

DMP - 11
DMR - 11

M8203 STATIC DIAG. #2
CZDMSDO

AH-E235D-MC
FICHE 1 OF 2

NOV 1980
COPYRIGHT © 79-80
MADE IN USA



DMP - 11
DMR - 11

M8203 STATIC DIAG.#2
CZDMSDO

AH-E235D-MC
FICHE 2 OF 2

NOV 1980
COPYRIGHT © 79-80
MADE IN USA



5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-E234D-MC
PRODUCT NAME: CZDMSD0 M8203 STATIC DIAG #2
PRODUCT DATE: AUGUST 1980
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: 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	
2	
3	
4	
5	
6	
7	
8	1.0 INTRODUCTION
9	
10	2.0 HARDWARE REQUIREMENTS
11	
12	3.0 PRELIMINARY PROGRAM REQUIREMENTS
13	
14	4.0 GENERAL PROGRAM CONSIDERATIONS
15	4.1 DIAGNOSTIC SUPERVISOR
16	4.2 EXECUTION TIME
17	4.3 XXDP+
18	4.4 ACT/SLIDE
19	4.5 APT
20	4.6 MEMORY MANAGEMENT
21	4.7 MEMORY PARITY OPTION
22	4.8 ERROR LOGGING
23	
24	5.0 PROGRAM LOAD MEDIA
25	
26	6.0 OPERATING INSTRUCTIONS
27	6.1 LOADING AND STARTING PROCEDURES
28	6.1.1 LOADING PROCEDURES
29	6.1.2 STARTING PROCEDURES
30	6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
31	6.2 INITIAL DIALOGUE
32	6.3 PROGRAM OPTIONS
33	6.3.1 START COMMAND
34	6.3.1.1 TESTS SWITCH
35	6.3.1.2 PASS SWITCH
36	6.3.1.3 FLAGS SWITCH
37	6.3.1.4 END OF PASS SWITCH
38	6.3.1.5 EFFECT OF START COMMAND
39	6.3.2 RESTART COMMAND
40	6.3.2.1 TESTS, PASS, AND FLAG SWITCHES
41	6.3.2.2 UNITS SWITCH
42	6.3.2.3 EFFECT OF RESTART COMMAND
43	6.3.3 CONTINUE COMMAND
44	6.3.3.1 PASS SWITCH
45	6.3.3.2 FLAGS SWITCH
46	6.3.3.3 EFFECT OF CONTINUE COMMAND
47	6.3.4 PROCEED COMMAND
48	6.3.4.1 FLAGS SWITCH
49	6.3.4.2 EFFECT OF PROCEED COMMAND
50	6.3.5 ADD COMMAND
51	6.3.5.1 UNITS SWITCH
52	6.3.5.2 EFFECT OF ADD COMMAND
53	6.3.6 DROP COMMAND
54	6.3.6.1 UNITS SWITCH
55	6.3.6.2 EFFECT OF DROP COMMAND
56	6.3.7 PRINT COMMAND
57	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

H3254 AND H3255 TEST CONNECTORS (IF NOT PRESENT, SOME TESTS
WILL BE SKIPPED)

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
CZDMS-D-0
M8203 STATIC LOGIC TESTS - PART 2 OF 2
UNIT IS M8203
DR>

THE OPERATOR THEN RESPONDS 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 "N 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

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

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.

NOTE THAT IF THE MESSAGE 'TOO MANY UNITS' IS ISSUED, TWO OR MORE CORE IMAGES MUST BE CREATED (WITH DIFFERENT NAMES) TO TEST ALL UNITS.

NOTE THAT ALTHOUGH THE CHAINABLE IMAGE CAN BE EXECUTED ON A 15K MACHINE, THE ORIGINAL CCI CREATION MUST BE DONE ON A LARGE MACHINE, THE EXACT SIZE BEING DEPENDENT ON WHICH UPDATE UTILITY IS USED.

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.

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

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

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.

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

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

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 8 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. M8203 REG 11 (E121 SW10,9 , E134 SW9,10) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF REG 11 SWITCHES. THE ALLOWABLE RANGE IS 000-056, AND THE DEFAULT VALUE IS 000. BITS 1,2 ARE E121 SW10,9, AND BITS 3,5 ARE E134 SW9,10.

5. M8203 REG 15 (E134 SW1-8) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF REG 15 SWITCHES. THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000. BITS 0-7 ARE E134 SW1-8.

6. M8203 REG 16 (E121 SW1-8) : (O) 0 ?

THIS IS THE EXPECTED CONTENT (OCTAL) OF REG 16 SWITCHES.

THE ALLOWABLE RANGE IS 000-377, AND THE DEFAULT VALUE IS 000.
BITS 0-7 ARE E121 SW1-8.

7. TURNAROUND TYPE -
(0=H3254&H3255, 1=CABLE, 2=MOD LOC, 3=MOD REM, 4=NONE)
: (0) 0 ?

THIS INDICATOR TELLS THE PROGRAM WHETHER TEST CONNECTOR(S)
WILL BE MOUNTED SO THAT CERTAIN TESTS CAN BE RUN IN EXTERNAL
LOOPBACK MODE. THE ALLOWABLE ANSWERS ARE 0-4, AND THE
DEFAULT VALUE IS 0. 0 MEANS THAT THE H3254 AND H3255 TEST
CONNECTORS WILL BE USED. 1 MEANS THAT EXTERNAL TURNAROUND
WILL BE PROVIDED AT THE FAR END OF A CABLE ATTACHED TO THE
J1 OR J2 CONNECTOR. 2 MEANS THAT MODEM LOCAL LOOPBACK WILL
BE USED, AND 3 MEANS THAT MODEM REMOTE LOOPBACK WILL BE
USED. 4 MEANS THAT NO EXTERNAL TURNAROUND WILL BE PROVIDED.
WHEN 0 IS SELECTED, ALL TESTS WILL BE RUN, AND IF 1-4 IS
SELECTED, CERTAIN TESTS CANNOT BE RUN, AND THE PROGRAM WILL
TYPE THE NUMBER(S) OF TEST(S) TO BE SKIPPED.

8. PLEASE SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K;
'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K;
'6' FOR 500K; OR '7' FOR 1 MEG BAUD : (0) 4?

THIS IS THE BAUD RATE WHICH IS SELECTED IN THE SWITCH PACK ON
THE M8203. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS
4 (FOR 56K).

6.3.13 SOFTWARE PARAMETERS

FOUR SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THE M8203
STATIC LOGIC TESTS PROGRAM, PART 2. THESE QUESTIONS ARE THE
FOLLOWING:

1. IS MAN. INTERVEN. DESIRED TO MOUNT TEST CONNECTOR(S)
(L) N?

IF THE OPERATOR ANSWERS THE QUESTION WITH Y (YES), THE
PROGRAM WILL LATER PAUSE BEFORE TESTING EACH LOGICAL UNIT
AND INFORM THE OPERATOR TO INSTALL THE APPROPRIATE TEST
CONNECTOR(S) ON THAT UNIT, AND THEN PROCEED TO TEST THAT
UNIT. IF THE OPERATOR ANSWERS N (NO) TO THE ABOVE QUESTION,
THE PROGRAM WILL PERFORM TESTING ON ALL UNITS WITHOUT
ALLOWING MANUAL INTERVENTION BETWEEN UNITS. IN THIS CASE,
ALL TEST CONNECTOR(S) ON ALL UNITS SHOULD BE INSTALLED PRIOR
TO RUNNING THE PROGRAM.

2. SHOULD SWITCH PACK AND AX3-15 PRINTOUT BE ALLOWED (L) N
?

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE
PRINTOUT OF SWITCHES IN REGS 11,15,16 AND MODEM INTERFACE REG AX3-15
ON ANY PASS IN WHICH THE CORRESPONDING TESTS ARE RUN. THE
DEFAULT IS NO, WHICH ONLY ALLOWS THE PRINTOUT ON THE FIRST
PASS AFTER LOADING, IF THE TESTS ARE RUN.

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

3. SHOULD SWITCH PACK TESTS BE ALLOWED (L) N ?

IF THE OPERATOR ANSWERS YES, THE PROGRAM WILL ALLOW THE READING AND COMPARISON OF SWITCHES IN REGS 11,15,16 TO VALUES ENTERED INTO THE HARDWARE P-TABLE FOR THIS UNIT, IF THE CORRESPONDING TEST IS RUN. IF ALLOWED, SWITCH PACK ERRORS WILL BE REPORTED. THE DEFAULT IS NO, AND THE TESTS ARE NOT RUN.

4. MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : (0) 0 ?

THIS VALUE CONTROLS THE DURATION OF THE RECEIVER MESSAGE TIME-OUT IN A NUMBER OF TESTS WHICH SEND AND RECEIVE MESSAGES ON THE VARIOUS MODEM INTERFACES WITH EXTERNAL LOOPBACK. THE SMALLER THE VALUE, THE LONGER THE TIME-OUT (UP TO SEVERAL SECONDS). THE DEFAULT IS 0.

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 '# 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 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

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

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. 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
PCLR     = BIT6
STEPLU   = BIT4
LULOOK   = BIT3
ROMO     = BIT2
ROMI     = BIT1
STEPMP   = BIT0

*****
* CBUS 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
SELFR    = BIT5
HDX      = BIT4
MAINT1   = BIT3
MAINT2   = BIT2
SELSBY   = BIT1
```

```
58      ;*****
59      * OBUS REG 14
60      ;*****
61      TXEN      = BIT6
62      DISS1     = BIT5
63      RDAX      = BIT4
64      WAX       = BIT3
65      ENAX      = BIT2
66      AX2       = BIT1
67      AX1       = BIT0
68
69      ;*****
70      * OBUS REG 17
71      ;*****
72      CRC2      = BIT7
73      CRC1      = BIT6
74      IDLE      = BIT5
75      SECA      = BIT4
76      STRIP     = BIT3
77      RDALL     = BIT2
78      IERR      = BIT1
79      DDCMP     = BIT0
80
81      ;*****
82      * IBUS REG 10 - RECEIVER BUFFER
83      ;*****
84      RX7       = BIT7
85      RX6       = BIT6
86      RX5       = BIT5
87      RX4       = BIT4
88      RX3       = BIT3
89      RX2       = BIT2
90      RX1       = BIT1
91      RX0       = BIT0
92
93      ;*****
94      * IBUS REG 11
95      ;*****
96      OC        = BIT7
97      OACT      = BIT6
98      SW3       = BIT5
99      ORDY      = BIT4
100     SW2       = BIT3
101     SW1       = BIT2
102     SW0       = BIT1
103     UNRR      = BIT0
104
105     ;*****
106     * IBUS REG 12
107     ;*****
108     IC         = BIT7
109     IACT       = BIT6
110     LULP      = BIT5
111     IRDY      = BIT4
112     OVRR      = BIT3
113     RAB       = BIT2
114     EBLK      = BIT1
```

```
115      BCC      = BIT0
116
117      ;*****
118      * IBUS REG 13
119      ;*****
120      RING      = BIT7
121      DTR       = BIT6
122      RTS       = BIT5
123      HDX       = BIT4
124      MODR      = BIT3
125      CS        = BIT2
126      STBY      = BIT1
127      CARR      = BIT0
128
129      ;*****
130      * IBUS REG 14
131      ;*****
132      READY     = BIT7
133      TXEN      = BIT6
134      DISSI     = BIT5
135      RDAX      = BIT4
136      WAX       = BIT3
137      ENAX      = BIT2
138      AX2       = BIT1
139      AX1       = BIT0
140
141      ;*****
142      * IBUS REG 17
143      ;*****
144      SIGH      = BIT7
145      SIGO      = BIT6
146      TXDATA    = BIT5
147      OCOR      = BIT4
148      ICIR      = BIT3
149      TESTMD    = BIT2
150      MCLK      = BIT1
151      DDCMP     = BIT0
152
153      ;*****
154      * AX0-15 - USYRT REG 0 (READ ONLY)
155      ;*****
156      RX7       = BIT7
157      RX6       = BIT6
158      RX5       = BIT5
159      RX4       = BIT4
160      RX3       = BIT3
161      RX2       = BIT2
162      RX1       = BIT1
163      RX0       = BIT0
164
165      ;*****
166      * AX0-16 - USYRT REG 1 (READ ONLY)
167      ;*****
168      RERR      = BIT7
169      ASBC2     = BIT6
170      ASBC1     = BIT5
171      ASBC0     = BIT4
```

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

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250

V35 = BIT4
INTGRL = BIT3
C32ENB = BIT2
OP = BIT1
TEST = BIT0
AX315U = I422!XYZ!C32BCC!V35!INTGRL!OP

* AX3-16 - USYRT REG 7

TXLEN2 = BIT7
TXLEN1 = BIT6
TXLEN0 = BIT5
RXLEN2 = BIT2
RXLEN1 = BIT1
RXLEN0 = BIT0

8.0 TEST DESCRIPTIONS

TEST 1 - BIT STUFFING TEST
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,
* FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN
* CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS
* WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH
* ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED
* BY THE RECEIVER.
* PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,
* 177,376,377

TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST
*
* IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
* IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
* RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
* OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
* CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
* THEN, IC IS SET TO CLEAR THE ERROR, AND THIS IS VERIFIED.
*
* IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, AND A MASTER CLEAR
* IS ISSUED TO CLEAR THE ERROR, AND THIS IS VERIFIED.

TEST 3 - ABORT SEQUENCE TEST
*
* SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
* TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).
* AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
* SET THE ABORT BIT (REG 11).
* ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
* AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
* ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
* IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
* RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
*

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

```

* REPEAT THE ABOVE SEQUENCE, SET IC, AND CHECK THAT
* THIS CLEARS RAB.
*
* REPEAT THE ABOVE SEQUENCE, ISSUE MASTER CLEAR, CHECK THAT THIS
* CLEARS RAB.
*
;*****

;*****
TEST 4 - ABORT AND IDLE FLAGS TEST
*
* TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
* WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
* (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
* NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
;*****

;*****
TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
*
* A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
* IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
* UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
* CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
* THE DEVICE IDLES ABORT CHARACTERS.
;*****

;*****
TEST 6 - RECEIVER DISABLE TEST
*
* TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
* ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
* TO BE RECEIVED, IC IS SET.
* THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
* THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.
;*****

;*****
TEST 7 - ASSEMBLED BIT COUNT TEST
*
* THE FOLLOWING SEQUENCE IS PERFORMED 8 TIMES, EACH TIME USING A

```

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

* DIFFERENT TX CHAR LENGTH (FROM 2 TO 8 BITS) AND A RCV CHAR LENGTH = 8
* BITS :
* A MESSAGE IS INITIATED IN BIT MODE, NO CRC.
* 2 FLAGS ARE SENT, FOLLOWED BY 3 000 DATA CHARACTERS AND A
* TERMINATING FLAG. AFTER THE RECEIVER HAS RECEIVED THE MESSAGE, AX0-16
* IS READ TO RETRIEVE THE ASSEMBLED BIT COUNT. THIS COUNT IS CHECKED TO INSURE
* THAT IT IS CORRECT FOR THE TX CHAR LENGTH USED IN THAT TRANSMISSION.
;*****

;*****
TEST 8 - SECONDARY STATION ADDRESS BIT TEST

*
* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
* BIT MODE, AND THE SECA BIT (REG 17) IS SET.
* 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
* RECEIVED.
*
* NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
* WITH THE FIRST WORD OF DATA PATTERN T. 2 FLAGS ARE SENT,
* FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
* AND A TERMINATING FLAG.
* THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
* ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
*
* THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
* DATA PATTERN T.
* PATTERN T = 000,125,252,176,177
;*****

;*****
TEST 9 - RDALL (ALL PARTIES ADDRESS) BIT TEST

*
* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
* BIT MODE, AND THE SECA BIT IS SET.
* 2 FLAGS ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
* RECEIVED.
* NEXT, THE RDALL BIT IN REG 17 IS SET TO 1. 2 FLAGS
* ARE SENT, FOLLOWED BY 377, 125, AND A TERMINATING FLAG.
* THEN, THE REC'D DATA IS CHECKED TO MAKE SURE THAT 377
* IS REC'D AS THE FIRST DATA CHAR, FOLLOWED BY 125.
;*****

;*****

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

TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC

*
* THE LINE UNIT IS PLACED IN DDCMP MODE WITH NO ERROR DETECTION, AND 2
* SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
* TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
* CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
* THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
* AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
* 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
* AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
* THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
;*****

;*****
TEST 11 - SWITCH PACK PRINTOUT AND TEST

*
* - READ AND PRINT SWITCHES IN REG 11 (E121 SW10,9 , E134 SW9,10) :
* THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 1,2
* ARE E121 SW10,9 , AND BITS 3,5 ARE E134 SW9,10.
*
* - READ AND PRINT SWITCHES IN REG 15 (E134 SW1-8) :
* THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E134 SW1-8.
*
* - READ AND PRINT SWITCHES IN REG 16 (E121 SW1-8) :
* THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
* (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
* THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E121 SW1-8.
;*****

;*****
TEST 12 - REG AX3-15 PRINTOUT

*
* IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
* THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
* PRINT THE REG.
;*****

;*****
TEST 13 - CRC GENERATION TEST

*
* - CRC-16, CHAR MODE:

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

* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOP AND STEPLU
* TO CLOCK THE DATA. AT THE END OF THE MESSAGE THE
* PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
*
* - CRC-CCITT - 1'S PRESET:
* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
*
* - CRC-CCITT - 0'S PRESET:
* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
;*****

;*****
TEST 14 - CRC ERROR DETECTION TEST
*
* - CRC-16, CHAR MODE :
* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOP AND STEPLU
* TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
* THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
* ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
* AN ERROR.
*
* - CRC-CCITT - 1'S PRESET :
* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
*
* - CRC-CCITT - 0'S PRESET :
* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
;*****

;*****
TEST 15 - VRC PARITY GENERATION TEST
*
* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
* LAST 4 CHARS IT SHOULD = 0.
*
* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
* THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
* OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
* FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE

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

* LAST 4 CHARS IT SHOULD = 1.

* DATA PATTERN Q = 000,120,125,137,040,052,057,177

;*****

;*****

TEST 16 - VRC ERROR DETECTION TEST

* SUBTEST 1 - FORCING OF BCC USING ODD VRC

* THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.

* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER

* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM

* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING

* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE

* RECEIVED (INDICATING AN ERROR).

* SUBTEST 2 - FORCING OF BCC USING EVEN VRC

* THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.

* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER

* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM

* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING

* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE

* RECEIVED (INDICATING AN ERROR).

* DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
172,176.

;*****

;*****

TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC

* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A

* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR

* ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF

* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT

* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE

* SKIPPED FOR THAT UNIT.

;*****

;*****

TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC

* THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A

* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR

* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,

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

* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
* SKIPPED FOR THAT UNIT.

;*****

;*****

TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC

*
* THE RS232C RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
* AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
* AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
* OR A MODEM TEST MODE. THE MESSAGE CONSISTS
* OF 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
* P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
* PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.

;*****

;*****

TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC

*
* THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
* ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
* OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
* SKIPPED FOR THAT UNIT.

;*****

;*****

TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST

*
* THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 DOES NOT INHIBIT
* LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
* A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
* BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
* CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
* THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
* BE UNLOADED (ALL CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
* IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.

;*****

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 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED
*
* THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND
* THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.
* THE DATA IS SENT WITH LULOOK SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS
* THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED
* WHEN THE HDX BIT IS SET.
;*****

;*****
TEST 23 - INTERACTION OF MODEM CONTROL BITS
*
* THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT
* THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. OTHERWISE, THE TEST WILL
* BE SKIPPED FOR THE UNIT.
* THE FOLLOWING SUBTESTS ARE PERFORMED:
* - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED
*   STATE, WITH LULOOK SET TO 1. THEN, LULOOK IS CLEARED AND REG 13 IS READ
*   AND CHECKED FOR THE PROPER STATE, WITH LULOOK CLEARED.
*   REG 13 IS THEN LOADED WITH 0'S, AND READ AND CHECKED FOR THE
*   INITIALIZED STATE.
*   REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.
* - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.
* - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGQ SET.
* - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.
* - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.
* - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.
* - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.
* - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.
* - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.
* - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED INTO THE TX SILO, THE LINE
*   UNIT IS Clocked UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND
*   CHECKED FOR RTS, CS, CARR SET.
;*****

;*****
TEST 24 - DATA TEST - BIT MODE, NO ERR DET
*
* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
* INHIBITED. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376.
```

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

* 375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
* INHIBITED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
*
* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
*
* A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
* DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
* AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,

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

* 375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 28 - DATA TEST - CHAR MODE, CRC-16
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
* DETECTION. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* 8-BIT CHARACTERS ARE USED.
;*****

;*****
TEST 29 - DATA TEST - CHAR MODE, ODD VRC
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
;*****

;*****
TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,

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

* 375,373,367,357,337,277,177
* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
;*****

;*****
* TEST 31 - CONTIGUOUS ONES IN SEC. STA. ADRS. MODE, BIT MODE
*
* IN THIS TEST, A MESSAGE CONSISTING OF 5 ONES CHARS (377 OCT)
* IS SENT IN SECONDARY STATION ADDRESS MODE, WITH THE STATION ADRS
* FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
* THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
* ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
* ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
* AND TRANSMITTED AND RECEIVED CORRECTLY.
;*****

;*****
* TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
*
* IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
* DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
* FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
* AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
* AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1' IS
* SENT WITH ITS CORRESPONDING CRC.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,SYNCH,SYNCH
;*****

8.1 DATA PATTERNS USED

***** DATA PATTERN A *****
PATA:

.BYTE	125
.BYTE	252
.BYTE	000
.BYTE	377
.BYTE	001
.BYTE	002
.BYTE	004
.BYTE	010
.BYTE	020

628	.BYTE	040
629	.BYTE	100
630	.BYTE	200
631	.BYTE	376
632	.BYTE	375
633	.BYTE	373
634	.BYTE	367
635	.BYTE	357
636	.BYTE	337
637	.BYTE	277
638	.BYTE	177

***** DATA PATTERN Q *****

PATQ:	.BYTE	000
	.BYTE	120
	.BYTE	125
	.BYTE	137
	.BYTE	040
	.BYTE	052
	.BYTE	057
	.BYTE	177

***** DATA PATTERN R *****

PATR:	.BYTE	000
	.BYTE	100
	.BYTE	120
	.BYTE	124
	.BYTE	164
	.BYTE	172
	.BYTE	176
	.BYTE	177
	.BYTE	000
	.BYTE	100
	.BYTE	120
	.BYTE	124
	.BYTE	164
	.BYTE	172
	.BYTE	176

***** DATA PATTERN S *****

PATS:	.BYTE	000
	.BYTE	017
	.BYTE	036
	.BYTE	074
	.BYTE	170
	.BYTE	360
	.BYTE	037
	.BYTE	076
	.BYTE	174
	.BYTE	370
	.BYTE	077
	.BYTE	176
	.BYTE	374
	.BYTE	177
	.BYTE	376
	.BYTE	377

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

***** DATA PATTERN T *****
PATT: .BYTE 000
 .BYTE 125
 .BYTE 252
 .BYTE 176
 .BYTE 177

CZDMSD M8203 STATIC DIAG #2
PROGRAM DOCUMENT

MACRO V03.01 7-AUG-80 11:18:21 PAGE 6^{J 3}

SEQ 35

1

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 :

CZDMS 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 :

CZDMS 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

CZDMSD M8203 STATIC DIAG #2
PROGRAM DOCUMENT

MACRO V03.01 7-AUG-80 11:18:21 PAGE 8^{L 3}

SEQ 37

1
2
3
4
5
6
7
8
9
10

a

```
1          .TITLE CZDMSD M8203 STATIC TESTS #2
10         002000          .=2000
11
12
13
14
15
16
17         .MCALL  SVC
18 002000  SVC              ; INITIALIZE SUPERVISOR MACROS
19
20
21
22
23
24 002000  BGNMOD  LU2MOD
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
```

1
2
3
4
5
6
7 002000
8
10
11
12
13
14
16
17 002000
002000
002000 103
002001 132
002002 104
002003 115
002004 123
002005 000
002006 000
002007 000
002010
002010 104
002011
002011 060
002012
002012 000000
002014
002014 000055
002016
002016 036430
002020
002020 037374
002022
002022 002226
002024
002024 002256
002026
002026 040044
002030
002030 000000
002032
002032 000000
002034
002034 000000
002036
002036 000000
002040
002040 002124
002042
002042 000000
002044
002044 000000
002046

.SBTTL PROGRAM HEADER

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

POINTER BGNSW,BGNSFT,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 CZDMS,D,0,45.,0

LSNAME::

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

LSREV::

.ASCII /D/

LSDEPO::

.ASCII /O/

LSUNIT::

.WORD 0

LSTIML::

.WORD 45.

LSHPCP::

.WORD LSHARD

LSSPCP::

.WORD LSSOFT

LSHPTP::

.WORD LSHW

LSSPTP::

.WORD LSSW

LSLADP::

.WORD LSLAST

LSSSTA::

.WORD 0

LSCO::

.WORD 0

LSDTYP::

.WORD 0

LSAPT::

.WORD 0

LSDTP::

.WORD LSDISPATCH

LSPRIO::

.WORD 0

LSENV1::

.WORD 0

LSEXP1::

002046 000000
002050
002050 003
002051 003
002052
002052 000000
002054 000000
002056
002056 000000
002060
002060 003162
002062
002062 000000
002064
002064 000000
002066
002066 000000
002070
002070 023222
002072
002072 023140
002074
002074 000000
002076
002076 003170
002100
002100 104035
002102
002102 000000
002104
002104 022104
002106
002106 023136
002110
002110 023056
002112
002112 022076
002114
002114 000000
002116
002116 000000
002120
002120 000000

LSMREV:: .WORD 0
 .BYTE C\$REVISION
 .BYTE C\$EDIT
LSEF:: .WORD 0
 .WORD 0
LSSPC:: .WORD 0
L\$DEVP:: .WORD L\$DVTYP
LSREPP:: .WORD 0
L\$EXP4:: .WORD 0
L\$EXP5:: .WORD 0
LSAUT:: .WORD L\$AU
L\$DUT:: .WORD L\$DU
L\$LUN:: .WORD 0
L\$DESP:: .WORD L\$DESC
L\$LOAD:: EMT E\$LOAD
L\$ETP:: .WORD 0
L\$ICP:: .WORD L\$INIT
L\$CCP:: .WORD L\$CLEAN
L\$ACP:: .WORD L\$AUTO
L\$PRT:: .WORD L\$PROT
L\$TEST:: .WORD 0
L\$DLY:: .WORD 0
L\$HIME:: .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

[illegible]

```

        .WORD      32
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

```

[illegible]

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

.SBTTL DEFAULT HARDWARE P-TABLE

:/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
 :/ THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
 :/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
 :/

BGNHW DFPTBL

002224
 002224 000013
 002226
 002226

.WORD L10000-L\$HW/2
 L\$HW::
 DFPTBL::

.WORD 7
 .WORD 160170
 .WORD 300
 .WORD 5000
 .WORD 3
 .WORD 000
 .WORD 000
 .WORD 000
 .WORD 0
 .WORD 4
 .WORD 1

;MICROPROCESSOR TYPE = M8207
 ;M8207 CSR UNIBUS ADDRESS
 ;M8207 INTERRUPT VECTOR
 ;M8207 INTERRUPT PRIORITY LEVEL = 5
 ;LINE UNIT = M8203
 ;M8203 REG 11 (E121 SW10,9 , E134 SW9,10)
 ;M8203 REG 15 (E134 SW1-8)
 ;M8203 REG 16 (E121 SW1-8)
 ;H3254&H3255 USED
 ;BAUD RATE = 56 K
 ;M8207 RUN SWITCH (E28 SW7) IS ON

ENDHW

L10000:

```

1      .SBTTL  SOFTWARE P-TABLE
2
3      ;////////////////////////////////////
4      ;/  THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
5      ;/  PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
6      ;////////////////////////////////////
7
8      002254      BGNSW  SFPTBL
9      002254      000004
10     002256
11     002256
12     002256
13     002256
14
15     002256      MIFLAG: .WORD  0      ; =1 IF MAN. INTERVENTION DESIRED, =0 IF NOT
16     002260      PRNFLG: .WORD  0      ; =1 IF SW PACK AND AX3-15 PRINTOUT ALLOWED ALWAYS
17     002262      SWIFLG: .WORD  0      ; =1 IF SWITCH PACK VERIFICATION TEST SHOULD BE RUN
18     002264      TCOUNT: .WORD  0     ; INITIAL MSG TIME-OUT VALUE (0=LONGEST TIME-OUT)
19
20     ENDSW
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
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
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
2
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 002266

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS

: BIT DEFINITIONS

100000	B1T15== 100000
040000	B1T14== 40000
020000	B1T13== 20000
010000	B1T12== 10000
004000	B1T11== 4000
002000	B1T10== 2000
001000	B1T09== 1000
000400	B1T08== 400
000200	B1T07== 200
000100	B1T06== 100
000040	B1T05== 40
000020	B1T04== 20
000010	B1T03== 10
000004	B1T02== 4
000002	B1T01== 2
000001	B1T00== 1

001000	B1T9== B1T09
000400	B1T8== B1T08
000200	B1T7== B1T07
000100	B1T6== B1T06
000040	B1T5== B1T05
000020	B1T4== B1T04
000010	B1T3== B1T03
000004	B1T2== B1T02
000002	B1T1== B1T01
000001	B1T0== B1T00

: 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

```

; A NEW PASS HAS BEEN STARTED
; A POWER-FAIL/POWER-UP OCCURRED

```

:
 :
 : PRIORITY LEVEL DEFINITIONS

```

PRI07== 340
PRI06== 300
PRI05== 240
PRI04== 200
PRI03== 140
PRI02== 100
PRI01== 40
PRI00== 0

```

: OPERATOR FLAG BITS

```

EVL==          4
LOF==         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
*****

```

```

RUN      = BIT7
MCLR     = BIT6
STEPLU   = BIT4
LULOOOP  = 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	STR:P	=	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          OVRR    = 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          DISS1    = BIT5
160          000020          RDAX     = BIT4
```

161	000010	WAX	=	BIT3
162	000004	ENAX	=	BIT2
163	000002	AX2	=	BIT1
164	000001	AX1	=	BIT0

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

* IBUS REG 17

SIGR = BIT7
SIGQ = BIT6
TXDATA = BIT5
CCOR = 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
RABT = 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

```
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
277
278      004000
279      002000
280      001000
281      000400
282
283
284
285
286
287
288
289
290      004000
291      002000
292      001000
293      000400
294
295
296
297
298
299
300
301      002266
302      002270
303      002272
304      002274
305      002276
306      002300
307      002302
308      002304
309      002306
310      002310
311      002312
312      002314
313      002316
314      002320
315      002322
316      002324
317
318
319
320
321
322      100000
323
324      100000
325      100000
326
327      100000
328
329
330
331
```

```
*****
;* TX CONTROL BITS DEFINED ON WORD BASIS
*****
TXGOA  = BIT11
TXABT  = BIT10
TXEOM  = BIT9
TXSOM  = BIT8

*****
;* RCV CONTROL BITS DEFINED ON WORD BASIS
*****
RXOVR  = BIT11
RXABT  = BIT10
RXEBL  = BIT9
RXBCC  = BIT8

*****
;* ADDRESS EQUATES FOR REGISTER STORAGE TABLE (LUREG:)
*****
LUR10  = LUREG+0      ;LINE UNIT IBUS REG 10
LUR11  = LUREG+2      ;LINE UNIT IBUS REG 11
LUR12  = LUREG+4      ;LINE UNIT IBUS REG 12
LUR13  = LUREG+6      ;LINE UNIT IBUS REG 13
LUR14  = LUREG+10     ;LINE UNIT IBUS REG 14
LUR15  = LUREG+12     ;LINE UNIT IBUS REG 15
LUR16  = LUREG+14     ;LINE UNIT IBUS REG 16
LUR17  = LUREG+16     ;LINE UNIT IBUS REG 17
AX0.15 = LUREG+20     ;USYRT REG 0
AX0.16 = LUREG+22     ;USYRT REG 1
AX1.15 = LUREG+24     ;USYRT REG 2
AX1.16 = LUREG+26     ;USYRT REG 3
AX2.15 = LUREG+30     ;USYRT REG 4
AX2.16 = LUREG+32     ;USYRT REG 5
AX3.15 = LUREG+34     ;USYRT REG 6
AX3.16 = LUREG+36     ;USYRT REG 7

CHPCHK = BIT15
BCCCHK = BIT15
CRCCHK = BIT15
TCCHEK = BIT15
```

332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348

021000
122000

000001
000002

```
*****  
; * MICROINSTRUCTION DEFINITIONS *  
*****  
MVI0X  = 021000      ;MOVE IBUS TO OBUS*  
MVI0X  = 122000      ;MOVE IBUS* TO OBUS  
  
;***** ERROR1 BIT FLAG DEFINITIONS *****  
RRDYTO = BIT0  
WRDYTO = BIT1
```

```
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     002266  LUREG: .BLKW 16.
12
13     ;*****
14     ;* MISCELLANEOUS STORAGE
15     ;*****
16     002326  000000  SCRACH: .WORD 0          ;GEN'L PURPOSE SCRATCH WORD
17     002330  000000  LOGDEV: .WORD 0         ;LOGICAL DEVICE NUMBER
18     002332  000000  PSTACK: .WORD 0        ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
19     002334  000000  PRIOR: .WORD 0         ;CPU PRIORITY FOR PRINTOUT
20     002336  000000  SUBRPC: .WORD 0        ;PC OF SUBR CALL FOR ERROR REPORTS
21     002340  000000  INTFLG: .WORD 0        ;INTERRUPT RECEIVED FLAGS
22                                     ; BIT 0 FOR TX, BIT 1 FOR RCV
23     002342  000000  ERRFLG: .WORD 0        ;SUBROUTINE ERROR FLAG
24     002344  000000  TIMFLG: .WORD 0        ;EVENT TIME-OUT FLAG
25     002346  000000  RETADR: .WORD 0        ;SUBR ERROR RETURN ADDRESS
26     002350  000000  REDBYT: .WORD 0        ;LO BYTE CONTAINS BYTE READ FROM LU REG
27     002352  000000  WRIBYT: .WORD 0        ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
28     002354  000000  RAX15: .WORD 0         ;LO BYTE CONTAINS BYTE READ FROM REG 15
29     002356  000000  RAX16: .WORD 0         ;LO BYTE CONTAINS BYTE READ FROM REG 16
30     002360  000000  WAX15: .WORD 0         ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 15
31     002362  000000  WAX16: .WORD 0         ;LO BYTE CONTAINS BYTE TO LOAD INTO REG 16
32     002364  000000  REGNUM: .WORD 0        ;NUMBER (10-17) OF LINE UNIT REG BEING TESTED
33     002366  000000  AXNUM: .WORD 0        ;NUMBER (0-7) OF EXTENDED REG BYTE BEING TESTED
34     002370  000000  GOODAT: .WORD 0        ;STORAGE FOR EXPECTED DATA
35     002372  000000  BADDAT: .WORD 0        ;STORAGE FOR ACTUAL DATA
36     002374  000000  LOADAT: .WORD 0        ;CONTAINS TEST DATA LOADED INTO REG
37     002376  000000  FRSTIM: .WORD 0        ;FLAG=0 IF PROGRAM JUST LOADED
38     002400  000000  FRSPAS: .WORD 0        ;FLAG=0 IF FIRST PASS AFTER LOAD
39     002402  000000  STARES: .WORD 0        ;FLAG=0 IF FIRST TIME THRU AFTER STA OR RES
40     002404  000000  SAVE4: .WORD 0         ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
41     002406  000000  SAVE6: .WORD 0         ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
42     002410  000000  ERROR1: .WORD 0        ;SUBR ERROR BIT FLAGS (DEF'D IN GLOBAL EQUATES)
43     002412  000000  TXWORD: .WORD 0       ;BITS 0-11 CONTAIN DATA TO LOAD INTO TX SILO
44     002414  000000  RXWORD: .WORD 0       ;BITS 0-11 CONTAIN DATA READ FROM RCV SILO
45     002416  000000  DISILO: .WORD 0        ;CONTAINS CURRENT STATE OF DISSI IN BIT 5
46     002420  000000  CHPTYP: .WORD 0        ;USYRT CHIP TYPE, =0 FOR SIG, ELSE =1
47     002422  000000  MODINT: .WORD 0        ;MODEM INTERFACE SELECTION
48     002424  000000  SAVLEN: .WORD 0        ;SAVED TX AND RCV CHAR LENGTHS
49     002426  000000  DEVMAP: .WORD 0        ;BIT MAP OF ACTIVE DEVICES
50     002430  000000  DEVPTR: .WORD 0        ;DEVICE MAP BIT POINTER
51     002432  000000  UNIT: .WORD 0          ;CONTAINS UNIT NUMBER (1 TO N)
52     002434  000000  TSTNUM: .WORD 0        ;CONTAINS TEST NUMBER FOR SOME TESTS
53
54     ;***** CURRENT DEVICE PARAMETERS *****
55     002436  160170  MPCSR: .WORD 160170    ;POINTER TO MICROPROCESSOR CSR'S
56     002440  160171  BSEL1: .WORD 160171    ;POINTER TO BSEL1
57     002442  BSEL4:
```

```
58 002442 160174      SEL4:  .WORD 160174      ;POINTER TO SEL4
59 002444 160176      SEL6:  .WORD 160176      ;POINTER TO SEL6
60 002446 000300      MPIVEC: .WORD 300         ;MICROPROCESSOR INPUT INTERRUPT VECTOR
61 002450 000304      MPOVEC: .WORD 304        ;MICROPROCESSOR OUTPUT INTERRUPT VECTOR
62 002452 000240      MPRIOR: .WORD 240        ;MICROPROCESSOR DEVICE PRIORITY
63 002454 000000      LUSW11: .WORD 0          ;LINE UNIT SWITCH PACK #1
64 002456 000000      LUSW12: .WORD 0          ;LINE UNIT SWITCH PACK #2
65 002460 000000      LUSW13: .WORD 0          ;LINE UNIT SWITCH PACK #3
66 002462 000000      TSTCON: .WORD 0          ;TEST CONNECTOR INDICATOR
67 002464 000004      BDRATE: .WORD 4          ;BAUD RATE
68
69      ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
70 002466 000      REDDAT: .BYTE 0
71 002467 000      .BYTE 0
72 002470 000      .BYTE 0
73 002471 000      .BYTE 0
74 002472 000      .BYTE 0
75 002473 000      .BYTE 0
76 002474 000      .BYTE 0
77 002475 000      .BYTE 0
78
79      ;***** GEN'L PURPOSE SCRATCH STORAGE *****
80 002476 000000      REG0:  .WORD 0
81 002500 000000      REG1:  .WORD 0
82 002502 000000      REG2:  .WORD 0
83 002504 000000      REG3:  .WORD 0
84 002506 000000      REG4:  .WORD 0
85 002510 000000      REG5:  .WORD 0
86 002512 000000      REG6:  .WORD 0
87 002514 000000      REG7:  .WORD 0
88
89      ;***** SCRATCH STORAGE FOR MESSAGE REPORTING *****
90 002516 000000      TMP0:  .WORD 0
91 002520 000000      TMP1:  .WORD 0
92 002522 000000      TMP2:  .WORD 0
93 002524 000000      TMP3:  .WORD 0
94 002526 000000      TMP4:  .WORD 0
95 002530 000000      TMP5:  .WORD 0
96 002532 000000      TMP6:  .WORD 0
97 002534 000000      TMP7:  .WORD 0
98
99      ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
100 002536 000      UPBITS: .BYTE 000          ;MASK FOR REG 10
101 002536 000      .BYTE 056          ;MASK FOR REG 11
102 002537 056      .BYTE 000          ;MASK FOR REG 12
103 002540 000      .BYTE 257          ;MASK FOR REG 13
104 002541 257      .BYTE 100          ;MASK FOR REG 14
105 002542 100      .BYTE 377          ;MASK FOR REG 15
106 002543 377      .BYTE 377          ;MASK FOR REG 16
107 002544 377      .BYTE 306          ;MASK FOR REG 17
108 002545 306
109
110 002546 200      R14NRW: .BYTE 200          ;REG 14 NON-R/W BITS
111
112      ;***** MASKS FOR EXTENDED REGISTER NON-READ/WRITE BITS *****
113 002547 377      ANBITS: .BYTE 377          ;MASK FOR AX0-15
114 002547 377
```

115	002550	377	.BYTE	377	;MASK FOR AX0-16
116	002551	000	.BYTE	000	;MASK FOR AX1-15
117	002552	360	.BYTE	360	;MASK FOR AX1-16
118	002553	000	.BYTE	000	;MASK FOR AX2-15
119	002554	000	.BYTE	000	;MASK FOR AX2-16
120	002555	004	.BYTE	004	;MASK FOR AX3-15
121	002556	030	.BYTE	030	;MASK FOR AX3-16

122

123

;***** DATA PATTERN A *****
PATA:

124	002557		.BYTE	125
125	002557	125	.BYTE	252
126	002560	252	.BYTE	000
127	002561	000	.BYTE	377
128	002562	377	.BYTE	001
129	002563	001	.BYTE	002
130	002564	002	.BYTE	004
131	002565	004	.BYTE	010
132	002566	010	.BYTE	020
133	002567	020	.BYTE	040
134	002570	040	.BYTE	100
135	002571	100	.BYTE	200
136	002572	200	.BYTE	376
137	002573	376	.BYTE	375
138	002574	375	.BYTE	373
139	002575	373	.BYTE	367
140	002576	367	.BYTE	357
141	002577	357	.BYTE	337
142	002600	337	.BYTE	277
143	002601	277	.BYTE	177
144	002602	177		

145

146

;***** DATA PATTERN B *****
PATB:

147	002603		.BYTE	000
148	002603	000	.BYTE	000
149	002604	000	.BYTE	040
150	002605	040	.BYTE	100
151	002606	100	.BYTE	220
152	002607	220	.BYTE	000
153	002610	000	.BYTE	000
154	002611	000	.BYTE	051
155	002612	051		

156

157

;***** DATA PATTERN Q *****
PATQ:

158			.BYTE	000
159	002613	000	.BYTE	120
160	002614	120	.BYTE	125
161	002615	125	.BYTE	137
162	002616	137	.BYTE	040
163	002617	040	.BYTE	052
164	002620	052	.BYTE	057
165	002621	057	.BYTE	177
166	002622	177		

167

168

;***** DATA PATTERN R *****
PATR:

169	002623	000	.BYTE	000
170	002624	100	.BYTE	100
171	002625	120	.BYTE	120

172	002626	124	.BYTE	124
173	002627	164	.BYTE	164
174	002630	172	.BYTE	172
175	002631	176	.BYTE	176
176	002632	177	.BYTE	177
177	002633	000	.BYTE	000
178	002634	100	.BYTE	100
179	002635	120	.BYTE	120
180	002636	124	.BYTE	124
181	002637	164	.BYTE	164
182	002640	172	.BYTE	172
183	002641	176	.BYTE	176

;***** DATA PATTERN S *****

184				
185				
186	002642	000	PATS: .BYTE	000
187	002643	017	.BYTE	017
188	002644	036	.BYTE	036
189	002645	074	.BYTE	074
190	002646	170	.BYTE	170
191	002647	360	.BYTE	360
192	002650	037	.BYTE	037
193	002651	076	.BYTE	076
194	002652	174	.BYTE	174
195	002653	370	.BYTE	370
196	002654	077	.BYTE	077
197	002655	176	.BYTE	176
198	002656	374	.BYTE	374
199	002657	177	.BYTE	177
200	002660	376	.BYTE	376
201	002661	377	.BYTE	377

;***** DATA PATTERN T *****

202				
203				
204	002662	000	PATT: .BYTE	000
205	002663	125	.BYTE	125
206	002664	252	.BYTE	252
207	002665	176	.BYTE	176
208	002666	177	.BYTE	177

ENDPAT:
.EVEN

209
210
211
212
213
214
215
216
217

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

218				
219	002670	000400	MSG1: TXSOM	
220	002672	000400	TXSOM	
221	002674	000000	000	
222	002676	000125	125	
223	002700	000252	252	
224	002702	000377	377	
225	002704	000000	000	
226	002706	001000	TXEOM	
227	002710	001000	TXEOM	
228	002712	001000	TXEOM	

229	002714	001000	TXEOM
230			
231	002716	000400	MSG2: TXSOM
232	002720	000400	TXSOM
233	002722	000000	000
234	002724	000377	377
235	002726	001000	TXEOM
236	002730	001000	TXEOM
237			
238	002732	000001	MSG3: 001
239	002734	000001	001
240	002736	000001	001
241	002740	000001	001
242	002742	002000	TXABT
243	002744	000400	TXSOM
244	002746	000400	TXSOM
245	002750	000003	003
246	002752	000003	003
247	002754	000003	003
248	002756	000003	003
249	002760	000003	003

250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266

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

```
1          .SBTTL GLOBAL TEXT SECTION
2
3          :XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4          :X      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
5          :X      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
6          :X      MORE THAN ONE TEST.
7          :XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8
9          :*****
10         :* NAMES OF DEVICES SUPPORTED BY PROGRAM
11         :*****
12         003162      DEVTYP <M8203>
13
14         003162      LSDVTYP::
15         003162      .ASCIZ /M8203/
16         003165      115      070      062
17         003165      060      063      000
18
19
20
21
22
23
24
25         :*****
26         :* TITLE OF PROGRAM
27         :*****
28         003170      DESCRIPT      <M8203 STATIC LOGIC TESTS - PART 2 OF 2>
29         003170      LSDESC::
30         003170      .ASCIZ /M8203 STATIC LOGIC
31         003173      115      070      062
32         003173      060      063      040
33         003176      123      124      101
34         003201      124      111      103
35         003204      040      114      117
36         003207      107      111      103
37         003212      040      124      105
38         003215      123      124      123
39         003220      040      055      040
40         003223      120      101      122
41         003226      124      040      062
42         003231      040      117      106
43         003234      040      062      000
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
```

```
1      .SBTTL  GLOBAL SUBROUTINES
2
3      ;////////////////////////////////////////////////////
4      ;//  THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
5      ;////////////////////////////////////////////////////
6
7
8
9
10
11
12      ;*****
13      ;* STPCLK - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
14      ;*   EXECUTE AN INSTRUCTION WHICH IS PASSED IN THE WORD FOLLOWING THE CALL.
15      ;*****
16      STPCLK:
17          BISB    #ROMO!ROMI,@BSEL1      ;SET ROMO, ROMI BITS IN BSEL1
18          MOV     @ (SP),@SEL6           ;PUT INSTRUCTION INTO SEL6
19          BISB    #ROMO!ROMI!STEPMP,@BSEL1 ;SET ROMO, ROMI, STEPMP IN BSEL1
20          BICB    #ROMO!ROMI!STEPMP,@BSEL1 ;CLEAR ROMO, ROMI, STEPMP IN BSEL1
21          ADD     #2,(SP)                ;FIX UP RETURN PC
22          RTS     PC                    ;RETURN
23
24
25
26
27
28      ;*****
29      ;* MSTCLR - THIS SUBROUTINE ISSUES A MASTER CLEAR AND SETS LULOOP
30      ;*****
31      MSTCLR:
32          MOV     R1,-(SP)                ;SAVE R1
33          MOV     REGNUM,-(SP)            ;SAVE LU REG. NO.
34          MOV     #MCLR,@BSEL1           ;SET MASTER CLEAR BIT
35          BICB    #RUN!MCLR,@BSEL1       ;CLEAR RUN AND MCLR BITS
36          MOV     #20.,R1                ;INITIALIZE STALL COUNTER
37          2$:    NOP                     ;STALL IN LOOP FOR SEVERAL MICRO-SEC
38          DEC     R1
39          BNE     2$
40          BISB    #LULOOP,@BSEL1         ;SET LU LOOP
41          MOV     #13,REGNUM              ;SET LU REG NO. = 13
42          CLR     WRIBYT                  ;CLEAR REG 13
43          JSR     PC,WRITLU              ;RESTORE LU REG NO.
44          MOV     (SP)+,REGNUM            ;RESTORE R1
45          MOV     (SP)+,R1                ;CLEAR SAVED CHAR LENGTH FROM SETUP
46          CLR     SAVLEN                  ;RETURN
47          RTS     PC
48
49
50
51
52      ;*****
53      ;* READLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR
54      ;*   TO EXECUTE AN INSTRUCTION WHICH READS THE LINE UNIT REG WHOSE
55      ;*   NUMBER IS PASSED IN REGNUM, INTO REDBYT.
56      ;*****
57      READLU:
```

GLOBAL SUBROUTINES

```

58 003372 010146      MOV      R1,-(SP)      ;SAVE R1
59 003374 013701 002364  MOV      REGNUM,R1      ;GET LINE UNIT REG NUMBER
60 003400 006301      ASL      R1      ;SHIFT INTO SOURCE BITS 4-7
61 003402 006301      ASL      R1
62 003404 006301      ASL      R1
63 003406 006301      ASL      R1
64 003410 052701 000004  BIS      #4,R1      ;SET DESTINATION = BSEL4
65 003414 052701 021000  BIS      #MVIOX,R1      ;SET REST OF MOVE INSTRUCTION
66 003420 010137 003430  MOV      R1,2$      ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
67 003424 004737 003240  JSR      PC,STPCLK      ;EXECUTE MOVE INSTRUCTION
68 003430 000000      .WORD      0      ;INSTRUCTION GOES HERE
69 003432 117737 177004 002350 2$:  MOVB      @BSEL4,REDBYT      ;GET LU REG CONTENTS INTO REDBYT
70 003440 105037 002351      CLRB      REDBYT+1      ;CLR HI BYTE OF STORAGE
71 003444 012601      MOV      (SP)+,R1      ;RESTORE R1
72 003446 000207      RTS      PC      ;RETURN

```

73
74
75
76
77

```

78 *****
79 ;* WRITLU - THIS SUBROUTINE FORCES THE DMC11 OR KMC11 MICROPROCESSOR TO
80 ;* EXECUTE AN INSTRUCTION WHICH LOADS THE BYTE CONTAINED IN WRIBYT
81 ;* INTO THE LU REG WHOSE NUMBER IS PASSED IN REGNUM,
82 *****

```

```

83 003450      WRITLU:
84 003450 010146      MOV      R1,-(SP)      ;SAVE R1
85 003452 013701 002364  MOV      REGNUM,R1      ;GET LINE UNIT REG NUMBER
86 003456 052701 000100  BIS      #100,R1      ;SET SOURCE = BSEL4
87 003462 052701 122000  BIS      #MVIOX,R1      ;SET REST OF MOVE INSTRUCTION
88 003466 010137 003510  MOV      R1,2$      ;SET INSTRUCTION AS SUBROUTINE ARGUMENT
89 003472 105037 002353      CLRB      WRIBYT+1      ;CLR HI BYTE OF STORAGE
90 003476 113777 002352 176736  MOVB      WRIBYT,@BSEL4      ;LOAD BYTE INTO BSEL4
91 003504 004737 003240  JSR      PC,STPCLK      ;EXECUTE MOVE INSTRUCTION
92 003510 000000      .WORD      0
93 003512 012601      MOV      (SP)+,R1      ;RESTORE R1
94 003514 000207      RTS      PC      ;RETURN

```

95
96
97
98
99

```

100 *****
101 ;* GETREG - THIS SUBROUTINE READS THE LINE UNIT REGISTERS 10-17 INTO THE
102 ;* REGISTER STORAGE TABLE (LUREG:).
103 *****

```

```

104 003516 010146      GETREG: MOV      R1,-(SP)      ;SAVE R1
105 003520 013746 002364  MOV      REGNUM,-(SP)      ;SAVE CURRENT REG NO.
106 003524 012701 002266  MOV      #LUR10,R1      ;INIT POINTER TO REG STORAGE TABLE
107 003530 012737 000010 002364  MOV      #10,REGNUM      ;INIT LU REG NO. TO 10
108 003536 004737 003372      JSR      PC,READLU      ;READ A LINE UNIT REG
109 003542 113721 002350      MOVB      REDBYT,(R1)+      ;PUT BYTE READ INTO TABLE
110 003546 105021      CLRB      (R1)+      ;CLEAR UPPER BYTE OF TABLE ENTRY
111 003550 005237 002364      INC      REGNUM      ;INCREMENT REG NO.
112 003554 023727 002364 000020  CMP      REGNUM,#20      ;SEE IF ALL REGS READ YET
113 003562 002765      BLT      3$      ;BR IF NOT
114 003564 012637 002364      MOV      (SP)+,REGNUM      ;RESTORE CURRENT REG NO.

```

115 003570 012601
116 003572 000207MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

117

118

119

120

121

122

123

124

125

126

127

128

129

130 003574

131 003574 152777 000006 176636

132 003602 017677 000000 176634

133 003610 152777 000206 176622

134 003616 062716 000002

135 003622 000207

136

137

138

139

140

141

142

143

144

145

146

147

148 003624 010146

149 003626 013746 002364

150 003632 042737 000001 002410

151 003640 012737 000014 002364

152 003646 113737 002366 002352

153 003654 006237 002352

154 003660 152737 000024 002352

155 003666 053737 002416 002352

156 003674 004737 003450

157 003700 005001

158 003702 004737 003372 6\$:

159 003706 132737 000200 002350

160 003714 001006

161 003716 005201

162 003720 001370

163 003722 052737 000001 002410

164 003730 000424

165 003732 012737 000015 002364 9\$:

166 003740 004737 003372

167 003744 113737 002350 002354

168 003752 105037 002355

169 003756 012737 000016 002364

170 003764 004737 003372

171 003770 113737 002350 002356

* 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.

LOOPIN:

BISB #ROMO!ROMI,BSEL1 ;SET ROMO, ROMI BITS IN BSEL1
MOV @ (SP),@SEL6 ;PUT MICROINSTRUCTION INTO SEL6
BISB #RUN!ROMO!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*****
* 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.
*****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
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 003776 105037 002357
173 004002 012637 002364
174 004006 012601
175 004010 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

* 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.

185

186 004012 010146
187 004014 013746 002364
188 004020 042737 000002 002410
189 004026 012737 000014 002364
190 004034 113737 002366 002352
191 004042 006237 002352
192 004046 053737 002416 002352
193 004054 004737 003450
194 004060 012737 000015 002364
195 004066 105037 002361
196 004072 113737 002360 002352
197 004100 004737 003450
198 004104 005237 002364
199 004110 105037 002363
200 004114 113737 002362 002352
201 004122 004737 003450
202 004126 012737 000014 002364
203 004134 113737 002366 002352
204 004142 006237 002352
205 004146 152737 000014 002352
206 004154 053737 002416 002352
207 004162 004737 003450
208 004166 005001
209 004170 004737 003372
210 004174 132737 000200 002350
211 004202 001005
212 004204 005201
213 004206 001370
214 004210 052737 000002 002410
215 004216 012637 002364
216 004222 012601
217 004224 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
6\$: 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 004226 010146
228 004230 013746 002366

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

```
229 004234 012737 015250 002516      MOV    #DH5,TMPO      ;SET AX LO BYTE NO.
230 004242 032737 000001 002366      BIT    #BIT0,AXNUM   ;SEE IF LO OR HI BYTE
231 004250 001403                BEQ     1$              ;BR IF LO BYTE
232 004252 012737 015253 002516      MOV    #DH6,TMPO      ;SET AX HI BYTE NO.
233 004260 004737 003516 1$:      JSR     PC,GETREG      ;READ AND STORE REGS 10-17
234 004264 142777 000010 176146      BICB   #LULOOP,ABSEL1 ;CLEAR LULOOP
235 004272 012701 002306      MOV    #AX0.15,R1    ;INIT POINTER TO REG STORAGE TABLE
236 004276 005037 002366      CLR     AXNUM          ;INIT AX REG BYTE NO. TO 0
237 004302 004737 003624 3$:      JSR     PC,READAX      ;READ 2 AX REG BYTES
238 004306 113721 002354      MOVB    RAX15,(R1)+    ;PUT LO BYTE READ INTO TABLE
239 004312 105021                CLRB                    ;CLEAR UPPER BYTE OF TABLE ENTRY
240 004314 113721 002356      MOVB    RAX16,(R1)+    ;PUT HI BYTE READ INTO TABLE
241 004320 105021                CLRB                    ;CLEAR UPPER BYTE OF TABLE ENTRY
242 004322 062737 000002 002366      ADD     #2,AXNUM        ;INCR AX REG BYTE NO.
243 004330 023727 002366 000010      CMP     AXNUM,#10      ;SEE IF ALL REGS READ YET
244 004336 002761                BLT     3$              ;BR IF NOT
245 004340 012637 002366      MOV     (SP)+,AXNUM      ;RESTORE CURRENT AX REG BYTE NO.
246 004344 012601                MOV     (SP)+,R1        ;RESTORE R1
247 004346 013737 002366 002520      MOV     AXNUM,TMP1
248 004354 006237 002520      ASR     TMP1              ;GET EXTENDED REG NO. FOR PRINTOUT
249 004360 000207                RTS     PC              ;RETURN
```

```
250
251
252
253
254
255 *****
256 * OSIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ORDY (REG 11)
257 * AND OCOR (REG 17) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
258 * AS PASSED IN BIT 0 (ORDY) AND BIT 1 (OCOR) OF THE WORD FOLLOWING THE
259 * CALL.
260 * IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST, AT THE ADDRESS IN
261 * RETADR.
262 *****
```

```
263 004362 013746 002364      OSIRDY: MOV    REGNUM,-(SP)    ;SAVE LU REG NO.
264 004366 013746 002336      MOV    SUBRPC,-(SP)
265 004372 005737 002336      TST     SUBRPC          ;SEE IF THIS IS A NESTED CALL
266 004376 001006                BNE     1$              ;BR IF YES
267 004400 016637 000004 002336      MOV    4(SP),SUBRPC
268 004406 162737 000004 002336      SUB     #4,SUBRPC      ;GET PC OF SUBROUTINE CALL
269 004414 012737 000011 002364 1$:  MOV    #11,REGNUM      ;SET REG NO. TO 11
270 004422 004737 003372      JSR     PC,READLU      ;READ REG 11
271 004426 032776 000001 000004      BIT    #BIT0,24(SP)    ;GET EXPECTED STATE OF ORDY
272 004434 001413                BEQ     3$              ;BR IF EXPECTED ORDY = 0
273 004436 132737 000020 002350      BITB   #ORDY,REDBYT    ;SEE IF ORDY = 1
274 004444 001022                BNE     9$              ;BR IF ORDY = 1
275 004446 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
276      ;REPORT ORDY NOT SET
277      ERRDF 7,EM7,ERR4

      TRAP    CSERDF
      .WORD   7
      .WORD   EM7
      .WORD   ERR4

278 004462 000451                BR     16$              ;TAKE ERROR RETURN
279 004464 132737 000020 002350 3$:  BITB   #ORDY,REDBYT    ;SEE IF ORDY = 0
280 004472 001407                BEQ     9$              ;BR IF ORDY = 0
281 004474 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
```

```
282                                     ;REPORT ORDY NOT CLEARED
283 004500 ERRDF 8,EM8,ERR4
      004500 104455
      004502 000010
      004504 013360
      004506 016434
284 004510 000436
285 004512 012737 000017 002364 9$: BR 16$ ;TAKE ERROR RETURN
286 004520 004737 003372 JSR PC,READLU ;SET REG NO. = 17
287 004524 132776 000002 000004 BITB #BIT1,24(SP) ;READ LU REG 17
288 004532 001413 BEQ 12$ ;GET EXPECTED STATE OF OCOR
289 004534 132737 000020 002350 BITB #OCOR,REDBYT ;BR IF EXPECTED OCOR = 0
290 004542 001031 BNE 20$ ;SEE IF OCOR = 1
291 004544 004737 004226 JSR PC,GETALL ;BR IF OCOR = 1
292                                     ;GET REGS FOR PRINTOUT
293                                     ;REPORT OCOR NOT SET
      004550 ERRDF 9,EM9,ERR4
      004550 104455
      004552 000011
      004554 013401
      004556 016434
294 004560 000412
295 004562 132737 000020 002350 12$: BR 16$ ;TAKE ERROR RETURN
296 004570 001416 BEQ 20$ ;SEE IF OCOR = 0
297 004572 004737 004226 JSR PC,GETALL ;BR IF OCOR = 0
298                                     ;GET REGS FOR PRINTOUT
299                                     ;REPORT OCOR NOT CLEARED
      004576 ERRDF 10,EM10,ERR4
      004576 104455
      004600 000012
      004602 013416
      004604 016434
300 004606 016637 000002 002364 16$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.
301 004614 013706 002332 MOV PSTACK,SP ;RESTORE STACK POINTER TO BASE LEVEL
302 004620 013746 002346 MOV RETADR,-(SP) ;FIX ERROR RETURN PC
303 004624 000407 BR 23$
304 004626 062766 000002 000004 20$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
305 004634 012637 002336 MOV (SP)+,SUBRPC
306 004640 012637 002364 MOV (SP)+,REGNUM ;RESTORE LU REG NO.
307 004644 000207 23$: RTS PC ;RETURN
308
309
310
311
312
313
314
315
316 004646 010146
317 004650 012701 000310
318 004654 005301
319 004656 001376
320 004660 012601
321 004662 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 ;* STALL - THIS SUBROUTINE STALLS FOR ABOUT A MICRO-SEC.
329 ;*****
330 004664 000240 STALL: NOP
331 004666 000240 NOP
332 004670 000240 NOP
333 004672 000207 RTS PC
334
335
336
337
338
339 ;*****
340 ;* LDTXSI - THIS SUBROUTINE LOADS THE TX SILO (REGS 10,11) WITH THE DATA PASSED
341 ;* IN BITS 0-11 OF TXWORD.
342 ;*****
343 004674 013746 002364 LDTXSI: MOV REGNUM, -(SP) ;SAVE LU REG NO.
344 004700 042737 170000 002412 BIC #170000, TXWORD ;CLEAR UNUSED BITS
345 004706 012737 000011 002364 MOV #11, REGNUM ;SET REG NO. = 11
346 004714 113737 002413 002352 MOVB TXWORD+1, WRIBY ;SET DATA TO BE WRITTEN INTO REG 11
347 004722 004737 003450 JSR PC, WRITLU ;LOAD DATA INTO REG 11
348 004726 012737 000010 002364 MOV #10, REGNUM ;SET REG NO. = 10
349 004734 113737 002412 002352 MOVB TXWORD, WRIBY ;SET DATA TO BE WRITTEN INTO REG 10
350 004742 004737 003450 JSR PC, WRITLU ;LOAD DATA INTO REG 10
351 004746 012637 002364 MOV (SP)+, REGNUM ;RESTORE LU REG NO.
352 004752 000207 RTS PC ;RETURN
353
354
355
356
357
358 ;*****
359 ;* STPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NO. OF CYCLES PASSED
360 ;* IN BITS 0-14 OF THE WORD FOLLOWING THE CALL.
361 ;* IF BIT 15 = 1, A CHECK IS MADE TO DETERMINE IF THE USYRT CHIP TYPE
362 ;* REQUIRES DECREMENTING THE NO. OF CYCLES BY 1.
363 ;*****
364 004754 010146 STPLU: MOV R1, -(SP) ;SAVE R1
365 004756 017601 000002 MOV @2(SP), R1 ;GET DESIRED NO. OF CYCLES
366 004762 001426 BEQ 6$ ;IF DESIRED CYCLES = 0, RETURN
367 004764 100006 BPL 2$ ;BR IF CHIP TYPE CHECK NOT NECESSARY
368 004766 042701 100000 BIC #BIT15, R1 ;CLEAR FLAG BIT
369 004772 005737 002420 TST CHPTYP ;SEE IF SIG USYRT
370 004776 001401 BEQ 2$ ;BR IF YES
371 005000 005301 DEC R1 ;DECREMENT CYCLE COUNT
372 005002 152777 000010 175430 2$: BISB #LULOOP, @BSEL1 ;SET LU LOOP BIT
373 005010 152777 000020 175422 3$: BISB #STEPLU, @BSEL1 ;SET THE STEPLU BIT (CLOCK THE TRANSMITTER)
374 005016 004737 004664 JSR PC, STALL ;STALL
375 005022 142777 000020 175410 BICB #STEPLU, @BSEL1 ;CLEAR THE STEPLU BIT (CLOCK THE RECEIVER)
376 005030 004737 004664 JSR PC, STALL ;STALL
377 005034 005301 DEC R1 ;DECREMENT CYCLE COUNTER
378 005036 001364 BNE 3$ ;BR IF NOT DONE YET
379 005040 062766 000002 000002 6$: ADD #2, 2(SP) ;FIX UP RETURN PC
380 005046 012601 MOV (SP)+, R1 ;RESTORE R1
381 005050 000207 RTS PC ;RETURN
382
383
```

```
384
385
386
387
388
389
390
391
392 005052 013746 002364
393 005056 013746 002336
394 005062 005737 002336
395 005066 001006
396 005070 016637 000004 002336
397 005076 162737 000004 002336
398 005104 012737 000011 002364
399 005112 004737 003372
400 005116 032776 000001 000004
401 005124 001413
402 005126 132737 000100 002350
403 005134 001031
404 005136 004737 004226
405
406 005142
    005142 104455
    005144 000013
    005146 013437
    005150 016434
407 005152 000412
408 005154 132737 000100 002350
409 005162 001416
410 005164 004737 004226
411
412 005170
    005170 104455
    005172 000014
    005174 013454
    005176 016434
413 005200 016637 000002 002364
414 005206 013706 002332
415 005212 013746 002346
416 005216 000407
417 005220 062766 000002 000004
418 005226 012637 002336
419 005232 012637 002364
420 005236 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
        MOV     #11, REGNUM      ;SET REG NO. = 11
        JSR     PC, READLU       ;READ REG 11
        BIT     #BIT0, 24(SP)    ;GET EXPECTED STATE OF OACT
        BEQ     3$              ;BR IF EXPECTED OACT = 0
        BITB    #OACT, REDBYT    ;SEE IF OACT = 1
        BNE     9$              ;BR IF OACT = 1
        JSR     PC, GETALL       ;GET REGS FOR PRINTOUT
;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

        TRAP    C$ERDF
        .WORD   11
        .WORD   EM11
        .WORD   ERR4

        BR     6$              ;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
        RTS     PC              ;RETURN

        TRAP    C$ERDF
        .WORD   12
        .WORD   EM12
        .WORD   ERR4

*****
* 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 USVRT. THE PROGRAM MONITORS ORDY, OCOR, AND OACT FOR VALID STATES.
* THROUGHOUT THE PROCESS.
*****
```

