	16-202
	16-202	16-202	16-202#	16-202#	16-202#	16-202#	16-203	16-203	16-203	16-203	16-203	16-203	16-203#	16-203#
	16-203#	16-203#	16-203#	16-203#	16-204	16-204	16-204	16-204#	16-204#	16-204#	16-204#	16-205	16-205	16-205
	16-205	16-205	16-205#	16-205#	16-205#	16-205#	16-205#	16-205#	16-205#	16-206	16-206	16-206	16-206#	16-206#
	16-206#	16-206#	16-207	16-207	16-207	16-207	16-207	16-207	16-207#	16-207#	16-207#	16-207#	16-207#	16-207#
	16-208	16-208	16-208	16-208#	16-208#	16-208#	16-209	16-209	16-209	16-209	16-209	16-209	16-209#	16-209#
	16-209#	16-209#	16-209#	16-209#	16-217	16-217	16-217	16-217#	16-217#	16-217#	16-217#	16-218	16-218	16-218
	16-218#	16-218#	16-218#	16-218#	16-219	16-219	16-219#	16-219#	16-219#	16-220	16-220	16-220	16-220#	16-220#
	16-220#	16-220#	16-221	16-221	16-221	16-221	16-221#	16-221#	16-221#	16-221#	16-221#	16-222	16-222	16-222
	16-222#	16-222#	16-222#	16-222#	16-223	16-223	16-223	16-223	16-223	16-223	16-223#	16-223#	16-223#	16-223#
	16-223#	16-223#	16-224	16-224	16-224	16-224#	16-224#	16-224#	16-224#	16-225	16-225	16-225	16-225	16-225
	16-225#	16-225#	16-225#	16-225#	16-225#	16-225#	16-226	16-226	16-226	16-226	16-226#	16-226#	16-226#	16-226#
	16-227	16-227	16-227	16-227	16-227	16-227	16-227#	16-227#	16-227#	16-227#	16-227#	16-227#	16-227#	16-228
	16-228	16-228#	16-228#	16-228#	16-229	16-229	16-229	16-229	16-229	16-229	16-229#	16-229#	16-229#	16-229#
	16-229#	16-229#	16-237	16-237	16-237	16-237	16-237#	16-237#	16-237#	16-237#	16-237#	16-238	16-238	16-238#
	16-239	16-239	16-239	16-239	16-239#	16-239#	16-239#	16-239#	16-239#	16-240	16-240	16-240	16-240#	16-240#
	16-240#	16-240#	16-241	16-241	16-241	16-241	16-241	16-241	16-241#	16-241#	16-241#	16-241#	16-241#	16-241#
	16-242	16-242	16-242	16-242#	16-242#	16-242#	16-243	16-243	16-243	16-243	16-243	16-243	16-243#	16-243#
	16-243#	16-243#	16-243#	16-243#	22-12	22-12	22-12	22-12#	22-12#	22-12#	22-12#	24-456	24-456	24-456#
	24-456#	24-456#	24-482	24-482	24-482#	24-482#	24-501	24-501	24-501#	24-501#	24-501#	24-520	24-520	24-520#
MSRADI	1-D77#	7-18#	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#	25-19#	25-19#	25-19#	25-19#
MSRBRO	1-C52#	7-18#												
MSRNRO	1-C62#	7-18#	19-51	19-51#										
MSSETS	1-D32#	7-18#	7-24	7-24#	10-9	10-9#	11-8	11-8#	16-100	16-100#	16-106	16-106#	16-121	16-121#
	16-140	16-140#	16-159	16-159#	16-179	16-179#	16-198	16-198#	16-216	16-216#	16-236	16-236#	17-9	17-9#
	18-8	18-8#	19-8	19-8#	20-7	20-7#	21-8	21-8#	22-8	22-8#	23-9	23-9#	24-14	24-14#
