

DQ11

CHAR L & INTR  
CZDQHEO

AH-8637E-MC  
FICHE 1 OF 1

AUG 1981  
COPYRIGHT © 75-81  
MADE IN USA





IDENTIFICATION  
-----

PRODUCT CODE: AC-8635E-MC  
PRODUCT NAME: CZDQHEO CHAR L & INTR  
DATE: NOV 1980  
MAINTAINER: DIAGNOSTIC GROUP

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1975, 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL  
DEC

PDP  
DECUS

UNIBUS  
DECTAPE

MASSBUS

## ABSTRACT

THE FUNCTION OF THE DQ11 DIAGNOSTICS ARE TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS.

CURRENTLY THERE ARE SEVEN OFF LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO INSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE AND INSURING THAT DIAGNOSIS OF ERROR WILL BE IMMEDIATE TO PROBLEM  
 NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

THE SEVEN DIAGNOSTICS ARE:

1. (ZDQA [REV] BASIC R/W TEST #1
2. (ZDOB [REV] BASIC R/W TEST #2
3. (ZDQC [REV] BASIC NPR AND INTERRUPT TEST
4. (ZDQD [REV] RECEIVER TRANSMITTER EXERCISER TEST
5. (ZDQE [REV] MISC. RX AND TX TESTS. PLUS BCC TESTS.
6. (ZDQF [REV] CHARACTER DETECT TESTS.
7. (ZDQH [REV] CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.  
 1. (ZDQG [REV] ONLINE TEST. (ITEP OVERLAY)

AND A PARAMETER INPUT PROGRAM IS AVAILABLE  
 1. (ZDQG [REV] DQ11 TRIAL PROGRAM (PARAMETER INPUT) REQUIREMENTS

## EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 4K MEMORY)-WITH OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570) ASR 33 (OR EQUIVALENT)  
 DQ11  
 SYNC MODEM (ONLY REQUIRED FOR ONLINE TEST)

## STORAGE

PROGRAM WILL LOAD AND RUN IN 4K OF MEMORY.  
 LOCATION 1400 THRU 1600 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER DQ11 TRIAL PROGRAM HAS BEEN EXECUTED. OR AFTER THE 'AUTO SIZING' HAS BEEN DONE.

## LOADING PROCEEDURE

## METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND

ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS \*500

MEMORY \*  
 SIZE

|     |     |
|-----|-----|
| 4k  | 17  |
| 8k  | 37  |
| 12k | 57  |
| 16k | 77  |
| 20k | 117 |
| 24k | 137 |
| 28k | 157 |

3.1.1 LOAD THE ADDRESS OF ABS. LOADER (LOC.XXX500)

3.1.2 THEN START

4. STARTING PROCEDURE

A. LOAD LOC. 200

B. SET SWR TO ZERO FOR 'AUTO SIZING' OR LEAVE  
 LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP  
 BY DQ11 TRIAL PROGRAM OR A PREVIOUSLY RUN DQ11 DIAGNOSTIC  
 THAT USED THE 'AUTO SIZING'.  
 \*\*\*\*REFER TO SECTION 4.1 FOR SOFTWARE SWITCH REGISTER OPERATION  
 AND OPTIONS.\*\*\*\*

NOTE: THE SOFTWARE SWITCH REGISTER IS LOCATED AT LOC.176  
 SOFTWARE DISPLAY REGISTER IS LOCATED AT LOC.174

C. THEN START

THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME  
 IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO  
 THE FOLLOWING:

```
'MAP OF DQ11 STATUS'
1400 160010
1402 152300
1404 160020
1406 150310
```

THE ABOVE IS ONLY AN EXAMPLE!

THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD.  
 1400 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE  
 USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS  
 TABLE SEE SECTION 8.4 FOR HELP.

\*\*\*\*IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING  
 WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:  
 SWR-XXXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR'S OPTION)\*\*\*\*  
 NOTE: IF USING THE SOFTWARE SWITCH REGISTER WHEN A HARDWARE  
 SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL NOT  
 TYPE OUT THE TITLE.

THE PROGRAM WILL TYPE 'R'  
AND PROCEED TO RUN THE DIAGNOSTIC

CONTROL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
  - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
  - B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

SW 15 SET: HALT ON ERROR  
SW 14 SET: LOOP ON CURRENT TEST  
SW 13 SET: INHIBIT ERROR PRINT OUT  
SW 12 SET: INHIBIT TYPE OUT/BELL ON ERROR.  
SW 11 SET: INHIBIT ITERATIONS  
SW 10 SET: ESCAPE TO NEXT TEST  
SW 09 SET: LOOP WITH CURRENT DATA  
SW 08 SET: CATCH ERROR AND LOOP ON IT  
SW 07 SET: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.  
SW 06 SET:  
SW 05 SET:  
SW 04 SET:  
SW 03 SET:  
SW 02 SET: LOCK ON SELECTED TEST  
SW 01 SET: RESTART PROGRAM AT SEL: TESTED TEST  
SW 00 SET: RESELECT DQ11'S DESIRED ACTIVE.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DQ11'S DESIRED ACTIVE.  
PLEASE NOTE THAT A MESSAGE IS TYPED  
OUT FOR SWITCH REGISTER BEING EQUAL TO DQ11'S  
ACTIVE. THIS MEANS IF THE SYSTEM HAS  
FOUR DQ11S; BITS 00,01,02,03 WILL  
BE SET IN LOC 'DQACTV'. USING THIS  
SWITCH ALTERS THAT LOCATION; THEREFORE  
IF FOUR DQ11S ARE IN THE SYSTEM  
\*\*\*DO NOT\*\*\* SET SWITCHS GREATER THAN  
SW 03 IN THE UP POSITION. THIS WOULD BE  
A FATAL ERROR. DO NOT SELECT MORE ACTIVE  
DQ11S THAN HAS BEEN GIVEN INFORMATION  
ABOUT IN TRIAL PROGRAM.

- A: LOAD ADDRESS 200
- B: START WITH SW 00=1
- C: PROGRAM WILL TYPE MESSAGE
- D: CONTINUE THE BINARY NUMBER OF DQ11S DESIRED ACTIVE  
EXAMPLE: 1=1 DQ11; 3=2 DQ11; 7=3 DQ11; 17=4 DQ11 37=5 DQ11 ETC.
- E: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05, 11/04, 11/34)
- F: CONTINUE WITH ANY OTHER SWITCH SETTINGS DESIRED.

SW 01 IT IS STRONGLY SUGGESTED THAT  
AT LEAST ONE PASS HAS BEEN MADE  
BEFORE TRYING TO SELECT A TEST  
THAT IS NOT IN THE ORDER OF SEQUENCE  
THE REASON BEING IS THAT THE  
PROGRAM HAS TO CLEAR AREAS AND SET  
UP PARAMETERS. ALSO WHEN A TEST IS  
SELECTED ALWAYS START AT THE VERY  
BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA:  
THIS SWITCH WILL ONLY WORK IF  
CALL 'SCOPI' IS IN THE TEST.  
THE REASON BEING THAT MOST TESTS  
DEAL WITH BLOCKS OF DIFFERENT DATA  
TO BE SENT OR RECEIVED ALL AT ONCE  
THUS IN BLOCK DATA; ONE PATTERN CANN'T BE SINGLED OUT.

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

- 1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
- 2. SW 13 DELETE ERROR PRINTOUT.
- 3. SW 15 HALT ON THE ERROR.
- 4. SW 08 GOTO BEGINNING OF THE TEST.
- 5. SW 10 GOTO NEXT TEST ON ERROR.

\*\*\*HALT (ERROR) ROUTINE SUPPORTS <^G> OPERATION\*\*\*

SCOPE SWITCHES

1. SW 09 (IF ENABLED BY "SCOPI")
2. SW 14
3. SW 11

\*\*\*\*SCOPE ROUTINE WILL SUPPORT <^G> OPERATION\*\*\*\*

#### STARTING ADDRESS

STARTING ADDRESS IS AT 000200  
THERE ARE NO OTHER STARTING ADDRESSES  
FOR THE DQ11 DIAGNOSTICS PREVIOUSLY MENTIONED

NOTE: IF ADDRESS 000042 IS NON-ZERO  
THE PROGRAM ASSUMES IT IS UNDER  
ACT11 OR DDP CONTROL AND WILL ACT ACCORDINGLY  
AFTER \*ALL\* AVAILABLE DQ11'S ARE TESTED  
THE PROGRAM WILL RETURN TO "DDP2" OR "ACT-11".

#### OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION  
FOUR WILL BE PRINTED.

AND PROGRAM WILL BEGIN RUNNING THE  
DIAGNOSTIC

#### PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15-1)  
WHEN EVER AN ERROR OCCURS
2. CLEAR SW 15
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

THE TEST NUMBER AND PC WILL BE TYPED OUT AND  
POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST)  
TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE  
PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION  
CONCERNING THE ERROR REPORT; LOOK IN THE LISTING  
FOR THAT TEST NUMBER WHICH WAS TYPED OUT  
AND THEN NOTE THE PC OF THE ERROR REPORT  
THIS WAY THE EXACT FUNCTIONING OF THE TEST  
CAN BE INTERPRETED

#### ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE  
A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN  
ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL  
INFORMATION WILL BE SUPPLIED THROUGH THE ERROR MESSAGE  
WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE  
ERROR.

#### ERROR RECOVERY

IF FOR SOME REASON THE DQ11 SHOULD  
 "HANG THE BUS" (GAIN CONTROL OF BUS SO THAT  
 CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT  
 OR POWER DOWN/UP IS NECESSARY FOR OPERATOR  
 TO REGAIN CONTROL OF CPU.  
 IF THIS SHOULD HAPPEN; LOOK IN LOCATION  
 "TSTNO" (ADDRESS 1226) FOR THE NUMBER OF THE TEST THAT  
 WAS RUNNING AT THE TIME OF THE CATASTROPHIC  
 ERROR.  
 IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO  
 WHAT THE DQ11 WAS DOING AT THE TIME OF THE ERROR.

4.3 \*\*\*\*\*HALT RECOVERY WHEN USING SOFTWARE SWITCH REGISTER\*\*\*\*\*

IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT  
 THE OPERATOR IS REQUIRED TO TYPE A <^G> BEFORE DEPRESSING CONTINUE.  
 THE FOLLOWING WILL BE TYPED:  
 SWR-XXXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR OPTION)

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

7.2 OPERATING RESTRICTIONS

DQ11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE  
 FIRST AND ONLY THE FIRST RUNNING OF ANY DQ11 DIAGNOSTIC  
 NOTE: IF NO PROGRAM OTHER THAN A  
 DQ11 DIAGNOSTIC WAS LOADED AFTER DQ11 TRIAL OR  
 IF CORE MEMORY HAS NOT BEEN CHANGED; OR IF THERE  
 IS NO DQ11 CONFIGURATION CHANGES; THE  
 DQ11 TRIAL PROGRAM NEED NEVER BE RUN AGAIN.  
 HOWEVER IF ANY OF THE ABOVE HAVE BEEN VIOLATED  
 THE DQ11 TRIAL PROGRAM MUST BE RUN AGAIN  
 BEFORE RUNNING THE DIAGNOSTICS  
 NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING  
 THE "AUTO SIZING" WHEN PROGRAM IS INITIALLY STARTED  
 WITH SW07-0.

8. MISCELLANEOUS

8.1 EXECUTION TIME

8.2 PASS COMPLETE

WHEN THE DIAGNOSTIC HAS COMPLETED  
 A PASS THE FOLLOWING IS AN EXAMPLE  
 OF THE PRINT OUT TO BE EXPECTED.

END PASS AC-8635E-MC CSR: 160000 VEC: 300 PASSES: 000001 ERRORS: 000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE  
 NOT NECESSARILY THE VALUES FOR THE DEVICE



THEY ARE ONLY FOR THIS EXAMPLE.

4.3 TST1 (MINI MONITOR)

THE VERY FIRST 'TEST' (TST1)  
IS ~~NOT~~ A TEST OF THE DQ11 HARDWARE  
IT IS A MINI-MONITOR USED TO CYCLE DQ11 IN THE  
SYSTEM THROUGH THE DIAGNOSTIC.

REMEMBER: TST1 IS NOT A TEST OF DQ11 HARDWARE!!!!!!!

4.4 KEY LOCATIONS

RETURN (1214) CONTAINS THE ADDRESS WHERE PROGRAM WILL  
RETURN WHEN ITERATION COUNT IS REACHED  
OR IF LOOP ON TEST IS ASSERTED.  
NEXT (1216) CONTAINS THE ADDRESS OF THE NEXT TEST  
TO BE PERFORMED.  
TSTNO (1226) CONTAINS THE NUMBER OF THE TEST NOW  
BEING PERFORMED.  
RUN (1304) THE BIT IN 'RUN' ALWAYS POINTS ONE  
PAST THE DQ11 CURRENTLY BEING TESTED.  
EXAMPLE:  
(RUN) 1304/0000000001000000  
MEANS THAT DQ11 NO.05 IS THE DQ11 NOW  
RUNNING.

DQCR00-DQCR17  
DQST00-DQST17  
(1400)-(1476)

THESE LOCATIONS CONTAIN THE INFORMATION  
NEEDED TO TEST UP TO 16 (DECIMAL) DQ11S  
SEQUENTIALLY. THEY CONTAIN THE CSR, VECTOR  
AND STATUS CONCERNING THE CONFIGURATION  
OF EACH DQ11.

DQACTV (1500) EACH BIT SET IN THIS LOCATION INDICATES  
THAT THE ASSOCIATED DQ11 WILL BE TESTED  
IN TURN.  
EXAMPLE:  
(DQACTV) 1500/0000000000011111  
MEANS THAT DQ11 NO. 00,01,02,03,04  
WILL BE TESTED.  
EXAMPLE:  
(DQACTV) 1500/0000000000010001  
MEANS THAT DQ11 NO. 00,04  
WILL BE TESTED.

DQCSR (1506) CONTAINS THE RECEIVER CSR OF THE  
CURRENT DQ11 UNDER TEST.

DQSTAT (1510) CONTAINS THE STATUS OF THE CURRENT  
DQ11 UNDER TEST.

BIT 15 SET: TWO SYNC CHARS/ONE SYNC CHAR  
BIT 14 SET: TEST JUMPER INSTALLED/NOT INSTALLED  
BIT 13 SET: BB OPTION INSTALLED/NOT INSTALLED  
BIT 12 SET: BA OPTION INSTALLED/NOT INSTALLED  
BIT 11 SET: ACTIVE ON FIRST NON-SYNC/ACTIVE AFTER NO. OF SYNC  
BIT 10 SET: AB OPTION INSTALLED/NOT INSTALLED  
BIT 09 SET: ODD VRC/EVEN VRC

## BIT 00-08 VECTOR 'A' OF DEVICE

8.5 \*\*\* METHOD OF AUTO SIZING \*\*\*

## 8.5.1 FINDING THE CONTROL STATUS REGISTER.

WHEN LOOKING FOR THE CSR IT IS NECESSARY TO TAKE CARE THAT WHEN A CSR IS FOUND THAT IT IS INDEED A DQ11. THAT IS THE METHOD OF MY MADNESS FOR THIS ROUTINE. AN ATTEMPT TO CLEAR THE MISC. REGISTER IS TRIED IF A TIME-OUT TRAP OCCURS POINTERS ARE UPDATED AND ATTEMPTED AGAIN. IF NO TIME-OUT; THE RECEIVER 'ACTIVE BIT' (BIT 12) IS SET AND A \*COMPARE\* FOR BOTH SYNC1 AND SYNC 2 IS DONE AT THE MISC. REGISTER. IF THEY ARE THERE THIS IS A DQ11. THE INFORMATION IS STORED AWAY.

## 8.5.2 ONE SYNC BIT OR TWO?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE THE PRESENTS OF ONE SYNC OR TWO. THE PROGRAM ASSUMES TWO SYNC CHARS. NOTE: THIS ASSUMPTION MAY BE ALTERED AFTER AUTO SIZING BY ALTERING BIT 15 IN APPRIOATE DQSTXX: LOCATION.

## 8.5.3 'BB' OPTION INSTALLED?

TO SENSE FOR THE 'BB' OPTION THE PROGRAM SELECTS THE CHARACTER DET. REGISTER AND THE LOADS IN ALL 1'S; IF ANY ONE OR COMBINATION OF BITS ARE SET THE BB OPTION IS ASSUMED TO EXIST.

## 8.5.4 'AB' OPTION INSTALLED?

TO SENSE FOR THE 'AB' OPTION THE PROGRAM SELECTS THE POLYNOMIAL REGISTER AND WRITES ALL 1'S INTO IT; IF ANY ONE OR COMBINATION OF BITS ARE SET THE AB OPTION IS ASSUMED TO EXIST.

## 8.5.5 'BA' OPTION INSTALLED?

TO SENSE FOR 'BA' OPTION REQUEST TO SEND AND DATA TERMINAL READY ARE SET; IF EITHER ONE OR BOTH ARE SET THE PROGRAM ASSUMES THE BA OPTION EXISTES

## 8.5.6 JUMPER ON END OF CABLE? \*\*\*NOTE:CZDQE ONLY\*\*\*

THE PROGRAM CHECKS TO SEE IF EITHER OR BOTH CLEAR TO SEND AND CARRIER ARE SET; IF SO THE PROGRAM ASSUMES THE TEST JUMPER IS ON THE END OF THE CABLE.

## 8.5.7 ACTIVE ON FIRST NON-SYNC?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE FOR WHEN THE DQ11 GOES ACTIVE THE PROGRAM ASSUMES 'ACTIVE ON FIRST NON-SYNC'. NOTE: THIS CAN BE CHANGED BY ALTERING BIT 11 IN THE APPRIOATE DQSTXX: AFTER AUTO SIZING

## 8.5.8 SET FOR ODD OR EVEN PARITY?

AS ABOVE TOO MUCH HARDWARE IS NEEDED TO SENSE WHICH PARITY WAS SELECTED. SO THE PROGRAM ASSEMBLES ODD PARITY.  
NOTE: THIS CAN BE CHANGED BY ALTERING BIT 9 IN APPROPRIATE DQSTXX: LOCATION. AFTER AUTO SIZING

## 8.5.9 FINDING THE VECTOR.

THE PROGRAM SETS 'PRIMARY DONE', 'SECONDARY DONE', AND 'INTERUPT ENABLE' AND LOOKS FOR AN INTERRUPT. IF IT INTERRUPTS IT IS PICKED UP AND STORED AWAY. IF NO INTERRUPT OCCURS THE PROGRAM ASSUMES VECTOR = 300. THIS PROBLEM WILL BE FIXED IN ONE OF THE DIAGNOSTICS AND \*AUTO SIZING\* SHOULD BE REDONE TO GET THE CORRECT VECTOR.

9. PROGRAM DESCRIPTION  
CONTAINED WITHIN LISTING
10. LISTING  
FOLLOWING



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

```
:CZDQHE0/<377>/CHAR L & INTR
:COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
:REVISED 16-DEC-76 BY R. BLACK
: A)SUPPORTS SOFTWARE SWITCH REGISTER
: B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
: BY <^G>.
: STARTING PROCEDURE
: LOAD PROGRAM
: LOAD ADDRESS 000200
: PRESS START
: PROGRAM WILL TYPE 'CZDQHE0/<377>/CHAR L & INTR ''
: PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
: AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
: AND THEN RESUME TESTING
```

: SWITCH REGISTER OPTIONS

```
100000 SW15=100000 :-1,HALT ON ERROR
040000 SW14=40000 :=1,LOOP ON CURRENT TEST
020000 SW13=20000 : 1,INHIBIT ERROR TYPEOUT
010000 SW12=10000 :=1,DELETE TYPEOUT/BELL ON ERROR.
004000 SW11=4000 :=1,INHIBIT ITERATIONS
002000 SW10=2000 :-1,ESCAPE TO NEXT TEST ON ERROR
001000 SW09=1000 :-1, OOP WITH CURRENT DATA
000400 SW08=400 :=1,LOOP ON ERROR
000100 SW06=100
000040 SW05=40
000020 SW04=20
000010 SW03=10
000004 SW02=4 :LOCK ON TEST SELECT
000002 SW01=2 :RESTART PROGRAM AT SELECTED TEST
000001 SW00=1 :RESELECT DQ11 DESIRED ACTIVE
:NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
```

```
558
559
560      ;REGISTER DEFINITIONS
561
562      000000      R0=%0      ;GENERAL REGISTER
563      000001      R1=%1      ;GENERAL REGISTER
564      000002      R2=%2      ;GENERAL REGISTER
565      000003      R3=%3      ;GENERAL REGISTER
566      000004      R4=%4      ;GENERAL REGISTER
567      000005      R5=%5      ;GENERAL REGISTER
568      000006      SP=%6      ;PROCESSOR STACK POINTER
569      000007      PC=%7      ;PROGRAM COUNTER
570
571      ;LOCATION EQUIVALENCIES
572
573      177570      DSWR= 177570 ;HARDWARE SWITCH REGISTER LOC.
574      177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
575      177776      PS=177776 ;PROCESSOR STATUS WORD
576      001200      STACK=1200 ;START OF PROCESSOR STACK
577
578      ;INSTRUCTION DEFINITIONS
579
580      005746      PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
581      005726      POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD
582      010046      PUSHRO=10046 ;SAVE R0 ON STACK
583      012600      POPRO=12600 ;RESTORE R0 FROM STACK
584      024646      PUSH2SP=24646 ;DECREMENT STACK TWICE
585      022626      POP2SP=22626 ;INCREMENT STACK TWICE
586      .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
587
588
589      100000      BIT15=100000
590      040000      BIT14=40000
591      020000      BIT13=20000
592      010000      BIT12=10000
593      004000      BIT11=4000
594      002000      BIT10=2000
595      001000      BIT9=1000
596      000400      BIT8=400
597      000200      BIT7=200
598      000100      BIT6=100
599      000040      BIT5=40
600      000020      BIT4=20
601      000010      BIT3=10
602      000004      BIT2=4
603      000002      BIT1=2
604      000001      BIT0=1
605
606
607      ;DQ11 OPTIONAL DEFINITIONS
608
609      002000      ABBIT=2000
610      004000      ACTBIT=4000
611      010000      BABIT=10000
612      020000      BBBIT=20000
613      040000      JUMBIT=40000
```