```
433                                     ;* IF THE SUBROUTINE DETECTS AN ERROR, A RETURN IS MADE TO THE TEST, AT THE
434                                     ;* ADDRESS CONTAINED IN RETADR.
435                                     ;*****
436 005240 010146 INITRN: MOV R1,-(SP) ;SAVE R1
437 005242 013746 002364 MOV REGNUM,-(SP) ;SAVE LU REG NO.
438 005246 013746 002366 MOV AXNUM,-(SP) ;SAVE AX BYTE NO.
439 005252 016637 000006 002336 MOV 6(SP),SUBRPC
440 005260 162737 000004 002336 SUB #4,SUBRPC ;GET PC OF SUBR CALL
441 005266 004737 003276 JSR PC,MSTCLR ;ISSUE A MASTER CLEAR
442 005272 004737 004362 JSR PC,OSIRDY ;CHECK ORDY=1, OCOR=0
443 005276 000001 1 JSR PC,OACTIV ;CHK OACT=0
444 005300 004737 005052 0
445 005304 000000 0 MOV #4,AXNUM ;SET AX BYTE NO. = 4 FOR AX2
446 005306 012737 000004 002366 MOVB @6(SP),WAX15 ;SET DATA BYTE TO LOAD INTO AX2-15
447 005314 117637 000006 002360 MOV #TXSOM,TXWORD ;SET TSOM BIT
448 005322 012737 000400 002412 MOVB WAX15,TXWORD ;SET SYNCH CHAR
449 005330 113737 002360 002412 CLW WAX16
450 005336 005037 002362 JSR PC,WRITAX ;LOAD AX2
451 005342 004737 004012 MOV #17,REGNUM ;SET REG NO. = 17
452 005346 012737 000017 002364 ADD #2,6(SP) ;INCR POINTER TO NEXT DATA BYTE
453 005354 062766 000002 000006 MOVB @6(SP),WRIBYT ;SET DATA BYTE TO LOAD INTO REG 17
454 005362 117637 000006 002352 JSR PC,WRITLU ;LOAD REG 17
455 005370 004737 003450 JSR PC,LDTXSI ;LOAD THE SILO WITH SOM CHAR
456 005374 004737 004674 JSR PC,LDTXSI ;LOAD ANOTHER SOM INTO SILC
457 005400 004737 004674 JSR PC,WAIT50 ;WAIT FOR DATA TO RIPPLE
458 005404 004737 004646 JSR PC,OSIRDY ;CHK ORDY=1, OCOR=1
459 005410 004737 004362 3 JSR PC,OACTIV ;CHK FOR OACT = 0
460 005414 000003 0
461 005416 004737 005052 CLR R1 ;INIT CYCLE COUNTER
462 005422 000000 0 MOV #11,REGNUM ;SET LU REG NO. = 11
463 005424 005001 000011 002364 BISB #LULOOP,@BSEL1 ;SET LINE UNIT LOOP BIT
464 005426 012737 000010 174776 6$: BISB #STEPLU,@BSEL1 ;SET CLOCK BIT
465 005434 152777 000010 174776 JSR PC,STALL ;STALL FOR MICRO-SEC
466 005442 152777 000020 174770 JSR PC,READLU ;READ REG 11
467 005450 004737 004664 BITB #OACT,REDBYT ;SEE IF OACT = 1 YET
468 005454 004737 003372 BNE 9$ ;BR IF OACT = 1
469 005460 132737 000100 002350 BICB #STEPLU,@BSEL1 ;CLEAR CLOCK BIT
470 005466 001014 0 JSR PC,STALL ;STALL FOR A MICRO-SEC
471 005470 142777 000020 174742 INC R1 ;INCR CYCLE COUNT
472 005476 004737 004664 CMP R1,#3 ;SEE IF 3 CYCLES DONE YET
473 005502 005201 000003 BLT 6$ ;BR IF NOT
474 005504 020127 000003 JSR PC,OACTIV ;CHK FOR OACT = 1
475 005510 002751 1
476 005512 004737 005052 MOV #17,REGNUM ;SET REG NO. = 17
477 005516 000001 0 CLR CHPTYP ;CLEAR USYRT CHIP INDICATOR
478 005520 012737 000017 002364 9$: JSR PC,READLU ;READ REG 17
479 005526 005037 002420 BITB #OCOR,REDBYT ;CHK FOR OCOR CLEARED YET
480 005532 004737 003372 BEQ 12$ ;BR IF YES - IT IS SIG CHIP
481 005536 132737 000020 002350 MOV #1,CHPTYP ;SET INDICATOR FOR OTHER CHIP TYPE
482 005544 001403 0 BICB #STEPLU,@BSEL1 ;CLEAR CLOCK BIT
483 005546 012737 000001 002420 12$: JSR PC,STALL ;STALL FOR MICRO-SEC
484 005554 142777 000020 174656 JSR PC,OSIRDY ;CHK FOR ORDY = 1, OCOR = 0
485 005562 004737 004664 1
486 005566 004737 004362 ADD #2,6(SP) ;FIX UP RETURN PC
487 005572 000001 0 MOV (SP)+,AXNUM ;RESTORE AX BYTE NO.
488 005574 062766 000002 000006
489 005602 012637 002366
```

```
490 005606 012637 002364      MOV      (SP)+,REGNUM      ;RESTORE LU REG NO.
491 005612 012601              MOV      (SP)+,R1          ;RESTORE R1
492 005614 005037 002336      CLR      SUBRPC          ;CLEAR SUBR CALL PC
493 005620 000207              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 005622 010146
511 005624 010246
512 005626 016637 000004 002336
513 005634 162737 000004 002336
514 005642 017637 000004 002412
515 005650 004737 004674
516 005654 004737 004646
517 005660 062766 000002 000004
518 005666 005001
519 005670 017602 000004
520 005674 005702
521 005676 100006
522 005700 042702 100000
523 005704 005737 002420
524 005710 001401
525 005712 005302
526 005714 004737 005052
527 005720 000001
528 005722 020102
529 005724 001410
530 005726 004737 004362
531 005732 000003
532 005734 004737 004754
533 005740 000001
534 005742 005201
535 005744 000763
536 005746 004737 004362
537 005752 000001
538 005754 062766 000002 000004
539 005762 005037 002336
540 005766 012602
541 005770 012601
542 005772 000207
```

```
TXCHAR: MOV      R1,-(SP)      ;SAVE R1
        MOV      R2,-(SP)      ;SAVE R2
        MOV      4(SP),SUBRPC    ;GET PC OF SUBR CALL
        SUB      #4,SUBRPC      ;GET DATA TO BE TRANSMITTED
        MOV      @4(SP),TXWORD  ;LOAD THE TX SILO WITH THE DATA
        JSR      PC,LDTXSI      ;WAIT FOR DATA TO RIPPLE DOWN SILO
        JSR      PC,WAIT50
        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
        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 * ISIRDY - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF ICIR (REG 17)
551 * AND IRDY (REG 12) AND REPORTS AN ERROR IF EITHER IS NOT PROPERLY SET
552 * AS PASSED IN BIT 0 (ICIR) AND BIT 1 (IRDY) OF THE WORD FOLLOWING THE
553 * CALL.
554 * IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS
555 * IN RETADR.
556 *****
556 005774 013746 002364 ISIRDY: MOV REGNUM,-(SP) ;SAVE LU REG NO.
557 006000 013746 002336 MOV SUBRPC,-(SP)
558 006004 005737 002336 TST SUBRPC ;SEE IF THIS IS A NESTED CALL
559 006010 001006 BNE 1$ ;BR IF YES
560 006012 016637 000004 002336 MOV 4(SP),SUBRPC
561 006020 162737 000004 002336 SUB #4,SUBRPC ;GET PC OF SUBR CALL
562 006026 012737 000012 002364 1$: MOV #12,REGNUM ;SET REG NO. TO 12
563 006034 004737 003372 JSR PC,READLU ;READ REG 12
564 006040 032776 000002 000004 BIT #BIT1,24(SP) ;GET EXPECTED STATE OF IRDY
565 006046 001413 BEQ 3$ ;BR IF EXPECTED IRDY = 0
566 006050 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY = 1
567 006056 001022 BNE 9$ ;BR IF IRDY = 1
568 006060 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
569 ;REPORT IRDY NOT SET
570 006064 104455 ERRDF 17,EM17,ERR4
571 006064 000021 TRAP C$ERDF
572 006070 013475 .WORD 17
573 006072 016434 .WORD EM17
574 006074 000451 .WORD ERR4
575 006076 132737 000020 002350 3$: BR 16$ ;TAKE ERROR EXIT
576 006104 001407 BITB #IRDY,REDBYT ;SEE IF IRDY = 0
577 006106 004737 004226 BEQ 9$ ;BR IF IRDY = 0
578 ;REPORT IRDY NOT CLEARED
579 JSR PC,GETALL ;GET REGS FOR PRINTOUT
580 ERRDF 18,EM18,ERR4
581 006112 104455 TRAP C$ERDF
582 006112 000022 .WORD 18
583 006114 013512 .WORD EM18
584 006116 016434 .WORD ERR4
585 006120 000436
586 006122 012737 000017 002364 9$: BR 16$ ;TAKE ERROR RETURN
587 006124 004737 003372 MOV #17,REGNUM ;SET REG NO. = 17
588 006132 132776 000001 000004 JSR PC,READLU ;READ REG 17
589 006136 001413 BITB #BIT0,24(SP) ;GET EXPECTED STATE OF ICIR
590 006144 132737 000010 002350 BEQ 12$ ;BR IF EXPECTED ICIR = 0
591 006146 001031 BITB #ICIR,REDBYT ;SEE IF ICIR = 1
592 006154 004737 004226 BNE 20$ ;BR IF ICIR = 1
593 ;REPORT ICIR NOT SET
594 JSR PC,GETALL ;GET REGS FOR PRINTOUT
595 ERRDF 19,EM19,ERR4
596 006162 104455 TRAP C$ERDF
597 006162 000023 .WORD 19
598 006164 013533 .WORD EM19
599 006166 016434 .WORD ERR4
600 006170 000412
601 006172 132737 000010 002350 12$: BR 16$ ;TAKE ERROR RETURN
602 006174 001416 BITB #ICIR,REDBYT ;SEE IF ICIR = 0
603 006202 004737 004226 BEQ 20$ ;BR IF ICIR = 0
604 ;REPORT ICIR NOT CLEARED
605 JSR PC,GETALL ;GET REGS FOR PRINTOUT
606 ERRDF 19,EM19,ERR4
607 006204 004737 004226
```

GLOBAL SUBROUTINES

```

592 006210          ERRDF 20,EM20,ERR4
    006210 104455
    006212 000024
    006214 013550
    006216 016434
593 006220 016637 000002 002364 16$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.
594 006226 013706 002332          MOV PSTACK,SP ;RESTORE STACK POINTER TO BASE LEVEL
595 006232 013746 002346          MOV RETADR,-(SP) ;FIX ERROR RETURN PC
596 006236 000407          BR 23$
597 006240 062766 000002 000004 20$: ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
598 006246 012637 002336          MOV (SP)+,SUBRPC
599 006252 012637 002364          MOV (SP)+,REGNUM ;RESTORE LU REG NO.
600 006256 000207          23$: RTS PC ;RETURN
601
602
603
604
605
606
607
608
609
610
611
612
        ;*****
        ;* 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
        ;* WORD FOLLOWING THE CALL.
        ;* IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN
        ;* RETADR.
        ;*****
613 006260 013746 002364 IACTIV: MOV REGNUM,-(SP) ;SAVE LU REG NO.
614 006264 013746 002336          MOV SUBRPC,-(SP)
615 006270 005737 002336          TST SUBRPC
616 006274 001006          BNE 1$ ;SEE IF THIS IS A NESTED CALL
617 006276 016637 000004 002336          MOV 4(SP),SUBRPC ;BR IF YES
618 006304 162737 000004 002336          SL'9 #4,SUBRPC ;GET PC OF SUBR CALL
619 006312 012737 000012 002364 1$: MOV #12,REGNUM ;SET REG NO. = 12
620 006320 004737 003372          JSR PC,READL ;READ REG 12
621 006324 032776 000001 000004          BIT #BIT0,24(SP) ;GET EXPECTED STATE OF IACT
622 006332 001413          BEQ 3$ ;BR IF EXPECTED IACT = 0
623 006334 132737 000100 002350          BITB #IACT,REDBYT ;SEE IF IACT = 1
624 006342 001031          BNE 9$ ;BR IF IACT = 1
625 006344 004737 004226          JSR PC,GETALL ;GET REGS FOR PRINTOUT
626
627 006350          ;REPORT IACT NOT SET
    006350 104455          ERRDF 21,EM21,ERR4
    006352 000025
    006354 013571
    006356 016434
628 006360 000412
629 006362 132737 000100 002350 3$: BR 6$ ;TAKE ERROR EXIT
630 006370 001416          BEQ 9$ ;SEE IF IACT = 0
631 006372 004737 004226          JSR PC,GETALL ;BR IF IACT = 0
632          ;REPORT IACT NOT CLEARED
633 006376          ERRDF 22,EM22,ERR4
    006376 104455
    006400 000026
    006402 013606
    006404 016434
634 006406 016637 000002 002364 6$: MOV 2(SP),REGNUM ;RESTORE LU REG NO.
635 006414 013706 002332          MOV PSTACK,SP ;RESTORE PROGRAM STACK TO BASE LEVEL
636 006420 013746 002346          MOV RETADR,-(SP) ;FIX UP ERROR RETURN PC

```

TRAP CSERDF
.WORD 20
.WORD EM20
.WORD ERR4

TRAP CSERDF
.WORD 21
.WORD EM21
.WORD ERR4

TRAP CSERDF
.WORD 22
.WORD EM22
.WORD ERR4

```
637 006424 000407
638 006426 062766 000002 000004 9$: BR 12$
639 006434 012637 002336 ADD #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
640 006440 012637 002364 MOV (SP)+,SUBRPC
641 006444 000207 12$: MOV (SP)+,REGNUM ;RESTORE LU REG NO.
RTS PC ;RETURN
```

642
643
644
645
646
647

```
*****
; * RSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM AND REOM IN
; * AX0-16, AND REPORTS AN ERROR IF EITHER IS NOT SET TO THE STATE PASSED IN BITS
; * 0,1, RESPECTIVELY, OF THE WORD FOLLOWING THE CALL.
; * IF AN ERROR OCCURS, A RETURN IS MADE TO THE TEST AT THE ADDRESS IN RETADR.
*****
```

```
652
653 006446 013746 002366 RSEOM: MOV AXNUM,-(SP) ;SAVE AX BYTE NO.
654 006452 013746 002336 MOV SUBRPC,-(SP)
655 006456 005737 002336 TST SUBRPC ;SEE IF THIS IS A NESTED CALL
656 006462 001006 BNE 1$ ;BR IF YES
657 006464 016637 000004 002336 MOV 4(SP),SUBRPC
658 006472 162737 000004 002336 SUB #4,SUBRPC ;GET PC OF SUBR CALL
659 006500 012737 000001 002366 1$: MOV #1,AXNUM ;SET AX BYTE NO. FOR AX0-16
660 006506 004737 003624 JSR PC,READAX ;READ AX0
661 006512 032776 000001 000004 BIT #BIT0,#4(SP) ;GET EXPECTED STATE OF RSOM
662 006520 001413 BEQ 3$ ;BR IF EXPECTED RSOM = 0
663 006522 132737 000001 002356 BITB #RSOM,RAX16 ;SEE IF RSOM = 1
664 006530 001022 BNE 9$ ;BR IF RSOM = 1
665 006532 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT RSOM NOT SET
ERRDF 29,EM29,ERR6
```

```
666
667 006536
006536 104455
006540 000035
006542 013650
006544 017624
668 006546 000444
669 006550 132737 000001 002356 3$: BR 16$ ;TAKE ERROR EXIT
670 006556 001407 BEQ 9$ ;SEE IF RSOM = 0
671 006560 004737 004226 JSR PC,GETALL ;BR IF RSOM = 0
;REPORT RSOM NOT CLEARED ;GET REGS FOR PRINTOUT
ERRDF 28,EM28,ERR6
```

```
672
673 006564
006564 104455
006566 000034
006570 013627
006572 017624
674 006574 000431
675 006576 132776 000002 000004 9$: BR 16$ ;TAKE ERROR RETURN
676 006604 001413 BEQ 12$ ;GET EXPECTED STATE OF REOM
677 006606 132737 000002 002356 BITB #REOM,RAX16 ;BR IF EXPECTED REOM = 0
678 006614 001031 BNE 20$ ;SEE IF REOM = 1
679 006616 004737 004226 JSR PC,GETALL ;BR IF REOM = 1
;REPORT REOM NOT SET ;GET REGS FOR PRINTOUT
ERRDF 31,EM31,ERR6
```

```
680
681 006622
006622 104455
006624 000037
006626 013706
006630 017624
```

```
TRAP CSERDF
.WORD 29
.WORD EM29
.WORD ERR6

TRAP CSERDF
.WORD 28
.WORD EM28
.WORD ERR6

TRAP CSERDF
.WORD 31
.WORD EM31
.WORD ERR6
```

```
682 006632 000412          BR      16$      ;TAKE ERROR RETURN
683 006634 132737 000002 002356 12$:  BITB    #REOM,RAX16 ;SEE IF REOM = 0
684 006642 001416          BEQ     20$      ;BR IF REOM = 0
685 006644 004737 004226          JSR     PC,GETALL ;GET REGS FOR PRINTOUT
686                                     ;REPORT REOM NOT CLEARED
687 006650          ERRDF    30,EM30,ERR6
      006650 104455
      006652 000036
      006654 013665
      006656 017624
688 006660 016637 000002 002366 16$:  MOV     2(SP),AXNUM ;RESTORE AX BYTE NO.
689 006666 013706 002332          MOV     PSTACK,SP ;RESTORE STACK POINTER TO BASE LEVEL
690 006672 013746 002346          MOV     RETADR,-(SP) ;FIX ERROR RETURN PC
691 006676 000407          BR      23$
692 006700 062766 000002 000004 20$:  ADD     #2,4(SP) ;FIX UP ERROR-FREE RETURN PC
693 006706 012637 002336          MOV     (SP)+,SUBRPC
694 006712 012637 002366          MOV     (SP)+,AXNUM ;RESTORE AX BYTE NO.
695 006716 000207          23$:  RTS      PC ;RETURN
696
697
698
699
700
701
702
703
704
705 006720 013746 002364          RDRXSI: MOV    REGNUM,-(SP) ;SAVE LU REG NO.
706 006724 012737 000012 002364    MOV     #12,REGNUM ;SET REG NO. = 12
707 006732 004737 003372          JSR     PC,READLU ;READ LU REG 12
708 006736 113737 002350 002415    MOV     REDBYT,RXWORD+1 ;GET HI BITS OF SILO ENTRY
709 006744 042737 170000 002414    BIC     #170000,RXWORD ;CLEAR UNUSED BITS
710 006752 012737 000010 002364    MOV     #10,REGNUM ;SET REG NO. = 10
711 006760 004737 003372          JSR     PC,READLU ;READ REG 10
712 006764 113737 002350 002414    MOV     REDBYT,RXWORD ;GET LOW BITS OF SILO ENTRY
713 006772 012637 002364          MOV     (SP)+,REGNUM ;RESTORE LU REG NO.
714 006776 000207          RTS      PC ;RETURN
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729 007000 010146          RCV1ST: MOV     R1,-(SP) ;SAVE R1
730 007002 010346          MOV     R3,-(SP) ;SAVE R3
731 007004 013746 002364          MOV     REGNUM,-(SP) ;SAVE LU REG NO.
732 007010 016637 000006 002336    MOV     6(SP),SUBRPC
733 007016 162737 000004 002336    SUB     #4,SUBRPC ;GET PC OF SUBROUTINE CALL
734 007024 012737 000012 002364    MOV     #12,REGNUM ;SET LU REG NO. = 12

;*****
;* RDRXSI - THIS SUBROUTINE READS THE RCV SILO (REGS 10,12) AND RETURNS THE
;* SILO ENTRY IN BITS 0-11 OF RXWORD.
;*****

;*****
;* RCV1ST - 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.
;*****
```

```

735 007032 005001          CLR      R1          ;INIT CYCLE COUNT TO 0
736 007034 017603 000006    MOV      @6(SP),R3    ;GET CYCLE COUNT LIMIT
737 007040 062703 000003    ADD      #3,R3
738 007044 005776 000006    TST      @6(SP)      ;SEE IF DESIRED CYCLES = 0
739 007050 001414          BEQ      8$          ;BR IF YES
740 007052 004737 006260    JSR      PC,IACTIV    ;CHK FOR IACT = 0
741 007056 000000          0
742 007060 004737 005774    JSR      PC,ISIRDY    ;CHK FOR ICIR = 1, IRDY = 0
743 007064 000001          1
744 007066 004737 006446    JSR      PC,RSEOM    ;CHK RSOM = 0, REOM = 0 IN AX0-16
745 007072 000000          0
746 007074 004737 004754    6$: JSR      PC,STPLU    ;CLOCK LU FOR 1 CYCLE
747 007100 000001          1
748 007102 004737 004646    8$: JSR      PC,WAIT50   ;ALLOW SILO DATA TO RIPPLE
749 007106 005201          INC      R1          ;INCREMENT CYCLE COUNT
750 007110 004737 003372    JSR      PC,READLU   ;READ REG 12
751 007114 132737 000020 002350 BITB     #IRDY,REDBYT ;SEE IF IRDY = 1 YET
752 007122 001005          BNE      9$          ;BR IF IRDY = 1
753 007124 020103          CMP      R1,R3        ;SEE IF LIMIT EXCEEDED
754 007126 002762          BLT      6$          ;BR IF NOT YET
755 007130 004737 005774    JSR      PC,ISIRDY    ;CHK FOR ICIR = 1, IRDY = 1
756 007134 000003          3
757 007136 020176 000006    9$: CMP      R1,@6(SP) ;SEE IF LESS THAN REQUIRED CYCLES
758 007142 002003          BGE      12$         ;BR IF NOT
759 007144 004737 005774    JSR      PC,ISIRDY    ;CHK FOR ICIR = 1, IRDY = 0
760 007150 000001          1
761 007152 004737 006260    12$: JSR      PC,IACTIV ;CHK FOR IACT = 1
762 007156 000001          1
763 007158 004737 005774    JSR      PC,ISIRDY    ;CHK FOR ICIR = 1, IRDY = 1
764 007164 000003          3
765 007166 062766 000002 000006 ADD      #2,6(SP) ;FIX UP RETURN PC
766 007174 012637 002364    MOV      (SP)+,REGNUM ;RESTORE LU REG NO.
767 007200 012603          MOV      (SP)+,R3     ;RESTORE R3
768 007202 012601          MOV      (SP)+,R1     ;RESTORE R1
769 007204 005037 002336    CLR      SUBRPC    ;CLEAR SUBR CALL PC
770 007210 000207          RTS      PC          ;RETURN

```

771
772
773
774
775
776
777
778
779
780
781

```

*****
* 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.
*****

```

```

782 007212 013746 002364    STPERR: MOV      REGNUM,-(SP) ;SAVE LU REG NO.
783 007216 012737 000017 002364 MOV      #17,REGNUM ;SET LJ REG NO. = 17
784 007224 017637 000002 002352 MOV      @2(SP),WRIBYT
785 007232 152737 000002 002352 BISB     #IERR,WRIBYT
786 007240 004737 003450    JSR      PC,WRITLU ;SET IERR BIT IN REG 17
787 007244 062766 000002 000002 ADD      #2,2(SP) ;INCREMENT SUBR ARGUMENT POINTER
788 007252 017637 000002 007264 MOV      @2(SP),3$ ;GET DESIRED NO. OF CYCLES
789 007260 004737 004754    JSR      PC,STPLU ;CLOCK LU FOR DESIRED NO. OF CYCLES
790 007264 000000          3$: .WORD 0 ;NO. OF CYCLES GOES HERE
791 007266 142737 000002 002352 BICB     #IERR,WRIBYT

```

```
792 007274 004737 003450      JSR    PC,WRITLU      ;CLEAR IERR BIT IN REG 17
793 007300 062766 000002 000002  ADD    #2,2(SP)      ;FIX UP RETURN PC
794 007306 012637 002364      MOV    (SP)+,REGNUM    ;RESTORE LU REG NO.
795 007312 000207          RTS    PC      ;RETURN
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810 007314 010146          CKDATA: MOV    R1,-(SP)      ;SAVE R1
811 007316 013746 002364      MOV    REGNUM,-(SP)      ;SAVE LU REG NO.
812 007322 016637 000004 002336  MOV    4(SP),SUBRPC
813 007330 162737 000004 002336  SUB     #4,SUBRPC      ;GET PC OF SUBR CALL
814 007336 017601 000004      MOV    @4(SP),R1      ;GET EXPECTED SILO ENTRY
815 007342 042701 170000      BIC     #170000,R1      ;CLEAR UNUSED BITS FOR COMPARE
816 007346 004737 006720      JSR    PC,RDRXSI      ;READ RCV SILO
817 007352 023727 002424 000347  CMP     SAVLEN,#TXLEN2!TXLEN1!TXLEN0!RXLEN2!RXLEN1!RXLEN0
818 007360 001005          BNE     4$      ;BR IF CHAR LENGTH NOT = 7
819 007362 042701 000200      BIC     #BIT7,R1      ;MASK OFF 8TH BIT
820 007366 042737 000200 002414  BIC     #BIT7,RXWORD
821 007374 120137 002414 4$:  CMPB    R1,RXWORD      ;COMPARE EXPECTED BITS 0-7 TO ACTUAL
822 007400 001445          BEQ     6$      ;BR IF MATCH
823 007402 005037 002370      CLR     GOODAT
824 007406 110137 002370      MOVB    R1,GOODAT      ;GET EXPECTED DATA
825 007412 005037 002372      CLR     BADDAT
826 007416 113737 002414 002372  MOVB    RXWORD,BADDAT      ;GET ACTUAL DATA
827 007424 012737 000011 002364  MOV     #11,REGNUM      ;SET REG NO. = 11
828 007432 004737 003372      JSR    PC,READLU      ;READ REG 11
829 007436 132737 000001 002350  BITB    #UNRR,REDBYT      ;SEE IF TX UNDERRUN ERROR
830 007444 001410          BEQ     5$      ;BR IF NOT, TO REPORT DATA ERROR
831 007446 004737 004226      JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
832
833
834 007452          ;REPORT TX UNDERRUN ERROR
835 007452 104455          ERRDF    54,EM54,ERR4
836 007454 000066
837 007456 014617
838 007460 016434
839 007462 000137 010144      JMP     36$      ;TAKE ERROR EXIT
840 007466 012737 000010 002364 5$:  MOV     #10,REGNUM      ;SET REG NO. = 10
841 007474 004737 004226      JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
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
```

```
841 007516 012737 000012 002364      MOV    #12,REGNUM      ;SET LU REG NO. FOR ERROR REPORTS
842 007524 120137 002415      CMPB   R1,RXWORD+1    ;COMPARE EXPECTED SILO BITS 8-11 TO ACTUAL
843 007530 001002              BNE     7$                ;BR IF MISMATCH
844 007532 000137 010120      JMP     22$                ;CONTINUE
845 007536 005037 002370      7$:    CLR     GOODAT
846 007542 110137 002370      MOVB   R1,GOODAT      ;SET EXPECTED DATA
847 007546 005037 002372      CLR     BADDAT
848 007552 113737 002415 002372 MOVB   RXWORD+1,BADDAT ;SET ACTUAL DATA
849 007560 032776 100000 000004 BIT    #BCCCHK,24(SP) ;SEE IF BCC SHOULD BE IGNORED
850 007566 001433              BEQ     10$                ;BR IF YES
851 007570 132701 000001      BITB   #BCC,R1          ;SEE IF EXPECTED BIT = 1
852 007574 001014              BNE     8$                ;BR IF YES
853 007576 132737 000001 002415 BITB   #BCC,RXWORD+1 ;SEE IF ACTUAL BIT = 0
854 007604 001424              BEQ     10$                ;BR IF YES
855 007606 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
856                      ;REPORT BCC NOT CLEARED
857 007612 104455              ERRDF   35,EM35,ERR8
                                TRAP    C$ERDF
                                .WORD   35
                                .WORD   EM35
                                .WORD   ERR8
858 007622 000137 010144      JMP     36$                ;TAKE ERROR EXIT
859 007626 132737 000001 002415 8$:    BITB   #BCC,RXWORD+1 ;SEE IF ACTUAL BIT = 1
860 007634 001010              BNE     10$                ;BR IF YES
861 007636 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
862                      ;REPORT BCC NOT SET
863 007642 104455              ERRDF   36,EM36,ERR8
                                TRAP    C$ERDF
                                .WORD   36
                                .WORD   EM36
                                .WORD   ERR8
864 007652 000137 010144      JMP     36$                ;TAKE ERROR EXIT
865 007656 132701 000002      10$:   BITB   #EBLK,R1          ;SEE IF EXPECTED BIT = 1
866 007662 001014              BNE     12$                ;BR IF YES
867 007664 132737 000002 002415 BITB   #EBLK,RXWORD+1 ;SEE IF ACTUAL BIT = 0
868 007672 001424              BEQ     14$                ;BR IF YES
869 007674 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
870                      ;REPORT EBLK NOT CLEARED
871                      ERRDF   37,EM37,ERR8
                                TRAP    C$ERDF
                                .WORD   37
                                .WORD   EM37
                                .WORD   ERR8
872 007700 104455              007700 000045
                                007702 014005
                                007706 020734
873 007710 000137 010144      JMP     36$                ;TAKE ERROR EXIT
874 007714 132737 000002 002415 12$:  BITB   #EBLK,RXWORD+1 ;SEE IF ACTUAL BIT = 1
875 007722 001010              BNE     14$                ;BR IF YES
876 007724 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
877                      ;REPORT EBLK NOT SET
878 007730 104455              ERRDF   38,EM38,ERR8
                                TRAP    C$ERDF
                                .WORD   38
                                .WORD   EM38
                                .WORD   ERR8
879 007740 000137 010144      JMP     36$                ;TAKE ERROR EXIT
880 007744 132701 000004      14$:   BITB   #RAB,R1          ;SEE IF EXPECTED BIT = 1
881 007744 132701 000004
```

882	007750	001014			BNE	16\$		;BR IF YES	
883	007752	132737	000004	002415	BITB	#RAB,RXWORD+1		;SEE IF ACTUAL BIT = 0	
884	007760	001424			BEQ	18\$		;BR IF YES	
885	007762	004737	004226		JSR	PC,GETALL		;GET REGS FOR PRINTOUT	
886					;REPORT RAB NOT	CLEARED			
887	007766				ERRDF	39,EM39,ERR8			
	007766	104455						TRAP	C\$ERDF
	007770	000047						.WORD	39
	007772	014043						.WORD	EM39
	007774	020734						.WORD	ERR8
888	007776	000137	010144		JMP	36\$		;TAKE ERROR EXIT	
889	010002	132737	000004	002415	16\$: BITB	#RAB,RXWORD+1		;SEE IF ACTUAL BIT = 1	
890	010010	001010			BNE	18\$		;BR IF YES	
891	010012	004737	004226		JSR	PC,GETALL		;GET REGS FOR PRINTOUT	
892					;REPORT RAB NOT	SET			
893	010016				ERRDF	40,EM40,ERR8			
	010016	104455						TRAP	C\$ERDF
	010020	000050						.WORD	40
	010022	014063						.WORD	EM40
	010024	020734						.WORD	ERR8
894	010026	000137	010144		JMP	36\$		;TAKE ERROR EXIT	
895	010032				18\$: BITB	#OVRR,R1		;SEE IF EXPECTED BIT = 1	
896	010032	132701	000010		BNE	20\$		;BR IF YES	
897	010036	001014			BITB	#OVRR,RXWORD+1		;SEE IF ACTUAL BIT = 0	
898	010040	132737	000010	002415	BEQ	22\$		;BR IF YES	
899	010046	001424			JSR	PC,GETALL		;GET REGS FOR PRINTOUT	
900	010050	004737	004226		;REPORT OVRR NOT	CLEARED			
901					ERRDF	41,EM41,ERR8			
902	010054							TRAP	C\$ERDF
	010054	104455						.WORD	41
	010056	000051						.WORD	EM41
	010060	014077						.WORD	ERR8
	010062	020734							
903	010064	000137	010144		JMP	36\$		;TAKE ERROR EXIT	
904	010070	132737	000010	002415	20\$: BITB	#OVRR,RXWORD+1		;SEE IF ACTUAL BIT = 1	
905	010076	001010			BNE	22\$		;BR IF YES	
906	010100	004737	004226		JSR	PC,GETALL		;GET REGS FOR PRINTOUT	
907					;REPORT OVRR NOT	SET			
908	010104				ERRDF	42,EM42,ERR8			
	010104	104455						TRAP	C\$ERDF
	010106	000052						.WORD	42
	010110	014120						.WORD	EM42
	010112	020734						.WORD	ERR8
909	010114	000137	010144		JMP	36\$		;TAKE ERROR EXIT	
910	010120				22\$: ADD	#2,4(SP)		;INCR SUBROUTINE ARGUMENT POINTER	
911	010120	062766	000002	000004	MOV	24(SP),24\$		;GET DESIRED CYCLE COUNT	
912	010126	017637	000004	010140	JSR	PC,STPLU		;CLOCK LU FOR DESIRED CYCLES	
913	010134	004737	004754						
914	010140	000000			24\$: .WORD	0			
915	010142	000407			BR	38\$		;TAKE ERROR-FREE EXIT	
916	010144	011637	002364		36\$: MOV	(SP),REGNUM		;RESTORE LU REG NO.	
917	010150	013706	002332		MOV	PSTACK,SP		;RESTORE PROGRAM STACK TO BASE LEVEL	
918	010154	013746	002346		MOV	RETADR,-(SP)		;FIX UP ERROR RETURN PC	
919	010160	000406			BR	40\$			
920	010162	062766	000002	000004	38\$: ADD	#2,4(SP)		;FIX UP ERROR-FREE RETURN PC	
921	010170	012637	002364		MOV	(SP)+,REGNUM		;RESTORE LU REG NO.	
922	010174	012601			MOV	(SP)+,R1		;RESTORE R1	

```

923 010176 005037 002336      40$: CLR      SUBRPC      ;CLEAR SUBROUTINE PC
924 010202 000207              RTS      PC      ;RETURN
925
926
927
928
929
930
931      ;*****
932      ;* LODATA - THIS SUBROUTINE LOADS THE TRANSMITTER SILO WITH 5 SOM'S, THE DATA
933      ;* IN PATTERN A REPEATED 2 TIMES (40 CHARS), AND 2 EOM'S. IN ADDITION, THE
934      ;* DATA CHARS ARE ALSO LOADED INTO THE RECEIVED MSG BUFFER (RCVBUF:), AS
935      ;* EXPECTED DATA FOR LATER COMPARISON.
936      ;*****
936 010204 010146      LODATA: MOV      R1,-(SP)      ;SAVE R1
937 010206 010346      MOV      R3,-(SP)      ;SAVE R3
938 010210 010446      MOV      R4,-(SP)      ;SAVE R4
939 010212 004737 010746      JSR      PC,LODSIL      ;LOAD 5 SOM'S INTO TX SILO
940 010216 000400      TXSOM
941 010220 000005      5
942 010222 012701 000002      MOV      #2,R1      ;INIT COUNTER
943 010226 012704 002762      MOV      #RCVBUF,R4      ;GET POINTER TO RCV BUF
944 010232 012703 002557      3$: MOV      #PATA,R3      ;GET POINTER TO DATA PATTERN
945 010236 005037 002412      6$: CLR      TXWORD
946 010242 112337 002412      MOV      (R3)+,TXWORD      ;GET A DATA CHAR
947 010246 013724 002412      MOV      TXWORD,(R4)+      ;LOAD A DATA CHAR INTO RCV BUF
948 010252 004737 004674      JSR      PC,LDTXSI      ;LOAD DATA CHAR INTO TX SILO
949 010256 020327 002603      CMP      R3,#PATB      ;SEE IF AT END OF PATTERN A YET
950 010262 103765      BLO      6$      ;BR IF NOT YET
951 010264 005301      DEC      R1      ;DECREMENT COUNTER
952 010266 001361      BNE      3$      ;BR IF NOT DONE YET
953 010270 052764 100400 177776      BIS      #CRCCHK!RXBCC,-2(R4) ;SET UP TO CHK BCC = 1 ON LAST DATA CHAR
954 010276 012737 001000 002412      MOV      #TXEOM,TXWORD
955 010304 004737 004674      JSR      PC,LDTXSI      ;LOAD AN EOM INTO TX SILO
956 010310 004737 004674      JSR      PC,LDTXSI      ;LOAD ANOTHER EOM
957 010314 012604      MOV      (SP)+,R4      ;RESTORE R4
958 010316 012603      MOV      (SP)+,R3      ;RESTORE R3
959 010320 012601      MOV      (SP)+,R1      ;RESTORE R1
960 010322 000207      RTS      PC      ;RETURN
961
962
963
964
965
966      ;*****
967      ;* SETUP - THIS SUBROUTINE LOADS THE FIRST WORD AFTER THE CALL INTO AX2-15
968      ;* (SYNCH CHAR), LOADS THE SECOND WORD AFTER THE CALL INTO REG 17
969      ;* LOADS THE THIRD WORD INTO AX3-15, AND LOADS THE FOURTH WORD INTO AX3-16.
970      ;*****
971 010324 013746 002366      SETUP: MOV      AXNUM,-(SP)      ;SAVE AX BYTE NO.
972 010330 013746 002364      MOV      REGNUM,-(SP)      ;SAVE LU REG NO.
973 010334 012737 000004 002366      MOV      #4,AXNUM      ;SET AX BYTE NO. FOR AX2
974 010342 017637 000004 002360      MOV      @4(SP),WAX15
975 010350 005037 002362      CLR      WAX16
976 010354 004737 004012      JSR      PC,WRITAX      ;SET SYNCH CHAR IN AX2-15, CLEAR AX2-16
977 010360 012737 000017 002364      MOV      #17,REGNUM      ;SET LU REG NO. = 17
978 010366 062766 000002 000004      ADD      #2,4(SP)      ;INCREMENT ARGUMENT POINTER
979 010374 017637 000004 002352      MOV      @4(SP),WRIBY

```

```
980 010402 004737 003450 JSR PC,WRITLU ;LOAD REG 17
981 010406 012737 000006 002366 MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
982 010414 062766 000002 000004 ADD #2,4(SP) ;INCREMENT ARGUMENT POINTER
983 010422 017637 000004 002360 MOV @4(SP),WAX15
984 010430 062766 000002 000004 ADD #2,4(SP) ;INCREMENT ARGUMENT POINTER
985 010436 017637 000004 002362 MOV @4(SP),WAX16
986 010444 013737 002362 002424 MOV WAX16,SAVLEN ;STORE TX AND RCV CHAR LENGTHS
987 010452 142777 000010 171760 BICB #LLOOP,@BSEL1 ;CLEAR LLOOP
988 010460 004737 004012 JSR PC,WRITAX ;LOAD AX3-15, AX3-16
989 010464 152777 000010 171746 BISB #LLOOP,@BSEL1 ;SET LLOOP
990 010472 062766 000002 000004 ADD #2,4(SP) ;FIX RETURN PC
991 010500 012637 002364 MOV (SP)+,REGNUM ;RESTORE LU REG NO.
992 010504 012637 002366 MOV (SP)+,AXNUM ;RESTORE AX BYTE NO.
993 010510 005037 002336 CLR SUBRPC ;CLEAR SUBROUTINE PC STORAGE
994 010514 000207 RTS PC ;RETURN
995
996
997
998
999
```

```
1000 *****
1001 * LODMSG - THIS SUBROUTINE LOADS THE NO. OF WORDS PASSED IN THE SECOND WORD
1002 * FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
1003 * WORD FOLLOWING THE CALL, INTO THE TRANSMITTER SILO.
1004 *****
1005 010516 010146 LODMSG: MOV R1,-(SP) ;SAVE R1
1006 010520 010246 MOV R2,-(SP) ;SAVE R2
1007 010522 017601 000004 000004 MOV @4(SP),R1 ;GET MSG POINTER INTO R1
1008 010526 062766 000002 000004 ADD #2,4(SP) ;INCR ARG POINTER
1009 010534 017602 000004 000004 MOV @4(SP),R2 ;GET WORD COUNT INTO R2
1010 010540 062766 000002 000004 ADD #2,4(SP) ;FIX UP RETURN PC
1011 010546 012137 002412 6$: MOV (R1)+,TXWORD ;GET NEXT MSG WORD
1012 010552 004737 004674 JSR PC,LDTXSI ;LOAD A WORD INTO TX SILO
1013 010556 005302 DEC R2 ;DECR COUNT
1014 010560 001372 BNE 6$ ;BR IF NOT DONE YET
1015 010562 004737 004646 JSR PC,WAIT50 ;WAIT FOR SILO TO RIPPLE
1016 010566 012602 MOV (SP)+,R2 ;RESTORE R2
1017 010570 012601 MOV (SP)+,R1 ;RESTORE R1
1018 010572 000207 RTS PC ;RETURN
1019
1020
1021
1022
1023
```

```
1024 *****
1025 * LDBYTS - THIS SUBROUTINE LOADS THE NO. OF BYTES PASSED IN THE SECOND WORD
1026 * FOLLOWING THE CALL FROM THE MSG BUFFER WHOSE ADDRESS IS IN THE FIRST
1027 * WORD FOLLOWING THE CALL, INTO THE LOW BYTE OF THE TX SILO. FOR EACH
1028 * BYTE LOADED, A 0 IS LOADED INTO THE HI 4 BITS OF THE TX SILO.
1029 *****
1030 010574 010146 LDBYTS: MOV R1,-(SP) ;SAVE R1
1031 010576 010246 MOV R2,-(SP) ;SAVE R2
1032 010600 017601 000004 000004 MOV @4(SP),R1 ;GET DATA POINTER INTO R1
1033 010604 062766 000002 000004 ADD #2,4(SP) ;INCR ARGUMENT POINTER
1034 010612 017602 000004 000004 MOV @4(SP),R2 ;GET BYTE COUNT INTO R2
1035 010616 062766 000002 000004 ADD #2,4(SP) ;FIX UP RETURN PC
1036 010624 112137 002412 6$: MOVB (R1)+,TXWORD ;GET NEXT DATA BYTE
```

1037 010630 105037 002413
1038 010634 004737 004674
1039 010640 005302
1040 010642 001370
1041 010644 004737 004646
1042 010650 012602
1043 010652 012601
1044 010654 000207

CLRB TXWORD+1 ;CLEAR HI BYTE
JSR PC,LDTXSI ;LOAD A SILO ENTRY
DEC R2 ;DECR BYTE 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

1045
1046
1047
1048
1049

1050
1051
1052
1053

; * 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.

1054
1055 010656 010146
1056 010660 010246
1057 010662 004737 010746
1058 010666 000400
1059 010670 000003
1060 010672 004737 010516
1061 010676 002670
1062 010700 000010
1063 010702 012701 002674
1064 010706 012702 002762
1065 010712 012122
1066 010714 020127 002706
1067 010720 103774
1068 010722 052762 100400 177776
1069 010730 012722 000160
1070 010734 012722 000034
1071 010740 012602
1072 010742 012601
1073 010744 000207

LDMSG1: MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
JSR PC,LODSIL ;LOAD 3 SOM'S INTO TX SILO
TXSOM
3
JSR PC,LODMSG ;LOAD MSG1 INTO TX SILO (WITH 2 SOM'S, 1 EOM)
MSG1
8.
MOV #MSG1+4,R1 ;GET POINTER TO MSG1
MOV #RCVBUF,R2 ;GET POINTER TO MSG BUF
3\$: MOV (R1)+,(R2)+ ;LOAD A CHAR INTO MSG BUF
CMP R1,#MSG1+14. ;SEE IF DID LAST DATA CHAR YET
BLO 3\$;BR IF NOT
BIS #CRCCHK!RXBCC,-2(R2) ;SET EXPECTED BCC
MOV #160,(R2)+ ;LOAD HI CRC BYTE
MOV #034,(R2)+ ;LOAD LO CRC BYTE
MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS PC ;RETURN

1074
1075
1076
1077
1078
1079

1080
1081
1082
1083

; * LODSIL - THIS SUBROUTINE REPEATEDLY LOADS THE DATA PASSED IN THE FIRST WORD
; * FOLLOWING THE CALL INTO THE TX SILO, FOR THE NO. OF TIMES PASSED IN THE
; * SECOND WORD FOLLOWING THE CALL.

1084 010746 010146
1085 010750 017637 000002 002412
1086 010756 062766 000002 000002
1087 010764 017601 000002
1088 010770 004737 004674
1089 010774 005301
1090 010776 001374
1091 011000 004737 004646
1092 011004 062766 000002 000002
1093 011012 012601

LODSIL: MOV R1,-(SP) ;SAVE R1
MOV @2(SP),TXWORD ;GET DATA
ADD #2,2(SP) ;INCR ARGUMENT POINTER
MOV @2(SP),R1 ;GET COUNT
3\$: JSR PC,LDTXSI ;LOAD WORD INTO TX SILO
DEC R1 ;DECR COUNT
BNE 3\$;BR IF NOT ALL LOADED YET
JSR PC,WAIT50 ;ALLOW SILO DATA TO RIPPLE
ADD #2,2(SP) ;FIX UP RETURN PC
MOV (SP)+,R1 ;RESTORE R1

1094 011014 000207

RTS

PC

;RETURN

1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118 011016 013746 002364
1119 011022 013746 002366
1120 011026 010246
1121 011030 016637 000006 002336
1122 011036 162737 000004 002336
1123 011044 012737 000333 002422
1124 011052 032776 100000 000006
1125 011060 001405
1126 011062 005737 002462
1127 011066 001074
1128 011070 000137 012122
1129
1130 011074 023727 002462 000004
1131 011102 001466
1132 011104 142777 000010 171326
1133 011112 012737 000006 002366
1134 011120 004737 003624
1135
1136
1137
1138 011124 027627 000006 000010
1139 011132 001422
1140 011134 027627 000006 000020
1141 011142 001512
1142 011144 027627 000006 000200
1143 011152 001002
1144 011154 000137 011454
1145 011160 027627 000006 000100
1146 011166 001002
1147 011170 000137 011510
1148 011174 000137 011602
1149
1150

```
*****
* CKLPBK - THIS SUBROUTINE DETERMINES IF THE TEST CALLING IT CAN BE RUN. THE
* TEST PASSES THE DESIRED MODEM INTERFACE TYPE IN THE WORD FOLLOWING THE
* CALL, AND IF THE PROPER EXTERNAL LOOPBACK HAS BEEN PROVIDED BY THE
* OPERATOR FOR THAT INTERFACE, AND IF THE BAUD RATE IS CORRECT, A NORMAL
* RETURN IS MADE TO RUN THE TEST. IF NOT, A RETURN IS MADE TO THE TEST,
* AT THE ADDRESS IN RETADR (RETADR CONTAINS THE TEST EXIT ADDRESS, SO
* THE TEST GETS SKIPPED).
* IF BIT 15 IS SET IN THE WORD FOLLOWING THE CALL, THE TEST WILL NOT
* BE RUN UNLESS THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. IF THE
* TEST IS TO BE RUN, THE SUBROUTINE RETURNS THE MODEM SELECT BITS FOR
* AX3-15 IN MODINT. IF NECESSARY, THE SUBROUTINE WILL SET MAINT1, MAINT2,
* OR DTR.
* IF THE PROGRAM PASSES '0' IN THE WORD FOLLOWING THE CALL, THE SUBROUTINE
* WILL ATTEMPT TO RUN WHICHEVER MODEM INTERFACE IS SELECTED BY CABLE,
* SWITCH, OR TEST CONNECTOR. IF SUCCESSFUL, THE SELECTED INTERFACE WILL
* BE PASSED BACK TO THE TEST IN MODINT.
*****
CKLPBK: MOV     REGNUM, -(SP)      ;SAVE REG NO.
        MOV     AXNUM, -(SP)     ;SAVE AX BYTE NO.
        MOV     R2, -(SP)       ;SAVE R2
        MOV     6(SP), SUBRPC    ;GET PC OF SUBR CALL
        SUB     #4, SUBRPC
        MOV     #1422!XYZ!V35!INTGRL!OP!TEST, MODINT ;INIT MODEM SELECT BITS
        BIT     #TCCHEK, @6(SP)  ;SEE IF H3254,5 CHECK IS DESIRED
        BEQ     1$,              ;BR IF NOT
        TST     TSTCON           ;SEE IF H3254,5 INSTALLED
        BNE     7$,              ;BR IF NOT, TO SKIP TEST
        JMP     32$,             ;BR TO RUN TEST
;IF NO EXTERNAL LPBK, SKIP TEST
1$:      CMP     TSTCON, #4       ;SEE IF NO LPBK
        BEQ     7$,              ;BR IF NO LPBK, TO SKIP TEST
        BICB    #LULoop, @BSEL1 ;CLEAR LULoop
        MOV     #6, AXNUM        ;SET AX BYTE NO. FOR AX3-15
        JSR     PC, READAX       ;READ AX3-15

;*** SEE IF AN INTERFACE IS REQUESTED ***
        CMP     @6(SP), #INTGRL  ;SEE IF INTEGRAL MODEM REQUESTED
        BEQ     4$,              ;BR IF INTGRL MODEM REQUESTED
        CMP     @6(SP), #V35     ;SEE IF V.35 REQUESTED
        BEQ     10$,             ;BR IF V.35 REQUESTED
        CMP     @6(SP), #1422    ;SEE IF 422 REQUESTED
        BNE     2$,              ;BR IF 422 NOT REQUESTED
        JMP     14$,             ;422 REQUESTED
2$:      CMP     @6(SP), #XYZ     ;SEE IF XYZ REQUESTED
        BNE     3$,              ;BR IF XYZ NOT REQUESTED
        JMP     17$,             ;XYZ REQUESTED
3$:      JMP     21$,             ;NONE REQUESTED, FIND AN INTERFACE TO TEST

;SEE IF INTEGRAL MODEM CAN BE RUN
```

```
1151 011200 005737 002462      4$:   TST      TSTCON      ;SEE IF H3254 AND H3255 USED
1152 011204 001050                BNE      8$          ;BR IF NOT
1153 011206 023727 002464 000004 5$:   CMP      BDRATE,#4      ;SEE IF BAUD RATE > OR = 56K
1154 011214 002405                BLT      6$          ;BR IF BAUD RATE TOO SLOW FOR INTGRL MODM
1155 011216 042737 000010 002422 BIC      #INTGRL,MODINT ;ASSERT INTEGRAL MODEM
1156 011224 000137 012122      JMP      32$        ;GO TO RUN TEST
1157                                ;PRINT 'BAUD RATE INCORRECT'
1158 011230                6$:                                ;
1159 011230 023727 002402 000001      CMP      STARES,#1      ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
1160 011236 001026                BNE      40$          ;BR IF NOT, TO SKIP PRINTING
1161 011240                PRINTF    #FMT25
                                MOV      #FMT25,-(SP)
                                MOV      #1,-(SP)
                                MOV      SP,R0
                                TRAP     C$PNTF
                                ADD      #4,SP
                                011240 012746 013174
                                011244 012746 000001
                                011250 010600
                                011252 104417
                                011254 062706 000004
1162                                ;PRINT 'TEST XX NOT RUN'
1163 011260                7$:                                ;
1164 011260 023727 002402 000001      CMP      STARES,#1      ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
1165 011266 001012                BNE      40$          ;BR IF NOT, TO SKIP PRINTING
1166 011270                PRINTF    #FMT19,TSTNUM
                                MOV      TSTNUM,-(SP)
                                MOV      #FMT19,-(SP)
                                MOV      #2,-(SP)
                                MOV      SP,R0
                                TRAP     C$PNTF
                                ADD      #6,SP
                                011270 013746 002434
                                011274 012746 013143
                                011300 012746 000002
                                011304 010600
                                011306 104417
                                011310 062706 000006
1167 011314 013766 002346 000006 40$:  MOV      RETADR,6(SP) ;SET TEST EXIT ADDRESS FOR ERRORS
1168 011322 000137 012130                JMP      33$        ;GO TO SKIP TEST
1169 011326 032737 000010 002354 8$:   BIT      #INTGRL,RAX15 ;SEE IF INTEGRAL MODEM IS SELECTED
1170 011334 001324                BNE      5$          ;BR IF YES, TO CHECK BAUD RATE
1171                                ;PRINT 'MODEM INTERFACE NOT SELECTED'
1172 011336                9$:                                ;
1173 011336 023727 002402 000001      CMP      STARES,#1      ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
1174 011344 001363                BNE      40$          ;BR IF NOT, TO SKIP PRINTING
1175 011346                PRINTF    #FMT26
                                MOV      #FMT26,-(SP)
                                MOV      #1,-(SP)
                                MOV      SP,R0
                                TRAP     C$PNTF
                                ADD      #4,SP
                                011346 012746 013224
                                011352 012746 000001
                                011356 010600
                                011360 104417
                                011362 062706 000004
1176 011366 000734                BR      7$          ;GO TO PRINT 'TEST NOT RUN'
1177
1178                                ;SEE IF V.35 CAN BE RUN
1179 011370 032737 000200 002354 10$:  BIT      #I422,RAX15 ;SEE IF 422 IS SELECTED
1180 011376 001357                BNE      9$          ;BR IF YES, TO SKIP TEST
1181 011400 005737 002462                TST      TSTCON      ;SEE IF H3254 AND H3255 USED
1182 011404 001010                BNE      12$         ;BR IF H3254 AND H3255 NOT USED
1183 011406 012737 000004 002352 11$:  MOV      #MAINT2,WRIBYT ;SET MAINT2 FOR MANUFACTURING TEST CONN.
1184 011414 042737 000020 002422 BIC      #V35,MODINT ;ASSERT V.35
1185 011422 000137 011704                JMP      42$        ;GO SET DTR AND RUN THE TEST
1186 011426 032737 000020 002354 12$:  BIT      #V35,RAX15 ;SEE IF V.35 IS SELECTED
1187 011434 001405                BEQ      13$         ;BR IF NOT
1188 011436 042737 000020 002422 BIC      #V35,MODINT ;ASSERT V.35
1189 011444 000137 011700                JMP      27$        ;GO SET DTR AND RUN THE TEST
1190 011450 000137 011336                13$:  JMP      9$          ;WRONG INTRFCE, GO SKIP TEST
1191
```

```
1192      ;SEE IF 422 CAN BE RUN
1193 011454 005737 002462      14$: TST TSTCON      ;SEE IF H3254 AND H3255 USED
1194 011460 001005      BNE 16$      ;BR IF NOT
1195 011462 042737 000200 002422 15$: BIC #1422,MODINT ;ASSERT 422
1196 011470 000137 011700      JMP 27$      ;GO TO RUN TEST
1197 011474 032737 000200 002354 16$: BIT #1422,RAX15 ;SEE IF 422 IS SELECTED
1198 011502 001367      BNE 15$      ;IF YES, GO ASSERT 422 AND RUN TEST
1199 011504 000137 011336      JMP 9$      ;WRONG INTRFCE, GO SKIP TEST
1200
1201      ;SEE IF XYZ CAN BE RUN
1202 011510 032737 000200 002354 17$: BIT #1422,RAX15 ;SEE IF 422 IS SELECTED
1203 011516 001402      BEQ 18$      ;BR IF NOT
1204 011520 000137 011336      JMP 9$      ;WRONG INTRFCE, GO SKIP TEST
1205 011524 032737 000100 002354 18$: BIT #XYZ,RAX15 ;SEE IF XYZ IS SELECTED
1206 011532 001002      BNE 19$      ;BR IF YES
1207 011534 000137 011336      JMP 9$      ;WRONG INTRFCE, GO SKIP TEST
1208 011540 023727 002464 000004 19$: CMP BDRATE,#4 ;SEE IF BAUD RATE < OR = 56K
1209 011546 003402      BLE 20$      ;BR IF YES
1210 011550 000137 011230      JMP 6$      ;BAUD RATE TOO FAST FOR XYZ
1211 011554 042737 000100 002422 20$: BIC #XYZ,MODINT ;ASSERT XYZ
1212 011562 005737 002462      TST TSTCON      ;SEE IF H3254,5 BEING USED
1213 011566 001044      BNE 27$      ;BR IF NOT
1214 011570 012737 000004 002352      MOV #MAINT2,WRIBYT
1215 011576 000137 011704      JMP 42$      ;GO SET MAINT2 FOR MANUFACTURING TEST CONN.
1216
1217      ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
1218
1219 011602 032737 000010 002354 21$: BIT #INTGRL,RAX15 ;SEE IF INTEGRAL MODEM SELECTED
1220 011610 001402      BEQ 22$      ;BR IF NOT
1221 011612 000137 011206      JMP 5$      ;SEE IF INTEGRAL MODEM CAN BE RUN
1222 011616 032737 000020 002354 22$: BIT #V35,RAX15 ;SEE IF V.35 SELECTED
1223 011624 001402      BEQ 23$      ;BR IF NOT
1224 011626 000137 011370      JMP 10$      ;GO SEE IF V.35 CAN BE RUN
1225 011632 032737 000200 002354 23$: BIT #1422,RAX15 ;SEE IF 422 SELECTED
1226 011640 001402      BEQ 24$      ;BR IF NOT
1227 011642 000137 011462      JMP 15$      ;GO ASSERT AND RUN 422
1228 011646 005737 002462      24$: TST TSTCON      ;SEE IF H3254 AND H3255 USED
1229 011652 001002      BNE 25$      ;BR IF NOT
1230 011654 000137 011406      JMP 11$      ;GO ASSERT AND RUN V.35 BY DEFAULT
1231 011660 032737 000100 002354 25$: BIT #XYZ,RAX15 ;SEE IF XYZ SELECTED
1232 011666 001402      BEQ 26$      ;BR IF NOT
1233 011670 000137 011540      JMP 19$      ;GO SEE IF XYZ CAN BE RUN
1234 011674 000137 011336      26$: JMP 9$      ;GO SKIP TEST
1235
1236      ;*** SET MAINT1 OR MAINT2 IF NEEDED, AND SET DTR ***
1237
1238      ;SET MAINT1 IF LOCAL LOOPBACK DESIRED
1239 011700 005037 002352      27$: CLR WRIBYT      ;INIT DATA TO BE WRITTEN
1240 011704 012737 000013 002364 42$: MOV #13,REGNUM ;SET REG NO. = 13
1241 011712 023727 002462 000002      CMP TSTCON,#2 ;SEE IF MODEM LOCAL LPBK DESIRED
1242 011720 001032      BNE 29$      ;BR IF NOT, TO CHK REMOTE LPBK
1243 011722 112737 000010 002352      MOVB #MAINT1,WRIBYT
1244 011730 004737 003450      JSR PC,WRITLU ;SET MAINT1 IN REG 13
1245 011734 012737 000017 002364      MOV #17,REGNUM ;SET REG NO. = 17
1246 011742 012702 000000      MOV #0,R2 ;INIT TIMER
1247 011746 004737 003372      28$: JSR PC,READLU ;READ REG 17
1248 011752 132737 000004 002350      BITB #TESTMD,REDBYT ;SEE IF TEST MODE BIT SET IN REG 17 YET
```

```
1249 011760 001050      BNE      31$      ;BR IF YES, TO SET DTR
1250 011762 005202      INC      R2      ;INCREMENT TIMER
1251 011764 001370      BNE      28$      ;BR IF NO TIME-OUT YET
1252 011766 004737 004226 JSR      PC,GETALL ;GET REGS FOR PRINTOUT
1253                      ;REPORT TEST MODE NOT SET BY MAINT1
1254                      ERRDF 52,EM52,ERR4
1254 011772      104455
1254 011772 000064      TRAP      C$ERDF
1254 011774 000064      .WORD    52
1254 011776 014527      .WORD    EM52
1254 012000 016434      .WORD    ERR4
1254 012000 016434
1255 012002 000137 011260 JMP      7$      ;BR TO SKIP TEST
1256                      ;SET MAINT2 IF REMOTE LOOPBACK DESIRED
1257 012006 023727 002462 000003 29$: CMP      TSTCON,#3 ;SEE IF MODEM REMOTE LPBK DESIRED
1258 012014 001032      BNE      31$      ;BR IF NOT, TO SET DTR
1259 012016 112737 000004 002352 MOV      #MAINT2,WRIBYT
1260 012024 004737 003450 JSR      PC,WRITLU ;SET MAINT2 IN REG 13
1261 012030 012737 000017 002364 MOV      #17,REGNUM ;SET REG NO. = 17
1262 012036 012702 000000      MOV      #0,R2 ;INIT TIMER
1263 012042 004737 003372 30$: JSR      PC,READLU ;READ REG 17
1264 012046 132737 000004 002350 BITB     #TESTMD,REDBYT ;SEE IF TEST MODE BIT SET IN REG 17 YET
1265 012054 001012      BNE      31$      ;BR IF YES, TO SET DTR
1266 012056 005202      INC      R2      ;INCREMENT TIMER
1267 012060 001370      BNE      30$      ;BR IF NO TIME-OUT YET
1268 012062 004737 004226 JSR      PC,GETALL ;GET REGS FOR PRINTOUT
1269                      ;REPORT TEST MODE NOT SET BY MAINT2
1270                      ERRDF 53,EM53,ERR4
1270 012066      104455
1270 012066 000065      TRAP      C$ERDF
1270 012070 000065      .WORD    53
1270 012072 014563      .WORD    EM53
1270 012074 016434      .WORD    ERR4
1271 012076 000137 011260 JMP      7$      ;BR TO SKIP TEST
1272                      ;SET DTR
1273 012102 012737 000013 002364 31$: MOV      #13,REGNUM ;SET REG NO. = 13
1274 012110 152737 000100 002352 BISB     #DTR,WRIBYT ;SET DTR BIT TO BE WRITTEN
1275 012116 004737 003450 JSR      PC,WRITLU ;LOAD REG 13
1276
1277                      ;*** BRANCH HERE TO RUN TEST ***
1278 012122 062766 000002 000006 32$: ADD      #2,6(SP) ;INCREMENT RETURN ADRS
1279
1280                      ;*** BRANCH HERE TO SKIP TEST ***
1281 012130 012602 33$: MOV      (SP)+,R2 ;RESTORE R2
1282 012132 012637 002366 MOV      (SP)+,AXNUM ;RESTORE AX BYTE NO.
1283 012136 012637 002364 MOV      (SP)+,REGNUM ;RESTORE LU REG NO.
1284 012142 152777 000010 170270 BISB     #LULoop,ABSEL1 ;SET LULoop
1285 012150 005037 002336 CLR      SUBRPC ;CLEAR SUBROUTINE CALL PC
1286 012154 000207 RTS      PC ;RETURN
1287
1288
1289
1290
1291
1292                      ;*****
1293                      ;* CHKABT - THIS SUBROUTINE READS AX0-16 AND CHECKS FOR RAB, REOM SET. IF
1294                      ;* EITHER IS NOT SET, A RETURN IS MADE TO THE TEST, AT THE ADDRESS
1295                      ;* CONTAINED IN RETADR.
1296                      ;*****
1297 012156 013746 002366 CHKABT: MOV      AXNUM,-(SP) ;SAVE AX BYTE NO.
```

```
1298 012162 016637 000002 002336      MOV      2(SP),SUBRPC
1299 012170 162737 000004 002336      SUB      #4,SUBRPC      ;GET PC OF SUBROUTINE CALL
1300 012176 005037 002366      CLR      AXNUM      ;SET AX0 ADDRESS
1301 012202 004737 003624      JSR      PC,READAX      ;READ REG AX0
1302 012206 032737 000004 002356      BIT      #RABT,RAX16      ;CHK FOR RAB SET
1303 012214 001007      BNE      6$      ;BR IF RAB SET
1304 012216 004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
1305      ;REPORT RAB NOT SET
1306      ERRDF      40,EM40,ERR6

      012222      104455
      012224      000050
      012226      014063
      012230      017624
      1307 012232 000412      BR      8$
      1308 012234 032737 000002 002356 6$: BIT      #REOM,RAX16      ;CHK FOR REOM SET
      1309 012242 001015      BNE      9$      ;BR IF REOM SET
      1310 012244 004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
      1311      ;REPORT REOM NOT SET
      1312      ERRDF      31,EM31,ERR6

      012250      104455
      012252      000037
      012254      013706
      012256      017624
      1313 012260 011637 002366      8$: MOV      (SP),AXNUM      ;RESTORE AX BYTE NO.
      1314 012264 013706 002332      MOV      PSTACK,SP      ;RESTORE STACK POINTER TO BASE LEVEL
      1315 012270 013746 002346      MOV      RETADR,-(SP)      ;FIX ERROR RETURN PC
      1316 012274 000402      BR      16$
      1317 012276 012637 002366      9$: MOV      (SP)+,AXNUM      ;RESTORE AX BYTE NO.
      1318 012302 000207      16$: RTS      PC      ;RETURN
1319
1320
1321
1322
1323
```