	24-39	24-39#	24-67	24-67#	24-72	24-72	24-72#	24-72#	24-101	24-101#	24-128	24-128#	24-166	24-166#
	24-203	24-203#	24-247	24-247#	24-298	24-298#	24-303	24-303	24-303#	24-303#	24-332	24-332#	24-337	24-337
	24-337#	24-337#	24-366	24-366#	24-371	24-371	24-371#	24-371#	24-400	24-400#	24-405	24-405	24-405#	24-405#
	24-449	24-449#	24-539	24-539#	24-610	24-610#	24-676	24-676#	24-680	24-680	24-680#	24-680#	24-727	24-727#
	24-731	24-731	24-731#	24-731#	24-774	24-774#	24-778	24-778	24-778#	24-778#	24-832	24-832#	24-838	24-838
	24-838#	24-838#	24-889	24-889#	24-898	24-898#	24-900	24-900	24-900#	24-900#	24-924	24-924#	24-957	24-957#
	24-960	24-960#	24-:22	24-:22#	24-:52	24-:52#	24-<01	24-<01#	24-<49	24-<49#	24-=07	24-=07#	24-=91	24-=91#
	24->48	24->48#	24-?27	24-?27#	24-?63	24-?63#	24-@43	24-@43#	24-A26	24-A26#	24-A89	24-A89#	24-B69	24-B69#
	24-C05	24-C05#	24-D58	24-D58#	24-E53	24-E53#	24-F05	24-F05#	24-F70	24-F70#	24-G08	24-G08#	24-G59	24-G59#
	24-H13	24-H13#	25-14	25-14#	26-13	26-13#								
MSSTAR	1-A33#	7-18#												
MS SVC	1-C33#	7-18#	15-277	15-283	15-293	15-299	15-406	15-412	15-570	15-602	15-608	15-618	15-624	15-659
	15-665	15-699	15-705	15-713	15-719	15-865	15-870	15-889	15-895	15-904	15-910	15-919	15-925	15-934
	15-940	15-:72	15-:78	16-101	16-101#	16-102	16-102#	16-107	16-107#	16-108	16-108#	16-109	16-109#	16-110
	16-110#	16-111	16-111#	16-112	16-112#	16-113	16-113#	16-114	16-114#	16-115	16-115#	16-122	16-122#	16-123
	16-123#	16-124	16-124#	16-125	16-125#	16-126	16-126#	16-127	16-127#	16-128	16-128#	16-129	16-129#	16-130
	16-130#	16-131	16-131#	16-132	16-132#	16-133	16-133#	16-134	16-134#	16-141	16-141#	16-142	16-142#	16-143

	16-143#	16-144	16-144#	16-145	16-145#	16-146	16-146#	16-147	16-148	16-148#	16-149	16-149#	16-150
	16-150#	16-151	16-151#	16-152	16-152#	16-153	16-153#	16-160	16-161	16-161#	16-162	16-162#	16-163
	16-163#	16-164	16-164#	16-165	16-165#	16-166	16-166#	16-167	16-168	16-168#	16-169	16-169#	16-170
	16-170#	16-171	16-171#	16-172	16-172#	16-173	16-173#	16-180	16-181	16-181#	16-182	16-182#	16-183
	16-183#	16-184	16-184#	16-185	16-185#	16-186	16-186#	16-187	16-188	16-188#	16-189	16-189#	16-190
	16-190#	16-191	16-191#	16-192	16-192#	16-199	16-199#	16-200	16-200#	16-201	16-201#	16-202	16-203
	16-203#	16-204	16-204#	16-205	16-205#	16-206	16-206#	16-207	16-207#	16-208	16-208#	16-209	16-210
	16-210#	16-217	16-217#	16-218	16-218#	16-219	16-219#	16-220	16-220#	16-221	16-221#	16-222	16-223
	16-223#	16-224	16-224#	16-225	16-225#	16-226	16-226#	16-227	16-227#	16-228	16-228#	16-229	16-230