GENERAL DEFINITIONS AND EQUIVALENCIES

614 001000 ODDBIT=1000  
615 100000 SYNBIT=100000  
616  
617

:DQ11 SECONDARY REGISTER DEFINITIONS

|     |        |           |                                     |
|-----|--------|-----------|-------------------------------------|
| 620 | 000000 | RXBA.P=0  | :RECEIVER BUS ADDRESS PRIMARY.      |
| 621 | 000001 | RXWC.P=1  | :RECEIVER WORD COUNT PRIMARY.       |
| 622 | 000002 | TXBA.P=2  | :TRANSMITTER BUS ADDRESS PRIMARY.   |
| 623 | 000003 | TXWC.P=3  | :TRANSMITTER BUS ADDRESS PRIMARY.   |
| 624 | 000004 | RXBA.S=4  | :RECEIVER BUS ADDRESS SECONDARY.    |
| 625 | 000005 | RXWC.S=5  | :RECEIVER WORD COUNT SECONDARY.     |
| 626 | 000006 | TXBA.S=6  | :TRANSMITTER BUS ADDRESS SECONDARY. |
| 627 | 000007 | TXWC.S=7  | :TRANSMITTER WORD COUNT SECONDARY.  |
| 628 |        |           |                                     |
| 629 | 000010 | CHARDT=10 | :CHARACTER DETECT REGISTER.         |
| 630 | 000011 | SYNC.=11  | :SYNC REGISTER.                     |
| 631 | 000012 | MISC.-12  | :MISCELLANEOUS REGISTER.            |
| 632 | 000013 | TX.MUX=13 | :TRANSMITTER MUX REGISTER.          |
| 633 | 000014 | SEQ.=14   | :SEQUENCE REGISTER.                 |
| 634 | 000015 | RX.BCC=15 | :RECEIVER BCC REGISTER.             |
| 635 | 000016 | TX.BCC=16 | :TRANSMITTER BCC REGISTER.          |
| 636 | 000017 | POLY.-17  | :POLYNOMIAL REGISTER.               |
| 637 |        |           |                                     |
| 638 |        |           |                                     |



```

639 ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
640 000000 .=0
641 ;STANDARD INTERRUPT VECTORS
642
643 .-24
644 000024 016222 .PFAIL ;POWER FAIL HANDLER
645 000026 000340 340 ;SERVICE AT LEVEL 7
646 000030 015672 .HLT ;ERROR HANDLER
647 000032 000340 340 ;SERVICE AT LEVEL 7
648 000034 015640 .TRPSRV ;GENERAL HANDLER DISPATCH SERVICE
649 000036 000340 340 ;SERVICE AT LEVEL 7
650 000046 .=46
651 000046 014420 LOGICAL ;ACT HOOKS
652 . 52
653 000052 000000 .WORD 0
654 ;THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
655 ;TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
656 ;FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
657 ;NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
658 ;EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
659 ;TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
660 000056 .-56
661
662 000056 VECMAP:
663 000056 010120 1$: MOV R1,(R0)+ ;START FILLING THE VECTOR AREA
664 000060 012721 000004 MOV #4,(R1)+ ;WITH .+2: IOT (4)
665 000064 022021 CMP (R0)+,(R1)+ ;UPDATE THE POINTERS
666 000066 020127 001000 CMP R1,#1000 ;IS ALL FLOATING VECTOR AREA DONE
667 000072 101771 BLOS 1$ ;BR IF NOT ALL DONE
668 000074 012737 000146 000020 MOV #4$,@#20 ;SET FOR IOT TRAP BY DQ11
669 000102 013737 001500 001244 MOV DQACTV,TEMP1 ;GET THE ACTIVE DQ11 S
670 000110 006037 001244 2$: ROR TEMP1 ;ARE YOU ACTIVE.. DQ11
671 000114 103023 BCC 5$ ;IF CARRY CLEAR.. NO MORE DQ11S
672 000116 005037 177776 CLR PS ;CLEAR PS
673 000122 005722 TST (R2)+ ;PUT POINTER TO STATUS TABLE
674 000124 012772 000340 177776 MOV #340,@-2(R2) ;TRY AND SET PRI/SEC DONE AND IE
675 000132 105200 INCB R0 ;DELAY.....
676 000134 001376 BNE .-2 ;.....DELAY
677 000136 112712 000300 MOVB #300,(R2) ;NO INTERRUPT ASSUME 300 FIX IN TEST C
678 000142 005722 3$: TST (R2)+ ;UPDATE POINTERS
679 000144 000761 BR 2$ ;GO DO IT AGAIN
680 000146 051612 4$: BIS (SP),(R2) ;ENTERED BY IOT TRAP BY DQ11
681 000150 042712 000007 BIC #7,(R2) ;CLEAR UNWANTED BITS
682 000154 022626 CMP (SP)+,(SP)+ ;POP IOT JUNK OFF STACK
683 000156 012716 000142 MOV #3$,(SP) ;SET RETURN PC ON STACK
684 000162 000002 RTI ;GO HOME.
685 000164 000207 5$: RTS PC ;ALL SIZING IS DONE
686
687 ;****SOFTWARE SWITCH REGISTER****
688 .=174
689 000174 000000 DISPREG: 0 ;SOFTWARE DISPLAY REGISTER
690 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
691
692 ;PROGRAM START
693
694 000200 .-200

```

```

695 000200 000137 001512          JMP      .START          ;GO TO START OF PROGRAM
696
697          000220          ;=220
698 000220 012702 001400  CSRMAP: MOV      #1400,R2          ;CLEAR ALL STATUS TABLE
699 000224 005022          CLR      (R2)+          ;DO CLEAR
700 000226 022702 001512          CMP      #1512,R2          ;ALL TABLE DONE
701 000232 001374          BNE     .-6              ;BR IF MORE TO GO
702 000234 005037 001504          CLR      DQNUM          ;SET NUMBER OF DQ11S TO 0
703 000240 012702 001400          MOV      #1400,R2          ;SET TABLE POINTER
704 000244 012701 160000          MOV      #160000,R1       ;GET FIRST FLOATING ADDRESS
705 000250 012737 000614 000004          MOV      #5$,@#4         ;SET FOR TIME OUT TRAP--NO DEVICE--
706 000256 112761 000012 000005 1$:  MOVVB   #12,5(R1)        ;TRY AND SEL MISC REGISTER
707 000264 005061 000006          CLR      6(R1)           ;TRY AND CLEAR MISC REG
708 000270 012711 010000          MOV      #10000,(R1)      ;TRY AND SET RX ACTIVE
709 000274 022761 030000 000006          CMP      #30000,6(R1)    ;LOOK FOR SYNC 1 AND SYNC 2
710 000302 001071          BNE     2$              ;THIS IS NOT A DQ11 IF I BRANCH
711 000304 010122          MOV      R1,(R2)+        ;NOW THIS IS A DQ11 --STORE CSR
712 000306 052712 100000          BIS      #SYNBIT,(R2)    ;SET FOR TWO SYNC (HARS
713 000312 005011          CLR      (R1)           ;CLEAR DQ ACTIVE BIT
714 000314 112761 000010 000005          MOVVB   #10,5(R1)        ;SEL CHAR DET REGISTER
715 000322 012761 177777 000006          MOV      #-1,6(R1)       ;WRITE INTO CHAR DET REG
716 000330 005761 000006          TST     6(R1)           ;WAS THE REGISTER WRITTEN?
717 000334 001402          BEQ     .+(             ;APPARENTLY NO BB OPTION.
718 000336 052712 020000          BIS      #BBBIT,(R2)    ;SET FOR BB OPTION
719 000342 112761 000017 000005          MOVVB   #17,5(R1)        ;SEL POLYNO. REGISTER
720 000350 012761 177777 000006          MOV      #-1,6(R1)       ;WRITE POLYNO.REGISTER
721 000356 005761 000006          TST     6(R1)           ;WAS REG WRITTEN??
722 000362 001402          BEQ     .+6            ;BR IF NO AB OPTION
723 000364 052712 002000          BIS      #ABBIT,(R2)    ;SET FOR AB OPTION
724 000370 012761 001400 000002          MOV      #1400,2(R1)     ;TRY TO SET .DTR. .RS.
725 000376 032761 001400 000002          BIT      #1400,2(R1)    ;DID ANY OF THEM SET
726 000404 001402          BEQ     .+6            ;BR IF NO BA OPTION
727 000406 052712 010000          BIS      #BABIT,(R2)    ;SET FOR BA OPTION
728 000412 032761 030000 000002          BIT      #30000,2(R1)   ;DID .CS. .CO. SET
729 000420 001402          BEQ     .+6            ;BR IF NO JUMPER
730 000422 052712 040000          BIS      #JUMBIT,(R2)   ;SET FOR JUMPER
731 000426 052712 004000          BIS      #ACTBIT,(R2)   ;SET FOR ACTIVE ON FIRST NON-SYNC
732 000432 052712 001000          BIS      #ODDBIT,(R2)  ;SET FOR ODD VRC.....
733 000436 005722          TST     (R2)+          ;POP POINTER
734 000440 005011          CLR      (R1)           ;CLEAR RCSR
735 000442 005061 000002          CLR      2(R1)          ;CLEAR TCSR
736 000446 005061 000002          CLR      2(R1)          ;CLEAR AGAIN
737 000452 005061 000004          CLR      4(R1)          ;CLEAR ERROR REG
738 000456 005061 000006          CLR      6(R1)          ;CLEAR SEC REG
739 000462 005237 001504          INC     DQNUM          ;UPDATE NUMBER OF DQ11S
740 000466 062701 000010 2$:  ADD     #10,R1          ;UPDATE CSR POINTER BY 10 (8)
741 000472 022701 164000          CMP     #164000,R1      ;HAVE ALL FLOATING ADDRESSES BEEN CHECKED??
742 000476 001267          BNE     1$              ;BR IF NOT ALL DONE
743 000500 005037 001500          CLR     DQACTV          ;ZERO ACTIVE DQ11S
744 000504 005737 001504          TST     DQNUM          ;WERE ANY DQ11S FOUND
745 000510 001434          BEQ     4$              ;HEY BUDDY. NO DQ11S FOUND IN SYSTEM
746 000512 013701 001504          MOV     DQNUM,R1        ;SAVE NUMBER OF DQ11S
747 000516 010137 001276          MOV     R1,SAVNUM       ;SAVE NUMBER FOR ACT11
748 000522 000241 3$:  CLC          ;CLEAR CARRY
749 000524 006137 001500          ROL     DQACTV          ;   ACTIVE ADDRESS
750 000530 005237 001500          INC     DQACTV          ;SET BIT 0

```

```

751 000534 005301          DEC      R1          :DEC NUMBER OF DQ11S
752 000536 001371          BNE      3$          :BR IF MORE TO GO
753 000540 012737 000006 000004  MOV      #6,@#4      :RESET TIME OUT VECTOR
754 000546 013737 001500 001502  MOV      DQACTV,SAVACT :SAVE ACTIVE
755 000554 012737 000340 000022  MOV      #340,@#22   :SET IOT TRAP PRIO: TO 7
756 000562 012702 001400          MOV      #1400,R2    :SET TABLE POINTER
757 000566 012700 000300          MOV      #300,R0     :SET VECTOR START
758 000572 012701 000302          MOV      #302,R1     :SET VECTOR+2 START
759 000576 000137 000056          JMP      VECMAP      :GO FIND THE VECTORS
760 000602 104402          4$: TYPE          :TYPE MESSAGE
761 000604 016563          MERR2          :I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
762 000606 005000          CLR      R0         :
763 000610 000000          HALT          :HOW CAN I TEST NO DQ11S
764 000612 000776          BR          :DON'T LET OPR HIT CONT. SW
765 000614 012716 000466 5$: MOV      #-2          :ENTERED BY TIME OUT TRAP
766 000620 000002          RTI          :GO HOME.
767
768
769          001000          .=1000
770 001000 005377 055103 050504  MTITLE: .ASCIZ <377><12>/CZDQHE0/<377>/CHAR L & INTR /<377>
771 001006 042510 177460 044103
772 001014 051101 046040 023040
773 001022 044440 052116 020122
774 001030 000377
775
776          001200          .=1200
777          :INDIRECT POINTERS
778
779 001200 177570          SWR: 177570          :SWITCH REGISTER POINTER
780 001202 177570          LIGHTS: 177570      :DISPLAY REGISTER POINTER
781 001204 177560          TKCSR: 177560      :TELETYPE KEYBOARD CONTROL REGISTER
782 001206 177562          TKDBR: 177562      :TELETYPE KEYBOARD DATA BUFFER
783 001210 177564          TPCSR: 177564      :TELEPRINTER CONTROL REGISTER
784 001212 177566          TPDBR: 177566      :TELEPRINTER DATA BUFFER
785
786          :PROGRAM CONTROL PARAMETERS
787
788 001214 000000          RETURN: 0          :SCOPE ADDRESS FOR LOOP ON TEST
789 001216 000000          NEXT: 0           :ADDRESS OF NEXT TEST TO BE EXECUTED
790 001220 000000          LOCK: 0           :ADDRESS FOR LOCK ON CURRENT DATA
791 001222 000003          ICOUNT: 3         :NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
792 001224 000000          LPCNT: 0          :NUMBER OF ITERATIONS COMPLETED
793 001226 000000          TSTNG: 0          :NUMBER OF TEST IN PROGRESS
794 001230 000000          PASCNT: 0         :NUMBER OF PASSES COMPLETED
795 001232 000000          ERRCNT: 0         :TOTAL NUMBER OF ERRORS
796 001234 000000          LSTERR: 0         :PC OF LAST ERROR CALL
797
798          :PROGRAM VARIABLES
799
800 001236 000000          CHAR1: 0
801 001240 000000          CHAR2: 0
802 001242 000000          CHAR3: 0
803 001244 000000          TEMP1: 0         :TEMPORARY STORAGE
804 001246 000000          TEMP2: 0         :TEMPORARY STORAGE
805 001250 000000          TEMP3: 0         :TEMPORARY STORAGE
806 001252 000000          TEMP4: 0         :TEMPORARY STORAGE

```



CZDQH MACY11 30A(1052) 03-DEC-80 08:29  
CZDQHE.P11 03-DEC-80 08:27

PAGE 18  
PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0017

|     |        |        |         |         |                          |
|-----|--------|--------|---------|---------|--------------------------|
| 807 | 001254 | 000000 | TEMP5:  | 0       | :TEMPORARY STORAGE       |
| 808 | 001256 | 000000 | SAVR0:  | 0       | :R0 STORAGE              |
| 809 | 001260 | 000000 | SAVR1:  | 0       | :R1 STORAGE              |
| 810 | 001262 | 000000 | SAVR2:  | 0       | :R2 STORAGE              |
| 811 | 001264 | 000000 | SAVR3:  | 0       | :R3 STORAGE              |
| 812 | 001266 | 000000 | SAVR4:  | 0       | :R4 STORAGE              |
| 813 | 001270 | 000000 | SAVR5:  | 0       | :R5 STORAGE              |
| 814 | 001272 | 000000 | SAVSP:  | 0       | :STACK POINTER STORAGE   |
| 815 | 001274 | 000000 | SAVPC:  | 0       | :PROGRAM COUNTER STORAGE |
| 816 | 001276 | 000000 | SAVNUM: | 0       |                          |
| 817 | 001300 | 000001 | CREAM:  | .BLKW 1 |                          |
| 818 | 001302 | 000000 | RUNFLG: | 0       |                          |
| 819 | 001304 | 000000 | RUN:    | 0       |                          |
| 820 | 001306 | 000000 | RUNCNT: | 0       |                          |

```

821
822                ;PROGRAM CONTROL FLAGS
823
824 001310         000      INIFLG: .BYTE 0           ;PROGRAM INITIALIZATION FLAG
825 001311         000      STFLG:  .BYTE 0           ;TEST START FLAG
826 001312         000      ERRFLG: .BYTE 0           ;ERROR OCCURED FLAG
827 001313         000      LOKFLG: .BYTE 0           ;LOCK ON CURRENT TEST FLAG
828                000000  $Y-0
829
830                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
831                ;POINTERS TO SUBROUTINES CAN BE FOUND
832                ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
833
834                ;*****
835                ;*****
836 001314         104400  .TRPTAB:
837                SCOPE=TRAP+0           ;CALL TO SCOPE LOOP AND ITERATION HANDLER
838 001314         014474  .SCOPE
839                SCOPI=TRAP+1          ;CALL TO LOOP ON CURRENT DATA HANDLER
840 001316         014606  .SCOPI
841                TYPE-TRAP+2          ;CALL TO TELETYPE OUTPUT ROUTINE
842 001320         014626  .TYPE
843                INSTR-TRAP+3         ;CALL TO ASCII STRING INPUT ROUTINE
844 001322         014734  .INSTR
845                INSTER=TRAP+4        ;CALL TO INPUT ERROR HANDLER
846 001324         015052  .INSTER
847                PARAM=TRAP+5         ;CALL TO NUMERICAL DATA INPUT ROUTINE
848 001326         015104  .PARAM
849                SAV05=TRAP+6         ;CALL TO REGISTER SAVE ROUTINE
850 001330         015320  .SAV05
851                RES05=TRAP+7         ;CALL TO REGISTER RESTORE ROUTINE
852 001332         015360  .RES05
853                CONVRT=TRAP+10       ;CALL TO DATA OUTPUT ROUTINE
854 001334         015412  .CONVRT
855                CNVRT-TRAP+11        ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
856 001336         015416  .CNVRT
857                MSTCLR-TRAP+12       ;CALL TO ISSUE MASTER CLEAR
858 001340         013032  .MSTCLR
859                MEMCLR-TRAP+13       ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
860 001342         012706  .MEMCLR
861                CKSWR=TRAP+14        ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
862 001344         016320  .CKSWR
863                CNTLU-TRAP+15        ;CALL TO ALLOW LOADING OF SWREG FROM TTY
864 001346         016374  .CNTLU
865
866                ;*****
867                ;*****
868
869                ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
870
871 001350         000000  DQRVEC: 0           ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
872 001352         000000  DQRLVL: 0          ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
873 001354         000000  DQTVEC: 0          ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
874 001356         000000  DQTLVL: 0          ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
875 001360         000000  DQRCSR: 0          ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
876 001362         000000  DQRCSH: 0          ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER

```

CZDQH MACY11 30A(1052) 03-DEC-80 08:29 PAGE 20  
 CZDQHE.P11 03-DEC-80 08:27

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

SEQ 0019

```

877 001364 000000      DQTCR: 0          ; POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
878 001366 000000      DQERR: 0         ; POINTER TO DQ11 ERROR REGISTER
879 001370 000000      DQREG: 0         ; POINTER TO HIGH BYTE OF ERROR REGISTER
880 001372 000000      DQSEC: 0         ; POINTER TO DQ11 SECONDARY REGISTER
881 001374 000000      DQSECH: 0        ; POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
882
883
884
885                      ;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS
886
887                      .=1400
888 001400 000001      DQCR00: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 00
889 001402 000001      DQST00: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
890 001404 000001      DQCR01: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 01
891 001406 000001      DQST01: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
892 001410 000001      DQCR02: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 02
893 001412 000001      DQST02: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
894 001414 000001      DQCR03: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 03
895 001416 000001      DQST03: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
896 001420 000001      DQCR04: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 04
897 001422 000001      DQST04: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
898 001424 000001      DQCR05: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 05
899 001426 000001      DQST05: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
900 001430 000001      DQCR06: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 06
901 001432 000001      DQST06: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
902 001434 000001      DQCR07: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 07
903 001436 000001      DQST07: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
904 001440 000001      DQCR10: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 10
905 001442 000001      DQST10: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
906 001444 000001      DQCR11: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 11
907 001446 000001      DQST11: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
908 001450 000001      DQCR12: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 12
909 001452 000001      DQST12: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
910 001454 000001      DQCR13: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 13
911 001456 000001      DQST13: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
912 001460 000001      DQCR14: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 14
913 001462 000001      DQST14: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
914 001464 000001      DQCR15: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 15
915 001466 000001      DQST15: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
916 001470 000001      DQCR16: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 16
917 001472 000001      DQST16: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
918 001474 000001      DQCR17: .BLKW 1   ; CONTROL STATUS REGISTER FOR DEVICE NO: 17
919 001476 000001      DQST17: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
920 001500 000001      DQACTV: .BLKW 1   ; HOLD ACTIVE BITS FOR TESTING
921 001502 000001      SAVACT: .BLKW 1   ; SAVE NUMBER OF ACTIVE DQ11S
922 001504 000001      DQNUM: .BLKW 1   ; OCTAL NUMBER OF TOTAL NUMBER OF DQ11S
923 001506 000001      DQCSR: .BLKW 1   ; CSR OF DQ11 UNDER TEST
924 001510 000001      DQSTAT: .BLKW 1   ; VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
925
926                      ;PROGRAM INITIALIZATION
927                      ;LOCK OUT INTERRUPTS
928                      ;SET UP PROCESSOR STACK
929                      ;SET UP POWER FAIL VECTOR
930                      ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
931                      ;TYPE TITLE MESSAGE
932

```

```

933 001512 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
934 001520 012706 001200 MOV #STACK,SP ;SET UP STACK
935 001524 012737 016222 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
936 001532 013737 001504 001276 MOV DQNUM,SAVNUM
937 001540 105037 001311 CLR#B STFLG ;CLEAR START FLAG
938 001544 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
939 001550 105037 001312 CLR#B ERRFLG ;CLEAR ERROR FLAG
940 001554 005037 001302 CLR RUNFLG
941 001560 012737 001400 001300 MOV #1400,CREAM
942 001566 005037 001232 CLR ERXCNT ;CLEAR ERROR COUNT
943 001572 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
944 001576 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
945 001604 012737 001512 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
946 ;TESTING STARTS
947 001612 012737 177570 001200 MOV #DSWR,SWR ;MOV HARDWARE SWR TO SWR
948 001620 012737 177570 001202 MOV #DLIGHTS,LIGHTS ;MOV DISPLAY LIGHTS TO LIGHTS
949 001626 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
950 001632 013746 000004 MOV @#4,-(SP)
951 001636 012737 001656 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
952 001644 022777 177777 177326 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
953 001652 001402 BEQ 65$
954 001654 000407 BR 66$
955 001656 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
956 001660 012737 000176 001200 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
957 001666 012737 000174 001202 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
958 001674 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
959 001700 012637 000006 MOV (SP)+,@#6
960 001704 005737 000042 TST @#42 ;UNDER MONITOR
961 001710 001014 BNE 67$
962 ;:*****THE NEXT 4 LINES OF CODE MOVED TO SOLVE PR#2757 (JUNE 78)*****
963 001712 105737 001310 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED?
964 001716 001035 BNE 12$ ;IF YES, BR
965 001720 104402 001000 TYPE ,MTITLE ;TYPE TITLE MESSAGE
966 001724 105137 001310 COMB INIFLG ;IF NOT SET FLAG AND INIT
967 001730 022737 000176 001200 CMP #SWREG,SWR ;IS SWREG USED
968 001736 001001 BNE 67$
969 001740 104415 CNTLU
970 001742 105777 177232 67$: TSTB @SWR
971 001746 100402 BMI .+6
972 001750 004737 000220 JSR PC,CSRMAP
973 001754 104402 017050 TYPE ,XHEAD
974 001760 012737 001400 001244 MOV #1400,TEMP1
975 001766 017737 177252 001246 MOV @TEMP1,TEMP2
976 001774 001406 BEQ .+16
977 001776 104410 CONVRT
978 002000 017076 XSTATQ
979 002002 062737 000002 001244 ADD #2,TEMP1
980 002010 000766 BR .-22
981 002012 032777 000001 177160 12$: BIT #SW00,@SWR
982 002020 001424 BEQ 1$
983 002022 104402 TYPE
984 002024 016771 MNEW
985 002026 005000 CLR R0
986 002030 000000 HALT
987 002032 104414 CKSWR
988 002034 027737 177140 001502 CMP @SWR,SAVACT

```

```

989 002042 101404 BLOS 11$
990 002044 104402 TYPE
991 002046 016632 MERR3
992 002050 000000 HALT
993 002052 000776 BR -2
994 002054 017737 177120 001500 11$: MOV @SWR,DQACTV
995 002062 013700 001500 MOV DQACTV,R0
996 002066 000000 HALT
997 002070 104414 CKSWR
998 002072 012700 000300 1$: MOV #300,R0
999 002076 012701 000302 MOV #302,R1
1000 002102 010120 2$: MOV R1,(R0)+
1001 002104 005021 CLR (R1)+
1002 002106 022021 CMP (R0)+,(R1)+
1003 002110 022700 001000 CMP #1000,R0
1004 002114 001372 BNE 2$
1005
1006 ;TEST START AND RESTART
1007
1008 C02116 012737 000340 177776 .BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
1009 002124 012706 001200 MOV #STACK,SP ;SET UP STACK
1010 002130 005737 000042 TST @#42 ;IS PROGRAM UNDER MONITOR CONTROL
1011 002134 001040 BNE 3$
1012 002136 104414 CKSWR ;CHECK FOR <^G>
1013 002140 032777 000004 177032 BIT #RIT2,@SWR ;CHECK FOR LOCK ON TEST
1014 002146 001411 BEQ 1$
1015 002150 104402 016670 TYPE ,MLOCK
1016 002154 012737 000240 014504 MOV #NOP,TTST
1017 002162 012737 000240 014506 MOV #NOP,TTST+2 ;SET UP TO LOCK
1018 002170 000406 BR 2$
1019 002172 013737 014602 014504 1$: MOV BRW,TTST
1020 002200 013737 014604 014506 MOV BRX,TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1021 002206 032777 000002 176764 2$: BIT #SW01,@SWR ;IF SW01=1, GET STARTING PC
1022 002214 001410 BEQ 3$
1023 002216 104403 INSTR
1024 002220 016656 MTSTPC
1025 002222 104405 PARAM
1026 002224 002254 TST1
1027 002226 012040 TLAST
1028 002230 001214 #RETURN
1029 002232 001 .BYTE 1
1030 002233 001 .BYTE 1
1031 002234 000403 BR 4$
1032 002236 012737 002254 001214 3$: MOV #TST1,RETURN ;START AT TEST 1
1033 002244 104402 016560 4$: TYPE ,MR ;TYPE R
1034 002250 000177 176740 JMP @RETURN ;START TESTING
1035 ; TEST 1
1036 *****
1037 002254 012737 000001 001226 TST1: MOV #1,TSTNO
1038 002262 012737 002644 001214 MOV #TST2,RETURN
1039 002270 012737 002644 001216 MOV #TST2,NEXT
1040 002276 105737 001302 TSTB RUNFLG ;IS THIS MY FIRST TIME HERE?
1041 002302 001010 BNE 1$ ;BR IF FLAG IS SET
1042 002304 012737 000001 001304 MOV #BIT0,RUN ;SET RUN POINTER.
1043 002312 012737 000020 001306 MOV #16.,RUNCNT ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1044 002320 105137 001302 COMB RUNFLG ;SET RUN FLAG
  