```
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 012304    045    124    045  FMT1:  .ASCIIZ  /%T%06%N/
012307    117    066    045
012312    116    000
10 012314    045    116    045  FMT2:  .ASCIIZ  /%N%AFAILING REG: /
012317    101    106    101
012322    111    114    111
012325    116    107    040
012330    122    105    107
012333    072    040    000
11 012336    045    101    105  FMT3:  .ASCIIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%N/
012341    130    120    105
012344    103    124    105
012347    104    072    040
012352    045    117    063
012355    045    123    065
012360    045    101    101
012363    103    124    125
012366    101    114    072
012371    040    045    117
012374    063    045    116
012377    000
12 012400    045    116    045  FMT4:  .ASCIIZ  /%N%T%N%T%N/
012403    124    045    116
012406    045    124    045
012411    116    000
13 012413    045    117    063  FMT5:  .ASCIIZ  /%03%S5%03%S5%03%S5%03%N/
012416    045    123    065
012421    045    117    063
012424    045    123    065
012427    045    117    063
012432    045    123    065
012435    045    117    063
012440    045    116    000
14 012443    045    123    064  FMT6:  .ASCIIZ  /%S4%03%S5%03%S5%03%S5%03%N/
012446    045    117    063
012451    045    123    065
012454    045    117    063
012457    045    123    065
012462    045    117    063
012465    045    123    065
012470    045    117    063
012473    045    116    000
15 012476    045    124    045  FMT7:  .ASCIIZ  /%T%02%N/
012501    117    062    045
012504    116    000
16 012506    045    101    105  FMT8:  .ASCIIZ  /%AEXTENDED REG AX%01%A-%T%N/
012511    130    124    105
012514    116    104    105
012517    104    040    122
```

	012522	105	107	040	
	012525	101	130	045	
	012530	117	061	045	
	012533	101	055	045	
	012536	124	045	116	
	012541	000			
17	012542	045	124	045	FMT9: .ASCIZ /%T%N/
	012545	116	000		
18	012547	045	101	120	FMT10: .ASCIZ /%APC OF SUBR CALL: %06%N/
	012552	103	040	117	
	012555	106	040	123	
	012560	125	102	122	
	012563	040	103	101	
	012566	114	114	072	
	012571	040	045	117	
	012574	066	045	116	
	012577	000			
19	012600	045	101	122	FMT11: .ASCIZ /%AREG %02%A LOADED WITH: %03%N/
	012603	105	107	040	
	012606	045	117	062	
	012611	045	101	040	
	012614	114	117	101	
	012617	104	105	104	
	012622	040	127	111	
	012625	124	110	072	
	012630	040	045	117	
	012633	063	045	116	
	012636	000			
20	012637	045	101	115	FMT12: .ASCIZ /%AM8203 REG 11 (E121 SW10,9 , E134 SW9,10) = %03%N/
	012642	070	062	060	
	012645	063	040	122	
	012650	105	107	040	
	012653	061	061	040	
	012656	050	105	061	
	012661	062	061	040	
	012664	123	127	061	
	012667	060	054	071	
	012672	040	054	040	
	012675	105	061	063	
	012700	064	040	123	
	012703	127	071	054	
	012706	061	060	051	
	012711	040	075	040	
	012714	045	117	063	
	012717	045	116	000	
21	012722	045	101	115	FMT13: .ASCIZ /%AM8203 REG 15 (E134 SW1-8) = %03%N/
	012725	070	062	060	
	012730	063	040	122	
	012733	105	107	040	
	012736	061	065	040	
	012741	050	105	061	
	012744	063	064	040	
	012747	123	127	061	
	012752	055	070	051	
	012755	040	075	040	
	012760	045	117	063	
	012763	045	116	000	

22	012766	045	101	115	FMT14: .ASCIIZ /%AM8203 REG 16 (E121 SW1-8) = %03%N/
	012771	070	062	060	
	012774	063	040	122	
	012777	105	107	040	
	013002	061	066	040	
	013005	050	105	061	
	013010	062	061	040	
	013013	123	127	061	
	013016	055	070	051	
	013021	040	075	040	
	013024	045	117	063	
	013027	045	116	000	
23	013032	045	101	115	FMT15: .ASCIIZ /%AMODEM INTERFACE REG (AX3-15) = %03%N/
	013035	117	104	105	
	013040	115	040	111	
	013043	116	124	105	
	013046	122	106	101	
	013051	103	105	040	
	013054	122	105	107	
	013057	040	050	101	
	013062	130	063	055	
	013065	061	065	051	
	013070	040	075	040	
	013073	045	117	063	
	013076	045	116	000	
24	013101	045	116	045	FMT18: .ASCIIZ /%N%AFOR DEVICE AT ADRS %06%A ,%N/
	013104	101	106	117	
	013107	122	040	104	
	013112	105	126	111	
	013115	103	105	040	
	013120	101	124	040	
	013123	101	104	122	
	013126	123	040	040	
	013131	045	117	066	
	013134	045	101	040	
	013137	054	045	116	
	013142	000			
25	013143	045	116	045	FMT19: .ASCIIZ /%N%ATFST %D2%A NOT RUN%N/
	013146	101	124	105	
	013151	123	124	040	
	013154	045	104	062	
	013157	045	101	040	
	013162	116	117	124	
	013165	040	122	125	
	013170	116	045	116	
	013173	000			
26	013174	045	116	045	FMT25: .ASCIIZ /%N%ABAUd RATE INCORRECT/
	013177	101	102	101	
	013202	125	104	040	
	013205	122	101	124	
	013210	105	040	111	
	013213	116	103	117	
	013216	122	122	105	
	013221	103	124	000	
27	013224	045	116	045	FMT26: .ASCIIZ /%N%AMODEM INTERFACE NOT SELECTED/
	013227	101	115	117	
	013232	104	105	115	

	013235	040	111	116	
	013240	124	105	122	
	013243	106	101	103	
	013246	105	040	116	
	013251	117	124	040	
	013254	123	105	114	
	013257	105	103	124	
	013262	105	104	000	
28					
29					
30					
31	013265	122	105	107	EM2: .ASCIZ /REG NOT INITIALIZED BY MST CLR/
	013270	040	116	117	
	013273	124	040	111	
	013276	116	111	124	
	013301	111	101	114	
	013304	111	132	105	
	013307	104	040	102	
	013312	131	040	115	
	013315	123	124	040	
	013320	103	114	122	
	013323	000			
32	013324	122	105	107	EM3: .ASCIZ /REG MISCOMPARE/
	013327	040	115	111	
	013332	123	103	117	
	013335	115	120	101	
	013340	122	105	000	
33	013343	117	122	104	EM7: .ASCIZ /ORDY NOT SET/
	013346	131	040	116	
	013351	117	124	040	
	013354	123	105	124	
	013357	000			
34	013360	117	122	104	EM8: .ASCIZ /ORDY NOT CLEARED/
	013363	131	040	116	
	013366	117	124	040	
	013371	103	114	105	
	013374	101	122	105	
	013377	104	000		
35	013401	117	103	117	EM9: .ASCIZ /OCOR NOT SET/
	013404	122	040	116	
	013407	117	124	040	
	013412	123	105	124	
	013415	000			
36	013416	117	103	117	EM10: .ASCIZ /OCOR NOT CLEARED/
	013421	122	040	116	
	013424	117	124	040	
	013427	103	114	105	
	013432	101	122	105	
	013435	104	000		
37	013437	117	101	103	EM11: .ASCIZ /OACT NOT SET/
	013442	124	040	116	
	013445	117	124	040	
	013450	123	105	124	
	013453	000			
38	013454	117	101	103	EM12: .ASCIZ /OACT NOT CLEARED/
	013457	124	040	116	
	013462	117	124	040	

	013465	103	114	105	
	013470	101	122	105	
	013473	104	000		
39	013475	111	122	104	EM17: .ASCIZ /IRDY NOT SET/
	013500	131	040	116	
	013503	117	124	040	
	013506	123	105	124	
	013511	000			
40	013512	111	122	104	EM18: .ASCIZ /IRDY NOT CLEARED/
	013515	131	040	116	
	013520	117	124	040	
	013523	103	114	105	
	013526	101	122	105	
	013531	104	000		
41	013533	111	103	111	EM19: .ASCIZ /ICIR NOT SET/
	013536	122	040	116	
	013541	117	124	040	
	013544	123	105	124	
	013547	000			
42	013550	111	103	111	EM20: .ASCIZ /ICIR NOT CLEARED/
	013553	122	040	116	
	013556	117	124	040	
	013561	103	114	105	
	013564	101	122	105	
	013567	104	000		
43	013571	111	101	103	EM21: .ASCIZ /IACT NOT SET/
	013574	124	040	116	
	013577	117	124	040	
	013602	123	105	124	
	013605	000			
44	013606	111	101	103	EM22: .ASCIZ /IACT NOT CLEARED/
	013611	124	040	116	
	013614	117	124	040	
	013617	103	114	105	
	013622	101	122	105	
	013625	104	000		
45	013627	122	123	117	EM28: .ASCIZ /RSOM NOT CLEARED/
	013632	115	040	116	
	013635	117	124	040	
	013640	103	114	105	
	013643	101	122	105	
	013646	104	000		
46	013650	122	123	117	EM29: .ASCIZ /RSOM NOT SET/
	013653	115	040	116	
	013656	117	124	040	
	013661	123	105	124	
	013664	000			
47	013665	122	105	117	EM30: .ASCIZ /REOM NOT CLEARED/
	013670	115	040	116	
	013673	117	124	040	
	013676	103	114	105	
	013701	101	122	105	
	013704	104	000		
48	013706	122	105	117	EM31: .ASCIZ /REOM NOT SET/
	013711	115	040	116	
	013714	117	124	040	
	013717	123	105	124	

	013722	000			
49	013723	122	103	126	EM34: .ASCIIZ /RCV'D DATA MISCOMPARE/
	013726	047	104	040	
	013731	104	101	124	
	013734	101	040	115	
	013737	111	123	103	
	013742	117	115	120	
	013745	101	122	105	
	013750	000			
50	013751	102	103	103	EM35: .ASCIIZ /BCC NOT CLEARED/
	013754	040	116	117	
	013757	124	040	103	
	013762	114	105	101	
	013765	122	105	104	
	013770	000			
51	013771	102	103	103	EM36: .ASCIIZ /BCC NOT SET/
	013774	040	116	117	
	013777	124	040	123	
	014002	105	124	000	
52	014005	105	102	114	EM37: .ASCIIZ /EBLK NOT CLEARED/
	014010	113	040	116	
	014013	117	124	040	
	014016	103	114	105	
	014021	101	122	105	
	014024	104	000		
53	014026	105	102	114	EM38: .ASCIIZ /EBLK NOT SET/
	014031	113	040	116	
	014034	117	124	040	
	014037	123	105	124	
	014042	000			
54	014043	122	101	102	EM39: .ASCIIZ /RAB NOT CLEARED/
	014046	040	116	117	
	014051	124	040	103	
	014054	114	105	101	
	014057	122	105	104	
	014062	000			
55	014063	122	101	102	EM40: .ASCIIZ /RAB NOT SET/
	014066	040	116	117	
	014071	124	040	123	
	014074	105	124	000	
56	014077	117	126	122	EM41: .ASCIIZ /OVRR NOT CLEARED/
	014102	122	040	116	
	014105	117	124	040	
	014110	103	114	105	
	014113	101	122	105	
	014116	104	000		
57	014120	117	126	122	EM42: .ASCIIZ /OVRR NOT SET/
	014123	122	040	116	
	014126	117	124	040	
	014131	123	105	124	
	014134	000			
58	014135	122	105	107	EM43: .ASCIIZ /REG 11 SWITCHES INCORRECT/
	014140	040	061	061	
	014143	040	123	127	
	014146	111	124	103	
	014151	110	105	123	
	014154	040	111	116	

	014157	103	117	122	
	014162	122	105	103	
	014165	124	000		
59	014167	122	105	107	EM44: .ASCIIZ /REG 15 SWITCHES INCORRECT/
	014172	040	061	065	
	014175	040	123	127	
	014200	111	124	103	
	014203	110	105	123	
	014206	040	111	116	
	014211	103	117	122	
	014214	122	105	103	
	014217	124	000		
60	014221	122	105	107	EM45: .ASCIIZ /REG 16 SWITCHES INCORRECT/
	014224	040	061	066	
	014227	040	123	127	
	014232	111	124	103	
	014235	110	105	123	
	014240	040	111	116	
	014243	103	117	122	
	014246	122	105	103	
	014251	124	000		
61	014253	122	103	126	EM46: .ASCIIZ /RCV SILO NOT CLEARED BY IC/
	014256	040	123	111	
	014261	114	117	040	
	014264	116	117	124	
	014267	040	103	114	
	014272	105	101	122	
	014275	105	104	040	
	014300	102	131	040	
	014303	111	103	000	
62	014306	101	123	123	EM47: .ASCIIZ /ASSEMB BIT COUNT INCORRECT/
	014311	105	115	102	
	014314	040	102	111	
	014317	124	040	103	
	014322	117	125	116	
	014325	124	040	111	
	014330	116	103	117	
	014333	122	122	105	
	014336	103	124	000	
63	014341	117	104	104	EM48: .ASCIIZ /ODD VRC PARITY BIT NOT SET/
	014344	040	126	122	
	014347	103	040	120	
	014352	101	122	111	
	014355	124	131	040	
	014360	102	111	124	
	014363	040	116	117	
	014366	124	040	123	
	014371	105	124	000	
64	014374	117	104	104	EM49: .ASCIIZ /ODD VRC PARITY BIT NOT CLEARED/
	014377	040	126	122	
	014402	103	040	120	
	014405	101	122	111	
	014410	124	131	040	
	014413	102	111	124	
	014416	040	116	117	
	014421	124	040	103	
	014424	114	105	101	

CZDMSD M8203 STATIC TESTS #2
GLOBAL ERROR REPORT SECTION

MACRO V03.01 7-AUG-80 11:18:21 PAGE 18-7

N 7

SEQ 91

	014427	122	105	104	
	014432	000			
65	014433	105	126	105	EM50: .ASCIZ /EVEN VRC PARITY BIT NOT SET/
	014436	116	040	126	
	014441	122	103	040	
	014444	120	101	122	
	014447	111	124	131	
	014452	040	102	111	
	014455	124	040	116	
	014460	117	124	040	
	014463	123	105	124	
	014466	000			
66	014467	105	126	105	EM51: .ASCIZ /EVEN VRC PARITY BIT NOT CLEARED/
	014472	116	040	126	
	014475	122	103	040	
	014500	120	101	122	
	014503	111	124	131	
	014506	040	102	111	
	014511	124	040	116	
	014514	117	124	040	
	014517	103	114	105	
	014522	101	122	105	
	014525	104	000		
67	014527	124	105	123	EM52: .ASCIZ /TEST MODE NOT SET BY MAINT1/
	014532	124	040	115	
	014535	117	104	105	
	014540	040	116	117	
	014543	124	040	123	
	014546	105	124	040	
	014551	102	131	040	
	014554	115	101	111	
	014557	116	124	061	
	014562	000			
68	014563	124	105	123	EM53: .ASCIZ /TEST MODE NOT SET BY MAINT2/
	014566	124	040	115	
	014571	117	104	105	
	014574	040	116	117	
	014577	124	040	123	
	014602	105	124	040	
	014605	102	131	040	
	014610	115	101	111	
	014613	116	124	062	
	014616	000			
69	014617	124	130	040	EM54: .ASCIZ /TX UNDERRUN ERROR/
	014622	125	116	104	
	014625	105	122	122	
	014630	125	116	040	
	014633	105	122	122	
	014636	117	122	000	
70	014641	104	124	122	EM55: .ASCIZ /DTR NOT SET/
	014644	040	116	117	
	014647	124	040	123	
	014652	105	124	000	
71	014655	122	111	116	EM56: .ASCIZ /RING NOT SET/
	014660	107	040	116	
	014663	117	124	040	
	014666	123	105	124	

	014671	000			
72	014672	115	117	104	EM57: .ASCIZ /MODR NOT SET/
	014675	122	040	116	
	014700	117	124	040	
	014703	123	105	124	
	014706	000			
73	014707	110	104	130	EM58: .ASCIZ /HDX NOT SET/
	014712	040	116	117	
	014715	124	040	123	
	014720	105	124	000	
74	014723	123	124	102	EM59: .ASCIZ /STBY NOT SET/
	014726	131	040	116	
	014731	117	124	040	
	014734	123	105	124	
	014737	000			
75	014740	122	124	123	EM60: .ASCIZ /RTS NOT SET/
	014743	040	116	117	
	014746	124	040	123	
	014751	105	124	000	
76	014754	103	123	040	EM61: .ASCIZ /CS NOT SET/
	014757	116	117	124	
	014762	040	123	105	
	014765	124	000		
77	014767	103	101	122	EM62: .ASCIZ /CARR NOT SET/
	014772	122	040	116	
	014775	117	124	040	
	015000	123	105	124	
	015003	000			
78	015004	123	111	107	EM63: .ASCIZ /SIGQ NOT SET/
	015007	121	040	116	
	015012	117	124	040	
	015015	123	105	124	
	015020	000			
79	015021	123	111	107	EM64: .ASCIZ /SIGR NOT SET/
	015024	122	040	116	
	015027	117	124	040	
	015032	123	105	124	
	015035	000			
80	015036	122	124	123	EM65: .ASCIZ /RTS NOT CLEARED/
	015041	040	116	117	
	015044	124	040	103	
	015047	114	105	101	
	015052	122	105	104	
	015055	000			
81	015056	103	101	122	EM66: .ASCIZ /CARR NOT CLEARED/
	015061	122	040	116	
	015064	117	124	040	
	015067	103	114	105	
	015072	101	122	105	
	015075	104	000		
82					
83					
84					
85	015077	111	116	102	DH1: .ASCIZ &INBUS/OUTBUS REG &
	015102	125	123	057	
	015105	117	125	124	
	015110	102	125	123	

	015113	040	122	105		
	015116	107	040	000		
86	015121	114	111	116	DH2:	.ASCIZ /LINE UNIT INBUS REGS :/
	015124	105	040	125		
	015127	116	111	124		
	015132	040	111	116		
	015135	102	125	123		
	015140	040	122	105		
	015143	107	123	040		
	015146	072	000			
87	015150	122	105	107	DH3:	.ASCIZ /REG10 REG11 REG12 REG13/
	015153	061	060	040		
	015156	040	040	122		
	015161	105	107	061		
	015164	061	040	040		
	015167	040	122	105		
	015172	107	061	062		
	015175	040	040	040		
	015200	122	105	107		
	015203	061	063	000		
88	015206	040	040	040	DH4:	.ASCIZ / REG14 REG15 REG16 REG17/
	015211	040	122	105		
	015214	107	061	064		
	015217	040	040	040		
	015222	122	105	107		
	015225	061	065	040		
	015230	040	040	122		
	015233	105	107	061		
	015236	066	040	040		
	015241	040	122	105		
	015244	107	061	067		
	015247	000				
89	015250	061	065	000	DH5:	.ASCIZ /15/
90	015253	061	066	000	DH6:	.ASCIZ /16/
91	015256	114	111	116	DH7:	.ASCIZ /LINE UNIT EXTENDED REGS :/
	015261	105	040	125		
	015264	116	111	124		
	015267	040	105	130		
	015272	124	105	116		
	015275	104	105	104		
	015300	040	122	105		
	015303	107	123	040		
	015306	072	000			
92	015310	101	130	060	DH8:	.ASCIZ /AX0-15 AX0-16 AX1-15 AX1-16/
	015313	055	061	065		
	015316	040	040	101		
	015321	130	060	055		
	015324	061	066	040		
	015327	040	101	130		
	015332	061	055	061		
	015335	065	040	040		
	015340	101	130	061		
	015343	055	061	066		
	015346	000				
93	015347	040	040	040	DH9:	.ASCIZ / AX2-15 AX2-16 AX3-15 AX3-16/
	015352	040	101	130		
	015355	062	055	061		

015360	065	040	040
015363	101	130	062
015366	055	061	066
015371	040	040	101
015374	130	063	055
015377	061	065	040
015402	040	101	130
015405	063	055	061
015410	066	000	

94
95
96
97
98
99

100
101 015412 BGNMSG ERR1
015412
102 015412 PRINTB #FMT1,#ADDRES,MPCSR
015412 013746 002436
015416 012746 036544
015422 012746 012304
015426 012746 000003
015432 010600
015434 104414
015436 062706 000010
103 015442 ENDMSG
015442
015442 104423

104
105
106
107 015444 BGNMSG ERR2
015444
108 015444 PRINTB #FMT1,#ADDRES,MPCSR
015444 013746 002436
015450 012746 036544
015454 012746 012304
015460 012746 000003
015464 010600
015466 104414
015470 062706 000010
109 015474 PRINTB #FMT2
015474 012746 012314
015500 012746 000001
015504 010600
015506 104414
015510 062706 000004
110 015514 PRINTB #FMT7,#DH1,REGNUM
015514 013746 002364
015520 012746 015077
015524 012746 012476
015530 012746 000003
015534 010600
015536 104414
015540 062706 000010
111 015544 PRINTB #FMT3,GOODAT,BADDAT

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

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

015544 013746 002372
015550 013746 002370
015554 012746 012336
015560 012746 000003
015564 010600
015566 104414
015570 062706 000010

112 015574 012746 015150
015600 012746 015121
015604 012746 012400
015610 012746 000003
015614 010600
015616 104415

015620 062706 000010

113 015624 013746 002274
015630 013746 002272
015634 013746 002270
015640 013746 002266
015644 012746 012413
015650 012746 000005
015654 010600
015656 104415

015660 062706 000014

114 015664 012746 015206
015670 012746 012542
015674 012746 000002
015700 010600
015702 104415

015704 062706 000006

115 015710 013746 002304
015714 013746 002302
015720 013746 002300
015724 013746 002276
015730 012746 012443
015734 012746 000005
015740 010600
015742 104415

015744 062706 000014

116 015750
015750
015750 104423

117

118

119

120

121

122 015752

015752

123 015752

015752 013746 002436

015756 012746 036544

015762 012746 012304

015766 012746 000003

PRINTX #FMT4,#DH2,#DH3

PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13

PRINTX #FMT9,#DH4

PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17

ENDMSG

BGNMSG ERR3

PRINTB #FMT1,#ADDRES,MPCSR

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::

MOV MPCSR,-(SP)
MOV #ADDRES,-(SP)
MOV #FMT1,-(SP)
MOV #3,-(SP)

```
015772 010600
015774 104414
015776 062706 000010
124 016002 PRINTB #FMT2
016002 012746 012314
016006 012746 000001
016012 010600
016014 104414
016016 062706 000004
125 016022 PRINTB #FMT8,TMP1,TMP0
016022 013746 002516
016026 013746 002520
016032 012746 012506
016036 012746 000003
016042 010600
016044 104414
016046 062706 000010
126 016052 PRINTB #FMT3,GOODAT,BADDAT
016052 013746 002372
016056 013746 002370
016062 012746 012336
016066 012746 000003
016072 010600
016074 104414
016076 062706 000010
127 016102 PRINTX #FMT4,#DH2,#DH3
016102 012746 015150
016106 012746 015121
016112 012746 012400
016116 012746 000003
016122 010600
016124 104415
016126 062706 000010
128 016132 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
016132 013746 002274
016136 013746 002272
016142 013746 002270
016146 013746 002266
016152 012746 012413
016156 012746 000005
016162 010600
016164 104415
016166 062706 000014
129 016172 PRINTX #FMT9,#DH4
016172 012746 015206
016176 012746 012542
016202 012746 000002
016206 010600
016210 104415
016212 062706 000006
130 016216 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
016216 013746 002304
016222 013746 002302
016226 013746 002300
016232 013746 002276
016236 012746 012443
016242 012746 000005
```

```
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 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)
```

CZDMSD M8203 STATIC TESTS #2
GLOBAL ERROR REPORT SECTION

MACRO V03.01 7-AUG-80 11:18:21 PAGE 18-13

G 8

SEQ 97

131 016246 010600
016250 104415
016252 062706 000014

PRINTX #FMT4,#DH7,#DH8

016256 012746 015310
016262 012746 015256
016266 012746 012400
016272 012746 000003
016276 010600
016300 104415
016302 062706 000010

132 016306 013746 002314
016312 013746 002312
016316 013746 002310
016322 013746 002306
016326 012746 012413
016332 012746 000005
016336 010600
016340 104415
016342 062706 000014

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

133 016346 012746 015347
016352 012746 012542
016356 012746 000002
016362 010600
016364 104415
016366 062706 000006

PRINTX #FMT9,#DH9

134 016372 013746 002324
016376 013746 002322
016402 013746 002320
016406 013746 002316
016412 012746 012443
016416 012746 000005
016422 010600
016424 104415
016426 062706 000014

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

135 016432 000014
016432
016432 104423

ENDMSG

136
137
138
139
140

141 016434
016434
142 016434 013746 002336
016440 012746 012547
016444 012746 000002
016450 010600
016452 104414
016454 062706 000006

BGNMSG ERR4

PRINTB #FMT10,SUBRPC

143 016460 002436
016460 013746

PRINTB #FMT1,#ADDRES,MPCSR

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)
MOV SP,R0
TRAP C\$PNTB
ADD #6,SP

MOV MPCSR,-(SP)

144	016464	012746	036544		MOV	#ADDRES,-(SP)
	016470	012746	012304		MOV	#FMT1,-(SP)
	016474	012746	000003		MOV	#3,-(SP)
	016500	010600			MOV	SP,R0
	016502	104414			TRAP	C\$PNTB
	016504	062706	000010		ADD	#10,SP
	016510			PRINTB	#FMT2	
	016510	012746	012314		MOV	#FMT2,-(SP)
	016514	012746	000001		MOV	#1,-(SP)
	016520	010600			MOV	SP,R0
	016522	104414			TRAP	C\$PNTB
	016524	062706	000004		ADD	#4,SP
145	016530			PRINTB	#FMT7,#DH1,REGNUM	
	016530	013746	002364		MOV	REGNUM,-(SP)
	016534	012746	015077		MOV	#DH1,-(SP)
	016540	012746	012476		MOV	#FMT7,-(SP)
	016544	012746	000003		MOV	#3,-(SP)
	016550	010600			MOV	SP,R0
	016552	104414			TRAP	C\$PNTB
	016554	062706	000010		ADD	#10,SP
146	016560			PRINTX	#FMT4,#DH2,#DH3	
	016560	012746	015150		MOV	#DH3,-(SP)
	016564	012746	015121		MOV	#DH2,-(SP)
	016570	012746	012400		MOV	#FMT4,-(SP)
	016574	012746	000003		MOV	#3,-(SP)
	016600	010600			MOV	SP,R0
	016602	104415			TRAP	C\$PNTX
	016604	062706	000010		ADD	#10,SP
147	016610			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	016610	013746	002274		MOV	LUR13,-(SP)
	016614	013746	002272		MOV	LUR12,-(SP)
	016620	013746	002270		MOV	LUR11,-(SP)
	016624	013746	002266		MOV	LUR10,-(SP)
	016630	012746	012413		MOV	#FMT5,-(SP)
	016634	012746	000005		MOV	#5,-(SP)
	016640	010600			MOV	SP,R0
	016642	104415			TRAP	C\$PNTX
	016644	062706	000014		ADD	#14,SP
148	016650			PRINTX	#FMT9,#DH4	
	016650	012746	015206		MOV	#DH4,-(SP)
	016654	012746	012542		MOV	#FMT9,-(SP)
	016660	012746	000002		MOV	#2,-(SP)
	016664	010600			MOV	SP,R0
	016666	104415			TRAP	C\$PNTX
	016670	062706	000006		ADD	#6,SP
149	016674			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	016674	013746	002304		MOV	LUR17,-(SP)
	016700	013746	002302		MOV	LUR16,-(SP)
	016704	013746	002300		MOV	LUR15,-(SP)
	016710	013746	002276		MOV	LUR14,-(SP)
	016714	012746	012443		MOV	#FMT6,-(SP)
	016720	012746	000005		MOV	#5,-(SP)
	016724	010600			MOV	SP,R0
	016726	104415			TRAP	C\$PNTX
	016730	062706	000014		ADD	#14,SP
150	016734			PRINTX	#FMT4,#DH7,#DH8	
	016734	012746	015310		MOV	#DH8,-(SP)

CZDMSD M8203 STATIC TESTS #2
GLOBAL ERROR REPORT SECTION

MACRO V03.01 7-AUG-80 11:18:21 PAGE 18-15

1 8

SEQ 99

151	016740	012746	015256			MOV	#DH7,-(SP)
	016744	012746	012400			MOV	#FMT4,-(SP)
	016750	012746	000003			MOV	#3,-(SP)
	016754	010600				MOV	SP,R0
	016756	104415				TRAP	C\$PNTX
	016760	062706	000010			ADD	#10,SP
	016764			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16		
	016764	013746	002314			MOV	AX1.16,-(SP)
	016770	013746	002312			MOV	AX1.15,-(SP)
	016774	013746	002310			MOV	AX0.16,-(SP)
	017000	013746	002306			MOV	AX0.15,-(SP)
	017004	012746	012413			MOV	#FMT5,-(SP)
	017010	012746	000005			MOV	#5,-(SP)
	017014	010600				MOV	SP,R0
	017016	104415				TRAP	C\$PNTX
	017020	062706	000014			ADD	#14,SP
152	017024			PRINTX	#FMT9,#DH9		
	017024	012746	015347			MOV	#DH9,-(SP)
	017030	012746	012542			MOV	#FMT9,-(SP)
	017034	012746	000002			MOV	#2,-(SP)
	017040	010600				MOV	SP,R0
	017042	104415				TRAP	C\$PNTX
	017044	062706	000006			ADD	#6,SP
153	017050			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16		
	017050	013746	002324			MOV	AX3.16,-(SP)
	017054	013746	002322			MOV	AX3.15,-(SP)
	017060	013746	002320			MOV	AX2.16,-(SP)
	017064	013746	002316			MOV	AX2.15,-(SP)
	017070	012746	012443			MOV	#FMT6,-(SP)
	017074	012746	000005			MOV	#5,-(SP)
	017100	010600				MOV	SP,R0
	017102	104415				TRAP	C\$PNTX
	017104	062706	000014			ADD	#14,SP
154	017110			ENDMSG			
	017110					L10005:	
	017110	104423				TRAP	C\$MSG
155							
156							
157							
158							
159							
160	017112			BGNMSG	ERR5		
	017112					ERR5::	
161	017112			PRINTB	#FMT1,#ADDRES,MPCSR		
	017112	013746	002436			MOV	MPCSR,-(SP)
	017116	012746	036544			MOV	#ADDRES,-(SP)
	017122	012746	012304			MOV	#FMT1,-(SP)
	017126	012746	000003			MOV	#3,-(SP)
	017132	010600				MOV	SP,R0
	017134	104414				TRAP	C\$PNTB
	017136	062706	000010			ADD	#10,SP
162	017142			PRINTB	#FMT11,REGNUM,LOADAT		
	017142	013746	002374			MOV	LOADAT,-(SP)
	017146	013746	002364			MOV	REGNUM,-(SP)
	017152	012746	012600			MOV	#FMT11,-(SP)
	017156	012746	000003			MOV	#3,-(SP)
	017162	010600				MOV	SP,R0

163 017164 104414 000010
017166 062706 000010
017172 012746 012314

PRINTB #FMT2

TRAP C\$PNTB
ADD #10,SP017176 012746 000001
017202 010600MOV #FMT2,-(SP)
MOV #1,-(SP)017204 104414
017206 062706 000004MOV SP,RO
TRAP C\$PNTB
ADD #4,SP164 017212 013746 002516
017216 013746 002520

PRINTB #FMT8,TMP1,TMP0

MOV TMP0,-(SP)
MOV TMP1,-(SP)017222 012746 012506
017226 012746 000003MOV #FMT8,-(SP)
MOV #3,-(SP)017232 010600
017234 104414MOV SP,RO
TRAP C\$PNTB

017236 062706 000010

ADD #10,SP

165 017242 013746 002372
017246 013746 002370

PRINTB #FMT3,GOODAT,BADDAT

MOV BADDAT,-(SP)
MOV GOODAT,-(SP)017252 012746 012336
017256 012746 000003MOV #FMT3,-(SP)
MOV #3,-(SP)017262 010600
017264 104414MOV SP,RO
TRAP C\$PNTB

017266 062706 000010

ADD #10,SP

166 017272 012746 015150
017276 012746 015121

PRINTX #FMT4,#DH2,#DH3

MOV #DH3,-(SP)
MOV #DH2,-(SP)017302 012746 012400
017306 012746 000003MOV #FMT4,-(SP)
MOV #3,-(SP)017312 010600
017314 104415MOV SP,RO
TRAP C\$PNTX

017316 062706 000010

ADD #10,SP

167 017322 013746 002274
017326 013746 002272

PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13

MOV LUR13,-(SP)
MOV LUR12,-(SP)017332 013746 002270
017336 013746 002266MOV LUR11,-(SP)
MOV LUR10,-(SP)017342 012746 012413
017346 012746 000005MOV #FMT5,-(SP)
MOV #5,-(SP)017352 010600
017354 104415MOV SP,RO
TRAP C\$PNTX

017356 062706 000014

ADD #14,SP

168 017362 012746 015206
017366 012746 012542

PRINTX #FMT9,#DH4

MOV #DH4,-(SP)
MOV #FMT9,-(SP)017372 012746 000002
017376 010600MOV #2,-(SP)
MOV SP,RO017400 104415
017402 062706 000006TRAP C\$PNTX
ADD #6,SP169 017406 013746 002304
017412 013746 002302

PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17

MOV LUR17,-(SP)
MOV LUR16,-(SP)017416 013746 002300
017422 013746 002276MOV LUR15,-(SP)
MOV LUR14,-(SP)017426 012746 012443
017432 012746 000005MOV #FMT6,-(SP)
MOV #5,-(SP)

017436 010600

MOV SP,RO

017660	012746	012304		MOV	#FMT1,-(SP)
017664	012746	000003		MOV	#3,-(SP)
017670	010600			MOV	SP,R0
017672	104414			TRAP	C\$PNTB
017674	062706	000010		ADD	#10,SP
183 017700			PRINTB #FMT2		
017700	012746	012314		MOV	#FMT2,-(SP)
017704	012746	000001		MOV	#1,-(SP)
017710	010600			MOV	SP,R0
017712	104414			TRAP	C\$PNTB
017714	062706	000004		ADD	#4,SP
184 017720			PRINTB #FMT8,TMP1,TMP0		
017720	013746	002516		MOV	TMP0,-(SP)
017724	013746	002520		MOV	TMP1,-(SP)
017730	012746	012506		MOV	#FMT8,-(SP)
017734	012746	000003		MOV	#3,-(SP)
017740	010600			MOV	SP,R0
017742	104414			TRAP	C\$PNTB
017744	062706	000010		ADD	#10,SP
185 017750			PRINTX #FMT4,#DH2,#DH3		
017750	012746	015150		MOV	#DH3,-(SP)
017754	012746	015121		MOV	#DH2,-(SP)
017760	012746	012400		MOV	#FMT4,-(SP)
017764	012746	000003		MOV	#3,-(SP)
017770	010600			MOV	SP,R0
017772	104415			TRAP	C\$PNTX
017774	062706	000010		ADD	#10,SP
186 020000			PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13		
020000	013746	002274		MOV	LUR13,-(SP)
020004	013746	002272		MOV	LUR12,-(SP)
020010	013746	002270		MOV	LUR11,-(SP)
020014	013746	002266		MOV	LUR10,-(SP)
020020	012746	012413		MOV	#FMT5,-(SP)
020024	012746	000005		MOV	#5,-(SP)
020030	010600			MOV	SP,R0
020032	104415			TRAP	C\$PNTX
020034	062706	000014		ADD	#14,SP
187 020040			PRINTX #FMT9,#DH4		
020040	012746	015206		MOV	#DH4,-(SP)
020044	012746	012542		MOV	#FMT9,-(SP)
020050	012746	000002		MOV	#2,-(SP)
020054	010600			MOV	SP,R0
020056	104415			TRAP	C\$PNTX
020060	062706	000006		ADD	#6,SP
188 020064			PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17		
020064	013746	002304		MOV	LUR17,-(SP)
020070	013746	002302		MOV	LUR16,-(SP)
020074	013746	002300		MOV	LUR15,-(SP)
020100	013746	002276		MOV	LUR14,-(SP)
020104	012746	012443		MOV	#FMT6,-(SP)
020110	012746	000005		MOV	#5,-(SP)
020114	010600			MOV	SP,R0
020116	104415			TRAP	C\$PNTX
020120	062706	000014		ADD	#14,SP
189 020124			PRINTX #FMT4,#DH7,#DH8		
020124	012746	015310		MOV	#DH8,-(SP)
020130	012746	015256		MOV	#DH7,-(SP)

CZDMSD M8203 STATIC TESTS #2
GLOBAL ERROR REPORT SECTION

MACRO V03.01 7-AUG-80 11:18:21 PAGE 18-19

M 8

SEQ 103

	020134	012746	012400		MOV	#FMT4,-(SP)
	020140	012746	000003		MOV	#3,-(SP)
	020144	010600			MOV	SP,R0
	020146	104415			TRAP	C\$PNTX
	020150	062706	000010		ADD	#10,SP
190	020154			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16	
	020154	013746	002314		MOV	AX1.16,-(SP)
	020160	013746	002312		MOV	AX1.15,-(SP)
	020164	013746	002310		MOV	AX0.16,-(SP)
	020170	013746	002306		MOV	AX0.15,-(SP)
	020174	012746	012413		MOV	#FMT5,-(SP)
	020200	012746	000005		MOV	#5,-(SP)
	020204	010600			MOV	SP,R0
	020206	104415			TRAP	C\$PNTX
	020210	062706	000014		ADD	#14,SP
191	020214			PRINTX	#FMT9,#DH9	
	020214	012746	015347		MOV	#DH9,-(SP)
	020220	012746	012542		MOV	#FMT9,-(SP)
	020224	012746	000002		MOV	#2,-(SP)
	020230	010600			MOV	SP,R0
	020232	104415			TRAP	C\$PNTX
	020234	062706	000006		ADD	#6,SP
192	020240			PRINTX	#FMT6,AX2.15,AX2.16,AX3.15,AX3.16	
	020240	013746	002324		MOV	AX3.16,-(SP)
	020244	013746	002322		MOV	AX3.15,-(SP)
	020250	013746	002320		MOV	AX2.16,-(SP)
	020254	013746	002316		MOV	AX2.15,-(SP)
	020260	012746	012443		MOV	#FMT6,-(SP)
	020264	012746	000005		MOV	#5,-(SP)
	020270	010600			MOV	SP,R0
	020272	104415			TRAP	C\$PNTX
	020274	062706	000014		ADD	#14,SP
193	020300			ENDMSG		
	020300					
	020300	104423				
194						
195						
196						
197						
198						
199	020302			BGNMSG	ERR7	
	020302					
200	020302			PRINTB	#FMT1,#ADDRES,MPCSR	
	020302	013746	002436			
	020306	012746	036544			
	020312	012746	012304			
	020316	012746	000003			
	020322	010600				
	020324	104414				
	020326	062706	000010			
201	020332			PRINTB	#FMT2	
	020332	012746	012314			
	020336	012746	000001			
	020342	010600				
	020344	104414				
	020346	062706	000004			
202	020352			PRINTB	#FMT7,#DH1,REGNUM	

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

	020352	013746	002364		MOV	REGNUM,-(SP)
	020356	012746	015077		MOV	#DH1,-(SP)
	020362	012746	012476		MOV	#FMT7,-(SP)
	020366	012746	000003		MOV	#3,-(SP)
	020372	010600			MOV	SP,R0
	020374	104414			TRAP	C\$PNTB
	020376	062706	000010		ADD	#10,SP
203	020402			PRINTX	#FMT4,#DH2,#DH3	
	020402	012746	015150		MOV	#DH3,-(SP)
	020406	012746	015121		MOV	#DH2,-(SP)
	020412	012746	012400		MOV	#FMT4,-(SP)
	020416	012746	000003		MOV	#3,-(SP)
	020422	010600			MOV	SP,R0
	020424	104415			TRAP	C\$PNTX
	020426	062706	000010		ADD	#10,SP
204	020432			PRINTX	#FMT5,LUR10,LUR11,LUR12,LUR13	
	020432	013746	002274		MOV	LUR13,-(SP)
	020436	013746	002272		MOV	LUR12,-(SP)
	020442	013746	002270		MOV	LUR11,-(SP)
	020446	013746	002266		MOV	LUR10,-(SP)
	020452	012746	012413		MOV	#FMT5,-(SP)
	020456	012746	000005		MOV	#5,-(SP)
	020462	010600			MOV	SP,R0
	020464	104415			TRAP	C\$PNTX
	020466	062706	000014		ADD	#14,SP
205	020472			PRINTX	#FMT9,#DH4	
	020472	012746	015206		MOV	#DH4,-(SP)
	020476	012746	012542		MOV	#FMT9,-(SP)
	020502	012746	000002		MOV	#2,-(SP)
	020506	010600			MOV	SP,R0
	020510	104415			TRAP	C\$PNTX
	020512	062706	000006		ADD	#6,SP
206	020516			PRINTX	#FMT6,LUR14,LUR15,LUR16,LUR17	
	020516	013746	002304		MOV	LUR17,-(SP)
	020522	013746	002302		MOV	LUR16,-(SP)
	020526	013746	002300		MOV	LUR15,-(SP)
	020532	013746	002276		MOV	LUR14,-(SP)
	020536	012746	012443		MOV	#FMT6,-(SP)
	020542	012746	000005		MOV	#5,-(SP)
	020546	010600			MOV	SP,R0
	020550	104415			TRAP	C\$PNTX
	020552	062706	000014		ADD	#14,SP
207	020556			PRINTX	#FMT4,#DH7,#DH8	
	020556	012746	015310		MOV	#DH8,-(SP)
	020562	012746	015256		MOV	#DH7,-(SP)
	020566	012746	012400		MOV	#FMT4,-(SP)
	020572	012746	000003		MOV	#3,-(SP)
	020576	010600			MOV	SP,R0
	020600	104415			TRAP	C\$PNTX
	020602	062706	000010		ADD	#10,SP
208	020606			PRINTX	#FMT5,AX0.15,AX0.16,AX1.15,AX1.16	
	020606	013746	002314		MOV	AX1.16,-(SP)
	020612	013746	002312		MOV	AX1.15,-(SP)
	020616	013746	002310		MOV	AX0.16,-(SP)
	020622	013746	002306		MOV	AX0.15,-(SP)
	020626	012746	012413		MOV	#FMT5,-(SP)
	020632	012746	000005		MOV	#5,-(SP)

020636	010600				MOV	SP,R0
020640	104415				TRAP	C\$PNTX
020642	062706	000014			ADD	#14,SP
209 020646			PRINTX	#FM19,#DH9		
020646	012746	015347			MOV	#DH9,-(SP)
020652	012746	012542			MOV	#FM19,-(SP)
020656	012746	000002			MOV	#2,-(SP)
020662	010600				MOV	SP,R0
020664	104415				TRAP	C\$PNTX
020666	062706	000006			ADD	#6,SP
210 020672			PRINTX	#FM16,AX2.15,AX2.16,AX3.15,AX3.16		
020672	013746	002324			MOV	AX3.16,-(SP)
020676	013746	002322			MOV	AX3.15,-(SP)
020702	013746	002320			MOV	AX2.16,-(SP)
020706	013746	002316			MOV	AX2.15,-(SP)
020712	012746	012443			MOV	#FM16,-(SP)
020716	012746	000005			MOV	#5,-(SP)
020722	010600				MOV	SP,R0
020724	104415				TRAP	C\$PNTX
020726	062706	000014			ADD	#14,SP
211 020732			ENDMSG			
020732						
020732	104423				L10010:	TRAP C\$MSG
212						
213						
214						
215						
216						
217 020734			BGNMSG	ERR8		
020734					ERR8::	
218 020734			PRINTB	#FM10,SUBRPC		
020734	013746	002336			MOV	SUBRPC,-(SP)
020740	012746	012547			MOV	#FM10,-(SP)
020744	012746	000002			MOV	#2,-(SP)
020750	010600				MOV	SP,R0
020752	104414				TRAP	C\$PNTB
020754	062706	000006			ADD	#6,SP
219 020760			PRINTB	#FM11,#ADDRES,MPCSR		
020760	013746	002436			MOV	MPCSR,-(SP)
020764	012746	036544			MOV	#ADDRES,-(SP)
020770	012746	012304			MOV	#FM11,-(SP)
020774	012746	000003			MOV	#3,-(SP)
021000	010600				MOV	SP,R0
021002	104414				TRAP	C\$PNTB
021004	062706	000010			ADD	#10,SP
220 021010			PRINTB	#FM12		
021010	012746	012314			MOV	#FM12,-(SP)
021014	012746	000001			MOV	#1,-(SP)
021020	010600				MOV	SP,R0
021022	104414				TRAP	C\$PNTB
021024	062706	000004			ADD	#4,SP
221 021030			PRINTB	#FM17,#DH1,REGNUM		
021030	013746	002364			MOV	REGNUM,-(SP)
021034	012746	015077			MOV	#DH1,-(SP)
021040	012746	012476			MOV	#FM17,-(SP)
021044	012746	000003			MOV	#3,-(SP)
021050	010600				MOV	SP,R0

```
021052 104414
021054 062706 000010
222 021060 PRINTB #FMT3,GOODAT,BADDAT
021060 013746 002372
021064 013746 002370
021070 012746 012336
021074 012746 000003
021100 010600
021102 104414
021104 062706 000010
223 021110 PRINTX #FMT4,#DH2,#DH3
021110 012746 015150
021114 012746 015121
021120 012746 012400
021124 012746 000003
021130 010600
021132 104415
021134 062706 000010
224 021140 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
021140 013746 002274
021144 013746 002272
021150 013746 002270
021154 013746 002266
021160 012746 012413
021164 012746 000005
021170 010600
021172 104415
021174 062706 000014
225 021200 PRINTX #FMT9,#DH4
021200 012746 015206
021204 012746 012542
021210 012746 000002
021214 010600
021216 104415
021220 062706 000006
226 021224 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
021224 013746 002304
021230 013746 002302
021234 013746 002300
021240 013746 002276
021244 012746 012443
021250 012746 000005
021254 010600
021256 104415
021260 062706 000014
227 021264 PRINTX #FMT4,#DH7,#DH8
021264 012746 015310
021270 012746 015256
021274 012746 012400
021300 012746 000003
021304 010600
021306 104415
021310 062706 000010
228 021314 PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
021314 013746 002314
021320 013746 002312
021324 013746 002310
```

```
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
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)
```


021546 012746 015121
021552 012746 012400
021556 012746 000003
021562 010600
021564 104415
021566 062706 000010
242 021572 PRINTX #FMT5,LUR10,LUR11,LUR12,LUR13
021572 013746 002274
021576 013746 002272
021602 013746 002270
021606 013746 002266
021612 012746 012413
021616 012746 000005
021622 010600
021624 104415
021626 062706 000014
243 021632 PRINTX #FMT9,#DH4
021632 012746 015206
021636 012746 012542
021642 012746 000002
021646 010600
021650 104415
021652 062706 000006
244 021656 PRINTX #FMT6,LUR14,LUR15,LUR16,LUR17
021656 013746 002304
021662 013746 002302
021666 013746 002300
021672 013746 002276
021676 012746 012443
021702 012746 000005
021706 010600
021710 104415
021712 062706 000014
245 021716 PRINTX #FMT4,#DH7,#DH8
021716 012746 015310
021722 012746 015256
021726 012746 012400
021732 012746 000003
021736 010600
021740 104415
021742 062706 000010
246 021746 PRINTX #FMT5,AX0.15,AX0.16,AX1.15,AX1.16
021746 013746 002314
021752 013746 002312
021756 013746 002310
021762 013746 002306
021766 012746 012413
021772 012746 000005
021776 010600
022000 104415
022002 062706 000014
247 022006 PRINTX #FMT9,#DH9
022006 012746 015347
022012 012746 012542
022016 012746 000002
022022 010600
022024 104415

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)
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

248 022026 062706 000006
022032
022032 013746 002324
022036 013746 002322
022042 013746 002320
022046 013746 002316
022052 012746 012443
022056 012746 000005
022062 010600
022064 104415
022066 062706 000014
249 022072
022072
022072 104423
250
251
252
253
254

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

ENDMSG

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

L10012:

TRAP C\$MSG

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

9 022074
022074
10
11 022074
022074
022074 104425

LSRPT::

L10013:

.EVEN

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 022104      BGNINIT
9 022104
10 022104 010637 002332      MOV      SP,PSTACK      ;SAVE BASE-LEVEL STACK POINTER
11 022110 005037 002336      CLR      SUBRPC        ;CLEAR SUBR CALL PC
12 022114 005037 002416      CLR      DISILO        ;CLEAR CURRENT STATE OF DISSI
13 022120 005037 002420      CLR      CHPTYP        ;CLEAR USYRT CHIP TYPE INDICATOR
14 022124 005037 002410      CLR      ERROR1        ;CLEAR ERROR FLAGS
15 022130 005037 002424      CLR      SAVLEN        ;CLEAR CHAR LENGTH FROM SETJP
16 022134 005737 002376      TST      FRSTIM        ;SEE IF FIRST TIME THROUGH AFTER LOAD
17 022140 001007              BNE      6$           ;BR IF NOT
18 022142 013737 000004 002404      MOV      @#4,SAVE4      ;SAVE ERROR TRAP VECTOR
19 022150 013737 000006 002406      MOV      @#6,SAVE6
20 022156 000406              BR      9$
21 022160 013737 002404 000004 6$:      MOV      SAVE4,@#4      ;RESTORE ERROR TRAP VECTOR
22 022166 013737 002406 000006      MOV      SAVE6,@#6
23 022174 012737 000001 002376 9$:      MOV      #1,FRSTIM      ;MARK FLAG FOR NEXT TIME THROUGH
24      ;SEE IF PROGRAM JUST STARTED, BR IF YES
25 022202      READEF  #EF.START
26 022202 012700 000040              MOV      #EF.START,RO
27 022206 104447              TRAP      C$REFG
28 022210      BCOMPLETE      STARST
29 022210 103415              BCS      STARST
30      ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
31 022212      READEF  #EF.RESTART
32 022212 012700 000037              MOV      #EF.RESTART,RO
33 022216 104447              TRAP      C$REFG
34 022220      BCOMPLETE      STARST
35 022220 103411              BCS      STARST
36      ;SEE IF THIS IS A NEW PASS, BR IF YES
37 022222      READEF  #EF.NEW
38 022222 012700 000035              MOV      #EF.NEW,RO
39 022226 104447              TRAP      C$REFG
40 022230      BCOMPLETE      NEWST
41 022230 103411              BCS      NEWST
42      ;SEE IF PROGRAM WAS JUST CONTINUED
43 022232      READEF  #EF.CONTINUE
44 022232 012700 000036              MOV      #EF.CONTINUE,RO
45 022236 104447              TRAP      C$REFG
46 022240      BCOMPLETE      ENDIT
47 022240 103556              BCS      ENDIT
48 022242 000416              BR      GETPRM
49 022244      STARST:      CLR      STARES      ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
50 022244 005037 002402      ;CLEAR DEVICE MAP
51 022244      CLR      DEVMAP
52 022250 005037 002426      NEWST:      MOV      #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
53 022254 012737 177777 002330      INC      FRSPAS      ;INCREMENT NO. OF PASSES AFTER LOAD
54 022262 005237 002400      INC      STARES      ;INCREMENT NO. OF PASSES SINCE STA OR RES
55 022266 005237 002402
```

```
45 022272 012737 000001 002430      MOV    #BIT0,DEVPTR      ;INIT DEVICE MAP BIT POINTER
46                                     ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
47                                     ; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
48 022300                                     GETPRM:
49 022300 005237 002330      INC    LOGDEV      ;INCREMENT LOGICAL DEVICE NUMBER
50 022304 023737 002330 002012      CMP    LOGDEV,LSUNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
51 022312 002360      BGE    NEWST      ;BR IF YES
52 022314      GPWARD LOGDEV,R1      ;GET P-TABLE POINTER INTO R1
      022314 013700 002330      MOV    LOGDEV,R0
      022320 104442      TRAP    C$GPHRD
      022322 010001      MOV    R0,R1
53 022324      BCOMPLETE      10$      ;BR IF DEVICE AVAILABLE
      022324 103403      BCS      10$
54 022326 006337 002430      ASL    DEVPTR      ;SHIFT DEVICE MAP BIT POINTER
55 022332 000762      BR    GETPRM      ;SKIP THIS DEVICE
56 022334 053737 002430 002426 10$:  BIS    DEVPTR,DEVMAP    ;SET BIT FOR THIS DEVICE IN DEVICE MAP
57 022342 006337 002430      ASL    DEVPTR      ;SHIFT DEVICE MAP BIT POINTER
58 022346 062701 000002      ADD    #2,R1      ;INCREMENT R1 PAST MICROPROCESSOR TYPE
59 022352 011137 002436      MOV    (R1),MPCSR    ;STORE POINTER TO MICROPROCESSOR CSR'S
60 022356 011137 002440      MOV    (R1),BSEL1
61 022362 005237 002440      INC    BSEL1      ;GET POINTER TO BSEL1 (MAINTENANCE REGISTER)
62 022366 011137 002442      MOV    (R1),SEL4
63 022372 062737 000004 002442      ADD    #4,SEL4      ;GET POINTER TO SEL4
64 022400 012137 002444      MOV    (R1)+,SEL6
65 022404 062737 000006 002444      ADD    #6,SEL6      ;STORE POINTER TO SEL6
66 022412 011137 002446      MOV    (R1),MPIVEC    ;GET MICROPROCESSOR INPUT INTRPT VECTOR
67 022416 012137 002450      MOV    (R1)+,MPOVEC
68 022422 062737 000004 002450      ADD    #4,MPOVEC    ;GET MICROPROCESSOR OUTPUT INTRPT VECTOR
69 022430 012137 002452      MOV    (R1)+,MPRIOR    ;GET MICROPROCESSOR DEVICE PRIORITY
70 022434 062701 000002      ADD    #2,R1      ;INCREMENT R1 PAST LU TYPE
71 022440 012137 002454      MOV    (R1)+,LUSW11    ;GET LU SWITCH PACK #1
72 022444 012137 002456      MOV    (R1)+,LUSW12    ;GET LU SWITCH PACK #2
73 022450 012137 002460      MOV    (R1)+,LUSW13    ;GET LU SWITCH PACK #3
74 022454 012137 002462      MOV    (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
75 022460 011137 002464      MOV    (R1),BDRATE    ;GET BAUD RATE
76                                     ;SEE IF MANUAL INTERVENTION DESIRED BETWEEN UNITS FOR INSTALLATION OR REMOVAL
77                                     ; OF TEST CONNECTORS, BR IF NOT
78 022464 005737 002256      TST    MIFLAG
79 022470 001442      BEQ    22$
80                                     ;SEE IF MANUAL INTERVENTION ALLOWED BY SUPERVISOR
81 022472      MANUAL
      022472 104450      TRAP    C$MANI
82                                     ;BR IF ALLOWED
83 022474      BCOMPLETE      18$      BCS      18$
      022474 103412
84                                     ;PRINT MSG THAT OPERATOR INTERVENTION IS NOT ALLOWED
85 022476      PRINTF #FMT16
      022476 012746 022600      MOV    #FMT16,-(SP)
      022502 012746 000001      MOV    #1,-(SP)
      022506 010600      MOV    SP,R0
      022510 104417      TRAP    C$PNTF
      022512 062706 000004      ADD    #4,SP
86 022516 16$:  BREAK      ;HANG UNTIL ^C TYPED
      022516 104422      TRAP    C$BRK
87 022520 000776      BR    16$
88 022522 18$:
89                                     ;TYPE "INSTALL TEST CONNECTOR(S) ON UNIT AT ADRS XXXXXX"
```

```
90 022522          PRINTF  #FMT17,MPCSR
    022522 013746 002436
    022526 012746 022722
    022532 012746 000002
    022536 010600
    022540 104417
    022542 062706 000006
91 022546 005037 002502
92 022552
93
94 022552          CLR      REG2
    022552 104443
    022554 000404
    022556 002502
    022560 000120
    022562 023011
    022564 000001
    022566
95 022566 023727 002502 000001
96 022574 001366
97 022576
98 022576
99 022576
    022576
    022576 104411
100
101 022600          045      116      045
    022603          101      115      101
    022606          116      125      101
    022611          114      040      111
    022614          116      124      105
    022617          122      126      105
    022622          116      124      111
    022625          117      116      040
    022630          111      123      040
    022633          116      117      124
    022636          040      101      114
    022641          114      117      127
    022644          105      104      041
    022647          045      116
102 022651          045      101      124
    022654          131      120      105
    022657          040      103      117
    022662          116      124      122
    022665          117      114      055
    022670          103      040      050
    022673          136      103      051
    022676          040      074      103
    022701          122      076      040
    022704          124      117      040
    022707          120      122      117
    022712          103      105      105
    022715          104      072      045
    022720          116      000
103 022722          045      116      045
    022725          101      111      116
    022730          123      124      101
```

20\$:
;ASK OPERATOR TO 'TYPE <Y> <CR> WHEN READY TO PROCEED'
GMANIL TYPEY,REG2,1,NO

22\$:
ENDIT:
ENDINIT

10000\$:
TRAP CSGMAN
BR 10000\$
.WORD REG2
.WORD TSCODE
.WORD TYPEY
.WORD 1

L10015:
TRAP CSINIT

FMT16: .ASCII /%N%AMANUAL INTERVENTION IS NOT ALLOWED!%N/

.ASCIIZ /%ATYPE CONTROL-C (^C) <CR> TO PROCEED:%N/

FMT17: .ASCIIZ /%N%AINSTALL TEST CONNECTOR(S) ON UNIT AT ADRS : %06%N/

	022733	114	114	040
	022736	124	105	123
	022741	124	040	103
	022744	117	116	116
	022747	105	103	124
	022752	117	122	050
	022755	123	051	040
	022760	117	116	040
	022763	125	116	111
	022766	124	040	101
	022771	124	040	101
	022774	104	122	123
	022777	040	072	040
	023002	040	045	117
	023005	066	045	116
	023010	000		
104	023011	124	131	120
	023014	105	040	074
	023017	131	076	074
	023022	103	122	076
	023025	040	127	110
	023030	105	116	040
	023033	122	105	101
	023036	104	131	040
	023041	124	117	040
	023044	120	122	117
	023047	103	105	105
	023052	104	040	000

TYPEY: .ASCIIZ /TYPE <Y><CR> WHEN READY TO PROCEED /

.EVEN

105
 106
 107
 108
 109
 110

```
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
8 023056      BGNAUTO
9 023056
10          ;ESTABLISH PRIORITY = 7
11          SETPRI  #PRI07
12          MOV     #6$,@#4      ;SET UP NON-EXISTENT MEMORY ERROR TRAP VECTOR
13          MOV     #PRI07,@#6
14          TSI     @MPCSR      ;ADDRESS SELO
15          BR      9$          ;TAKE THIS BRANCH IF DEVICE RESPONDS
16          ;COME HERE IF DEVICE CSR IS NON-EXISTENT
17 6$:      ADD     #4,SP      ;CLEAN UP THE STACK POINTER
18          DODU     LOGDEV    ;DROP THIS UNIT FROM TESTING
19          MOV     LOGDEV,R0
20          TRAP     C$DODU
21          MOV     SAVE4,@#4    ;RESTORE ERROR TRAP VECTOR
22          MOV     SAVE6,@#6
23          ENDAUTO
24
25          L10016:
26          TRAP     C$AUTO
```

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.
////////////////////////////////////

023136
023136

BGNCLN

L\$CLEAN::

023136
023136
023136

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 023140      BGNDU
9 023140      ;ISSUE UNIBUS RESET TO CLEAN UP
10 023140      BRESET
10 023140      104433
11
12 023142      ;PRINT 'UNIT XX DROPPED'
12 023142      PRINTF  #FMT27,LOGDEV
12 023146      013746  002330
12 023146      012746  023170
12 023152      012746  000002
12 023156      010600
12 023160      104417
12 023162      062706  000006
13 023166      ENDDU
13 023166
13 023166      104453
14
15 023170      045      116      045  FMT27:  .ASCII  /%N%AUNIT %D2%A DROPPED%N/
15 023173      101      125      116
15 023176      111      124      040
15 023201      045      104      062
15 023204      045      101      040
15 023207      104      122      117
15 023212      120      120      105
15 023215      104      045      116
15 023220      000
16
17      .EVEN
18
19
20
21
```

```
LSDU::
TRAP      C$RESET
MOV      LOGDEV,-(SP)
MOV      #FMT27,-(SP)
MOV      #2,-(SP)
MOV      SP,R0
TRAP      C$PRINTF
ADD      #6,SP
L10020:
TRAP      C$DU
```

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.
://

BGNAU

ENDAU

LSAU::

L10021:

TRAP

CSAU

023222
023222
023222
023222
023222 104452

.SBTTL HARDWARE TESTS

```
*****
.SBTTL      TEST 1 - BIT STUFFING TEST
*
* THE DEVICE IS ENABLED FOR TRANSMIT AND RECEIVE, AND A MESSAGE IS
* INITIATED IN BIT MODE . TWO LEADING FLAGS ARE SENT,
* FOLLOWED BY ALL SIXTEEN CHARS IN DATA PATTERN S. THIS PATTERN
* CONSISTS OF CHARACTERS WHICH REQUIRE NO BIT STUFFING AND CHARACTERS
* WHICH REQUIRE BIT STUFFING INDIVIDUALLY AND IN COMBINATION WITH
* ADJACENT CHARACTERS. ALL 16 CHARACTERS ARE READ AND COMPARED
* BY THE RECEIVER.
*   PATTERN S = 000,017,036,074,170,360,037,076,174,370,077,176,374,
*               177,376,377
*****
```

BGNTST

```
*****
T1::
21 023224 012737 023346 002346      MOV    #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
22 023232 004737 005240              JSR    PC,INTRN        ;MST CLR, LOAD 2 SOM'S
23 023236 000000                      000
24 023240 000300                      CRC2!CRC1          ;BIT MODE, NO ERR DETECTION
25 023242 004737 010574              JSR    PC,LDBYTS      ;LOAD PAT S INTO TX SILO
26 023246 002642                      PATS
27 023250 000020                      16.
28 023252 012737 001000 002412      MOV    #TXEOM,TXWORD
29 023260 004737 004674              JSR    PC,LDTXSI      ;LOAD 2 EOM'S INTO TX SILO
30 023264 004737 004674              JSR    PC,LDTXSI
31 023270 004737 004754              JSR    PC,STPLU       ;CLK MORE THAN ENTIRE MSG
32 023274 000300                      192.
33 023276 012701 002642              MOV    #PATS,R1        ;INIT PAT S POINTER
34 023302 112137 023312      6$:     MOVB   (R1)+,8$
35 023306 004737 007314              JSR    PC,CKDATA      ;CHK A RCV'D CHAR
36 023312 000000                      8$:     .WORD   0
37 023314 000000                      0
38 023316 020127 002661              CMP    R1,#PATS+15.    ;SEE IF 15 CHARS CHECKED YET
39 023322 103767                      BLO    6$             ;BR IF NOT YET
40 023324 111137 023342              MOVB   (R1),12$
41 023330 052737 001000 023342      BIS    #RXEBL,12$      ;GET SET TO CHK EBLK = 1
42 023336 004737 007314              JSR    PC,CKDATA      ;CHK LAST CHAR AND EBLK = 1
43 023342 000000                      12$:     .WORD   0
44 023344 000000                      0
45 023346 004737 003276      24$:     JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
46 023352                      ENDTST
023352
023352 104401                      L10022:
TRAP    C$ETST
*****
```

```
*****
.SBTTL      TEST 2 - RCV OVERRUN ERROR SET AND CLEAR TEST
*
*****
```

```

55      ;* IN THIS TEST, A RCV OVERRUN ERROR IS FORCED IN EACH OF 2 SUBTESTS.
56      ;* IN THE FIRST, A MESSAGE IS INITIATED, 64 001 CHARS ARE SENT, AND THE
57      ;* RECEIVER IS NOT SERVICED IN RESPONSE TO THE USYRT RCV FLAG, WHICH CAUSES RCV
58      ;* OVERRUN TO SET. THEN, A CHECK IS MADE TO INSURE THAT OVRR IS NOT
59      ;* CLEARED BY THE LINE UNIT READING THE USYRT STATUS.
60      ;* THEN, IC IS SET TO CLEAR THE ERROR, AND THIS IS VERIFIED.
61      ;*
62      ;* IN THE SECOND SUBTEST, RCV OVRUN IS FORCED AGAIN, AND A MASTER CLEAR
63      ;* IS ISSUED TO CLEAR THE ERROR, AND THIS IS VERIFIED.
64      ;*****
65      BGNTST
66      023354 012737 023722 002346      MOV      #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
67      ;-----T2::
68      ; CAUSE OVRR, SET IC TO CLEAR IT
69      ;-----
70      023362 004737 005240      JSR      PC,INITRN      ;MST CLR, LOAD 2 SOM'S
71      023366 000226      SYNCH
72      023370 000311      CRC2!CRC1!STRIP!DDCMP      ;DDCMP, NO ERR DET
73      023372 004737 010746      JSR      PC,LODSIL      ;LOAD 64 001 CHARS INTO TX SILO
74      023376 000001      001
75      023400 000100      64.
76      023402 004737 007000      JSR      PC,RCV1ST      ;CLOCK UNTIL FIRST DATA CHAR RCV'D
77      023406 000030      24.
78      023410 004737 004754      JSR      PC,STPLU      ;CLOCK UNTIL 59 MORE RCV'D
79      023414 000730      472.
80      023416 004737 010746      JSR      PC,LODSIL      ;LOAD 60 MORE INTO TX SILO
81      023422 000001      001
82      023424 000074      60.
83      023426 004737 004754      JSR      PC,STPLU      ;CLK 8 MORE TIMES TO FORCE UNDERRUN
84      023432 000100      64.
85      023434 012701 000100      MOV      #64.,R1      ;READ AND CHK 64 CHARS FROM RCV SILO
86      023440 004737 007314      6$: JSR      PC,CKDATA
87      023444 000001      001
88      023446 000000      0
89      023450 005301      DEC      R1
90      023452 001372      BNE      6$
91      023454 004737 007314      JSR      PC,CKDATA      ;READ CHAR, CHK OVRR = 1
92      023460 004001      4001
93      023462 000010      8.
94      023464 004737 007314      JSR      PC,CKDATA      ;READ CHAR, CHK OVRR STILL = 1
95      023470 004001      4001
96      023472 000010      8.
97      023474 012737 000012 002364      MOV      #12,REGNUM      ;SET REG NO. = 12
98      023502 012737 000200 002352      MOV      #IC,WRIBYT
99      023510 004737 003450      JSR      PC,WRITLU      ;SET IC TO CLEAR RCVR
100     023514 004737 006720      JSR      PC,RDRXSI      ;READ RCV SILO
101     023520 132737 000010 002415      BITB     #OVRR,RXWORD+1      ;CHK FOR OVRR CLEARED
102     023526 001407      BEQ      8$      ;BR IF OVRR CLEARED
103     023530 004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
104     ;REPORT OVRR NOT CLEARED
105     023534      ERRDF      41,EM41,ERR7
106     023534 104455      TRAP     CSERDF
107     023536 000051      .WORD    41
108     023540 014077      .WORD    EM41
109     023542 020302      .WORD    ERR7
110     023544 000466      BR      24$

```