	16-230#	16-237	16-237#	16-238	16-238#	16-239	16-239#	16-240	16-240#	16-241	16-241#	16-242	16-243
	16-243#	16-244	16-244#	17-11	17-11#	19-25	19-25#	19-28	19-28#	19-31	19-31#	19-34	19-51
	19-51#	19-74	19-74#	20-9	20-9#	20-16	20-16#	20-19	20-19#	21-11	21-11#	22-10	22-12
	22-12#	22-13	22-13#	23-10	23-10#	24-16	24-16#	24-24	24-27	24-27#	24-50	24-52	24-72
	24-72#	24-84	24-86	24-86#	24-90	24-90#	24-115	24-117	24-117#	24-133	24-133#	24-148	24-150
	24-184	24-186	24-186#	24-226	24-227	24-227#	24-232	24-232#	24-278	24-279	24-279#	24-286	24-303
	24-303#	24-314	24-316	24-316#	24-320	24-320#	24-337	24-337#	24-348	24-350	24-350#	24-354	24-371
	24-371#	24-382	24-384	24-384#	24-388	24-388#	24-405	24-405#	24-416	24-418	24-418#	24-422	24-456
	24-456#	24-480	24-482	24-482#	24-499	24-501	24-501#	24-518	24-520	24-520#	24-523	24-523#	24-551
	24-552#	24-565	24-566	24-566#	24-577	24-578	24-578#	24-588	24-589	24-589#	24-594	24-594#	24-637
	24-638#	24-647	24-648	24-648#	24-655	24-655#	24-680	24-680#	24-694	24-695	24-695#	24-704	24-706
	24-710	24-710#	24-731	24-731#	24-745	24-746	24-746#	24-754	24-756	24-756#	24-760	24-760#	24-778
	24-792	24-793	24-793#	24-801	24-803	24-803#	24-807	24-807#	24-838	24-838#	24-855	24-856	24-865
	24-867	24-867#	24-872	24-872#	24-898	24-898#	24-900	24-900#	24-913	24-915	24-915#	24-920	24-924
	24-924#	24-936	24-938	24-938#	24-939	24-939#	24-960	24-960#	24-983	24-984	24-984#	24-993	24-994
	24-999	24-999#	24-:00	24-:00#	24-:98	24-:99	24-:99#	24-:21	24-:22	24-:22#	24-:32	24-:32#	24-:81
	24-<27	24-<27#	24-<83	24-<87	24-<87#	24-=26	24-=35	24-=47	24-=66	24-=74	24-=76	24-=76#	24->32
	24-?12	24-?12#	24-?47	24-?47#	24-?97	24-?12	24-?27	24-?27#	24-?95	24-A10	24-A10#	24-A60	24-A73
	24-841	24-856	24-856#	24-885	24-888	24-888#	24-C89	24-D34	24-D34#	24-E35	24-E35#	24-E87	24-F55
	24-F55#	24-F87	24-F87#	24-G40	24-G40#	24-G90	24-G90#	24-H60	24-H60#				
MSTLAB	1-C29#	7-18#	15-277#	15-283#	15-293#	15-299#	15-406#	15-412#	15-570#	15-602#	15-608#	15-618#	15-624#
	15-665#	15-699#	15-705#	15-713#	15-719#	15-865#	15-870#	15-889#	15-895#	15-904#	15-910#	15-919#	15-925#
	15-940#	15-:72#	15-:78#	16-101#	16-102#	16-107#	16-108#	16-109#	16-110#	16-111#	16-112#	16-113#	16-114#
	16-122#	16-123#	16-124#	16-125#	16-126#	16-127#	16-128#	16-129#	16-130#	16-131#	16-132#	16-133#	16-134#
	16-142#	16-143#	16-144#	16-145#	16-146#	16-147#	16-148#	16-149#	16-150#	16-151#	16-152#	16-153#	16-160#
	16-162#	16-163#	16-164#	16-165#	16-166#	16-167#	16-168#	16-169#	16-170#	16-171#	16-172#	16-173#	16-180#