```

```

1045 002324 033737 001304 001500 1$: BIT RUN,DQACTV ;FIND AN ACTIVE DQ11 TO TEST.
1046 002332 001032 BNE 3$ ;BR IF I FOUND ONE TO TEST.
1047 002334 005737 001500 TST DQACTV ;FIND OUT IF THERE ARE NO DQ11 ACTIVE.
1048 002340 001423 BEQ 2$ ;BR TO FATAL ERROR. WHY AM I HERE IF NO ACTIVE DQ11'S???
1049 002342 000257 CCC ;CLEAR ALL THE CONDITION CODES OF CPU
1050 002344 006137 001304 ROL RUN ;UPDATE RUN POINTER
1051 002350 062737 000004 001300 ADD #4,CREAM ;UPDATE ADDRESS POINTER.
1052 002356 005337 001306 DEC RUNCNT ;DEC NUMBER OF TIMES I LOOKED AT ACTIVE.
1053 002362 001360 BNE 1$ ;BR AND KEEP LOOKING.
1054 002364 012737 000020 001306 MOV #16,RUNCNT ;START RESTORING MY POINTERS.
1055 002372 012737 001400 001300 MOV #1400,CREAM ;RESTORE ADDRESS POINTER
1056 002400 012737 000001 001304 MOV #1,RUN ;RESTORE RUN POINTER.
1057 002406 0007.6 BR 1$ ;KEEP ON TESTING.
1058 002410 104402 2$: TYPE ;ALLERT OPERATOR OF FATAL ERROR
1059 002412 016563 MERR2 ;NO DQ11 ACTIVE. WHY AM I HERE???
1060 002414 000000 HALT ;YOU MUST RELOAD DQ11 DIAGNOSTIC!.
1061 002416 000776 BR .-2 ;STICK HERE ON CONT.
1062 002420 000257 3$: CCC ;CLEAR CPU COND. CODES
1063 002422 006137 001304 ROL RUN ;UPDATE RUN. ACTIVE DQ11 FOUND.
1064 002426 017737 176646 001506 MOV @CREAM,DQCSR ;PLACE ADDRESS OF DQ11 AT DQCSR
1065 002434 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1066 002442 017737 176632 001510 MOV @CREAM,DQSTAT ;PLACE STATUS OF DQ11 AT DQSTAT
1067 002450 062737 000002 001300 ADD #2,CREAM ;UPDATE ADDRESS POINTER
1068 002456 013737 001506 001360 MOV DQCSR,DQRCSR
1069 002464 013737 001510 001350 MOV DQSTAT,DQRVEC
1070 002472 042737 177007 001350 BIC #177007,DQRVEC
1071 002500 013737 001350 001352 MOV DQRVEC,DQRLVL ;GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
1072 002506 062737 000002 001352 ADD #2,DQRLVL
1073 002514 013737 001352 001354 MOV DQRLVL,DQTVEC ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
1074 002522 062737 000002 001354 ADD #2,DQTVEC
1075 002530 013737 001354 001356 MOV DQTVEC,DQTLVL ;GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
1076 002536 062737 000002 001356 ADD #2,DQTLVL
1077 002544 013737 001360 001362 MOV DQRCSR,DQRCSH
1078 002552 005237 001362 INC DQRCSH ;GENERATE ADDRESS OF HIGH BYTE
1079 002556 013737 001360 001364 MOV DQRCSR,DQTCR ;GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
1080 002564 062737 000002 001364 ADD #2,DQTCR
1081 002572 013737 001364 001366 MOV DQTCR,DQERR ;GENERATE ADDRESS OF ERROR REGISTER
1082 002600 062737 000002 001366 ADD #2,DQERR
1083 002606 013737 001366 001370 MOV DQERR,DQREG ;GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
1084 002614 005237 001370 INC DQREG
1085 002620 013737 001370 001372 MOV DQREG,DQSEC ;GENERATE ADDRESS OF SECONDARY REGISTER
1086 002626 005237 001372 INC DQSEC
1087 002632 013737 001372 001374 MOV DQSEC,DQSECH ;GENERATE ADDRESS OF HIGH BYTE
1088 002640 005237 001374 INC DQSECH

```

```

1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100

```

```

;DQ11 HELL RAISER!!!
;THIS TEST WILL EXERCISE:
;DQ11 RECEIVER AND TRANSMITTER INTERUPTS
;ENTER T AND EXIT T (IF AB OPTION INSTALLED)
;VRC
;THE CABLE AND TURN AROUND (DATA ONLY)
;CHARACTER TRANSFERS.

```



```

1101      : TEST 2
1102      :*****
1103 002644 012737 000002 001226 TST2: MOV #2,TSTNO
1104 002652 012737 002734 001214 MOV #2$,RETURN
1105 002660 012737 000036 001222 MOV #30,ICOUNT
1106 002666 012737 003756 001216 MOV #TST3,NEXT
1107      :ADJUST SYNC CHARACTERS.
1108
1109 002674 032737 100000 001510 BIT #SYNBIT,DQSTAT ;ONE SYNC CHAR OR TWO?
1110 002702 001006 BNE 1$ ;BR IF TWO
1111 002704 112737 000377 013052 MOVB #377,SYNC ;SET ONE SYNC. ;:++D
1112 002712 005037 013656 CLR XSYNC ;DBL SYNC SET TO ONE.
1113 002716 000406 BR 2$ ;CONT.
1114 002720 112737 000026 013052 1$: MOVB #26,SYNC ;LOAD FOR TWO SYNC
1115 002726 012737 013026 013656 MOV #13026,XSYNC ;SAME FOR DBL SYNC
1116 002734 104413 2$: MEMCLR ;CLEAR ALL REGISTERS GIVE MSTCLR
1117 002736 005037 014302 CLR GDCHAR ;ZERO POINTER
1118 002742 005037 014274 CLR CHAR ;
1119 002746 005037 177776 CLR PS ;ZERO PROC. PRIO.
1120 002752 105077 176412 SETON: CLRB @DQREG ;SEL THE RX BA PRI.
1121 002756 012777 013256 176406 MOV #RXBUFF,@DQSEC ;LOAD RX BA PRI.
1122 002764 105277 176400 INCB @DQREG ;SEL RX WC PRI.
1123 002770 012777 177600 176374 MOV #-200,@DQSEC ;SET FOR 200 (8) CHARS
1124 002776 105277 176366 INCB @DQREG ;SEL THE TX BA PRI.
1125 003002 012777 013052 176362 MOV #SYNC,@DQSEC ;LOAD WITH SYNC POINTER
1126 003010 105277 176354 INCB @DQREG ;SEL THE TX WC PRI.
1127 003014 012777 177576 176350 MOV #-202,@DQSEC ;SET FOR 2 SYNC AND 200 (8) CHARS.
1128 003022 105277 176342 INCB @DQREG ;SEL THE RX BA SEC
1129 003026 012777 014064 176336 MOV #XRXBUF,@DQSEC ;LOAD RX BA SEC
1130 003034 105277 176330 INCB @DQREG ;SEL RX WC SEC
1131 003040 012777 177600 176324 MOV #-200,@DQSEC ;SET FOR 200(8) CHARS
1132 003046 105277 176316 INCB @DQREG ;SEL THE TX BA SEC
1133 003052 012777 013662 176312 MOV #XTXBUF,@DQSEC ;LOAD IT
1134 003060 105277 176304 INCB @DQREG ;SEL THE TX WC SEC
1135 003064 012777 177600 176300 MOV #-200,@DQSEC ;SET FOR 200 CHARS
1136 003072 112777 000011 176270 MOVB #11,@DQREG ;SEL THE SYNC REGISTER
1137 003100 013777 013050 176264 MOV .SYNC,@DQSEC ;LOAD SYNC
1138 003106 105277 176256 INCB @DQREG ;SEL THE MISC REGISTER
1139 003112 012777 104000 176252 MOV #104000,@DQSEC ;SET 8 BITS PER CHAR AND VRC ENABLE.
1140 003120 032737 040000 001510 BIT #JUMBIT,DQSTAT ;IS JUMPER AT END OF CABLE?
1141 003126 001003 BNE .+10 ;BR IF YES
1142 003130 052777 000010 176234 BIS #BIT3,@DQSEC ;NO CABLE SET TEST LOOP FOR DATA TURN AROUND
1143 003136 112777 000017 176224 MOVB #17,@DQREG ;SEL THE POLY REGISTER
1144 003144 012777 123456 176220 MOV #123456,@DQSEC ;SET PLOYNOMIAL.
1145
1146 003152 012700 013054 MOV #TXBFA,R0 ;START TO FILL TX BUFFERS
1147 003156 012703 000177 MOV #177,R3 ;COUNTER
1148 003162 110320 1$: MOVB R3,(R0)+ ;PRIMARY IS BINARY COUNT BACKWARDS.
1149 003164 105303 DECB R3 ;DONE?
1150 003166 001375 BNE 1$ ;NO
1151 003170 012700 013662 MOV #XTXBUF,R0 ;SET SEC BUFFER
1152 003174 005003 CLR R3 ;
1153 003176 110320 2$: MOVB R3,(R0)+ ;SECONDARY IS BINARY COUNT
1154 003200 105203 INCB R3 ;DONE?
1155 003202 100375 BPL 2$ ;NO
1156 003204 012777 003502 176136 MOV #RXISR,@DQREVC ;SET RECEIVER INTERRUPT POINTER

```

```

1157 003212 012777 000240 176132      MOV      #240,@DQRLVL      ;SET PRIO: TO 5
1158 003220 012777 003334 176126      MOV      #TXISR,@DQTVEC   ;SET TX VECTOR
1159 003226 012777 000240 176122      MOV      #240,@DQTLVL     ;SET PRIO TO 5
1160 003234 012777 000041 176116      MOV      #BIT5+BIT0,@DQRCSR ;SET RX GO AND IE
1161 003242 012777 000051 176114      MOV      #BIT5+BIT3+BIT0,@DQTCSR ;SET TX GO AND IE AND ERROR IE
1162 003250 005037 001246      CLR      TEMP2           ;SET TIMER
1163 003254 012737 000113 001250      MOV      #75.,TEMP3      ;SET NUMBER OF INTERUPTS WANTED
1164 003262 012737 000020 001252 4$:      MOV      #16.,TEMP4     ;SET FOR 16 REGISTERS
1165 003270 142777 000017 176072      BICB    #17,@DQREG       ;SEL RX BA PRI.
1166 003276 105777 176066      TSTB    @DQREG          ;SIT HERE AND MAKE WAVES
1167 003302 005777 176064      TST     @DQSEC          ;WHILE INTERUPTS OCCUR
1168 003306 105277 176056      INCB    @DQREG          ;*****
1169 003312 005337 001252      DEC     TEMP4           ;*****
1170 003316 001367      BNE     3$             ;SAME
1171 003320 005237 001246      INC     TEMP2          ;UPDATE COUNTER
1172 003324 001356      BNE     4$             ;KEEP GOING
1173 003326 104005      HLT     5              ;RX FAILED TO CONTINUSLY INTERUPT
1174                                     ;*****STRONGLY SUGGEST SW08 1 (GOTO TOP OF TEST OF ERROR
1175 003330 000754      BR      4$             ;KEEP IT GOING.
1176 003332 104400      ENDS2: SCOPE          ;SCOPE THIS TEST.....
1177
1178
1179 003334 017737 176026 014266 TXISR:  MOV      @DQERR,ERR     ;ANY ERRORS
1180 003342 100001      BPL     .+4            ;BR IF NO
1181 003344 104004      HLT     4              ;DQ11 ERROR FLAG IS SET.
1182                                     ;*****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR
1183 003346 032777 000004 176010      BIT     #BIT2,@DQTCSR   ;WHO SHOULD I SERVICE PRI OR SEC?
1184 003354 001425      BEQ     1$             ;BR IF SEC NEFDS SERVICE
1185 003356 112777 000002 176004      MOVB   #2,@DQREG       ;SEL TX BA PRI
1186 003364 042777 000200 175772      BIC     #BIT7,@DQTCSR   ;CLEAR TX PRI DONE.
1187 003372 012777 013054 175772      MOV     #TXBFA,@DQSEC   ;LOAD THE TX BA PRI
1188 003400 105277 175764      INCB   @DQREG          ;SEL THE TX WC PRI.
1189 003404 152777 000120 175756      BISB   #BIT6+BIT4,@DQREG ;SET WRITE EN. AND ENTER T
1190 003412 012777 177600 175752      MOV     #-200,@DQSEC    ;LOAD TX WC PRI.
1191 003420 142777 000017 175742      BICB   #17,@DQREG      ;CLEAR REG POINTER.
1192 003426 000002      RTI                                     ;EXIT STAGE RIGHT
1193 003430 042777 000100 175726 1$:      BIC     #BIT6,@DQTCSR   ;CLEAR TX SEC DONE
1194 003436 112777 000006 175724      MOVB   #6,@DQREG       ;SEL THE TX BA PRI.
1195 003444 012777 013662 175720      MOV     #XTXBUF,@DQSEC  ;LOAD THE TX BA SEC
1196 003452 105277 175712      INCB   @DQREG          ;SEL THE TX WC SEC
1197 003456 152777 000060 175704      BISB   #BIT5+BIT4,@DQREG ;SET WRITE EN. AND EXIT T
1198 003464 012777 177600 175700      MOV     #-200,@DQSEC    ;LOAD THE TX WC SEC
1199 003472 142777 000017 175670 2$:      BICB   #17,@DQREG      ;CLEAR REG POINTER
1200 003500 000002      RTI                                     ;EXIT STAGE LEFT.
1201
1202                                     RXISR:
1203 003502 005037 001246      CLR     TEMP2          ;LET TIMER KNOW THAT RX INTERUPTED
1204 003506 017737 175654 014266      MOV     @DQERR,ERR     ;ANY ERRORS
1205 003514 100001      BPL     .+4            ;BR IF NO
1206 003516 104004      HLT     4              ;DQ11 ERROR FLAG SET!!!!
1207                                     ;*****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR
1208 003520 032777 000004 175632      BIT     #BIT2,@DQRCSR   ;WHO SERVICE PRI OR SEC
1209 003526 001426      BEQ     2$             ;BR IF SEC NEEDS SERVICE
1210 003530 042777 000200 175622      BIC     #BIT7,@DQRCSR   ;CLEAR RX PRI. DONE
1211 003536 105077 175626      CLRB   @DQREG          ;SEL RX BA PRI.
1212 003542 012777 013256 175622      MOV     #RXBUFF,@DQSEC  ;LOAD IT

```

|      |        |        |        |        |       |                   |  |
|------|--------|--------|--------|--------|-------|-------------------|--|
| 1213 | 003550 | 105277 | 175614 |        | INCB  | @DQREG            | :SEL THE RX WC PRI.                                      |
| 1214 | 003554 | 152777 | 000120 | 175606 | BISB  | #BIT6+BIT4,@DQREG | :SET WRITE EN. AND ENTER T                               |
| 1215 | 003562 | 012777 | 177600 | 175602 | MOV   | #-200,@DQSEC      | :LOAD RX WC SEC  |
| 1216 | 003570 | 012701 | 013054 |        | MOV   | #TXBFA,R1         | :PREPARE TO CHECK DATA. SET TX POINTER                   |
| 1217 | 003574 | 012702 | 013256 |        | MOV   | #RXBUF,R2         | :SET RX POINTER  |
| 1218 | 003600 | 0001   | 003656 |        | JMP   | 3\$               | :GO AND CHECK DATA                                       |
| 1219 | 003604 | 042777 | 000100 | 175546 | BIC   | #BIT6,@DQRCR      | :CLEAR RX SEC DONE                                       |
| 1220 | 003612 | 112777 | 000004 | 175550 | MOVB  | #4,@DQREG         | :SEL RX BA SEC   |
| 1221 | 003620 | 012777 | 014064 | 175544 | MOV   | #XRABUF,@DQSEC    | :LOAD IT   |
| 1222 | 003626 | 105277 | 175536 |        | INCB  | @DQREG            | :SEL THE RX WC SEC                                       |
| 1223 | 003632 | 152777 | 000060 | 175530 | BISB  | #BIT5+BIT4,@DQREG | :SET WRITE EN. AND EXIT T                                |
| 1224 | 003640 | 012777 | 177600 | 175524 | MOV   | #-200,@DQSEC      | :WRITE RX WC SEC   |
| 1225 | 003646 | 012701 | 013662 |        | MOV   | #XTXBUF,R1        | :GET TX BUFFER POINTER                                   |
| 1226 | 003652 | 012702 | 014064 |        | MOV   | #XRXBUF,R2        | :GET RX POINTER  |
| 1227 | 003656 | 012700 | 000200 |        | MOV   | #200,R0           | :GET NUMBER OF CHARS                                     |
| 1228 | 003662 | 142711 | 000200 |        | BICB  | #BIT7,(R1)        | :CLEAR VRC   |
| 1229 | 003666 | 142712 | 000200 |        | BICB  | #BIT7,(R2)        | :CLEAR VRC   |
| 1230 | 003672 | 122122 |        |        | CMPB  | (R1)+,(R2)+       | :DATA OK?  |
| 1231 | 003674 | 001414 |        |        | BEQ   | 7\$               | :BR IF YES   |
| 1232 | 003676 | 112777 | 000012 | 175464 | MOVB  | #12,@DQREG        | :SEL MISC REG  |
| 1233 | 003704 | 052777 | 000002 | 175460 | BIS   | #BIT1,@DQSEC      | :STOP THE DQ11 CLOCK.                                    |
| 1234 | 003712 | 114137 | 014302 |        | MOVB  | -(R1),GDCHAR      | :STORE GOOD CHAR   |
| 1235 | 003716 | 114237 | 014274 |        | MOVB  | -(R2),CHAR        | :STORE BAD CHAR.   |
| 1236 | 003722 | 104003 |        |        | HLT   | 3                 | :DATA COMPARE ERROR                                      |
| 1237 |        |        |        |        |       |                   | :*****STRONGLY SUGGEST SW08 1 (GOTO TOP OF TEST OF ERROR |
| 1238 | 003724 | 122122 |        |        | CMPB  | (R1)+,(R2)+       | :POP PCINTERS  |
| 1239 | 003726 | 005300 |        |        | DEC   | R0                | :ALL DATA CHECKED?                                       |
| 1240 | 003730 | 001354 |        |        | BNE   | 4\$               | :BR IF NO  |
| 1241 | 003732 | 005337 | 001250 |        | DEC   | TEMP3             | :ALL INTERRUPTS DONE?                                    |
| 1242 | 003736 | 001003 |        |        | BNE   | 6\$               | :NO KEEP INTERRUPTING                                    |
| 1243 | 003740 | 000005 |        |        | RESET |                   | :STOP THE SHOW CLEAR THE WORLD                           |
| 1244 | 003742 | 012716 | 003332 |        | MOV   | #ENDTS2,(SP)      | :SET FOR END TEST RETURN                                 |
| 1245 | 003746 | 142777 | 000017 | 175414 | BICB  | #17,@DQREG        | :CLEAR REG POINTER                                       |
| 1246 | 003754 | 000002 |        |        | RTI   |                   | :EXIT STAGE MIDDLE                                       |
| 1247 |        |        |        |        |       |                   |  |

```
1248
1249
1250 :TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
1251 :THIS TEST WILL XMIT AND RECV CHARACTERS
1252 :AT 2 BITS/PER/CHAR.
1253 :DATA CHECKING WILL BE PERFORMED!
1254
1255 : TEST 3
1256 :*****
1257 003756 012737 000003 001226 TST3: MOV #3,TSTNO
1258 003764 012737 004324 001216 MOV #TST4,NEXT
1259 003772 104413 MEMCLR :CLEAR ALL THE DQ11
1260 003774 012700 013256 MOV #RXBUFF,R0 :LOAD THE BUFFER POINTER
1261 004000 005001 CLR R1 :SET UP TO CLEAR THE BUFFER
1262 004002 005020 5$: CLR (R0)+ :CLEAR IT
1263 004004 105201 INCB R1 :DONE?
1264 004006 100375 BPL 5$ :BRANCH IF NO
1265 004010 112777 000011 175352 MOVB #11,@DQREG :SELECT THE SYNC REG
1266 004016 013737 013052 001246 MOV SYNC,TEMP2 :LOAD SYNC
1267 004024 012737 177774 012370 MOV #177774,MASK :LOAD THE MASK
1268 004032 143737 012370 001246 BICB MASK,TEMP2 :SET UP A MASK TO GET THE
1269 004040 005737 001510 TST DOSTAT :SINGLE SYNC CHARACTER? :;+D
1270 004044 100003 BPL 10$ :IF YES,BR. :;+D
1271 004046 000241 CLC :CORRECT SYNC CHARACTER
1272 004050 106037 001246 RORB TEMP2 :FOR THIS CHARACTER LENGTH
1273 004054 143737 012370 001247 10$: BICB MASK,TEMP2+1 :MANIPULATE DATA TO :;+D
1274 004062 000241 CLC :COME UP WITH THE
1275 004064 106037 001247 RORB TEMP2+1 :PROPER SYNC CHARACTER
1276 004070 013737 001246 012372 MOV TEMP2,SYNC1 :LOAD THE CHARACTER
1277 004076 013737 001246 012374 MOV TEMP2,SYNC2 :DITTO
1278 004104 013777 001246 175260 MOV TEMP2,@DQSEC :LOAD THE SYNC REGISTER
1279 004112 105277 175252 INCB @DQREG :SEL THE MISC REGISTER
1280 004116 012777 000010 175246 MOV #BIT3,@DQSEC :SET TEST LOOP
1281 004124 012700 000016 MOV #16,R0
1282 004130 000300 SWAB R0 :FLIP THE BYTES
1283 004132 050077 175234 BIS RO,@DQSEC :SET CHARACTER LENGTH
1284 004136 052777 000002 175226 BIS #BIT1,@DQSEC :TURN CLOCK OFF...
1285 004144 042777 000002 175220 BIC #BIT1,@DQSEC :AND ON
1286 004152 105077 175212 CLRB @DQREG :SEL RX PRIMARY ADRESS
1287 004156 012777 013256 175206 MOV #RXBUFF,@DQSEC :SET ADRESS
1288 004164 105277 175200 INCB @DQREG :SEL RX PRIMARY CHAR COUNT
1289 004170 012777 177734 175174 MOV #-36,@DQSEC :SET CHAR COUNT
1290 004176 105277 175166 INCB @DQREG :SEL TX PRIMARY ADDRESS
1291 004202 012777 012374 175162 MOV #SYNC2,@DQSEC :LOAD THE SYNC CHAR
1292 004210 105277 175154 INCB @DQREG :SEL TX PRI CHAR COUNT
1293 004214 012777 177732 175150 MOV #-38,@DQSEC :SET CHAR COUNT
1294 004222 005277 175132 INC @DQCSR :SET RX GO
1295 004226 005277 175132 INC @DQCSR :SET TX GO
1296 004232 005005 CLR R5 :START TIMING
1297 004234 105777 175120 1$: TSTB @DQCSR :IS DONE UP?
1298 004240 100404 BMI 2$ :BRANCH IF YES
1299 004242 062705 000001 ADD #1,R5 :WAIT
1300 004246 001372 BNE 1$ :BR IF MORE TO GO
1301 004250 104001 HLT 1 :ERROR--NO RX DONE
1302 004252 012700 012376 2$: MOV #TXBUFF,R0 :LOAD BUFFER POINTER
1303 004256 012701 013256 MOV #RXBUFF,R1 :LOAD RX BUFFER POINTER
```

```

1304 004262 012702 000044          MOV    #36.,R2          ;SET UP TO COUNT CHARACTERS
1305 004266          3$:      MOVVB  (R0)+,R5          ;GET A CHARACTER TO COPMARE
1306 004266          CLR    TEMP2           ;
1307 004270          112005  MOVVB  (R1)+,TEMP2       ;GET REC CHARACTER
1308 004274          112137 001246  MOV    TEMP2,R4        ;MOVE TO R4
1309 004300          013704 001246  BIC    MASK,R5         ;MASK OUT UNWANTED BITS
1310 004304          043705 012370  CMP    R5,R4          ;DO THE CHARACTERS MATCH?
1311 004310          020504          BEQ    4$              ;BR IF OK
1312 004312          001401          HLT    2              ;ERROR--DATA DOESN'T MATCH
1313 004314          104002          4$:      DEC    R2          ;ALL DONE?
1314 004316          005302          BNE    3$            ;NO--GO BACK FOR MORE
1315 004320          001362          SCOPE  3$            ;SCOPE THIS TEST
1316 004322          104400

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 3 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED.