```

107 023546      8$:
108      :-----
109      : CAUSE OVR, SET MST CLR TO CLEAR IT
110      :-----
111 023546 004737 005240      JSR      PC,INITRN      ;MST CLR, LOAD 2 SOM'S
112 023552 000226      SYNCH
113 023554 000311      CRC2!CRC1!STRIP!DDCMP      ;DDCMP, NO ERR DET
114 023556 004737 010746      JSR      PC,LODSIL      ;LOAD 64 001 CHARS INTO TX SILO
115 023562 000001      001
116 023564 000100      64.
117 023566 004737 007000      JSR      PC,RCV1ST      ;CLOCK UNTIL FIRST DATA CHAR RCV'D
118 023572 000030      24.
119 023574 004737 004754      JSR      PC,STPLU      ;CLOCK UNTIL 59 MORE RCV'D
120 023600 000730      472.
121 023602 004737 010746      JSR      PC,LODSIL      ;LOAD 60 MORE INTO TX SILO
122 023606 000001      001
123 023610 000074      60.
124 023612 004737 004754      JSR      PC,STPLU      ;CLK 8 MORE TIMES TO FORCE UNDERRUN
125 023616 000100      64.
126 023620 012701 000100      MOV      #64.,R1      ;READ AND CHK 64 CHARS FROM RCV SILO
127 023624 004737 007314      9$: JSR      PC,CKDATA
128 023630 000001      001
129 023632 000000      0
130 023634 000301      DEC      R1
131 023636 001372      BNE      9$
132 023640 004737 007314      JSR      PC,CKDATA      ;READ CHAR, CHK OVR = 1
133 023644 004001      4001
134 023646 000010      8.
135 023650 004737 007314      JSR      PC,CKDATA      ;READ CHAR, CHK OVR STILL = 1
136 023654 004001      4001
137 023656 000010      8.
138 023660 012737 000012 002364      MOV      #12,REGNUM      ;SET REG NO. = 12
139 023666 004737 003276      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
140 023672 004737 006720      JSR      PC,RDRXSI      ;READ RCV SILO
141 023676 132737 000010 002415      BITB      #OVR, RXWORD+1      ;CHK FOR OVR CLEAR
142 023704 001406      BEQ      24$      ;BR IF OVR CLEARED
143 023706 004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
144      ;REPORT OVR NOT CLEARED
145      ERRDF 41,EM41,ERR7
146 023712      023712 104455
147 023714 000051
148 023716 014077
149 023720 020302
150 023722 004737 003276      24$: JSR      PC,MSTCLR      ;ISSUE CLEAN UP MST CLR
151 023726      023726
152 023726 104401
153      L10023:
154      TRAP      C$SETST
155
156      ;*****
157      ;SBTTL      TEST 3 - ABORT SEQUENCE TEST
158      ;*
159      ;* SET BIT MODE, CRC, AND ENABLE THE DEVICE FOR
160      ;* TRANSMIT AND RECEIVE. SEND 2 FLAGS AND 4 DATA CHARS (001).

```

158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

.* AS THE FIRST DATA CHAR IS BEING TRANSMITTED,
.* SET THE ABORT BIT (REG 11).
.* ON THE RECEIVER SIDE, CHECK FOR RECEPTION OF THE FIRST DATA CHAR
.* AND THEN THE SETTING OF RAB AND REOM A CHAR TIME LATER.
.* ALSO, CHECK FOR IACT = 0. THEN, CHECK THAT RAB
.* IS CLEARED BY READING THE USYRT STATUS, TRANSMITTING A NEW MSG,
.* RECEIVING THE FIRST CHAR (003) AND CHECKING FOR RAB CLEARED.
.*
.* REPEAT THE ABOVE SEQUENCE, SET IC, AND CHECK THAT
.* THIS CLEARS RAB.
.*
.* REPEAT THE ABOVE SEQUENCE, ISSUE MASTER CLEAR, CHECK THAT THIS
.* CLEARS RAB.
.*

BGNTST173 023730
023730

174 023730 012737 024260 002346

MOV #24\$,RETADR ;SET TEST EXIT ADRS FOR ERRORS

T3::

175
176
177

; CAUSE ABORT, START NEW MSG TO CLEAR IT

178 023736 004737 005240

JSR PC,INITRN ;MST CLR, LOAD 2 SOM S

179 023742 000000

000

180 023744 000000

000

;BIT MODE, CRC

181 023746 004737 010516

JSR PC,LODMSG

;LOAD MSG INTO TX SILO

182 023752 002732

MSG3

183 023754 000014

12.

184 023756 004737 007000

JSR PC,RCV1ST

;CLK AND RCV FIRST DATA CHAR

185 023762 000060

48.

186 023764 004737 007314

JSR PC,CKDATA

;CHK CHR = 001, CLK ABORT CHAR

187 023770 000001

001

188 023772 000010

8.

189 023774 004737 007314

JSR PC,CKDATA

;CHK FOR RAB, EBLK, AND 001 CHAR

190 024000 003001

RXABT.RXEBL.001

191 024002 000000

0

192 024004 004737 006260

JSR PC,IACTIV

;CHK FOR IACT = 0

193 024010 000000

0

194 024012 004737 007000

JSR PC,RCV1ST

;CLK AND RCV NEW MSG

195 024016 000060

48.

196 024020 004737 007314

JSR PC,CKDATA

;CHK CHAR - 003

197 024024 000003

003

198 024026 000000

0

199

; CAUSE ABORT, SET IC TO CLEAR IT

200

201

202 024030 004737 005240

JSR PC,INITRN

;MST CLR, LOAD 2 SOM'S

203 024034 000000

000

204 024036 000000

000

;BIT MODE, CRC

205 024040 004737 010516

JSR PC,LODMSG

;LOAD MSG INTO TX SILO

206 024044 002732

MSG3

207 024046 000014

12.

208 024050 004737 007000

JSR PC,RCV1ST

;CLK AND RCV FIRST DATA CHAR

209 024054 000060

48.

210 024056 004737 007314

JSR PC,CKDATA

;CHK CHR = 001, CLK ABORT CHAR

211 024062 000001

001

212 024064 000010

8.

213 024066 004737 007314

JSR PC,CKDATA

;CHK FOR RAB, EBLK, AND 001 CHAR

```
214 024072 003001 RXABT!RXEBL!001
215 024074 000000 0
216 024076 012737 000012 002364 MOV #12,REGNUM ;SET REG NO. = 12
217 024104 012737 000200 002352 MOV #IC,WRIBYT
218 024112 004737 003450 JSR PC,WRITLU ;SET IC TO CLEAR RCVR
219 024116 004737 006720 JSR PC,RDRXSI ;READ RCV SILO
220 024122 132737 000004 002415 BITB #RAB,RXWORD+1 ;CHK FOR RAB CLEARED
221 024130 001407 BEQ 8$ ;BR IF RAB CLEARED
222 024132 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
223 ;REPORT RAB NOT CLEARED
224 ERRDF 39,EM39,ERR7
225 024136 104455 TRAP C$ERDF
226 024136 000047 .WORD 39
227 024140 014043 .WORD EM39
228 024142 020302 .WORD ERR7
229 024146 000444
230 024150 004737 005240 BR 24$
231 024154 000000 8$:
232 024156 000000 ;-----
233 024160 004737 010516 ;CAUSE ABORT, ISSUE MASTER CLEAR TO CLEAR IT
234 024164 002732 ;-----
235 024166 000014 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
236 024170 004737 007000 000
237 024174 000060 000 ;BIT MODE, CRC
238 024176 004737 007314 JSR PC,LODMSG ;LOAD MSG INTO TX SILO
239 024202 000001 MSG3
240 024204 000010 12.
241 024206 004737 007314 JSR PC,RCV1ST ;CLK AND RCV FIRST DATA CHAR
242 024212 003001 48.
243 024214 000000 JSR PC,CKDATA ;CHK CHR = 001, CLK ABORT CHAR
244 024216 012737 000012 002364 RXABT!RXEBL!001
245 024224 004737 003276 0 MOV #12,REGNUM ;SET REG NO. = 12
246 024230 004737 006720 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
247 024234 132737 000004 002415 JSR PC,RDRXSI ;READ RCV SILO
248 024242 001406 BITB #RAB,RXWORD+1 ;CLK FOR RAB CLEARED
249 024244 004737 004226 BEQ 24$ ;BR IF RAB CLEARED
250 ;REPORT RAB NOT CLEARED
251 ERRDF 39,EM39,ERR7 ;GET REGS FOR PRINTOUT
252 024250 104455 TRAP C$ERDF
253 024252 000047 .WORD 39
254 024254 014043 .WORD EM39
255 024256 020302 .WORD ERR7
256 024260 004737 003276 24$: JSR PC,MSTCLR ;ISSUE MST CLR TO CLEAN UP
257 024264 ENDTST
258 024264 104401 L10024: TRAP C$ETST
259
260 ;*****
.SBTTL TEST 4 - ABORT AND IDLE FLAGS TEST
```

```

261
262
263
264
265
266
267 024266
    024266
268 024266 012737 024342 002346      MOV    #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
269 024274 004737 005240              JSR    PC,INITRN      ;MST CLR, LOAD 2 SOM'S
270 024300 000000                      000
271 024302 000040                      IDLE
272 024304 004737 010516              JSR    PC,LODMSG      ;BIT MODE, NO ERROR DET
273 024310 002732                      MSG3
274 024312 000005                      5
275 024314 004737 007000              JSR    PC,RCV1ST      ;CLK AND RCV FIRST DATA CHAR
276 024320 000060                      48.
277 024322 004737 007314              JSR    PC,CKDATA      ;CHK CHR = 001, CLK FLAG CHAR
278 024326 000001                      001
279 024330 000010                      8.
280 024332 004737 007314              JSR    PC,CKDATA      ;CHK RAB = 0, EBLK = 1
281 024336 001001                      RXEBL!001
282 024340 000000                      0
283 024342 004737 003276      24$: JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR
284 024346
    024346
    024346 104401
                                L10025:
                                IRAP    C$ETST

285
286
287
288
289
290
291
292
293
294
295
296
297
298
299 024350
    024350
300 024350 012737 024432 002346      MOV    #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
301 024356 004737 005240              JSR    PC,INITRN      ;MST CLR, LOAD 2 SOM'S
302 024362 000000                      000
303 024364 000000                      000
304 024366 012737 000100 002416      MOV    #TXEN,DISILO    ;SET TX ENB TO KEEP RTS HIGH
305 024374 004737 010746              JSR    PC,LODSIL      ;LOAD 4 001 CHARS INTO TX SILO
306 024400 000001                      001
307 024402 000004                      4
308 024404 004737 007000              JSR    PC,RCV1ST      ;CLK AND RCV FIRST CHAR
309 024410 000060                      48.
310 024412 004737 007314              JSR    PC,CKDATA      ;CHK DATA - 001, CLOCK ABORT CHAR
311 024416 000001                      001
312 024420 000011                      9.
313 024422 004737 007314              JSR    PC,CKDATA      ;CHK FOR RAB, EBLK, AND 001 CHAR

;*****
;* TRANSMIT THE SAME ABORT SEQUENCE AS IN THE PREVIOUS TEST, BUT
;* WITH THE IDLE BIT SET. CHECK THAT FLAGS ARE SENT AND RECEIVED
;* (NOT ABORT CHARACTERS) BY VERIFYING THAT RAB DOES
;* NOT SET, AND THAT THE MESSAGE TERMINATES WITH EBLK = 1.
;*****
BGNTST
T4::
;*****
SBTTL      TEST 5 - TRANSMITTER UNDERRUN ERROR, IDLE ABORT CHARS, BIT MODE
;*****
;* A MESSAGE IS INITIATED IN BIT MODE, 4 001 CHARS ARE SENT, AND THE TRANSMITTER
;* IS NOT SERVICED IN RESPONSE TO THE LAST TX FLAG, WHICH CAUSES TX
;* UNDERRUN ERROR TO SET. ON THE RECEIVER SIDE, CHECK THAT THE DATA
;* CHAR IS RECEIVED, AND THAT 8 CYCLES LATER THE RAB BIT SETS, AND
;* THE DEVICE IDLES ABORT CHARACTERS.
;*****
BGNTST
T5::

```

314 024426 003001 RXABT!RXEBL!001
 315 024430 000000 0
 316 024432 004737 003276 24\$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR
 317 024436 004737 003276 24\$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR
 024436 104401 L10026: TRAP C\$ETST
 024436 104401

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

024440

024440

333 024440 012737 024626 002346

334 024446 004737 005240

335 024452 000000

336 024454 000000

337 024456 004737 010746

338 024462 000252

339 024464 000005

340 024466 004737 007000

341 024472 000060

342 024474 004737 004754

343 024500 000004

344 024502 012737 000012 002364

345 024510 012737 000200 002352

346 024516 004737 003450

347 024522 004737 006260

348 024526 000000

349 024530 004737 005774

350 024534 000001

351 024536 005037 002370

352 024542 005037 002372

353 024546 004737 006720

354 024552 105737 002414

355 024556 001404

356 024560 012737 000010 002364

357 024566 000406

358 024570 105737 002415

359 024574 001414

360 024576 012737 000012 002364

361 024604 113737 002414 002372

362 024612 004737 004226

363

364 024616

024616 104455

024620 000056

024622 014253

 SBTTL TEST 6 - RECEIVER DISABLE TEST
 *
 * TRANSMIT AND RECEIVE ARE ENABLED IN BIT MODE, AND 2 FLAGS
 * ARE SENT, FOLLOWED BY 5 252 DATA CHARS. AFTER THE SECOND DATA CHAR HAS BEGUN
 * TO BE RECEIVED, IC IS SET.
 * THEN, THE PROGRAM CHECKS THAT A USYRT RCV FLAG IS NOT GENERATED, AND
 * THE RECEIVER DATA PATH STOPS OPERATING IN THE MIDDLE OF THE CHAR.

 BGNTST

T6::
 MOV #24\$,RETADR ;SET TEST EXIT ADRS FOR ERRORS
 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
 000
 000 ;BIT MODE, CRC
 JSR PC,LODSIL ;LOAD 5 252 CHARS
 252
 5
 JSR PC,RCV1ST ;CLK AND RCV FIRST DATA CHAR
 48.
 JSR PC,STPLU ;CLK TO MIDDLE OF 2ND CHAR
 4
 MOV #12,REGNUM ;SET REG NO. = 12
 MOV #IC,WRIBYT
 JSR PC,WRITLU ;SET IC IN REG 12
 JSR PC,IACTIV ;CHK IACT = 0
 0
 JSR PC,ISIRDY ;CHK ICIR = 1, IRDY = 0
 1
 CLR GOODAT ;SET EXPECTED DATA = 0
 CLR BADDAT
 JSR PC,RDRXSI ;READ RCV SILO
 TSTB RXWORD ;SEE IF SILO BITS 0-7 = 000
 BEQ 9\$;BR IF YES
 MOV #10,REGNUM ;SET REG NO. = 10
 BR 12\$
 TSTB RXWORD+1 ;SEE IF SILO BITS 8-11 = 000
 BEQ 24\$;BR IF YES
 MOV #12,REGNUM ;SET REG NO. = 12
 MOVB RXWORD,BADDAT ;GET ACTUAL DATA
 JSR PC,GETALL ;GET REGS FOR PRINTOUT
 ;REPORT RCV SILO NOT CLEARED BY IC
 ERRDF 46,EM46,ERR2

TRAP C\$ERDF
 .WORD 46
 .WORD EM46

```
024624 015444
365 024626 004737 003276      24$: JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP .WORD ERR2
366 024632                                L10027:
    024632                                TRAP    C$ETST
    024632 104401

367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384 024634
    024634
385 024634 012737 025372 002346      MOV    #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
386 024642 004737 005240              JSR    PC,INITRN      ;FIND OUT WHICH USYRT CHIP
387 024646 000000                      0
388 024650 000000                      0
389 024652 012701 0001C0              MOV    #TXLEN1,R1      ;SET INITIAL TX LENGTH TO 2 BITS
390 024656 004737 003276      6$: JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR
391 024662 004737 010324              JSR    PC,SETUP      ;PROGRAM THE USYRT
392 024666 000000                      000
393 024670 000300                      CRC2!CRC1
394 024672 000000                      000
395 024674 000000                      000
396 024676 012737 000014 002364      MOV    #14,REGNUM      ;SET REG NO. = 14
397 024704 012737 000140 002352      MOV    #TXEN!DISSI,WRIBYT
398 024712 004737 003450              JSR    PC,WRITLU      ;SET TXEN AND DISSI IN REG 14
399 024716 012737 000140 002416      MOV    #TXEN!DISSI,DISILO ;SET DISABLE SILO FLAG
400 024724 012737 000012 002364      MOV    #12,REGNUM      ;SET LU REG NO. = 12
401 024732 112737 000040 002352      MOVB   #LULP,WRIBYT
402 024740 004737 003450              JSR    PC,WRITLU      ;SET LULP IN REG 12
403 024744 012737 000002 002366      MOV    #2,AXNUM      ;SET AX BYTE NO. = 2
404 024752 105037 002360              CLRB   WAX15
405 024756 112737 000001 002362      MOVB   #TSOM,WAX16
406 024764 004737 004012              JSR    PC,WRITAX      ;LOAD SOM CHAR
407 024770 005004                      CLR    R4          ;INIT COUNTER
408 024772 012737 000011 002364      MOV    #11,REGNUM      ;SET REG NO. = 11
409 025000 004737 004754      7$: JSR    PC,STPLU      ;CLOCK LU FOR A CYCLE
410 025004 000001                      1
411 025006 004737 003372              JSR    PC,READLU      ;READ REG 11
412 025012 132737 000100 002350      BITB   #OACT,REDBYT      ;SEE IF OACT SET YET
413 025020 001014                      BNE    10$          ;BR IF OACT SET
414 025022 005204                      INC    R4          ;INCR COUNTER
415 025024 020427 000004              CMP    R4,#4          ;SEE IF COUNT TOO BIG
416 025030 002763                      BLT    7$          ;BR IF NOT
417 025032 004737 004226              JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
```

```

418                                     ;REPORT OACT NOT SET
419 025036                             ERRDF 11,EM11,ERR7
      025036 104455
      025040 000013
      025042 013437
      025044 020302
420 025046 000137 025372
421 025052 004737 004012
422 025056 004737 004754
423 025062 000010
424 025064 105037 002362
425 025070 004737 004012
426 025074 004737 004754
427 025100 000010
428 025102 004737 004012
429 025106 004737 004754
430 025112 000010
431 025114 012737 000006 002366
432 025122 010137 002362
433 025126 004737 004012
434 025132 012737 000002 002366
435 025140 105037 002362
436 025144 005737 002420
437 025150 001403
438 025152 112737 000002 002362
439 025160 004737 004012
440 025164 004737 004754
441 025170 000010
442 025172 005737 002420
443 025176 001005
444 025200 112737 000002 002362
445 025206 004737 004012
446 025212 012737 000001 002366
447 025220 005003
448 025222 004737 003624
449 025226 132737 000002 002356
450 025234 001016
451 025236 004737 004754
452 025242 000001
453 025244 005203
454 025246 020327 000023
455 025252 002763
456 025254 004737 004226
457                                     ;REPORT REOM NOT SET
458                                     ERRDF 31,EM31,ERR10
      025260 104455
      025262 000037
      025264 013706
      025266 021442
459 025270 000440
460 025272 013702 002356
461 025276 042702 000217
462 025302 006102
463 025304 120201
464 025306 001421
465 025310 010137 002370
466 025314 006237 002370

      10$: JMP 24$
      JSR PC,WRITAX ;LOAD ANOTHER SOM CHAR
      JSR PC,STPLU ;CLK FIRST FLAG
      8.
      CLRB WAX16
      JSR PC,WRITAX ;LOAD FIRST 000 CHAR
      JSR PC,STPLU ;CLK SECOND FLAG
      8.
      JSR PC,WRITAX ;LOAD SECOND 000 CHAR
      JSR PC,STPLU ;CLK FIRST 000 CHAR
      8.
      MOV #6,AXNUM ;SET AX BYTE NO. FOR AX 3
      MOV R1,WAX16 ;GET TX CHAR LENGTH
      JSR PC,WRITAX ;SET TX CHAR LENGTH
      MOV #2,AXNUM ;SET AX BYTE NO. = 2
      CLRB WAX16
      TST CHPTYP ;SEE IF SIG USYRT
      BEQ 5$ ;BR IF YES
      MOVB #TEOM,WAX16 ;SET TEOM WITH LAST DATA CHAR
      JSR PC,WRITAX ;LOAD 3RD 000 CHAR
      JSR PC,STPLU ;CLK 2ND 000 CHAR
      8.
      TST CHPTYP ;SEE IF SIG USYRT
      BNE 16$ ;BR IF NOT
      MOVB #TEOM,WAX16
      JSR PC,WRITAX ;LOAD AN EOM CHAR
      MOV #1,AXNUM ;SET AX BYTE NO. = 1
      CLR R3
      12$: JSR PC,READAX ;READ AX0
      BITB #REOM,RAX16 ;CHK FOR REOM = 1
      BNE 14$ ;BR IF YES
      JSR PC,STPLU ;CLOCK LU FOR A CYCLE
      1
      INC R3 ;INCR COUNT
      CMP R3,#19. ;SEE IF COUNT TOO BIG
      BLT 12$ ;BR IF NOT
      JSR PC,GETALL ;GET REGS FOR PRINTOUT

      14$: BR 24$
      MOV RAX16,R2 ;GET AX0-16 CONTENTS
      BIC #217,R2 ;MASK OFF ALL BUT ASSEMB BIT COUNT
      ROL R2
      CMPB R2,R1 ;CHK FOR CORRECT ASSEMB BIT COUNT
      BEQ 9$ ;BR IF MATCH
      MOV R1,GOODAT ;SET EXPECTED DATA
      ASR GOODAT
  
```

TRAP C\$ERDF
 .WORD 11
 .WORD EM11
 .WORD ERR7

TRAP C\$ERDF
 .WORD 31
 .WORD EM31
 .WORD ERR10

```

467 025320 152737 000002 002370      BISB    #REOM,GOODAT
468 025326 013737 002356 002372      MOV     RAX16,BADDAT      ;SET ACTUAL DATA
469 025334 004737 004226      JSR     PC,GETALL       ;GET REGS FOR PRINTOUT
470                                     ;REPORT ASSEMB BIT COUNT INCORRECT
471                                     ERRDF  47,EM47,ERR3
      025340 104455
      025342 000057
      025344 014306
      025346 015752
472 025350 000410
473 025352 005701
474 025354 001406
475 025356 062701 000040
476 025362 042701 000400
477 025366 000137 024656
478 025372 005037 002416
479 025376 004737 003276
480 025402
      025402 104401
      025402 104401

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 025404
      025404
507
508
509
510 025404
      025404
      025404 104402
511 025406 012737 025504 002346      MOV     #3$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
512 025414 004737 005240      JSR     PC,INITRN      ;MST CLR, LOAD 2 SOM'S
513 025420 000000
514 025422 000020      000      ;SEC ADRS = 000
                        SECA    ;BIT MODE, CRC, SEC ADRS MODE

      BR     24$
9$:  TST     R1           ;SEE IF ALL DONE YET
      BEQ    24$         ;BR IF YES
      ADD    #TXLEN0,R1  ;INCR TX LENGTH
      BIC    #400,R1     ;MASK OFF OVERFLOW IF 8 BITS
      JMP    6$          ;PROCEED
24$: CLR     DISILO
      JSR    PC,MSTCLR   ;ISSUE MASTER CLR TO CLEAN UP
ENDTST

                                L10030:
                                TRAP   C$SETST

;*****
;SBTTL      TEST 8 - SECONDARY STATION ADDRESS BIT TEST
;
;* FIRST, A MASTER CLEAR IS ISSUED. THEN, THE LINE UNIT IS PLACED IN
;* BIT MODE, AND THE SECA BIT (REG 17) IS SET.
;* 2 FLAGS ARE SENT, FOLLOWED BY 252, 000, AND A TERMINATING FLAG.
;* THEN, THE RECEIVER IS CHECKED TO MAKE SURE THAT NO DATA CHARS ARE
;* RECEIVED.
;*
;* NEXT, THE SECONDARY STATION ADDRESS BITS IN AX2-15 ARE LOADED
;* WITH THE FIRST WORD OF DATA PATTERN T. 2 FLAGS ARE SENT,
;* FOLLOWED BY THE FIRST WORD OF DATA PATTERN T, A 000 CHAR,
;* AND A TERMINATING FLAG.
;* THEN, THE RCV'D DATA IS CHECKED TO MAKE SURE THAT THE SEC STATION
;* ADDRESS IS RCV'D AS THE FIRST DATA CHAR, FOLLOWED BY 000.
;*
;* THEN, THE SUBTEST IS REPEATED FOR EACH OF THE REMAINING WORDS OF
;* DATA PATTERN T.
;* PATIERN T = 000,125,252,176,177
;*****
BGNTST
                                T8::
;-----
; SEND MSG WITH INVALID SEC STA ADRS
;-----
                                T8.1:
                                TRAP   C$BSUB
                                BGNSUB
MOV     #3$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
JSR     PC,INITRN      ;MST CLR, LOAD 2 SOM'S
000
SECA    ;SEC ADRS = 000
;BIT MODE, CRC, SEC ADRS MODE

```

```

515 025424 004737 010746 JSR PC,LODSIL ;LOAD 252 INTO TX SILO
516 025430 000252 252
517 025432 000001 1
518 025434 004737 010746 JSR PC,LODSIL ;LOAD 000 DATA INTO TX SILO
519 025440 000000 000
520 025442 000001 1
521 025444 004737 010746 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
522 025450 001000 TXEOM
523 025452 000002 2
524 025454 004737 004754 JSR PC,STPLU ;TRANSMIT THE MSG
525 025460 000060 48.
526 025462 004737 006260 JSR PC,IACTIV ;CHK IACT = 0
527 025466 000000 0
528 025470 004737 004754 JSR PC,STPLU ;CLOCK 8 MORE CYCLES
529 025474 000010 8.
530 025476 004737 006260 JSR PC,IACTIV ;CHK IACT = 0
531 025502 000000 0
532 025504 3$:
533 025504 ENDSUB
    025504 104403 L10032:
    025504 TRAP C$ESUB
534 -----
535 : SEND MSG'S WITH VALID SEC ADRS'S FROM PAT T
536 :-----
537 025506 012701 002662 MOV #PATT,R1 ;INIT DATA PATTERN POINTER
538 025512 A11:
539 025512 BGNSUB
    025512 T8.2:
    025512 104402 TRAP C$BSUB
540 025514 012737 025624 002346 MOV #24$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
541 025522 111137 025542 MOV (R1),5$ ;SET SEC ADRS
542 025526 111137 025552 MOV (R1),6$ ;SET FIRST DATA CHAR
543 025532 111137 025610 MOV (R1),9$ ;SET EXPECTED DATA CHAR
544 025536 004737 005240 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
545 025542 000000 5$:
546 025544 000020 .WORD 0
547 025546 004737 010746 SECA ;BIT MODE, CRC, SEC ADRS MODE
548 025552 000000 6$:
549 025554 000001 .WORD 0
550 025556 004737 010746 JSR PC,LODSIL ;LOAD A 000 CHAR INTO TX SILO
551 025562 000000 000
552 025564 000001 1
553 025566 004737 010746 JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
554 025572 001000 TXEOM
555 025574 000002 2
556 025576 004737 007000 JSR PC,RCV1ST ;CLOCK AND RCV FIRST DATA CHAR
557 025602 000060 48.
558 025604 004737 007314 JSR PC,CKDATA ;CHK FOR CORRECT RCV'D SEC STA ADRS
559 025610 000000 9$:
560 025612 000011 9.
561 025614 004737 007314 JSR PC,CKDATA ;READ AND CHK 000 CHAR, EBLK=1, BCC=0
562 025620 101000 CRCCCHK'RXEBL:000
563 025622 000000 0
564 025624 24$:
565 025624 ENDSUB
    025624 L10033:
    025624 104403 TRAP C$ESUB
  
```

```

566 025626 005201          INC      R1          ; INCR PATTERN POINTER
567 025630 020127 002667    CMP      R1, #ENDPAT ; SEE IF ALL DONE YET
568 025634 103726          BLO      A11        ; BR IF NO
569 025636 004737 003276    JSR      PC, MSTCLR ; ISSUE MASTER CLEAR
570 025642          ENDTST
      025642
      025642 104401          L10031: TRAP      C$ETST

571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589 025644
      025644
590
591
592
593 025644
      025644
      025644 104402          T9.:
594 025646 012737 025744 002346    MOV     #3$, RETADR ; SET SUBTEST EXIT ADRS FOR ERRORS
595 025654 004737 005240          JSR     PC, INITRN ; MST CLR, LOAD 2 SOM'S
596 025660 000000          000          ; SEC ADRS = 000
597 025662 000020          SECA          ; BIT MODE, CRC, SEC ADRS MODE
598 025664 004737 010746          JSR     PC, LODSIL ; LOAD 377 INTO TX SILO
599 025670 000377          377
600 025672 000001          1
601 025674 004737 010746          JSR     PC, LODSIL ; LOAD 125 DATA INTO TX SILO
602 025700 000125          125
603 025702 000001          1
604 025704 004737 010746          JSR     PC, LODSIL ; LOAD 2 EOM'S INTO TX SILO
605 025710 001000          TXEOM
606 025712 000002          2
607 025714 004737 004754          JSR     PC, STPLU ; TRANSMIT THE MSG
608 025720 000060          48.
609 025722 004737 006260          JSR     PC, IACTIV ; CHK IACT = 0
610 025726 000000          0
611 025730 004737 004754          JSR     PC, STPLU ; CLOCK 8 MORE CYCLES
612 025734 000010          8.
613 025736 004737 006260          JSR     PC, IACTIV ; CHK IACT = 0
614 025742 000000          0
615 025744
616 025744
      3$:
      ENDSUB
      L10035:

```

```

025744 104403                                TRAP  C$ESUB
617
618
619
620 025746                                BGNSUB
    025746                                T9.2:
    025746 104402                                TRAP  C$BSUB
621 025750 012737 026044 002346      MOV    #24$,RETADR      ;SET SUBTEST EXIT ADRS FOR ERRORS
622 025756 004737 005240      JSR    PC,INITRN      ;MST CLR, LOAD 2 SOM'S
623 025762 000000      000      ;SEC ADRS = 000
624 025764 000024      SECA!RDALL      ;BIT MODE, CRC, SEC ADRS MODE, RDALL
625 025766 004737 010746      JSR    PC,LODSIL      ;LOAD 1ST DATA CHAR INTO TX SILO
626 025772 000377      377
627 025774 000001      1
628 025776 004737 010746      JSR    PC,LODSIL      ;LOAD A 125 CHAR INTO TX SILO
629 026002 000125      125
630 026004 000001      1
631 026006 004737 010746      JSR    PC,LODSIL      ;LOAD 2 EOM'S INTO 1X SILO
632 026012 001000      TXEOM
633 026014 000002      2
634 026016 004737 007000      JSR    PC,RCV1ST      ;CLOCK AND RCV FIRST DATA CHAR
635 026022 000060      48.
636 026024 004737 007314      JSR    PC,CKDATA      ;CHK FOR 377 CHAR RCV'D
637 026030 000377      377
638 026032 000010      8.
639 026034 004737 007314      JSR    PC,CKDATA      ;READ AND CHK 125 CHAR, EBLK=1, BCC=0
640 026040 101125      CRCCHK!RXEBL!125
641 026042 000000      0
642 026044
643 026044      24$:
    026044      ENDSUB
    026044                                L10036:
    026044 104403                                TRAP  C$ESUB
644 026046      ENDTST                                L10034:
    026046                                TRAP  C$ETST
    026046 104401
  
```

645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662

```

*****
SBTTL      TEST 10 - INSERT ERROR (IERR) BIT TEST - CHAR MODE, NO CRC
*
* THE LINE UNIT IS PLACED IN DDMP MODE WITH NO ERROR DETECTION, AND 2
* SYNCHS, A 000 CHAR, A 377 CHAR, AND 2 SYNCHS ARE LOADED INTO THE
* TRANSMITTER SILO. THEN, THE LU IS CLOCKED UNTIL THE 2ND BIT OF THE 000
* CHAR IS ABOUT TO BE SENT AND THE IERR BIT IS SET FOR A CLOCK TIME AND
* THEN CLEARED. IN THE SAME WAY, IERR IS SET PRIOR TO THE SENDING OF THE 4TH
* AND 5TH BITS OF THE 000 CHAR. IT IS ALSO SET FOR THE SENDING OF THE FIRST
* 4 BITS OF THE 377 CHAR. THE PROGRAM READS THE FIRST RCV'D CHAR FROM AX0
* AND CHECKS IT TO BE 032, AND READS THE 2ND CHAR AND CHECKS IT TO BE 377.
* THEN, A MASTER CLEAR IS DONE TO IDLE THE DEVICE.
*****
BGNTEST
  
```

```

663 026050                                T10.:
    026050
664 026050 012737 026176 002346      MOV    #15$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
665 026056 004737 005240      JSR    PC,INITRN      ;LOAD 2 SOM'S, CLOCK THEM INTO USYRT
  
```

666	026062	000226		SYNCH	
667	026064	000011		STRIP!DDCMP	
668	026066	004737	010516	JSR PC,LODMSG	;LOAD MSG INTO TX SILO
669	026072	002722		MSG2+4	
670	026074	000004		4	
671	026076	004737	004754	JSR PC,STPLU	;CLOCK LU UNTIL 2ND BIT OF 000 CHAR
672	026102	100021		CHPCHK!17.	
673	026104	004737	007212	JSR PC,STPERR	;SET IERR 1 CYCLE
674	026110	000011		STRIP!DDCMP	
675	026112	000001		1	
676	026114	004737	004754	JSR PC,STPLU	;CLOCK LU UNTIL 4TH BIT OF 000 CHAR
677	026120	000001		1	
678	026122	004737	007212	JSR PC,STPERR	;SET IERR FOR 2 CYCLES
679	026126	000011		STRIP!DDCMP	
680	026130	000002		2	
681	026132	004737	004754	JSR PC,STPLU	;CLOCK LU UNTIL 1ST BIT OF 377 CHAR
682	026136	000003		3	
683	026140	004737	007212	JSR PC,STPERR	;SET IERR FOR 4 CYCLES
684	026144	000011		STRIP!DDCMP	
685	026146	000004		4	
686	026150	004737	007000	JSR PC,RCV1ST	;CLOCK AND RCV 1ST CHAR
687	026154	000014		12.	
688	026156	004737	007314	JSR PC,CKDATA	;READ AND COMPARE 1ST CHAR TO 032
689	026162	000032		032	
690	026164	000010		8.	
691	026166	004737	007314	JSR PC,CKDATA	;READ AND COMPARE 2ND CHAR TO 377
692	026172	000377		377	;377 CHAR
693	026174	000000		0	
694	026176	004737	003276	15\$: JSR PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
695	026202			ENDTST	
	026202				L10037:
	026202	104401			TRAP C\$ETST

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 026204

```

*****
:SBTTL      TEST 11 - SWITCH PACK PRINTOUT AND TEST
:
: * - READ AND PRINT SWITCHES IN REG 11 (E121 SW10,9 , E134 SW9,10) :
: * THE PROGRAM READS REG 11 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
: * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
: * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 1,2
: * ARE E121 SW10,9 , AND BITS 3,5 ARE E134 SW9,10.
:
: * - READ AND PRINT SWITCHES IN REG 15 (E134 SW1-8) :
: * THE PROGRAM READS REG 15 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
: * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
: * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E134 SW1-8.
:
: * - READ AND PRINT SWITCHES IN REG 16 (E121 SW1-8) :
: * THE PROGRAM READS REG 16 AND PRINTS THE CONTENTS. IF DESIRED BY THE OPERATOR,
: * (AS INDICATED IN THE SOFTWARE P-TABLE), THE PROGRAM WILL THEN COMPARE IT TO
: * THE EXPECTED VALUE (GIVEN IN THE HARDWARE P-TABLE). BITS 0-7 ARE E121 SW1-8.
:*****
:BGNTST
  
```

```

026204
721
722
723
724 026204
      026204
      026204 104402
725 026206 004737 003276
726 026212 012737 000011 002364
727 026220 004737 003372
728 026224 142737 000321 002350
729 026232 023727 002400 000001
730 026240 001403
731 026242 005737 002260
732 026246 001424
733 026250
734
735 026250
      026250 013746 002436
      026254 012746 013101
      026260 012746 000002
      026264 010600
      026266 104417
      026270 062706 000006
736
737 026274
      026274 013746 002350
      026300 012746 012637
      026304 012746 000002
      026310 010600
      026312 104417
      026314 062706 000006
738 026320 005737 002262
739 026324 001420
740 026326 123737 002350 002454
741 026334 001414
742 026336 013737 002454 002370
743 026344 013737 002350 002372
744 026352 004737 004226
745
746 026356
      026356 104455
      026360 000053
      026362 014135
      026364 015444
747 026366
748 026366
      026366
      026366 104403
749
750
751
752 026370
      026370
      026370 104402
753 026372 004737 003276
754 026376 012737 000015 002364

      -----
      : READ AND PRINT SWITCHES IN REG 11, IF DESIRED
      -----
      BGNSUB
      T11.1: TRAP CSBSUB
      JSR PC,MSTCLR ;ISSUE MASTER CLEAR
      MOV #11,REGNUM ;SET LU REG NO. = 11
      JSR PC,READLU ;READ LU REG 11
      BICB #321,REDBYT ;MASK OFF NON-SWITCH BITS
      CMP FRSPAS,#1 ;SEE IF IN FIRST PASS AFTER LOAD
      BEQ 3$ ;BR IF YES
      TST PRNFLG ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
      BEQ 4$ ;BR IF NOT
3$:
:PRINT DEVICE ADDRESS
  PRINTF #FMT18,MPCSR
      MOV MPCSR,-(SP)
      MOV #FMT18,-(SP)
      MOV #2,-(SP)
      MOV SP,RO
      TRAP CS$PNTF
      ADD #6,SP
:PRINT REG 11 SWITCHES
  PRINTF #FMT12,REDBYT
      MOV REDBYT,-(SP)
      MOV #FMT12,-(SP)
      MOV #2,-(SP)
      MOV SP,RO
      TRAP CS$PNTF
      ADD #6,SP
4$:
  TST SWIFLG ;SEE IF TEST IS ALLOWED
  BEQ 6$ ;BR IF NOT
  CMPB REDBYT,LUSW11 ;COMPARE SWITCHES TO EXPECTED
  BEQ 6$ ;BR IF MATCH
  MOV LUSW11,GOODAT ;SET EXPECTED DATA
  MOV REDBYT,BADDAT ;SET ACTUAL DATA
  JSR PC,GETALL ;GET REGS FOR PRINTOUT
:REPORT REG 11 SWITCHES INCORRECT
  ERDF 43,EM43,ERR2
      TRAP CS$ERDF
      .WORD 43
      .WORD EM43
      .WORD ERR2
6$:
  ENDSUB
      L10041: TRAP CS$ESUB
      -----
      : READ AND PRINT SWITCHES IN REG 15, IF DESIRED
      -----
      BGNSUB
      T11.2: TRAP CSBSUB
      JSR PC,MSTCLR ;ISSUE MASTER CLEAR
      MOV #15,REGNUM ;SET LU REG NO. = 15
  
```

```

755 026404 004737 003372      JSR    PC,READLU      ;READ LU REG 15
756 026410 023727 002400 000001  CMP    FRSPAS,#1      ;SEE IF IN FIRST PASS AFTER LOAD
757 026416 001403              BEQ     3$              ;BR IF YES
758 026420 005737 002260      TST     PRNFLG        ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
759 026424 001412              BEQ     4$              ;BR IF NOT
760 026426                      3$:
761                      ;PRINT REG 15 SWITCHES
762                      PRINTF  #FMT13,REDBYT
                                MOV     REDBYT,-(SP)
                                MOV     #FMT13,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTF
                                ADD     #6,SP
                                026426 013746 002350
                                026432 012746 012722
                                026436 012746 000002
                                026442 010600
                                026444 104417
                                026446 062706 000006
763 026452 005737 002262      4$:  TST     SWIFLG        ;SEE IF TEST IS ALLOWED
764 026456 001420              BEQ     6$              ;BR IF NOT
765 026460 123737 002350 002456  CMPB   REDBYT,LUSW12    ;COMPARE SWITCHES TO EXPECTED
766 026466 001414              BEQ     6$              ;BR IF MATCH
767 026470 013737 002456 002370  MOV     LUSW12,GOODAT    ;SET EXPECTED DATA
768 026476 013737 002350 002372  MOV     REDBYT,BADDAT    ;SET ACTUAL DATA
769 026504 004737 004226      JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
770                      ;REPORT REG 15 SWITCHES INCORRECT
771                      ERDF   44,EM44,ERR2
                                TRAP    C$ERDF
                                .WORD   44
                                .WORD   EM44
                                .WORD   ERR2
                                026510 104455
                                026512 000054
                                026514 014167
                                026516 015444
772 026520                      6$:
773 026520                      ENDSUB
                                L10042:
                                026520 104403              TRAP    C$ESUB
774                      ;-----
775                      ; READ AND PRINT SWITCHES IN REG 16, IF DESIRED
776                      ;-----
777                      BGNSUB
                                T11.3:
                                TRAP    C$BSUB
                                026522
                                026522 104402
778 026524 004737 003276      JSR     PC,MSTCLR      ;ISSUE MASTER CLEAR
779 026530 012737 000016 002364  MOV     #16,REGNUM    ;SET LU REG NO. - 16
780 026536 004737 003372      JSR     PC,READLU      ;READ LU REG 16
781 026542 023727 002400 000001  CMP    FRSPAS,#1      ;SEE IF IN FIRST PASS AFTER LOAD
782 026550 001403              BEQ     3$              ;BR IF YES
783 026552 005737 002260      TST     PRNFLG        ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
784 026556 001412              BEQ     4$              ;BR IF NOT
785 026560                      3$:
786                      ;PRINT REG 16 SWITCHES
787                      PRINTF  #FMT14,REDBYT
                                MOV     REDBYT,-(SP)
                                MOV     #FMT14,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTF
                                ADD     #6,SP
                                026560 013746 002350
                                026564 012746 012766
                                026570 012746 000002
                                026574 010600
                                026576 104417
                                026600 062706 000006
788 026604 005737 002262      4$:  TST     SWIFLG        ;SEE IF TEST IS ALLOWED
789 026610 001420              BEQ     6$              ;BR IF NOT
790 026612 123737 002350 002460  CMPB   REDBYT,LUSW13    ;COMPARE SWITCHES TO EXPECTED
791 026620 001414              BEQ     6$              ;BR IF MATCH

```

```

792 026622 013737 002460 002370      MOV    LUSW13,GOODAT ;SET EXPECTED DATA
793 026630 013737 002350 002372      MOV    REDBYT,BADDAT ;SET ACTUAL DATA
794 026636 004737 004226              JSR    PC,GETALL  ;GET REGS FOR PRINTOUT
795                                ;REPORT REG 16 SWITCHES INCORRECT
796                                ERRDF  45,EM45,ERR2

```

```

026642 104455
026644 000055
026646 014221
026650 015444

```

```

TRAP    C$ERDF
.WORD   45
.WORD   EM45
.WORD   ERR2

```

```

797 026652
798 026652
026652 104403
799 026654
026654
026654 104401

```

```

6$:
ENDSUB

```

```

L10043:
TRAP    C$ESUB
L10040:
TRAP    C$ETST

```

```

ENDTST

```

```

800
801
802
803
804
805
806
807
808
809
810
811

```

```

*****
.SBTTL      TEST 12 - REG AX3-15 PRINTOUT
*
* IN THIS TEST, REG AX3-15 IS READ AND THE CONTENTS PRINTED OUT IF DESIRED BY
* THE OPERATOR, AS INDICATED IN THE SOFTWARE P-TABLE. THE DEFAULT IS TO NOT
* PRINT THE REG.
*****
BGNTST

```

```

812 026656
026656
813 026656 004737 003276
814 026662 142777 000010 153550
815 026670 012737 000006 002366
816 026676 004737 003624
817 026702 023727 002400 000001
818 026710 001403
819 026712 005737 002260
820 026716 001424
821 026720
822
823 026720
026720 013746 002436
026724 012746 013101
026730 012746 000002
026734 010600
026736 104417
026740 062706 000006

```

```

T12::
JSR    PC,MSTCLR ;ISSUE MASTER CLEAR
BICB   #LULoop,@BSEL1 ;CLEAR LULoop
MOV    #6,AXNUM ;SET AX BYTE NO. FOR AX3-15
JSR    PC,READAX ;READ AX3-15,AX3-16
CMP    FRSPAS,#1 ;SEE IF FIRST PASS AFTER LOAD
BEQ    3$ ;BR IF YES
TST    PRNFLG ;SEE IF PRINTOUT IS ALLOWED ON ALL PASSES
BEQ    4$ ;BR IF NOT

```

```

3$:
;PRINT DEVICE ADDRESS
PRINTF #FMT18,MPCSR

```

```

MOV     MPCSR,-(SP)
MOV     #FMT18,-(SP)
MOV     #2,-(SP)
MOV     SP,R0
TRAP    C$PNTF
ADD     #6,SP

```

```

824
825 026744
026744 013746 002354
026750 012746 013032
026754 012746 000002
026760 010600
026762 104417
026764 062706 000006
826 026770
827 026770

```

```

;PRINT AX3-15
PRINTF #FMT15,RAX15

```

```

MOV     RAX15,-(SP)
MOV     #FMT15,-(SP)
MOV     #2,-(SP)
MOV     SP,R0
TRAP    C$PNTF
ADD     #6,SP

```

```

4$:
ENDTST

```

026770
026770 104401L10044:
TRAP C\$ETST828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849850 026772
026772851
852
853854 026772 012737 027322 002346
855 027000 004737 003276
856 027004 004737 010324
857 027010 000226
858 027012 000011
859 027014 000000
860 027016 000000
861 027020 004737 010516
862 027024 002670
863 027026 000011
864 027030 004737 004754
865 027034 000136
866 027036 004737 007314
867 027042 000000
868 027044 000000
869 027046 004737 007314
870 027052 000125
871 027054 000000
872 027056 004737 007314
873 027062 000252
874 027064 000000
875 027066 004737 007314
876 027072 000377
877 027074 000000
878 027076 004737 007314
879 027102 100400
880 027104 000000
881*****
SBTTL TEST 13 - CRC GENERATION TEST* - CRC-16, CHAR MODE:
* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE WITH CRC-16 SELECTED -
* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
* TO CLOCK THE DATA. AT THE END OF THE MESSAGE THE
* PROGRAM CHECKS FOR BCC = 1 (IN REG 12) INDICATING NO ERROR.
*
* - CRC-CCITT - 1'S PRESET:
* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT
* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.
*
* - CRC-CCITT - 0'S PRESET:
* THE ABOVE SUBTEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT
* THE END OF THE MESSAGE THE PROGRAM CHECKS FOR BCC = 0, INDICATING NO ERROR.

BGNST

T13::

CRC 16, CHAR MODE
-----MOV #24\$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
JSR PC,MSTCLR ;ISSUE MASTER CLEAR
JSR PC,SETUP ;PROGRAM THE USVRT
SYNCH
STRIP!DDCMP
000
000
JSR PC,LODMSG ;LOAD MSG INTO TX SILO
MSG1
9.
JSR PC,STPLU ;CLOCK THE MSG
94.
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000
000
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 125
125
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 252
252
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 377
377
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000, CHK BCC - 1
CRCCHK'400
0

```
882
883
884
885 027106 004737 003276
886 027112 004737 010324
887 027116 000000
888 027120 000000
889 027122 000000
890 027124 000000
891 027126 004737 010516
892 027132 002670
893 027134 000011
894 027136 004737 004754
895 027142 000146
896 027144 004737 007314
897 027150 000000
898 027152 000000
899 027154 004737 007314
900 027160 000125
901 027162 000000
902 027164 004737 007314
903 027170 000252
904 027172 000000
905 027174 004737 007314
906 027200 000377
907 027202 000000
908 027204 004737 007314
909 027210 101000
910 027212 000000
911
912
913
914
915
916 027214 004737 003276
917 027220 004737 010324
918 027224 000000
919 027226 000100
920 027230 000000
921 027232 000000
922 027234 004737 010516
923 027240 002670
924 027242 000011
925 027244 004737 004754
926 027250 000146
927 027252 004737 007314
928 027256 000000
929 027260 000000
930 027262 004737 007314
931 027266 000125
932 027270 000000
933 027272 004737 007314
934 027276 000252
935 027300 000000
936 027302 004737 007314
937 027306 000377
938 027310 000000
```

```
-----
: CRC-CCITT-1'S PRESET, BIT MODE
-----
JSR PC,MSTCLR ;ISSUE MASTER CLEAR
JSR PC,SETUP ;PROGRAM THE USYRT
000
000
000
000
JSR PC,LODMSG ;LOAD MSG INTO TX SILO
MSG1
9.
JSR PC,STPLU ;CLOCK THE MSG
102.
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000
000
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 125
125
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 252
252
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 377
377
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
CRCCHK:1000
0

-----
: CRC-CCITT-0'S PRESET, BIT MODE
-----
JSR PC,MSTCLR ;ISSUE MASTER CLEAR
JSR PC,SETUP ;PROGRAM THE USYRT
000
CRC1
000
000
000
JSR PC,LODMSG ;LOAD MSG INTO TX SILO
MSG1
9.
JSR PC,STPLU ;CLOCK THE MSG
102.
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000
000
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 125
125
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 252
252
0
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 377
377
0
```

939 027312 004737 007314
940 027316 101000
941 027320 000000
942
943 027322 004737 003276
944 027326
027326
027326 104401

JSR PC,CKDATA ;READ AND COMPARE CHAR TO 000, CHK BCC = 0
CRCCHK!1000
0

248: JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
ENDTST

L10045:
TRAP C\$ETST

945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968

:SBTTL TEST 14 - CRC ERROR DETECTION TEST
:
* - CRC-16, CHAR MODE :
* THE FOLLOWING MESSAGE IS SENT IN DDCMP MODE, WITH CRC-16 SELECTED -
* 2 SYNCHS, 000, 125, 252, 377, 000, AND 2 SYNCHS, USING LULOOP AND STEPLU
* TO CLOCK THE DATA. JUST BEFORE THE FIRST BIT OF THE LAST 000 CHAR IS SENT,
* THE IERR BIT IS SET IN REG 17 TO CAUSE A 1 TO BE SENT, INTRODUCING A DATA
* ERROR. AT THE END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 0, INDICATING
* AN ERROR.
:
* - CRC-CCITT - 1'S PRESET :
* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-1'S SELECTED. AT THE
* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
:
* - CRC-CCITT - 0'S PRESET :
* THE ABOVE TEST IS PERFORMED IN BIT MODE WITH CRC-CCITT-0'S SELECTED. AT THE
* END OF THE MESSAGE, THE PROGRAM CHECKS FOR BCC = 1, INDICATING AN ERROR.
:*****
BGNTST

969 027330
027330

T14::

970
971
972

: CRC 16, CHAR MODE

973 027330 012737 027732 002346
974 027336 004737 005240
975 027342 000226
976 027344 000011
977 027346 004737 005622
978 027352 000000
979 027354 100010
980 027356 004737 010516
981 027362 002676
982 027364 000006
983 027366 004737 004754
984 027372 000010
985 027374 004737 007212
986 027400 000011
987 027402 000001
988 027404 004737 004754
989 027410 000122
990 027412 004737 007314
991 027416 000001
992 027420 000000

MOV #248,RETADR ;SET TEST EXIT ADRS FOR ERRORS
JSR PC,INITRN ;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
SYNCH
STRIP!DDCMP
JSR PC,TXCHAR ;LOAD 000 CHAR, TX 1ST SYNCH
000
CHPCHK!8.
JSR PC,LODMSG ;LOAD MSG INTO TX SILO
MSG1+6
6
JSR PC,STPLU ;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
8.
JSR PC,STPERR ;MAKE 1ST BIT = 1 INSTEAD OF 0
STRIP!DDCMP
1
JSR PC,STPLU ;CLOCK REST OF MESSAGE
82.
JSR PC,CKDATA ;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
001
0

993	027422	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 125
994	027426	000125		125		
995	027430	000000		0		
996	027432	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 252
997	027436	000252		252		
998	027440	000000		0		
999	027442	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 377
1000	027446	000377		377		
1001	027450	000000		0		
1002	027452	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 000, CHK BCC = 0
1003	027456	100000		CRCCHK.000		
1004	027460	000000		0		
1005						
1006						
1007						
1008						

: CRC-CCITT-1'S PRESET, BIT MODE						

1009	027462	004737	005240	JSR	PC,INITRN	;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
1010	027466	000000		000		
1011	027470	000000		000		
1012	027472	004737	005622	JSR	PC,TXCHAR	;LOAD 000 CHAR, TX 1ST FLAG
1013	027476	000000		000		
1014	027500	100010		CHPCHK!8.		
1015	027502	004737	010516	JSR	PC,LODMSG	;LOAD MSG INTO TX SILO
1016	027506	002676		MSG1+6		
1017	027510	000006		6		
1018	027512	004737	004754	JSR	PC,STPLU	;CLOCK LINE UNIT UNTIL 1ST BIT OF 000 CHAR
1019	027516	000010		8.		
1020	027520	004737	007212	JSR	PC,STPERR	;MAKE 1ST BIT = 1 INSTEAD OF 0
1021	027524	000000		000		
1022	027526	000001		1		
1023	027530	004737	004754	JSR	PC,STPLU	;CLOCK REST OF MESSAGE
1024	027534	000122		82.		
1025	027536	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
1026	027542	000001		001		
1027	027544	000000		0		
1028	027546	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 125
1029	027552	000125		125		
1030	027554	000000		0		
1031	027556	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 252
1032	027562	000252		252		
1033	027564	000000		0		
1034	027566	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 377
1035	027572	000377		377		
1036	027574	000000		0		
1037	027576	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 000, CHK BCC = 1
1038	027602	101400		CRCCHK!1400		
1039	027604	000000		0		
1040						
1041						
1042						
1043						

: CRC-CCITT-0'S PRESET, BIT MODE						

1044	027606	004737	005240	JSR	PC,INITRN	;LOAD 2 SOM'S, CLOCK THEM INTO THE USYRT
1045	027612	000000		000		
1046	027614	000100		CRC1		
1047	027616	004737	005622	JSR	PC,TXCHAR	;LOAD 000 CHAR, TX 1ST FLAG
1048	027622	000000		000		
1049	027624	100010		CHPCHK!8.		

1050	027626	004737	010516	JSR	PC,LODMSG	;LOAD MSG INTO TX SILO
1051	027632	002676		MSG1+6		
1052	027634	000006		6		
1053	027636	004737	004754	JSR	PC,STPLU	;CLOCK LINE JNIT UNTIL 1ST BIT OF 000 CHAR
1054	027642	000010		8.		
1055	027644	004737	007212	JSR	PC,STPERR	;MAKE 1ST BIT = 1 INSTEAD OF 0
1056	027650	000100		CRC1		
1057	027652	000001		1		
1058	027654	004737	004754	JSR	PC,STPLU	;CLOCK REST OF MESSAGE
1059	027660	000122		82.		
1060	027662	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 001 (INTENDED ERROR)
1061	027666	000001		001		
1062	027670	000000		0		
1063	027672	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 125
1064	027676	000125		125		
1065	027700	000000		0		
1066	027702	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 252
1067	027706	000252		252		
1068	027710	000000		0		
1069	027712	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 377
1070	027716	000377		377		
1071	027720	000000		0		
1072	027722	004737	007314	JSR	PC,CKDATA	;READ AND COMPARE CHAR TO 000, CHK BCC - 1
1073	027726	101400		CRCCHK!1400		
1074	027730	000000		0		
1075						
1076	027732	004737	003276	248: JSR	PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP
1077	027736			ENDTST		
	027736					L10046: TRAP CSETST
	027736	104401				

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 027740
 027740
 1103

```

*****
:SBTTL      TEST 15 - VRC PARITY GENERATION TEST
:
: * SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
: * THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC AND 7-BIT CHARS SELECTED.
: * THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
: * OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
: * FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1 AND FOR THE
: * LAST 4 CHARS IT SHOULD = 0.
:
: * SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
: * THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
: * THE DATA CHARS IN PATTERN Q ARE TRANSMITTED, AND AS THE 8TH BIT (PARITY BIT)
: * OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TXDATA FOR THE PROPER STATE.
: * FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0 AND FOR THE
: * LAST 4 CHARS IT SHOULD = 1.
:
: * DATA PATTERN Q = 000,120,125,137,040,052,057,177
:*****
:BGNTST
:
:-----
:
  
```

T15::

```
1104                                     ; TEST ODD VRC GENERATION
1105                                     ;-----
1106 027740 012737 000006 002366      MOV    #6,AXNUM      ;SET AX BYTE NO. FOR AX3
1107 027746 012737 000017 002364      MOV    #17,REGNUM    ;SET REG NO. = 17
1108 027754                                     BGNSUB
```

```

1147 030160
1148 030160
      030160
      030160 104403
1149
1150
1151
1152 030162 012737 000006 002366
1153 030170 012737 000017 002364
1154 030176
      030176
      030176 104402
1155 030200 012737 030402 002346
1156 030206 004737 005240
1157 030212 000026
1158 030214 000211
1159 030216 004737 010574
1160 030222 002613
1161 030224 000010
1162 030226 004737 010746
1163 030232 001000
1164 030234 000002
1165 030236 005037 002360
1166 030242 012737 000347 002362
1167 030250 004737 004012
1168 030254 004737 004754
1169 030260 000010
1170 030262 004737 004754
1171 030266 000010
1172 030270 005001
1173 030272 004737 004754
1174 030276 000010
1175 030300 004737 003372
1176 030304 005201
1177 030306 020127 000004
1178 030312 003014
1179 030314 132737 000040 002350
1180 030322 001424
1181 030324 004737 004226
1182
1183 030330
      030330 104455
      030332 000063
      030334 014467
      030336 020302
1184 030340
      030340 104410
      030342 000040
1185 030344 132737 000040 002350
1186 030352 001010
1187 030354 004737 004226
1188
1189 030360
      030360 104455
      030362 000062
      030364 014433
      030366 020302

      8$:
      ENDSUB

      L10050:
      TRAP C$ESUB

      -----
      TEST EVEN VRC GENERATION
      -----

      MOV #6,AXNUM ;SET AX BYTE NO. FOR AX3
      MOV #17,REGNUM ;SET REG NO. = 17
      BGNSUB

      T15.2:
      TRAP C$BSUB

      MOV #18$,RETADR ;SET SUBTEST EXIT ADRS FOR ERRORS
      JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
      026
      CRC2!STRIP!DDCMP ;CHAR MODE, EVEN VRC
      JSR PC,LDBYTS ;LOAD DATA INTO TX SILO
      PATQ
      8.
      JSR PC,LODSIL ;LOAD 2 EOM'S INTO TX SILO
      TXEOM
      2
      CLR WAX15
      MOV #TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO,WAX16
      JSR PC,WRITAX ;SET TX AND RCV LENGTHS = 7
      JSR PC,STPLU ;CLOCK FIRST SYNCH
      8.
      JSR PC,STPLU ;CLOCK 2ND SYNCH
      8.
      CLR R1 ;INIT CHAR COUNT
      JSR PC,STPLU ;CLOCK A CHAR
      8.
      JSR PC,READLU ;READ REG 17
      INC R1 ;INCR CHAR COUNT
      CMP R1,#4 ;SEE IF 4 CHARS CLKD YET
      BGT 14$ ;BR IF YES
      BITB #TXDATA,REDBYT ;SEE IF PARITY BIT IS CLEARED
      BEQ 16$ ;BR IF YES
      JSR PC,GETALL ;GET REGS FOR PRINTOUT
      ;REPORT EVEN VRC PARITY BIT NOT CLEARED
      ERRDF 51,EM51,ERR7

      TRAP C$ERDF
      .WORD 51
      .WORD EM51
      .WORD ERR7

      ESCAPE SUB

      TRAP C$ESCAPE
      .WORD L10051-.