	16-182#	16-183#	16-184#	16-185#	16-186#	16-187#	16-188#	16-189#	16-190#	16-191#	16-192#	16-199#	16-200#
	16-202#	16-203#	16-204#	16-205#	16-206#	16-207#	16-208#	16-209#	16-210#	16-217#	16-218#	16-219#	16-220#
	16-222#	16-223#	16-224#	16-225#	16-226#	16-227#	16-228#	16-229#	16-230#	16-237#	16-238#	16-239#	16-240#
	16-242#	16-243#	16-244#	17-11#	19-25#	19-28#	19-31#	19-34#	19-51#	19-74#	20-9#	20-16#	20-19#
22-10#	22-12#	22-13#	23-10#	24-16#	24-24#	24-27#	24-50#	24-52#	24-72#	24-84#	24-86#	24-90#	
24-117#	24-133#	24-148#	24-150#	24-184#	24-186#	24-226#	24-227#	24-232#	24-278#	24-279#	24-286#	24-303#	
24-316#	24-320#	24-337#	24-348#	24-350#	24-354#	24-371#	24-382#	24-384#	24-388#	24-405#	24-416#	24-418#	
24-456#	24-480#	24-482#	24-499#	24-501#	24-518#	24-520#	24-523#	24-551#	24-552#	24-565#	24-566#	24-577#	
24-588#	24-589#	24-594#	24-637#	24-638#	24-647#	24-648#	24-655#	24-680#	24-694#	24-695#	24-704#	24-706#	
24-731#	24-745#	24-746#	24-754#	24-756#	24-760#	24-778#	24-792#	24-793#	24-801#	24-803#	24-807#	24-838#	
24-856#	24-865#	24-867#	24-872#	24-898#	24-900#	24-913#	24-915#	24-920#	24-924#	24-936#	24-938#	24-939#	
24-983#	24-984#	24-993#	24-994#	24-999#	24-:00#	24-:98#	24-:99#	24-:21#	24-:22#	24-:32#	24-:81#	24-<27#	
24-<87#	24-=26#	24-=35#	24-=47#	24-=66#	24-=74#	24-=76#	24->32#	24-?12#	24-?47#	24-?97#	24-?12#	24-?27#	
24-A10#	24-A60#	24-A73#	24-B41#	24-B56#	24-B85#	24-B88#	24-C89#	24-D34#	24-E35#	24-E87#	24-F55#	24-F87#	
24-G90#	24-H60#												
MSTSTL	1-C21#	7-18#	15-277	15-277#	15-277#	15-283	15-283#	15-283#	15-293	15-293#	15-293#	15-299	15-299#
	15-406	15-406#	15-406#	15-412	15-412#	15-412#	15-570	15-570#	15-570#	15-602	15-602#	15-602#	15-608
	15-608#	15-618	15-618#	15-618#	15-624	15-624#	15-624#	15-659	15-659#	15-659#	15-655	15-665#	15-665#
	15-699#	15-699#	15-705	15-705#	15-705#	15-713	15-713#	15-713#	15-719	15-719#	15-719#	15-865	15-865#
	15-870	15-870#	15-870#	15-889	15-889#	15-889#	15-895	15-895#	15-895#	15-904	15-904#	15-904#	15-910
	15-910#	15-919	15-919#	15-919#	15-925	15-925#	15-925#	15-934	15-934#	15-934#	15-940	15-940#	15-:72
	15-:72#	15-:72#	15-:78	15-:78#	15-:78#	16-101	16-101#	16-102	16-102#	16-107	16-107#	16-108	16-109

16-109#	16-110	16-110#	16-111	16-111#	16-112	16-112#	16-113	16-113#	16-114	16-114#	16-115	16-115#	16-122
16-122#	16-123	16-123#	16-124	16-124#	16-125	16-125#	16-126	16-126#	16-127	16-127#	16-128	16-128#	16-129
16-129#	16-130	16-130#	16-131	16-131#	16-132	16-132#	16-133	16-133#	16-134	16-134#	16-141	16-141#	16-142