```

: TEST 4

```

1324 : *****
1325 : *****
1326 004324 012737 000004 001226 TST4: MOV    #4,TSTNO
1327 004332 012737 004672 001216      MOV    #TST5,NEXT
1328 004340 104413          MEMCLR          ;CLEAR ALL THE DQ11
1329 004342 012700 013256          MOV    #RXBUFF,R0  ;LOAD THE BUFFER POINTER
1330 004346 005001          CLR    R1         ;SET UP TO CLEAR THE BUFFER
1331 004350 005020          5$:      CLR    (R0)+      ;CLEAR IT
1332 004352 105201          INCB   R1         ;DONE?
1333 004354 100375          BPL    5$        ;BRANCH IF NO
1334 004356 112777 000011 175004  MOVVB  #11,@DQREG  ;SELECT THE SYNC REG
1335 004364 013737 013052 001246  MOV    SYNC,TEMP2  ;LOAD SYNC
1336 004372 012737 177770 012370  MOV    #177770,MASK ;LOAD THE MASK
1337 004400 143737 012370 001246  BICB  MASK,TEMP2   ;SET UP A MASK TO GET THE
1338 004406 005737 001510          TST    DQSTAT     ;SINGLE SYNC CHARACTER?
1339 004412 100003          BPL    10$       ;IF YES,BR.
1340 004414 000241          CLC          ;CORRECT SYNC CHARACTER
1341 004416 106037 001246          RORB   TEMP2     ;FOR THIS CHARACTER LENGTH
1342 004422 143737 012370 001247 10$:  BICB  MASK,TEMP2+1 ;MANIPULATE DATA TO
1343 004430 000241          CLC          ;COME UP WITH THE
1344 004432 106037 001247          RORB   TEMP2+1  ;PROPER SYNC CHARACTER
1345 004436 013737 001246 012372  MOV    TEMP2,SYNC1 ;LOAD THE CHARACTER
1346 004444 013737 001246 012374  MOV    TEMP2,SYNC2 ;DITTO
1347 004452 013777 001246 174712  MOV    TEMP2,@DQSEC ;LOAD THE SYNC REGISTER
1348 004460 105277 174704          INCB  @DQREG     ;SEL THE MISC REGISTER
1349 004464 012777 000010 174700  MOV    #BIT3,@DQSEC ;SET TEST LOOP
1350 004472 012700 000015          MOV    #15,R0
1351 004476 000300          SWAB  R0         ;FLIP THE BYTES
1352 004500 050077 174666          BIS   R0,@DQSEC  ;SET CHARACTER LENGTH
1353 004504 052777 000002 174660  BIS   #BIT1,@DQSEC ;TURN CLOCK OFF...
1354 004512 042777 000002 174652  BIC   #BIT1,@DQSEC ;AND ON
1355 004520 105077 174644          CLRB  @DQREG     ;SEL RX PRIMARY ADRESS
1356 004524 012777 013256 174640  MOV    #RXBUFF,@DQSEC ;SET ADRESS
1357 004532 105277 174632          INCB  @DQREG     ;SEL RX PRIMARY CHAR COUNT
1358 004536 012777 177734 174626  MOV    #-36.,@DQSEC ;SET CHAR COUNT
1359 004544 105277 174620          INCB  @DQREG     ;SEL TX PRIMARY ADDRESS

```

PROGRAM INITIALIZATION AND START UP.

```

1360 004550 012777 012374 174614      MOV      #SYNC2,@DQSEC      ;LOAD THE SYNC CHAR
1361 004556 105277 174606      INCB     @DQREG             ;SEL TX PRI CHAR COUNT
1362 004562 012777 177732 174602      MOV      #-38,@DQSEC       ;SET CHAR COUNT
1363 004570 005277 174564      INC      @DQRCSR           ;SET RX GO
1364 004574 005277 174564      INC      @DQTCR           ;SET TX GO
1365 004600 005005      CLR      R5                ;START TIMING
1366 004602 105777 174552      1$:     TSTB     @DQRCSR         ;IS DONE UP?
1367 004606 100404      BMI     2$                 ;BRANCH IF YES
1368 004610 062705 000001      ADD      #1,R5             ;WAIT
1369 004614 001372      BNE     1$                 ;BR IF MORE TO GO
1370 004616 104001      HLT     1                  ;ERROR--NO RX DONE
1371 004620 012700 012376      2$:     MOV      #TXBUFF,R0    ;LOAD BUFFER POINTER
1372 004624 012701 013256      MOV      #RXBUFF,R1       ;LOAD RX BUFFER POINTER
1373 004630 012702 000044      MOV      #36.,R2          ;SET UP TO COUNT CHARACTERS
1374 004634      3$:
1375 004634 112005      MOVB    (R0)+,R5           ;GET A CHARACTER TO COPMARE
1376 004636 005037 001246      CLR     TEMP2              ;
1377 004642 112137 001246      MOVB    (R1)+,TEMP2        ;GET REC CHARACTER
1378 004646 013704 001246      MOV     TEMP2,R4           ;MOVE TO R4
1379 004652 043705 012370      BIC     MASK,R5            ;MASK OUT UNWANTED BITS
1380 004656 020504      CMP     R5,R4              ;DO THE CHARACTERS MATCH?
1381 004660 001401      BEQ     4$                 ;BR IF OK
1382 004662 104002      HLT     2                  ;ERROR--DATA DOESN'T MATCH
1383 004664 005302      4$:     DEC      R2            ;ALL DONE?
1384 004666 001362      BNE     3$                 ;NO--GO BACK FOR MORE
1385 004670 104400      SCOPE
1386
1387
1388      ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
1389      ;THIS TEST WILL XMIT AND RECV CHARACTERS
1390      ;AT 4 BITS/PER/CHAR.
1391      ;DATA CHECKING WILL BE PERFORMED.
1392
1393      ; TEST 5
1394      ;*****
1395 004672 012737 000005 001226      TST5:  MOV      #5,TSTNO
1396 004700 012737 005240 001216      MOV      #TST6,NEXT
1397 004706 104413      MEMCLR
1398 004710 012700 013256      MOV      #RXBUFF,R0       ;CLEAR ALL THE DQ11
1399 004714 005001      CLR      R1                ;LOAD THE BUFFER POINTER
1400 004716 005020      5$:     CLR      (R0)+            ;SET UP TO CLEAR THE BUFFER
1401 004720 105201      INCB     R1                ;CLEAR IT
1402 004722 100375      BPL     5$                 ;DONE?
1403 004724 112777 000011 174436      MOVB    #11,@DQREG        ;BRANCH IF NO
1404 004732 013737 013052 001246      MOV     SYNC,TEMP2         ;SELECT THE SYNC REG
1405 004740 012737 177760 012370      MOV     #177760,MASK      ;LOAD SYNC
1406 004746 143737 012370 001246      BICB    MASK,TEMP2        ;LOAD THE MASK
1407 004754 005737 001510      TST     DQSTAT             ;SET UP A MASK TO GET THE
1408 004760 100003      BPL     10$                ;SINGLE SYNC CHARACTER?
1409 004762 000241      CLC
1410 004764 106037 001246      RORB    TEMP2              ;IF YES,BR.
1411 004770 143737 012370 001247 10$:  BICB    MASK,TEMP2+1      ;CORRECT SYNC CHARACTER
1412 004776 000241      CLC
1413 005000 106037 001247      RORB    TEMP2+1           ;FOR THIS CHARACTER LENGTH
1414 005004 013737 001246 012372      MOV     TEMP2,SYNC1        ;MANIPULATE DATA TO
1415 005012 013737 001246 012374      MOV     TEMP2,SYNC2        ;COME UP WITH THE
                                ;PROPER SYNC CHARACTER
                                ;LOAD THE CHARACTER
                                ;DITTO

```



```

1416 005020 013777 001246 174344      MOV     TEMP2,@DQSEC      ;LOAD THE SYNC REGISTER
1417 005026 105277 174336                INCB   @DQREG            ;SEL THE MISC REGISTER
1418 005032 012777 000010 174332      MOV     #BIT3,@DQSEC     ;SET TEST LOOP
1419 005040 012700 000014                MOV     #14,R0           ;
1420 005044 000300                SWAB   R0                ;FLIP THE BYTES
1421 005046 050077 174320                BIS    RO,@DQSEC        ;SET CHARACTER LENGTH
1422 005052 052777 000002 174312      BIS    #BIT1,@DQSEC     ;TURN CLOCK OFF...
1423 005060 042777 000002 174304      BIC    #BIT1,@DQSEC     ;AND ON
1424 005066 105077 174276                CLR   @DQREG            ;SEL RX PRIMARY ADDRESS
1425 005072 012777 013256 174272      MOV     #RXBUFF,@DQSEC  ;SET ADDRESS
1426 005100 105277 174264                INCB   @DQREG            ;SEL RX PRIMARY CHAR COUNT
1427 005104 012777 177734 174260      MOV     #-36,@DQSEC     ;SET CHAR COUNT
1428 005112 105277 174252                INCB   @DQREG            ;SEL TX PRIMARY ADDRESS
1429 005116 012777 012374 174246      MOV     #SYNC2,@DQSEC   ;LOAD THE SYNC CHAR
1430 005124 105277 174240                INCB   @DQREG            ;SEL TX PRI CHAR COUNT
1431 005130 012777 177732 174234      MOV     #-38,@DQSEC     ;SET CHAR COUNT
1432 005136 005277 174216                INC    @DQRCR           ;SET RX GO
1433 005142 005277 174216                INC    @DQTCR           ;SET TX GO
1434 005146 005005                CLR    R5                ;START TIMING
1435 005150 105777 174204      1$:   TSTB   @DQRCR           ;IS DONE UP?
1436 005154 100404                BMI    2$                ;BRANCH IF YES
1437 005156 062705 000001                ADD    #1,R5             ;WAIT
1438 005162 001372                BNE    1$                ;BR IF MORE TO GO
1439 005164 104001                HLT    1                  ;ERROR--NO RX DONE
1440 005166 012700 012376      2$:   MOV     #TXBUFF,R0     ;LOAD BUFFER POINTER
1441 005172 012701 013256      MOV     #RXBUFF,R1     ;LOAD RX BUFFER POINTER
1442 005176 012702 000044      MOV     #36,R2         ;SET UP TO COUNT CHARACTERS
1443 005202      3$:
1444 005202 112005                MOV   (R0)+,R5          ;GET A CHARACTER TO COMPARE
1445 005204 005037 001246                CLR   TEMP2            ;
1446 005210 112137 001246      MOV   (R1)+,TEMP2     ;GET REC CHARACTER
1447 005214 013704 001246      MOV   TEMP2,R4        ;MOVE TO R4
1448 005220 043705 012370      BIC   MASK,R5         ;MASK OUT UNWANTED BITS
1449 005224 020504                CMP   R5,R4            ;DO THE CHARACTERS MATCH?
1450 005226 001401                BEQ   4$                ;BR IF OK
1451 005230 104002                HLT   2                  ;ERROR--DATA DOESN'T MATCH
1452 005232 005302      4$:   DEC    R2                ;ALL DONE?
1453 005234 001362                BNE   3$                ;NO--GO BACK FOR MORE
1454 005236 104400                SCOPE                    ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 5 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

: TEST 6

\*\*\*\*\*

```

1463      :
1464 005240 012737 000006 001226      TST6: MOV     #6,TSTNO
1465 005246 012737 005606 001216      MOV     #TST7,NEXT
1466 005254 104413                MEMCLR                    ;CLEAR ALL THE DQ11
1467 005256 012700 013256      MOV     #RXBUFF,R0     ;LOAD THE BUFFER POINTER
1468 005262 005001                CLR    R1                ;SET UP TO CLEAR THE BUFFER
1469 005264 005020      5$:   CLR    (R0)+            ;CLEAR IT
1470 005266 105201                INCB   R1                ;DONE?
1471 005270 100375                BPL    5$                ;BRANCH IF NO

```

CZDQH MACY11 30A(1052) 03-DEC-80 08:29 PAGE 31  
 CZDQHE.P11 03-DEC-80 08:27 PROGRAM INITIALIZATION AND START UP.

SEQ 0030

```

1472 005272 112777 000011 174070      MOVB    #11,@DQREG      ;SELECT THE SYNC REG
1473 005300 013737 013052 001246      MOV     SYNC,TEMP2     ;LOAD SYNC'S
1474 005306 012737 177740 012370      MOV     #177740,MASK   ;LOAD THE MASK
1475 005314 143737 012370 001246      BICB   MASK,TEMP2     ;SET UP A MASK TO GET THE
1476 005322 005737 001510                TST     DQSTAT        ;SINGLE SYNC CHARACTER?      ;:++D
1477 005326 100003                BPL     10$           ;IF YES,BR.                  ;:++D
1478 005330 000241                CLC                                     ;CORRECT SYNC CHARACTER
1479 005332 106037 001246      RORB   TEMP2          ;FOR THIS CHARACTER LENGTH
1480 005336 143737 012370 001247 10$:    BICB   MASK,TEMP2+1   ;MANIPULATE DATA TO      ;:++D
1481 005344 000241                CLC                                     ;COME UP WITH THE
1482 005346 106037 001247      RORB   TEMP2+1       ;PROPER SYNC CHARACTER
1483 005352 013737 001246 012372      MOV     TEMP2,SYNC1   ;LOAD THE CHARACTER
1484 005360 013737 001246 012374      MOV     TEMP2,SYNC2   ;DITTO
1485 005366 013777 001246 173776      MOV     TEMP2,@DQSEC  ;LOAD THE SYNC REGISTER
1486 005374 105277 173770      INCB   @DQREG        ;SEL THE MISC REGISTER
1487 005400 012777 000010 173764      MOV     #BIT3,@DQSEC  ;SET TEST LOOP
1488 005406 012700 000013      MOV     #13,R0       ;
1489 005412 000300                SWAB   R0            ;FLIP THE BYTES
1490 005414 050077 173752      BIS    R0,@DQSEC     ;SET CHARACTER LENGTH
1491 005420 052777 000002 173744      BIS    #BIT1,@DQSEC  ;TURN CLOCK OFF...
1492 005426 042777 000002 173736      BIC    #BIT1,@DQSEC  ;AND ON
1493 005434 105077 173730      CLRB   @DQREG        ;SEL RX PRIMARY ADDRESS
1494 005440 012777 013256 173724      MOV     #RXBUFF,@DQSEC ;SET ADDRESS
1495 005446 105277 173716      INCB   @DQREG        ;SEL RX PRIMARY CHAR COUNT
1496 005452 012777 177734 173712      MOV     #-36,@DQSEC   ;SET CHAR COUNT
1497 005460 105277 173704      INCB   @DQREG        ;SEL TX PRIMARY ADDRESS
1498 005464 012777 012374 173700      MOV     #SYNC2,@DQSEC ;LOAD THE SYNC CHAR
1499 005472 105277 173672      INCB   @DQREG        ;SEL TX PRI CHAR COUNT
1500 005476 012777 177732 173666      MOV     #-38,@DQSEC   ;SET CHAR COUNT
1501 005504 005277 173650      INC    @DQRCR        ;SET RX GO
1502 005510 005277 173650      INC    @DQTCR        ;SET TX GO
1503 005514 005005                CLR    R5            ;START TIMING
1504 005516 105777 173636 1$:    TSTB  @DQRCR        ;IS DONE UP?
1505 005522 100404                BMI    2$            ;BRANCH IF YES
1506 005524 062705 000001      ADD    #1,R5         ;WAIT
1507 005530 001372                BNE    1$            ;BR IF MORE TO GO
1508 005532 104001                HLT    1              ;ERROR--NO RX DONE
1509 005534 012700 012376 2$:    MOV    #TXBUFF,R0   ;LOAD BUFFER POINTER
1510 005540 012701 013256      MOV    #RXBUFF,R1   ;LOAD RX BUFFER POINTER
1511 005544 012702 000044      MOV    #36.,R2     ;SET UP TO COUNT CHARACTERS
1512 005550                3$:
1513 005550 112005                MOVB   (R0)+,R5     ;GET A CHARACTER TO COPMARE
1514 005552 005037 001246      CLR    TEMP2        ;
1515 005556 112137 001246      MOVB   (R1)+,TEMP2  ;GET REC CHARACTER
1516 005562 013704 001246      MOV    TEMP2,R4     ;MOVE TO R4
1517 005566 043705 012370      BIC    MASK,R5      ;MASK OUT UNWANTED BITS
1518 005572 020504                CMP    R5,R4        ;DO THE CHARACTERS MATCH?
1519 005574 001401                BEQ    4$            ;BR IF OK
1520 005576 104002                HLT    2              ;ERROR--DATA DOESN'T MATCH
1521 005600 005302 4$:    DEC    R2            ;ALL DONE?
1522 005602 001362                BNE    3$            ;NO--GO BACK FOR MORE
1523 005604 104400                SCOPE
1524
1525
1526
1527
;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS

```

```

1528                                     ;AT 6 BITS/PER/CHAR.
1529                                     ;DATA CHECKING WILL BE PERFORMED.
1530
1531                                     : TEST 7
1532                                     :*****
1533 005606 012737 000007 001226 ST7: MOV #7,TSTNO
1534 005614 012737 006154 001216 MOV #TST10,NEXT
1535 005622 104413 MEMCLR                                     ;CLEAR ALL THE DQ11
1536 005624 012700 013256 MOV #RXBUFF,R0                                     ;LOAD THE BUFFER POINTER
1537 005630 005001 CLR R1                                     ;SET UP TO CLEAR THE BUFFER
1538 005632 005020 5$: CLR (R0)+                                     ;CLEAR IT
1539 005634 105201 INCB R1                                     ;DONE?
1540 005636 100375 BPL 5$                                     ;BRANCH IF NO
1541 005640 112777 000011 173522 MOVB #11,@DQREG                                     ;SELECT THE SYNC REG
1542 005646 013737 013052 001246 MOV SYNC,TEMP2                                     ;LOAD SYNC'S
1543 005654 012737 177700 012370 MOV #177700,MASK                                     ;LOAD THE MASK
1544 005662 143737 012370 001246 BICB MASK,TEMP2                                     ;SET UP A MASK TO GET THE
1545 005670 005737 001510 TST DQSTAT                                     ;SINGLE SYNC CHARACTER?
1546 005674 100003 BPL 10$                                     ;IF YES,BR.
1547 005676 000241 CLC                                     ;CORRECT SYNC CHARACTER
1548 005700 106037 001246 RORB TEMP2                                     ;FOR THIS CHARACTER LENGTH
1549 005704 143737 012370 001247 10$: BICB MASK,TEMP2+1                                     ;MANIPULATE DATA TO
1550 005712 000241 CLC                                     ;COME UP WITH THE
1551 005714 106037 001247 RORB TEMP2+1                                     ;PROPER SYNC CHARACTER
1552 005720 013737 001246 012372 MOV TEMP2,SYNC1                                     ;LOAD THE CHARACTER
1553 005726 013737 001246 012374 MOV TEMP2,SYNC2                                     ;DITTO
1554 005734 013777 001246 173430 MOV TEMP2,@DQSEC                                     ;LOAD THE SYNC REGISTER
1555 005742 105277 173422 INCB @DQREG                                     ;SEL THE MISC REGISTER
1556 005746 012777 000010 173416 MOV #BIT3,@DQSEC                                     ;SET TEST LOOP
1557 005754 012700 000012 MOV #12,R0
1558 005760 000300 SWAB R0                                     ;FLIP THE BYTES
1559 005762 050077 173404 BIS R0,@DQSEC                                     ;SET CHARACTER LENGTH
1560 005766 052777 000002 173376 BIS #BIT1,@DQSEC                                     ;TURN CLOCK OFF...
1561 005774 042777 000002 173370 BIC #BIT1,@DQSEC                                     ;AND ON
1562 006002 105077 173362 CLRB @DQREG                                     ;SEL RX PRIMARY ADDRESS
1563 006006 012777 013256 173356 MOV #RXBUFF,@DQSEC                                     ;SET ADDRESS
1564 006014 105277 173350 INCB @DQREG                                     ;SEL RX PRIMARY CHAR COUNT
1565 006020 012777 177734 173344 MOV #-36,@DQSEC                                     ;SET CHAR COUNT
1566 006026 105277 173336 INCB @DQREG                                     ;SEL TX PRIMARY ADDRESS
1567 006032 012777 012374 173332 MOV #SYNC2,@DQSEC                                     ;LOAD THE SYNC CHAR
1568 006040 105277 173324 INCB @DQREG                                     ;SEL TX PRI CHAR COUNT
1569 006044 012777 177732 173320 MOV #-38,@DQSEC                                     ;SET CHAR COUNT
1570 006052 005277 173302 INC @DQRCR                                     ;SET RX GO
1571 006056 005277 173302 INC @DQTCR                                     ;SET TX GO
1572 006062 005005 CLR R5                                     ;START TIMING
1573 006064 105777 173270 1$: STB @DQRCR                                     ;IS DONE UP?
1574 006070 100404 BMI 2$                                     ;BRANCH IF YES
1575 006072 062705 000001 ADD #1,R5                                     ;WAIT
1576 006076 001372 BNE 1$                                     ;BR IF MORE TO GO
1577 006100 104001 HLT 1                                     ;ERROR--NO RX DONE
1578 006102 012700 012376 2$: MOV #TXBUFF,R0                                     ;LOAD BUFFER POINTER
1579 006106 012701 013256 MOV #RXBUFF,R1                                     ;LOAD RX BUFFER POINTER
1580 006112 012702 000044 MOV #36,R2                                     ;SET UP TO COUNT CHARACTERS
1581 006116 3$:
1582 006116 112005 MOVB (R0)+,R5                                     ;GET A CHARACTER TO COMPARE
1583 006120 005037 001246 CLR TEMP2

```