      14$:
      BITB #TXDATA,REDBYT ;SEE IF PARITY BIT IS SET
      BNE 16$ ;BR IF YES
      JSR PC,GETALL ;GET REGS FOR PRINTOUT
      ;REPORT EVEN VRC PARITY BIT NOT SET
      ERRDF 50,EM50,ERR7

      TRAP C$ERDF
      .WORD 50
      .WORD EM50
      .WORD ERR7
  
```

1190	030370			ESCAPE SUB			
	030370	104410				TRAP	C\$ESCAPE
	030372	000010				.WORD	L10051-
1191	030374	020127	000010	16\$: CMP R1,#8.	;SEE IF ALL CHARS TESTED YET		
1192	030400	002734		BLT 12\$	;BR IF NOT		
1193	030402			18\$: ENDSUB			
1194	030402					L10051:	
	030402	104403				TRAP	C\$ESUB
1195	030404	004737	003276	JSR PC,MSTCLR	;ISSUE MASTER CLEAR TO CLEAN UP		
1196	030410			ENDTST		L10047:	
	030410					TRAP	C\$ETST
	030410	104401					

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

```

*****
SBTTL      TEST 16 - VRC ERROR DETECTION TEST
*
* SUBTEST 1 - FORCING OF BCC USING ODD VRC
* THE LINE UNIT IS PLACED IN CHAR MODE WITH ODD VRC AND 7-BIT CHARS SELECTED.
* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
* RECEIVED (INDICATING AN ERROR).
*
* SUBTEST 2 - FORCING OF BCC USING EVEN VRC
* THE LINE UNIT IS PLACED IN CHAR MODE WITH EVEN VRC AND 7-BIT CHARS SELECTED.
* THE FIRST 8 DATA CHARS IN PATTERN R ARE TRANSMITTED NORMALLY, BUT THE OTHER
* 7 CHARS ARE TRANSMITTED WITH BIT 0 STUCK AT 1 (USING IERR BIT). THE PROGRAM
* CHECKS FOR BCC = 0 AFTER EACH OF THE FIRST 8 CHARS ARE RECEIVED (INDICATING
* NO ERROR) AND CHECKS FOR BCC = 1 AFTER EACH OF THE REMAINING 7 CHARS ARE
* RECEIVED (INDICATING AN ERROR).
*
* DATA PATTERN R = 000,100,120,124,164,172,176,177,000,100,120,124,164,
*                   172,176.
*****
BGN TST
  
```

1224 030412
 030412

T16::

1225
 1226
 1227

 : TEST ODD VRC ERROR DETECTION

1228	030412	012737	000006	002366
1229	030420	012737	000012	002364
1230	030426			
	030426			
	030426	104402		
1231	030430	012737	030646	002346
1232	030436	004737	005240	
1233	030442	000026		
1234	030444	000111		
1235	030446	004737	010746	
1236	030452	000400		
1237	030454	000001		

```

MOV #6,AXNUM      ;SET AX BYTE NO. FOR AX3
MOV #12,REGNUM    ;SET REG NO.
BGNSUB
  
```

T16.1:

TRAP C\$BSUB

```

MOV #10$,RETADR   ;SET SUBTEST EXIT ADRS FOR ERRORS
JSR PC,INITRN     ;MST CLR, LOAD 2 SOM'S
026
CRC1!STRIP!DDCMP  ;CHAR MODE, ODD VRC
JSR PC,LODSIL     ;LOAD A THIRD SOM INTO TX SILO
TXSOM
1
  
```

1238	030456	004737	010574		JSR	PC,LDBYTS	;LOAD DATA INTO TX BUFFER	
1239	030462	002623			PATR			
1240	030464	000017			15.			
1241	030466	004737	010746		JSR	PC,LODSIL	;LOAD 2 EOM'S INTO TX SILO	
1242	030472	001000			TXEOM			
1243	030474	000002			2			
1244	030476	005037	002360		CLR	WAX15		
1245	030502	012737	000347	002362	MOV	#TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO,WAX16		
1246	030510	013737	002362	002424	MOV	WAX16,SAVLEN	;STORE LENGTH 7	
1247	030516	004737	004012		JSR	PC,WRITAX	;SET TX AND RCV LENGTHS = 7	
1248	030522	004737	004754		JSR	PC,STPLU	;CLOCK 1ST 8 CHARS, WITH NO ERRORS	
1249	030526	000130			88.			
1250	030530	012701	000007		MOV	#7,R1	;INIT COUNTER FOR LAST 7 CHARS	
1251	030534	004737	007212		JSR	PC,STPERR	;ASSERT IERR BIT FOR 1 TIME	
1252	030540	000111			CRC1!STRIP!DDCMP			
1253	030542	000001			1			
1254	030544	004737	004754		JSR	PC,STPLU	;CLOCK REST OF CHAR	
1255	030550	000007			7			
1256	030552	005301			DEC	R1	;DECR COUNTER	
1257	030554	001367			BNE	3\$	;BR IF NOT DONE TRANSMITTING YET	
1258	030556	004737	004754		JSR	PC,STPLU	;CLOCK 2 TERMINATING SYNCHS	
1259	030562	000020			16.			
1260	030564	012701	000010		MOV	#8.,R1	;INIT COUNTER FOR ERROR-FREE CHARS	
1261	030570	012703	002623		MOV	#PATR,R3	;INIT DATA PATTERN POINTER	
1262	030574	112337	030604		MOVB	(R3)+,6\$	;GET AN EXPECTED DATA CHAR	
1263	030600	004737	007314		JSR	PC,CKDATA	;GO CHECK CHAR, CHK BCC=0	
1264	030604	100000			6\$:	BCCCHK!000		
1265	030606	000000			0			
1266	030610	005301			DEC	R1	;DECR COUNTER	
1267	030612	001370			BNE	5\$	;BR IF NOT DONE YET	
1268	030614	012701	000007		MOV	#7,R1	;INIT COUNTER FOR ERROR CHARS	
1269	030620	112337	030636		MOVB	(R3)+,9\$	;GET EXPECTED DATA CHAR	
1270	030624	052737	000001	030636	BIS	#BIT0,9\$	;EXPECT ERROR BIT 0 SET	
1271	030632	004737	007314		JSR	PC,CKDATA	;CHECK DATA, CHK BCC=1	
1272	030636	100400			9\$:	BCCCHK!RXBCC!000		
1273	030640	000000			0			
1274	030642	005301			DEC	R1	;DECR COUNTER	
1275	030644	001365			BNE	8\$	;BR IF NOT DONE YET	
1276	030646				10\$:			
1277	030646				ENDSUB			
	030646							
	030646	104403						
1278								
1279								
1280								
1281	030650	012737	000006	002366	MOV	#6,AXNUM	;SET AX BYTE NO. FOR AX3	
1282	030656	012737	000012	002364	MOV	#12,REGNUM	;SET REG NO.	
1283	030664				BGNSUB			
	030664							
	030664	104402						
1284	030666	012737	031104	002346	MOV	#30\$,RETADR	;SET SUBTEST EXIT ADRS FOR ERRORS	
1285	030674	004737	005240		JSR	PC,INITRN	;MST CLR, LOAD 2 SOM'S	
1286	030700	000026			026			
1287	030702	000211			CRC2!STRIP!DDCMP		;CHAR MODE, EVEN VRC	
1288	030704	004737	010746		JSR	PC,LODSIL	;LOAD A THIRD SOM INTO TX SILO	
1289	030710	000400			TXSOM			
1290	030712	000001			1			

L10053: TRAP C\$ESUB

 ; TEST EVEN VRC ERROR DETECTION

T16.2: TRAP C\$BSUB

```

1291 030714 004737 010574      JSR      PC,LDBYTS      ;LOAD DATA INTO TX BUFFER
1292 030720 002623      PATR
1293 030722 000017      15.
1294 030724 004737 010746      JSR      PC,LODSIL      ;LOAD 2 EOM'S INTO TX SILO
1295 030730 001000      TXEOM
1296 030732 000002      2
1297 030734 005037 002360      CLR      WAX15
1298 030740 012737 000347 002362 MOV      #TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO,WAX16
1299 030746 013737 002362 002424 MOV      WAX16,SAVLEN      ;STORE LENGTH 7
1300 030754 004737 004012      JSR      PC,WRITAX      ;SET TX AND RCV LENGTHS = 7
1301 030760 004737 004754      JSR      PC,STPLU      ;CLOCK 1ST 8 CHARS, WITH NO ERRORS
1302 030764 000130      88.
1303 030766 012701 000007      MOV      #7,R1      ;INIT COUNTER FOR LAST 7 CHARS
1304 030772 004737 007212      JSR      PC,STPERR      ;ASSERT IERR BIT FOR 1 TIME
1305 030776 000211      CRC2!STRIP!DDCMP
1306 031000 000001      1
1307 031002 004737 004754      JSR      PC,STPLU      ;CLOCK REST OF CHAR
1308 031006 000007      7
1309 031010 005301      DEC      R1      ;DECR COUNTER
1310 031012 001367      BNE      23$      ;BR IF NOT DONE TRANSMITTING YET
1311 031014 004737 004754      JSR      PC,STPLU      ;CLOCK 2 TERMINATING SYNCHS
1312 031020 000020      16.
1313 031022 012701 000010      MOV      #8.,R1      ;INIT COUNTER FOR ERROR-FREE CHARS
1314 031026 012703 002623      MOV      #PATR,R3      ;INIT DATA PATTERN POINTER
1315 031032 112337 031042      25$: MOV      (R3)+,26$      ;GET EXPECTED DATA CHAR
1316 031036 004737 007314      JSR      PC,CKDATA      ;CHK DATA, CHECK BCC=0
1317 031042 100000      26$: BCCCHK!000
1318 031044 000000      0
1319 031046 005301      DEC      R1      ;DECR COUNTER
1320 031050 001370      B      25$      ;BR IF NOT DONE YET
1321 031052 012701 000007      MOV      #7,R1      ;INIT COUNTER FOR ERROR CHARS
1322 031056 112337 031074      28$: MOV      (R3)+,29$      ;GET EXPECTED DATA CHAR
1323 031062 052737 000001 031074 BIS      #BIT0,29$      ;SET EXPECTED ERROR BIT 0
1324 031070 004737 007314      JSR      PC,CKDATA      ;CHK DATA, CHK BCC=1
1325 031074 100400      29$: BCCCHK!RXBCC!000
1326 031076 000000      0
1327 031100 005301      DEC      R1      ;DECR COUNTER
1328 031102 001365      BNE      28$      ;BR IF NOT DONE YET
1329 031104      30$:
1330 031104      ENDSUB
1331 031106 004737 003276      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
1332 031112      ENDTST
1333 031112 104401      L10054: TRAP      C$ESUB
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343

```

```

*****
.SBTTL      TEST 17 - INTEGRAL MODEM INTERFACE TEST - CHAR MODE, CRC
*
* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM IN AX3-15, AND A
* MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
* ON THE LINE UNIT OR AT THE CABLE. THE MESSAGE CONSISTS OF

```

```

1344      ;* 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
1345      ;* UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
1346      ;* SKIPPED FOR THAT UNIT.
1347      ;*****
1348      BGNTST
1349      031114      012737 000021 002434      MOV      #17.,TSTNUM      ;SET TEST NO.
1350      031114      012737 031374 002346      MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
1351      031130      004737 003276      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
1352      031134      004737 011016      JSR      PC,CKLPBK      ;CHECK LOOPBACK -
1353      031140      000010      INTGR      ;SEE IF TEST SHOULD BE RUN
1354      031142      012737 000323 031160      MOV      #1422!XYZ.V35!OP!TEST,6$ ;SET UP TO SELECT INTEGRAL MODEM
1355      031150      004737 010324      JSR      PC,SETUP      ;PROGRAM THE USYRT
1356      031154      000226      SYNCH
1357      031156      000011      STRIP!DDCMP
1358      031160      000000      6$:      .WORD      0
1359      031162      000000      000
1360      031164      004737 010656      JSR      PC,LDMSG1      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1361      031170      142777 000010 151242      BICB      #LULOP,0BSEL1      ;CLEAR LULOP, CLOCK MSG
1362      031176      012737 000012 002364      MOV      #12,REGNUM      ;SET LU REG NO. = 12
1363      031204      012703 002762      MOV      #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
1364      031210      013702 002264      9$:      MOV      TCOUNT,R2      ;INIT TIMER
1365      031214      004737 003372      10$:      JSR      PC,READLU      ;READ REG 12
1366      031220      132737 000020 002350      BITB      #IRDY,REDBYT      ;SEE IF IRDY IS SET YET
1367      031226      001011      BNE      12$      ;BR IF YES
1368      031230      005202      INC      R2      ;INCREMENT TIMER
1369      031232      001370      BNE      10$      ;BR IF NO TIME-OUT YET
1370      031234      004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
1371      ;REPORT IRDY NOT SET
1372      031240      031240 104455      ERRDF      17,EM17,ERR7
1373      031250      000451      TRAP      C$ERDF
1374      031252      012337 031262      12$:      MOV      (R3)+,16$      .WORD      17
1375      031256      004737 007314      16$:      JSR      PC,CKDATA      ;COMPARE RCV'D DATA CHAR TO EXPECTED
1376      031262      000000      0
1377      031264      000000      0
1378      031266      020327 003000      CMP      R3,#RCVBUF+14.      ;SEE IF ALL CHARS CHECKED YET
1379      031272      103746      BLO      9$      ;BR IF NOT YET
1380      031274      004737 004646      JSR      PC,WAIT50      ;STALL FOR 50 MICRO-SEC
1381      031300      004737 005052      JSR      PC,OACTIV      ;CHECK OACT = 0
1382      031304      000000      0
1383      031306      004737 006260      JSR      PC,IACTIV      ;CHECK IACT STILL = 1
1384      031312      000001      1
1385      031314      012737 000013 002364      MOV      #13,REGNUM      ;SET REG NO. = 13
1386      031322      004737 003372      JSR      PC,READLU      ;READ REG 13
1387      031326      042737 000232 002350      BIC      #RING!HDX!MODR!STBY,REDBYT      ;CLR UNUSED BITS
1388      031334      023727 002350 000000      CMP      REDBYT,#0      ;CHECK REG 13 FOR 000 (MODEM SIGNALS SHOULD BE CLEARED)
1389      031342      001414      BEQ      24$      ;BR IF CLEARED
1390      031344      012737 000000 002370      MOV      #0,GOODAT      ;SET EXPECTED DATA = 0
1391      031352      013737 002350 002372      MOV      REDBYT,BADDAT      ;SET ACTUAL DATA
1392      031360      004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
1393      ;REPORT REG MISCOMPARE
1394      031364      031364 104455      ERRDF      3,EM3,ERR2
1395      TRAP      C$ERDF

```

031366 000003
 031370 013324
 031372 015444
 1395 031374
 1396 031374
 031374
 031374 104401

24\$:
 ENDIST

.WORD 3
 .WORD EM3
 .WORD ERR2

L10055:
 TRAP CSETST

1397
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412

 SBTTL TEST 18 - V.35 MODEM INTERFACE TEST - CHAR MODE, CRC
 *
 * THE V.35 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
 * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
 * ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
 * OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
 * 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
 * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
 * SKIPPED FOR THAT UNIT.

 BGNTST

1413 031376
 031376
 1414 031376 012737 000022 002434
 1415 031404 012737 031646 002346
 1416 031412 004737 003276
 1417 031416 004737 011016
 1418 031422 000020
 1419 031424 012737 000313 031442
 1420 031432 004737 010324
 1421 031436 000226
 1422 031440 000011
 1423 031442 000000
 1424 031444 000000
 1425 031446 142777 000010 150764
 1426 031454 012737 000013 002364
 1427 031462 004737 003372
 1428 031466 132737 000001 002350
 1429 031474 001415
 1430 031476 012737 000000 002370
 1431 031504 013737 002350 002372
 1432 031512 004737 004226

T18::
 MOV #18,,TSTNUM ;SET TEST NO.
 MOV #24\$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
 JSR PC,CKLPBK ;CHECK LOOPBACK -
 V35 ; SEE IF TEST SHOULD BE RUN
 MOV #1422!XYZ!INTGRL.OP!TEST,6\$;SET UP TO SELECT V35
 JSR PC,SETUP ;PROGRAM THE USYRT
 SYNCH
 STRIP!DDCMP
 6\$: .WORD 0
 000
 BICB #LULOOB,ABSEL1 ;CLEAR LULOOB
 MOV #13,REGNUM ;SET LU REG NO. = 13
 JSR PC,READLU ;READ REG 13
 BITB #CARR,REDBYT ;CHECK FOR CARRIER FALSELY SET
 BEQ B\$;BR IF NOT SET
 MOV #000,GOODAT ;SET EXPECTED DATA
 MOV REDBYT,BADDAT ;SET ACTUAL DATA
 JSR PC,GETALL ;GET REGS FOR PRINTOUT
 ;REPORT CARRIER NOT CLEARED
 ERRDF 66,EM66,ERR7

TRAP C\$ERDF
 .WORD 66
 .WORD EM66
 .WORD ERR7

1433
 1434 031516
 031516 104455
 031520 000102
 031522 015056
 031524 020302
 1435 031526 000447
 1436 031530 152777 000010 150702
 1437 031536 004737 010656
 1438 031542 142777 000010 150670
 1439 031550 012737 000012 002364
 1440 031556 012703 002762
 1441 031562 013702 002264

8\$: BR 24\$
 BISB #LULOOB,ABSEL1 ;SET LULOOB AGAIN
 JSR PC,LDMG1 ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
 BICB #LULOOB,ABSEL1 ;CLEAR LULOOB, CLOCK MSG
 MOV #12,REGNUM ;SET LU REG NO. = 12
 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
 9\$: MOV TCOUNT,R2 ;INIT TIMER

```

1442 031566 004737 003372 10$: JSR PC,READLU ;READ REG 12
1443 031572 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1444 031600 001011 BNE 12$ ;BR IF YES
1445 031602 005202 INC R2 ;INCREMENT TIMER
1446 031604 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1447 031606 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1448 ;REPORT IRDY NOT SET
1449 031612 ERRDF 17,EM17,ERR7
031612 104455 TRAP C$ERDF
031614 000021 .WORD 17
031616 013475 .WORD EM17
031620 020302 .WORD ERR7
1450 031622 000411 BR 24$ ;ESCAPE TO END OF TEST
1451 031624 012337 031634 12$: MOV (R3)+,16$
1452 031630 004737 007314 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1453 031634 000000 16$: 0
1454 031636 000000 0
1455 031640 020327 003000 CMP R3,#RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1456 031644 103746 BLO 9$ ;BR IF NOT YET
1457 031646 24$:
1458 031646 ENDTST
031646 104401 L10056:
031646 104401 TRAP C$ETST

```

1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474

```

*****
:SBTTL TEST 19 - RS 232C AND RS 423 MODEM INTERFACE TEST - CHAR MODE, CRC
:
:* THE RS232C & RS423 (XYZ) MODEM INTERFACE IS SELECTED BY THE PROGRAM IN
:* AX3-15, AND A MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURN-
:* AROUND CONNECTOR ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
:* OR A MODEM TEST MODE. THE MESSAGE CONSISTS
:* OF 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE
:* P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS
:* PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
*****
:BGNTST

```

```

1475 031650 T19::
031650
1476 031650 012737 000023 002434 MOV #19.,TSTNUM ;SET TEST NO.
1477 031656 012737 032030 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1478 031664 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1479 031670 004737 011016 JSR PC,CKLPBK ;CHECK LOOPBACK -
1480 031674 000100 XYZ ;SEE IF TEST SHOULD BE RUN
1481 031676 012737 000233 031714 MOV #1422!V35!INTGRL!OP!TEST,6$ ;SET UP TO SELECT XYZ
1482 031704 004737 010324 JSR PC,SETUP ;PROGRAM THE USYRT
1483 031710 000226 SYNCH
1484 031712 000011 STRIP!DDCMP
1485 031714 000000 6$: .WORD 0
1486 031716 000000 000
1487 031720 004737 010656 JSR PC,LDMSG1 ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1488 031724 142777 000010 150506 BICB #LULOP,@BSEL1 ;CLEAR LULOP, CLOCK MSG
1489 031732 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
1490 031740 012703 002762 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
1491 031744 013702 002264 9$: MOV TCOUNT,R2 ;INIT TIMER

```

```

1492 031750 004737 003372 10$: JSR PC,READLU ;READ REG 12
1493 031754 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1494 031762 001011 BNE 12$ ;BR IF YES
1495 031764 005202 INC R2 ;INCREMENT TIMER
1496 031766 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1497 031770 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1498 ;REPORT IRDY NOT SET
1499 031774 ERRDF 17,EM17,ERR7
031774 104455 TRAP C$ERDF
031776 000021 .WORD 17
032000 013475 .WORD EM17
032002 020302 .WORD ERR7
1500 032004 000411 BR 24$ ;ESCAPE TO END OF TEST
1501 032006 012337 032016 12$: MOV (R3)+,16$
1502 032012 004737 007314 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1503 032016 000000 16$: 0
1504 032020 000000 0
1505 032022 020327 003000 CMP R3,#RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1506 032026 103746 BLO 9$ ;BR IF NOT YET
1507 032030 24$:
1508 032030 ENDTST
032030 104401 L10057:
032030 TRAP C$ETST

```

1509
1510
1511
1512
1513

```

1514 *****
1515 .SBTTL TEST 20 - RS 422 MODEM INTERFACE TEST - CHAR MODE, CRC
1516 *
1517 * THE RS 422 MODEM INTERFACE IS SELECTED BY THE PROGRAM IN AX3-15, AND A
1518 * MESSAGE IS TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR
1519 * ON THE LINE UNIT OR AT THE MODEM SIDE OF THE CABLE,
1520 * OR A MODEM TEST MODE. THE MESSAGE CONSISTS OF
1521 * 5 SYNCHS, 000,125,252,377,000, AND 1 SYNCH. IF THE P-TABLE FOR THE CURRENT
1522 * UNIT INDICATES THAT NO EXTERNAL TURNAROUND IS PROVIDED, THE TEST WILL BE
1523 * SKIPPED FOR THAT UNIT.
1524 *****
1525 BGNTST

```

```

1526 032032 012737 000024 002434 MOV #20.,TSTNUM ;SET TEST NO.
1527 032040 012737 032212 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1528 032046 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1529 032052 004737 011016 JSR PC,CKLPBK ;CHECK LOOPBACK -
1530 032056 000200 1422 ;SEE IF TEST SHOULD BE RUN
1531 032060 012737 000133 032076 MOV #XYZ!V35!INTGRL'OP!TEST,6$ ;SET UP TO SELECT 422
1532 032066 004737 010324 JSR PC,SETUP ;PROGRAM THE USYRT
1533 032072 000226 SYNCH
1534 032074 000011 STRIP!DDCMP
1535 032076 000000 6$: .WORD 0
1536 032100 000000 000
1537 032102 004737 010656 JSR PC,LDMSG1 ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1538 032106 142777 000010 150324 BICB #LULOOK,ABSEL1 ;CLEAR LULOOK, CLOCK MSG
1539 032114 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
1540 032122 012703 002762 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
1541 032126 013702 002264 9$: MOV TCOUNT,R2 ;INIT TIMER

```

```

1542 032132 004737 003372 10$: JSR PC,READLU ;READ REG 12
1543 032136 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
1544 032144 001011 BNE 12$ ;BR IF YES
1545 032146 005202 INC R2 ;INCREMENT TIMER
1546 032150 001370 BNE 10$ ;BR IF NO TIME-OUT YET
1547 032152 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1548 ;REPORT IRDY NOT SET
1549 ERRDF 17,EM17,ERR7

                                TRAP C$ERDF
                                .WORD 17
                                .WORD EM17
                                .WORD ERR7

1550 032166 000411 BR 24$ ;ESCAPE TO END OF TEST
1551 032170 012337 032200 12$: MOV (R3)+,16$
1552 032174 004737 007314 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
1553 032200 000000 16$: 0
1554 032202 000000 0
1555 032204 020327 003000 CMP R3,#RCVBUF+14. ;SEE IF ALL CHARS CHECKED YET
1556 032210 103746 BLO 9$ ;BR IF NOT YET
1557 032212 24$:
1558 032212 ENDTST
                                L10060:
                                TRAP C$ETST

1559 032212 104401
1560
1561
1562
1563
1564
1565 ;*****
1566 ;SBTTL TEST 21 - HALF-DUPLEX BIT (HALF DUPX) TEST
1567 ;*
1568 ;* THIS TEST VERIFIES THAT SETTING HALF-DUPLEX BIT IN REG 13 DOES NOT INHIBIT
1569 ;* LOADING OF THE USYRT TRANSMITTER FROM THE TRANSMITTER SILO.
1570 ;* A MASTER CLEAR IS ISSUED, DDCMP MODE IS ENTERED, AND THE HALF DUPX
1571 ;* BIT IN REG 13 IS SET. A MESSAGE IS LOADED INTO THE TX SILO
1572 ;* CONSISTING OF 2 SYNCHS, 000,125,252,377,000, AND 2 MORE SYNCHS.
1573 ;* THE LINE UNIT IS THEN CLOCKED EXTENSIVELY, AND THE TX SILO IS CHECKED TO
1574 ;* BE UNLOADED (ALL CHARS SHOULD HAVE BEEN REMOVED) AND THE RECEIVER
1575 ;* IS MONITORED TO INSURE THAT NO RCV FLAGS ARE GENERATED.
1576 ;*****
1576 032214 BGNTST
                                T21::
1577 032214 012737 032304 002346 MOV #24$,RETADR ;SET TEST EXIT ADRS FOR ERRORS
1578 032222 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1579 032230 004737 005240 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
1580 032234 000226 SYNCH
1581 032236 000011 STRIP!DDCMP
1582 032240 112737 000020 002352 MOVB #4DX,WRIBYT
1583 032246 004737 003450 JSR PC,WRITLU ;SET HDX BIT IN REG 13
1584 032252 004737 010516 JSR PC,LODMSG ;LOAD MSG INTO TX SILO
1585 032256 002674 MSG1+4
1586 032260 000007 7
1587 032262 004737 004754 JSR PC,STPLU ;CLK MORE THAN ENTIRE MSG
1588 032266 000136 94.
1589 032270 004737 004362 JSR PC,OSIRDY ;CHK ORDY = 1, OCOR 0
1590 032274 000001 1
1591 032276 004737 005774 JSR PC,ISIRDY ;CHK ICIR = 1, IRDY = 0

```

1592 032302 000001
 1593 032304 004737 003276 24\$: JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
 1594 032310
 032310
 032310 104401 L10061: TRAP C\$ETST

1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608

 :SBTTL TEST 22 - HALF-DUPLEX RCV DISABLED TEST WITH SILOS DISABLED
 :
 :* THIS TEST SENDS A MESSAGE IN HDX, CHAR MODE, WITH NO ERROR DETECTION, AND
 :* THE SILOS DISABLED. THE MSG CONSISTS OF 2 SYNCHS AND 2 000 CHARS.
 :* THE DATA IS SENT WITH LULCOP SET FOR TTL DATA LOOPBACK. THE PROGRAM CHECKS
 :* THAT THE RECEIVER NEVER BECOMES ACTIVE, BECAUSE THE RCV CLOCK IS INHIBITED
 :* WHEN THE HDX BIT IS SET.
 :*****

1609 032312 BGNTST
 032312
 1610 032312 012737 032510 002346 MOV #24\$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
 1611 032320 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
 1612 032324 004737 010324 JSR PC,SETUP ;PROGRAM USYRT FOR CHAR MODE, NO CRC
 1613 032330 000226 SYNCH
 1614 032332 000301 CRC2!CRC1!DDCMP
 1615 032334 000000 000
 1616 032336 000000 000
 1617 032340 012737 000013 002364 MOV #13,REGNUM ;SET LU REG NO. = 13
 1618 032346 112737 000020 002352 MOVB #HDX,WRIBYT ;SET HDX BIT IN REG 13
 1619 032354 004737 003450 JSR PC,WRITLU
 1620 032360 012737 000014 002364 MOV #14,REGNUM ;SET LU REG NO. = 14
 1621 032366 112737 000140 002352 MOVB #TXEN!DISSI,WRIBYT ;DISABLE SILOS
 1622 032374 004737 003450 JSR PC,WRITLU
 1623 032400 012737 000140 002416 MOV #TXEN!DISSI,DISILO
 1624 032406 012737 000002 002366 MOV #2,AXNUM ;SET AX BYTE NO FOR AX1
 1625 032414 112737 000000 002360 MOVB #000,WAX15 ;SET TSOM IN USYRT
 1626 032422 112737 000001 002362 MOVB #TSOM,WAX16
 1627 032430 004737 004012 JSR PC,WRITAX
 1628 032434 004737 004754 JSR PC,STPLU ;CLOCK FIRST SYNCH OUT
 1629 032440 000013 11.
 1630 032442 112737 000000 002360 MOVB #000,WAX15 ;LOAD FIRST 000 DATA CHAR INTO USYRT
 1631 032450 112737 000000 002362 MOVB #000,WAX16
 1632 032456 004737 004012 JSR PC,WRITAX
 1633 032462 004737 004754 JSR PC,STPLU ;CLOCK SECOND SYNCH
 1634 032466 000010 8.
 1635 032470 004737 004012 JSR PC,WRITAX ;LOAD SECOND 000 CHAR
 1636 032474 004737 004754 JSR PC,STPLU ;CLOCK FIRST 000 CHAR OUT
 1637 032500 000013 11.
 1638 032502 004737 006260 JSR PC,IACTIV ;CHK FOR IACT = 0 (RECEIVER NOT ACTIVE)
 1639 032506 000000 0
 1640 032510 005037 002416 24\$: CLR DISILO ;CLEAR DISABLE SILO FLAG
 1641 032514 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
 1642 032520
 032520
 032520 104401 L10062: TRAP C\$ETST
 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

```

*****
SBTTL      TEST 23 - INTERACTION OF MODEM CONTROL BITS
*
* THIS TEST WILL BE RUN ONLY IF THE P-TABLE FOR THIS UNIT INDICATES THAT THE
* H3254 AND H3255 TEST CONNECTORS ARE INSTALLED. OTHERWISE, THE TEST WILL BE
* SKIPPED FOR THE UNIT.
* THE FOLLOWING SUBTESTS ARE PERFORMED:
* - A MASTER CLEAR IS DONE AND REG 13 IS READ AND CHECKED FOR INITIALIZED
*   STATE, WITH LULOOK SET TO 1. THEN, LULOOK IS CLEARED AND REG 13 IS READ
*   AND CHECKED FOR THE PROPER STATE, WITH LULOOK CLEARED.
*   REG 13 IS THEN LOADED WITH 0'S, AND READ AND CHECKED FOR THE INITIALIZED
*   STATE.
*   REG 17 IS THEN READ AND CHECKED FOR INITIALIZED STATE.
* - RUN IS SET IN BSEL1, AND REG 13 IS READ AND CHECKED FOR RING SET.
* - POLL IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGO SET.
* - BPOLL IS SET IN REG 12, ONLY TO LIGHT THE LED FOR THIS SIGNAL.
* - DTR IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR DTR AND MODR SET.
* - SELFR IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR SIGR SET.
* - HDX IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR HDX SET.
* - MAINT1 IS SET IN REG 13, AND REG 17 IS READ AND CHECKED FOR TEST MODE SET.
* - SELSBY IS SET IN REG 13, AND REG 13 IS READ AND CHECKED FOR STBY SET.
* - A MASTER CLEAR IS DONE, 2 TSOM'S ARE LOADED INTO THE TX SILO, THE LINE
*   UNIT IS Clocked UNTIL THE TRANSMITTER IS ACTIVE, AND REG 13 IS READ AND
*   CHECKED FOR RTS, CS, CARR SET.
*****
BGNTST
  
```

1673 032522
 032522
 1674 032522 012737 000027 002434
 1675 032530 012737 034022 002346
 1676 032536 004737 011016
 1677 032542 100000

```

T23::
MOV      #23,,TSTNUM      ;SET TEST NO.
MOV      #A12,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
JSR      PC,CKLPBK        ;SEE IF H3254,5 INSTALLED - SKIP TEST IF NOT
TCCHK
  
```

 : DO MASTER CLEAR, CHK REGS 13,17 FOR INITIALIZED STATES

1678
1679
1680
1681

1682 032544
 032544
 032544 104402
 1683 032546 004737 003276
 1684 032552 012737 000013 002364
 1685 032560 004737 003372
 1686 032564 023727 002350 000210
 1687 032572 001416
 1688 032574 012737 000210 002370
 1689 032602 013737 002350 002372
 1690 032610 004737 004226

```

BGNSUB
T23.1:
TRAP      C$BSUB
JSR      PC,MSTCLR        ;ISSUE MASTER CLEAR
MOV      #13,REGNUM       ;SET REG NO. = 13
JSR      PC,READLU        ;READ REG 13
CMP      REDBYT,#RING!MODR ;CHECK REG 13 FOR INIT'D STATE
BEQ      6$               ;BR IF REG 13 INIT'D
MOV      #RING!MODR,GOODAT ;SET EXPECTED DATA
MOV      REDBYT,BADDAT     ;SET ACTUAL DATA
3$:      JSR      PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT REG MISCOMPARE
ERRDF    3,EM3,ERR2
  
```

1691

1692 032614
 032614 104455
 032616 000003
 032620 013324
 032622 015444

```

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

1693 032624

ESCAPE SUB

PC	Address	Offset	Value	Instruction	Comment	Trap	Word	Escape
1694	032624	104410	000010	147602	6\$: BICB #LULOOK,@BSEL1 ;CLEAR LULOOK			
1695	032626	000170	003372		JSR PC,READLU ;READ REG 13			
1696	032630	142777	002350	000000	CMP REDBYT,#0 ;CHECK FOR INITIALIZED STATE			
1697	032636	004737	000000		BEQ 8\$;BR IF OK			
1698	032642	023727	000000	002370	MOV #0,GOODAT ;GET EXPECTED DATA			
1699	032650	001416	002350	002372	MOV REDBYT,BADDAT ;GET ACTUAL DATA			
1700	032652	012737	004226		JSR PC,GETALL ;GET REGS FOR PRINTOUT			
1701	032660	013737			;PEPORT REG NOT INITIALIZED BY MASTER CLEAR			
1702	032666	004737			ERRDF 2,EM2,ERR2			
1703	032672	104455			ESCAPE SUB			
1704	032672	000002						
1705	032674	000002						
1706	032676	013265						
1707	032700	015444						
1708	032702	104410						
1709	032704	000112						
1710	032706	005037	002352		8\$: CLR WRIBYT ;SET DATA = 0 TO BE WRITTEN			
1711	032712	004737	003450		JSR PC,WRITLU ;LOAD 0'S INTO REG 13			
1712	032716	004737	003372		JSR PC,READLU ;READ REG 13			
1713	032722	023727	002350	000000	CMP REDBYT,#000 ;CHECK FOR REG 13 CLEARED			
1714	032730	001407			BEQ 9\$;BR IF CLEARED			
1715	032732	012737	000000	002370	MOV #000,GOODAT ;SET EXPECTED DATA			
1716	032740	013737	002350	002372	MOV REDBYT,BADDAT ;SET ACTUAL DATA			
1717	032746	000720			BR 3\$;GO PRINT ERROR			
1718	032750	012737	000017	002364	9\$: MOV #17,REGNUM ;SET REG NO. = 17			
1719	032756	004737	003372		JSR PC,READLU ;READ REG 17			
1720	032762	042737	000002	002350	BIC #MCLK,REDBYT ;IGNORE MCLK BIT			
1721	032770	123727	002350	000051	CMPB REDBYT,#TXDATA!ICIR!DDCMP ;CHK REG 17 FOR INIT'D STATE			
1722	032776	001407			BEQ 10\$;BR IF REG 17 INITIALIZED			
1723	033000	012737	000051	002370	MOV #TXDATA!ICIR!DDCMP,GOODAT ;SET EXPECTED DATA			
1724	033006	013737	002350	002372	MOV REDBYT,BADDAT ;SET ACTUAL DATA			
1725	033014	000675			BR 3\$;GO REPORT ERROR			
1726	033016				10\$: ENDSUB			
1727	033016							
1728	033016	104403						
1729	033020							
1730	033020							
1731	033020	104402						
1732	033022	004737	003276					
1733	033026	105077	147406					
1734	033032	112777	000200	147400				
1735	033040	112777	000010	147372				
1736	033046	012737	000013	002364				
1737	033054	004737	003372					
1738	033060	132737	000200	002350				
1739	033066	001010						
1740	033070	004737	004226					
1741	033074							

```

          ESCAPE SUB
98:
          ENDSUB

```

L10065: TRAP CSESUB

```

:-----:
: SET POLL IN REG 13, CHK FOR SIGQ SET IN REG 17
:-----:
      BGNSUB

```

```
T23.3:      TRAP      CSBSUB
```

```

JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
MOV      #POLL,WRIBYT
MOV      #13,REGNUM     ;SET REG NO. = 13
JSR      PC,WRITLU      ;SET POLL IN REG 13
MOV      #17,REGNUM     ;SET REG NO. = 17
JSR      PC,READLU      ;READ REG 17
BITB     #SIGQ,REDBYT   ;SEE IF SIGQ = 1
BNE      6$             ;BR IF SIGQ = 1
JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
;REPORT SIGQ NOT SET
ERRDF    63,EM63,ERR7

```

```
TRAP      C$ERDF
.WORD     63
.WORD     EM63
.WORD     ERR7
```

68: ENDSUB

```

Li0066:      TRAP      CSESUB

```

```

; SET BPOLL IN REG 12, TO LIGHT LED ONLY
;-----

```

```
T23.4:      TRAP      CSBSUB
```

```

JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
MOV      #12,REGNUM     ;SET LU REG NO. = 12
MOVB     #BPOLL,WRIBYT  ;SET BPOLL IN LU REG 12
JSR      PC,WRITLU
ENDSUB

```

L10067: TRAP CSESUB

```

; SET DTR IN REG 13, CHECK FOR DTR AND MODR SET IN REG 13
-----

```

123.5: TRAP CSBSUB

10071

```

1812 033432 104403
1813
1814 -----
1815 : SET HDX IN REG 13, CHK FOR HDX SET IN REG 13
1816 -----
1816 033434 BGNSUB
1816 033434 104402
1817 033436 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1818 033442 112737 000020 002352 MOVB #HDX,WRIBYT
1819 033450 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1820 033456 004737 003450 JSR PC,WRITLU ;SET HDX IN REG 13
1821 033462 004737 003372 JSR PC,READLU ;READ REG 13
1822 033466 132737 000020 002350 BITB #HDX,REDBYT ;SEE IF HDX = 1
1823 033474 001006 BNE 6$ ;BR IF HDX = 1
1824 033476 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1825 ;REPORT HDX NOT SET
1826 033502 ERRDF 58,EM58,ERR7
1826 033502 104455 TRAP C$ERDF
1826 033504 000072 .WORD 58
1826 033506 014707 .WORD EM58
1826 033510 020302 .WORD ERR7
1827 033512 6$:
1828 033512 ENDSUB
1828 033512 L10072:
1828 033512 104403 TRAP C$ESUB
1829
1830 -----
1831 : SET MAINT1 IN REG 13, CHK FOR TEST MODE SET IN REG 17
1832 -----
1833 033514 BGNSUB
1833 033514 104402
1834 033516 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1835 033522 112737 000010 002352 MOVB #MAINT1,WRIBYT
1836 033530 012737 000013 002364 MOV #13,REGNUM ;SET REG NO. = 13
1837 033536 004737 003450 JSR PC,WRITLU ;SET MAINT1 IN REG 13
1838 033542 012737 000017 002364 MOV #17,REGNUM ;SET REG NO. = 17
1839 033550 142777 000010 146662 BICB #LULOOP,@BSEL1 ;CLEAR LULOOP
1840 033556 004737 003372 JSR PC,READLU ;READ REG 17
1841 033562 132737 000004 002350 BITB #TESTMD,REDBYT ;SEE IF TESTMD = 1
1842 033570 001006 BNE 6$ ;BR IF TESTMD = 1
1843 033572 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1844 ;REPORT TEST MODE NOT SET BY MAINT1
1845 033576 ERRDF 52,EM52,ERR7
1845 033576 104455 TRAP C$ERDF
1845 033600 000064 .WORD 52
1845 033602 014527 .WORD EM52
1845 033604 020302 .WORD ERR7
1846 033606 6$:
1847 033606 ENDSUB
1847 033606 L10073:
1847 033606 104403 TRAP C$ESUB
1848
1849 -----
1850 : SET SELSBY IN REG 13, CHK FOR STBY SET IN REG 13
1851 -----

```

1852	033610				BGNSUB				
	033610							T23.9:	
	033610	104402						TRAP	C\$BSUB
1853	033612	004737	003276		JSR	PC,MSTCLR	;ISSUE MASTER CLEAR		
1854	033616	112737	000002	002352	MOVB	#SELSBY,WRIBYT			
1855	033624	012737	000013	002364	MOV	#13,REGNUM	;SET REG NO. = 13		
1856	033632	004737	003450		JSR	PC,WRITLU	;SET SELSBY IN REG 13		
1857	033636	004737	003372		JSR	PC,READLU	;READ REG 13		
1858	033642	132737	000002	002350	BITB	#STBY,REDBYT	;SEE IF STBY = 1		
1859	033650	001006			BNE	6\$	;BR IF STBY = 1		
1860	033652	004737	004226		JSR	PC,GETALL	;GET REGS FOR PRINTOUT		
1861					;REPORT	STBY NOT SET			
1862	033656				ERRDF	59,EM59,ERR7			
	033656	104455						TRAP	C\$ERDF
	033660	000073						.WORD	59
	033662	014723						.WORD	EM59
	033664	020302						.WORD	ERR7
1863	033666				6\$:				
1864	033666				ENDSUB				
	033666							L10074:	
	033666	104403						TRAP	C\$ESUB
1865					-----				
1866					DO MASTER CLEAR, LOAD 2 TSOM'S INTO TX SILO, CLOCK TRANSMITTER UNTIL ACTIVE,				
1867					CHECK FOR RTS, CS, CARR SET IN REG 13				
1868					-----				
1869	033670				BGNSUB				
	033670							T23.10:	
	033670	104402						TRAP	C\$BSUB
1870	033672	004737	005240		JSR	PC,INITRN	;MST CLR, LOAD SOM'S, CLK TRANSMITTER		
1871	033676	000000			000				
1872	033700	000000			000				
1873	033702	012737	000013	002364	MOV	#13,REGNUM	;SET REG NO. = 13		
1874	033710	004737	003372		JSR	PC,READLU	;READ REG 13		
1875	033714	132737	000040	002350	BITB	#RTS,REDBYT	;SEE IF RTS = 1		
1876	033722	001010			BNE	6\$	;BR IF RTS = 1		
1877	033724	004737	004226		JSR	PC,GETALL	;GET REGS FOR PRINTOUT		
1878					;REPORT	RTS NOT SET			
1879	033730				ERRDF	60,EM60,ERR7			
	033730	104455						TRAP	C\$ERDF
	033732	000074						.WORD	60
	033734	014740						.WORD	EM60
	033736	020302						.WORD	ERR7
1880	033740				ESCAPE	SUB			
	033740	104410						TRAP	C\$ESCAPE
	033742	000056						.WORD	L10075-
1881	033744	132737	000004	002350	6\$:	BITB	#CS,REDBYT	;SEE IF CS = 1	
1882	033752	001010			BNE	9\$	;BR IF CS = 1		
1883	033754	004737	004226		JSR	PC,GETALL	;GET REGS FOR PRINTOUT		
1884					;REPORT	CS NOT SET			
1885	033760				ERRDF	61,EM61,ERR7			
	033760	104455						TRAP	C\$ERDF
	033762	000075						.WORD	61
	033764	014754						.WORD	EM61
	033766	020302						.WORD	ERR7
1886	033770				ESCAPE	SUB			
	033770	104410						TRAP	C\$ESCAPE
	033772	000026						.WORD	L10075-

```

1887 033774 132737 000001 002350 9$: BITB #CARR,REDBYT ;SEC IF CARR = 1
1888 034002 001006 BNE 12$ ;BR IF CARR = 1
1889 034004 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
1890 ;REPORT CARR NOT SET
1891 034010 ERRDF 62,EM62,ERR7
      034010 104455 TRAP C$ERDF
      034012 000076 .WORD 62
      034014 014767 .WORD EM62
      034016 020302 .WORD ERR7
1892 034020 12$:
1893 034020 ENDSUB
      034020 L10075:
      034020 104403 TRAP C$ESUB
1894
1895 034022 A12:
1896 034022 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR TO CLEAN UP
1897 034026 ENDTST
      034026 L10063:
      034026 104401 TRAP C$ESTST
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917 034030
      034030
1918 034030 012737 000030 002434 MOV #24,,TSTNUM ;SET TEST NO.
1919 034036 012737 034216 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
1920 034044 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
1921 034050 004737 011016 JSR PC,CKLPBK ;CHECK LOOPBACK, GET MODEM SELECTION
1922 034054 000000 0
1923 034056 013737 002422 034074 MOV MODINT,6$ ;SET MODEM SELECTION
1924 034064 004737 010324 JSR PC,SETUP ;PROGRAM THE USYRT
1925 034070 000000 000
1926 034072 000300 CRC2!CRC1 ;BIT MODE, NO ERR DET
1927 034074 000000 6$: .WORD 0 ;MODEM SELECTION GOES HERE
1928 034076 000000 000
1929 034100 004737 010204 JSR PC,LODATA ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
1930 034104 012737 001177 003100 MOV #RXE!177,RCVBUF+78. ; SET LAST DATA CHAR IN BUFFER
1931 034112 142777 000010 146320 BICB #LULOOP,@BSEL1 ;CLEAR LULOOP, CLOCK MSG
1932 034120 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
1933 034126 012703 002762 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
1934 034132 013702 002264 9$: MOV TCOUNT,R2 ;INIT TIMER
  
```

 SBTTL TEST 24 - DATA TEST - BIT MODE, NO ERR DET
 *
 * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH ERROR DETECTION
 * INHIBITED. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
 * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
 * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 * TEST WILL NOT BE RUN.
 * PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 * 375,373,367,357,337,277,177
 * 8-BIT CHARACTERS ARE USED.

 BGNTST

124::

1935	034136	004737	003372	10\$:	JSR	PC,READLU	:READ REG 12	
1936	034142	132737	000020		BITB	#IRDY,REDBYT	:SEE IF IRDY IS SET YET	
1937	034150	001011			BNE	12\$	:BR IF YES	
1938	034152	005202			INC	R2	:INCREMENT TIMER	
1939	034154	001370			BNE	10\$	:BR IF NO TIME-OUT YET	
1940	034156	004737	004226		JSR	PC,GETALL	:GET REGS FOR PRINTOUT	
1941				:REPORT	IRDY NOT SET			
1942	034162				ERRDF	17,EM17,ERR7		
	034162	104455						TRAP C\$ERDF
	034164	000021						.WORD 17
	034166	013475						.WORD EM17
	034170	020302						.WORD ERR7
1943	034172	000411			BR	24\$	:ESCAPE TO END OF TEST	
1944	034174	012337	034204	12\$:	MOV	(R3)+,16\$		
1945	034200	004737	007314		JSR	PC,CKDATA	:COMPARE RCV'D DATA CHAR TO EXPECTED	
1946	034204	000000		16\$:	0			
1947	034206	000000			0			
1948	034210	020327	003102		CMP	R3,#RCVBUF+80.	:SEE IF ALL CHARS CHECKED YET	
1949	034214	103746			BLO	9\$	:BR IF NOT YET	
1950	034216			24\$:				
1951	034216			ENDTST				
	034216							L10076:
	034216	104401						TRAP C\$ETST

1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970

```

*****
:SBITL      TEST 25 - DATA TEST - CHAR MODE, NO ERR DET
:
:  A MESSAGE IS INITIATED IN CHAR MODE, WITH ERROR DETECTION
:  INHIBITED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
:  AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
:  THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
:  IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
:  TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
:  TEST WILL NOT BE RUN.
:  PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
:              375,373,367,357,337,277,177
:  8-BIT CHARACTERS ARE USED.
*****

```

1971	034220			BGN1ST				
	034220							T25::
1972	034220	012737	000031	002434	MOV	#25,,TSTNUM	:SET TEST NO.	
1973	034226	012737	034406	002346	MOV	#24\$,RETADR	:SET TEST EXIT ADDRESS FOR ERRORS	
1974	034234	004737	003276		JSR	PC,MSTCLR	:ISSUE MASTER CLEAR	
1975	034240	004737	011016		JSR	PC,CKLPBK	:CHECK LOOPBACK, GET MODEM SELECTION	
1976	034244	000000			0			
1977	034246	013737	002422	034264	MOV	MODINT,6\$	:SET MODEM SELECTION	
1978	034254	004737	010324		JSR	PC,SETUP	:PROGRAM THE USYRT	
1979	034260	000226			SYNCH			
1980	034262	000311			CRC2!CRC1!STRIP!DDCMP		:CHAR MODE, NO ERR DET	
1981	034264	000000			.WORD 0		:MODEM SELECTION GOES HERE	
1982	034266	000000			000			
1983	034270	004737	010204		JSR	PC,LODATA	:LOAD MSG INTO TX SILO AND RCV'D DATA BUF	
1984	034274	012737	000177	003100	MOV	#177,RCVBUF+78.	:SET LAST DATA CHAR IN BUFFER	

1985	034302	142777	000010	146130	BICB	#LULOOB, @BSEL1	;CLEAR LULOOB, CLOCK MSG		
1986	034310	012737	000012	002364	MOV	#12, REGNUM	;SET LU REG NO. = 12		
1987	034316	012703	002762		MOV	#RCVBUF, R3	;GET POINTER TO RCV MSG BUF		
1988	034322	013702	002264		9\$: MOV	TCOUNT, R2	;INIT TIMER		
1989	034326	004737	003372		10\$: JSR	PC, READLU	;READ REG 12		
1990	034332	132737	000020	002350	BITB	#IRDY, REDBYT	;SEE IF IRDY IS SET YET		
1991	034340	001011			BNE	12\$	;BR IF YES		
1992	034342	005202			INC	R2	;INCREMENT TIMER		
1993	034344	001370			BNE	10\$	;BR IF NO TIME-OUT YET		
1994	034346	004737	004226		JSR	PC, GETALL	;GET REGS FOR PRINTOUT		
1995					;REPORT	IRDY NOT SET			
1996	034352				ERRDF	17, EM17, ERR7			
	034352	104455						TRAP	C\$ERDF
	034354	000021						.WORD	17
	034356	013475						.WORD	EM17
	034360	020302						.WORD	ERR7
1997	034362	000411			BR	24\$	;ESCAPE TO END OF TEST		
1998	034364	012337	034374		12\$: MOV	(R3)+, 16\$			
1999	034370	004737	007314		JSR	PC, CKDATA	;COMPARE RCV'D DATA CHAR TO EXPECTED		
2000	034374	000000			16\$: 0				
2001	034376	000000			0				
2002	034400	020327	003102		CMP	R3, #RCVBUF+80.	;SEE IF ALL CHARS CHECKED YET		
2003	034404	103746			BLO	9\$	;BR IF NOT YET		
2004	034406				24\$:				
2005	034406				ENDTST				
	034406							L10077:	
	034406	104401						IRAP	C\$ETST
2006									
2007									
2008									
2009									
2010									
2011									
2012									
2013									
2014									
2015									
2016									
2017									
2018									
2019									
2020									
2021									
2022									
2023									
2024									
2025	034410								
	034410								
2026	034410	012737	000032	002434	MOV	#26, TSTNUM	;SET TEST NO.		
2027	034416	012737	034576	002346	MOV	#24\$, RETADR	;SET TEST EXIT ADDRESS FOR ERRORS		
2028	034424	004737	003276		JSR	PC, MSTCLR	;ISSUE MASTER CLEAR		
2029	034430	004737	011016		JSR	PC, CKLPBK	;CHECK LOOPBACK, GET MODEM SELECTION		
2030	034434	000000			0				
2031	034436	013737	002422	034454	MOV	MODINT, 6\$	;SET MODEM SELECTION		
2032	034444	004737	010324		JSR	PC, SETUP	;PROGRAM THE USYRT		
2033	034450	000000			000				
2034	034452	000000			000		;BIT MODE CRC-CCITT-1		

```

*****
;SBTTL      TEST 26 - DATA TEST - BIT MODE, CRC-CCITT-1
;
; * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-1 ERROR
; * DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
; * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
; * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
; * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
; * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
; * TEST WILL NOT BE RUN.
; * PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
; *             375,373,367,357,337,277,177
; * 8-BIT CHARACTERS ARE USED.
*****
;BGNST

```

T26::

```

2035 034454 000000      6$: .WORD 0 ;MODEM SELECTION GOES HERE
2036 034456 000000      000
2037 034460 004737 010204 JSR PC,LODATA ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2038 034464 012737 101177 003100 MOV #CRCCHK!RXEBL!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
2039 034472 142777 000010 145740 BICB #LULOOB, @BSEL1 ;CLEAR LULOOB, CLOCK MSG
2040 034500 012737 000012 002364 MOV #12,REGNUM ;SET LU REG NO. = 12
2041 034506 012703 002762 MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
2042 034512 013702 002264 9$: MOV TCOUNT,R2 ;INIT TIMER
2043 034516 004737 003372 10$: JSR PC,READLU ;READ REG 12
2044 034522 132737 000020 002350 BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
2045 034530 001011 BNE 12$ ;BR IF YES
2046 034532 005202 INC R2 ;INCREMENT TIMER
2047 034534 001370 BNE 10$ ;BR IF NO TIME-OUT YET
2048 034536 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
2049 ;REPORT IRDY NOT SET
2050 ERRDF 17,EM17,ERR7
      TRAP C$ERDF
      .WORD 17
      .WORD EM17
      .WORD ERR7
2051 034552 000411 BR 24$ ;ESCAPE TO END OF TEST
2052 034554 012337 034564 12$: MOV (R3)+,16$
2053 034560 004737 007314 JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
2054 034564 000000 16$: 0
2055 034566 000000 0
2056 034570 020327 003102 CMP R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
2057 034574 103746 BLO 9$ ;BR IF NOT YET
2058 034576 24$:
2059 034576 ENDTST
      L10100.
      TRAP C$ETST
      .WORD 10100
2060
2061
2062
2063
2064
2065 *****
2066 .SBTTL TEST 27 - DATA TEST - BIT MODE, CRC-CCITT-0
2067 *
2068 * A MESSAGE IS INITIATED IN BIT-STUFF MODE, WITH CRC-CCITT-0 ERROR
2069 * DETECTION. THE MESSAGE CONSISTS OF 5 FLAGS, PAT A REPEATED 2 TIMES,
2070 * AND 2 FLAGS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
2071 * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
2072 * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
2073 * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
2074 * TEST WILL NOT BE RUN.
2075 * PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
2076 * 375,373,367,357,337,277,177
2077 * 8-BIT CHARACTERS ARE USED.
2078 *****
2079 034600 BGNTST
      T27::
      034600
2080 034600 012737 000033 002434 MOV #27.,TSTNUM ;SET TEST NO.
2081 034606 012737 034766 002346 MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
2082 034614 004737 003276 JSR PC,MSTCLR ;ISSUE MASTER CLEAR
2083 034620 004737 011016 JSR PC,CKLPBK ;CHECK LOOPBACK, GET MODEM SELECTION
2084 034624 000000 0

```

```

2085 034626 013737 002422 034644      MOV      MODINT,6$      ;SET MODEM SELECTION
2086 034634 004737 010324      JSR      PC,SETUP      ;PROGRAM THE USYRT
2087 034640 000000
2088 034642 000100
2089 034644 000000      6$:      .WORD      0      ;BIT MODE, CRC-CCITT-0
2090 034646 000000      ;MODEM SELECTION GOES HERE
2091 034650 004737 010204      JSR      PC,LODATA      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2092 034654 012737 101177 003100      MOV      #CRCCHK.RXEBL:177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
2093 034662 142777 000010 145550      BICB      #LULOOP,@BSEL1 ;CLEAR LULOOP, CLOCK MSG
2094 034670 012737 000012 002364      MOV      #12,REGNUM      ;SET LU REG NO. = 12
2095 034676 012703 002762      MOV      #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
2096 034702 013702 002264      9$:      MOV      TCOUNT,R2      ;INIT TIMER
2097 034706 004737 003372      10$:     JSR      PC,READLU      ;READ REG 12
2098 034712 132737 000020 002350      BITB      #IRDY,REDBYT      ;SEE IF IRDY IS SET YET
2099 034720 001011      BNE      12$      ;BR IF YES
2100 034722 005202      INC      R2      ;INCREMENT TIMER
2101 034724 001370      BNE      10$      ;BR IF NO TIME-OUT YET
2102 034726 004737 004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
2103      ;REPORT IRDY NOT SET
2104      ERRDF      17,EM17,ERR7
2105 034732 104455
2106 034734 000021
2107 034736 013475
2108 034740 020302
2109 034742 000411
2110 034744 012337 034754      12$:     MOV      24$      ;ESCAPE TO END OF TEST
2111 034750 004737 007314      JSR      PC,CKDATA      ;COMPARE RCV'D DATA CHAR TO EXPECTED
2112 034754 000000      16$:     0
2113 034756 000000      0
2114 034760 020327 003102      CMP      R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
2115 034762 103746      BLO      9$      ;BR IF NOT YET
2116 034766
2117 034766      24$:     ENDTST
2118 034766
2119 034766      _10101:
2120 104401      TRAP      C$E1ST
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133 034770
2134 034770 012737 000034 002434      MOV      #28.,TSTNUM      ;SET TEST NO.

```

 SBTTL TEST 28 - DATA TEST - CHAR MODE, CRC-16

 * A MESSAGE IS INITIATED IN CHAR MODE, WITH CRC-16 ERROR
 * DETECTION. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
 * AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
 * THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
 * IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
 * TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
 * TEST WILL NOT BE RUN.
 * PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
 * 375,373,367,357,337,277,177
 * 8-BIT CHARACTERS ARE USED.

 BGNST

128::

```

2135 034776 012737 035156 002346      MOV    #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
2136 035004 004737 003276              JSR    PC,MSTCLR      ;ISSUE MASTER CLEAR
2137 035010 004737 011016              JSR    PC,CKLPBK      ;CHECK LOOPBACK, GET MODEM SELECTION
2138 035014 000000 0              0
2139 035016 013737 002422 035034      MOV    MODINT,6$      ;SET MODEM SELECTION
2140 035024 004737 010324              JSR    PC,SETUP      ;PROGRAM THE USYRT
2141 035030 000226              SYNCH
2142 035032 000011              STRIP!DDCMP
2143 035034 000000 6$: .WORD 0      ;MODEM SELECTION GOES HERE
2144 035036 000000 000
2145 035040 004737 010204              JSR    PC,L0DATA      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2146 035044 012737 100577 003100      MOV    #CRCCHK!RXBCC!177,RCVBUF+78. ;SET LAST DATA CHAR IN BUFFER
2147 035052 142777 000010 145360      BICB   #!LLOOP,0BSEL1 ;CLEAR LLOOP, CLOCK MSG
2148 035060 012737 000012 002364      MOV    #12,REGNUM    ;SET LU REG NO. = 12
2149 035066 012703 002762              MOV    #RCVBUF,R3     ;GET POINTER TO RCV MSG BUF
2150 035072 013702 002264 9$: MOV    TCOUNT,R2      ;INIT TIMER
2151 035076 004737 003372 10$: JSR    PC,READLU      ;READ REG 12
2152 035102 132737 090020 002350      BITB   #IRDY,REDBYT  ;SEE IF IRDY IS SFT YET
2153 035110 001011              BNE     12$           ;BR IF YES
2154 035112 005202              INC     R2           ;INCREMENT TIMER
2155 035114 001370              BNE     10$           ;BR IF NO TIME-OUT YET
2156 035116 004737 004226              JSR    PC,GETALL      ;GET REGS FOR PRINTOUT
2157 ;REPORT IRDY NOT SET
2158 ERRDF 17,EM17,ERR7
2159 035122 104455
2160 035124 000021 TRAP C$ERDF
2161 035126 013475 .WORD 17
2162 035130 020302 .WORD EM17
2163 035132 000411 .WORD ERR7
2164 035134 012337 035144 12$: BR 24$      ;ESCAPE TO END OF TEST
2165 035140 004737 007314      MOV    (R3)+,16$
2166 035144 000000 16$: JSR    PC,CKDATA      ;COMPARE RCV'D DATA CHAR TO EXPECTED
2167 035146 000000 0
2168 035150 020327 003102      CMP    R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
2169 035154 103746 24$: BLO 9$      ;BR IF NOT YET
2170 035156
2171 035156
2172 035156
2173 035156 104401 L10102: TRAP C$ETST
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185

```

```

*****
SBTTL TEST 29 - DATA TEST - CHAR MODE, ODD VRC
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH ODD VRC ERROR DETECTION
* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE
* TEST WILL NOT BE RUN.
* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
* 375,373,367,357,337,277,177
* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).

```

```

2186
2187 035160
2188 035160 012737 000035 002434
2189 035166 012737 035342 002346
2190 035174 004737 003276
2191 035200 004737 011016
2192 035204 000000
2193 035206 013737 002422 035224
2194 035214 004737 010324
2195 035220 000026
2196 035222 000111
2197 035224 000000
2198 035226 000347
2199 035230 004737 010204
2200 035234 142777 000010 145176
2201 035242 012737 000012 002364
2202 035250 012703 002762
2203 035254 013702 002264
2204 035260 004737 003372
2205 035264 132737 000020 002350
2206 035272 001011
2207 035274 005202
2208 035276 001370
2209 035300 004737 004226
2210
2211 035304
2212 035314 000412
2213 035316 112337 035330
2214 035322 005203
2215 035324 004737 007314
2216 035330 100000
2217 035332 000000
2218 035334 020327 003102
2219 035340 103745
2220 035342
2221 035342
2222 035342 104401
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235

*****
BGNTST
T29::
MOV #29.,TSTNUM ;SET TEST NO.
MOV #24$,RETADR ;SET TEST EXIT ADDRESS FOR ERRORS
JSR PC,MSTCLR ;ISSUE MASTER CLEAR
JSR PC,CKLPBK ;CHECK LOOPBACK, GET MODEM SELECTION
0
MOV MODINT,6$ ;SET MODEM SELECTION
JSR PC,SETUP ;PROGRAM THE USYRT
026
CRC1!STRIP!DDCMP
6$: .WORD 0 ;MODEM SELECTION GOES HERE
TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO
JSR PC,LODATA ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
BICB #LULOOP,@BSEL1 ;CLEAR LULOOP, CLOCK MSG
MOV #12,REGNUM ;SET LU REG NO. = 12
MOV #RCVBUF,R3 ;GET POINTER TO RCV MSG BUF
9$: MOV TCOUNT,R2 ;INIT TIMER
10$: JSR PC,READLU ;READ REG 12
BITB #IRDY,REDBYT ;SEE IF IRDY IS SET YET
BNE 12$ ;BR IF YES
INC R2 ;INCREMENT TIMER
BNE 10$ ;BR IF NO TIME-OUT YET
JSR PC,GETALL ;GET REGS FOR PRINTOUT
;REPORT IRDY NOT SET
ERRDF 17,EM17,ERR7
TRAP C$ERDF
.WORD 17
.WORD EM17
.WORD ERR7
12$: BR 24$ ;ESCAPE TO END OF TEST
MOV (R3)+,16$ ;GET AN EXPECTED DATA BYTE
INC R3 ;INCREMENT POINTER
JSR PC,CKDATA ;COMPARE RCV'D DATA CHAR TO EXPECTED
16$: BCCCHK
0
CMP R3,#RCVBUF+80. ;SEE IF ALL CHARS CHECKED YET
BLO 9$ ;BR IF NOT YET
24$:
ENDTST
L10103:
TRAP C$ETST

*****
SBTTL TEST 30 - DATA TEST - CHAR MODE, EVEN VRC
*
* A MESSAGE IS INITIATED IN CHAR MODE, WITH EVEN VRC ERROR DETECTION
* SELECTED. THE MESSAGE CONSISTS OF 5 SYNCHS, PAT A REPEATED 2 TIMES,
* AND 2 SYNCHS. IF THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED,
* THE TEST WILL BE RUN WITH THE V.35 INTERFACE SELECTED.
* IF EXTERNAL TURNAROUND IS PROVIDED ON A PARTICULAR INTERFACE, THE
* TEST WILL BE RUN ON THAT INTERFACE. IF THERE IS NO EXTERNAL TURNAROUND, THE

```

```

2236      ;* TEST WILL NOT BE RUN.
2237      ;*   PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
2238      ;*                   375,373,367,357,337,277,177
2239      ;* 7-BIT CHARACTERS ARE USED. (HI BIT OF A PATTERN CHAR IS NOT USED).
2240      ;*****
2241      BGNTST
2242      035344      012737      000036      002434      MOV      #30.,TSTNUM      ;SET TEST NO.
2243      035344      012737      035526      002346      MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
2244      035360      004737      003276      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR
2245      035364      004737      011016      JSR      PC,CKLPBK      ;CHECK LOOPBACK, GET MODEM SELECTION
2246      035370      000000      0 MOV      0
2247      035372      013737      002422      035410      MOV      MODINT,6$      ;SET MODEM SELECTION
2248      035400      004737      010324      JSR      PC,SETUP      ;PROGRAM THE USYRT
2249      035404      000026      026
2250      035406      000211      CRC2,STRIP!DDCMP
2251      035410      000000      6$:      .WORD      0      ;MODEM SELECTION GOES HERE
2252      035412      000347      TXLEN2!TXLEN1!TXLENO!RXLEN2!RXLEN1!RXLENO
2253      035414      004737      010204      JSR      PC,LODATA      ;LOAD MSG INTO TX SILO AND RCV'D DATA BUF
2254      035420      142777      000010      145012      BICB      #LULOP,ABSEL1      ;CLEAR LULOP, CLOCK MSG
2255      035426      012737      000012      002364      MOV      #12,REGNUM      ;SET LU REG NO. = 12
2256      035434      012703      002762      MOV      #RCVBUF,R3      ;GET POINTER TO RCV MSG BUF
2257      035440      013702      002264      9$:      MOV      TCOUNT,R2      ;INIT TIMER
2258      035444      004737      003372      10$:      JSR      PC,READLU      ;READ REG 12
2259      035450      132737      000020      002350      BITB      #IRDY,REDBYT      ;SEE IF IRDY IS SET YET
2260      035456      001011      BNE      12$      ;BR IF YES
2261      035460      005202      INC      R2      ;INCREMENT TIMER
2262      035462      001370      BNE      10$      ;BR IF NO TIME-OUT YET
2263      035464      004737      004226      JSR      PC,GETALL      ;GET REGS FOR PRINTOUT
2264      ;REPORT IRDY NOT SET
2265      035470      104455      ERRDF      17,EM17,ERR7
2266      035470      000021      TRAP      C$ERDF
2267      035472      013475      .WORD      17
2268      035474      020302      .WORD      EM17
2269      035476      000412      .WORD      ERR7
2270      035500      112337      035514      12$:      BR      24$      ;ESCAPE TO END OF TEST
2271      035502      005203      INC      R3      ;GET AN EXPECTED DATA CHAR
2272      035506      004737      007314      JSR      PC,CKDATA      ;INCREMENT POINTER
2273      035510      100000      16$:      BCCCHK      0      ;COMPARE RCV'D DATA CHAR TO EXPECTED
2274      035514      000000      0
2275      035516      020327      003102      CMP      R3,#RCVBUF+80.      ;SEE IF ALL CHARS CHECKED YET
2276      035520      103745      BLO      9$      ;BR IF NOT YET
2277      035526      24$:
2278      035526      104401      ENDTST
2279      035526
2280      035526
2281      035526
2282      035526
2283      035526
2284      035526
2285      035526
2286      035526
2287      035526
2288      035526
2289      035526
2290      035526
2291      035526
2292      035526
2293      035526
2294      035526
2295      035526
2296      035526
2297      035526
2298      035526
2299      035526
2300      035526
2301      035526
2302      035526
2303      035526
2304      035526
2305      035526
2306      035526
2307      035526
2308      035526
2309      035526
2310      035526
2311      035526
2312      035526
2313      035526
2314      035526
2315      035526
2316      035526
2317      035526
2318      035526
2319      035526
2320      035526
2321      035526
2322      035526
2323      035526
2324      035526
2325      035526
2326      035526
2327      035526
2328      035526
2329      035526
2330      035526
2331      035526
2332      035526
2333      035526
2334      035526
2335      035526
2336      035526
2337      035526
2338      035526
2339      035526
2340      035526
2341      035526
2342      035526
2343      035526
2344      035526
2345      035526
2346      035526
2347      035526
2348      035526
2349      035526
2350      035526
2351      035526
2352      035526
2353      035526
2354      035526
2355      035526
2356      035526
2357      035526
2358      035526
2359      035526
2360      035526
2361      035526
2362      035526
2363      035526
2364      035526
2365      035526
2366      035526
2367      035526
2368      035526
2369      035526
2370      035526
2371      035526
2372      035526
2373      035526
2374      035526
2375      035526
2376      035526
2377      035526
2378      035526
2379      035526
2380      035526
2381      035526
2382      035526
2383      035526
2384      035526
2385      035526
2386      035526
2387      035526
2388      035526
2389      035526
2390      035526
2391      035526
2392      035526
2393      035526
2394      035526
2395      035526
2396      035526
2397      035526
2398      035526
2399      035526
2400      035526
2401      035526
2402      035526
2403      035526
2404      035526
2405      035526
2406      035526
2407      035526
2408      035526
2409      035526
2410      035526
2411      035526
2412      035526
2413      035526
2414      035526
2415      035526
2416      035526
2417      035526
2418      035526
2419      035526
2420      035526
2421      035526
2422      035526
2423      035526
2424      035526
2425      035526
2426      035526
2427      035526
2428      035526
2429      035526
2430      035526
2431      035526
2432      035526
2433      035526
2434      035526
2435      035526
2436      035526
2437      035526
2438      035526
2439      035526
2440      035526
2441      035526
2442      035526
2443      035526
2444      035526
2445      035526
2446      035526
2447      035526
2448      035526
2449      035526
2450      035526
2451      035526
2452      035526
2453      035526
2454      035526
2455      035526
2456      035526
2457      035526
2458      035526
2459      035526
2460      035526
2461      035526
2462      035526
2463      035526
2464      035526
2465      035526
2466      035526
2467      035526
2468      035526
2469      035526
2470      035526
2471      035526
2472      035526
2473      035526
2474      035526
2475      035526
2476      035526
2477      035526
2478      035526
2479      035526
2480      035526
2481      035526
2482      035526
2483      035526
2484      035526
2485      035526
2486      035526
2487      035526
2488      035526
2489      035526
2490      035526
2491      035526
2492      035526
2493      035526
2494      035526
2495      035526
2496      035526
2497      035526
2498      035526
2499      035526
2500      035526
2501      035526
2502      035526
2503      035526
2504      035526
2505      035526
2506      035526
2507      035526
2508      035526
2509      035526
2510      035526
2511      035526
2512      035526
2513      035526
2514      035526
2515      035526
2516      035526
2517      035526
2518      035526
2519      035526
2520      035526
2521      035526
2522      035526
2523      035526
2524      035526
2525      035526
2526      035526
2527      035526
2528      035526
2529      035526
2530      035526
2531      035526
2532      035526
2533      035526
2534      035526
2535      035526
2536      035526
2537      035526
2538      035526
2539      035526
2540      035526
2541      035526
2542      035526
2543      035526
2544      035526
2545      035526
2546      035526
2547      035526
2548      035526
2549      035526
2550      035526
2551      035526
2552      035526
2553      035526
2554      035526
2555      035526
2556      035526
2557      035526
2558      035526
2559      035526
2560      035526
2561      035526
2562      035526
2563      035526
2564      035526
2565      035526
2566      035526
2567      035526
2568      035526
2569      035526
2570      035526
2571      035526
2572      035526
2573      035526
2574      035526
2575      035526
2576      035526
2577      035526
2578      035526
2579      035526
2580      035526
2581      035526
2582      035526
2583      035526
2584      035526
2585      035526
2586      035526
2587      035526
2588      035526
2589      035526
2590      035526
2591      035526
2592      035526
2593      035526
2594      035526
2595      035526
2596      035526
2597      035526
2598      035526
2599      035526
2600      035526
2601      035526
2602      035526
2603      035526
2604      035526
2605      035526
2606      035526
2607      035526
2608      035526
2609      035526
2610      035526
2611      035526
2612      035526
2613      035526
2614      035526
2615      035526
2616      035526
2617      035526
2618      035526
2619      035526
2620      035526
2621      035526
2622      035526
2623      035526
2624      035526
2625      035526
2626      035526
2627      035526
2628      035526
2629      035526
2630      035526
2631      035526
2632      035526
2633      035526
2634      035526
2635      035526
2636      035526
2637      035526
2638      035526
2639      035526
2640      035526
2641      035526
2642      035526
2643      035526
2644      035526
2645      035526
2646      035526
2647      035526
2648      035526
2649      035526
2650      035526
2651      035526
2652      035526
2653      035526
2654      035526
2655      035526
2656      035526
2657      035526
2658      035526
2659      035526
2660      035526
2661      035526
2662      035526
2663      035526
2664      035526
2665      035526
2666      035526
2667      035526
2668      035526
2669      035526
2670      035526
2671      035526
2672      035526
2673      035526
2674      035526
2675      035526
2676      035526
2677      035526
2678      035526
2679      035526
2680      035526
2681      035526
2682      035526
2683      035526
2684      035526
2685      035526
2686      035526
2687      035526
2688      035526
2689      035526
2690      035526
2691      035526
2692      035526
2693      035526
2694      035526
2695      035526
2696      035526
2697      035526
2698      035526
2699      035526
2700      035526
2701      035526
2702      035526
2703      035526
2704      035526
2705      035526
2706      035526
2707      035526
2708      035526
2709      035526
2710      035526
2711      035526
2712      035526
2713      035526
2714      035526
2715      035526
2716      035526
2717      035526
2718      035526
2719      035526
2720      035526
2721      035526
2722      035526
2723      035526
2724      035526
2725      035526
2726      035526
2727      035526
2728      035526
2729      035526
2730      035526
2731      035526
2732      035526
2733      035526
2734      035526
2735      035526
2736      035526
2737      035526
2738      035526
2739      035526
2740      035526
2741      035526
2742      035526
2743      035526
2744      035526
2745      035526
2746      035526
2747      035526
2748      035526
2749      035526
2750      035526
2751      035526
2752      035526
2753      035526
2754      035526
2755      035526
2756      035526
2757      035526
2758      035526
2759      035526
2760      035526
2761      035526
2762      035526
2763      035526
2764      035526
2765      035526
2766      035526
2767      035526
2768      035526
2769      035526
2770      035526
2771      035526
2772      035526
2773      035526
2774      035526
2775      035526
2776      035526
2777      035526
2778      035526
2779      035526
2780      035526
2781      035526
2782      035526
2783      035526
2784      035526
2785      035526
2786      035526
2787      035526
2788      035526
2789      035526
2790      035526
2791      035526
2792      035526
2793      035526
2794      035526
2795      035526
2796      035526
2797      035526
2798      035526
2799      035526
2800      035526
2801      035526
2802      035526
2803      035526
2804      035526
2805      035526
2806      035526
2807      035526
2808      035526
2809      035526
2810      035526
2811      035526
2812      035526
2813      035526
2814      035526
2815      035526
2816      035526
2817      035526
2818      035526
2819      035526
2820      035526
2821      035526
2822      035526
2823      035526
2824      035526
2825      035526
2826      035526
2827      035526
2828      035526
2829      035526
2830      035526
2831      035526
2832      035526
2833      035526
2834      035526
2835      035526
2836      035526
2837      035526
2838      035526
2839      035526
2840      035526
2841      035526
2842      035526
2843      035526
2844      035526
2845      035526
2846      035526
2847      035526
2848      035526
2849      035526
2850      035526
2851      035526
2852      035526
2853      035526
2854      035526
2855      035526
2856      035526
2857      035526
2858      035526
2859      035526
2860      035526
2861      035526
2862      035526
2863      035526
2864      035526
2865      035526
2866      035526
2867      035526
2868      035526
2869      035526
2870      035526
2871      035526
2872      035526
2873      035526
2874      035526
2875      035526
2876      035526
2877      035526
2878      035526
2879      035526
2880      035526
2881      035526
2882      035526
2883      035526
2884      035526
2885      035526
2886      035526
2887      035526
2888      035526
2889      035526
2890      035526
2891      035526
2892      035526
2893      035526
2894      035526
2895      035526
2896      035526
2897      035526
2898      035526
2899      035526
2900      035526
2901      035526
2902      035526
2903      035526
2904      035526
2905      035526
2906      035526
2907      035526
2908      035526
2909      035526
2910      035526
2911      035526
2912      035526
2913      035526
2914      035526
2915      035526
2916      035526
2917      035526
2918      035526
2919      035526
2920      035526
2921      035526
2922      035526
2923      035526
2924      035526
2925      035526
2926      035526
2927      035526
2928      035526
2929      035526
2930      035526
2931      035526
2932      035526
2933      035526
2934      035526
2935      035526
2936      035526
2937      035526
2938      035526
2939      035526
2940      035526
2941      035526
2942      035526
2943      035526
2944      035526
2945      035526
2946      035526
2947      035526
2948      035526
2949      035526
2950      035526
2951      035526
2952      035526
2953      035526
2954      035526
2955      035526
2956      035526
2957      035526
2958      035526
2959      035526
2960      035526
2961      035526
2962      035526
2963      035526
2964      035526
2965      035526
2966      035526
2967      035526
2968      035526
2969      035526
2970      035526
2971      035526
2972      035526
2973      035526
2974      035526
2975      035526
2976      035526
2977      035526
2978      035526
2979      035526
2980      035526
2981      035526
2982      035526
2983      035526
2984      035526
2985      035526
2986      035526
2987      035526
2988      035526
2989      035526
2990      035526
2991      035526
2992      035526
2993      035526
2994      035526
2995      035526
2996      035526
2997      035526
2998      035526
2999      035526
3000      035526
3001      035526
3002      035526
3003      035526
3004      035526
3005      035526
3006      035526
3007      035526
3008      035526
3009      035526
3010      035526
3011      035526
3012      035526
3013      035526
3014      035526
3015      035526
3016      035526
3017      035526
3018      035526
3019      035526
3020      035526
3021      035526
3022      035526
3023      035526
3024      035526
3025      035526
3026      035526
3027      035526
3028      035526
3029      035526
3030      035526
3031      035526
3032      035526
3033      035526
3034      035526
3035      035526
3036      035526
3037      035526
3038      035526
3039      035526
3040      035526
3041      035526
3042      035526
3043      035526
3044      035526
3045      035526
3046      035526
3047      035526
3048      035526
3049      035526
3050      035526
3051      035526
3052      035526
3053      035526
3054      035526
3055      035526
3056      035526
3057      035526
3058      035526
3059      035526
3060      035526
3061      035526
3062      035526
3063      035526
3064      035526
3065      035526
3066      035526
3067      035526
3068      035526
3069      035526
3070      035526
3071      035526
3072      035526
3073      035526
3074      035526
3075      035526
3076      035526
3077      035526
3078      035526
3079      035526
3080      035526
3081      035526
3082      035526
3083      035526
3084      035526
3085      035526
3086      035526
3087      035526
3088      035526
3089      035526
3090     
```

```

2286      ;* FOR THIS LINE = 377. THE PROGRAM CHECKS FOR CORRECT RECEPTION OF
2287      ;* THE FIRST CHARACTER (STATION ADDRESS) AND THE REMAINING 4
2288      ;* ONES CHARACTERS (DATA). THIS TEST EXERCISES THE SECONDARY STATION
2289      ;* ADDRESS LOGIC, AND CHECKS THAT THE SEC. STA. ADRS. CAN BE BIT-STUFFED
2290      ;* AND TRANSMITTED AND RECEIVED CORRECTLY.
2291      ;*****
2292      BGNTST
2293      035530      012737      035634      002346      MOV      #24$,RETADR      ;SET TEST EXIT ADDRESS FOR ERRORS
2294      035530      004737      005240      JSR      PC,INITRN      ;MST CLR, LOAD 2 SOM'S
2295      035542      000377      377      ;
2296      035544      000320      CRC2!CRC1!SECA      ;BIT MODE, NO ERROR DET, SEC ADR MODE
2297      035546      004737      010746      JSR      PC,LODSIL      ;LOAD 5 377-CHARS INTO TX SILO
2298      035552      000377      377      ;
2299      035554      000005      5      JSR      PC,LODSIL      ;LOAD 2 EOM'S INTO TX SILO
2300      035556      004737      010746      TXEOM
2301      035562      001000      2      JSR      PC,STPLU      ;CLOCK MORE THAN ENTIRE MSG
2302      035564      000002      160.      JSR      PC,CKDATA      ;RCV SEC ADRS = 377
2303      035566      004737      004754      377
2304      035572      000240      0      MOV      #3,R1      ;RCV 3 MORE 377 CHARS
2305      035574      004737      007314      JSR      PC,CKDATA
2306      035600      000377      6$:      377
2307      035602      000000      0      DEC      R1
2308      035604      012701      000003      BNE      6$
2309      035610      004737      007314      JSR      PC,CKDATA      ;RCV LAST 377 CHAR, CHK EBLK = 1
2310      035614      000377      1377
2311      035616      000000      0      JSR      PC,MSTCLR      ;ISSUE MASTER CLEAR TO CLEAN UP
2312      035620      005301      24$:      0
2313      035622      001372      6$:      0
2314      035624      004737      007314      JSR      PC,CKDATA
2315      035630      001377      1377
2316      035632      000000      0
2317      035634      004737      003276      JSR      PC,MSTCLR
2318      035640      035640      104401      ENDTST
2319      035640
2320      035640
2321      035640
2322      035640
2323      035640
2324      035640
2325      035640
2326      035640
2327      035640
2328      035640
2329      035640
2330      035640
2331      035640
2332      035640
2333      035640
2334      035640
2335      035640
2336      035640
2337      035640
2338      035640      012737      036420      002346      MOV      #24$,RETADR      ;SET TEST EXIT ADRS FOR ERRORS
  