16-142#	16-143	16-143#	16-144	16-144#	16-145	16-145#	16-146	16-146#	16-147	16-147#	16-148	16-148#	16-149
16-149#	16-150	16-150#	16-151	16-151#	16-152	16-152#	16-153	16-153#	16-160	16-160#	16-161	16-161#	16-162
16-162#	16-163	16-163#	16-164	16-164#	16-165	16-165#	16-166	16-166#	16-167	16-167#	16-168	16-168#	16-169
16-169#	16-170	16-170#	16-171	16-171#	16-172	16-172#	16-173	16-173#	16-180	16-180#	16-181	16-181#	16-182
16-182#	16-183	16-183#	16-184	16-184#	16-185	16-185#	16-186	16-186#	16-187	16-187#	16-188	16-188#	16-189
16-189#	16-190	16-190#	16-191	16-191#	16-192	16-192#	16-199	16-199#	16-200	16-200#	16-201	16-201#	16-202
16-202#	16-203	16-203#	16-204	16-204#	16-205	16-205#	16-206	16-206#	16-207	16-207#	16-208	16-208#	16-209
16-209#	16-210	16-210#	16-217	16-217#	16-218	16-218#	16-219	16-219#	16-220	16-220#	16-221	16-221#	16-222
16-222#	16-223	16-223#	16-224	16-224#	16-225	16-225#	16-226	16-226#	16-227	16-227#	16-228	16-228#	16-229
16-229#	16-230	16-230#	16-237	16-237#	16-238	16-238#	16-239	16-239#	16-240	16-240#	16-241	16-241#	16-242
16-242#	16-243	16-243#	16-244	16-244#	17-11	17-11#	19-25	19-25#	19-28	19-28#	19-31	19-31#	19-34
19-34#	19-51	19-51#	19-74	19-74#	20-9	20-9#	20-16	20-16#	20-19	20-19#	21-11	21-11#	22-10
22-10#	22-12	22-12#	22-13	22-13#	23-10	23-10#	24-16	24-16#	24-24	24-24#	24-24#	24-27	24-27#
24-50	24-50#	24-50#	24-52	24-52#	24-72	24-72#	24-84	24-84#	24-84#	24-86	24-86#	24-90	24-90#
24-115	24-115#	24-115#	24-117	24-117#	24-133	24-133#	24-148	24-148#	24-148#	24-150	24-150#	24-184	24-184#
24-184#	24-186	24-186#	24-226	24-226#	24-226#	24-227	24-227#	24-232	24-232#	24-278	24-278#	24-278#	24-279
24-279#	24-286	24-286#	24-303	24-303#	24-314	24-314#	24-314#	24-316	24-316#	24-320	24-320#	24-337	24-337#
24-348	24-348#	24-348#	24-350	24-350#	24-354	24-354#	24-371	24-371#	24-382	24-382#	24-382#	24-384	24-384#
24-388	24-388#	24-405	24-405#	24-416	24-416#	24-416#	24-418	24-418#	24-422	24-422#	24-456	24-456#	24-480
24-480#	24-480#	24-482	24-482#	24-499	24-499#	24-499#	24-501	24-501#	24-518	24-518#	24-518#	24-520	24-520#
24-523	24-523#	24-551	24-551#	24-551#	24-552	24-552#	24-565	24-565#	24-565#	24-566	24-566#	24-577	24-577#
24-577#	24-578	24-578#	24-588	24-588#	24-588#	24-589	24-589#	24-594	24-594#	24-637	24-637#	24-637#	24-638
24-638#	24-647	24-647#	24-647#	24-648	24-648#	24-655	24-655#	24-680	24-680#	24-694	24-694#	24-694#	24-695
24-695#	24-704	24-704#	24-704#	24-706	24-706#	24-710	24-710#	24-731	24-731#	24-745	24-745#	24-745#	24-746