```

1584 006124 112137 001246      MOV  B   (R1)+,TEMP2      ;GET REC CHARACTER
1585 006130 013704 001246      MOV      TEMP2,R4        ;MOVE TO R4
1586 006134 043705 012370      BIC     MASK,R5          ;MASK OUT UNWANTED BITS
1587 006140 020504          CMP      R5,R4           ;DO THE CHARACTERS MATCH?
1588 006142 001401          BEQ     4$               ;BR IF OK
1589 006144 104002          HLT     2                ;ERROR--DATA DOESN'T MATCH
1590 006146 005302      4$: DEC     R2            ;ALL DONE?
1591 006150 001362      BNE     3$               ;NO--GO BACK FOR MORE
1592 006152 104400          SCOPE                    ;SCOPE THIS TEST
1593
1594
1595                                ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
1596                                ;THIS TEST WILL XMIT AND RECV CHARACTERS
1597                                ;AT 7 BITS/PER/CHAR.
1598                                ;DATA CHECKING WILL BE PERFORMED!
1599
1600                                ; TEST 10
1601                                ;*****
1602 006154 012737 000010 001226  TST10: MOV     #10,TSTNO
1603 006162 012737 006522 001216      MOV     #TST11,NEXT
1604 006170 104413          MEMCLR                    ;CLEAR ALL THE DQ11
1605 006172 012700 013256      MOV     #RXBUFF,R0       ;LOAD THE BUFFER POINTER
1606 006176 005001          CLR     R1                ;SET UP TO CLEAR THE BUFFER
1607 006200 005020      5$: CLR     (R0)+           ;CLEAR IT
1608 006202 105201          INCB   R1                 ;DONE?
1609 006204 100375          BPL     5$                ;BRANCH IF NO
1610 006206 112777 000011 173154  MOV  B   #11,@DQREG       ;SELECT THE SYNC REG
1611 006214 013737 013052 001246  MOV      SYNC,TEMP2       ;LOAD SYNC
1612 006222 012737 177600 012370  MOV     #177600,MASK      ;LOAD THE MASK
1613 006230 143737 012370 001246  BIC  B   MASK,TEMP2       ;SET UP A MASK TO GET THE
1614 006236 005737 001510          TST     DQSTAT            ;SINGLE SYNC CHARACTER?
1615 006242 100003          BPL     10$              ;IF YES,BR.
1616 006244 000241          CLC                    ;CORRECT SYNC CHARACTER
1617 006246 106037 001246      RORB   TEMP2              ;FOR THIS CHARACTER LENGTH
1618 006252 143737 012370 001247 10$: BIC  B   MASK,TEMP2+1     ;MANIPULATE DATA TO
1619 006260 000241          CLC                    ;COME UP WITH THE
1620 006262 106037 001247          RORB   TEMP2+1           ;PROPER SYNC CHARACTER
1621 006266 013737 001246 012372  MOV     TEMP2,SYNC1       ;LOAD THE CHARACTER
1622 006274 013737 001246 012374  MOV     TEMP2,SYNC2       ;DITTO
1623 006302 013777 001246 173062  MOV     TEMP2,@DQSEC      ;LOAD THE SYNC REGISTER
1624 006310 105277 173054          INCB   @DQREG             ;SEL THE MISC REGISTER
1625 006314 012777 000010 173050  MOV     #BIT3,@DQSEC      ;SET TEST LOOP
1626 006322 012700 000011          MOV     #11,R0
1627 006326 000300          SWAB   R0                ;FLIP THE BYTES
1628 006330 050077 173036      BIS     R0,@DQSEC         ;SET CHARACTER LENGTH
1629 006334 052777 000002 173030  BIS     #BIT1,@DQSEC      ;TURN CLOCK OFF...
1630 006342 042777 000002 173022  BIC     #BIT1,@DQSEC      ;AND ON
1631 006350 105077 173014          CLRB   @DQREG            ;SEL RX PRIMARY ADDRESS
1632 006354 012777 013256 173010  MOV     #RXBUFF,@DQSEC    ;SET ADDRESS
1633 006362 105277 173002          INCB   @DQREG            ;SEL RX PRIMARY CHAR COUNT
1634 006366 012777 177734 172776  MOV     #-36,@DQSEC       ;SET CHAR COUNT
1635 006374 105277 172770          INCB   @DQREG            ;SEL TX PRIMARY ADDRESS
1636 006400 012777 012374 172764  MOV     #SYNC2,@DQSEC     ;LOAD THE SYNC CHAR
1637 006406 105277 172756          INCB   @DQREG            ;SEL TX PRI CHAR COUNT
1638 006412 012777 177732 172752  MOV     #-38,@DQSEC       ;SET CHAR COUNT
1639 006420 005277 172734          INC     @DQRCR           ;SET RX GO

```

```

1640 006424 005277 172734      INC      @DQTCR      :SET TX GO
1641 006430 005005              CLR      R5          :START TIMING
1642 006432 105777 172722      1$:     TSTB      @DQRCR      :IS DONE UP?
1643 006436 100404              BMI      2$         :BRANCH IF YES
1644 006440 062705 000001      ADD      #1,R5      :WAIT
1645 006444 001372              BNE      1$         :BR IF MORE TO GO
1646 006446 104001              HLT      1          :ERROR--NO RX DONE
1647 006450 012700 012376      2$:     MOV      #TXBUFF,R0  :LOAD BUFFER POINTER
1648 006454 012701 013256      MOV      #RXBUFF,R1  :LOAD RX BUFFER POINTER
1649 006460 012702 000044      MOV      #36.,R2     :SET UP TO COUNT CHARACTERS
1650 006464
1651 006464 112005              3$:     MOV      (R0)+,R5    :GET A CHARACTER TO COMPARE
1652 006466 005037 001246      CLR      TEMP2       :
1653 006472 112137 001246      MOV      (R1)+,TEMP2  :GET REC CHARACTER
1654 006476 013704 001246      MOV      TEMP2,R4    :MOVE TO R4
1655 006502 043705 012370      BIC      MASK,R5     :MASK OUT UNWANTED BITS
1656 006506 020504              CMP      R5,R4       :DO THE CHARACTERS MATCH?
1657 006510 001401              BEQ      4$         :BR IF OK
1658 006512 104002              HLT      2          :ERROR--DATA DOESN'T MATCH
1659 006514 005302              4$:     DEC      R2          :ALL DONE?
1660 006516 001362              BNE      3$         :NO--GO BACK FOR MORE
1661 006520 104400              SCOPE

```

```

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 8 BITS/PER/CHAR.
:DATA CHECKING WILL BE PERFORMED.

```

: TEST 11

```

*****
TST11: 1671 006522 012737 000011 001226      MOV      #1,TSTNO
1672 006530 012737 007070 001216      MOV      #TST12,NEXT
1673 006536 104413              MEMCLR
1674 006540 012700 013256      MOV      #RXBUFF,R0  :CLEAR ALL THE DQ11
1675 006544 005001              CLR      R1          :LOAD THE BUFFER POINTER
1676 006546 005020              5$:     CLR      (R0)+     :SET UP TO CLEAR THE BUFFER
1677 006550 105201              INCB     R1          :CLEAR IT
1678 006552 100375              BPL      5$         :DONE?
1679 006554 112777 000011 172606      MOV      #11,@DQREG  :BRANCH IF NO
1680 006562 013737 013052 001246      MOV      SYNC,TEMP2  :SELECT THE SYNC REG
1681 006570 012737 177400 012370      MOV      #177400,MASK :LOAD SYNC
1682 006576 143737 012370 001246      BICB     MASK,TEMP2  :LOAD THE MASK
1683 006604 005737 001510              TST      DQSTAT     :SET UP A MASK TO GET THE
1684 006610 100003              BPL      10$        :SINGLE SYNC CHARACTER?
1685 006612 000241              CLC
1686 006614 106037 001246              RORB     TEMP2       :CORRECT SYNC CHARACTER
1687 006620 143737 012370 001247      10$:    BICB     MASK,TEMP2+1 :FOR THIS CHARACTER LENGTH
1688 006626 000241              CLC
1689 006630 106037 001247              RORB     TEMP2+1    :MANIPULATE DATA TO
1690 006634 013737 001246 012372      MOV      TEMP2,SYNC1 :COME UP WITH THE
1691 006642 013737 001246 012374      MOV      TEMP2,SYNC2 :PROPER SYNC CHARACTER
1692 006650 013777 001246 172514      MOV      TEMP2,@DQSEC :LOAD THE CHARACTER
1693 006656 105277 172506      INCB     @DQREG     :DITTO
1694 006662 012777 000010 172502      MOV      #BIT3,@DQSEC :LOAD THE SYNC REGISTER
1695 006670 012700 000010              MOV      #10,R0     :SEL THE MISC REGISTER
:SET TEST LOOP
:

```

```

::+D
::+D
::+D

```

```

1696 006674 000300 SWAB R0 ;FLIP THE BYTES
1697 006676 050077 172470 BIS R0,@DQSEC ;SET CHARACTER LENGTH
1698 006702 052777 000002 172462 BIS #BIT1,@DQSEC ;TURN CLOCK OFF...
1699 006710 042777 000002 *72454 BIC #BIT1,@DQSEC ;AND ON
1700 006716 105077 172446 CLR @DQREG ;SEL RX PRIMARY ADDRESS
1701 006722 012777 013256 172442 MOV #RXBUFF,@DQSEC ;SET ADDRESS
1702 006730 105277 172434 INCB @DQREG ;SEL RX PRIMARY CHAR COUNT
1703 006734 012777 177734 172430 MOV #-36,@DQSEC ;SET CHAR COUNT
1704 006742 105277 172422 INCB @DQREG ;SEL TX PRIMARY ADDRESS
1705 006746 012777 012374 172416 MOV #SYNC2,@DQSEC ;LOAD THE SYNC CHAR
1706 006754 105277 172410 INCB @DQREG ;SEL TX PRI CHAR COUNT
1707 006760 012777 177732 172404 MOV #-38,@DQSEC ;SET CHAR COUNT
1708 006766 005277 172366 INC @DQRCR ;SET RX GO
1709 006772 005277 172366 INC @DQTCR ;SET TX GO
1710 006776 005005 CLR R5 ;START TIMING
1711 007000 105777 172354 1$: TSTB @DQRCR ;IS DONE UP?
1712 007004 100404 BMJ 2$ ;BRANCH IF YES
1713 007006 062705 000001 ADD #1,R5 ;WAIT
1714 007012 001372 BNE 1$ ;BR IF MORE TO GO
1715 007014 104001 HLT 1 ;ERROR--NO RX DONE
1716 007016 012700 012376 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
1717 007022 012701 013256 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
1718 007026 012702 000044 MOV #36,R2 ;SET UP TO COUNT CHARACTERS
1719 007032 3$:
1720 007032 112005 MOVB (R0)+,R5 ;GET A CHARACTER TO COMPARE
1721 007034 005037 00246 CLR TEMP2 ;
1722 007040 112137 001246 MOVB (R1)+,TEMP2 ;GET REC CHARACTER
1723 007044 013704 001246 MOV TEMP2,R4 ;MOVE TO R4
1724 007050 043705 012370 BIC MASK,R5 ;MASK OUT UNWANTED BITS
1725 007054 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
1726 007056 001401 BEQ 4$ ;BR IF OK
1727 007060 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
1728 007062 005302 4$: DEC R2 ;ALL DONE?
1729 007064 001362 BNE 3$ ;NO--GO BACK FOR MORE
1730 007066 104400 SCOPE ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 9 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED.

```

; TEST 12

```

1739 *****
1740 007070 012737 000012 001226 TST12: MOV #12,TSTNO
1741 007076 012737 007420 001216 MOV #TST13,NEXT
1742 007104 104413 MEMCLR ;CLEAR ALL THE DQ11
1743 007106 012700 013256 MOV #RXBUFF,R0 ;LOAD THE BUFFER POINTER
1744 007112 005001 CLR R1 ;SET UP TO CLEAR THE BUFFER
1745 007114 005020 5$: CLR (R0)+ ;CLEAR IT
1746 007116 105201 INCB R1 ;DONE?
1747 007120 100375 BPL 5$ ;BRANCH IF NO
1748 007122 112777 000011 172240 MOVB #11,@DQREG ;SELECT THE SYNC REG
1749 007130 013737 013052 001246 MOV SYNC,TEMP2 ;LOAD SYNC
1750 007136 012737 177000 012370 MOV #177000,MASK ;LOAD THE MASK
1751 007144 043737 012370 001246 BIC MASK,TEMP2 ;SET UP THE MASK FOR THE

```



```

1752 007152 000241          CLC          ;CORRECT SYNC CHARACTER
1753 007154 006037 001246  ROR          TEMP2       ;SHIFT IT
1754 007160 005737 001510  TST          DQSTAT     ;SINGLE SYNC CHARACTER?      ;:++D
1755 007164 100404          BMI          10$        ;IF NO,BR.                  ;:++D
1756 007166 012737 177777 012372  MOV          #-1,SYNC1   ;IF YES, MARK.              ;:++D
1757 007174 000403          BR           20$        ;CONTINUE.                  ;:++D
1758 007176 013737 001246 012372 10$: MOV          TEMP2,SYNC1 ;LOAD THE CHARACTER        ;:++D
1759 007204 013737 001246 012374 20$: MOV          TEMP2,SYNC2 ;DITTO                      ;:++D
1760 007212 013777 001246 172152  MOV          TEMP2,@DQSEC ;LOAD THE SYNC REGISTER
1761 007220 105277 172144  INCB        @DQREG      ;SEL THE MISC REGISTER
1762 007224 012777 000010 172140  MOV          #BIT3,@DQSEC ;SET TEST LOOP
1763 007232 012700 000007  MOV          #7,R0
1764 007236 000300          SWAB        R0         ;FLIP THE BYTES
1765 007240 050077 172126  BIS          R0,@DQSEC   ;SET CHARACTER LENGTH
1766 007244 052777 000002 172120  BIS          #BIT1,@DQSEC ;TURN CLOCK OFF...
1767 007252 042777 000002 172112  BIC          #BIT1,@DQSEC ;AND ON
1768 007260 105077 172104  CLRB        @DQREG      ;SEL RX PRIMARY ADRESS
1769 007264 012777 013256 172100  MOV          #RXBUFF,@DQSEC ;SET ADDRESS
1770 007272 105277 172072  INCB        @DQREG      ;SEL RX PRIMARY CHAR COUNT
1771 007276 012777 177734 172066  MOV          #-36,@DQSEC ;SET CHAR COUNT
1772 007304 105277 172060  INCB        @DQREG      ;SEL TX PRIMARY ADDRESS
1773 007310 012777 012372 172054  MOV          #SYNC1,@DQSEC ;LOAD THE SYNC CHAR
1774 007316 105277 172046  INCB        @DQREG      ;SEL TX PRI CHAR COUNT
1775 007322 012777 177732 172042  MOV          #-38,@DQSEC ;SET CHAR COUNT
1776 007330 005277 172024  INC          @DQRCR      ;SET RX GO
1777 007334 005277 172024  INC          @DQTCR      ;SET TX GO
1778 007340 005005          CLR          R5        ;START TIMING
1779 007342 105777 172012 1$: TSTB        @DQRCR      ;IS DONE UP?
1780 007346 100404          BMI          2$        ;BRANCH IF YES
1781 007350 062705 000001  ADD          #1,R5       ;WAIT
1782 007354 001372          BNE          1$        ;BR IF MORE TO GO
1783 007356 104001          HLT          1         ;ERROR--NO RX DONE
1784 007360 012700 012376 2$: MOV          #TXBUFF,R0 ;LOAD BUFFER POINTER
1785 007364 012701 013256  MOV          #RXBUFF,R1  ;LOAD RX BUFFER POINTER
1786 007370 012702 000044  MOV          #36,R2     ;SET UP TO COUNT CHARACTERS
1787 007374          3$:
1788 007374 012005  MOV          (R0)+,R5    ;GET ANOTHER CHAR
1789 007376 012104  MOV          (R1)+,R4    ;GET A REC CHAR
1790 007400 043705 012370  BIC          MASK,R5     ;MASK OUT UNWANTED BITS
1791 007404 020504  CMP          R5,R4       ;DO THE CHARACTERS MATCH?
1792 007406 001401  BEQ          4$        ;BR IF OK
1793 007410 104002  HLT          2         ;ERROR--DATA DOESN'T MATCH
1794 007412 005302 4$: DEC          R2       ;ALL DONE?
1795 007414 001367  BNE          3$        ;NO--GO BACK FOR MORE
1796 007416 104400  SCOPE
1797
1798
1799          ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
1800          ;THIS TEST WILL XMIT AND RECV CHARACTERS
1801          ;AT 10 BITS/PER/CHAR.
1802          ;DATA CHECKING WILL BE PERFORMED!
1803
1804          ; TEST 13
1805          ;*****
1806 007420 012737 000013 001226  TST13: MOV          #13,ISTNO
1807 007426 012737 007750 001216  MOV          #TST14,NEXT

```



```

1864
1865                                     :TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
1866                                     :THIS TEST WILL XMIT AND RECV CHARACTERS
1867                                     :AT 11 BITS/PER/CHAR.
1868                                     :DATA CHECKING WILL BE PERFORMED!
1869
1870                                     : TEST 14
1871                                     :*****
1872 007750 012737 000014 001226 TST14: MOV #14,TSTNO
1873 007756 012737 010300 001216      MOV #TST15,NEXT
1874 007764 104413      MEMCLR                                     :CLEAR ALL THE DQ11
1875 007766 012700 013256      MOV #RXBUFF,RO      :LOAD THE BUFFER POINTER
1876 007772 005001      CLR R1              :SET UP TO CLEAR THE BUFFER
1877 007774 005020 5$: CLR (R0)+      :CLEAR IT
1878 007776 105201      INCB R1             :DONE?
1879 010000 100375      BPL 5$              :BRANCH IF NO
1880 010002 112777 000011 171360      MOVB #11,@DQREG    :SELECT THE SYNC REG
1881 010010 013737 013052 001246      MOV SYNC,TEMP2     :LOAD SYNCs
1882 010016 012737 174000 012370      MOV #174000,MASK   :LOAD THE MASK
1883 010024 043737 012370 001246      BIC MASK,TEMP2    :SET UP THE MASK FOR THE
1884 010032 000241      CLC                :CORRECT SYNC CHARACTER
1885 010034 006037 001246      ROR TEMP2          :SHIFT IT
1886 010040 005737 001510      TST DQSTAT         :SINGLE SYNC CHARACTER?      :++D
1887 010044 100404      BMI 10$            :IF NO,BR.                  :++D
1888 010046 012737 177777 012372      MOV #-1,SYNC1      :IF YES, MARK.              :++D
1889 010054 000403      BR 20$             :CONTINUE.                  :++D
1890 010056 013737 001246 012372 10$: MOV TEMP2,SYNC1     :LOAD THE CHARACTER         :++D
1891 010064 013737 001246 012374 20$: MOV TEMP2,SYNC2     :DITTO                      :++D
1892 010072 013777 001246 171272      MOV TEMP2,@DQSEC   :LOAD THE SYNC REGISTER
1893 010100 105277 171264      INCB @DQREG        :SEL THE MISC REGISTER
1894 010104 012777 000010 171260      MOV #BIT3,@DQSEC   :SET TEST LOOP
1895 010112 012700 000005      MOV #5,RO          :
1896 010116 000300      SWAB RO            :FLIP THE BYTES
1897 010120 050077 171246      BIS RO,@DQSEC      :SET CHARACTER LENGTH
1898 010124 052777 000002 171240      BIS #BIT1,@DQSEC   :TURN CLOCK OFF...
1899 010132 042777 000002 171232      BIC #BIT1,@DQSEC   :AND ON
1900 010140 105077 171224      CLR @DQREG         :SEL RX PRIMARY ADDRESS
1901 010144 012777 013256 171220      MOV #RXBUFF,@DQSEC :SET ADDRESS
1902 010152 105277 171212      INCB @DQREG        :SEL RX PRIMARY CHAR COUNT
1903 010156 012777 177734 171206      MOV #-36,@DQSEC    :SET CHAR COUNT
1904 010164 105277 171200      INCB @DQREG        :SEL TX PRIMARY ADDRESS
1905 010170 012777 012372 171174      MOV #SYNC1,@DQSEC  :LOAD THE SYNC CHAR
1906 010176 105277 171166      INCB @DQREG        :SEL TX PRI CHAR COUNT
1907 010202 012777 177732 171162      MOV #-38,@DQSEC    :SET CHAR COUNT
1908 010210 005277 171144      INC @DQRCR         :SET RX GO
1909 010214 005277 171144      INC @DQTCR         :SET TX GO
1910 010220 005005      CLR R5             :START TIMING
1911 010222 105777 171132 1$: TSTB @DQRCR        :IS DONE UP?
1912 010226 100404      BMI 2$             :BRANCH IF YES
1913 010230 062705 000001      ADD #1,R5          :WAIT
1914 010234 001372      BNE 1$             :BR IF MORE TO GO
1915 010236 104001      HLT 1              :ERROR--NO RX DONE
1916 010240 012700 012376 2$: MOV #TXBUFF,RO     :LOAD BUFFER POINTER
1917 010244 012701 013256      MOV #RXBJFF,R1     :LOAD RX BUFFER POINTER
1918 010250 012702 000044      MOV #36.,R2        :SET UP TO COUNT CHARACTERS
1919 010254 3$:

```

```

1920 010254 012005          MOV      (R0)+,R5      ;GET ANOTHER CHAR
1921 010256 012104          MOV      (R1)+,R4      ;GET A REC CHAR
1922 010260 043705 012370  BIC      MASK,R5      ;MASK OUT UNWANTED BITS
1923 010264 020504          CMP      R5,R4        ;DO THE CHARACTERS MATCH?
1924 010266 001401          BEQ     4$            ;BR IF OK
1925 010270 104002          HLT     2            ;ERROR--DATA DOESN'T MATCH
1926 010272 005302 4$:    DEC      R2          ;ALL DONE?
1927 010274 001367          BNE     3$            ;NO--GO BACK FOR MORE
1928 010276 104400          SCOPE                    ;SCOPE THIS TEST
1929
1930
1931                                ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
1932                                ;THIS TEST WILL XMIT AND RECV CHARACTERS
1933                                ;AT 12 BITS/PER/CHAR.
1934                                ;DATA CHECKING WILL BE PERFORMED!
1935
1936                                : TEST 15
1937                                :*****
1938 010300 012737 000015 001226 TST15: MOV      #15,TSTNO
1939 C10306 012737 010630 001216    MOV      #TST16,NEXT
1940 010314 104413          MEMCLR                    ;CLEAR ALL THE DQ11
1941 010316 012700 013256    MOV      #RXBUFF,R0      ;LOAD THE BUFFER POINTER
1942 010322 005001          CLR     R1              ;SET UP TO CLEAR THE BUFFER
1943 010324 005020 5$:    CLR     (R0)+          ;CLEAR IT
1944 010326 105201          INCB   R1              ;DONE?
1945 010330 100375          BPL     5$            ;BRANCH IF NO
1946 010332 112777 000011 171030 MOVB    #11,@DQREG      ;SELECT THE SYNC REG
1947 010340 013737 013052 001246 MOV     SYNC,TEMP2      ;LOAD SYNC
1948 010346 012737 170000 012370 MOV     #170000,MASK    ;LOAD THE MASK
1949 010354 043737 012370 001246 BIC     MASK,TEMP2      ;SET UP THE MASK FOR THE
1950 010362 000241          CLC                    ;CORRECT SYNC CHARACTER
1951 010364 006037 001246    ROR     TEMP2          ;SHIFT IT
1952 010370 005737 001510    TST     DQSTAT         ;SINGLE SYNC CHARACTER?
1953 010374 100404          BMI     10$           ;IF NO, BR.
1954 010376 012737 177777 012372 MOV     #-1,SYNC1      ;IF YES, MARK.
1955 010404 000403          BR     20$           ;CONTINUE.
1956 010406 013737 001246 012372 10$:  MOV     TEMP2,SYNC1    ;LOAD THE CHARACTER
1957 010414 013737 001246 012374 20$:  MOV     TEMP2,SYNC2    ;DITTO
1958 010422 013777 001246 170742    MOV     TEMP2,@DQSEC   ;LOAD THE SYNC REGISTER
1959 010430 105277 170734          INCB   @DQREG         ;SEL THE MISC REGISTER
1960 010434 012777 000010 170730    MOV     #BIT3,@DQSEC   ;SET TEST LOOP
1961 010442 012700 000004          MOV     #4,R0
1962 010446 000300          SWAB   R0            ;FLIP THE BYTES
1963 010450 050077 170716          BIS    R0,@DQSEC      ;SET CHARACTER LENGTH
1964 010454 052777 000002 170710    BIS    #BIT1,@DQSEC    ;TURN CLOCK OFF...
1965 010462 042777 000002 170702    BIC    #BIT1,@DQSEC    ;AND ON
1966 010470 105077 170674          CLRB  @DQREG         ;SEL RX PRIMARY ADDRESS
1967 010474 012777 013256 170670    MOV     #RXBUFF,@DQSEC ;SET ADDRESS
1968 010502 105277 170662          INCB  @DQREG         ;SEL RX PRIMARY CHAR COUNT
1969 010506 012777 177734 170656    MOV     #-36,@DQSEC    ;SET CHAR COUNT
1970 010514 105277 170650          INCB  @DQREG         ;SEL TX PRIMARY ADDRESS
1971 010520 012777 012372 170644    MOV     #SYNC1,@DQSEC  ;LOAD THE SYNC CHAR
1972 010526 105277 170636          INCB  @DQREG         ;SEL TX PRI CHAR COUNT
1973 010532 012777 177732 170632    MOV     #-38,@DQSEC    ;SET CHAR COUNT
1974 010540 005277 170614          INC   @DQRCR         ;SET RX GO
1975 010544 005277 170614          INC   @DQTCR         ;SET TX GO

```

```

1976 010550 005005          CLR      R5          ;START TIMING
1977 010552 105777 170602 1$:  TSTB    @DQRCR     ;IS DONE UP?
1978 010556 100404          BMI      2$          ;BRANCH IF YES
1979 010560 062705 000001  ADD     #1,R5        ;WAIT
1980 010564 001372          BNE     1$          ;BR IF MORE TO GO
1981 010566 104001          HLT     1           ;ERROR--NO RX DONE
1982 010570 012700 012376 2$:  MOV     #TXBUFF,RO   ;LOAD BUFFER POINTER
1983 010574 012701 013256  MOV     #RXBUFF,R1   ;LOAD RX BUFFER POINTER
1984 010600 012702 000044  MOV     #36.,R2      ;SET UP TO COUNT CHARACTERS
1985 010604
1986 010604 012005          MOV     (R0)+,R5     ;GET ANOTHER CHAR
1987 010606 012104          MOV     (R1)+,R4     ;GET A REC CHAR
1988 010610 043705 012370  BIC     MASK,R5      ;MASK OUT UNWANTED BITS
1989 010614 020504          CMP     R5,R4        ;DO THE CHARACTERS MATCH?
1990 010616 001401          BEQ     4$          ;BR IF OK
1991 010620 104002          HLT     2           ;ERROR--DATA DOESN'T MATCH
1992 010622 005302 4$:  DEC     R2           ;ALL DONE?
1993 010624 001367          BNE     3$          ;NO--GO BACK FOR MORE
1994 010626 104400          SCOPE
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

```

```

:TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 13 BITS/PER/CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

: TEST 16

:\*\*\*\*\*

```

2004 010630 012737 000016 001226 TST16: MOV     #16,TSTNO
2005 010636 012737 011160 001216  MOV     #TST17,NEXT
2006 010644 104413          MEMCLR
2007 010646 012700 013256  MOV     #RXBUFF,RO   ;CLEAR ALL THE DQ11
2008 010652 005001          CLR     R1           ;LOAD THE BUFFER POINTER
2009 010654 005020 5$:  CLR     (R0)+        ;SET UP TO CLEAR THE BUFFER
2010 010656 105201          INCB   R1           ;CLEAR IT
2011 010660 100375          BPL    5$          ;DONE?
2012 010662 112777 000011 170500  MOVB   #11,@DQREG   ;BRANCH IF NO
2013 010670 013737 013052 001246  MOV     SYNC,TEMP2   ;SELECT THE SYNC REG
2014 010676 012737 160000 012370  MOV     #160000,MASK ;LOAD SYNCs
2015 010704 043737 012370 001246  BIC     MASK,TEMP2   ;LOAD THE MASK
2016 010712 000241          CLC
2017 010714 006037 001246          ROR    TEMP2        ;SET UP THE MASK FOR THE
2018 010720 005737 001510          TST   DQSTAT        ;CORRECT SYNC CHARACTER
2019 010724 100404          BMI   10$          ;SHIFT IT
2020 010726 012737 177777 012372  MOV     #-1,SYNC1    ;SINGLE SYNC CHARACTER?
2021 010734 000403          BR    20$          ;IF NO,BR.
2022 010736 013737 001246 012372 10$:  MOV     TEMP2,SYNC1  ;IF YES, MARK.
2023 010744 013737 001246 012374 20$:  MOV     TEMP2,SYNC2  ;CONTINUE.
2024 010752 013777 001246 170412  MOV     TEMP2,@DQSEC ;LOAD THE CHARACTER
2025 010760 105277 170404          INCB  @DQREG        ;DITTO
2026 010764 012777 000010 170400  MOV     #BIT3,@DQSEC ;LOAD THE SYNC REGISTER
2027 010772 012700 000003          MOV   #3,R0         ;SEL THE MISC REGISTER
2028 010776 000300          SWAB  R0           ;SET TEST LOOP
2029 011000 050077 170366          BIS   RC,@DQSEC     ;FLIP THE BYTES
2030 011004 052777 000002 170360  BIS   #BIT1,@DQSEC  ;SET CHARACTER LENGTH
2031 011012 042777 000002 170352  BIC   #BIT1,@DQSEC  ;TURN CLOCK OFF...