```

```

;*****
;SBTTL      TEST 32 - DDCMP MESSAGE TEST - CHAR MODE
;
;* IN THIS TEST, THREE USYRT MESSAGES ARE SENT TO SIMULATE A DDCMP HEADER,
;* DDCMP DATA MESSAGE, AND THE START OF A NEW DDCMP HEADER.
;* FIRST, THE DATA IN PATTERN A IS TRANSMITTED AND RECEIVED
;* AND THEN CRC (CRC-16) IS SENT, FOLLOWED BY THE DATA IN PATTERN A
;* AGAIN AND THE CRC ON THAT DATA, AND FINALLY THE DATA IN 'MSG1' IS
;* SENT WITH ITS CORRESPONDING CRC.
;* PATTERN A = 125,252,000,377,001,002,004,010,020,040,100,200,376,
;*             375,373,367,357,337,277,177
;* MSG1 = SYNCH,SYNCH,SYNCH,SYNCH,000,125,252,377,000,SYNCH,SYNCH
;*****
BGNTST
  
```

T32::

```

2339
2340 -----
2341 TRANSMIT AND RCV ENTIRE MSG
2342 035650 004737 005240 JSR PC,INITRN ;MST CLR, LOAD 2 SOM'S
2343 035654 000226 SYNCH
2344 035656 000011 STRIP!DDCMP
2345 035660 004737 010574 JSR PC,LDBYTS ;LOAD 20 WORDS OF PAT A INTO TX SILO
2346 035664 002557 PATA
2347 035666 000024 20.
2348 035670 004737 010746 JSR PC,LODSIL ;LOAD AN EOM INTO TX SILO
2349 035674 001000 TXEOM
2350 035676 000001 1
2351 035700 004737 010574 JSR PC,LDBYTS ;LOAD 20 WORDS OF PAT A INTO TX SILO
2352 035704 002557 PATA
2353 035706 000024 20.
2354 035710 004737 010746 JSR PC,LODSIL ;LOAD 1 EOM INTO TX SILO
2355 035714 001000 TXEOM
2356 035716 000001 1
2357 035720 004737 010746 JSR PC,LODSIL ;LOAD 3 SOM'S INTO TX SILO
2358 035724 000400 TXSOM
2359 035726 000003 3
2360 035730 004737 010516 JSR PC,LODMSG ;LOAD MSG1 INTO TX SILO
2361 035734 002670 MSG1
2362 035736 000013 11.
2363 035740 004737 004754 JSR PC,STPLU ;CLOCK HDR MSG AND CRC CHARS
2364 035744 000300 192.
2365 035746 012737 000013 002364 MOV #13,REGNUM ;SET REG. NO. = 13
2366 035754 004737 003372 JSR PC,READLU ;READ REG 13
2367 035760 032737 000040 002350 BIT #RTS,REDBYT ;SEE IF RTS SET
2368 035766 001010 BNE 2$ ;BR IF RTS SET
2369 035770 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
2370 ;REPORT RTS NOT SET
2371 035774 ERRDF 60,EM60,ERR7
2372 035774 104455 TRAP CSERDF
2373 035776 000074 .WORD 60
2374 036000 014740 .WORD EM60
2375 036002 020302 .WORD ERR7
2376 036004 000137 036420 JMP 24$ ;EXIT TEST
2377 036010 004737 004754 2$: JSR PC,STPLU ;CLK DATA MSG AND FIRST CRC CHAR
2378 036014 000250 168.
2379 036016 012703 000040 MOV #32,R3 ;SET COUNTER FOR CHECKING RTS
2380 036022 004737 004754 4$: JSR PC,STPLU ;CLOCK LINE UNIT FOR 1 CYCLE
2381 036026 000001 1
2382 036030 004737 003372 JSR PC,READLU ;READ REG 13
2383 036034 032737 000040 002350 BIT #RTS,REDBYT ;CHK FOR RTS SET
2384 036042 001007 BNE 5$ ;BR IF RTS SET
2385 036044 004737 004226 JSR PC,GETALL ;GET REGS FOR PRINTOUT
2386 ;REPORT RTS NOT SET
2387 ERRDF 60,EM60,ERR7
2388 036050 104455 TRAP CSERDF
2389 036052 000074 .WORD 60
2390 036054 014740 .WORD EM60
2391 036056 020302 .WORD ERR7
2392 036060 000557
2393 036062 005303 5$: BR 24$
2394 036064 001356 DEC R3 ;DECR COUNTER
2395 BNE 4$ ;BR IF NOT DONE YET
2396 -----
  
```

```

2388 ; READ AND CHK HEADER AND CRC
2389 -----
2390 036066 012701 002557      MOV    #PATA,R1      ;INIT PATTERN A POINTER
2391 036072 112137 036102    7$:  MOV    (R1)+,8$      ;GET AN EXPECTED CHAR
2392 036076 004737 007314    JSR     PC,CKDATA      ;READ AND CHK A CHAR
2393 036102 000000           8$:  .WORD    0
2394 036104 000000           0
2395 036106 020127 002601    CMP     R1,#PATB-2      ;SEE IF CHKING NEXT-TO-LAST CHAR YET
2396 036112 103767           BLO     7$              ;BR IF NOT YET
2397 036114 004737 007314    JSR     PC,CKDATA      ;READ AND CHK CHAR, BCC=0
2398 036120 100277           CRCCHK!277
2399 036122 000000           0
2400 036124 004737 007314    JSR     PC,CKDATA      ;READ AND CHK LAST CHAR, BCC=1
2401 036130 100577           CRCCHK!RXBCC!177
2402 036132 000000           0
2403 036134 004737 007314    JSR     PC,CKDATA      ;READ AND CHK HI LRC BYTE
2404 036140 000156           156
2405 036142 000000           0
2406 036144 004737 007314    JSR     PC,CKDATA      ;READ AND CHK LO CRC BYTE
2407 036150 000236           236
2408 036152 000000           0
2409 -----
2410 ; READ AND CHK DATA MSG AND CRC
2411 -----
2412 036154 012701 002557      MOV    #PATA,R1      ;INIT PATTERN A POINTER
2413 036160 112137 036170    9$:  MOV    (R1)+,12$     ;GET AN EXPECTED CHAR
2414 036164 004737 007314    JSR     PC,CKDATA      ;READ AND CHK A CHAR
2415 036170 000000           12$: .WORD    0
2416 036172 000000           0
2417 036174 020127 002601    CMP     R1,#PATB-2      ;SEE IF CHKING NEXT-TO-LAST CHAR YET
2418 036200 103767           BLO     9$              ;BR IF NOT YET
2419 036202 004737 007314    JSR     PC,CKDATA      ;READ AND CHK CHAR, BCC=0
2420 036206 100277           CRCCHK!277
2421 036210 000000           0
2422 036212 004737 007314    JSR     PC,CKDATA      ;READ AND CHK LAST CHAR, BCC=1
2423 036216 100577           CRCCHK!RXBCC!177
2424 036220 000000           0
2425 036222 004737 007314    JSR     PC,CKDATA      ;READ AND CHK HI CRC BYTE
2426 036226 000156           156
2427 036230 000000           0
2428 036232 004737 007314    JSR     PC,CKDATA      ;READ AND CHK LO CRC BYTE
2429 036236 000236           236
2430 036240 000000           0
2431 -----
2432 ; CLOCK 3RD MESSAGE ('MSG1' DATA)
2433 -----
2434 036242 012737 000012 002364  MOV    #12,REGNUM      ;SET REG NO. = 12
2435 036250 112737 000200 002352  MOV    #1C,WRIBYT      ;SET 1C TO CLEAR RECEIVER FOR NEW MSG
2436 036256 004737 003450           JSR     PC,WRITLU
2437 036262 012737 000013 002364  MOV    #13,REGNUM      ;RESTORE REG NO. TO 13
2438 036270 004737 004754           JSR     PC,STPLU      ;CLOCK THE REST OF MSG
2439 036274 000150           104.
2440 036276 004737 003372           JSR     PC,READLU      ;READ REG 13
2441 036302 032737 000040 002350  BIT     #RTS,REDBYT      ;SEE IF RTS IS CLEARED
2442 036310 001407           BEQ     14$              ;BR IF RTS CLEARED
2443 036312 004737 004226           JSR     PC,GETALL      ;GET REGS FOR PRINTOUT
2444 ;REPORT RTS NOT CLEARED

```

2445	036316		ERRDF	65,EM65,ERR7	
	036316	104455			TRAP C\$ERDF
	036320	000101			.WORD 65
	036322	015036			.WORD EM65
	036324	020302			.WORD ERR7
2446	036326	000434			
2447			BR	248	
2448			-----		
2449			; READ AND CHECK 3RD MESSAGE AND CRC		
2450	036330	004737	007314	148:	JSR PC,CKDATA ; READ AND CHECK 000 DATA CHAR
2451	036334	000000		000	
2452	036336	000000		0	
2453	036340	004737	007314	JSR PC,CKDATA ; READ AND CHECK 125 DATA CHAR	
2454	036344	000125		125	
2455	036346	000000		0	
2456	036350	004737	007314	JSR PC,CKDATA ; READ AND CHECK 252 DATA CHAR	
2457	036354	000252		252	
2458	036356	000000		0	
2459	036360	004737	007314	JSR PC,CKDATA ; READ AND CHECK 377 DATA CHAR, AND BCC=0	
2460	036364	100377		CRCCHK.377	
2461	036366	000000		0	
2462	036370	004737	007314	JSR PC,CKDATA ; READ AND CHECK 000 DATA CHAR, AND BCC=1	
2463	036374	100400		CRCCHK!RXBCC.000	
2464	036376	000000		0	
2465	036400	004737	007314	JSR PC,CKDATA ; READ AND CHK HI CRC BYTE	
2466	036404	000160		160	
2467	036406	000000		0	
2468	036410	004737	007314	JSR PC,CKDATA ; READ AND CHK LO CRC BYTE	
2469	036414	000034		034	
2470	036416	000000		0	
2471	036420	004737	003276	JSR PC,MSTCLR ; ISSUE MASTER CLEAR TO CLEAN UP	
2472	036424				
	036424				
	036424	104401			L10106: TRAP C\$ETST
2473					
2474					
2475					
2476					
2477					

```

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     036426      BGNHRD
15     036426      000046
16     036430      .WORD L10107-L$HARD/2
17     036430      L$HARD::
18     036432      GPRMA  ADDRES,2,0,160000,177776,YES
19     036434      .WORD  T$CODE
20     036436      .WORD  ADDRES
21     036440      .WORD  T$LOLIM
22     036442      .WORD  T$HILIM
23     036444      GPRMA  VECTOR,4,0,0,674,YES
24     036446      .WORD  T$CODE
25     036448      .WORD  VECTOR
26     036450      .WORD  T$LOLIM
27     036452      .WORD  T$HILIM
28     036454      GPRMD  PRIRTY,6,0,7000,4,7,YES
29     036456      .WORD  T$CODE
30     036458      .WORD  PRIRTY
31     036460      .WORD  7000
32     036462      .WORD  T$LOLIM
33     036464      .WORD  T$HILIM
34     036466      GPRMD  SWPAC1,12,0,377,0,056,YES
35     036468      .WORD  T$CODE
36     036470      .WORD  SWPAC1
37     036472      .WORD  377
38     036474      .WORD  T$LOLIM
39     036476      .WORD  T$HILIM
40     036478      GPRMD  SWPAC2,14,0,377,0,377,YES
41     036480      .WORD  T$CODE
42     036482      .WORD  SWPAC2
43     036484      .WORD  377
44     036486      .WORD  T$LOLIM
45     036488      .WORD  T$HILIM
46     036490      GPRMD  SWPAC3,16,0,377,0,377,YES
47     036492      .WORD  T$CODE
48     036494      .WORD  SWPAC3
49     036496      .WORD  377
50     036498      .WORD  T$LOLIM
51     036500      .WORD  T$HILIM
52     036502      GPRMD  LOOPBK,20,0,7,0,4,YES
53     036504      .WORD  T$CODE
54     036506      .WORD  LOOPBK
55     036508      .WORD  7
56     036510      .WORD  T$LOLIM
57     036512      .WORD  T$HILIM
58     036514
59     036516
60     036520
61     036522
62     036524
63     036526
64     036530

```

23 036532 GPRMD BAUDRT,22,0,7,0,7,YES
 036532 011032
 036534 037137
 036536 000007
 036540 000000
 036542 000007

.WORD T\$CODE
 .WORD BAUDRT
 .WORD 7
 .WORD T\$LOLIM
 .WORD T\$HILIM

24
 25 036544 ENDHRD

.EVEN

L10107:

26
 27 036544 104 105 126 ADDR: .ASCIIZ /DEVICE CSR ADDRESS : /
 036547 111 103 105
 036552 040 103 123
 036555 122 040 101
 036560 104 104 122
 036563 105 123 123
 036566 040 072 040
 036571 000

28 036572 104 105 126 VECTOR: .ASCIIZ /DEVICE VECTOR ADDRESS : /
 036575 111 103 105
 036600 040 126 105
 036603 103 124 117
 036606 122 040 101
 036611 104 104 122
 036614 105 123 123
 036617 040 072 040
 036622 000

29 036623 104 105 126 PRIRTY: .ASCIIZ /DEVICE PRIORITY LEVEL : /
 036626 111 103 105
 036631 040 120 122
 036634 111 117 122
 036637 111 124 131
 036642 040 114 105
 036645 126 105 114
 036650 040 072 040
 036653 000

30 036654 115 070 062 SWPAC1: .ASCIIZ /M8203 REG 11 (E121 SW10,9 , E134 SW9,10) : /
 036657 060 063 040
 036662 122 105 107
 036665 040 061 061
 036670 040 050 105
 036673 061 062 061
 036676 040 123 127
 036701 061 060 054
 036704 071 040 054
 036707 040 105 061
 036712 063 064 040
 036715 123 127 071
 036720 054 061 060
 036723 051 040 072
 036726 040 000

31 036730 115 070 062 SWPAC2: .ASCIIZ /M8203 REG 15 (E134 SW1-8) : /
 036733 060 063 040
 036736 122 105 107
 036741 040 061 065
 036744 040 050 105

	036747	061	063	064	
	036752	040	123	127	
	036755	061	055	070	
	036760	051	040	072	
	036763	040	000		
32	036765	115	070	062	SWPAC3: .ASCII /MB203 REG 16 (E121 SW1-8) : /
	036770	060	063	040	
	036773	122	105	107	
	036776	040	061	066	
	037001	040	050	105	
	037004	061	062	061	
	037007	040	123	127	
	037012	061	055	070	
	037015	051	040	072	
	037020	040	000		
33	037022	124	125	112	LOOPBK: .ASCII /TURNAROUND TYPE -/<15><12>
	037025	116	101	122	
	037030	117	125	116	
	037033	104	040	124	
	037036	131	120	105	
	037041	040	055	015	
	037044	012			
34	037045	050	060	075	.ASCII /(0=H3254&H3255, 1=CABLE, 2=MOD LOC, 3=MOD REM, 4=NONE) : /
	037050	110	063	062	
	037053	065	064	046	
	037056	110	063	062	
	037061	065	065	054	
	037064	040	061	075	
	037067	103	101	102	
	037072	114	105	054	
	037075	040	062	075	
	037100	115	117	104	
	037103	040	114	117	
	037106	103	054	040	
	037111	063	075	115	
	037114	117	104	040	
	037117	122	105	115	
	037122	054	040	064	
	037125	075	116	117	
	037130	116	105	051	
	037133	040	072	040	
	037136	000			
35	037137	120	114	105	BAUDRT: .ASCII /PLEASE SELECT BAUD RATE; TYPE '0' FOR 2.4K; '1' FOR 4.8K;/<15><12>
	037142	101	123	105	
	037145	040	123	105	
	037150	114	105	103	
	037153	124	040	102	
	037156	101	125	104	
	037161	040	122	101	
	037164	124	105	073	
	037167	040	124	131	
	037172	120	105	040	
	037175	047	060	047	
	037200	040	106	117	
	037203	122	040	062	
	037206	056	064	113	
	037211	073	040	047	

	037214	061	047	040
	037217	106	117	122
	037222	040	064	056
	037225	070	113	073
	037230	015	012	
36	037232	047	062	047
	037235	040	106	117
	037240	122	040	071
	037243	056	066	113
	037246	073	040	047
	037251	063	047	040
	037254	106	117	122
	037257	040	061	071
	037262	056	062	113
	037265	073	040	047
	037270	064	047	040
	037273	106	117	122
	037276	040	065	066
	037301	113	073	040
	037304	047	065	047
	037307	040	106	117
	037312	122	040	062
	037315	065	060	113
	037320	073	015	012
37	037323	047	066	047
	037326	040	106	117
	037331	122	040	065
	037334	060	060	113
	037337	073	040	117
	037342	122	040	047
	037345	067	047	040
	037350	106	117	122
	037353	040	061	040
	037356	115	105	107
	037361	040	102	101
	037364	125	104	040
	037367	072	040	000

.ASCII /'2' FOR 9.6K; '3' FOR 19.2K; '4' FOR 56K; '5' FOR 250K; /<15><12>

.ASCII /'6' FOR 500K; OR '7' FOR 1 MEG BAUD : /

.EVEN

38
 39
 40
 41
 42
 43
 44

```

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 037372          BGNSFT
14 037372          000016
15 037374          GPRML  ISMANI,0,1,YES
16 037402          GPRML  ISPRNT,2,1,YES
17 037410          GPRML  ISWPAK,4,1,YES
18 037416          GPRMD  TIMCNT,6,0,177777,0,177777,YES
19
20 037430          ENDSFT
21
22 037430          L10110: .EVEN
23 037430          L10110: .WORD L10110-L$SOFT/2
24 037433          .WORD T$CODE
25 037436          .WORD ISMANI
26 037441          .WORD 1
27 037444          .WORD T$CODE
28 037447          .WORD ISPRNT
29 037452          .WORD 1
30 037455          .WORD T$CODE
31 037460          .WORD ISWPAK
32 037463          .WORD 1
33 037466          .WORD TIMCNT
34 037471          .WORD 177777
35 037474          .WORD T$LOLIM
36 037477          .WORD T$HILIM
37 037502
38 037505
39 037510
40 037513
41 111
42 115
43 056
44 116
45 122
46 116
47 104
48 111
49 104
50 117
51 117
52 124
53 105
54 040
55 116
56 103
57 122
58 122
59 103
60 122
61 051
62 123
63 101
64 040
65 124
66 126
67 056
68 105
69 122
70 105
71 040
72 115
73 125
74 040
75 124
76 103
77 116
78 116
79 124
80 050
81 040
82 040
83 116
84 117
85 123
86 105
87 117
88 123
89 000
90
91 ISMANI: .ASCIZ  /IS MAN. INTERVEN. DESIRED TO MOUNT TEST CONNECTOR(S) /

```

SOFTWARE PARAMETER CODING SECTION

23	037516	123	110	117	ISPRNT: .ASCIIZ /SHOULD SWITCH PACK AND AX3-15 PRINTOUT BE ALLOWED /	
	037521	125	114	104		
	037524	040	123	127		
	037527	111	124	103		
	037532	110	040	120		
	037535	101	103	113		
	037540	040	101	116		
	037543	104	040	101		
	037546	130	063	055		
	037551	061	065	040		
	037554	120	122	111		
	037557	116	124	117		
	037562	125	124	040		
	037565	102	105	040		
	037570	101	114	114		
	037573	117	127	105		
	037576	104	040	000		
24	037601	123	110	117		ISWPAK: .ASCIIZ /SHOULD SWITCH PACK TESTS BE ALLOWED /
	037604	125	114	104		
	037607	040	123	127		
	037612	111	124	103		
	037615	110	040	120		
	037620	101	103	113		
	037623	040	124	105		
	037626	123	124	123		
	037631	040	102	105		
	037634	040	101	114		
	037637	114	117	127		
	037642	105	104	040		
	037645	000				
25	037646	115	123	107	TIMCNT: .ASCIIZ /MSG TIMER VALUE (0-177777), 0 = LONGEST TIME-OUT : /	
	037651	040	124	111		
	037654	115	105	122		
	037657	040	126	101		
	037662	114	125	105		
	037665	040	050	060		
	037670	055	061	067		
	037673	067	067	067		
	037676	067	051	054		
	037701	040	060	040		
	037704	075	040	114		
	037707	117	116	107		
	037712	105	123	124		
	037715	040	124	111		
	037720	115	105	055		
	037723	117	125	124		
	037726	040	072	040		
	037731	000				

.EVEN

26
27
28
29
30
31
32
33
34

```
35  
36 037732           ;***** PATCH AREA FOR DEBUG *****  
37           PATCH:  
38                .=.+100  
39                NOP  
40                NOP  
41                NOP  
42           ;*****  
43  
44  
45 040040           ENDMOD  
46  
47 040040           LASTAD  
  
          040040   000000  
          040042   000000  
          040044           L$LAST::  
48  
49           000001           .END
```

```
.EVEN  
.WORD 0  
.WORD 0
```

ABORT = 000004	BSEL4 = 002442	CSRDBU= 000007	EM40 = 014063	FMT18 = 013101
ADDRES = 036544	CARR = 000001	CSREFG= 000047	EM41 = 014077	FMT19 = 013143
ADR = 000020 G	CHKABT = 012156	CSRESE= 000033	EM42 = 014120	FMT2 = 012314
ANBITS = 002547	CHPCHK= 100000	CSREVI= 000003	EM43 = 014135	FMT25 = 013174
APA = 000200	CHPTYP = 002420	CSRFLA= 000021	EM44 = 014167	FMT26 = 013224
ASBC0 = 000020	CKDATA = 007314	CSRPT = 000025	EM45 = 014221	FMT27 = 023170
ASBC1 = 000040	CKLPBK = 011016	CSSEFG= 000046	EM46 = 014253	FMT3 = 012336
ASBC2 = 000100	CRCHK= 100000	CSSPRI= 000041	EM47 = 014306	FMT4 = 012400
ASSEMB= 000010	CRCTY0= 000001	CS\$VEC= 000037	EM48 = 014341	FMT5 = 012413
AXNUM = 002366	CRCTY1= 000002	CSTPRI= 000013	EM49 = 014374	FMT6 = 012443
AX0.15= 002306	CRCTY2= 000004	C32BCC= 000040	EM50 = 014433	FMT7 = 012476
AX0.16= 002310	CRC1 = 000100	C32ENB= 000004	EM51 = 014467	FMT8 = 012506
AX1 = 000001	CRC2 = 000200	DDC = 000100	EM52 = 014527	FMT9 = 012542
AX1.15= 002312	CS = 000004	DDCMP = 000001	EM53 = 014563	FRSPAS = 002400
AX1.16= 002314	CSAU = 000052	DEVMAP = 002426	EM54 = 014617	FRSTIM = 002376
AX2 = 000002	CSAUTO= 000061	DEVPTR = 002430	EM55 = 014641	FSAU = 000015
AX2.15= 002316	CSBRK = 000022	DFPTBL = 002226 G	EM56 = 014655	FSAUTO= 000020
AX2.16= 002320	CSBSEG= 000004	DH1 = 015077	EM57 = 014672	FSBGN = 000040
AX3.15= 002322	CSBSUB= 000002	DH2 = 015121	EM58 = 014707	FSCLEA= 000007
AX3.16= 002324	CSCEFG= 000045	DH3 = 015150	EM59 = 014723	FSDU = 000016
AX315U= 000372	CSCLCK= 000062	DH4 = 015206	EM60 = 014740	FSEND = 000041
A11 = 025512	CSCLEA= 000012	DH5 = 015250	EM61 = 014754	FSHARD= 000004
A12 = 034022	CSCLOS= 000035	DH6 = 015253	EM62 = 014767	FSHW = 000013
BADDAT = 002372	CSCLP1= 000006	DH7 = 015256	EM63 = 015004	FSINIT= 000006
BAUDRT = 037137	CS\$VEC= 000036	DH8 = 015310	EM64 = 015021	FSJMP = 000050
BCC = 000001	CS\$DCLN= 000044	DH9 = 015347	EM65 = 015036	F\$MOD = 000000
BCCCHK= 100000	CS\$DODU= 000051	DIAGMC= 000000	EM66 = 015056	F\$MSG = 000011
BDRATE = 002464	CS\$DRPT= 000024	DISILO = 002416	EM7 = 013343	F\$PROT= 000021
BIT0 = 000001 G	CS\$DU = 000053	DISSI = 000040	EM8 = 013360	F\$PWR = 000017
BIT00 = 000001 G	CS\$EDIT= 000003	DTR = 000100	EM9 = 013401	F\$RPT = 000012
BIT01 = 000002 G	CS\$ERDF= 000055	EBLK = 000002	ENAX = 000004	F\$SEG = 000003
BIT02 = 000004 G	CS\$ERHR= 000056	EF.CON= 000036 G	ENDIT = 022576	F\$SOFT= 000005
BIT03 = 000010 G	CS\$ERRO= 000060	EF.NEW= 000035 G	ENDPAT = 002667	F\$SRV = 000010
BIT04 = 000020 G	CS\$ERSF= 000054	EF.PWR= 000034 G	EOM = 000002	F\$SUB = 000002
BIT05 = 000040 G	CS\$ERSO= 000057	EF.RES= 000037 G	ERRFLG = 002342	F\$SW = 000014
BIT06 = 000100 G	CS\$ESCA= 000010	EF.STA= 000040 G	ERROR1 = 002410	F\$TEST= 000001
BIT07 = 000200 G	CS\$ESEG= 000005	EM10 = 013416	ERR1 = 015412 G	GETALL = 004226
BIT08 = 000400 G	CS\$ESUB= 000003	EM11 = 013437	ERR10 = 021442 G	GETPRM = 022300
BIT09 = 001000 G	CS\$ETST= 000001	EM12 = 013454	ERR2 = 015444 G	GETREG = 003516
BIT1 = 000002 G	CS\$EXIT= 000032	EM17 = 013475	ERR3 = 015752 G	GOAH = 000010
BIT10 = 002000 G	CS\$GETB= 000026	EM18 = 013512	ERR4 = 016434 G	GOODAT = 002370
BIT11 = 004000 G	CS\$GETW= 000027	EM19 = 013533	ERR5 = 017112 G	G\$CNT0= 000200
BIT12 = 010000 G	CS\$GMAN= 000043	EM2 = 013265	ERR6 = 017624 G	G\$DELM= 000372
BIT13 = 020000 G	CS\$GPHR= 000042	EM20 = 013550	ERR7 = 020302 G	G\$DISP= 000003
BIT14 = 040000 G	CS\$GPLO= 000030	EM21 = 013571	ERR8 = 020734 G	G\$EXCP= 000400
BIT15 = 100000 G	CS\$GPRI= 000040	EM22 = 013606	EVL = 000004 G	G\$HILI= 000002
BIT2 = 000004 G	CS\$INIT= 000011	EM28 = 013627	E\$END = 002100	G\$LOLI= 000001
BIT3 = 000010 G	CS\$INLP= 000020	EM29 = 013650	E\$LOAD= 000035	G\$NO = 000000
BIT4 = 000020 G	CS\$MANI= 000050	EM3 = 013324	FMT1 = 012304	G\$OFFS= 000400
BIT5 = 000040 G	CS\$MEM = 000031	EM30 = 013665	FMT10 = 012547	G\$OFFSI= 000376
BIT6 = 000100 G	CS\$MSG = 000023	EM31 = 013706	FMT11 = 012600	G\$PRMA= 000001
BIT7 = 000200 G	CS\$OPEN= 000034	EM34 = 013723	FMT12 = 012637	G\$PRMD= 000002
BIT8 = 000400 G	CS\$PNTB= 000014	EM35 = 013751	FMT13 = 012722	G\$PRML= 000000
BIT9 = 001000 G	CS\$PNTF= 000017	EM36 = 013771	FMT14 = 012766	G\$RADA= 000140
BOE = 000400 G	CS\$PNTS= 000016	EM37 = 014005	FMT15 = 013032	G\$RADB= 000000
BPOLL = 000100	CS\$PNTX= 000015	EM38 = 014026	FMT16 = 022600	G\$RADD= 000040
BSEL1 = 002440	CS\$QIO = 000377	EM39 = 014043	FMT17 = 022722	G\$RADL= 000120

GSRADO= 000020	LOT = 000010 G	LSREPP 002062 G	L10056 031646	OSDU = 000001
GSXFER= 000004	LULOP= 000010	LSREV 002010 G	L10057 032030	OSERRT= 000000
GYES = 000010	LULP = 000040	LSRPT 022074 G	L10060 032212	OSGNSW= 000001
HDX = 000020	LUREG 002266	LSSOFT 037374 G	L10061 032310	OSPOIN= 000001
HELP = 000001	LUR10 = 002266	LSSPC 002056 G	L10062 032520	OSSETU= 000000
HOE = 100000 G	LUR11 = 002270	LSSPCP 002020 G	L10063 034026	PATA 002557
IAC = 000100	LUR12 = 002272	LSSPTP 002024 G	L10064 033016	PATB 002603
IACTIV 006260	LUR13 = 002274	LSSTA 002030 G	L10065 033110	PATCH 037732
IBE = 010000 G	LUR14 = 002276	LSSW 002256 G	L10066 033176	PATQ 002613
IC = 000200	LUR15 = 002300	LSTEST 002114 G	L10067 033226	PATR 002623
ICIR = 000010	LUR16 = 002302	LSTIML 002014 G	L10070 033344	PATS 002642
IDL = 000010	LUR17 = 002304	LSUNIT 002012 G	L10071 033432	PATT 002662
IDLE = 000040	LUSW11 002454	L10000 002254	L10072 033512	PNT = 001000 G
IDU = 000040 G	LUSW12 002456	L10001 002266	L10073 033606	POLL = 000200
IER = 020000 G	LUSW13 002460	L10002 015442	L10074 033666	PRI = 002000 G
IERR = 000002	LU2MOD 002000 G	L10003 015750	L10075 034020	PRIOR 002334
INITRN 005240	LSACP 002110 G	L10004 016432	L10076 034216	PRIPTY 036623
INTFLG 002340	LSAPT 002036 G	L10005 017110	L10077 034406	PRI00 = 000000 G
INTGRL= 000010	LSAU 023222 G	L10006 017622	L10100 034576	PRI01 = 000040 G
IRDY = 000020	LSAUT 002070 G	L10007 020300	L10101 034766	PRI02 = 000100 G
ISIRDY 005774	LSAUTO 023056 G	L10010 020732	L10102 035156	PRI03 = 000140 G
ISMANI 037430	LSCCP 002106 G	L10011 021440	L10103 035342	PRI04 = 000200 G
ISPRNT 037516	LSCLEA 023136 G	L10012 022072	L10104 035526	PRI05 = 000240 G
ISR = 000100 G	LSCO 002032 G	L10013 022074	L10105 035640	PRI06 = 000300 G
ISWPAK 037601	LSDEPO 002011 G	L10015 022576	L10106 036424	PRI07 = 000340 G
IXE = 004000 G	LSDESC 003170 G	L10016 023134	L10107 036544	PRNFLAG 002260
ISAU = 000041	LSDESP 002076 G	L10017 023136	L10110 037430	PSTACK 002332
ISAUTO= 000041	LSDEVP 002060 G	L10020 023166	MAINT1= 000010	RAB = 000004
ISCLN = 000041	LSDISP 002124 G	L10021 023222	MAINT2= 000004	RABT = 000004
ISDU = 000041	LSDLY 002116 G	L10022 023352	MCLK = 000002	RAX15 002354
ISHRD = 000041	LSDTP 002040 G	L10023 023726	MCLR = 000100	RAX16 002356
ISINIT= 000041	LSDTYP 002034 G	L10024 024264	MIFLAG 002256	RCVBUF 002762
ISMOD = 000041	LSDU 023140 G	L10025 024346	MODINT 002422	RCV1ST 007000
ISMSG = 000041	LSDUT 002072 G	L10026 024436	MODR = 000010	RDALL = 000004
ISPROT= 000040	LSDVTY 003162 G	L10027 024632	MPCSR 002436	RDAX = 000020
ISPTAB= 000041	LSEF 002052 G	L10030 025402	MPIVEC 002446	RDRXSI 006720
ISPR = 000041	LSENVI 002044 G	L10031 025642	MPOVEC 002450	READAX 003624
ISRPT = 000041	LSETP 002102 G	L10032 025504	MPRIOR 002452	READLU 003372
ISSEG = 000041	LSEXP1 002046 G	L10033 025624	MSG1 002670	READY = 000200
ISSETU= 000041	LSEXP4 002064 G	L10034 026046	MSG2 002716	REDBYT 002350
ISSFT = 000041	LSEXP5 002066 G	L10035 025744	MSG3 002732	REDDAT 002466
ISSRV = 000041	LSHARD 036430 G	L10036 026044	MSTCLR 003276	REGNUM 002364
ISSUB = 000041	LSHIME 002120 G	L10037 026202	MVIOX = 021000	REG0 002476
ISTST = 000041	LSHPCP 002016 G	L10040 026654	MVIXO = 122000	REG1 002500
I422 = 000200	LSHPTP 002022 G	L10041 026366	NEWST 022254	REG2 002502
JSJMP = 000167	LSHW 002226 G	L10042 026520	OACT = 000100	REG3 002504
LDBYTS 010574	LSICP 002104 G	L10043 026652	OACTIV 005052	REG4 002506
LDMSG1 010656	LSINIT 022104 G	L10044 026770	OC = 000200	REG5 002510
LDXSI 004674	LSLADP 002026 G	L10045 027326	OCOR = 000020	REG6 002512
LOADAT 002374	LSLAST 040044 G	L10046 027736	OP = 000002	REG7 002514
LODATA 010204	LSLOAD 002100 G	L10047 030410	ORDY = 000020	REOM = 000002
LODMSG 010516	LSLUN 002074 G	L10050 030160	OSIRDY 004362	RERR = 000200
LODSIL 010746	LSMREV 002050 G	L10051 030402	OVRR = 000010	RETADR 002346
LOE = 040000 G	LSNAME 002000 G	L10052 031112	OSAPTS= 000000	RING = 000200
LOGDEV 002330	LSPRIO 002042 G	L10053 030646	OSAU = 000001	ROMI = 000002
LOOPBK 037022	LSPROT 022076 G	L10054 031104	OSBGNR= 000000	ROMO = 000004
LOOPIN 003574	LSPT 002112 G	L10055 031374	OSBGNS= 000001	ROR = 000010

CZDMSD M8203 STATIC TESTS #2
SYMBOL TABLE

MACRO V03.01 7-AUG-80 11:18:21 PAGE 28-5

L 14

SEQ 180

RRDYTO= 000001	STR = 000040	TXAB = 000004	TSTSTM= 177777	T23.6 033346
RSEOM 006446	STRIP = 000010	TXABT = 002000	TSTSTS= 000001	T23.7 033434
RSOM = 000001	SUBRPC 002336	TXCHAR 005622	TSSAU = 010021	T23.8 033514
RTS = 000040	SVCGBL= 000000	TXDATA= 000040	TSSAUT= 010016	T23.9 033610
RUN = 000200	SVCINS= 000001	TXEN = 000100	TSSCLE= 010017	T24 034030 G
RXABT = 002000	SVCSUB= 000001	TXEOM = 001000	TSSDU = 010020	T25 034220 G
RXBCC = 000400	SVCTAG= 000001	TXGA = 000010	TSSHAR= 010107	T26 034410 G
RXEBL = 001000	SVCTST= 000001	TXGOA = 004000	TSSHW = 010000	T27 034600 G
RXLENO= 000001	SWIFLG 002262	TXLENO= 000040	TSSINI= 010015	T28 034770 G
RXLEN1= 000002	SWPAC1 036654	TXLEN1= 000100	TSSMSG= 010012	T29 035160 G
RXLEN2= 000004	SWPAC2 036730	TXLEN2= 000200	TSSPRO= 010014	T3 023730 G
RXOVR = 004000	SWPAC3 036765	TXSOM = 000400	TSSRPT= 010013	T30 035344 G
RXWORD 002414	SW0 = 000002	TXWORD 002412	TSSSOF= 010110	T31 035530 G
RX0 = 000001	SW1 = 000004	TX0 = 000001	TSSSUB= 010075	T32 035642 G
RX1 = 000002	SW2 = 000010	TX1 = 000002	TSSSW = 010001	T4 024266 G
RX2 = 000004	SW3 = 000040	TX2 = 000004	TSSTES= 010106	T5 024350 G
RX3 = 000010	SYNCH = 000226	TX3 = 000010	T1 023224 G	T6 024440 G
RX4 = 000020	SYNO = 000001	TX4 = 000020	T10 026050 G	T7 024634 G
RX5 = 000040	SYN1 = 000002	TX5 = 000040	T11 026204 G	T8 025404 G
RX6 = 000100	SYN2 = 000004	TX6 = 000100	T11.1 026204	T8.1 025404
RX7 = 000200	SYN3 = 000010	TX7 = 000200	T11.2 026370	T8.2 025512
R14NRW 002546	SYN4 = 000020	TYPEY 023011	T11.3 026522	T9 025644 G
SAVE4 002404	SYN5 = 000040	T\$ARGC= 000002	T12 026656 G	T9.1 025644
SAVE6 002406	SYN6 = 000100	T\$CODE= 003032	T13 026772 G	T9.2 025746
SAVLEN 002424	SYN7 = 000200	T\$ERRN= 000101	T14 027330 G	UAM = 000200 G
SCRACH 002326	SSLSYM= 010000	T\$EXCP= 000000	T15 027740 G	UNIT 002432
SEC = 000020	TCCHEK= 100000	T\$FLAG= 000040	T15.1 027754	UNRR = 000001
SECA = 000020	TCOUNT 002264	T\$GMAN= 000000	T15.2 030176	UPBITS 002536
SELFR = 000040	TEOM = 000002	T\$HILI= 177777	T16 030412 G	VECTOR 036572
SELSBY= 000002	TERR = 000200	T\$LAST= 000001	T16.1 030426	V35 = 000020
SEL4 002442	TEST = 000001	T\$LOLI= 000000	T16.2 030664	WAIT50 004646
SEL6 002444	TESTMD= 000004	T\$LSYM= 010000	T17 031114 G	WAX = 000010
SETUP 010324	TIMCNT 037646	T\$LTNO= 000040	T18 031376 G	WAX15 002360
SFPTBL 002256 G	TIMFLG 002344	T\$NEST= 177777	T19 031650 G	WAX16 002362
SIG0 = 000100	TMP0 002516	T\$NS0 = 000000	T2 023354 G	WRDYTO= 000002
SIGR = 000200	TMP1 002520	T\$NS1 = 000005	T20 032032 G	WRIBYT 002352
SOM = 000001	TMP2 002522	T\$NS2 = 000002	T21 032214 G	WRITAX 004012
STALL 004664	TMP3 002524	T\$PTNU= 000000	T22 032312 G	WRITLU 003450
STARES 002402	TMP4 002526	T\$SAVL= 177777	T23 032522 G	XYZ = 000100
STARST 022244	TMP5 002530	T\$SEGL= 177777	T23.1 032544	X\$ALWA= 000000
STBY = 000002	TMP6 002532	T\$SUBN= 000000	T23.10 033670	X\$FALS= 000040
STEPLU= 000020	TMP7 002534	T\$TAGL= 177777	T23.2 033020	X\$OFFS= 000400
STEPMP= 000001	TSOM = 000001	T\$TAGN= 010111	T23.3 033112	X\$TRUE= 000020
STPCLK 003240	TSTCON 002462	T\$TEMP= 000000	T23.4 033200	\$LSTIN= 000001
STPERR 007212	TSTNUM 002434	T\$TEST= 000040	T23.5 033230	\$LSTTA= 000001
STPLU 004754				

. ABS. 040044 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 21865 WORDS (86 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 69 PAGES
CZDMSD.BIC,CZDMSD.SEQ/C/N:TOC=SVC34R.MLB,CZDMSD.P11

SLSTIN	9-27#														
SLSTTA	9-28#														
A11	26-538#	26-568													
A12	26-275	26-895#													
ABORT	14-61#														
ADDRES	18-102	18-108	18-123	18-143	18-161	18-182	18-200	18-219	18-238	27-16	27-27#				
ADR	14-20#														
ANBITS	15-113#														
APA	14-239#														
ASBC0	14-196#														
ASBC1	14-195#														
ASBC2	14-194#														
ASSEMB	9-18	9-18													
AX0.15	14-309#	17-235	18-132	18-151	18-171	18-190	18-208	18-228	18-246						
AX0.16	14-310#	18-132	18-151	18-171	18-190	18-208	18-228	18-246							
AX1	14-92#	14-164#													
AX1.15	14-311#	18-132	18-151	18-171	18-190	18-208	18-228	18-246							
AX1.16	14-312#	18-132	18-151	18-171	18-190	18-208	18-228	18-246							
AX2	14-91#	14-163#													
AX2.15	14-313#	18-134	18-153	18-173	18-192	18-210	18-230	18-248							
AX2.16	14-314#	18-134	18-153	18-173	18-192	18-210	18-230	18-248							
AX3.15	14-315#	18-134	18-153	18-173	18-192	18-210	18-230	18-248							
AX3.16	14-316#	18-134	18-153	18-173	18-192	18-210	18-230	18-248							
AX315U	14-259#														
AXNUM	15-33#	17-152	17-190	17-203	17-228	17-230	17-236*	17-242*	17-243	17-245*	17-247	17-438	17-446*	17-489*	
	17-653	17-659*	17-688*	17-694*	17-971	17-973*	17-981*	17-992*	17-;19	17-;33*	17-<82*	17-<97	17-=00*	17-=13*	
	17-=17*	26-403*	26-431*	26-434*	26-446*	26-815*	26-;06*	26-;52*	26-<28*	26-<81*	26-224*				
BADDAT	15-35#	17-825*	17-826*	17-847*	17-848*	18-111	18-126	18-165	18-222	26-352*	26-361*	26-468*	26-743*	26-768*	
	26-793*	26-=91*	26->31*	26-289*	26-299*	26-A10*	26-A18*								
BAUDRT	27-23	27-35#													
BCC	14-140#	17-851	17-853	17-859											
BCCCHK	14-324#	17-849	26-<64	26-<72	26-=17	26-=25	26-F16	26-F70							
BDRATE	15-67#	17-;53	17-<08	21-75*											
BIT0	14-20#	14-42	14-54	14-63	14-92	14-104	14-116	14-128	14-140	14-152	14-164	14-176	14-188	14-200	
	14-212	14-221	14-233	14-246	14-258	14-269	14-342	17-230	17-271	17-400	17-580	17-621	17-661	21-45	
	26-<70	26-=23													
BIT00	14-20	14-20#													
BIT01	14-20	14-20#													
BIT02	14-20	14-20#													
BIT03	14-20	14-20#													
BIT04	14-20	14-20#													
BIT05	14-20	14-20#													
BIT06	14-20	14-20#													
BIT07	14-20	14-20#													
BIT08	14-20	14-20#													
BIT09	14-20	14-20#													
BIT1	14-20#	14-41	14-53	14-62	14-81	14-91	14-103	14-115	14-127	14-139	14-151	14-163	14-175	14-187	
	14-199	14-211	14-220	14-232	14-245	14-257	14-268	14-343	17-287	17-564	17-675				
BIT10	14-20#	14-279	14-291												
BIT11	14-20#	14-278	14-290												
BIT12	14-20#														
BIT13	14-20#														
BIT14	14-20#														
BIT15	14-20#	14-322	14-324	14-325	14-327	17-368	17-522								
BIT2	14-20#	14-40	14-52	14-61	14-80	14-90	14-102	14-114	14-126	14-138	14-150	14-162	14-174	14-186	
	14-198	14-210	14-219	14-231	14-244	14-256	14-267								

BIT3	14-20#	14-39	14-51	14-60	14-79	14-89	14-101	14-113	14-125	14-137	14-149	14-161	14-173	14-185
	14-197	14-209	14-218	14-230	14-243	14-255								
BIT4	14-20#	14-38	14-50	14-78	14-88	14-100	14-112	14-124	14-136	14-148	14-160	14-172	14-184	14-196
	14-208	14-229	14-242	14-254										
BIT5	14-20#	14-49	14-70	14-77	14-87	14-99	14-111	14-123	14-135	14-147	14-159	14-171	14-183	14-195
	14-207	14-228	14-241	14-253	14-266									
BIT6	14-20#	14-37	14-48	14-69	14-76	14-86	14-98	14-110	14-122	14-134	14-146	14-158	14-170	14-182
	14-194	14-206	14-227	14-240	14-252	14-265								
BIT7	14-20#	14-36	14-47	14-59	14-68	14-75	14-97	14-109	14-121	14-133	14-145	14-157	14-169	14-181
	14-193	14-205	14-217	14-226	14-239	14-251	14-264	17-819	17-820					
BIT8	14-20#	14-281	14-293											
BIT9	14-20#	14-280	14-292											
BOE	14-20#													
BPOLL	14-69#	26-A66												
BSEL1	15-56#	17-16*	17-18*	17-19*	17-33*	17-34*	17-39*	17-131*	17-133*	17-234*	17-372*	17-373*	17-375*	17-465*
	17-466*	17-471*	17-484*	17-987*	17-989*	17-;32*	17- ~~84~~ *	21-60*	21-61*	26-814*	26- ~~61~~ *	26- ~~25~~ *	26- ~~36~~ *	26- ~~38~~ *
	26- ~~88~~ *	26- ~~38~~ *	26- ~~94~~ *	26-A28*	26-A29*	26-A30*	26-A79*	26-B39*	26-C31*	26-C85*	26-D39*	26-D93*	26-E47*	26-F00*
	26-F54*													
BSEL4	15-57#	17-69	17-90*											
C\$AU	9-18#	25-10												
C\$AUTO	9-18#	22-20												
C\$BRK	9-18#	21-86												
C\$BSEG	9-18#													
C\$BSUB	9-18#	26-510	26-539	26-593	26-620	26-724	26-752	26-777	26-;08	26-;54	26- ~~30~~	26- ~~83~~	26- ~~82~~	26-A26
	26-A45	26-A63	26-A73	26-A98	26-B16	26-B33	26-B52	26-B69						
C\$CEFG	9-18#													
C\$CLCK	9-18#													
C\$CLEA	9-18#	23-11												
C\$CLOS	9-18#													
C\$CLP1	9-18#													
C\$CVEC	9-18#													
C\$DCLN	9-18#													
C\$DODU	9-18#	22-17												
C\$DRPT	9-18#													
C\$DU	9-18#	24-13												
C\$EDIT	9-18#	10-17												
C\$ERDF	9-18#	17-277	17-283	17-293	17-299	17-406	17-412	17-570	17-576	17-586	17-592	17-627	17-633	17-667
	17-673	17-681	17-687	17-833	17-838	17-857	17-863	17-872	17-878	17-887	17-893	17-902	17-908	17- ~~54~~
	17- ~~70~~	17- ~~06~~	17- ~~12~~	26-105	26-145	26-224	26-251	26-364	26-419	26-458	26-471	26-746	26-771	26-796
	26-;37	26-;43	26-;83	26-;89	26- ~~72~~	26- ~~94~~	26- ~~34~~	26- ~~49~~	26- ~~99~~	26- ~~49~~	26- ~~92~~	26-A02	26-A37	26-A56
	26-A85	26-A91	26-B09	26-B26	26-B45	26-B62	26-B79	26-B85	26-B91	26-C42	26-C96	26-D50	26-E04	26-E58
	26-F11	26-F65	26-G71	26-G83	26-H45									
C\$ERHR	9-18#													
C\$ERRO	9-18#													
C\$ERSF	9-18#													
C\$ERSO	9-18#													
C\$ESCA	9-18#	26-;38	26-;44	26-;84	26-;90	26- ~~93~~	26-A03	26-A38	26-A86	26-B80	26-B86			
C\$ESEG	9-18#													
C\$ESUB	9-18#	26-533	26-565	26-616	26-643	26-748	26-773	26-798	26-;48	26-;94	26- ~~77~~	26- ~~30~~	26-A21	26-A40
	26-A58	26-A68	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93						
C\$ETST	9-18#	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-570	26-644	26-695	26-799	26-827	26-944
	26-;77	26-;96	26- ~~32~~	26- ~~96~~	26- ~~58~~	26- ~~08~~	26- ~~58~~	26- ~~94~~	26- ~~42~~	26-B97	26-C51	26-D05	26-D59	26-E13
	26-E67	26-F21	26-F75	26-G18	26-H72									
C\$EXIT	9-18#													
C\$GETB	9-18#													
C\$GETW	9-18#													
C\$GMAN	9-18#	21-94												

CSGPHR	9-18#	21-52													
CSGPLO	9-18#														
CSGPRI	9-18#														
CSINIT	9-18#	21-99													
CSINLP	9-18#														
CSMANI	9-18#	21-81													
CSMEM	9-18#														
CSMSG	9-18#	18-103	18-116	18-135	18-154	18-174	18-193	18-211	18-231	18-249					
CSOPEN	9-18#														
CSPNTB	9-18#	18-102	18-108	18-109	18-110	18-111	18-123	18-124	18-125	18-126	18-142	18-143	18-144	18-145	
	18-161	18-162	18-163	18-164	18-165	18-181	18-182	18-183	18-184	18-200	18-201	18-202	18-218	18-219	
	18-220	18-221	18-222	18-238	18-239	18-240									
CSPNTF	9-18#	17-;61	17-;66	17-;75	21-85	21-90	24-12	26-735	26-737	26-762	26-787	26-823	26-825		
CSPNTS	9-18#														
CSPNTX	9-18#	18-112	18-113	18-114	18-115	18-127	18-128	18-129	18-130	18-131	18-132	18-133	18-134	18-146	
	18-147	18-148	18-149	18-150	18-151	18-152	18-153	18-166	18-167	18-168	18-169	18-170	18-171	18-172	
	18-173	18-185	18-186	18-187	18-188	18-189	18-190	18-191	18-192	18-203	18-204	18-205	18-206	18-207	
	18-208	18-209	18-210	18-223	18-224	18-225	18-226	18-227	18-228	18-229	18-230	18-241	18-242	18-243	
	18-244	18-245	18-246	18-247	18-248										
CSQIO	9-18#														
CSRDBU	9-18#														
CSREFG	9-18#	21-25	21-28	21-31	21-34										
CSRESE	9-18#	9-18#	24-10												
CSREVI	9-18#	10-17													
CSRFLA	9-18#														
CSRPT	9-18#	19-11													
CSSEFG	9-18#														
CSSPRI	9-18#	22-10													
CSSEVC	9-18#														
CSIPRI	9-18#														
C32BCC	14-253#	14-259													
C32ENB	14-256#														
CARR	14-152#	26->28	26-B87												
CHKABT	17-<97#														
CHPCHK	14-322#	26-672	26-975	26-:14	26-:49										
CHPTYP	15-46#	17-369	17-479*	17-483*	17-523	21-13*	26-436	26-442							
CKDATA	17-810#	26-35	26-42	26-86	26-91	26-94	26-127	26-132	26-135	26-186	26-189	26-196	26-210	26-213	
	26-238	26-241	26-277	26-280	26-310	26-313	26-558	26-561	26-636	26-639	26-688	26-691	26-866	26-869	

[illegible]

[illegible]

	26-317	26-317	26-332	26-332	26-332	26-366	26-366	26-384	26-384	26-384	26-480	26-480	26-506	26-506
	26-506	26-510	26-510	26-533	26-533	26-539	26-539	26-565	26-565	26-570	26-570	26-589	26-589	26-589
	26-593	26-593	26-616	26-616	26-620	26-620	26-643	26-643	26-644	26-644	26-663	26-663	26-663	26-695
	26-695	26-720	26-720	26-720	26-724	26-724	26-748	26-748	26-752	26-752	26-773	26-773	26-777	26-777
	26-798	26-798	26-799	26-799	26-812	26-812	26-812	26-827	26-827	26-850	26-850	26-850	26-944	26-944
	26-969	26-969	26-969	26-:77	26-:77	26-:02	26-:02	26-:02	26-:08	26-:08	26-:38	26-:44	26-:48	26-:48
	26-:54	26-:54	26-:84	26-:90	26-:94	26-:94	26-:96	26-:96	26-<24	26-<24	26-<24	26-<30	26-<30	26-<77
	26-<77	26-<83	26-<83	26-=30	26-=30	26-=32	26-=32	26-=48	26-=48	26-=48	26-=96	26-=96	26->13	26->13
	26->13	26->58	26->58	26->75	26->75	26->75	26-?08	26-?08	26-?25	26-?25	26-?25	26-?58	26-?58	26-?76
	26-?76	26-?76	26-?94	26-?94	26-@09	26-@09	26-@09	26-@42	26-@42	26-@73	26-@73	26-@73	26-@82	26-@82
	26-@93	26-A03	26-A21	26-A21	26-A26	26-A26	26-A38	26-A40	26-A40	26-A45	26-A45	26-A58	26-A58	26-A63
	26-A63	26-A68	26-A68	26-A73	26-A73	26-A86	26-A93	26-A93	26-A98	26-A98	26-B11	26-B11	26-B16	26-B16
	26-B28	26-B28	26-B33	26-B33	26-B47	26-B47	26-B52	26-B52	26-B64	26-B64	26-B69	26-B69	26-B80	26-B86
	26-B93	26-B93	26-B97	26-B97	26-C17	26-C17	26-C17	26-C51	26-C51	26-C71	26-C71	26-C71	26-D05	26-D05
	26-D25	26-D25	26-D25	26-D59	26-D59	26-D79	26-D79	26-D79	26-E13	26-E13	26-E33	26-E33	26-E33	26-E67
	26-E67	26-E87	26-E87	26-E87	26-F21	26-F21	26-F41	26-F41	26-F41	26-F75	26-F75	26-F92	26-F92	26-F92
	26-G18	26-G18	26-G37	26-G37	26-G37	26-H72	26-H72	27-25	28-20	28-45				
FSHARD	9-18#	27-14	27-25											
FSHW	9-18#	12-9	12-23											
FSINIT	9-18#	21-8	21-99											
FSJMP	9-18#													
FSMOD	9-18#	9-24	28-45											
FSMSG	9-18#	18-101	18-103	18-107	18-116	18-122	18-135	18-141	18-154	18-160	18-174	18-180	18-193	18-199
	18-211	18-217	18-231	18-237	18-249									
FSPROT	9-18#	20-8	20-12											
FSPWR	9-18#													
FSRPT	9-18#	19-9	19-11											
FSSEG	9-18#													
FSOFT	9-18#	28-13	28-20											
FSSRV	9-18#													
FSSUB	9-18#	26-510	26-533	26-539	26-565	26-593	26-616	26-620	26-643	26-724	26-748	26-752	26-773	26-777
	26-798	26-:08	26-:48	26-:54	26-:94	26-<30	26-<77	26-<83	26-=30	26-@82	26-A21	26-A26	26-A40	26-A45
	26-A58	26-A63	26-A68	26-A73	26-A93	26-A98	26-B11	26-B16	26-B28	26-B33	26-B47	26-B52	26-B64	26-B69
	26-B93													
FSSW	9-18#	13-8	13-15											
FSTEST	9-18#	26-20	26-46	26-65	26-147	26-173	26-253	26-267	26-284	26-299	26-317	26-332	26-366	26-384
	26-480	26-506	26-570	26-589	26-644	26-663	26-695	26-720	26-799	26-812	26-827	26-850	26-944	26-969
	26-:77	26-:02	26-:96	26-<24	26-=32	26-=48	26-=96	26->13	26->58	26->75	26-?08	26-?25	26-?58	26-?76
	26-?94	26-@09	26-@42	26-@73	26-B97	26-C17	26-C51	26-C71	26-D05	26-D25	26-D59	26-D79	26-E13	26-E33
	26-E67	26-E87	26-F21	26-F41	26-F75	26-F92	26-G18	26-G37	26-H72					
FMT1	18-9#	18-102	18-108	18-123	18-143	18-161	18-182	18-200	18-219	18-238				
FMT10	18-18#	18-142	18-181	18-218										
FMT11	18-19#	18-162												
FMT12	18-20#	26-737												
FMT13	18-21#	26-762												
FMT14	18-22#	26-787												
FMT15	18-23#	26-825												
FMT16	21-85	21-101#												
FMT17	21-90	21-103#												
FMT18	18-24#	26-735	26-823											
FMT19	17-:66	18-25#												
FMT2	18-10#	18-109	18-124	18-144	18-163	18-183	18-201	18-220	18-239					
FMT25	17-:61	18-26#												
FMT26	17-:75	18-27#												
FMT27	24-12	24-15#												
FMT3	18-11#	18-111	18-126	18-165	18-222									
FMT4	18-12#	18-112	18-127	18-131	18-146	18-150	18-166	18-170	18-185	18-189	18-203	18-207	18-223	18-227

FMT5	18-241 18-13# 18-242	18-245 18-113 18-246	18-128	18-132	18-147	18-151	18-167	18-171	18-186	18-190	18-204	18-208	18-224	18-228
FMT6	18-14# 18-244	18-115 18-248	18-130	18-134	18-149	18-153	18-169	18-173	18-188	18-192	18-206	18-210	18-226	18-230
FMT7	18-15#	18-110	18-145	18-202	18-221									
FMT8	18-16#	18-125	18-164	18-184	18-240									
FMT9	18-17# 18-243	18-114 18-247	18-129	18-133	18-148	18-152	18-168	18-172	18-187	18-191	18-205	18-209	18-225	18-229
FRSPAS	15-38#	21-43*	26-729	26-756	26-781	26-817								
FRSTIM	15-37#	21-16	21-23*											
GSCNTO	9-18#													
G\$DELM	9-18#													
G\$DISP	9-18#													
G\$EXCP	9-18#													
G\$HILI	9-18#													
G\$LOLI	9-18#													
G\$NO	9-18#	21-94												
G\$OFFS	9-18#	21-94	27-16	27-17	27-18	27-19	27-20	27-21	27-22	27-23	28-15	28-16	28-17	28-18
G\$OF SI	9-18#	21-94	27-16	27-17	27-18	27-19	27-20	27-21	27-22	27-23	28-15	28-16	28-17	28-18
G\$PRMA	9-18#	27-16	27-17											
G\$PRMD	9-18#	27-18	27-19	27-20	27-21	27-22	27-23	28-18						
G\$PRML	9-18#	21-94	28-15	28-16	28-17									
G\$RADA	9-18#													
G\$RADB	9-18#													
G\$RADD	9-18#													
G\$RADL	9-18#	21-94	28-15	28-16	28-17									
G\$RADO	9-18#	27-16	27-17	27-18	27-19	27-20	27-21	27-22	27-23	28-18				
G\$XFER	9-18#													
G\$YES	9-18#	27-16	27-17	27-18	27-19	27-20	27-21	27-22	27-23	28-15	28-16	28-17	28-18	
GETALL	17-227# 17-671 17-<68 26-;35 26-A83 26-F09	17-275 17-679 17-=04 26-;41 26-A89 26-F63	17-281 17-685 17-=10 26-;81 26-B07 26-G69	17-291 17-831 26-103 26-;87 26-B24 26-G81	17-297 17-836 26-143 26-=70 26-B43 26-H43	17-404 17-855 26-222 26-=92 26-B60	17-410 17-861 26-249 26->32 26-B77	17-568 17-870 26-362 26->47 26-B83	17-574 17-876 26-417 26->97 26-B89	17-584 17-885 26-456 26-?47 26-C40	17-590 17-891 26-469 26-290 26-C94	17-625 17-900 26-744 26-A00 26-D48	17-631 17-906 26-769 26-A35 26-E02	17-665 17-<52 26-794 26-A54 26-E56
GETPRM	21-36	21-48#	21-55											
GE1REG	17-104#	17-233												
GOAH	14-60#													
GOODAT	15-34# 26-767*	17-823* 26-792*	17-824* 26-=90*	17-845* 26->30*	17-846* 26-288*	18-111 26-298*	18-126 26-A09*	18-165 26-A17*	18-222	26-351*	26-465*	26-466*	26-467*	26-742*
HDX	14-78#	14-148#	26-=87	26-?82	26-218	26-B18	26-B22							
HELP	9-4#	10-9	10-19	11-10	16-24									
HOE	14-20#													
ISAU	9-18#	25-9#	25-10#											
ISAUTO	9-18#	22-8#	22-20#											
ISCLN	9-18#	23-8#	23-11#											
ISDU	9-18#	24-8#	24-13#											
ISHRD	27-14#	27-25#												
ISINIT	9-18#	21-8#	21-99#											
ISMOD	9-18#	9-24	9-24#	28-45	28-45#									
ISMSG	9-18# 18-211#	18-101# 18-217#	18-103# 18-231#	18-107# 18-237#	18-116# 18-249#	18-122#	18-135#	18-141#	18-154#	18-160#	18-174#	18-180#	18-193#	18-199#
ISPROT	9-18#	20-8#												
ISPTAB	9-18#													
ISPUR	9-18#													
ISRPT	9-18#	19-9#	19-11#											