24-746#	24-754	24-754#	24-754#	24-756	24-756#	24-760	24-760#	24-778	24-778#	24-792	24-792#	24-792#	24-793
24-793#	24-801	24-801#	24-801#	24-803	24-803#	24-807	24-807#	24-838	24-838#	24-855	24-855#	24-855#	24-856
24-856#	24-865	24-865#	24-865#	24-867	24-867#	24-872	24-872#	24-898	24-898#	24-900	24-900#	24-913	24-913#
24-913#	24-915	24-915#	24-920	24-920#	24-924	24-924#	24-936	24-936#	24-936#	24-938	24-938#	24-939	24-939#
24-960	24-960#	24-983	24-983#	24-983#	24-984	24-984#	24-993	24-993#	24-993#	24-994	24-994#	24-999	24-999#
24-:00	24-:00#	24-:98	24-:98#	24-:98#	24-:99	24-:99#	24-:21	24-:21#	24-:21#	24-:22	24-:22#	24-:32	24-:32#
24-:81	24-:81#	24-<27	24-<27#	24-<83	24-<83#	24-<83#	24-<87	24-<87#	24-<87#	24-=26	24-=26#	24-=35	24-=35#
24-=35#	24-=47	24-=47#	24-=47#	24-=66	24-=66#	24-=66#	24-=74	24-=74#	24-=74#	24-=76	24-=76#	24->32	24->32#
24-?12	24-?12#	24-?47	24-?47#	24-?97	24-?97#	24-?97#	24-@12	24-@12#	24-@12#	24-@27	24-@27#	24-@95	24-@95#
24-@95#	24-A10	24-A10#	24-A60	24-A60#	24-A60#	24-A73	24-A73#	24-B41	24-B41#	24-B41#	24-B56	24-B56#	24-B85
24-B85#	24-B85#	24-B88	24-B88#	24-C89	24-C89#	24-C89#	24-D34	24-D34#	24-E35	24-E35#	24-E87	24-E87#	24-F55
24-F55#	24-F87	24-F87#	24-G40	24-G40#	24-G90	24-G90#	24-H60	24-H60#					
MSWORD	1-C94#	7-18#	8-17	8-17#	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8	9-8
	9-8	9-8	9-8	9-8	9-8	9-8	15-277	15-277	15-277	15-277#	15-283	15-283	15-283#
15-293	15-293	15-293	15-293#	15-299	15-299	15-299	15-299#	15-406	15-406	15-406	15-406#	15-412	15-412
15-412	15-412#	15-570	15-570	15-570	15-570#	15-602	15-602	15-602	15-602#	15-608	15-608	15-608	15-608#
15-618	15-618	15-618	15-618#	15-624	15-624	15-624	15-624#	15-659	15-659	15-659	15-659#	15-665	15-665
15-665	15-665#	15-699	15-699	15-699	15-699#	15-705	15-705	15-705	15-705#	15-713	15-713	15-713	15-713#
15-719	15-719	15-719	15-719#	15-865	15-865	15-865	15-865#	15-870	15-870	15-870	15-870#	15-889	15-889
15-889	15-889#	15-895	15-895	15-895	15-895#	15-904	15-904	15-904	15-904#	15-910	15-910	15-910	15-910#
15-919	15-919	15-919	15-919#	15-925	15-925	15-925	15-925#	15-934	15-934	15-934	15-934#	15-940	15-940
15-940	15-940#	15-:72	15-:72	15-:72	15-:72#	15-:78	15-:78	15-:78	15-:78#	24-24	24-24	24-24	24-24#
24-50	24-50	24-50	24-50#	24-84	24-84	24-84	24-84#	24-115	24-115	24-115	24-115#	24-148	24-148
24-148	24-148#	24-184	24-184	24-184	24-184#	24-226	24-226	24-226	24-226#	24-278	24-278	24-278	24-278#
24-314	24-314	24-314	24-314#	24-348	24-348	24-348	24-348#	24-382	24-382	24-382	24-382#	24-416	24-416