```

```

:++D
:++D
:++D
:++D
:++D
:++D

```

```

2032 011020 105077 170344          CLR B @DQREG          ;SEL RX PRIMARY ADRESS
2033 011024 012777 013256 170340  MOV #RXBUFF,@DQSEC  ;SET ADDRESS
2034 011032 105277 170332          INC B @DQREG          ;SEL RX PRIMARY CHAR COUNT
2035 011036 012777 177734 170326  MOV #-36,@DQSEC      ;SET CHAR COUNT
2036 011044 105277 170320          INC B @DQREG          ;SEL TX PRIMARY ADDRESS
2037 011050 012777 012372 170314  MOV #SYNC1,@DQSEC    ;LOAD THE SYNC CHAR
2038 011056 105277 170306          INC B @DQREG          ;SEL TX PRI CHAR COUNT
2039 011062 012777 177732 170302  MOV #-38,@DQSEC      ;SET CHAR COUNT
2040 011070 005277 170264          INC @DQRCR           ;SET RX GO
2041 011074 005277 170264          INC @DQTCR           ;SET TX GO
2042 011100 005005                   CLR R5                ;START TIMING
2043 011102 105777 170252          1$: TST B @DQRCR        ;IS DONE UP?
2044 011106 100404                   BMI 2$                ;BRANCH IF YES
2045 011110 062705 000001          ADD #1,R5             ;WAIT
2046 011114 001372                   BNE 1$                ;BR IF MORE TO GO
2047 011116 104001                   HLT 1                 ;ERROR--NO RX DONE
2048 011120 012700 012376          2$: MOV #TXBUFF,R0      ;LOAD BUFFER POINTER
2049 011124 012701 013256          MOV #RXBUFF,R1       ;LOAD RX BUFFER POINTER
2050 011130 012702 000044          MOV #36.,R2          ;SET UP TO COUNT CHARACTERS
2051 011134                   3$:
2052 011134 012005                   MOV (R0)+,R5         ;GET ANOTHER CHAR
2053 011136 012104                   MOV (R1)+,R4         ;GET A REC CHAR
2054 011140 043705 012370          BIC MASK,R5          ;MASK OUT UNWANTED BITS
2055 011144 020504                   CMP R5,R4            ;DO THE CHARACTERS MATCH?
2056 011146 001401                   BEQ 4$                ;BR IF OK
2057 011150 104002                   HLT 2                 ;ERROR--DATA DOESN'T MATCH
2058 011152 005302          4$: DEC R2             ;ALL DONE?
2059 011154 001367                   BNE 3$                ;NO--GO BACK FOR MORE
2060 011156 104400                   SCOPE                 ;SCOPE THIS TEST
2061
2062
2063                   ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
2064                   ;THIS TEST WILL XMIT AND RECV CHARACTERS
2065                   ;AT 14 BITS/PER/CHAR.
2066                   ;DATA CHECKING WILL BE PERFORMED.
2067
2068                   ; TEST 17
2069                   ;*****
2070 011160 012737 000017 001226  TST17: MOV #17,TSTNO
2071 011166 012737 011510 001216  MOV #TST20,NEXT
2072 011174 104413                   MEMCLR                ;CLEAR ALL THE DQ11
2073 011176 012700 013256          MOV #RXBUFF,R0       ;LOAD THE BUFFER POINTER
2074 011202 005001                   CLR R1                ;SET UP TO CLEAR THE BUFFER
2075 011204 005020          5$: CLR (R0)+          ;CLEAR IT
2076 011206 105201                   INCB R1                ;DONE?
2077 011210 100375                   BPL 5$                ;BRANCH IF NO
2078 011212 112777 000011 170150  MOV B #11,@DQREG     ;SELECT THE SYNC REG
2079 011220 013737 013052 001246  MOV SYNC,TEMP2        ;LOAD SYNC
2080 011226 012737 140000 012370  MOV #140000,MASK      ;LOAD THE MASK
2081 011234 043737 012370 001246  BIC MASK,TEMP2        ;SET UP THE MASK FOR THE
2082 011242 000241                   CLC                    ;CORRECT SYNC CHARACTER
2083 011244 006037 001246          ROR TEMP2             ;SHIFT IT
2084 011250 005737 001510          TST DQSTAT            ;SINGLE SYNC CHARACTER?
2085 011254 100404                   BMI 10$                ;IF NO.BR.
2086 011256 012737 177777 012372  MOV #-1,SYNC1         ;IF YES, MARK.
2087 011264 000403                   BR 20$                ;CONTINUE.

```



```

2088 011266 013757 001246 012372 10$: MOV TEMP2,SYNC1 ;LOAD THE CHARACTER ;:++D
2089 011274 013737 001246 012374 20$: MOV TEMP2,SYNC2 ;DITTO ;:++D
2090 011302 013777 001246 170062 MOV TEMP2,@DQSEC ;LOAD THE SYNC REGISTER
2091 011310 105277 170054 INCB @DQREG ;SEL THE MISC REGISTER
2092 011314 012777 000010 170050 MOV #BIT3,@DQSEC ;SET TEST LOOP
2093 011322 012700 000002 MOV #2,R0 ;
2094 011326 000300 SWAB R0 ;FLIP THE BYTES
2095 011330 050077 170036 BIS R0,@DQSEC ;SET CHARACTER LENGTH
2096 011334 052777 000002 170030 BIS #BIT1,@DQSEC ;TURN CLOCK OFF...
2097 011342 042777 000002 170022 BIC #BIT1,@DQSEC ;AND ON
2098 011350 105077 170014 CLRB @DQREG ;SEL RX PRIMARY ADDRESS
2099 011354 012777 013256 170010 MOV #RXBUFF,@DQSEC ;SET ADDRESS
2100 011362 105277 170002 INCB @DQREG ;SEL RX PRIMARY CHAR COUNT
2101 011366 012777 177734 167776 MOV #-36.,@DQSEC ;SET CHAR COUNT
2102 011374 105277 167770 INCB @DQREG ;SEL TX PRIMARY ADDRESS
2103 011400 012777 012372 167764 MOV #SYNC1,@DQSEC ;LOAD THE SYNC CHAR
2104 011406 105277 167756 INCB @DQREG ;SEL TX PRI CHAR COUNT
2105 011412 012777 177732 167752 MOV #-38.,@DQSEC ;SET CHAR COUNT
2106 011420 005277 167734 INC @DQRCR ;SET RX GO
2107 011424 005277 167734 INC @DQTCR ;SET TX GO
2108 011430 005005 CLR R5 ;START TIMING
2109 011432 105777 167722 1$: TSTB @DQRCR ;IS DONE UP?
2110 011436 100404 BMI 2$ ;BRANCH IF YES
2111 011440 062705 000001 ADD #1,R5 ;WAIT
2112 011444 001372 BNE 1$ ;BR IF MORE TO GO
2113 011446 104001 HLT 1 ;ERROR--NO RX DONE
2114 011450 012700 012376 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
2115 011454 012701 013256 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
2116 011460 012702 000044 MOV #36.,R2 ;SET UP TO COUNT CHARACTERS
2117 011464 3$:
2118 011464 012005 MOV (R0)+,R5 ;GET ANOTHER CHAR
2119 011466 012104 MOV (R1)+,R4 ;GET A REC CHAR
2120 011470 043705 012370 BIC MASK,R5 ;MASK OUT UNWANTED BITS
2121 011474 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
2122 011476 001401 BEQ 4$ ;BR IF OK
2123 011500 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
2124 011502 005302 4$: DEL R2 ;ALL DONE?
2125 011504 001367 BNE 3$ ;NO--GO BACK FOR MORE
2126 011506 104400 SCOPE ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 15 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

; TEST 20

\*\*\*\*\*

```

2136 011510 012737 000020 001226 TST20: MOV #20,TSTNO
2137 011516 012737 012040 001216 MOV #TST21,NEXT
2138 011524 104413 MEMCLR ;CLEAR ALL THE DQ11
2139 011526 012700 013256 MOV #RXBUFF,R0 ;LOAD THE BUFFER POINTER
2140 011532 005001 CLR R1 ;SET UP TO CLEAR THE BUFFER
2141 011534 005020 5$: CLR (R0)+ ;CLEAR IT
2142 011536 105201 INCB R1 ;DONE?
2143 011540 100375 BPL 5$ ;BRANCH IF NO

```

```

2144 011542 112777 000011 167620      MOV#B #11,@DQREG      ;SELECT THE SYNC REG
2145 011550 013737 013052 001246      MOV SYNC,TEMP2      ;LOAD SYNC
2146 011556 012737 100000 012370      MOV #100000,MASK    ;LOAD THE MASK
2147 011564 043737 012370 001246      BIC MASK,TEMP2      ;SET UP THE MASK FOR THE
2148 011572 000241                CLC                  ;CORRECT SYNC CHARACTER
2149 011574 006037 001246      ROR TEMP2           ;SHIFT IT
2150 011600 005737 001510      TST DQSTAT          ;SINGLE SYNC CHARACTER?      ;:++D
2151 011604 100404                BMI 10$             ;IF NO,BR.                  ;:++D
2152 011606 012737 177777 012372      MOV #-1,SYNC1       ;IF YES, MARK.              ;:++D
2153 011614 000403                BR 20$             ;CONTINUE.                  ;:++D
2154 011616 013737 001246 012372 10$:  MOV TEMP2,SYNC1     ;LOAD THE CHARACTER        ;:++D
2155 011624 013737 001246 012374 20$:  MOV TEMP2,SYNC2     ;DITTO                      ;:++D
2156 011632 013777 001246 167532      MOV TEMP2,@DQSEC    ;LOAD THE SYNC REGISTER
2157 011640 105277 167524      INCB @DQREG         ;SEL THE MISC REGISTER
2158 011644 012777 000010 167520      MOV #BIT3,@DQSEC    ;SET TEST LOOP
2159 011652 012700 000001                MOV #1,R0
2160 011656 000300                SWAB R0            ;FLIP THE BYTES
2161 011660 050077 167506      BIS R0,@DQSEC       ;SET CHARACTER LENGTH
2162 011664 052777 000002 167500      BIS #BIT1,@DQSEC    ;TURN CLOCK OFF...
2163 011672 042777 000002 167472      BIC #BIT1,@DQSEC    ;AND ON
2164 011700 105077 167464      CLRB @DQREG         ;SEL RX PRIMARY ADDRESS
2165 011704 012777 013256 167460      MOV #RXBUFF,@DQSEC ;SET ADDRESS
2166 011712 105277 167452      INCB @DQREG         ;SEL RX PRIMARY CHAR COUNT
2167 011716 012777 177734 167446      MOV #-36,@DQSEC     ;SET CHAR COUNT
2168 011724 105277 167440      INCB @DQREG         ;SEL TX PRIMARY ADDRESS
2169 011730 012777 012372 167434      MOV #SYNC1,@DQSEC   ;LOAD THE SYNC CHAR
2170 011736 105277 167426      INCB @DQREG         ;SEL TX PRI CHAR COUNT
2171 011742 012777 177732 167422      MOV #-38,@DQSEC     ;SET CHAR COUNT
2172 011750 005277 167404      INC @DQRCR          ;SET RX GO
2173 011754 005277 167404      INC @DQTCR          ;SET TX GO
2174 011760 005005                CLR R5             ;START TIMING
2175 011762 105777 167372 1$:  TSTB @DQRCR         ;IS DONE UP?
2176 011766 100404                BMI 2$             ;BRANCH IF YES
2177 011770 062705 000001      ADD #1,R5           ;WAIT
2178 011774 001372                BNE 1$             ;BR IF MORE TO GO
2179 011776 104001                HLT 1              ;ERROR--NO RX DONE
2180 012000 012700 012376 2$:  MOV #TXBUFF,R0     ;LOAD BUFFER POINTER
2181 012004 012701 013256      MOV #RXBUFF,R1     ;LOAD RX BUFFER POINTER
2182 012010 012702 000044      MOV #36.,R2        ;SET UP TO COUNT CHARACTERS
2183 012014                3$:
2184 012014 012005                MOV (R0)+,R5       ;GET ANOTHER CHAR
2185 012016 012104                MOV (R1)+,R4       ;GET A REC CHAR
2186 012020 043705 012370      BIC MASK,R5        ;MASK OUT UNWANTED BITS
2187 012024 020504                CMP R5,R4          ;DO THE CHARACTERS MATCH?
2188 012026 001401                BEQ 4$            ;BR IF OK
2189 012030 104002                HLT 2              ;ERROR--DATA DOESN'T MATCH
2190 012032 005302 4$:  DEC R2             ;ALL DONE?
2191 012034 001367                BNE 3$            ;NO--GO BACK FOR MORE
2192 012036 104400                SCOPE              ;SCOPE THIS TEST
2193
2194
2195                ;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
2196                ;THIS TEST WILL XMIT AND RECV CHARACTERS
2197                ;AT 16 BITS/PER/CHAR.
2198                ;DATA CHECKING WILL BE PERFORMED.
2199

```