[illegible]

LSAU	10-17	25-9#	
LSAUT	10-17#		
LSAUTO	10-17	22-8#	
LSCCP	10-17#		
LSCLEA	10-17	23-8#	
LSCO	10-17#		
LSDEPO	10-17#		
LSDESC	10-17	16-17#	
LSDESP	10-17#		
LSDEVP	10-17#		
LSDISP	10-17	11-8#	
LSDLY	10-17#		
LSDTP	10-17#		
LSDTYP	10-17#		
LSDU	10-17	24-8#	
LSDUT	10-17#		
LSDVTY	10-17	16-12#	
LSEF	10-17#		
LEENVI	10-17#		
LEETP	10-17#		
LEEXP1	10-17#		
LEEXP4	10-17#		
LEEXP5	10-17#		
LSHARD	10-17	27-14	27-14#
LSHIME	10-17#		
LSHPCP	10-17#		
LSHPTP	10-17#		
LSHW	10-17	12-9	12-9#
LSICP	10-17#		
LSINIT	10-17	21-8#	
LSLADP	10-17#		
LSLAST	10-17	28-47#	
LSLOAD	10-17#		
LSLUN	10-17#		
LSMREV	10-17#		
LSNAME	10-17#		
LSPRIO	10-17#		
LSPROT	10-17	20-8#	
LSPT	10-17#		
LSREPP	10-17#		
LSREV	10-17#		
LSRPT	19-9#		
LSOFT	10-17	28-13	28-13#
LSPC	10-17#		
LSPCP	10-17#		
LSPTP	10-17#		
LSSTA	10-17#		
LSW	10-17	13-8	13-8#
LSTEST	10-17#		
LSTIML	10-17#		
LSUNIT	10-17#	21-50	
L10000	12-9	12-23#	
L10001	13-8	13-15#	
L10002	18-103#		
L10003	18-116#		
L10004	18-135#		
L10005	18-154#		

L10006	18-174#		
L10007	18-193#		
L10010	18-211#		
L10011	18-231#		
L10012	18-249#		
L10013	19-11#		
L10015	21-99#		
L10016	22-20#		
L10017	23-11#		
L10020	24-13#		
L10021	25-10#		
L10022	26-46#		
L10023	26-147#		
L10024	26-253#		
L10025	26-284#		
L10026	26-317#		
L10027	26-366#		
L10030	26-480#		
L10031	26-570#		
L10032	26-533#		
L10033	26-565#		
L10034	26-644#		
L10035	26-616#		
L10036	26-643#		
L10037	26-695#		
L10040	26-799#		
L10041	26-748#		
L10042	26-773#		
L10043	26-798#		
L10044	26-827#		
L10045	26-944#		
L10046	26-:77#		
L10047	26-:96#		
L10050	26-:38	26-:44	26-:48#
L10051	26-:84	26-:90	26-:94#
L10052	26-=32#		
L10053	26-<77#		
L10054	26-=30#		
L10055	26-=96#		
L10056	26->58#		
L10057	26-?08#		
L10060	26-?58#		
L10061	26-?94#		
L10062	26-@42#		
L10063	26-B97#		
L10064	26-@93	26-A03	26-A21#
L10065	26-A38	26-A40#	
L10066	26-A58#		
L10067	26-A68#		
L10070	26-A86	26-A93#	
L10071	26-B11#		
L10072	26-B28#		
L10073	26-B47#		
L10074	26-B64#		
L10075	26-B80	26-B86	26-B93#
L10076	26-C51#		
L10077	26-D05#		

L10100	26-D59#														
L10101	26-E13#														
L10102	26-E67#														
L10103	26-F21#														
L10104	26-F75#														
L10105	26-G18#														
L10106	26-H72#														
L10107	27-14	27-25#													
L10110	28-13	28-20#													
LDBYTS	17-:30#	26-25	26-:13	26-:59	26-<38	26-<91	26-G45	26-G51							
LDMSG1	17-:55#	26-60	26->37	26->87	26-?37										
LDTXSI	17-343#	17-456	17-457	17-515	17-948	17-955	17-956	17-:12	17-:38	17-:88	26-29	26-30			
LOADAT	15-36#	18-162													
LODATA	17-936#	26-C29	26-C83	26-D37	26-D91	26-E45	26-E99	26-F53							
LODMSG	17-:05#	17-:60	26-181	26-205	26-233	26-272	26-668	26-861	26-891	26-922	26-980	26-:15	26-:50	26-?84	
	26-G60														
LODSIL	17-939	17-:57	17-:84#	26-73	26-80	26-114	26-121	26-305	26-337	26-515	26-518	26-521	26-547	26-550	
	26-553	26-598	26-601	26-604	26-625	26-628	26-631	26-:16	26-:62	26-<35	26-<41	26-<88	26-<94	26-F97	
	26-G00	26-G48	26-G54	26-G57											
LOE	14-20#														
LOGDEV	15-17#	21-42*	21-49*	21-50	21-52	22-17	24-12								
LOOPBK	27-22	27-33#													
LOOPIN	17-130#														
LOT	14-20#														
LU2MOD	9-24#														
LULoop	14-39#	17-39	17-234	17-372	17-465	17-987	17-989	17-:32	17-<84	26-814	26-=61	26->25	26->36	26->38	
	26->88	26-?38	26-294	26-A30	26-A79	26-839	26-C31	26-C85	26-D39	26-D93	26-E47	26-F00	26-F54		
LULP	14-70#	14-135#	26-401												
LUR10	14-301#	17-106	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242					
LUR11	14-302#	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242						
LUR12	14-303#	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242						
LUR13	14-304#	18-113	18-128	18-147	18-167	18-186	18-204	18-224	18-242						
LUR14	14-305#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244						
LUR15	14-306#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244						
LUR16	14-307#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244						
LUR17	14-308#	18-115	18-130	18-149	18-169	18-188	18-206	18-226	18-244						
LUREG	14-301	14-302	14-303	14-304	14-305	14-306	14-307	14-308	14-309	14-310	14-311	14-312	14-313	14-314	
	14-315	14-316	15-11#												
LUSW11	15-63#	21-71*	26-740	26-742											
LUSW12	15-64#	21-72*	26-765	26-767											
LUSW13	15-65#	21-73*	26-790	26-792											
MAINT1	14-79#	17-<43	26-835												
MAINT2	14-80#	17-:83	17-<14	17-<59	26-A76										
MCLK	14-175#	26-A14													
MCLR	14-37#	17-33	17-34												
MIFLAG	13-10#	21-78													
MODINT	15-47#	17-:23*	17-:55*	17-:84*	17-:88*	17-:95*	17-<11*	26-C23	26-C77	26-D31	26-D85	26-E39	26-E93	26-F47	
MODR	14-149#	26-=87	26-286	26-288	26-A87										
MPCSR	15-55#	18-102	18-108	18-123	18-143	18-161	18-182	18-200	18-219	18-238	21-59*	21-90	22-13	26-735	
	26-823														
MPIVEC	15-60#	21-66*													
MPOVEC	15-61#	21-67*	21-68*												
MPRIOR	15-62#	21-69*													
MSG1	15-219#	17-:61	17-:63	17-:66	26-862	26-892	26-923	26-981	26-:16	26-:51	26-?85	26-G61			
MSG2	15-231#	26-669													
MSG3	15-238#	26-182	26-206	26-234	26-273										
MSTCLR	17-30#	17-441	26-45	26-139	26-146	26-245	26-252	26-283	26-316	26-365	26-390	26-479	26-569	26-694	

	26-725	26-753	26-778	26-813	26-855	26-885	26-916	26-943	26-:76	26-:95	26-:31	26-:51	26->16	26->78
	26-728	26-793	26-811	26-841	26-883	26-A27	26-A46	26-A64	26-A74	26-A99	26-B17	26-B34	26-B53	26-B96
	26-C20	26-C74	26-D28	26-D82	26-E36	26-E90	26-F44	26-G17	26-H71					
MVIOX	14-336#	17-65												
MVIXO	14-337#	17-87												
NEWST	21-32	21-41#	21-51											
OSAPTS	9-18#	10-17												
OSAU	9-18#	10-7#	10-17											
OSBGNR	9-18#	10-17												
OSBGNS	9-18#	10-7#	10-17											
OSDU	9-18#	10-7#	10-17											
OSERRT	9-18#	10-17												
OSGNSW	9-18#	10-7#	10-17											
OSPOIN	9-18#	10-7	10-7#	10-7#	10-7#	10-7#	10-17							
OSSETU	9-18#	10-17	28-47											
OACT	14-122#	17-402	17-408	17-469	26-412									
OACTIV	17-392#	17-444	17-461	17-476	17-526	26-:81								
OC	14-59#	14-121#												
OCOR	14-172#	17-289	17-295	17-481										
OP	14-257#	14-259	17-:23	26-:54	26->19	26->81	26-?31							
ORDY	14-124#	17-273	17-279											
OSIRDY	17-263#	17-442	17-459	17-486	17-530	17-536	26-?89							
OVRR	14-137#	17-896	17-898	17-904	26-101	26-141								
PATA	15-124#	17-944	26-G46	26-G52	26-G90	26-H12								
PATB	15-147#	17-949	26-G95	26-H17										
PATCH	28-36#													
PATQ	15-159#	26-:14	26-:60											
PATR	15-169#	26-<39	26-<61	26-<92	26-:14									
PATS	15-186#	26-26	26-33	26-38										
PATT	15-204#	26-537												
PNT	14-20#													
POLL	14-75#	26-A47												
PRI	14-20#													
PRI00	14-20#													
PRI01	14-20#													
PRI02	14-20#													
PRI03	14-20#													
PRI04	14-20#													
PRI05	14-20#													
PRI06	14-20#													
PRI07	14-20#	22-10	22-12											
PRIOR	15-19#													
PRIRTY	27-18	27-29#												
PRNFLG	13-11#	26-731	26-758	26-783	26-819									
PSTACK	15-18#	17-301	17-414	17-594	17-635	17-689	17-917	17-:14	21-10*					
R14NRW	15-110#													
RAB	14-138#	17-881	17-883	17-889	26-220	26-247								
RABT	14-198#	17-:02												
RAX15	15-28#	17-167*	17-168*	17-238	17-:69	17-:79	17-:86	17-:97	17-<02	17-<05	17-<19	17-<22	17-<25	17-<31
	26-825													
RAX16	15-29#	17-171*	17-172*	17-240	17-663	17-669	17-677	17-683	17-:02	17-:08	26-449	26-460	26-468	
RCV1ST	17-729#	26-76	26-117	26-184	26-194	26-208	26-236	26-275	26-308	26-340	26-556	26-634	26-686	
RCVBUF	15-256#	17-943	17-:64	26-:63	26-:78	26->40	26->55	26->90	26-?05	26-?40	26-?55	26-C30*	26-C33	26-C48
	26-C84*	26-C87	26-D02	26-D38*	26-D41	26-D56	26-D92*	26-D95	26-E10	26-E46*	26-E49	26-E64	26-F02	26-F18
	26-F56	26-F72												
RDALL	14-102#	26-624												
RDAX	14-88#	14-160#	17-154											

RDRXSI	17-705#	17-816	26-100	26-140	26-219	26-246	26-353							
READAX	17-148#	17-237	17-660	17-:34	17-=01	26-448	26-816							
READLU	17-57#	17-108	17-158	17-166	17-170	17-209	17-270	17-286	17-399	17-468	17-480	17-563	17-579	17-620
	17-707	17-711	17-750	17-828	17-<47	17-<63	26-411	26-727	26-755	26-780	26-;29	26-;75	26-=65	26-=86
	26->27	26->42	26->92	26-?42	26-@85	26-@95	26-A06	26-A13	26-A32	26-A51	26-A80	26-B04	26-B21	26-B40
	26-B57	26-B74	26-C35	26-C89	26-D43	26-D97	26-E51	26-F04	26-F58	26-G66	26-G78	26-H40		
READY	14-157#	17-159	17-210											
REDBYT	15-26#	17-69*	17-70*	17-109	17-159	17-167	17-171	17-210	17-273	17-279	17-289	17-295	17-402	17-408
	17-469	17-481	17-566	17-572	17-582	17-588	17-623	17-629	17-708	17-712	17-751	17-829	17-<48	17-<64
	26-412	26-728*	26-737	26-740	26-743	26-762	26-765	26-768	26-787	26-790	26-793	26-;33	26-;39	26-;79
	26-;85	26-=66	26-=87*	26-=8E	26-=91	26->28	26->31	26->43	26->93	26-?43	26-@86	26-@89	26-@96	26-@99
	26-A07	26-A10	26-A14*	26-A15	26-A18	26-A33	26-A52	26-A81	26-A87	26-B05	26-B22	26-B41	26-B58	26-B75
	26-B81	26-B87	26-C36	26-C90	26-D44	26-D98	26-E52	26-F05	26-F59	26-G67	26-G79	26-H41		
REDDAT	15-70#													
REG0	15-80#													
REG1	15-81#													
REG2	15-82#	21-91*	21-94	21-95										
REG3	15-83#													
REG4	15-84#													
REG5	15-85#													
REG6	15-86#													
REG7	15-87#													
REGNUM	15-32#	17-32	17-40*	17-43*	17-59	17-85	17-105	17-107*	17-111*	17-112	17-114*	17-149	17-151*	17-165*
	17-169*	17-173*	17-187	17-189*	17-194*	17-198*	17-202*	17-215*	17-263	17-269*	17-285*	17-300*	17-306*	17-343
	17-345*	17-348*	17-351*	17-392	17-398*	17-413*	17-419*	17-437	17-452*	17-464*	17-478*	17-490*	17-556	17-562*
	17-578*	17-593*	17-599*	17-613	17-619*	17-634*	17-640*	17-705	17-706*	17-710*	17-713*	17-731	17-734*	17-766*
	17-782	17-783*	17-794*	17-811	17-827*	17-835*	17-841*	17-916*	17-921*	17-972	17-977*	17-991*	17-;18	17-<40*
	17-<45*	17-<61*	17-<73*	17-<83*	18-110	18-145	18-162	18-202	18-221	26-97*	26-138*	26-216*	26-244*	26-344*
	26-356*	26-360*	26-396*	26-400*	26-408*	26-726*	26-754*	26-779*	26-;07*	26-;53*	26-<29*	26-<82*	26-=62*	26-=85*
	26->26*	26->39*	26->89*	26-?39*	26-?78*	26-@17*	26-@20*	26-@84*	26-A12*	26-A31*	26-A48*	26-A50*	26-A65*	26-A75*
	26-B01*	26-B03*	26-B19*	26-B36*	26-B38*	26-B55*	26-B73*	26-C32*	26-C86*	26-D40*	26-D94*	26-E48*	26-F01*	26-F55*
	26-G65*	26-H34*	26-H37*											
REOM	14-199#	17-677	17-683	17-=08	26-449	26-467								
RERR	14-193#													
RETADR	15-25#	17-302	17-415	17-595	17-636	17-690	17-918	17-;67	17-=15	26-21*	26-66*	26-174*	26-268*	26-300*
	26-333*	26-385*	26-511*	26-540*	26-594*	26-621*	26-664*	26-854*	26-973*	26-;09*	26-;55*	26-<31*	26-<84*	26-=50*
	26->15*	26->77*	26-?27*	26-?77*	26-@10*	26-@75*	26-C19*	26-C73*	26-D27*	26-D81*	26-E35*	26-E89*	26-F43*	26-F93*
	26-G38*													
RING	14-145#	26-=87	26-@86	26-@88	26-A33									
ROMI	14-41#	17-16	17-18	17-19	17-131	17-133								
ROMO	14-40#	17-16	17-18	17-19	17-131	17-133								
ROR	14-197#													
RRDYTO	14-342#	17-150	17-163											
RSEOM	17-653#	17-744												
RSOM	14-200#	17-663	17-669											
RTS	14-147#	26-B75	26-G67	26-G79	26-H41									
RUN	14-36#	17-34	17-133	26-A29										
RX0	14-116#	14-188#												
RX1	14-115#	14-187#												
RX2	14-114#	14-186#												
RX3	14-113#	14-185#												
RX4	14-112#	14-184#												
RX5	14-111#	14-183#												
RX6	14-110#	14-182#												
RX7	14-109#	14-181#												
RXABT	14-291#	26-190	26-214	26-242	26-314									
RXBCC	14-293#	17-953	17-:68	26-<72	26-=25	26-E46	26-H01	26-H23	26-H63					

RXEBL	14-292#	26-41	26-190	26-214	26-242	26-281	26-314	26-562	26-640	26-C30	26-D38	26-D92		
RXLEN0	14-269#	17-817	26-;20	26-;66	26-<45	26-<98	26-E98	26-F52						
RXLEN1	14-268#	17-817	26-;20	26-;66	26-<45	26-<98	26-E98	26-F52						
RXLEN2	14-267#	17-817	26-;20	26-;66	26-<45	26-<98	26-E98	26-F52						
RXOVR	14-290#													
RXWORD	15-44#	17-708*	17-709*	17-712*	17-820*	17-821	17-826	17-842	17-848	17-853	17-859	17-868	17-874	17-883
	17-889	17-898	17-904	26-101	26-141	26-220	26-247	26-354	26-358	26-361				
S&LSYM	9-18#	12-23#	13-15#	18-103#	18-116#	18-135#	18-154#	18-174#	18-193#	18-211#	18-231#	18-249#	19-11#	21-94
	21-94	21-94	21-94#	21-99#	22-20#	23-11#	24-13#	25-10#	26-46#	26-147#	26-253#	26-284#	26-317#	26-366#
	26-480#	26-533#	26-565#	26-570#	26-616#	26-643#	26-644#	26-695#	26-748#	26-773#	26-798#	26-799#	26-827#	26-944#
	26-;77#	26-;48#	26-;94#	26-;96#	26-<77#	26-;30#	26-;32#	26-;96#	26->58#	26-?08#	26-?58#	26-?94#	26-242#	26-A21#
	26-A40#	26-A58#	26-A68#	26-A93#	26-B11#	26-B28#	26-B47#	26-B64#	26-B93#	26-B97#	26-C51#	26-D05#	26-D59#	26-E13#
	26-E67#	26-F21#	26-F75#	26-G18#	26-H72#	27-25#	28-20#							
SAVE4	15-40#	21-18*	21-21	22-18										
SAVE6	15-41#	21-19*	21-22	22-19										
SAVLEN	15-48#	17-45*	17-817	17-986*	21-15*	26-<46*	26-<99*							
SCRACH	15-16#													
SEC	14-242#													
SECA	14-100#	26-514	26-546	26-597	26-624	26-F96								
SEL4	15-58#	21-62*	21-63*											
SEL6	15-59#	17-17*	17-132*	21-64*	21-65*									
SEIFR	14-77#	26-B00												
SELSBY	14-81#	26-B54												
SETUP	17-971#	26-391	26-856	26-886	26-917	26-;55	26->20	26->82	26-?32	26-212	26-C24	26-C78	26-D32	26-D86
	26-E40	26-E94	26-F48											
SFPTBL	13-8#													
SIGQ	14-170#	26-A52												
SIGR	14-169#	26-B05												
SOM	14-63#													
STALL	17-330#	17-374	17-376	17-467	17-472	17-485								
STARES	15-39#	17-;59	17-;64	17-;73	21-38*	21-44*								
STARST	21-26	21-29	21-37#											
STBY	14-151#	26-;87	26-B58											
STEPLU	14-38#	17-373	17-375	17-466	17-471	17-484								
STEPMP	14-42#	17-18	17-19											
STPCLK	17-15#	17-67	17-91											
STPERR	17-782#	26-673	26-678	26-683	26-985	26-;20	26-;55	26-<51	26-;04					
STPLU	17-364#	17-532	17-746	17-789	17-913	26-31	26-78	26-83	26-119	26-124	26-342	26-409	26-422	26-426
	26-429	26-440	26-451	26-524	26-528	26-607	26-611	26-671	26-676	26-681	26-864	26-894	26-925	26-983
	26-988	26-;18	26-;23	26-;53	26-;58	26-;22	26-;24	26-;27	26-;68	26-;70	26-;73	26-<48	26-<54	26-<58
	26-;01	26-;07	26-;11	26-?87	26-228	26-233	26-236	26-G03	26-G63	26-G73	26-G76	26-H38		
STR	14-241#													
STRIP	14-101#	26-72	26-113	26-667	26-674	26-679	26-684	26-858	26-976	26-986	26-;12	26-;58	26-<34	26-<52
	26-<87	26-;05	26-;57	26->22	26->84	26-?34	26-?81	26-C80	26-E42	26-E96	26-F50	26-G44		
SUBRPC	15-20#	17-264	17-265	17-267*	17-268*	17-305*	17-393	17-394	17-396*	17-397*	17-418*	17-439*	17-440*	17-492*
	17-512*	17-513*	17-539*	17-557	17-558	17-560*	17-561*	17-598*	17-614	17-615	17-617*	17-618*	17-639*	17-654
	17-655	17-657*	17-658*	17-693*	17-732*	17-733*	17-769*	17-812*	17-813*	17-923*	17-993*	17-;21*	17-;22*	17-<85*
	17-<98*	17-<99*	18-142	18-181	18-218	21-11*								
SVCGBL	9-18#	9-24	9-24	9-24	9-32#	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	11-8	12-9	12-9	13-8	13-8	16-12	16-17	18-101	18-107	18-122	18-141	18-160	18-180
	18-199	18-217	18-237	19-9	20-8	21-8	22-8	23-8	24-8	25-9	27-14	28-13	28-47	28-47
	28-47	28-47#												
SVCINS	9-18#	9-29#	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

	21-94	21-99	22-20	23-11	24-13	25-10	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-533
	26-565	26-570	26-616	26-643	26-644	26-695	26-748	26-773	26-798	26-799	26-827	26-944	26-:77	26-:48
	26-:94	26-:96	26-:77	26-:30	26-:32	26-:96	26-:58	26-:08	26-:58	26-:94	26-:42	26-A21	26-A40	26-A58
	26-A68	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93	26-B97	26-C51	26-D05	26-D59	26-E13	26-E67	26-F21
	26-F75	26-G18	26-H72	27-25	28-20									
SVCTST	9-18#	9-30#	26-20	26-65	26-173	26-267	26-299	26-332	26-384	26-506	26-589	26-663	26-720	26-812
	26-850	26-969	26-:02	26-:24	26-:48	26-:13	26-:75	26-:25	26-:76	26-:09	26-:73	26-C17	26-C71	26-D25
	26-D79	26-E33	26-E87	26-F41	26-F92	26-G37								
SW0	14-127#													
SW1	14-126#													
SW2	14-125#													
SW3	14-123#													
SWIFLG	13-12#	26-738	26-763	26-788										
SWPAC1	27-19	27-30#												
SWPAC2	27-20	27-31#												
SWPAC3	27-21	27-32#												
SYNO	14-233#													
SYN1	14-232#													
SYN2	14-231#													
SYN3	14-230#													
SYN4	14-229#													
SYN5	14-228#													
SYN6	14-227#													
SYN7	14-226#													
SYNCH	14-234#	26-71	26-112	26-666	26-857	26-975	26-:56	26-:21	26-:83	26-:33	26-:80	26-:13	26-C79	26-E41
	26-G43													
TSSAU	25-9#	25-10												
TSSAUT	22-8#	22-20												
TSSCLE	23-8#	23-11												
TSSDU	24-8#	24-13												
TSSHAR	27-14	27-14#	27-25											
TSSHW	12-9	12-9#	12-23											
TSSINI	21-8#	21-99												
TSSMSG	18-101#	18-103	18-107#	18-116	18-122#	18-135	18-141#	18-154	18-160#	18-174	18-180#	18-193	18-199#	18-211
	18-217#	18-231	18-237#	18-249										
TSSPRO	20-8#													
TSSRPT	19-9#	19-11												
TSSOF	28-13	28-13#	28-20											
TSSSUB	26-510#	26-533	26-539#	26-565	26-593#	26-616	26-620#	26-643	26-724#	26-748	26-752#	26-773	26-777#	26-798
	26-:08#	26-:38	26-:44	26-:48	26-:54#	26-:84	26-:90	26-:94	26-:30#	26-:77	26-:83#	26-:30	26-:82#	26-:93
	26-A03	26-A21	26-A26#	26-A38	26-A40	26-A45#	26-A58	26-A63#	26-A68	26-A73#	26-A86	26-A93	26-A98#	26-B11
	26-B16#	26-B28	26-B33#	26-B47	26-B52#	26-B64	26-B69#	26-B80	26-B86	26-B93				
TSSSW	13-8	13-8#	13-15											
TSTES	26-20#	26-46	26-65#	26-147	26-173#	26-253	26-267#	26-284	26-299#	26-317	26-332#	26-366	26-384#	26-480
	26-506#	26-570	26-589#	26-644	26-663#	26-695	26-720#	26-799	26-812#	26-827	26-850#	26-944	26-969#	26-:77
	26-:02#	26-:96	26-:24#	26-:32	26-:48#	26-:96	26-:13#	26-:58	26-:75#	26-:08	26-:25#	26-:58	26-:76#	26-:94
	26-:09#	26-:42	26-:73#	26-B97	26-C17#	26-C51	26-C71#	26-D05	26-D25#	26-D59	26-D79#	26-E13	26-E33#	26-E67
	26-E87#	26-F21	26-F41#	26-F75	26-F92#	26-G18	26-G37#	26-H72						
TSARGC	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	17-:61	17-:61	17-:61#	17-:66	17-:66	17-:66	17-:66#	17-:66#	17-:75	17-:75	17-:75#
	18-102	18-102	18-102	18-102	18-102#	18-102#	18-102#	18-108	18-108	18-108	18-108	18-108#	18-108#	18-108#
	18-109	18-109	18-109#	18-110	18-110	18-110	18-110	18-110#	18-110#	18-110#	18-110#	18-111	18-111	18-111
	18-111#	18-111#	18-111#	18-112	18-112	18-112	18-112	18-112#	18-112#	18-112#	18-112#	18-113	18-113	18-113
	18-113	18-113	18-113#	18-113#	18-113#	18-113#	18-113#	18-114	18-114	18-114	18-114#	18-114#	18-115	18-115
	18-115	18-115	18-115	18-115	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#
	18-123#	18-123#	18-124	18-124	18-124#	18-125	18-125	18-125	18-125	18-125#	18-125#	18-125#	18-125#	18-126
	18-126	18-126	18-126#	18-126#	18-126#	18-127	18-127	18-127	18-127	18-127#	18-127#	18-127#	18-128	18-128

[illegible]

	17-<70#	17-=06	17-=06#	17-=12	17-=12#	26-105	26-105#	26-145	26-145#	26-224	26-224#	26-251	26-251#	26-364
	26-364#	26-419	26-419#	26-458	26-458#	26-471	26-471#	26-746	26-746#	26-771	26-771#	26-79J	26-796#	26-;37
	26-;37#	26-;43	26-;43#	26-;83	26-;83#	26-;89	26-;89#	26-=72	26-=72#	26-=94	26-=94#	26->34	26->34#	26->49
	26->49#	26->99	26->99#	26-?49	26-?49#	26-@92	26-@92#	26-A02	26-A02#	26-A37	26-A37#	26-A56	26-A56#	26-A85
	26-A85#	26-A91	26-A91#	26-B09	26-B09#	26-B26	26-B26#	26-B45	26-B45#	26-B62	26-B62#	26-B79	26-B79#	26-B85
	26-B85#	26-B91	26-B91#	26-C42	26-C42#	26-C96	26-C96#	26-D50	26-D50#	26-E04	26-E04#	26-E58	26-E58#	26-F11
	26-F11#	26-F65	26-F65#	26-G71	26-G71#	26-G83	26-G83#	26-H45	26-H45#					
TSEXCP	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#	27-21	27-21#	27-22	27-22#
	27-23	27-23#	28-18	28-18#										
T\$FLAG	26-;38	26-;38#	26-;38#	26-;44	26-;44#	26-;44#	26-;84	26-;84#	26-;84#	26-;90	26-;90#	26-;90#	26-@93	26-@93#
	26-@93#	26-A03	26-A03#	26-A03#	26-A38	26-A38#	26-A38#	26-A86	26-A86#	26-A86#	26-B80	26-B80#	26-B80#	26-B86
	26-B86#	26-B86#												
T\$GMAN	9-18#													
T\$HILI	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#	27-21	27-21#	27-22	27-22#
	27-23	27-23#	28-18	28-18#										
T\$LAST	9-18#	28-47#												
T\$LOLI	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#	27-21	27-21#	27-22	27-22#
	27-23	27-23#	28-18	28-18#										
T\$LSYM	9-18	9-18#	12-23	13-15	18-103	18-116	18-135	18-154	18-174	18-193	18-211	18-231	18-249	19-11
	21-99	22-20	23-11	24-13	25-10	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-533	26-565
	26-570	26-616	26-643	26-644	26-695	26-748	26-773	26-798	26-799	26-827	26-944	26-;77	26-;48	26-;94
	26-;96	26-<77	26-=30	26-=32	26-=96	26->58	26-?08	26-?58	26-?94	26-@42	26-A21	26-A40	26-A58	26-A68
	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93	26-B97	26-C51	26-D05	26-D59	26-E13	26-E67	26-F21	26-F75
	26-G18	26-H72	27-25	28-20										
T\$LTNO	28-47#													
T\$NEST	9-18#	9-24	9-24	9-24#	12-9	12-9	12-9#	12-23	12-23	12-23	12-23#	13-8	13-8	13-8#
	13-15	13-15	13-15	13-15#	18-101	18-101	18-101#	18-103	18-103	18-103	18-103#	18-107	18-107	18-107#
	18-116	18-116	18-116	18-116#	18-122	18-122	18-122#	18-135	18-135	18-135	18-135#	18-141	18-141	18-141#
	18-154	18-154	18-154	18-154#	18-160	18-160	18-160#	18-174	18-174	18-174	18-174#	18-180	18-180	18-180#
	18-193	18-193	18-193	18-193#	18-199	18-199	18-199#	18-211	18-211	18-211	18-211#	18-217	18-217	18-217#
	18-231	18-231	18-231	18-231#	18-237	18-237	18-237#	18-249	18-249	18-249	18-249#	19-9	19-9	19-9#
	19-11	19-11	19-11	19-11#	20-8	20-8	20-8#	20-12	20-12	20-12	20-12#	21-8	21-8	21-8#
	21-99	21-99	21-99	21-99#	22-8	22-8	22-8#	22-20	22-20	22-20	22-20#	23-8	23-8	23-8#
	23-11	23-11	23-11	23-11#	24-8	24-8	24-8#	24-13	24-13	24-13	24-13#	25-9	25-9	25-9#
	25-10	25-10	25-10	25-10#	26-20	26-20	26-20#	26-46	26-46	26-46	26-46#	26-65	26-65	26-65#
	26-147	26-147	26-147	26-147#	26-173	26-173	26-173#	26-253	26-253	26-253	26-253#	26-267	26-267	26-267#
	26-284	26-284	26-284	26-284#	26-299	26-299	26-299#	26-317	26-317	26-317	26-317#	26-332	26-332	26-332#
	26-366	26-366	26-366	26-366#	26-384	26-384	26-384#	26-480	26-480	26-480	26-480#	26-506	26-506	26-506#
	26-510	26-510	26-510#	26-533	26-533	26-533	26-533#	26-539	26-539	26-539	26-539#	26-565	26-565	26-565#
	26-570	26-570	26-570	26-570#	26-589	26-589	26-589#	26-593	26-593	26-593	26-593#	26-616	26-616	26-616#
	26-620	26-620	26-620#	26-643	26-643	26-643	26-643#	26-644	26-644	26-644	26-644#	26-663	26-663	26-663#
	26-695	26-695	26-695	26-695#	26-720	26-720	26-720#	26-724	26-724	26-724	26-724#	26-748	26-748	26-748#
	26-752	26-752	26-752#	26-773	26-773	26-773	26-773#	26-777	26-777	26-777	26-777#	26-798	26-798	26-798#
	26-799	26-799	26-799	26-799#	26-812	26-812	26-812#	26-827	26-827	26-827	26-827#	26-850	26-850	26-850#
	26-944	26-944	26-944	26-944#	26-969	26-969	26-969#	26-;77	26-;77	26-;77	26-;77#	26-;02	26-;02	26-;02#
	26-;08	26-;08	26-;08#	26-;48	26-;48	26-;48	26-;48#	26-;54	26-;54	26-;54	26-;54#	26-;94	26-;94	26-;94#
	26-;96	26-;96	26-;96	26-;96#	26-<24	26-<24	26-<24#	26-<30	26-<30	26-<30	26-<30#	26-<77	26-<77	26-<77#
	26-<83	26-<83	26-<83#	26-=30	26-=30	26-=30	26-=30#	26-=32	26-=32	26-=32	26-=32#	26-=48	26-=48	26-=48#
	26-=96	26-=96	26-=96	26-=96#	26->13	26->13	26->13#	26->58	26->58	26->58	26->58#	26->75	26->75	26->75#
	26-?08	26-?08	26-?08	26-?08#	26-?25	26-?25	26-?25#	26-?58	26-?58	26-?58	26-?58#	26-?76	26-?76	26-?76#
	26-?94	26-?94	26-?94	26-?94#	26-@09	26-@09	26-@09#	26-@42	26-@42	26-@42	26-@42#	26-@73	26-@73	26-@73#
	26-@82	26-@82	26-@82#	26-A21	26-A21	26-A21	26-A21#	26-A26	26-A26	26-A26	26-A26#	26-A40	26-A40	26-A40#
	26-A45	26-A45	26-A45#	26-A58	26-A58	26-A58	26-A58#	26-A63	26-A63	26-A63	26-A63#	26-A68	26-A68	26-A68#
	26-A73	26-A73	26-A73#	26-A93	26-A93	26-A93	26-A93#	26-A98	26-A98	26-A98	26-A98#	26-B11	26-B11	26-B11#
	26-B16	26-B16	26-B16#	26-B28	26-B28	26-B28	26-B28#	26-B33	26-B33	26-B33	26-B33#	26-B47	26-B47	26-B47#
	26-B52	26-B52	26-B52#	26-B64	26-B64	26-B64	26-B64#	26-B69	26-B69	26-B69	26-B69#	26-B93	26-B93	26-B93#
	26-B97	26-B97	26-B97	26-B97#	26-C17	26-C17	26-C17#	26-C51	26-C51	26-C51	26-C51#	26-C71	26-C71	26-C71#

	26-D05	26-D05	26-D05	26-D05#	26-D25	26-D25	26-D25#	26-D59	26-D59	26-D59	26-D59#	26-D79	26-D79	26-D79#
	26-E13	26-E13	26-E13	26-E13#	26-E33	26-E33	26-E33#	26-E67	26-E67	26-E67	26-E67#	26-E87	26-E87	26-E87#
	26-F21	26-F21	26-F21	26-F21#	26-F41	26-F41	26-F41#	26-F75	26-F75	26-F75	26-F75#	26-F92	26-F92	26-F92#
	26-G18	26-G18	26-G18	26-G18#	26-G37	26-G37	26-G37#	26-H72	26-H72	26-H72	26-H72#	27-14	27-14	27-14#
	27-25	27-25	27-25	27-25#	28-13	28-13	28-13#	28-20	28-20	28-20	28-20#	28-45	28-45	28-45
	28-45#													
TSNSO	9-24#	28-45												
TSNS1	12-9#	12-23	13-8#	13-15	18-101#	18-103	18-107#	18-116	18-122#	18-135	18-141#	18-154	18-160#	18-174
	18-180#	18-193	18-199#	18-211	18-217#	18-231	18-237#	18-249	19-9#	19-11	20-8#	20-12	21-8#	21-99
	22-8#	22-20	23-8#	23-11	24-8#	24-13	25-9#	25-10	26-20#	26-46	26-65#	26-147	26-173#	26-253
	26-267#	26-284	26-299#	26-317	26-332#	26-366	26-384#	26-480	26-506#	26-570	26-589#	26-644	26-663#	26-695
	26-720#	26-799	26-812#	26-827	26-850#	26-944	26-969#	26-:77	26-:02#	26-:96	26-<24#	26-:32	26-:48#	26-:96
	26->13#	26->58	26->75#	26-?08	26-?25#	26-?58	26-?76#	26-?94	26-@09#	26-@42	26-@73#	26-B97	26-C17#	26-C51
	26-C71#	26-D05	26-D25#	26-D59	26-D79#	26-E13	26-E33#	26-E67	26-E87#	26-F21	26-F41#	26-F75	26-F92#	26-G18
	26-G37#	26-H72	27-14#	27-25	28-13#	28-20								
TSNS2	26-510#	26-533	26-539#	26-565	26-593#	26-616	26-620#	26-643	26-724#	26-748	26-752#	26-773	26-777#	26-798
	26-:08#	26-:48	26-:54#	26-:94	26-<30#	26-<77	26-<83#	26-:30	26-@82#	26-A21	26-A26#	26-A40	26-A45#	26-A58
	26-A63#	26-A68	26-A73#	26-A93	26-A98#	26-B11	26-B16#	26-B28	26-B33#	26-B47	26-B52#	26-B64	26-B69#	26-B93
TSPTNU	9-18#													
TSSAVL	9-18#													
TSSGL	9-18#													
TSSUBN	9-18#	26-20#	26-65#	26-173#	26-267#	26-299#	26-332#	26-384#	26-506#	26-510	26-510	26-510#	26-539	26-539
	26-539#	26-589#	26-593	26-593	26-593#	26-620	26-620	26-620#	26-663#	26-720#	26-724	26-724	26-724#	26-752
	26-752	26-752#	26-777	26-777	26-777#	26-812#	26-850#	26-969#	26-:02#	26-:08	26-:08	26-:08#	26-:54	26-:54
	26-:54#	26-<24#	26-<30	26-<30	26-<30#	26-<83	26-<83	26-<83#	26-:48#	26->13#	26->75#	26-?25#	26-?76#	26-@09#
	26-@73#	26-@82	26-@82	26-@82#	26-A26	26-A26	26-A26#	26-A45	26-A45	26-A45#	26-A63	26-A63	26-A63#	26-A73
	26-A73	26-A73#	26-A98	26-A98	26-A98#	26-B16	26-B16	26-B16#	26-B33	26-B33	26-B33#	26-B52	26-B52	26-B52#
	26-B69	26-B69	26-B69#	26-C17#	26-C71#	26-D25#	26-D79#	26-E33#	26-E87#	26-F41#	26-F92#	26-G37#		
TSAGL	9-18#													
TSAGN	9-18#	12-9	12-9	12-9#	13-8	13-8	13-8#	18-101	18-101	18-101#	18-107	18-107	18-107#	18-122
	18-122	18-122#	18-141	18-141	18-141#	18-160	18-160	18-160#	18-180	18-180	18-180#	18-199	18-199	18-199#
	18-217	18-217	18-217#	18-237	18-237	18-237#	19-9	19-9	19-9#	20-8	20-8	20-8#	21-8	21-8
	21-8#	22-8	22-8	22-8#	23-8	23-8	23-8#	24-8	24-8	24-8#	25-9	25-9	25-9#	26-20
	26-20	26-20#	26-65	26-65	26-65#	26-173	26-173	26-173#	26-267	26-267	26-267#	26-299	26-299	26-299#
	26-332	26-332	26-332#	26-384	26-384	26-384#	26-506	26-506	26-506#	26-510	26-510	26-510#	26-539	26-539
	26-539#	26-589	26-589	26-589#	26-593	26-593	26-593#	26-620	26-620	26-620#	26-663	26-663	26-663#	26-720
	26-720	26-720#	26-724	26-724	26-724#	26-752	26-752	26-752#	26-777	26-777	26-777#	26-812	26-812	26-812#
	26-850	26-850	26-850#	26-969	26-969	26-969#	26-:02	26-:02	26-:02#	26-:08	26-:08	26-:08#	26-:54	26-:54
	26-:54#	26-<24	26-<24	26-<24#	26-<30	26-<30	26-<30#	26-<83	26-<83	26-<83#	26-:48	26-:48	26-:48#	26->13
	26->13	26->13#	26->75	26->75	26->75#	26-?25	26-?25	26-?25#	26-?76	26-?76	26-?76#	26-@09	26-@09	26-@09#
	26-@73	26-@73	26-@73#	26-@82	26-@82	26-@82#	26-A26	26-A26	26-A26#	26-A45	26-A45	26-A45#	26-A63	26-A63
	26-A63#	26-A73	26-A73	26-A73#	26-A98	26-A98	26-A98#	26-B16	26-B16	26-B16#	26-B33	26-B33	26-B33#	26-B52
	26-B52	26-B52#	26-B69	26-B69	26-B69#	26-C17	26-C17	26-C17#	26-C71	26-C71	26-C71#	26-D25	26-D25	26-D25#
	26-D79	26-D79	26-D79#	26-E33	26-E33	26-E33#	26-E87	26-E87	26-E87#	26-F41	26-F41	26-F41#	26-F92	26-F92
	26-F92#	26-G37	26-G37	26-G37#	27-14	27-14	27-14#	28-13	28-13	28-13#				
TSTEMP	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#	11-8#
	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154	18-154#	18-174	18-174#	18-193
	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	20-12	20-12#	21-94	21-94	21-94
	21-94#	21-94#	21-94#	21-99	21-99#	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46
	26-46#	26-147	26-147#	26-253	26-253#	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533
	26-533#	26-565	26-565#	26-570	26-570#	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-748

	26-748#	26-773	26-773#	26-798	26-798#	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:38
	26-:38#	26-:44	26-:44#	26-:48	26-:48#	26-:84	26-:84#	26-:90	26-:90#	26-:94	26-:94#	26-:96	26-:96#	26-<77
	26-<77#	26-=30	26-=30#	26-=32	26-=32#	26-=96	26-=96#	26->58	26->58#	26-?08	26-?08#	26-?58	26-?58#	26-?94
	26-?94#	26-@42	26-@42#	26-@93	26-@93#	26-A03	26-A03#	26-A21	26-A21#	26-A38	26-A38#	26-A40	26-A40#	26-A58
	26-A58#	26-A68	26-A68#	26-A86	26-A86#	26-A93	26-A93#	26-B11	26-B11#	26-B28	26-B28#	26-B47	26-B47#	26-B64
	26-B64#	26-B80	26-B80#	26-B86	26-B86#	26-B93	26-B93#	26-B97	26-B97#	26-C51	26-C51#	26-D05	26-D05#	26-D59
	26-D59#	26-E13	26-E13#	26-E67	26-E67#	26-F21	26-F21#	26-F75	26-F75#	26-G18	26-G18#	26-H72	26-H72#	27-16
	27-16	27-16	27-16#	27-16#	27-16#	27-17	27-17#	27-17	27-17#	27-17#	27-17#	27-18	27-18#	27-18
	27-18#	27-18#	27-18#	27-19	27-19#	27-19	27-19#	27-19#	27-19#	27-20	27-20#	27-20	27-20#	27-20#
	27-20#	27-21	27-21#	27-21	27-21#	27-21#	27-21#	27-22	27-22#	27-22	27-22#	27-22#	27-22#	27-23
	27-23	27-23	27-23#	27-23#	27-23#	27-25	27-25#	28-15	28-15#	28-15	28-15#	28-15#	28-15#	28-16
	28-16	28-16	28-16#	28-16#	28-16#	28-17	28-17#	28-17	28-17#	28-17#	28-17#	28-18	28-18#	28-18
	28-18#	28-18#	28-18#	28-20	28-20#	28-45	28-45#							
T\$TEST	9-18#	26-20	26-20#	26-20#	26-65	26-65#	26-65#	26-173	26-173#	26-173#	26-267	26-267#	26-267#	26-299
	26-299	26-299#	26-332	26-332#	26-332#	26-384	26-384#	26-384#	26-506	26-506#	26-506#	26-510	26-539	26-589
	26-589	26-589#	26-593	26-620	26-663	26-663#	26-663#	26-720	26-720#	26-720#	26-724	26-752	26-777	26-812
	26-812	26-812#	26-850	26-850	26-850#	26-969	26-969#	26-969#	26-:02	26-:02#	26-:02#	26-:08	26-:54	26-<24
	26-<24	26-<24#	26-<30	26-<83	26-=48	26-=48#	26-=48#	26->13	26->13#	26->13#	26->75	26->75#	26->75#	26-?25
	26-?25	26-?25#	26-?76	26-?76	26-?76#	26-@09	26-@09#	26-@09#	26-@73	26-@73#	26-@73#	26-@82	26-A26	26-A45
	26-A63	26-A73	26-A98	26-B16	26-B33	26-B52	26-B69	26-C17	26-C17#	26-C17#	26-C71	26-C71#	26-C71#	26-D25
	26-D25	26-D25#	26-D79	26-D79	26-D79#	26-E33	26-E33#	26-E33#	26-E87	26-E87#	26-E87#	26-F41	26-F41#	26-F41#
	26-F92	26-F92#	26-F92#	26-G37	26-G37#	26-G37#	28-47							
T\$TSTM	9-18#	17-277	17-283	17-293	17-299	17-406	17-412	17-570	17-576	17-586	17-592	17-627	17-633	17-667
	17-673	17-681	17-687	17-833	17-838	17-857	17-863	17-872	17-878	17-887	17-893	17-902	17-908	17-:61
	17-:66	17-:75	17-<54	17-<70	17-=06	17-=12	18-102	18-103	18-108	18-109	18-110	18-111	18-112	18-113
	18-114	18-115	18-116	18-123	18-124	18-125	18-126	18-127	18-128	18-129	18-130	18-131	18-132	18-133
	18-134	18-135	18-142	18-143	18-144	18-145	18-146	18-147	18-148	18-149	18-150	18-151	18-152	18-153
	18-154	18-161	18-162	18-163	18-164	18-165	18-166	18-167	18-168	18-169	18-170	18-171	18-172	18-173
	18-174	18-181	18-182	18-183	18-184	18-185	18-186	18-187	18-188	18-189	18-190	18-191	18-192	18-193
	18-200	18-201	18-202	18-203	18-204	18-205	18-206	18-207	18-208	18-209	18-210	18-211	18-218	18-219
	18-220	18-221	18-222	18-223	18-224	18-225	18-226	18-227	18-228	18-229	18-230	18-231	18-238	18-239
	18-240	18-241	18-242	18-243	18-244	18-245	18-246	18-247	18-248	18-249	19-11	21-25	21-28	21-31
	21-34	21-52	21-81	21-85	21-86	21-90	21-94	21-99	22-10	22-17	22-20	23-11	24-10	24-12
	24-13	25-10	26-46	26-105	26-145	26-147	26-224	26-251	26-253	26-284	26-317	26-364	26-366	26-419
	26-458	26-471	26-480	26-510	26-533	26-539	26-565	26-570	26-593	26-616	26-620	26-643	26-644	26-695
	26-724	26-735	26-737	26-746	26-748	26-752	26-762	26-771	26-773	26-777	26-787	26-796	26-798	26-799
	26-823	26-825	26-827	26-944	26-:77	26-:08	26-:37	26-:38	26-:43	26-:44	26-:48	26-:54	26-:83	26-:84
	26-:89	26-:90	26-:94	26-:96	26-<30	26-<77	26-<83	26-=30	26-=32	26-=72	26-=94	26-=96	26->34	26->49
	26->58	26->99	26-?08	26-?49	26-?58	26-?94	26-@42	26-@82	26-@92	26-@93	26-A02	26-A03	26-A21	26-A26
	26-A37	26-A38	26-A40	26-A45	26-A56	26-A58	26-A63	26-A68	26-A73	26-A85	26-A86	26-A91	26-A93	26-A98
	26-B09	26-B11	26-B16	26-B26	26-B28	26-B33	26-B45	26-B47	26-B52	26-B62	26-B64	26-B69	26-B79	26-B80
	26-B85	26-B86	26-B91	26-B93	26-B97	26-C42	26-C51	26-C96	26-D05	26-D50	26-D59	26-E04	26-E13	26-E58
	26-E67	26-F11	26-F21	26-F65	26-F75	26-G18	26-G71	26-G83	26-H45	26-H72				
T\$TSTS	9-18#	26-20#	26-65#	26-173#	26-267#	26-299#	26-332#	26-384#	26-506#	26-589#	26-663#	26-720#	26-812#	26-850#
	26-969#	26-:02#	26-<24#	26-=48#	26->13#	26->75#	26-?25#	26-?76#	26-@09#	26-@73#	26-C17#	26-C71#	26-D25#	26-D79#
	26-E33#	26-E87#	26-F41#	26-F92#	26-G37#									
T1	11-8	26-20#												
T10	11-8	26-663#												
T11	11-8	26-720#												
T11.1	26-724#													
T11.2	26-752#													
T11.3	26-777#													
T12	11-8	26-812#												
T13	11-8	26-850#												
T14	11-8	26-969#												
T15	11-8	26-:02#												
T15.1	26-:08#													

T15.2	26-.54#
T16	11-8 26-<24#
T16.1	26-<30#
T16.2	26-<83#
T17	11-8 26-=48#
T18	11-8 26->13#
T19	11-8 26->75#
T2	11-8 26-65#
T20	11-8 26-?25#
T21	11-8 26-?76#
T22	11-8 26-a09#
T23	11-8 26-a73#
T23.1	26-a82#
T23.10	26-B69#
T23.2	26-A26#
T23.3	26-A45#
T23.4	26-A63#
T23.5	26-A73#
T23.6	26-A98#
T23.7	26-B16#
T23.8	26-B33#
T23.9	26-B52#
T24	11-8 26-C17#
T25	11-8 26-C71#
T26	11-8 26-D25#
T27	11-8 26-D79#
T28	11-8 26-E33#
T29	11-8 26-E87#
T3	11-8 26-173#
T30	11-8 26-F41#
T31	11-8 26-F92#
T32	11-8 26-G37#
T4	11-8 26-267#
T5	11-8 26-299#
T6	11-8 26-332#
T7	11-8 26-384#
T8	11-8 26-506#
T8.1	26-510#
T8.2	26-539#
T9	11-8 26-589#
T9.1	26-593#
T9.2	26-620#
TCCHEK	14-327# 17-;24 26-a77
TCOUNT	13-13# 26-=64 26->41 26->91 26-?41 26-C34 26-C88 26-D42 26-D96 26-E50 26-F03 26-F57
TEOM	14-220# 26-438 26-444
TERR	14-217#
TEST	14-258# 17-;23 26-=54 26->19 26->81 26-?31
TESTMD	14-174# 17-<48 17-<64 26-B41
TIMCNT	28-18 28-25#
TIMFLG	15-24#
TMP0	15-90# 17-229* 17-232* 18-125 18-164 18-184 18-240
TMP1	15-91# 17-247* 17-248* 18-125 18-164 18-184 18-240
TMP2	15-92#
TMP3	15-93#
TMP4	15-94#
TMP5	15-95#
TMP6	15-96#

TMP7	15-97#													
TSOM	14-221#	26-405	26-226											
TSTCON	15-66#	17-:26	17-:30	17-:51	17-:81	17-:93	17-<12	17-<28	17-<41	17-<57	21-74*	26-E34*	26-E88*	26-F42*
TSTNUM	15-52#	17-:66	26-249*	26->14*	26->76*	26-226*	26-274*	26-C18*	26-C72*	26-D26*	26-D80*			
TX0	14-54#	14-212#												
TX1	14-53#	14-211#												
TX2	14-52#	14-210#												
TX3	14-51#	14-209#												
TX4	14-50#	14-208#												
TX5	14-49#	14-207#												
TX6	14-48#	14-206#												
TX7	14-47#	14-205#												
TXAB	14-219#													
TXABT	14-279#	15-242												
TXCHAR	17-510#	26-977	26-:12	26-:47										
TXDATA	14-171#	26-:33	26-:39	26-:79	26-:85	26-A15	26-A17							
TXEN	14-86#	14-158#	26-304	26-397	26-399	26-221	26-223							
TXEOM	14-280#	15-226	15-227	15-228	15-229	15-235	15-236	17-954	26-28	26-522	26-554	26-605	26-632	26-:17
	26-:63	26-<42	26-<95	26-G01	26-G49	26-G55								
TXGA	14-218#													
TXGOA	14-278#													
TXLENO	14-266#	17-817	26-475	26-:20	26-:66	26-<45	26-<98	26-E98	26-F52					
TXLEN1	14-265#	17-817	26-389	26-:20	26-:66	26-<45	26-<98	26-E98	26-F52					
TXLEN2	14-264#	17-817	26-:20	26-:66	26-<45	26-<98	26-E98	26-F52						
TXSOM	14-281#	15-219	15-220	15-231	15-232	15-243	15-244	17-448	17-940	17-:58	26-<36	26-<89	26-G58	
TXWORD	15-43#	17-344*	17-346	17-349	17-448*	17-449*	17-514*	17-945*	17-946*	17-947	17-954*	17-:11*	17-:36*	17-:37*
	17-:85*	26-28*												
TYPEY	21-94	21-104#												
UAM	14-20#													
UNIT	15-51#													
UNRR	14-128#	17-829												
UPBITS	15-100#													
V35	14-254#	14-259	17-:23	17-:40	17-:84	17-:86	17-:88	17-<22	26-254	26->18	26->81	26-?31		
VECTOR	27-17	27-28#												
WAIT50	17-316#	17-458	17-516	17-748	17-:15	17-:41	17-:91	26-280						
WAX	14-89#	14-161#	17-205											
WAX15	15-30#	17-195*	17-196	17-447*	17-449	17-974*	17-983*	26-404*	26-:19*	26-:65*	26-<44*	26-<97*	26-225*	26-230*
WAX16	15-31#	17-199*	17-200	17-450*	17-975*	17-985*	17-986	26-405*	26-424*	26-432*	26-435*	26-438*	26-444*	26-:20*
	26-:66*	26-<45*	26-<46	26-<98*	26-<99	26-226*	26-231*							
WRDYTO	14-343#	17-188	17-214											
WRIBYT	15-27#	17-41*	17-89*	17-90	17-152*	17-153*	17-154*	17-155*	17-190*	17-191*	17-192*	17-196*	17-200*	17-203*
	17-204*	17-205*	17-206*	17-346*	17-349*	17-454*	17-784*	17-785*	17-791*	17-979*	17-:83*	17-<14*	17-<39*	17-<43*
	17-<59*	17-<74*	26-98*	26-217*	26-345*	26-397*	26-401*	26-?82*	26-218*	26-221*	26-A04*	26-A47*	26-A66*	26-A76*
	26-800*	26-818*	26-835*	26-854*	26-H35*									
WRITAX	17-186#	17-451	17-976	17-988	26-406	26-421	26-425	26-428	26-433	26-439	26-445	26-:21	26-:67	26-<47
	26-200	26-227	26-232	26-235										
WRITLU	17-42	17-83#	17-156	17-193	17-197	17-201	17-207	17-347	17-350	17-455	17-786	17-792	17-980	17-<44
	17-<60	17-<75	26-99	26-218	26-346	26-398	26-402	26-?83	26-219	26-222	26-A05	26-A49	26-A67	26-A77
	26-802	26-820	26-837	26-856	26-H36									
XSALWA	9-18#													
XSFALS	9-18#													
XSOFFS	9-18#													
XSTRUE	9-18#													
XYZ	14-252#	14-259	17-:23	17-:45	17-<05	17-<11	17-<31	26-254	26->19	26->80	26-?31			

[illegible]

ENDSFT	1-568#	9-18#	28-20											
ENDSRV	1-580#	9-18#												
ENDSUB	1-596#	9-18#	26-533	26-565	26-616	26-643	26-748	26-773	26-798	26-;48	26-;94	26-<77	26-=30	26-A21
	26-A40	26-A58	26-A68	26-A93	26-B11	26-B28	26-B47	26-B64	26-B93					
ENDSW	1-614#	9-18#	13-15											
ENDTST	1-624#	9-18#	26-46	26-147	26-253	26-284	26-317	26-366	26-480	26-570	26-644	26-695	26-799	26-827
	26-944	26-;77	26-;96	26-=32	26-=96	26->58	26-?08	26-?58	26-?94	26-@42	26-B97	26-C51	26-D05	26-D59
	26-E13	26-E67	26-F21	26-F75	26-G18	26-H72								
EQUALS	1-642#	9-18#	14-20											
ERRDF	1-714#	9-18#	17-277	17-283	17-293	17-299	17-406	17-412	17-570	17-576	17-586	17-592	17-627	17-633
	17-667	17-673	17-681	17-687	17-833	17-838	17-857	17-863	17-872	17-878	17-887	17-893	17-902	17-908
	17-<54	17-<70	17-=06	17-=12	26-105	26-145	26-224	26-251	26-364	26-419	26-458	26-471	26-746	26-771
	26-796	26-;37	26-;43	26-;83	26-;89	26-=72	26-=94	26->34	26->49	26->99	26-?49	26-@92	26-A02	26-A37
	26-A56	26-A85	26-A91	26-B09	26-B26	26-B45	26-B62	26-B79	26-B85	26-B91	26-C42	26-C96	26-D50	26-E04
	26-E58	26-F11	26-F65	26-G71	26-G83	26-H45								
ERRHRD	1-718#	9-18#												
ERROR	1-722#	9-18#												
ERRSF	1-726#	9-18#												
ERRSOF	1-730#	9-18#												
ERRTBL	1-734#	9-18#												
ESCAPE	1-744#	9-18#	26-;38	26-;44	26-;84	26-;90	26-@93	26-A03	26-A38	26-A86	26-B80	26-B86		
EXIT	1-771#	9-18#												
FEQUAL	1-810#	9-18#												
GETBYT	1-824#	9-18#												
GETPRI	1-834#	9-18#												
GETWOR	1-829#	9-18#												
GMANIA	1-839#	9-18#												
GMANID	1-848#	9-18#												
GMANIL	1-859#	9-18#	21-94											
GPHARD	1-868#	9-18#	21-52											
GPRMA	1-874#	9-18#	27-16	27-17										
GPRMD	1-903#	9-18#	27-18	27-19	27-20	27-21	27-22	27-23	28-18					
GPRML	1-934#	9-18#	21-94	21-94#	28-15	28-16	28-17							
HEADER	1-954#	9-18#	10-17											
INLOOP	1-962#	9-18#												
IOSETU	1-966#	9-18#												
IOSTAR	1-974#	9-18#												
KY11	1-982#	9-18#												
LASTAD	1-;47#	9-18#	28-47											
MSBYTE	1-D00#	9-18#	10-17	10-17	10-17	10-17#								
MSCHEC	1-E18#	9-18#												
MSCNTO	1-E82#	9-18#	21-94	21-94#	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#
	27-21	27-21#	27-22	27-22#	27-23	27-23#	28-15	28-15#	28-16	28-16#	28-17	28-17#	28-18	28-18#
MSCOUN	1-D66#	9-18#	17-;61	17-;61#	17-;66	17-;66#	17-;75	17-;75#	18-102	18-102	18-102#	18-108	18-108	18-108#
	18-109	18-109#	18-110	18-110	18-110#	18-111	18-111	18-111#	18-112	18-112	18-112#	18-113	18-113	18-113#
	18-113	18-113#	18-114	18-114#	18-115	18-115	18-115	18-115#	18-115	18-115#	18-123	18-123	18-123#	18-124
	18-125	18-125	18-125#	18-126	18-126	18-126#	18-127	18-127	18-127#	18-128	18-128	18-128	18-128	18-128#
	18-129	18-129#	18-130	18-130	18-130	18-130#	18-131	18-131	18-131#	18-132	18-132	18-132	18-132	18-132#
	18-132#	18-133	18-133#	18-134	18-134	18-134#	18-134	18-134#	18-142	18-142#	18-143	18-143	18-143#	18-144
	18-144#	18-145	18-145	18-145#	18-146	18-146	18-146#	18-147	18-147	18-147	18-147	18-147	18-147#	18-148
	18-149	18-149	18-149	18-149	18-149#	18-150	18-150	18-150#	18-151	18-151	18-151	18-151	18-151#	18-152
	18-152#	18-153	18-153	18-153	18-153	18-153#	18-161	18-161	18-161#	18-162	18-162	18-162#	18-163	18-163#
	18-164	18-164	18-164#	18-165	18-165	18-165#	18-166	18-166	18-166#	18-167	18-167	18-167	18-167	18-167#
	18-168	18-168#	18-169	18-169	18-169	18-169#	18-169	18-169#	18-170	18-170	18-170#	18-171	18-171	18-171#
	18-171#	18-172	18-172#	18-173	18-173	18-173#	18-173	18-173#	18-181	18-181#	18-182	18-182	18-182#	18-183
	18-183#	18-184	18-184	18-184#	18-185	18-185	18-185#	18-186	18-186	18-186	18-186	18-186#	18-187	18-187#
	18-188	18-188	18-188	18-188	18-188#	18-189	18-189	18-189#	18-190	18-190	18-190	18-190	18-190#	18-191

	18-191#	18-192	18-192	18-192	18-192	18-192#	18-200	18-200	18-200#	18-201	18-201#	18-202	18-202	18-202#
	18-203	18-203	18-203#	18-204	18-204	18-204	18-204	18-204#	18-205	18-205#	18-206	18-206	18-206	18-206
	18-206#	18-207	18-207	18-207#	18-208	18-208	18-208	18-208#	18-209	18-209#	18-210	18-210	18-210	18-210
	18-210	18-210#	18-218	18-218#	18-219	18-219	18-219#	18-220	18-220#	18-221	18-221	18-221#	18-222	18-222
	18-222#	18-223	18-223	18-223#	18-224	18-224	18-224	18-224#	18-225	18-225#	18-226	18-226	18-226	18-226
	18-226	18-226#	18-227	18-227#	18-228	18-228	18-228	18-228#	18-229	18-229#	18-230	18-230	18-230	18-230
	18-230	18-230	18-230#	18-238	18-238	18-238#	18-239	18-239#	18-240	18-240#	18-241	18-241	18-241#	18-241#
	18-242	18-242	18-242	18-242#	18-243	18-243	18-243#	18-244	18-244	18-244	18-244	18-244#	18-245	18-245
	18-245#	18-246	18-246	18-246	18-246	18-246#	18-247	18-247#	18-248	18-248	18-248	18-248	18-248#	21-85
	21-85#	21-90	21-90#	24-12	24-12#	26-735	26-735#	26-737	26-737#	26-762	26-762#	26-787	26-787#	26-823
	26-823#	26-825	26-825#											
MSDATA	1-867#	9-18#	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17#	10-17#	16-12
	16-12#	16-17	16-17#											
MSDECR	1-D29#	9-18#	12-23	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154	18-154#
	18-174	18-174#	18-193	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	20-12	20-12#
	21-99	21-99#	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-147	26-147#
	26-253	26-253#	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533	26-533#	26-565	26-565#
	26-570	26-570#	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-748	26-748#	26-773	26-773#
	26-798	26-798#	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:48	26-:48#	26-:94	26-:94#
	26-:96	26-:96#	26-<77	26-<77#	26-=30	26-=30#	26-=32	26-=32#	26-=96	26-=96#	26->58	26->58#	26-?08	26-?08#
	26-?58	26-?58#	26-?94	26-?94#	26-@42	26-@42#	26-A21	26-A21#	26-A40	26-A40#	26-A58	26-A58#	26-A68	26-A68#
	26-A93	26-A93#	26-B11	26-B11#	26-B28	26-B28#	26-B47	26-B47#	26-B64	26-B64#	26-B93	26-B93#	26-B97	26-B97#
	26-C51	26-C51#	26-D05	26-D05#	26-D59	26-D59#	26-E13	26-E13#	26-E67	26-E67#	26-F21	26-F21#	26-F75	26-F75#
	26-G18	26-G18#	26-H72	26-H72#	27-25	27-25#	28-20	28-20#	28-45	28-45#				
MSDEFA	1-E70#	9-18#	21-94	21-94#	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20#
	27-21	27-21#	27-22	27-22#	27-23	27-23#	28-15	28-15#	28-16	28-16#	28-17	28-17#	28-18	28-18#
MSENDE	1-D74#	9-18#	12-23#	13-15#	18-103#	18-116#	18-135#	18-154#	18-174#	18-193#	18-211#	18-231#	18-249#	19-11#
	21-99#	22-20#	23-11#	24-13#	25-10#	26-46#	26-147#	26-253#	26-284#	26-317#	26-366#	26-480#	26-533#	26-565#
	26-570#	26-616#	26-643#	26-644#	26-695#	26-748#	26-773#	26-798#	26-799#	26-827#	26-944#	26-:77#	26-:48#	26-:94#
	26-:96#	26-<77#	26-=30#	26-=32#	26-=96#	26->58#	26-?08#	26-?58#	26-?94#	26-@42#	26-A21#	26-A40#	26-A58#	26-A68#
	26-A93#	26-B11#	26-B28#	26-B47#	26-B64#	26-B93#	26-B97#	26-C51#	26-D05#	26-D59#	26-E13#	26-E67#	26-F21#	26-F75#
	26-G18#	26-H72#	27-25#	28-20#	28-45#									
MSERRI	1-@49#	9-18#	17-277	17-277#	17-283	17-283#	17-293	17-293#	17-299	17-299#	17-406	17-406#	17-412	17-412#
	17-570	17-570#	17-576	17-576#	17-586	17-586#	17-592	17-592#	17-627	17-627#	17-633	17-633#	17-667	17-667#
	17-673	17-673#	17-681	17-681#	17-687	17-687#	17-833	17-833#	17-838	17-838#	17-857	17-857#	17-863	17-863#
	17-872	17-872#	17-878	17-878#	17-887	17-887#	17-893	17-893#	17-902	17-902#	17-908	17-908#	17-<54	17-<54#
	17-<70	17-<70#	17-=06	17-=06#	17-=12	17-=12#	26-105	26-105#	26-145	26-145#	26-224	26-224#	26-251	26-251#
	26-364	26-364#	26-419	26-419#	26-458	26-458#	26-471	26-471#	26-746	26-746#	26-771	26-771#	26-796	26-796#
	26-:37	26-:37#	26-:43	26-:43#	26-:83	26-:83#	26-:89	26-:89#	26-=72	26-=72#	26-=94	26-=94#	26->34	26->34#
	26->49	26->49#	26->99	26->99#	26-?49	26-?49#	26-@92	26-@92#	26-A02	26-A02#	26-A37	26-A37#	26-A56	26-A56#
	26-A85	26-A85#	26-A91	26-A91#	26-B09	26-B09#	26-B26	26-B26#	26-B45	26-B45#	26-B62	26-B62#	26-B79	26-B79#
	26-B85	26-B85#	26-B91	26-B91#	26-C42	26-C42#	26-C96	26-C96#	26-D50	26-D50#	26-E04	26-E04#	26-E58	26-E58#
	26-F11	26-F11#	26-F65	26-F65#	26-G71	26-G71#	26-G83	26-G83#	26-H45	26-H45#				
MSESCA	1-D06#	9-18#	26-:38	26-:38#	26-:44	26-:44#	26-:84	26-:84#	26-:90	26-:90#	26-@93	26-@93#	26-A03	26-A03#
	26-A38	26-A38#	26-A86	26-A86#	26-B80	26-B80#	26-B86	26-B86#						
MSESCS	1-D10#	9-18#	26-:38#	26-:44#	26-:84#	26-:90#	26-@93#	26-A03#	26-A38#	26-A86#	26-B80#	26-B86#		
MSEXCP	1-E01#	9-18#	27-16	27-16	27-16#	27-17	27-17	27-17#	27-18	27-18	27-18#	27-19	27-19	27-19#
	27-20	27-20	27-20#	27-21	27-21	27-21#	27-22	27-22	27-22#	27-23	27-23	27-23#	28-18	28-18
	28-18#													
MSEXIT	1-D14#	9-18#												
MSEXSE	1-D22#	9-18#												
MSEXTJ	1-D18#	9-18#												
MSGEN	1-D38#	9-18#	9-24	9-24#	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17