24-416	24-416#	24-480	24-480	24-480	24-480#	24-499	24-499	24-499	24-499#	24-518	24-518	24-518	24-518#
24-551	24-551	24-551	24-551#	24-565	24-565	24-565	24-565#	24-577	24-577	24-577	24-577#	24-588	24-588
24-588	24-588#	24-637	24-637	24-637	24-637#	24-647	24-647	24-647	24-647#	24-694	24-694	24-694	24-694#

	24-704	24-704	24-704	24-704#	24-745	24-745	24-745	24-745#	24-754	24-754	24-754	24-754#	24-792	24-792
	24-792	24-792#	24-801	24-801	24-801	24-801#	24-855	24-855	24-855	24-855#	24-865	24-865	24-865	24-865#
	24-913	24-913	24-913	24-913#	24-936	24-936	24-936	24-936#	24-983	24-983	24-983	24-983#	24-993	24-993
	24-993	24-993#	24-:98	24-:98	24-:98	24-:98#	24-:21	24-:21	24-:21	24-:21#	24-<83	24-<83	24-<83	24-<83#
	24-=26	24-=26	24-=26	24-=26#	24-=35	24-=35	24-=35	24-=35#	24-=47	24-=47	24-=47	24-=47#	24-=66	24-=66
	24-=66	24-=66#	24-=74	24-=74	24-=74	24-=74#	24-?97	24-?97	24-?97	24-?97#	24-@12	24-@12	24-@12	24-@12#
	24-@95	24-@95	24-@95	24-@95#	24-A60	24-A60	24-A60	24-A60#	24-B41	24-B41	24-B41	24-B41#	24-B85	24-B85
	24-B85	24-B85#	24-C89	24-C89	24-C89	24-C89#	25-16	25-16#	25-17	25-17#	25-18	25-18#	25-19	25-19#
	26-39	26-39												
MSXFER	1-@82#	7-18#												
MANUAL	1-:62#	7-18#												
MEMORY	1-:66#	7-18#												
OPEN	1-:71#	7-18#												
POINTE	1-:76#	7-18#	8-7											
PRINTB	1-<39#	7-18#	16-101	16-107	16-108	16-109	16-110	16-122	16-123	16-124	16-125	16-141	16-142	16-143
	16-144	16-160	16-161	16-162	16-163	16-164	16-180	16-181	16-182	16-183	16-199	16-200	16-201	16-217
	16-218	16-219	16-220	16-221	16-237	16-238	16-239							
PRINTF	1-<79#	7-18#	22-12	24-456	24-482	24-501	24-520							
PRINTS	1-=19#	7-18#												
PRINTX	1-=59#	7-18#	16-111	16-112	16-113	16-114	16-126	16-127	16-128	16-129	16-130	16-131	16-132	16-133
	16-145	16-146	16-147	16-148	16-149	16-150	16-151	16-152	16-165	16-166	16-167	16-168	16-169	16-170
	16-171	16-172	16-184	16-185	16-186	16-187	16-188	16-189	16-190	16-191	16-202	16-203	16-204	16-205
	16-206	16-207	16-208	16-209	16-222	16-223	16-224	16-225	16-226	16-227	16-228	16-229	16-240	16-241
	16-242	16-243												
READBU	1-=99#	7-18#												
READEF	1->03#	7-18#	19-25	19-28	19-31	19-34								
RFLAGS	1->08#	7-18#												
SETPRI	1->13#	7-18#	20-9	24-16										
SETVEC	1->18#	7-18#												
SLASH	1->24#	7-18#												
STARS	1->38#	7-18#												
SVC	1->52#	7-17#	7-18											
XFER	1-@12#	7-18#												
XFERF	1-@16#	7-18#												
XFERT	1-@20#	7-18#												