```

2200      ; TEST 21
2201      ;*****
2202 012040 012737 000021 001226 TST21: MOV #21,TSTNO
2203 012046 012737 014306 001216      MOV #.EOP,NEXT
2204 012054 104413      MEMCLR      ;CLEAR ALL THE DQ11
2205 012056 012700 013256      MOV #RXBUFF,R0 ;LOAD THE BUFFER POINTER
2206 012062 005001      CLR R1      ;SET UP TO CLEAR THE BUFFER
2207 012064 005020      5$: CLR (R0)+ ;CLEAR IT
2208 012066 105201      INCB R1     ;DONE?
2209 012070 100375      BPL 5$     ;BRANCH IF NO
2210 012072 112777 000011 167270      MOV #11,@DQREG ;SELECT THE SYNC REG
2211 012100 013737 013052 001246      MOV SYNC,TEMP2 ;LOAD SYNC
2212 012106 012737 000000 012370      MOV #000000,MASK ;LOAD THE MASK
2213 012114 043737 012370 001246      BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
2214 012122 000241      CLC      ;CORRECT SYNC CHARACTER
2215 012124 006037 001246      ROR TEMP2  ;SHIFT IT
2216 012130 005737 001510      TST DQSTAT ;SINGLE SYNC CHARACTER?
2217 012134 100404      BMI 10$   ;IF NO,BR.
2218 012136 012737 177777 012372      MOV #-1,SYNC1 ;IF YES, MARK.
2219 012144 000403      BR 20$   ;CONTINUE.
2220 012146 013737 001246 012372      10$: MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
2221 012154 013737 001246 012374      20$: MOV TEMP2,SYNC2 ;DITTO
2222 012162 013777 001246 167202      MOV TEMP2,@DQSEC ;LOAD THE SYNC REGISTER
2223 012170 105277 167174      INCB @DQREG ;SEL THE MISC REGISTER
2224 012174 012777 000010 167170      MOV #BIT3,@DQSEC ;SET TEST LOOP
2225 012202 012700 000000      MOV #0,R0
2226 012206 000300      SWAB R0   ;FLIP THE BYTES
2227 012210 050077 167156      BIS R0,@DQSEC ;SET CHARACTER LENGTH
2228 012214 052777 000002 167150      BIS #BIT1,@DQSEC ;TURN CLOCK OFF...
2229 012222 042777 000002 167142      BIC #BIT1,@DQSEC ;AND ON
2230 012230 105077 167134      CLR @DQREG ;SEL RX PRIMARY ADDRESS
2231 012234 012777 013256 167130      MOV #RXBUFF,@DQSEC ;SET ADDRESS
2232 012242 105277 167122      INCB @DQREG ;SEL RX PRIMARY CHAR COUNT
2233 012246 012777 177734 167116      MOV #-36,@DQSEC ;SET CHAR COUNT
2234 012254 105277 167110      INCB @DQREG ;SEL TX PRIMARY ADDRESS
2235 012260 012777 012372 167104      MOV #SYNC1,@DQSEC ;LOAD THE SYNC CHAR
2236 012266 105277 167076      INCB @DQREG ;SEL TX PRI CHAR COUNT
2237 012272 012777 177732 167072      MOV #-38,@DQSEC ;SET CHAR COUNT
2238 012300 005277 167054      INC @DQRCR ;SET RX GO
2239 012304 005277 167054      INC @DQTCR ;SET TX GO
2240 012310 005005      CLR R5   ;START TIMING
2241 012312 105777 167042      1$: TSTB @DQRCR ;IS DONE UP?
2242 012316 100404      BMI 2$   ;BRANCH IF YES
2243 012320 062705 000001      ADD #1,R5 ;WAIT
2244 012324 001372      BNE 1$   ;BR IF MORE TO GO
2245 012326 104001      HLT 1    ;ERROR--NO RX DONE
2246 012330 012700 012376      2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
2247 012334 012701 013256      MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
2248 012340 012702 000044      MOV #36.,R2  ;SET UP TO COUNT CHARACTERS
2249 012344      3$:
2250 012344 012005      MOV (R0)+,R5 ;GET ANOTHER CHAR
2251 012346 012104      MOV (R1)+,R4 ;GET A REC CHAR
2252 012350 043705 012370      BIC MASK,R5  ;MASK OUT UNWANTED BITS
2253 012354 020504      CMP R5,R4   ;DO THE CHARACTERS MATCH?
2254 012356 001401      BEQ 4$     ;BR IF OK
2255 012360 104002      HLT 2     ;ERROR--DATA DOESN'T MATCH

```

2256 012362 005302  
 2257 012364 001367  
 2258 012366 104400  
 2259  
 2260  
 2261  
 2262 012370 000000  
 2263 012372 026 026  
 2264 012374 026 026  
 2265 012376  
 2266 012376 177777  
 2267 012400 000000  
 2268 012402 125252  
 2269 012404 052525  
 2270 012406 000001  
 2271 012410 000002  
 2272 012412 000004  
 2273 012414 000010  
 2274 012416 000020  
 2275 012420 000040  
 2276 012422 000100  
 2277 012424 000200  
 2278 012426 000400  
 2279 012430 001000  
 2280 012432 002000  
 2281 012434 004000  
 2282 012436 010000  
 2283 012440 020000  
 2284 012442 040000  
 2285 012444 100000  
 2286 012446 077777  
 2287 012450 137777  
 2288 012452 157777  
 2289 012454 167777  
 2290 012456 173777  
 2291 012460 175 77  
 2292 012462 176777  
 2293 012464 177377  
 2294 012466 177577  
 2295 012470 177677  
 2296 012472 177737  
 2297 012474 177757  
 2298 012476 177767  
 2299 012500 177773  
 2300 012502 177775  
 2301 012504 177776  
 2302 012506 000100  
 2303 012706  
 2304 012706 005077 166446  
 2305 012712 005077 166446  
 2306 012716 005077 166444  
 2307 012722 012705 000020  
 2308 012726 152777 000020 166434  
 2309 012734 142777 000140 166426  
 2310 012742 005077 166424  
 2311 012746 105277 166416

4\$: DEC R2 :ALL DONE?  
 BNE 3\$ :NO--GO BACK FOR MORE  
 SCOPE :SCOPE THIS TEST

MASK: 0  
 SYNC1: .BYTE 26,26  
 SYNC2: .BYTE 26,26  
 TXBUFF:  
 ^B<1111111111111111>  
 ^B<0000000000000000>  
 ^B<1010101010101010>  
 ^B<0101010101010101>  
 ^B<0000000000000001>  
 ^B<0000000000000010>  
 ^B<0000000000000100>  
 ^B<0000000000001000>  
 ^B<0000000000100000>  
 ^B<0000000010000000>  
 ^B<0000000100000000>  
 ^B<0000010000000000>  
 ^B<0001000000000000>  
 ^B<0010000000000000>  
 ^B<0100000000000000>  
 ^B<1000000000000000>  
 ^B<0111111111111111>  
 ^B<1011111111111111>  
 ^B<1101111111111111>  
 ^B<1110111111111111>  
 ^B<1111011111111111>  
 ^B<1111101111111111>  
 ^B<1111110111111111>  
 ^B<1111111011111111>  
 ^B<1111111101111111>  
 ^B<1111111110111111>  
 ^B<1111111111011111>  
 ^B<1111111111101111>  
 ^B<1111111111110111>  
 ^B<1111111111111011>  
 ^B<1111111111111101>  
 ^B<1111111111111110>

.BLKW 100  
 .MEMCLR:  
 CLR @DQRCR  
 CLR @DQTCR  
 CLR @DQERR  
 MOV #16, R5  
 BISB #BIT4, @DQREG  
 BICB #140, @DQREG  
 CLR @DQSEC  
 INCB @DQREG

|      |        |        |        |        |          |                           |               |                      |
|------|--------|--------|--------|--------|----------|---------------------------|---------------|----------------------|
| 2312 | 012752 | 005305 |        |        | DEC      | R5                        |               |                      |
| 2313 | 012754 | 001364 |        |        | BNF      | 1\$                       |               |                      |
| 2314 | 012756 | 105077 | 166406 |        | CLRB     | @DQREG                    |               |                      |
| 2315 | 012762 | 105077 | 166374 |        | CLRB     | @DQRCSH                   |               |                      |
| 2316 | 012766 | 012705 | 000020 |        | MOV      | #16.,R5                   |               |                      |
| 2317 | 012772 | 112777 | 000010 | 166370 | 2\$:     | MOVB                      | #10,@DQREG    |                      |
| 2318 | 013000 | 005077 | 166366 |        | CLR      | @DQSEC                    |               |                      |
| 2319 | 013004 | 112777 | 000014 | 166356 |          | MOVB                      | #14,@DQREG    |                      |
| 2320 | 013012 | 005077 | 166354 |        | CLR      | @DQSEC                    |               |                      |
| 2321 | 013016 | 105277 | 166340 |        | INCB     | @DJRCSH                   |               |                      |
| 2322 | 013022 | 005305 |        |        | DEC      | R5                        |               |                      |
| 2323 | 013024 | 001362 |        |        | BNE      | 2\$                       |               |                      |
| 2324 | 013026 | 105077 | 166330 |        | CLRB     | @DQRCSH                   |               |                      |
| 2325 | 013032 |        |        |        | .MSTCLR: |                           |               |                      |
| 2326 | 013032 | 112777 | 000012 | 166330 |          | MOVB                      | #MISC.,@DQREG |                      |
| 2327 | 013040 | 012777 | 000040 | 166324 |          | MOV                       | #BIT5,@DQSEC  |                      |
| 2328 | 013046 | 000002 |        |        |          | RTI                       |               |                      |
| 2329 | 013050 | 026    | 026    |        | .SYNC:   | .BYTE                     | 26,26         |                      |
| 2330 | 013052 | 026    | 026    |        | SYNC:    | .BYTE                     | 26,26         |                      |
| 2331 | 013054 | 000000 |        |        | TXBFA:   | 0                         |               |                      |
| 2332 |        | 013256 |        |        |          | .-.+200                   |               |                      |
| 2333 | 013256 |        |        |        | RXBUF:   |                           |               |                      |
| 2334 | 013256 | 000200 |        |        |          | .BLKW                     | 200           |                      |
| 2335 | 013656 | 026    | 026    |        | XSYNC:   | .BYTE                     | 26,26         |                      |
| 2336 | 013660 | 026    | 026    |        | XSYNC2:  | .BYTE                     | 26,26         |                      |
| 2337 | 013662 | 000000 |        |        | XTXBUF:  | 0                         |               |                      |
| 2338 |        | 014064 |        |        |          | .=.+200                   |               |                      |
| 2339 | 014064 | 000000 |        |        | XRBUF:   | 0                         |               |                      |
| 2340 |        | 014266 |        |        |          | .=.+200                   |               |                      |
| 2341 | 014266 | 000000 |        |        | ERR:     | 0                         |               |                      |
| 2342 | 014270 | 000000 |        |        | POLY:    | 0                         |               |                      |
| 2343 | 014272 | 000000 |        |        | XPOLY:   | 0                         |               |                      |
| 2344 | 014274 | 000000 |        |        | CHAR:    | 0                         |               |                      |
| 2345 | 014276 | 000000 |        |        | COUNT:   | 0                         |               |                      |
| 2346 | 014300 | 000000 |        |        | ADDR:    | 0                         |               |                      |
| 2347 | 014302 | 000000 |        |        | GDCHAR:  | 0                         |               |                      |
| 2348 | 014304 | 000000 |        |        | DETCAR:  | 0                         |               |                      |
| 2349 |        |        |        |        |          |                           |               |                      |
| 2350 |        |        |        |        |          | :END OF PASS              |               |                      |
| 2351 |        |        |        |        |          | :TYPE NAME OF TEST        |               |                      |
| 2352 |        |        |        |        |          | :UPDATE PASS COUNT        |               |                      |
| 2353 |        |        |        |        |          | :CHECK FOR EXIT TO ACT-11 |               |                      |
| 2354 |        |        |        |        |          | :RESTART TEST             |               |                      |
| 2355 |        |        |        |        |          |                           |               |                      |
| 2356 | 014306 | 005037 | 001234 |        | .EOP:    | CLR                       | LS*ERR        | :CLEAR LAST ERROR PC |
| 2357 | 014312 | 005037 | 001312 |        |          | CLR                       | ERRFLG        | :CLEAR ERROR FLAG    |
| 2358 | 014316 | 005237 | 001230 |        |          | INC                       | PASCNT        | :UPDATE PASS COUNT   |
| 2359 | 014322 | 104402 |        |        |          | TYPE                      |               |                      |
| 2360 | 014324 | 016536 |        |        |          | MEPASS                    |               |                      |
| 2361 | 014326 | 104402 |        |        |          | TYPE                      |               |                      |
| 2362 | 014330 | 016717 |        |        |          | MCSRX                     |               |                      |
| 2363 | 014332 | 104411 |        |        |          | CVRT                      |               |                      |
| 2364 | 014334 | 014444 |        |        |          | XCSR                      |               |                      |
| 2365 | 014336 | 104402 |        |        |          | TYPE                      |               |                      |
| 2366 | 014340 | 016725 |        |        |          | MVECX                     |               |                      |
| 2367 | 014342 | 104411 |        |        |          | CVRT                      |               |                      |

|      |        |        |        |        |          |                |                                    |
|------|--------|--------|--------|--------|----------|----------------|------------------------------------|
| 2368 | 014344 | 014452 |        |        | XVEC     |                |                                    |
| 2369 | 014346 | 104402 |        |        | TYPE     |                |                                    |
| 2370 | 014350 | 016733 |        |        | MPASSX   |                |                                    |
| 2371 | 014352 | 104411 |        |        | CNVRT    |                |                                    |
| 2372 | 014354 | 014460 |        |        | XPASS    |                |                                    |
| 2373 | 014356 | 104402 |        |        | TYPE     |                |                                    |
| 2374 | 014360 | 016744 |        |        | MERRX    |                |                                    |
| 2375 | 014362 | 104411 |        |        | CNVRT    |                |                                    |
| 2376 | 014364 | 014466 |        |        | XERR     |                |                                    |
| 2377 | 014366 | 013777 | 001230 | 164606 | MOV      | PASCNT,@LIGHTS | :DISPLAY PASS COUNT                |
| 2378 | 014374 | 005337 | 001276 |        | DEC      | SAVNUM         |                                    |
| 2379 | 014400 | 001013 |        |        | BNE      | RESTR          |                                    |
| 2380 | 014402 | 013737 | 001504 | 001276 | MOV      | DQNUM,SAVNUM   |                                    |
| 2381 | 014410 | 013701 | 000042 |        | MOV      | @42,R1         | :CHECK FOR ACT-11 OR DDP           |
| 2382 | 014414 | 001405 |        |        | BEQ      | RESTR          | :IF NOT, CONTINUE TESTING          |
| 2383 | 014416 | 000005 |        |        | RESET    |                |                                    |
| 2384 | 014420 |        |        |        | LOGICAL: |                |                                    |
| 2385 | 014420 | 004711 |        |        | JSR      | PC,(R1)        |                                    |
| 2386 | 014422 | 000240 |        |        | NOP      |                |                                    |
| 2387 | 014424 | 000240 |        |        | NOP      |                |                                    |
| 2388 | 014426 | 000240 |        |        | NOP      |                |                                    |
| 2389 | 014430 | 104414 |        |        | RESTR:   | CKSWR          |                                    |
| 2390 | 014432 | 012737 | 002254 | 001214 | MOV      | #TST1,RETURN   |                                    |
| 2391 | 014440 | 000137 | 002254 |        | JMP      | TST1           |                                    |
| 2392 | 014444 | 000001 |        |        | XCSR:    | 1              |                                    |
| 2393 | 014446 | 006    | 002    |        | .BYTE    | 6.2            |                                    |
| 2394 | 014450 | 001360 |        |        | DQRCSR   |                |                                    |
| 2395 | 014452 | 000001 |        |        | XVEC:    | 1              |                                    |
| 2396 | 014454 | 003    | 002    |        | .BYTE    | 3.2            |                                    |
| 2397 | 014456 | 001350 |        |        | DQRVEC   |                |                                    |
| 2398 | 014460 | 000001 |        |        | XPASS:   | 1              |                                    |
| 2399 | 014462 | 006    | 002    |        | .BYTE    | 6.2            |                                    |
| 2400 | 014464 | 001230 |        |        | PASCNT   |                |                                    |
| 2401 | 014466 | 000001 |        |        | XERR:    | 1              |                                    |
| 2402 | 014470 | 006    | 002    |        | .BYTE    | 6.2            |                                    |
| 2403 | 014472 | 001232 |        |        | ERRCNT   |                |                                    |
| 2404 |        |        |        |        |          |                |                                    |
| 2405 |        |        |        |        |          |                | :SCOPE LOOP AND INTERATION HANDLER |
| 2406 |        |        |        |        |          |                |                                    |
| 2407 | 014474 | 104414 |        |        | .SCOPE:  | CKSWR          |                                    |
| 2408 | 014476 | 032777 | 040000 | 164474 | BIT      | #BIT14,@SWR    |                                    |
| 2409 | 014504 | 001407 |        |        | TTST:    | BEQ            | 1\$                                |
| 2410 | 014506 | 000432 |        |        | BR       | 3\$            |                                    |
| 2411 | 014510 | 105777 | 164470 |        | TSTB     | @TKCSR         |                                    |
| 2412 | 014514 | 100027 |        |        | BPL      | 3\$            |                                    |
| 2413 | 014516 | 017700 | 164464 |        | MOV      | @TKDDBR,R0     |                                    |
| 2414 | 014522 | 000412 |        |        | BR       | 2\$            |                                    |
| 2415 | 014524 | 032777 | 004000 | 164446 | 1\$:     | BIT            | #SW11,@SWR                         |
| 2416 | 014532 | 001006 |        |        | BNE      | 2\$            |                                    |
| 2417 | 014534 | 005237 | 001224 |        | INC      | LPCNT          |                                    |
| 2418 | 014540 | 023737 | 001224 | 001222 | CMP      | LPCNT,I COUNT  |                                    |
| 2419 | 014546 | 001012 |        |        | BNE      | 3\$            |                                    |
| 2420 | 014550 | 105037 | 001312 |        | 2\$:     | CLRB           | ERRFLG                             |
| 2421 | 014554 | 005037 | 001224 |        | CLR      | LPCNT          |                                    |
| 2422 | 014560 | 012737 | 000017 | 001222 | MOV      | #15,I COUNT    |                                    |
| 2423 | 014566 | 013737 | 001216 | 001214 | MOV      | NEXT,RETURN    |                                    |

```

2424 014574 013716 001214 3$: MOV RETURN,(SP)
2425 014600 000002 RTI
2426 014602 001407 BRW: 1407
2427 014604 000432 BRX: 432
2428
2429 ;CHECK FOR FREEZE ON CURRENT DATA
2430
2431 014606 104414 .SCOPI: CKSWR
2432 014610 032777 001000 164362 BIT #SW09,@SWR
2433 014616 001402 BEQ 1$
2434 014620 013716 001220 MOV LOCK,(SP)
2435 014624 000002 1$: RTI
2436
2437 ;TELETYPE OUTPUT ROUTINE
2438
2439 014626 010546 .TYPE: MOV R5,-(SP)
2440 014630 017605 000002 MOV @2(SP),R5
2441 014634 062766 000002 000002 ADD #2,2(SP)
2442 014642 005737 016316 1$: TST @RDSW
2443 014646 001004 BNE 300$
2444 014650 032777 010000 164322 BIT #SW12,@SWR
2445 014656 001024 BNE 3$
2446 014660 105715 300$: TSTB (R5)
2447 014662 100014 BPL 2$
2448 014664 105777 164320 TSTB @TPCSR
2449 014670 100375 BPL -4
2450 014672 012777 000015 164312 MOV #15,@TPDBR
2451 014700 105777 164304 TSTB @TPCSR
2452 014704 100375 BPL -4
2453 014706 012777 000012 164276 MOV #12,@TPDBR
2454 014714 105777 164270 2$: TSTB @TPCSR
2455 014720 100375 BPL 2$
2456 014722 112577 164264 MOVB (R5)+,@TPDBR
2457 014726 001345 BNE 1$
2458 014730 012605 3$: MOV (SP)+,R5
2459 014732 000002 RTI
2460
2461 ;ASCII STRING INPUT ROUTINE
2462
2463 014734 010346 .INSTR: MOV R3,-(SP)
2464 014736 010446 MOV R4,-(SP)
2465 014740 017637 000004 014756 MOV @4(SP),MSG
2466 014746 062766 000002 000004 ADD #2,4(SP)
2467 014754 104402 .INST1: TYPE
2468 014756 000000 .MSG: 0
2469 014760 012704 017110 MOV #INBUF,R4
2470 014764 012703 000007 MOV #7,R3
2471 014770 105777 164210 1$: TSTB @TKCSR
2472 014774 100375 BPL 1$
2473 014776 117714 164204 MOVB @TKDBR,(R4)
2474 015002 142714 000200 BICB #200,(R4)
2475 015006 121427 000025 CMPB (R4),#25
2476 015012 001003 BNE 200$ ;IS IT <^G>
2477 015014 104402 016476 TYPE,MCRLF
2478 015020 000755 BR .INST1
2479 015022 122427 000015 200$: CMPB (R4)+,#15

```



```

2480 015026 001423      BEQ     INSTR2
2481 015030 117777 164152 164154    MOVB   @TKDBR,@TPDBR
2482 015036 105777 164146    2$:   TSTB   @TPCSR
2483 015042 100375      BPL     2$
2484 015044 005303      DEC     R3
2485 015046 001350      BNE     1$
2486 015050 000402      BR      .INSTG
2487 015052 010346      .INSTE: MOV   R3,-(SP)
2488 015054 010446      .INSTG: MOV   R4,-(SP)
2489 015056 104402      .INSTG: TYPE
2490 015060 016472      MQM
2491 015062 005737 016316    TST    @#RDSW
2492 015066 001402      BEQ     400$
2493 015070 104402 016476    TYPE,MCRLF
2494 015074 000727    400$:  BR     .INST1
2495 015076 012604    INSTR2: MOV  (SP)+,R4
2496 015100 012603      MOV    (SP)+,R3
2497 015102 000002      RTI
2498
2499
2500      ;CONVERT ASCII STRING TO OCTAL
2501 015104 010546      .PARAM: MOV   R5,-(SP)
2502 015106 010446      MOV    R4,-(SP)
2503 015110 016605 000004    MOV    4(SP),R5
2504 015114 012537 015310    MOV    (R5)+,LOLIM
2505 015120 012537 015312    MOV    (R5)+,HILIM
2506 015124 012537 015314    MOV    (R5)+,DEVADR
2507 015130 112537 015316    MOVB   (R5)+,LOBITS
2508 015134 112537 015317    MOVB   (R5)+,ADRCNT
2509 015140 010566 000004    MOV    R5,4(SP)
2510 015144 005005      PARAM1: CLR   R5
2511 015146 012704 017110    MOV    #INBUF,R4
2512 015152 122714 000015    CMPB   #15,(R4)
2513 015156 001420      BEQ    PARERR
2514 015160 121427 000060    1$:   CMPB   (R4),#60
2515 015164 002415      BLT    PARERR
2516 015166 121427 000067    CMPB   (R4),#67
2517 015172 003012      BGT    PARERR
2518 015174 142714 000060    BICB   #60,(R4)
2519 015200 152405      BISB   (R4)+,R5
2520 015202 122714 000015    CMPB   #15,(R4)
2521 015206 001414      BEQ    LIMITS
2522 015210 006305      ASL    R5
2523 015212 006305      ASL    R5
2524 015214 006305      ASL    R5
2525 015216 000760      BR     1$
2526 015220 122714 000015    PARERR: CMPB  #15,(R4)      ;IS FIRST CHARACTER A <CR>
2527 015224 001003      BNE    120$
2528 015226 005737 016316    TST    @#RDSW      ;IS CKSWR ROUTINE BEING USED
2529 015232 001023      BNE    PARTI
2530 015234 104404    120$:  INSTER
2531 015236 000742      BR     PARAM1
2532
2533      ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2534
2535 015240 020537 015312    LIMITS: CMP   R5,HILIM

```

CZDQH MACY11 30A(1052) 03-DEC-80 08:29 PAGE 50  
 CZDQHE.P11 03-DEC-80 08:27 GENERAL UTILITIES (TYPE OUT,ERROR,SCOPE,ETC.)

SEQ 0049

```

2536 015244 101365          BHI      PARERR
2537 015246 020537 015310    CMP      R5,LOLIM
2538 015252 103762          BLO      PARERR
2539 015254 133705 015316    BITB     LOBITS,R5
2540 015260 001357          BNE      PARERR
2541
2542          ;STORE NUMBER AT SPECIFIED ADDRESS
2543
2544 015262 013704 015314    1$:     MOV      DEVADR,R4
2545 015266 010524          MOV      R5,(R4)+
2546 015270 062705 000002    ADD      #2,R5
2547 015274 105337 015317    DECB     ADRCNT
2548 015300 001372          BNE      1$
2549 015302 012604    PARTI:  MOV      (SP)+,R4
2550 015304 012605          MOV      (SP)+,R5
2551 015306 000002          RTI
2552 015310 000000    LOLIM:  0
2553 015312 000000    HILIM:  0
2554 015314 000000    DEVADR: 0
2555 015316 000000    LOBITS: 0
2556          ADRCNT=LOBITS+1
2557
2558          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2559
2560 015320 016637 000004 001274 .SAV05: MOV      4(SP),SAVPC
2561
2562          ;SAVE R0-R5
2563
2564 015326 010537 001270    SV05:  MOV      R5,SAVR5
2565 015332 010437 001266    MOV      R4,SAVR4
2566 015336 010337 001264    MOV      R3,SAVR3
2567 015342 010237 001262    MOV      R2,SAVR2
2568 015346 010137 001260    MOV      R1,SAVR1
2569 015352 010037 001256    MOV      R0,SAVR0
2570 015356 000002          RTI
2571
2572          ;RESTORE R0-R5
2573
2574 015360 013700 001256    .RES05: MOV      SAVR0,R0
2575 015364 013701 001260    MOV      SAVR1,R1
2576 015370 013702 001262    MOV      SAVR2,R2
2577 015374 013703 001264    MOV      SAVR3,R3
2578 015400 013704 001266    MOV      SAVR4,R4
2579 015404 013705 001270    MOV      SAVR5,R5
2580 015410 000002          RTI
2581
2582          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2583
2584 015412 104402    .CONVR: TYPE
2585 015414 016476    MCRLF
2586 015416 010046    .CNVRT: MOV      R0,-(SP)
2587 015420 010146    MOV      R1,-(SP)
2588 015422 010346    MOV      R3,-(SP)
2589 015424 010446    MOV      R4,-(SP)
2590 015426 010546    MOV      R5,-(SP)
2591 015430 017601 000012    MOV      @12(SP),R1

```

|      |        |        |        |        |         |                                       |                              |
|------|--------|--------|--------|--------|---------|---------------------------------------|------------------------------|
| 2592 | 015434 | 013737 | 017152 | 001250 | MOV     | TEMP,TEMP3                            |                              |
| 2593 | 015442 | 062766 | 000002 | 000012 | ADD     | #2,12(SP)                             |                              |
| 2594 | 015450 | 012137 | 015632 |        | MOV     | (R1)+,WRDCNT                          |                              |
| 2595 | 015454 | 112137 | 015634 |        | 1\$:    | MOVB                                  | (R1)+,CHRCNT                 |
| 2596 | 015460 | 112137 | 015635 |        | MOVB    | (R1)+,SPACNT                          |                              |
| 2597 | 015464 | 013137 | 015636 |        | MOV     | @(R1)+,BINWRD                         |                              |
| 2598 | 015470 | 013704 | 015636 |        | 2\$:    | MOV                                   | BINWRD,R4                    |
| 2599 | 015474 | 113705 | 015634 |        | MOVB    | CHRCNT,R5                             |                              |
| 2600 | 015500 | 012700 | 017152 |        | MOV     | #TEMP,R0                              |                              |
| 2601 | 015504 | 010403 |        |        | 3\$:    | MOV                                   | R4,R3                        |
| 2602 | 015506 | 042703 | 177770 |        | BIC     | #177770,R3                            |                              |
| 2603 | 015512 | 062703 | 000060 |        | ADD     | #060,R3                               |                              |
| 2604 | 015516 | 110320 |        |        | MOVB    | R3,(R0)+                              |                              |
| 2605 | 015520 | 000241 |        |        | CLC     |                                       |                              |
| 2606 | 015522 | 006004 |        |        | ROR     | R4                                    |                              |
| 2607 | 015524 | 000241 |        |        | CLC     |                                       |                              |
| 2608 | 015526 | 006004 |        |        | ROR     | R4                                    |                              |
| 2609 | 015530 | 000241 |        |        | CLC     |                                       |                              |
| 2610 | 015532 | 006004 |        |        | ROR     | R4                                    |                              |
| 2611 | 015534 | 005305 |        |        | DEC     | R5                                    |                              |
| 2612 | 015536 | 001362 |        |        | BNE     | 3\$                                   |                              |
| 2613 | 015540 | 012703 | 017214 |        | MOV     | #MDATA,R3                             |                              |
| 2614 | 015544 | 114023 |        |        | 4\$:    | MOVB                                  | -(R0),(R3)+                  |
| 2615 | 015546 | 105337 | 015634 |        | DECB    | CHRCNT                                |                              |
| 2616 | 015552 | 001374 |        |        | BNE     | 4\$                                   |                              |
| 2617 | 015554 | 105737 | 015635 |        | TSTB    | SPACNT                                |                              |
| 2618 | 015560 | 001405 |        |        | BEQ     | 6\$                                   |                              |
| 2619 | 015562 | 112723 | 000040 |        | 5\$:    | MOVB                                  | #040,(R3)+                   |
| 2620 | 015566 | 105337 | 015635 |        | DECB    | SPACNT                                |                              |
| 2621 | 015572 | 001373 |        |        | BNE     | 5\$                                   |                              |
| 2622 | 015574 | 105013 |        |        | 6\$:    | CLRB                                  | (R3)                         |
| 2623 | 015576 | 104402 |        |        | TYPE    |                                       |                              |
| 2624 | 015600 | 017214 |        |        | MDATA   |                                       |                              |
| 2625 | 015602 | 005337 | 015632 |        | DEC     | WRDCNT                                |                              |
| 2626 | 015606 | 001322 |        |        | BNE     | 1\$                                   |                              |
| 2627 | 015610 | 013737 | 001250 | 017152 | MOV     | TEMP3,TEMP                            |                              |
| 2628 | 015616 | 012605 |        |        | MOV     | (SP)+,R5                              |                              |
| 2629 | 015620 | 012604 |        |        | MOV     | (SP)+,R4                              |                              |
| 2630 | 015622 | 012603 |        |        | MOV     | (SP)+,R3                              |                              |
| 2631 | 015624 | 012601 |        |        | MOV     | (SP)+,R1                              |                              |
| 2632 | 015626 | 012600 |        |        | MOV     | (SP)+,R0                              |                              |
| 2633 | 015630 | 000002 |        |        | RTI     |                                       |                              |
| 2634 | 015632 | 000000 |        |        | WRDCNT: | 0                                     |                              |
| 2635 | 015634 | 000000 |        |        | CHRCNT: | 0                                     |                              |
| 2636 |        | 015635 |        |        | SPACNT= | CHRCNT+1                              |                              |
| 2637 | 015636 | 000000 |        |        | BINWRD: | 0                                     |                              |
| 2638 |        |        |        |        |         | :TRAP DISPATCH SERVICE                |                              |
| 2639 |        |        |        |        |         | :ARGUMENT OF TRAP IS EXTRACTED        |                              |
| 2640 |        |        |        |        |         | :AND USED AS OFFSET TO OBTAIN POINTER |                              |
| 2641 |        |        |        |        |         | :TO SELECTED SUBROUTINE               |                              |
| 2642 |        |        |        |        |         |                                       |                              |
| 2643 | 015640 | 011646 |        |        | .TRPSR: | MOV                                   | (SP),-(SP) ;GET PC OF RETURN |
| 2644 | 015642 | 162716 | 000002 |        | SUB     | #2,(SP)                               | :=PC OF TRAP                 |
| 2645 | 015646 | 017616 | 000000 |        | MOV     | @(SP),(SP)                            | :GET TRP                     |
| 2646 | 015652 | 006316 |        |        | TRPOK:  | ASL                                   | (SP) ;MULTIPLY TRAP ARG BY 2 |
| 2647 | 015654 | 042716 | 177001 |        | BIC     | #177001,(SP)                          | :CLEAR UNWANTED BITS         |