	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	12-9#	12-9#	12-23	12-23#	13-8	13-8	13-8#	13-15	13-15#	16-12	16-12#	16-17	16-17#
	18-101	18-101#	18-103	18-103#	18-107	18-107#	18-116	18-116#	18-122	18-122#	18-135	18-135#	18-141
	18-154	18-154#	18-160	18-160#	18-174	18-174#	18-180	18-180#	18-193	18-193#	18-199	18-199#	18-211
	18-217	18-217#	18-231	18-231#	18-237	18-237#	18-249	18-249#	19-9	19-9#	19-11	19-11#	20-8
	21-8	21-8#	21-94	21-94#	21-99	21-99#	22-8	22-8#	22-20	22-20#	23-8	23-8#	23-11
	24-8	24-8#	24-13	24-13#	25-9	25-9#	25-10	25-10#	26-20	26-20#	26-46	26-46#	26-65
	26-147	26-147#	26-173	26-173#	26-253	26-253#	26-267	26-267#	26-284	26-284#	26-299	26-299#	26-317
	26-332	26-332#	26-366	26-366#	26-384	26-384#	26-480	26-480#	26-506	26-506#	26-510	26-510#	26-533
	26-539	26-539#	26-565	26-565#	26-570	26-570#	26-589	26-589#	26-593	26-593#	26-616	26-616#	26-620
	26-643	26-643#	26-644	26-644#	26-663	26-663#	26-695	26-695#	26-720	26-720#	26-724	26-724#	26-748
	26-752	26-752#	26-773	26-773#	26-777	26-777#	26-798	26-798#	26-799	26-799#	26-812	26-812#	26-827
	26-850	26-850#	26-944	26-944#	26-969	26-969#	26-:77	26-:77#	26-:02	26-:02#	26-:08	26-:08#	26-:48
	26-:54	26-:54#	26-:94	26-:94#	26-:96	26-:96#	26-<24	26-<24#	26-<30	26-<30#	26-<77	26-<77#	26-<83
	26-:30	26-:30#	26-:32	26-:32#	26-:48	26-:48#	26-:96	26-:96#	26->13	26->13#	26->58	26->58#	26->75
	26-?08	26-?08#	26-?25	26-?25#	26-?58	26-?58#	26-?76	26-?76#	26-?94	26-?94#	26-209	26-209#	26-242
	26-273	26-273#	26-282	26-282#	26-A21	26-A21#	26-A26	26-A26#	26-A40	26-A40#	26-A45	26-A45#	26-A58
	26-A63	26-A63#	26-A68	26-A68#	26-A73	26-A73#	26-A93	26-A93#	26-A98	26-A98#	26-B11	26-B11#	26-B16
	26-B28	26-B28#	26-B33	26-B33#	26-B47	26-B47#	26-B52	26-B52#	26-B64	26-B64#	26-B69	26-B69#	26-B93
	26-B97	26-B97#	26-C17	26-C17#	26-C51	26-C51#	26-C71	26-C71#	26-D05	26-D05#	26-D25	26-D25#	26-D59
	26-D79	26-D79#	26-E13	26-E13#	26-E33	26-E33#	26-E67	26-E67#	26-E87	26-E87#	26-F21	26-F21#	26-F41
	26-F75	26-F75#	26-F92	26-F92#	26-G18	26-G18#	26-G37	26-G37#	26-H72	26-H72#	27-14	27-14#	27-25
	28-13	28-13#	28-20	28-20#	28-47	28-47#							
MSGENB	1-C38#	9-18#	21-94	21-94#									
MSGETS	1-D35#	9-18#	12-23	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154
	18-174	18-174#	18-193	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	20-12
	21-99	21-99#	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-147
	26-253	26-253#	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533	26-533#	26-565
	26-570	26-570#	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-748	26-748#	26-773
	26-798	26-798#	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:48	26-:48#	26-:94
	26-:96	26-:96#	26-<77	26-<77#	26-:30	26-:30#	26-:32	26-:32#	26-:96	26-:96#	26->58	26->58#	26-?08
	26-?58	26-?58#	26-?94	26-?94#	26-242	26-242#	26-A21	26-A21#	26-A40	26-A40#	26-A58	26-A58#	26-A68
	26-A93	26-A93#	26-B11	26-B11#	26-B28	26-B28#	26-B47	26-B47#	26-B64	26-B64#	26-B93	26-B93#	26-B97
	26-C51	26-C51#	26-D05	26-D05#	26-D59	26-D59#	26-E13	26-E13#	26-E67	26-E67#	26-F21	26-F21#	26-F75
	26-G18	26-G18#	26-H72	26-H72#	27-25	27-25#	28-20	28-20#	28-45	28-45#			
MSGETT	1-B77#	9-18#	26-:38#	26-:44#	26-:84#	26-:90#	26-293#	26-A03#	26-A38#	26-A86#	26-B80#	26-B86#	
MSGNGB	1-C02#	9-18#	9-24	9-24#	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	12-9#	13-8	13-8	13-8#	16-12	16-12#	16-17	16-17#	18-101	18-101#	18-107	18-107#	18-122
	18-141	18-141#	18-160	18-160#	18-180	18-180#	18-199	18-199#	18-217	18-217#	18-237	18-237#	19-9
	20-8	20-8#	21-8	21-8#	22-8	22-8#	23-8	23-8#	24-8	24-8#	25-9	25-9#	27-14
	28-13	28-13#	28-47	28-47#									
MSGNIN	1-D49#	9-18#	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17	10-17
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#	10-17#
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8

[illegible]

18-247	18-247	18-247#	18-247#	18-247#	18-247#	18-247#	18-247#	18-248	18-248	18-248	18-248	18-248	18-248	18-248
18-248	18-248	18-248#	18-248#	18-248#	18-248#	18-248#	18-248#	18-248#	18-248#	18-248#	18-249	18-249#	19-11	19-11#
21-25	21-25	21-25#	21-25#	21-26	21-26#	21-28	21-28	21-28#	21-28#	21-28#	21-29	21-29#	21-31	21-31#
21-31#	21-31#	21-32	21-32#	21-34	21-34#	21-34#	21-34#	21-35	21-35#	21-35#	21-52	21-52#	21-52	21-52#
21-52#	21-52#	21-53	21-53#	21-81	21-81#	21-83	21-83#	21-85	21-85#	21-85#	21-85	21-85#	21-85	21-85#
21-85#	21-85#	21-85#	21-86	21-86#	21-90	21-90	21-90	21-90	21-90#	21-90#	21-90	21-90#	21-90#	21-90#
21-90#	21-90#	21-94	21-94#	21-94	21-94#	21-94	21-94#	21-94#	21-94#	21-94#	21-94#	21-94#	21-99	21-99#
22-10	22-10	22-10#	22-10#	22-17	22-17#	22-17#	22-17#	22-20	22-20#	22-20#	23-1i	23-11#	24-10	24-10#
24-12	24-12	24-12	24-12#	24-12	24-12#	24-12#	24-12#	24-12#	24-12#	24-12#	24-12#	24-13	24-13#	25-10
25-10#	26-46	26-46#	26-105	26-105	26-105#	26-105	26-105#	26-105#	26-105#	26-105#	26-105#	26-105#	26-145	26-145#
26-145	26-145	26-145#	26-145#	26-145#	26-145#	26-145#	26-145#	26-147	26-147#	26-224	26-224	26-224	26-224	26-224#
26-224#	26-224#	26-224#	26-224#	26-251	26-251#	26-251	26-251#	26-251#	26-251#	26-251#	26-251#	26-251#	26-251#	26-253
26-253#	26-284	26-284#	26-317	26-317#	26-364	26-364	26-364#	26-364	26-364#	26-364#	26-364#	26-364#	26-364#	26-364#
26-366	26-366#	26-419	26-419#	26-419	26-419#	26-419#	26-419#	26-419#	26-419#	26-419#	26-419#	26-458	26-458	26-458#
26-458	26-458#	26-458#	26-458#	26-458#	26-458#	26-471	26-471#	26-471	26-471#	26-471#	26-471#	26-471#	26-471#	26-471#
26-471#	26-480	26-480#	26-510	26-510#	26-533	26-533#	26-533#	26-539	26-539#	26-539#	26-565	26-565#	26-570	26-570#
26-593#	26-616	26-616#	26-620	26-620#	26-643	26-643#	26-643#	26-644	26-644#	26-644#	26-695	26-695#	26-724	26-724#
26-735	26-735	26-735#	26-735	26-735#	26-735#	26-735#	26-735#	26-735#	26-735#	26-735#	26-737	26-737#	26-737	26-737#
26-737	26-737	26-737#	26-737#	26-737#	26-737#	26-737#	26-737#	26-746	26-746#	26-746#	26-746	26-746#	26-746#	26-746#
26-746#	26-746#	26-748	26-748#	26-752	26-752#	26-762	26-762#	26-762	26-762#	26-762#	26-762	26-762#	26-762#	26-762#
26-762#	26-762#	26-762#	26-771	26-771#	26-771	26-771#	26-771#	26-771#	26-771#	26-771#	26-771#	26-771#	26-773	26-773#
26-777	26-777#	26-787	26-787#	26-787	26-787#	26-787	26-787#	26-787#	26-787#	26-787#	26-787#	26-787#	26-787#	26-796
26-796	26-796	26-796#	26-796#	26-796#	26-796#	26-796#	26-796#	26-798	26-798#	26-798#	26-799	26-799#	26-823	26-823#
26-823	26-823	26-823#	26-823#	26-823#	26-823#	26-823#	26-823#	26-823#	26-823#	26-823#	26-825	26-825#	26-825	26-825#
26-825	26-825#	26-825#	26-825#	26-825#	26-825#	26-827	26-827#	26-827#	26-827#	26-827#	26-825	26-825#	26-825	26-825#
26-:37	26-:37	26-:37	26-:37	26-:37#	26-:37#	26-:37#	26-:37#	26-:37#	26-:37#	26-:37#	26-:77	26-:77#	26-:08	26-:08#
26-:43	26-:43	26-:43	26-:43#	26-:43#	26-:43#	26-:43#	26-:43#	26-:43#	26-:43#	26-:43#	26-:38	26-:38#	26-:38#	26-:43
26-:54	26-:54#	26-:83	26-:83	26-:83	26-:83#	26-:83#	26-:83#	26-:83#	26-:83#	26-:83#	26-:84	26-:84#	26-:84	26-:84#
26-:84#	26-:89	26-:89	26-:89	26-:89	26-:89#	26-:89#	26-:89#	26-:89#	26-:89#	26-:89#	26-:90	26-:90#	26-:90#	26-:90#
26-:94	26-:94#	26-:96	26-:96#	26-<30	26-<30#	26-<77	26-<77#	26-<83	26-<83#	26-<83#	26-=30	26-=30#	26-=32	26-=32#
26-=72	26-=72	26-=72	26-=72#	26-=72#	26-=72#	26-=72#	26-=72#	26-=72#	26-=72#	26-=72#	26-=94	26-=94#	26-=94	26-=94#
26-=94#	26-=94#	26-=94#	26-=94#	26-=96	26-=96#	26->34	26->34#	26->34	26->34#	26->34#	26->34	26->34#	26->34#	26->34#
26->34#	26->49	26->49	26->49#	26->49#	26->49#	26->49#	26->49#	26->49#	26->49#	26->49#	26->58	26->58#	26->99	26->99#
26->99	26->99	26->99#	26->99#	26->99#	26->99#	26->99#	26->99#	26-?08	26-?08#	26-?49	26-?49	26-?49	26-?49	26-?49#
26-?49#	26-?49#	26-?49#	26-?49#	26-?58	26-?58#	26-?94	26-?94#	26-@42	26-@42#	26-@82	26-@82#	26-@82#	26-@92	26-@92#
26-@92	26-@92	26-@92#	26-@92#	26-@92#	26-@92#	26-@92#	26-@92#	26-@93	26-@93#	26-@93#	26-@93#	26-@93#	26-A02	26-A02#
26-A02	26-A02#	26-A02#	26-A02#	26-A02#	26-A02#	26-A03	26-A03#	26-A03#	26-A03#	26-A03#	26-A21	26-A21#	26-A26	26-A26#
26-A37	26-A37	26-A37	26-A37#	26-A37#	26-A37#	26-A37#	26-A37#	26-A37#	26-A37#	26-A37#	26-A38	26-A38#	26-A38#	26-A40
26-A40#	26-A45	26-A45#	26-A56	26-A56	26-A56#	26-A56	26-A56#	26-A56#	26-A56#	26-A56#	26-A56#	26-A56#	26-A58	26-A58#
26-A63	26-A63#	26-A68	26-A68#	26-A73	26-A73#	26-A85	26-A85#	26-A85	26-A85#	26-A85#	26-A85#	26-A85#	26-A85#	26-A85#
26-A85#	26-A86	26-A86	26-A86#	26-A86#	26-A91	26-A91	26-A91#	26-A91	26-A91#	26-A91#	26-A91#	26-A91#	26-A91#	26-A91#
26-A93	26-A93#	26-A98	26-A98#	26-B09	26-B09#	26-B09	26-B09#	26-B09#	26-B09#	26-B09#	26-B09#	26-B09#	26-B09#	26-B11
26-B11#	26-B16	26-B16#	26-B26	26-B26	26-B26#	26-B26	26-B26#	26-B26#	26-B26#	26-B26#	26-B26#	26-B26#	26-B28	26-B28#
26-B33	26-B33#	26-B45	26-B45#	26-B45	26-B45#	26-B45#	26-B45#	26-B45#	26-B45#	26-B45#	26-B45#	26-B45#	26-B47	26-B47#
26-B52#	26-B62	26-B62	26-B62#	26-B62	26-B62#	26-B62#	26-B62#	26-B62#	26-B62#	26-B62#	26-B64	26-B64#	26-B69	26-B69#
26-B79	26-B79	26-B79	26-B79#	26-B79#	26-B79#	26-B79#	26-B79#	26-B79#	26-B79#	26-B79#	26-B80	26-B80#	26-B80#	26-B85
26-B85	26-B85	26-B85	26-B85#	26-B85#	26-B85#	26-B85#	26-B85#	26-B86	26-B86#	26-B86#	26-B86#	26-B86#	26-B86#	26-B85
26-B91	26-B91	26-B91#	26-B91#	26-B91#	26-B91#	26-B91#	26-B91#	26-B93	26-B93#	26-B97	26-B97#	26-B97#	26-B91	26-B91#
26-C42	26-C42#	26-C42#	26-C42#	26-C42#	26-C42#	26-C51	26-C51#	26-C96	26-C96#	26-C96	26-C96	26-C96#	26-C42	26-C42#
26-C96#	26-C96#	26-C96#	26-D05	26-D05#	26-D50	26-D50	26-D50#	26-D50	26-D50#	26-D50#	26-D50#	26-D50#	26-D50#	26-D50#
26-D59	26-D59#	26-E04	26-E04#	26-E04	26-E04#	26-E04#	26-E04#	26-E04#	26-E04#	26-E04#	26-E04#	26-E04#	26-E13	26-E13#
26-E58	26-E58	26-E58	26-E58#	26-E58#	26-E58#	26-E58#	26-E58#	26-E67	26-E67#	26-E67#	26-F11	26-F11#	26-F11	26-F11#
26-F11#	26-F11#	26-F11#	26-F11#	26-F11#	26-F21	26-F21#	26-F21#	26-F65	26-F65#	26-F65#	26-F65	26-F65#	26-F65#	26-F65#
26-F65#	26-F65#	26-F75	26-F75#	26-G18	26-G18#	26-G71	26-G71#	26-G71	26-G71#	26-G71#	26-G71#	26-G71#	26-G71#	26-G71#
26-G71#	26-G83	26-G83	26-G83#	26-G83	26-G83#	26-G83#	26-G83#	26-G83#	26-G83#	26-G83#	26-H45	26-H45#	26-H45	26-H45#
26-H45#	26-H45#	26-H45#	26-H45#	26-H45#	26-H72	26-H72#	26-H72#	27-14	27-14#	27-16	27-16	27-16	27-16	27-16#
27-17	27-17	27-17	27-17#	27-17#	27-18	27-18	27-18#	27-18	27-18#	27-18	27-18#	27-18#	27-19	27-19#

	27-19	27-19	27-19#	27-20	27-20	27-20	27-20	27-20	27-20#	27-21	27-21	27-21	27-21	27-21
	27-21#	27-22	27-22	27-22	27-22	27-22	27-22#	27-23	27-23	27-23	27-23	27-23	27-23#	27-25
	27-25#	28-13	28-13#	28-15	28-15	28-15	28-15#	28-16	28-16	28-16	28-16#	28-17	28-17	28-17
	28-17#	28-18	28-18	28-18	28-18	28-18	28-18#	28-20	28-20#	28-47	28-47	28-47	28-47#	28-47#
	28-47#													
MSGNLS	1-C13#	9-18#	21-94	21-94#										
MSGNSU	1-B98#	9-18#	26-510	26-510#	26-539	26-539#	26-593	26-593#	26-620	26-620#	26-724	26-724#	26-752	26-752#
	26-777	26-777#	26-:08	26-:08#	26-:54	26-:54#	26-<30	26-<30#	26-<83	26-<83#	26-282	26-282#	26-A26	26-A26#
	26-A45	26-A45#	26-A63	26-A63#	26-A73	26-A73#	26-A98	26-A98#	26-B16	26-B16#	26-B33	26-B33#	26-B52	26-B52#
	26-B69	26-B69#												
MSGNTA	1-B90#	9-18#	12-23	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154	18-154#
	18-174	18-174#	18-193	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	21-99	21-99#
	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-147	26-147#	26-253	26-253#
	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533	26-533#	26-565	26-565#	26-570	26-570#
	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-748	26-748#	26-773	26-773#	26-798	26-798#
	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:48	26-:48#	26-:94	26-:94#	26-:96	26-:96#
	26-<77	26-<77#	26-30	26-30#	26-32	26-32#	26-96	26-96#	26->58	26->58#	26-?08	26-?08#	26-?58	26-?58#
	26-?94	26-?94#	26-242	26-242#	26-A21	26-A21#	26-A40	26-A40#	26-A58	26-A58#	26-A68	26-A68#	26-A93	26-A93#
	26-B11	26-B11#	26-B28	26-B28#	26-B47	26-B47#	26-B64	26-B64#	26-B93	26-B93#	26-B97	26-B97#	26-C51	26-C51#
	26-D05	26-D05#	26-D59	26-D59#	26-E13	26-E13#	26-E67	26-E67#	26-F21	26-F21#	26-F75	26-F75#	26-G18	26-G18#
	26-H72	26-H72#	27-25	27-25#	28-20	28-20#								
MSGNTE	1-B94#	9-18#	26-20	26-20#	26-65	26-65#	26-173	26-173#	26-267	26-267#	26-299	26-299#	26-332	26-332#
	26-384	26-384#	26-506	26-506#	26-589	26-589#	26-663	26-663#	26-720	26-720#	26-812	26-812#	26-850	26-850#
	26-969	26-969#	26-:02	26-:02#	26-<24	26-<24#	26-48	26-48#	26->13	26->13#	26->75	26->75#	26-?25	26-?25#
	26-?76	26-?76#	26-209	26-209#	26-273	26-273#	26-C17	26-C17#	26-C71	26-C71#	26-D25	26-D25#	26-D79	26-D79#
	26-E33	26-E33#	26-E87	26-E87#	26-F41	26-F41#	26-F92	26-F92#	26-G37	26-G37#				
MSHAPT	1-A39#	9-18#	10-17	10-17#										
MSHNAP	1-B24#	9-18#	10-17	10-17#										
MSINCR	1-D26#	9-18#	9-24	9-24#	12-9	12-9	12-9#	12-9#	13-8	13-8	13-8#	13-8#	17-277#	17-283#
	17-293#	17-299#	17-406#	17-412#	17-570#	17-576#	17-586#	17-592#	17-627#	17-633#	17-667#	17-673#	17-681#	17-687#
	17-833#	17-838#	17-857#	17-863#	17-872#	17-878#	17-887#	17-893#	17-902#	17-908#	17-:61#	17-:66#	17-:75#	17-<54#
	17-<70#	17-06#	17-12#	18-101	18-101	18-101#	18-101#	18-102#	18-103#	18-107	18-107	18-107#	18-107#	18-108#
	18-109#	18-110#	18-111#	18-112#	18-113#	18-114#	18-115#	18-116#	18-122	18-122	18-122#	18-122#	18-123#	18-124#
	18-125#	18-126#	18-127#	18-128#	18-129#	18-130#	18-131#	18-132#	18-133#	18-134#	18-135#	18-141	18-141	18-141#
	18-141#	18-142#	18-143#	18-144#	18-145#	18-146#	18-147#	18-148#	18-149#	18-150#	18-151#	18-152#	18-153#	18-154#
	18-160	18-160	18-160#	18-160#	18-161#	18-162#	18-163#	18-164#	18-165#	18-166#	18-167#	18-168#	18-169#	18-170#
	18-171#	18-172#	18-173#	18-174#	18-180	18-180	18-180#	18-180#	18-181#	18-182#	18-183#	18-184#	18-185#	18-186#
	18-187#	18-188#	18-189#	18-190#	18-191#	18-192#	18-193#	18-199	18-199	18-199#	18-199#	18-200#	18-201#	18-202#
	18-203#	18-204#	18-205#	18-206#	18-207#	18-208#	18-209#	18-210#	18-211#	18-217	18-217	18-217#	18-217#	18-218#
	18-219#	18-220#	18-221#	18-222#	18-223#	18-224#	18-225#	18-226#	18-227#	18-228#	18-229#	18-230#	18-231#	18-237
	18-237	18-237#	18-237#	18-238#	18-239#	18-240#	18-241#	18-242#	18-243#	18-244#	18-245#	18-246#	18-247#	18-248#
	18-249#	19-9	19-9	19-9#	19-9#	19-11#	20-8	20-8	20-8#	20-8#	21-8	21-8	21-8#	21-8#
	21-25#	21-28#	21-31#	21-34#	21-52#	21-81#	21-85#	21-86#	21-90#	21-94	21-94#	21-94#	21-99#	22-8
	22-8	22-8#	22-8#	22-10#	22-17#	22-20#	23-8	23-8	23-8#	23-8#	23-11#	24-8	24-8	24-8#
	24-8#	24-10#	24-12#	24-13#	25-9	25-9	25-9#	25-9#	25-10#	26-20	26-20	26-20	26-20#	26-20#
	26-20#	26-46#	26-65	26-65	26-65	26-65#	26-65#	26-65#	26-105#	26-145#	26-147#	26-173	26-173	26-173
	26-173#	26-173#	26-173#	26-224#	26-251#	26-253#	26-267	26-267	26-267	26-267#	26-267#	26-267#	26-284#	26-299
	26-299	26-299	26-299#	26-299#	26-299#	26-317#	26-332	26-332	26-332	26-332#	26-332#	26-332#	26-364#	26-366#
	26-384	26-384	26-384	26-384#	26-384#	26-384#	26-419#	26-458#	26-471#	26-480#	26-506	26-506	26-506	26-506#
	26-506#	26-506#	26-510	26-510	26-510	26-510#	26-510#	26-510#	26-533#	26-539	26-539	26-539	26-539#	26-539#
	26-539#	26-565#	26-570#	26-589	26-589	26-589	26-589#	26-589#	26-589#	26-593	26-593	26-593	26-593#	26-593#
	26-593#	26-616#	26-620	26-620	26-620	26-620#	26-620#	26-620#	26-643#	26-644#	26-663	26-663	26-663	26-663#
	26-663#	26-663#	26-695#	26-720	26-720	26-720	26-720#	26-720#	26-720#	26-724	26-724	26-724	26-724#	26-724#
	26-724#	26-735#	26-737#	26-746#	26-748#	26-752	26-752	26-752	26-752#	26-752#	26-752#	26-762#	26-771#	26-773#
	26-777	26-777	26-777	26-777#	26-777#	26-777#	26-787#	26-796#	26-798#	26-799#	26-812	26-812	26-812	26-812#
	26-812#	26-812#	26-823#	26-825#	26-827#	26-850	26-850	26-850	26-850#	26-850#	26-850#	26-850#	26-850#	26-850#
	26-969	26-969#	26-969#	26-969#	26-:77#	26-:02	26-:02	26-:02	26-:02#	26-:02#	26-:02#	26-:02#	26-:08	26-:08

	26-:08#	26-:08#	26-:08#	26-:37#	26-:38#	26-:43#	26-:44#	26-:48#	26-:54	26-:54	26-:54	26-:54#	26-:54#	26-:54#
	26-:83#	26-:84#	26-:89#	26-:90#	26-:94#	26-:96#	26-<24	26-<24	26-<24	26-<24#	26-<24#	26-<24#	26-<30	26-<30
	26-<30	26-<30#	26-<30#	26-<30#	26-<77#	26-<83	26-<83	26-<83	26-<83#	26-<83#	26-<83#	26-<83#	26-<30#	26-<32#
	26-=48	26-=48	26-=48#	26-=48#	26-=48#	26-=72#	26-=94#	26-=96#	26->13	26->13	26->13	26->13#	26->13#	26->13#
	26->34#	26->49#	26->58#	26->75	26->75	26->75	26->75#	26->75#	26->75#	26->99#	26-?08#	26-?25	26-?25	26-?25
	26-?25#	26-?25#	26-?25#	26-?49#	26-?58#	26-?76	26-?76	26-?76	26-?76#	26-?76#	26-?76#	26-?94#	26-@09	26-@09
	26-@09	26-@09#	26-@09#	26-@09#	26-@42#	26-@73	26-@73	26-@73	26-@73#	26-@73#	26-@73#	26-@82	26-@82	26-@82
	26-@82#	26-@82#	26-@82#	26-@92#	26-@93#	26-A02#	26-A03#	26-A21#	26-A26	26-A26	26-A26	26-A26#	26-A26#	26-A26#
	26-A37#	26-A38#	26-A40#	26-A45	26-A45	26-A45	26-A45#	26-A45#	26-A45#	26-A56#	26-A58#	26-A63	26-A63	26-A63
	26-A63#	26-A63#	26-A63#	26-A68#	26-A73	26-A73	26-A73	26-A73#	26-A73#	26-A73#	26-A85#	26-A86#	26-A91#	26-A93#
	26-A98	26-A98	26-A98	26-A92#	26-A98#	26-A98#	26-B09#	26-B11#	26-B16	26-B16	26-B16	26-B16#	26-B16#	26-B16#
	26-B26#	26-B28#	26-B33	26-B33	26-B33	26-B33#	26-B33#	26-B33#	26-B45#	26-B47#	26-B52	26-B52	26-B52	26-B52#
	26-B52#	26-B52#	26-B62#	26-B64#	26-B69	26-B69	26-B69	26-B69#	26-B69#	26-B69#	26-B79#	26-B80#	26-B85#	26-B86#
	26-B91#	26-B93#	26-B97#	26-C17	26-C17	26-C17	26-C17#	26-C17#	26-C17#	26-C42#	26-C51#	26-C71	26-C71	26-C71
	26-C71#	26-C71#	26-C71#	26-C96#	26-D05#	26-D25	26-D25	26-D25	26-D25#	26-D25#	26-D25#	26-D50#	26-D59#	26-D79
	26-D79	26-D79	26-D79#	26-D79#	26-D79#	26-E04#	26-E13#	26-E33	26-E33	26-E33	26-E33#	26-E33#	26-E33#	26-E58#
	26-E67#	26-E87	26-E87	26-E87	26-E87#	26-E87#	26-E87#	26-F11#	26-F21#	26-F41	26-F41	26-F41	26-F41#	26-F41#
	26-F41#	26-F65#	26-F75#	26-F92	26-F92	26-F92	26-F92#	26-F92#	26-F92#	26-G18#	26-G37	26-G37	26-G37	26-G37#
	26-G37#	26-G37#	26-G71#	26-G83#	26-H45#	26-H72#	27-14	27-14	27-14#	27-14#	28-13	28-13	28-13#	28-13#
MSIOSE	1-A00#	9-18#												
MSLDRO	1-C42#	9-18#	21-25	21-25#	21-28	21-28#	21-31	21-31#	21-34	21-34#	21-52	21-52#	22-10	22-10#
	22-17	22-17#												
MSMASK	1-@71#	9-18#												
MSMCHI	1-4#	9-18	9-18#	9-18#										
MSMCLO	1-@24#	9-18	9-18#	9-18#										
MSMSK1	1-@77#	9-18#												
MSPOP	1-B81#	9-18#	12-23	12-23#	13-15	13-15#	18-103	18-103#	18-116	18-116#	18-135	18-135#	18-154	18-154#
	18-174	18-174#	18-193	18-193#	18-211	18-211#	18-231	18-231#	18-249	18-249#	19-11	19-11#	20-12	20-12#
	21-99	21-99#	22-20	22-20#	23-11	23-11#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-147	26-147#
	26-253	26-253#	26-284	26-284#	26-317	26-317#	26-366	26-366#	26-480	26-480#	26-533	26-533#	26-565	26-565#
	26-570	26-570#	26-616	26-616#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-748	26-748#	26-773	26-773#
	26-798	26-798#	26-799	26-799#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:48	26-:48#	26-:94	26-:94#
	26-:96	26-:96#	26-<77	26-<77#	26-=30	26-=30#	26-=32	26-=32#	26-=96	26-=96#	26->58	26->58#	26-?08	26-?08#
	26-?58	26-?58#	26-?94	26-?94#	26-@42	26-@42#	26-A21	26-A21#	26-A40	26-A40#	26-A58	26-A58#	26-A68	26-A68#
	26-A93	26-A93#	26-B11	26-B11#	26-B28	26-B28#	26-B47	26-B47#	26-B64	26-B64#	26-B93	26-B93#	26-B97	26-B97#
	26-C51	26-C51#	26-D05	26-D05#	26-D59	26-D59#	26-E13	26-E13#	26-E67	26-E67#	26-F21	26-F21#	26-F75	26-F75#
	26-G18	26-G18#	26-H72	26-H72#	27-25	27-25#	28-20	28-20#	28-45	28-45#				
MSPRIN	1-@36#	9-18#	17-:61	17-:61#	17-:66	17-:66#	17-:75	17-:75#	18-102	18-102#	18-108	18-108#	18-109	18-109#
	18-110	18-110#	18-111	18-111#	18-112	18-112#	18-113	18-113#	18-114	18-114#	18-115	18-115#	18-123	18-123#
	18-124	18-124#	18-125	18-125#	18-126	18-126#	18-127	18-127#	18-128	18-128#	18-129	18-129#	18-130	18-130#
	18-131	18-131#	18-132	18-132#	18-133	18-133#	18-134	18-134#	18-142	18-142#	18-143	18-143#	18-144	18-144#
	18-145	18-145#	18-146	18-146#	18-147	18-147#	18-148	18-148#	18-149	18-149#	18-150	18-150#	18-151	18-151#
	18-152	18-152#	18-153	18-153#	18-161	18-161#	18-162	18-162#	18-163	18-163#	18-164	18-164#	18-165	18-165#
	18-166	18-166#	18-167	18-167#	18-168	18-168#	18-169	18-169#	18-170	18-170#	18-171	18-171#	18-172	18-172#
	18-173	18-173#	18-181	18-181#	18-182	18-182#	18-183	18-183#	18-184	18-184#	18-185	18-185#	18-186	18-186#
	18-187	18-187#	18-188	18-188#	18-189	18-189#	18-190	18-190#	18-191	18-191#	18-192	18-192#	18-200	18-200#
	18-201	18-201#	18-202	18-202#	18-203	18-203#	18-204	18-204#	18-205	18-205#	18-206	18-206#	18-207	18-207#
	18-208	18-208#	18-209	18-209#	18-210	18-210#	18-218	18-218#	18-219	18-219#	18-220	18-220#	18-221	18-221#
	18-222	18-222#	18-223	18-223#	18-224	18-224#	18-225	18-225#	18-226	18-226#	18-227	18-227#	18-228	18-228#
	18-229	18-229#	18-230	18-230#	18-238	18-238#	18-239	18-239#	18-240	18-240#	18-241	18-241#	18-242	18-242#
	18-243	18-243#	18-244	18-244#	18-245	18-245#	18-246	18-246#	18-247	18-247#	18-248	18-248#	21-85	21-85#
	21-90	21-90#	24-12	24-12#	26-735	26-735#	26-737	26-737#	26-762	26-762#	26-787	26-787#	26-823	26-823#
	26-825	26-825#												
MSPUSH	1-@31#	9-18#	9-24	9-24#	12-9	12-9#	13-8	13-8#	18-101	18-101#	18-107	18-107#	18-122	18-122#
	18-141	18-141#	18-160	18-160#	18-180	18-180#	18-199	18-199#	18-217	18-217#	18-237	18-237#	19-9	19-9#
	20-8	20-8#	21-8	21-8#	22-8	22-8#	23-8	23-8#	24-8	24-8#	25-9	25-9#	26-20	26-20#
	26-65	26-65#	26-173	26-173#	26-267	26-267#	26-299	26-299#	26-332	26-332#	26-384	26-384#	26-506	26-506#

	26-510	26-510#	26-539	26-539#	26-589	26-589#	26-593	26-593#	26-620	26-620#	26-663	26-663#	26-720	26-720#
	26-724	26-724#	26-752	26-752#	26-777	26-777#	26-812	26-812#	26-850	26-850#	26-969	26-969#	26-:02	26-:02#
	26-:08	26-:08#	26-:54	26-:54#	26-<24	26-<24#	26-<30	26-<30#	26-<83	26-<83#	26-=48	26-=48#	26->13	26->13#
	26->75	26->75#	26-?25	26-?25#	26-?76	26-?76#	26-a09	26-a09#	26-a73	26-a73#	26-a82	26-a82#	26-A26	26-A26#
	26-A45	26-A45#	26-A63	26-A63#	26-A73	26-A73#	26-A98	26-A98#	26-B16	26-B16#	26-B33	26-B33#	26-B52	26-B52#
	26-B69	26-B69#	26-C17	26-C17#	26-C71	26-C71#	26-D25	26-D25#	26-D79	26-D79#	26-E33	26-E33#	26-E87	26-E87#
	26-F41	26-F41#	26-F92	26-F92#	26-G37	26-G37#	27-14	27-14#	28-13	28-13#				
MSPUT	1-C72#	9-18#	17-:61	17-:61	17-:61#	17-:66	17-:66	17-:66	17-:66#	17-:75	17-:75	17-:75#	18-102	18-102
	18-102	18-102	18-102#	18-108	18-108	18-108	18-108	18-108#	18-109	18-109	18-109#	18-110	18-110	18-110
	18-110	18-110#	18-111	18-111	18-111	18-111	18-111#	18-112	18-112	18-112	18-112	18-112#	18-113	18-113
	18-113	18-113	18-113	18-113	18-113#	18-114	18-114	18-114	18-114#	18-115	18-115	18-115	18-115	18-115
	18-115	18-115#	18-123	18-123	18-123	18-123	18-123#	18-124	18-124	18-124#	18-125	18-125	18-125	18-125
	18-125#	18-126	18-126	18-126	18-126	18-126#	18-127	18-127	18-127	18-127#	18-128	18-128	18-128	18-128
	18-128	18-128	18-128	18-128#	18-129	18-129	18-129	18-129#	18-130	18-130	18-130	18-130	18-130	18-130
	18-130#	18-131	18-131	18- 31	18-131	18-131#	18-132	18-132	18-132	18-132	18-132	18-132#	18-132#	18-133
	18-133	18-133	18-133#	18-134	18-134	18-134	18-134	18-134	18-134	18-134#	18-142	18-142	18-142	18-142#
	18-143	18-143	18-143	18-143	18-143#	18-144	18-144	18-144#	18-145	18-145	18-145	18-145#	18-145#	18-146
	18-146	18-146	18-146	18-146#	18-147	18-147	18-147	18-147	18-147	18-147#	18-148	18-148	18-148	18-148
	18-148#	18-149	18-149	18-149	18-149	18-149	18-149	18-149#	18-150	18-150	18-150	18-150	18-150#	18-151
	18-151	18-151	18-151	18-151	18-151	18-151#	18-152	18-152	18-152	18-152#	18-153	18-153	18-153	18-153
	18-153	18-153	18-153#	18-161	18-161	18-161	18-161	18-161#	18-162	18-162	18-162	18-162#	18-162#	18-163
	18-163	18-163#	18-164	18-164	18-164	18-164	18-164#	18-165	18-165	18-165	18-165	18-165#	18-166	18-166
	18-166	18-166	18-166#	18-167	18-167	18-167	18-167	18-167	18-167	18-167#	18-168	18-168	18-168	18-168#
	18-169	18-169	18-169	18-169	18-169	18-169	18-169#	18-170	18-170	18-170	18-170	18-170#	18-171	18-171
	18-171	18-171	18-171	18-171	18-171#	18-172	18-172	18-172	18-172#	18-173	18-173	18-173	18-173	18-173
	18-173	18-173#	18-181	18-181	18-181	18-181#	18-182	18-182	18-182	18-182#	18-183	18-183	18-183	18-183#
	18-184	18-184	18-184	18-184	18-184#	18-185	18-185	18-185	18-185	18-185#	18-186	18-186	18-186	18-186
	18-186	18-186	18-186#	18-187	18-187	18-187	18-187#	18-188	18-188	18-188	18-188	18-188	18-188	18-188#
	18-189	18-189	18-189	18-189	18-189#	18-190	18-190	18-190	18-190	18-190	18-190	18-190#	18-191	18-191
	18-191	18-191#	18-192	18-192	18-192	18-192	18-192	18-192	18-192#	18-200	18-200	18-200	18-200	18-200#
	18-201	18-201	18-201#	18-202	18-202	18-202	18-202	18-202#	18-203	18-203	18-203	18-203	18-203#	18-204
	18-204	18-204	18-204	18-204	18-204	18-204#	18-205	18-205	18-205	18-205#	18-206	18-206	18-206	18-206
	18-206	18-206	18-206#	18-207	18-207	18-207	18-207	18-207#	18-208	18-208	18-208	18-208	18-208	18-208
	18-208#	18-209	18-209	18-209	18-209#	18-210	18-210	18-210	18-210	18-210	18-210	18-210#	18-218	18-218
	18-218	18-218#	18-219	18-219	18-219	18-219	18-219#	18-220	18-220	18-220#	18-221	18-221	18-221	18-221
	18-221#	18-222	18-222	18-222	18-222	18-222#	18-223	18-223	18-223	18-223	18-223#	18-224	18-224	18-224
	18-224	18-224	18-224	18-224#	18-225	18-225	18-225	18-225#	18-226	18-226	18-226	18-226	18-226	18-226
	18-226#	18-227	18-227	18-227	18-227	18-227#	18-228	18-228	18-228	18-228	18-228	18-228	18-228#	18-229
	18-229	18-229	18-229#	18-230	18-230	18-230	18-230	18-230	18-230	18-230#	18-238	18-238	18-238	18-238
	18-238#	18-239	18-239	18-239#	18-240	18-240	18-240	18-240	18-240#	18-241	18-241	18-241	18-241	18-241#
	18-242	18-242	18-242	18-242	18-242	18-242#	18-243	18-243	18-243	18-243	18-243#	18-244	18-244	18-244
	18-244	18-244	18-244	18-244#	18-245	18-245	18-245	18-245	18-245#	18-246	18-246	18-246	18-246	18-246
	18-246	18-246#	18-247	18-247	18-247	18-247#	18-248	18-248	18-248	18-248	18-248	18-248	18-248#	21-85
	21-85	21-85#	21-90	21-90	21-90	21-90#	24-12	24-12	24-12	24-12#	26-735	26-735	26-735	26-735#
	26-737	26-737	26-737	26-737#	26-762	26-762	26-762	26-762#	26-787	26-787	26-787	26-787#	26-823	26-823
	26-823	26-823#	26-825	26-825	26-825	26-825#								
MSPUT1	1-C81#	9-18#	17-:61	17-:61	17-:61#	17-:66	17-:66	17-:66	17-:66#	17-:66#	17-:66#	17-:66#	17-:75	17-:75
	17-:75#	17-:75#	18-102	18-102	18-102	18-102	18-102#	18-102#	18-102#	18-102#	18-108	18-108	18-108	18-108
	18-108#	18-108#	18-108#	18-108#	18-109	18-109	18-109	18-109#	18-109#	18-110	18-110	18-110	18-110#	18-110#
	18-110#	18-110#	18-111	18-111	18-111	18-111	18-111#	18-111#	18-111#	18-111#	18-112	18-112	18-112	18-112
	18-112#	18-112#	18-112#	18-112#	18-113	18-113	18-113	18-113	18-113	18-113	18-113#	18-113#	18-113#	18-113#
	18-113#	18-113#	18-114	18-114	18-114	18-114#	18-114#	18-114#	18-115	18-115	18-115	18-115	18-115	18-115
	18-115#	18-115#	18-115#	18-115#	18-115#	18-115#	18-123	18-123	18-123	18-123	18-123#	18-123#	18-123#	18-123#
	18-124	18-124	18-124#	18-124#	18-125	18-125	18-125	18-125	18-125#	18-125#	18-125#	18-125#	18-126	18-126
	18-126	18-126	18-126#	18-126#	18-126#	18-126#	18-127	18-127	18-127	18-127	18-127#	18-127#	18-127#	18-127#
	18-128	18-128	18-128	18-128	18-128	18-128	18-128#	18-128#	18-128#	18-128#	18-128#	18-128#	18-129	18-129
	18-129	18-129#	18-129#	18-129#	18-130	18-130	18-130	18-130	18-130	18-130	18-130#	18-130#	18-130#	18-130#

MSRAD I

MSRBRO

MSRNR0

ASSETS

	20-8	20-8#	21-8	21-8#	22-8	22-8#	23-8	23-8#	24-8	24-8#	25-9	25-9#	26-20	26-20#
	26-65	26-65#	26-173	26-173#	26-267	26-267#	26-299	26-299#	26-332	26-332#	26-384	26-384#	26-506	26-506#
	26-510	26-510#	26-539	26-539#	26-589	26-589#	26-593	26-593#	26-620	26-620#	26-663	26-663#	26-720	26-720#
	26-724	26-724#	26-752	26-752#	26-777	26-777#	26-812	26-812#	26-850	26-850#	26-969	26-969#	26-:02	26-:02#
	26-:08	26-:08#	26-:54	26-:54#	26-<24	26-<24#	26-<30	26-<30#	26-<83	26-<83#	26-=48	26-=48#	26->13	26->13#
	26->75	26->75#	26-?25	26-?25#	26-?76	26-?76#	26-@09	26-@09#	26-@73	26-@73#	26-@82	26-@82#	26-A26	26-A26#
	26-A45	26-A45#	26-A63	26-A63#	26-A73	26-A73#	26-A98	26-A98#	26-B16	26-B16#	26-B33	26-B33#	26-B52	26-B52#
	26-B69	26-B69#	26-C17	26-C17#	26-C71	26-C71#	26-D25	26-D25#	26-D79	26-D79#	26-E33	26-E33#	26-E87	26-E87#
	26-F41	26-F41#	26-F92	26-F92#	26-G37	26-G37#	27-14	27-14#	28-13	28-13#				
MSSTAR	1-A33#	9-18#												
MS SVC	1-C33#	9-18#	17-277	17-283	17-293	17-299	17-406	17-412	17-570	17-576	17-586	17-592	17-627	17-633
	17-667	17-673	17-681	17-687	17-833	17-838	17-857	17-863	17-872	17-878	17-887	17-893	17-902	17-908
	17-:61	17-:61#	17-:66	17-:66#	17-:75	17-:75#	17-<54	17-<70	17-=06	17-=12	18-102	18-102#	18-103	18-103#
	18-108	18-108#	18-109	18-109#	18-110	18-110#	18-111	18-111#	18-112	18-112#	18-113	18-113#	18-114	18-114#
	18-115	18-115#	18-116	18-116#	18-123	18-123#	18-124	18-124#	18-125	18-125#	18-126	18-126#	18-127	18-127#
	18-128	18-128#	18-129	18-129#	18-130	18-130#	18-131	18-131#	18-132	18-132#	18-133	18-133#	18-134	18-134#
	18-135	18-135#	18-142	18-142#	18-143	18-143#	18-144	18-144#	18-145	18-145#	18-146	18-146#	18-147	18-147#
	18-148	18-148#	18-149	18-149#	18-150	18-150#	18-151	18-151#	18-152	18-152#	18-153	18-153#	18-154	18-154#
	18-161	18-161#	18-162	18-162#	18-163	18-163#	18-164	18-164#	18-165	18-165#	18-166	18-166#	18-167	18-167#
	18-168	18-168#	18-169	18-169#	18-170	18-170#	18-171	18-171#	18-172	18-172#	18-173	18-173#	18-174	18-174#
	18-181	18-181#	18-182	18-182#	18-183	18-183#	18-184	18-184#	18-185	18-185#	18-186	18-186#	18-187	18-187#
	18-188	18-188#	18-189	18-189#	18-190	18-190#	18-191	18-191#	18-192	18-192#	18-193	18-193#	18-200	18-200#
	18-201	18-201#	18-202	18-202#	18-203	18-203#	18-204	18-204#	18-205	18-205#	18-206	18-206#	18-207	18-207#
	18-208	18-208#	18-209	18-209#	18-210	18-210#	18-211	18-211#	18-218	18-218#	18-219	18-219#	18-220	18-220#
	18-221	18-221#	18-222	18-222#	18-223	18-223#	18-224	18-224#	18-225	18-225#	18-226	18-226#	18-227	18-227#
	18-228	18-228#	18-229	18-229#	18-230	18-230#	18-231	18-231#	18-238	18-238#	18-239	18-239#	18-240	18-240#
	18-241	18-241#	18-242	18-242#	18-243	18-243#	18-244	18-244#	18-245	18-245#	18-246	18-246#	18-247	18-247#
	18-248	18-248#	18-249	18-249#	19-11	19-11#	21-25	21-25#	21-28	21-28#	21-31	21-31#	21-34	21-34#
	21-52	21-52#	21-81	21-81#	21-85	21-85#	21-86	21-86#	21-90	21-90#	21-94	21-94#	21-99	21-99#
	22-10	22-10#	22-17	22-17#	22-20	22-20#	23-11	23-11#	24-10	24-10#	24-12	24-12#	24-13	24-13#
	25-10	25-10#	26-46	26-46#	26-105	26-145	26-147	26-147#	26-224	26-251	26-253	26-253#	26-284	26-284#
	26-317	26-317#	26-364	26-366	26-366#	26-419	26-458	26-471	26-480	26-480#	26-510	26-510#	26-533	26-533#
	26-539	26-539#	26-565	26-565#	26-570	26-570#	26-593	26-593#	26-616	26-616#	26-620	26-620#	26-643	26-643#
	26-644	26-644#	26-695	26-695#	26-724	26-724#	26-735	26-735#	26-737	26-737#	26-746	26-748	26-748#	26-752
	26-752#	26-762	26-762#	26-771	26-773	26-773#	26-777	26-777#	26-787	26-787#	26-796	26-798	26-798#	26-799
	26-799#	26-823	26-823#	26-825	26-825#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:08	26-:08#	26-:37
	26-:38	26-:38#	26-:43	26-:44	26-:44#	26-:48	26-:48#	26-:54	26-:54#	26-:83	26-:84	26-:84#	26-:89	26-:90
	26-:90#	26-:94	26-:94#	26-:96	26-:96#	26-<30	26-<30#	26-<77	26-<77#	26-<83	26-<83#	26-=30	26-=30#	26-=32
	26-=32#	26-=72	26-=94	26-=96	26-=96#	26->34	26->49	26->58	26->58#	26->99	26-?08	26-?08#	26-?49	26-?58
	26-?58#	26-?94	26-?94#	26-@42	26-@42#	26-@82	26-@82#	26-@92	26-@93	26-@93#	26-A02	26-A03	26-A03#	26-A21
	26-A21#	26-A26	26-A26#	26-A37	26-A38	26-A38#	26-A40	26-A40#	26-A45	26-A45#	26-A56	26-A58	26-A58#	26-A63
	26-A63#	26-A68	26-A68#	26-A73	26-A73#	26-A85	26-A86	26-A86#	26-A91	26-A93	26-A93#	26-A98	26-A98#	26-B09
	26-B11	26-B11#	26-B16	26-B16#	26-B26	26-B28	26-B28#	26-B33	26-B33#	26-B45	26-B47	26-B47#	26-B52	26-B52#
	26-B62	26-B64	26-B64#	26-B69	26-B69#	26-B79	26-B80	26-B80#	26-B85	26-B86	26-B86#	26-B91	26-B93	26-B93#
	26-B97	26-B97#	26-C42	26-C51	26-C51#	26-C96	26-D05	26-D05#	26-D50	26-D59	26-D59#	26-E04	26-E13	26-E13#
	26-E58	26-E67	26-E67#	26-F11	26-F21	26-F21#	26-F65	26-F75	26-F75#	26-G18	26-G18#	26-G71	26-G83	26-H45
	26-H72	26-H72#												
MS LAB	1-C29#	9-18#	17-277#	17-283#	17-293#	17-299#	17-406#	17-412#	17-570#	17-576#	17-586#	17-592#	17-627#	17-633#
	17-667#	17-673#	17-681#	17-687#	17-833#	17-838#	17-857#	17-863#	17-872#	17-878#	17-887#	17-893#	17-902#	17-908#
	17-:61#	17-:66#	17-:75#	17-<54#	17-<70#	17-=06#	17-=12#	18-102#	18-103#	18-108#	18-109#	18-110#	18-111#	18-112#
	18-113#	18-114#	18-115#	18-116#	18-123#	18-124#	18-125#	18-126#	18-127#	18-128#	18-129#	18-130#	18-131#	18-132#
	18-133#	18-134#	18-135#	18-142#	18-143#	18-144#	18-145#	18-146#	18-147#	18-148#	18-149#	18-150#	18-151#	18-152#
	18-153#	18-154#	18-161#	18-162#	18-163#	18-164#	18-165#	18-166#	18-167#	18-168#	18-169#	18-170#	18-171#	18-172#
	18-173#	18-174#	18-181#	18-182#	18-183#	18-184#	18-185#	18-186#	18-187#	18-188#	18-189#	18-190#	18-191#	18-192#
	18-193#	18-200#	18-201#	18-202#	18-203#	18-204#	18-205#	18-206#	18-207#	18-208#	18-209#	18-210#	18-211#	18-218#
	18-219#	18-220#	18-221#	18-222#	18-223#	18-224#	18-225#	18-226#	18-227#	18-228#	18-229#	18-230#	18-231#	18-238#
	18-239#	18-240#	18-241#	18-242#	18-243#	18-244#	18-245#	18-246#	18-247#	18-248#	18-249#	19-11#	21-25#	21-28#

MSSTL

21-31#	21-34#	21-52#	21-81#	21-85#	21-86#	21-90#	21-94#	21-99#	22-10#	22-17#	22-20#	23-11#	24-10#
24-12#	24-13#	25-10#	26-46#	26-105#	26-145#	26-147#	26-224#	26-251#	26-253#	26-284#	26-317#	26-364#	26-366#
26-419#	26-458#	26-471#	26-480#	26-510#	26-533#	26-539#	26-565#	26-570#	26-593#	26-616#	26-620#	26-643#	26-644#
26-695#	26-724#	26-735#	26-737#	26-746#	26-748#	26-752#	26-762#	26-771#	26-773#	26-777#	26-787#	26-796#	26-798#
26-799#	26-823#	26-825#	26-827#	26-944#	26-:77#	26-:08#	26-:37#	26-:38#	26-:43#	26-:44#	26-:48#	26-:54#	26-:83#
26-:84#	26-:89#	26-:90#	26-:94#	26-:96#	26-<30#	26-<77#	26-<83#	26-30#	26-32#	26-72#	26-94#	26-96#	26->34#
26->49#	26->58#	26->99#	26-?08#	26-?49#	26-?58#	26-?94#	26-242#	26-282#	26-292#	26-293#	26-A02#	26-A03#	26-A21#
26-A26#	26-A37#	26-A38#	26-A40#	26-A45#	26-A56#	26-A58#	26-A63#	26-A68#	26-A73#	26-A85#	26-A86#	26-A91#	26-A93#
26-A98#	26-B09#	26-B11#	26-B16#	26-B26#	26-B28#	26-B33#	26-B45#	26-B47#	26-B52#	26-B62#	26-B64#	26-B69#	26-B79#
26-B80#	26-B85#	26-B86#	26-B91#	26-B93#	26-B97#	26-C42#	26-C51#	26-C96#	26-D05#	26-D50#	26-D59#	26-E04#	26-E13#
26-E58#	26-E67#	26-F11#	26-F21#	26-F65#	26-F75#	26-G18#	26-G71#	26-G83#	26-H45#	26-H72#			
1-C21#	9-18#	17-277	17-277#	17-277#	17-283	17-283#	17-283#	17-293	17-293#	17-293#	17-299	17-299#	17-299#
17-406	17-406#	17-406#	17-412	17-412#	17-412#	17-570	17-570#	17-570#	17-576	17-576#	17-576#	17-586	17-586#
17-586#	17-592	17-592#	17-592#	17-627	17-627#	17-627#	17-633	17-633#	17-633#	17-667	17-667#	17-667#	17-673
17-673#	17-673#	17-681	17-681#	17-681#	17-687	17-687#	17-687#	17-833	17-833#	17-833#	17-838	17-838#	17-838#
17-857	17-857#	17-857#	17-863	17-863#	17-863#	17-872	17-872#	17-872#	17-878	17-878#	17-878#	17-887	17-887#
17-887#	17-893	17-893#	17-893#	17-902	17-902#	17-902#	17-908	17-908#	17-908#	17-:61	17-:61#	17-:66	17-:66#
17-:75	17-:75#	17-<54	17-<54#	17-<54#	17-<70	17-<70#	17-<70#	17-06	17-06#	17-06#	17-12	17-12#	17-12#
18-102	18-102#	18-103	18-103#	18-108	18-108#	18-109	18-109#	18-110	18-110#	18-111	18-111#	18-112	18-112#
18-113	18-113#	18-114	18-114#	18-115	18-115#	18-116	18-116#	18-123	18-123#	18-124	18-124#	18-125	18-125#
18-126	18-126#	18-127	18-127#	18-128	18-128#	18-129	18-129#	18-130	18-130#	18-131	18-131#	18-132	18-132#
18-133	18-133#	18-134	18-134#	18-135	18-135#	18-142	18-142#	18-143	18-143#	18-144	18-144#	18-145	18-145#
18-146	18-146#	18-147	18-147#	18-148	18-148#	18-149	18-149#	18-150	18-150#	18-151	18-151#	18-152	18-152#
18-153	18-153#	18-154	18-154#	18-161	18-161#	18-162	18-162#	18-163	18-163#	18-164	18-164#	18-165	18-165#
18-166	18-166#	18-167	18-167#	18-168	18-168#	18-169	18-169#	18-170	18-170#	18-171	18-171#	18-172	18-172#
18-173	18-173#	18-174	18-174#	18-181	18-181#	18-182	18-182#	18-183	18-183#	18-184	18-184#	18-185	18-185#
18-186	18-186#	18-187	18-187#	18-188	18-188#	18-189	18-189#	18-190	18-190#	18-191	18-191#	18-192	18-192#
18-193	18-193#	18-200	18-200#	18-201	18-201#	18-202	18-202#	18-203	18-203#	18-204	18-204#	18-205	18-205#
18-206	18-206#	18-207	18-207#	18-208	18-208#	18-209	18-209#	18-210	18-210#	18-211	18-211#	18-218	18-218#
18-212	18-212#	18-220	18-220#	18-221	18-221#	18-222	18-222#	18-223	18-223#	18-224	18-224#	18-225	18-225#
18-226	18-226#	18-227	18-227#	18-228	18-228#	18-229	18-229#	18-230	18-230#	18-231	18-231#	18-238	18-238#
18-239	18-239#	18-240	18-240#	18-241	18-241#	18-242	18-242#	18-243	18-243#	18-244	18-244#	18-245	18-245#
18-246	18-246#	18-247	18-247#	18-248	18-248#	18-249	18-249#	19-11	19-11#	21-25	21-25#	21-28	21-28#
21-31	21-31#	21-34	21-34#	21-52	21-52#	21-81	21-81#	21-85	21-85#	21-86	21-86#	21-90	21-90#
21-94	21-94#	21-99	21-99#	22-10	22-10#	22-17	22-17#	22-20	22-20#	23-11	23-11#	24-10	24-10#
24-12	24-12#	24-13	24-13#	25-10	25-10#	26-46	26-46#	26-105	26-105#	26-105#	26-145	26-145#	26-145#
26-147	26-147#	26-224	26-224#	26-224#	26-251	26-251#	26-251#	26-253	26-253#	26-284	26-284#	26-317	26-317#
26-364	26-364#	26-364#	26-366	26-366#	26-419	26-419#	26-419#	26-458	26-458#	26-458#	26-471	26-471#	26-471#
26-480	26-480#	26-510	26-510#	26-533	26-533#	26-539	26-539#	26-565	26-565#	26-570	26-570#	26-593	26-593#
26-616	26-616#	26-620	26-620#	26-643	26-643#	26-644	26-644#	26-695	26-695#	26-724	26-724#	26-735	26-735#
26-737	26-737#	26-746	26-746#	26-746#	26-748	26-748#	26-752	26-752#	26-762	26-762#	26-771	26-771#	26-771#
26-773	26-773#	26-777	26-777#	26-787	26-787#	26-796	26-796#	26-796#	26-798	26-798#	26-799	26-799#	26-823
26-823#	26-825	26-825#	26-827	26-827#	26-944	26-944#	26-:77	26-:77#	26-:08	26-:08#	26-:37	26-:37#	26-:37#
26-:38	26-:38#	26-:43	26-:43#	26-:43#	26-:44	26-:44#	26-:48	26-:48#	26-:54	26-:54#	26-:83	26-:83#	26-:83#
26-:84	26-:84#	26-:89	26-:89#	26-:89#	26-:90	26-:90#	26-:94	26-:94#	26-:96	26-:96#	26-<30	26-<30#	26-<77
26-<77#	26-<83	26-<83#	26-30	26-30#	26-32	26-32#	26-72	26-72#	26-72#	26-72#	26-94	26-94#	26-96
26-96#	26->34	26->34#	26->34#	26->49	26->49#	26->49#	26->58	26->58#	26->58#	26->99	26->99#	26->99#	26-?08
26-?49	26-?49#	26-?49#	26-?58	26-?58#	26-?94	26-?94#	26-242	26-242#	26-282	26-282#	26-292	26-292#	26-292#
26-293	26-293#	26-A02	26-A02#	26-A02#	26-A03	26-A03#	26-A21	26-A21#	26-A26	26-A26#	26-A37	26-A37#	26-A37#
26-A38	26-A38#	26-A40	26-A40#	26-A45	26-A45#	26-A56	26-A56#	26-A56#	26-A58	26-A58#	26-A63	26-A63#	26-A68
26-A68#	26-A73	26-A73#	26-A85	26-A85#	26-A85#	26-A86	26-A86#	26-A86#	26-A91	26-A91#	26-A93	26-A93#	26-A98
26-A98#	26-B09	26-B09#	26-B09#	26-B11	26-B11#	26-B16	26-B16#	26-B16#	26-B26	26-B26#	26-B26#	26-B28	26-B33
26-B33#	26-B45	26-B45#	26-B45#	26-B47	26-B47#	26-B52	26-B52#	26-B52#	26-B62	26-B62#	26-B62#	26-B64	26-B69
26-B69#	26-B79	26-B79#	26-B79#	26-B80	26-B80#	26-B85	26-B85#	26-B85#	26-B86	26-B86#	26-B86#	26-B91	26-B91#
26-B93	26-B93#	26-B97	26-B97#	26-C42	26-C42#	26-C42#	26-C51	26-C51#	26-C96	26-C96#	26-C96#	26-D05	26-D05#
26-D50	26-D50#	26-D50#	26-D59	26-D59#	26-E04	26-E04#	26-E04#	26-E13	26-E13#	26-E13#	26-E58	26-E58#	26-E67
26-E67#	26-F11	26-F11#	26-F11#	26-F21	26-F21#	26-F65	26-F65#	26-F65#	26-F75	26-F75#	26-G18	26-G18#	26-G71

MSWORD	26-G71#	26-G71#	26-G83	26-G83#	26-G83#	26-H45	26-H45#	26-H45#	26-H72	26-H72#				
	1-C94#	9-18#	10-17	10-17#	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8
	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8	11-8#	17-277	17-277	17-277	17-277#
	17-283	17-283	17-283	17-283#	17-293	17-293	17-293	17-293#	17-299	17-299	17-299	17-299	17-299#	17-406
	17-406	17-406#	17-412	17-412	17-412	17-412#	17-570	17-570	17-570	17-570#	17-576	17-576	17-576	17-576#
	17-586	17-586	17-586	17-586#	17-592	17-592	17-592	17-592#	17-627	17-627	17-627	17-627	17-627#	17-633
	17-633	17-633#	17-667	17-667	17-667	17-667#	17-673	17-673	17-673	17-673#	17-681	17-681	17-681	17-681#
	17-687	17-687	17-687	17-687#	17-833	17-833	17-833	17-833#	17-838	17-838	17-838	17-838	17-838#	17-857
	17-857	17-857#	17-863	17-863	17-863	17-863#	17-872	17-872	17-872	17-872#	17-878	17-878	17-878	17-878#
	17-887	17-887	17-887	17-887#	17-893	17-893	17-893	17-893#	17-902	17-902	17-902	17-902	17-902#	17-908
	17-908	17-908#	17-<54	17-<54	17-<54	17-<54#	17-<70	17-<70	17-<70	17-<70#	17-=06	17-=06	17-=06	17-=06#
	17-=12	17-=12	17-=12	17-=12#	21-94	21-94	21-94#	21-94#	26-105	26-105	26-105	26-105	26-105#	26-145
	26-145	26-145#	26-224	26-224	26-224	26-224#	26-251	26-251	26-251	26-251#	26-364	26-364	26-364	26-364#
	26-419	26-419	26-419	26-419#	26-458	26-458	26-458	26-458#	26-471	26-471	26-471	26-471	26-471#	26-746
	26-746	26-746#	26-771	26-771	26-771	26-771#	26-796	26-796	26-796	26-796#	26-;37	26-;37	26-;37	26-;37#
	26-;43	26-;43	26-;43	26-;43#	26-;83	26-;83	26-;83	26-;83#	26-;89	26-;89	26-;89	26-;89	26-;89#	26-=72
	26-=72	26-=72#	26-=94	26-=94	26-=94	26-=94#	26->34	26->34	26->34	26->34#	26->49	26->49	26->49	26->49#
	26->99	26->99	26->99	26->99#	26-?49	26-?49	26-?49	26-?49#	26-@92	26-@92	26-@92	26-@92	26-@92#	26-A02
	26-A02	26-A02#	26-A37	26-A37	26-A37	26-A37#	26-A56	26-A56	26-A56	26-A56#	26-A85	26-A85	26-A85	26-A85#
	26-A91	26-A91	26-A91	26-A91#	26-B09	26-B09	26-B09	26-B09#	26-B26	26-B26	26-B26	26-B26	26-B26#	26-B45
26-B45	26-B45#	26-B62	26-B62	26-B62	26-B62#	26-B79	26-B79	26-B79	26-B79#	26-B85	26-B85	26-B85	26-B85#	
26-B91	26-B91	26-B91	26-B91#	26-C42	26-C42	26-C42	26-C42#	26-C96	26-C96	26-C96	26-C96	26-C96#	26-D50	
26-D50	26-D50#	26-E04	26-E04	26-E04	26-E04#	26-E58	26-E58	26-E58	26-E58#	26-F11	26-F11	26-F11	26-F11#	
26-F65	26-F65	26-F65	26-F65#	26-G71	26-G71	26-G71	26-G71#	26-G83	26-G83	26-G83	26-G83	26-G83#	26-H45	
26-H45	26-H45#	27-16	27-16#	27-17	27-17#	27-18	27-18#	27-19	27-19#	27-20	27-20	27-20#	27-21	
27-22	27-22#	27-23	27-23#	28-15	28-15#	28-16	28-16#	28-17	28-17#	28-18	28-18#	28-18#	28-47	
MSXFER	1-@82#	9-18#												
MANUAL	1-;62#	9-18#	21-81											
MEMORY	1-;66#	9-18#												
OPEN	1-;71#	9-18#												
POINTE	1-;76#	9-18#	10-7											
PRINTB	1-<39#	9-18#	18-102	18-108	18-109	18-110	18-111	18-123	18-124	18-125	18-126	18-142	18-143	18-144
	18-145	18-161	18-162	18-163	18-164	18-165	18-181	18-182	18-183	18-184	18-200	18-201	18-202	18-218
	18-219	18-220	18-221	18-222	18-238	18-239	18-240							
PRINTF	1-<79#	9-18#	17-;61	17-;66	17-;75	21-85	21-90	24-12	26-735	26-737	26-762	26-787	26-823	26-825
PRINTS	1-=19#	9-18#												
PRINTX	1-=59#	9-18#	18-112	18-113	18-114	18-115	18-127	18-128	18-129	18-130	18-131	18-132	18-133	18-134
	18-146	18-147	18-148	18-149	18-150	18-151	18-152	18-153	18-166	18-167	18-168	18-169	18-170	18-171
	18-172	18-173	18-185	18-186	18-187	18-188	18-189	18-190	18-191	18-192	18-203	18-204	18-205	18-206
	18-207	18-208	18-209	18-210	18-223	18-224	18-225	18-226	18-227	18-228	18-229	18-230	18-241	18-242
	18-243	18-244	18-245	18-246	18-247	18-248								
READBU	1-=99#	9-18#												
READBF	1->03#	9-18#	21-25	21-28	21-31	21-34								
RFLAGS	1->08#	9-18#												
SETPRI	1->13#	9-18#	22-10											
SETVEC	1->18#	9-18#												
SLASH	1->24#	9-18#												
STARS	1->38#	9-18#												
SVC	1->52#	9-17#	9-18											
XFER	1-@12#	9-18#												
XFERF	1-@16#	9-18#												
XFERT	1-@20#	9-18#												