```

2648 015660 062716 001314      ADD    #.TRPTAB,(SP)      ;POINTER TO SUBROUTINE ADDRESS
2649 015664 017616 C00000      MOV    @ (SP), (SP)      ;SUBROUTINE ADDRESS
2650 015670 000136      JMP    @ (SP)+           ;GO TO SUBROUTINE
2651
2652                               ;ERROR HANDLER
2653
2654 015672 104414      .HLT:  CKSWR
2655 015674 032777 010000 163276  BIT    #SW12,@SWR
2656 015702 001406      BEQ    XBX
2657 015704 105777 163300      TSTB  @TPCSR
2658 015710 100003      BPL    XBX
2659 015712 112777 000207 163272  MOVB  #207,@TPDBR
2660 015720 032777 020000 163252  XBX:  BIT    #SW13,@SWR
2661 015726 001074      BNE    HALTS
2662 015730 021637 001234      CMP   (SP),LSTERR
2663 015734 001404      BEQ    1$
2664 015736 011637 001234      MOV   (SP),LSTERR
2665 015742 105037 001312      CLRB  ERRFLG
2666 015746 104406      1$:  SAVO5
2667 015750 011605      MOV   (SP),R5
2668 015752 162705 000002      SUB   #2,R5
2669 015756 011504      MOV   (R5),R4
2670 015760 006304      ASL   R4
2671 015762 061504      ADD   (R5),R4
2672 015764 006304      ASL   R4
2673 015766 042704 177001      BIC   #177001,R4
2674 015772 062704 017624      ADD   #.ERRTAB,R4
2675 015776 012437 016070      MOV   (R4)+,ERRMSG
2676 016002 012437 016102      MOV   (R4)+,DATAHD
2677 016006 011437 016114      MOV   (R4),DATABP
2678 016012 105737 001312      TSTB  ERRFLG
2679 016016 001403      BEQ   TYPMSG
2680 016020 005737 016114      TST   DATABP
2681 016024 001027      BNE   TYPDAT
2682 016026 104402      TYPMSG: TYPE
2683 016030 016755      MTSTN
2684 016032 104411      CNVRT
2685 016034 016214      XTSTN
2686 016036 104402      TYPE
2687 016040 017043      MERRPC
2688 016042 104411      CNVRT
2689 016044 016206      ERTABO
2690 016046 104402      TYPE
2691 016050 016476      MCRLF
2692 016052 112737 177777 001312  MOVB  #-1,ERRFLG
2693 016060 005737 016070      TST   ERRMSG
2694 016064 001402      BEQ   WRKO.FM
2695 016066 104402      TYPE
2696 016070 000000      ERRMSG: 0
2697 016072      WRKO.FM:
2698 016072 005737 016102      TST   DATAHD
2699 016076 001402      BEQ   TYPDAT
2700 016100 104402      TYPE
2701 016102 000000      DATAHD: 0
2702 016104 005737 016114      TYPDAT: TST   DATABP
2703 016110 001402      BEQ   RESREG

```

```

2704 016112 104410
2705 016114 000000
2706 016116 104407
2707 016120 005777 163054
2708 016124 100005
2709 016126 010046
2710 016130 016600 000002
2711 016134 000000
2712 016136 012600
2713 016140 104414
2714 016142 005237 001232
2715 016146 032777 000400 163024
2716 016154 001007
2717 016156 032777 002000 163014
2718 016164 001407
2719 016166 013737 001216 001214
2720 016174 012706 001200
2721 016200 000177 163010
2722 016204 000002
2723 016206 000001
2724 016210 006 002
2725 016212 001274
2726 016214 000001
2727 016216 003 002
2728 016220 001226
2729
2730
2731
2732 016222
2733 016222 012737 016234 000024
2734 016230 000000
2735 016232 000777
2736
2737
2738
2739 016234
2740 016234 012737 016222 000024
2741 016242 012706 001200
2742 016246 005037 017152
2743 016252 005237 017152
2744 016256 001375
2745 016260 104402
2746 016262 016500
2747 016264 104411
2748 016266 016310
2749 016270 005037 001312
2750 016274 005037 001234
2751 016300 104412
2752 016302 104413
2753 016304 000177 162704
2754 016310 000001
2755 016312 003 002
2756 016314 001226
2757
2758
2759

```

CONVRT  
DATABP: 0  
RESREG: RES05  
HALTS: TST @SWR  
BPL EXITER  
PUSHRO  
MOV 2(SP),R0  
HALT  
POPPO  
EXITER: CKSWR  
INC ERRCNT  
BIT #SW08,@SWR  
RNE 1\$  
BIT #SW10,@SWR  
BEQ 2\$  
MOV NEXT,RETURN  
1\$: MOV #STACK,SP  
JMP @RETURN  
2\$: RTI  
ERTAB0: 1  
.BYTE 6,2  
SAVPC  
XTSTN: 1  
.BYTE 3,2  
TSTNO  
:ENTER HERE ON POWER FAILURE  
.PFAIL:  
MOV #RESTART,24 ;SET UP FOR POWER UP TRAP  
HALT ;HALT ON POWER DOWN NORMAL  
BR .  
:PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED  
RESTAR:  
MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE  
MOV #STACK,SP  
CLR TEMP  
INC TEMP  
BNE .-4  
TYPE  
MPFAIL  
CNVRT  
PFTAB  
CLR ERRFLG  
CLR LSTERR  
MSTCLR  
MEMCLR  
JMP @RETURN  
PFTAB: 1  
.BYTE 3,2  
TSTNO  
:CHECK SWITCH REGISTER ROUTINE, CHECKS FOR ^G TO ALLOW CHANGING

```

2760                                     :OF LOC.176.
2761                                     :LOCATIONS USED:
2762 016316 000000                       :DSW: .WORD 0
2763
2764
2765 016320 005737 000042                 .CKSWR: TST      @#42
2766 016324 001042                       BNE      OUT
2767 016326 022737 000176 001200         CMP      #SWREG,SWR ;SOFTWARE SWITCH REGISTER PRESENT
2768 016334 001036                       BNE      OUT ;NO, GET OUT
2769 016336 105777 162642                 TSTB    @T:CSR ;YES, WAIT FOR
2770 016342 100033                       BPL     OUT ;READY, GET CHARACTER
2771 016344 017737 162636 014756         MOV     @TKDBR,.MSG ;AND STRIP OFF
2772 016352 042737 177600 014756         BIC    #177600,.MSG ;THE GARBAGE
2773 016360 122737 000007 014756         CMPB   #7,.MSG ;IS IT '<^G>
2774 016366 001021                       BNE     OUT
2775 016370 104402 016446                 TYPE , $CNTG
2776 016374 005137 016316                 .CNTLU: COM    @#RDSW
2777 016400 104402 016452                 TYPE , $MSWR
2778 016404 104411 016440                 CNVRT , SWREGC
2779 016410 104403 016461                 INSTR , $MNEW
2780 016414 104405                       PARAM
2781 016416 000000                       0
2782 016420 177777                       177777
2783 016422 000176                       SWREG
2784 016424 000 001                       .BYTE 0,1
2785 016426 104402 016476                 TYPE , MCRLF
2786 016432 005037 016316                 OUT: CLR      @#RDSW
2787 016436 000002                       RTJ
2788 016440 000001                       SWREGC 1
2789 016442 006 002                       .BYTE 6,2
2790 016444 000176                       SWREG
2791 016446 057377 000107                 $CNTG: .ASCIZ <377>/^G/
2792 016452 051777 051127 020075         $MSWR: .ASCIZ <377>/SWR= /
2793 016460 000
2794 016461 040 047040 053505         $MNEW: .ASCIZ / NEW= /
2795 016466 020075 000
2796 016472
2797 016472 020040 000077                 .EVEN
2798 016476 000377                       MQM: .ASCIZ / ?/
2799 016500 050377 051127 043040         MCRLF: .ASCIZ <377>
2800 016506 044501 042514 027104         MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST /
2801 016514 051040 051505 040524
2802 016522 052122 040440 020124
2803 016530 042524 052123 000040
2804 016536 042777 042116 050040         MEPASS: .ASCIZ <377>/END PASS DZDQH /
2805 016544 051501 020123 055104
2806 016552 050504 020110 000040
2807 016560 051377 000
2808 016563 377 051120 043517         MR .ASCIZ <377>/R/
2809 016570 040522 020115 047111         MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
2810 016576 044504 040503 042524
2811 016604 020123 047516 042040
2812 016612 053105 041511 051505
2813 016620 050040 042522 042523
2814 016626 052116 000056
2815 016632 044777 051516 043125         MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/

```

```

2816 016640 044506 044503 047105
2817 016646 020124 040504 040524
2818 016654 000041
2819 016656 052377 051505 020124 MTSTPC: .ASCIZ <377>/TEST PC-/
2820 016664 041520 000055
2821 016670 046377 041517 020113 MLOCK: .ASCIZ <377>/LOCK ON! SELECTED TEST/
2822 016676 047117 051440 046105
2823 016704 041505 042524 020104
2824 016712 042524 052123 000
2825 016717 103 051123 020072 MCSRX: .ASCIZ /CSR: /
2826 016724 000
2827 016725 126 041505 020072 MVECX: .ASCIZ /VEC: /
2828 016732 000
2829 016733 120 051501 042523 MPASSX: .ASCIZ /PASSES: /
2830 016740 035123 000040
2831 016744 051105 047522 051522 MERRX: .ASCIZ /ERRORS: /
2832 016752 020072 000
2833 016755 377 052377 051505 MTSTN: .ASCIZ <377><377> /TEST NO: /
2834 016762 020124 047516 020072
2835 016770 000
2836 016771 377 042523 020124 MNEW: .ASCIZ <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE./
2837 016776 053523 052111 044103
2838 017004 051040 043505 052040
2839 017012 020117 050504 030461
2840 017020 051447 042040 051505
2841 017026 051111 042105 040440
2842 017034 052103 053111 027105
2843 017042 000
2844 017043 120 035103 000040 MERRPC: .ASCIZ /PC: /
2845 017050 046777 050101 047440 XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>
2846 017056 020106 050504 030461
2847 017064 051440 040524 052524
2848 017072 177523 000
2849 017076
2850 017076 000002 .EVEN
XSTATQ: 2
2851 017100 006 003 .BYTE 6,3
2852 017102 001244 TEMP1
2853 017104 006 002 .BYTE 6,2
2854 017106 001246 TEMP2
2855 .EVEN
2856 ;BUFFERS FOR INPUT-OUTPUT
2857
2858
2859 017110 000000 INBUF: 0
2860 017152 .+.40
2861 017152 000000 TEMP: 0
2862 017214 017214 .+.40
2863 017214 000000 MDATA: 0
2864 017256 .+.40
2865 017256 005015 042522 042503 EM0: .ASCIZ <15><12>/RECEIVER DONE PRIMARY NOT SET./
017317 015 042012 052101 EM1: .ASCIZ <15><12>/DATA COMPARISON ERROR.../
017352 005015 050504 042440 EM2: .ASCIZ <15><12>/DQ ERROR FLAG SET. /
017400 005015 047516 051040 EM3: .ASCIZ <15><12>/NO RECEIVER INTERRUPTS!!! /
017436 005015 054105 042520 DH0: .ASCIZ <15><12>/EXPECTED FOUND RX ADDR. TX ADDR. MASK /
017513 015 042412 050130 DH1: .ASCIZ <15><12>/EXPECTED RECEIVED /
017541 015 042012 042521 DH2: .ASCIZ <15><12>/DQERR /

```



|      |        |        |     |          |        |       |   |
|------|--------|--------|-----|----------|--------|-------|---|
|      |        | 017556 |     | .EVEN    |        |       |   |
|      |        | 000005 |     | DT0:     | 5      |       |   |
| 2866 | 017560 | 006    | 004 |          | .BYTE  | 6.4   |   |
| 2867 | 017562 | 001270 |     |          | SAVR5  |       |   |
| 2868 | 017564 | 006    | 001 |          | .BYTE  | 6.1   |   |
| 2869 | 017566 | 001266 |     |          | SAVR4  |       |   |
| 2870 | 017570 | 006    | 004 |          | .BYTE  | 6.4   |   |
| 2871 | 017572 | 001260 |     |          | SAVR1  |       |   |
| 2872 | 017574 | 006    | 004 |          | .BYTE  | 6.4   |   |
| 2873 | 017576 | 001256 |     |          | SAVR0  |       |   |
| 2874 | 017600 | 006    | 002 |          | .BYTE  | 6.2   |   |
| 2875 | 017602 | 012370 |     |          | MASK   |       |   |
| 2876 | 017604 | 000002 |     | DT1:     | 2      |       |   |
| 2877 | 017606 | 003    | 006 |          | .BYTE  | 3.6   |   |
| 2878 | 017610 | 014302 |     |          | GDCHAR |       |   |
| 2879 | 017612 | 003    | 002 |          | .BYTE  | 3.2   |   |
| 2880 | 017614 | 014274 |     |          | CHAR   |       |   |
| 2881 | 017616 | 000001 |     | DT2:     | 1      |       |   |
| 2882 | 017620 | 006    | 002 |          | .BYTE  | 6.2   |   |
| 2883 | 017622 | 014266 |     |          | ERR    |       |   |
| 2884 | 017624 |        |     | .ERRTAB: |        |       |   |
| 2885 | 017624 | 000000 |     |          | 0      |       |   |
| 2886 | 017626 | 000000 |     |          | 0      |       |   |
| 2887 | 017630 | 000000 |     |          | 0      |       |   |
| 2888 | 017632 | 017256 |     |          | EM0    |       |   |
| 2889 | 017634 | 000000 |     |          | 0      | :HALT | 1 |
| 2890 | 017636 | 000000 |     |          | 0      |       |   |
| 2891 | 017640 | 017317 |     |          | EM1    |       |   |
| 2892 | 017642 | 017436 |     |          | DH0    | :HALT | 2 |
| 2893 | 017644 | 017556 |     |          | DT0    |       |   |
| 2894 | 017646 | 017317 |     |          | EM1    |       |   |
| 2895 | 017650 | 017513 |     |          | DH1    | :HALT | 3 |
| 2896 | 017652 | 017604 |     |          | DT1    |       |   |
| 2897 | 017654 | 017352 |     |          | EM2    |       |   |
| 2898 | 017656 | 017541 |     |          | DH2    | :HALT | 4 |
| 2899 | 017660 | 017616 |     |          | DT2    |       |   |
| 2900 | 017662 | 017400 |     |          | EM3    |       |   |
| 2901 | 017664 | 000000 |     |          | 0      | :HALT | 5 |
| 2902 | 017666 | 000000 |     |          | 0      |       |   |
| 2903 |        | 000001 |     | .END     |        |       |   |









|         |        |       |       |       |       |       |       |       |       |       |       |       |       |       |
|---------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| SAVNUM  | 001276 | 747*  | 816#  | 936*  | 2378* | 2380* |       |       |       |       |       |       |       |       |
| SAVPC   | 001274 | 815#  | 2560* | 2725  |       |       |       |       |       |       |       |       |       |       |
| SAVRO   | 001256 | 808#  | 2569* | 2574  | 2873  |       |       |       |       |       |       |       |       |       |
| SAVR1   | 001260 | 809#  | 2568* | 2575  | 2871  |       |       |       |       |       |       |       |       |       |
| SAVR2   | 001262 | 810#  | 2567* | 2576  |       |       |       |       |       |       |       |       |       |       |
| SAVR3   | 001264 | 811#  | 2566* | 2577  |       |       |       |       |       |       |       |       |       |       |
| SAVR4   | 001266 | 812#  | 2565* | 2578  | 2869  |       |       |       |       |       |       |       |       |       |
| SAVR5   | 001270 | 813#  | 2564* | 2579  | 2867  |       |       |       |       |       |       |       |       |       |
| SAVSP   | 001272 | 814#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SAVOS = | 104406 | 849#  | 2666  |       |       |       |       |       |       |       |       |       |       |       |
| SCOPE = | 104400 | 837#  | 1176  | 1316  | 1385  | 1454  | 1523  | 1592  | 1661  | 1730  | 1796  | 1862  | 1928  | 1994  |
|         |        | 2060  | 2126  | 2192  | 2258  |       |       |       |       |       |       |       |       |       |
| SCOPI = | 104401 | 839#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SEQ. =  | 000014 | 633#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SETON   | 002752 | 1120# |       |       |       |       |       |       |       |       |       |       |       |       |
| SPACNT= | 015635 | 2596* | 2617  | 2620* | 2636# |       |       |       |       |       |       |       |       |       |
| STACK = | 001200 | 576#  | 934   | 1009  | 2720  | 2741  |       |       |       |       |       |       |       |       |
| STFLG   | 001311 | 825#  | 937*  |       |       |       |       |       |       |       |       |       |       |       |
| SV05    | 015326 | 2564# |       |       |       |       |       |       |       |       |       |       |       |       |
| SWR     | 001200 | 779#  | 947*  | 952   | 956*  | 967   | 970   | 981   | 988   | 994   | 1013  | 1021  | 2408  | 2415  |
|         |        | 2432  | 2444  | 2655  | 2660  | 2707  | 2715  | 2717  | 2767  |       |       |       |       |       |
| SWREG   | 000176 | 690#  | 956   | 967   | 2767  | 2783  | 2790  |       |       |       |       |       |       |       |
| SWREGC  | 016440 | 2778  | 2788# |       |       |       |       |       |       |       |       |       |       |       |
| SW00 =  | 000001 | 556#  | 981   |       |       |       |       |       |       |       |       |       |       |       |
| SW01 =  | 000002 | 555#  | 1021  |       |       |       |       |       |       |       |       |       |       |       |
| SW02 =  | 000004 | 554#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SW03 =  | 000010 | 553#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SW04 =  | 000020 | 552#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SW05 =  | 000040 | 551#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SW06 =  | 000100 | 550#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SW08 =  | 000400 | 549#  | 2715  |       |       |       |       |       |       |       |       |       |       |       |
| SW09 =  | 001000 | 548#  | 2432  |       |       |       |       |       |       |       |       |       |       |       |
| SW10 =  | 002000 | 547#  | 2717  |       |       |       |       |       |       |       |       |       |       |       |
| SW11 =  | 004000 | 546#  | 2415  |       |       |       |       |       |       |       |       |       |       |       |
| SW12 =  | 010000 | 545#  | 2444  | 2655  |       |       |       |       |       |       |       |       |       |       |
| SW13 =  | 020000 | 544#  | 2660  |       |       |       |       |       |       |       |       |       |       |       |
| SW14 =  | 040000 | 543#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SW15 =  | 100000 | 542#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SYNBIT= | 100000 | 615#  | 712   | 1109  |       |       |       |       |       |       |       |       |       |       |
| SYNC    | 013052 | 1111* | 1114* | 1125  | 1266  | 1335  | 1404  | 1473  | 1542  | 1611  | 1680  | 1749  | 1815  | 1881  |
|         |        | 1947  | 2013  | 2079  | 2145  | 2211  | 2330# |       |       |       |       |       |       |       |
| SYNC. = | 000011 | 630#  |       |       |       |       |       |       |       |       |       |       |       |       |
| SYNC1   | 012372 | 1276* | 1345* | 1414* | 1483* | 1552* | 1621* | 1690* | 1756* | 1758* | 1773  | 1822* | 1824* | 1839  |
|         |        | 1888* | 1890* | 1905  | 1954* | 1956* | 1971  | 2020* | 2022* | 2037  | 2086* | 2088* | 2103  | 2152* |
|         |        | 2154* | 2169  | 2218* | 2220* | 2235  | 2263# |       |       |       |       |       |       |       |
| SYNC2   | 012374 | 1277* | 1291  | 1346* | 1360  | 1415* | 1429  | 1484* | 1498  | 1553* | 1567  | 1622* | 1636  | 1691* |
|         |        | 1705  | 1759* | 1825* | 1891* | 1957* | 2023* | 2089* | 2155* | 2221* | 2264# |       |       |       |
| TEMP    | 017152 | 2592  | 2600  | 2627* | 2742* | 2743* | 2861# |       |       |       |       |       |       |       |
| TEMP1   | 001244 | 669*  | 670*  | 803#  | 974*  | 975   | 979*  | 2852  |       |       |       |       |       |       |
| TEMP2   | 001246 | 804#  | 975*  | 1162* | 1171* | 1203* | 1266* | 1268* | 1272* | 1273* | 1275* | 1276  | 1277  | 1278  |
|         |        | 1307* | 1308* | 1309  | 1335* | 1337* | 1341* | 1342* | 1344* | 1345  | 1346  | 1347  | 1376* | 1377* |
|         |        | 1378  | 1404* | 1406* | 1410* | 1411* | 1413* | 1414  | 1415  | 1416  | 1445* | 1446* | 1447  | 1473* |
|         |        | 1475* | 1479* | 1480* | 1482* | 1483  | 1484  | 1485  | 1514* | 1515* | 1516  | 1542* | 1544* | 1548* |
|         |        | 1549* | 1551* | 1552  | 1553  | 1554  | 1583* | 1584* | 1585  | 1611* | 1613* | 1617* | 1618* | 1620* |
|         |        | 1621  | 1622  | 1623  | 1652* | 1653* | 1654  | 1680* | 1682* | 1686* | 1687* | 1689* | 1690  | 1691  |
|         |        | 1692  | 1721* | 1722* | 1723  | 1749* | 1751* | 1753* | 1758  | 1759  | 1760  | 1815* | 1817* | 1819* |







|         |       |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
|---------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|--|--|
| DQEND   | 1#    | 2349 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| DQFRNT  | 1#    | 522  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| HLT     | 586#  | 1173 | 1181 | 1206 | 1236 | 1301 | 1313 | 1370 | 1382 | 1439 | 1451 | 1508 | 1520 | 1577 | 1589 |  |  |
|         | 1646  | 1658 | 1715 | 1727 | 1783 | 1793 | 1849 | 1859 | 1915 | 1925 | 1981 | 1991 | 2047 | 2057 | 2113 |  |  |
|         | 2123  | 2179 | 2189 | 2245 | 2255 |      |      |      |      |      |      |      |      |      |      |  |  |
| IDENT   | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| ORANGE  | 1#    | 1035 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTA1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTB1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTC1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTD1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTE1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTF1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTH1  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| TESTH2  | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$BEGIN | 1#    | 1005 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$BUFFE | 1#    | 2856 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$CATCH | 1#    | 639  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$CLRVE | 1#    | 970  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$CONVR | 1#    | 2581 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$DQCHR | 1090# | 1249 | 1318 | 1387 | 1456 | 1525 | 1594 | 1663 | 1732 | 1798 | 1864 | 1930 | 1996 | 2062 | 2128 |  |  |
|         | 2194  |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$EOP   | 1#    | 2349 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$GETFL | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$GETPA | 1#    | 1023 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$HEADE | 1#    | 522  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$HLT   | 1#    | 2651 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$INSTR | 1#    | 2460 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$INTNP | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$MAINT | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$MSG   | 1#    | 2797 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$PARAM | 1#    | 2498 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$PFAIL | 1#    | 2729 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$REG   | 1#    | 2557 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$SCOPE | 1#    | 2404 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$SCOPI | 1#    | 2428 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$SETFL | 1#    |      |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$SETVE | 1#    | 641  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$START | 1#    | 925  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$SYMBO | 1#    | 539  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$TRAPS | 1#    | 828  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$TRPDE | 1#    | 837  | 839  | 841  | 843  | 845  | 847  | 849  | 851  | 853  | 855  | 857  | 859  | 861  | 863  |  |  |
| \$TRPSR | 1#    | 2638 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$TSTN  | 1#    | 1035 | 1101 | 1255 | 1324 | 1393 | 1462 | 1531 | 1600 | 1669 | 1738 | 1804 | 1870 | 1936 | 2002 |  |  |
|         | 2068  | 2134 | 2200 |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$TYPE  | 1#    | 2436 |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |
| \$VARIA | 1#    | 768  |      |      |      |      |      |      |      |      |      |      |      |      |      |  |  |

. ABS. 017670 000

ERRORS DETECTED: 0

CZDQHE.BIN,CZDQHE.LST/CRF/SOL/NL:TOC=CZDQXX.MAC,CZDQHE.P11  
RUN-TIME: 19 28 2 SECONDS

CZDQH MACY11 30A(1052) 03-DEC-80 08:29 PAGE 67  
CZDQHE.P11 03-DEC-80 08:27 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0064

RUN-TIME RATIO: 78/50=1.5  
CORE USED: 19K (37 PAGES)