IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBB-E-D

PRODUCT NAME: 11/40 - 11/45 CPU PARITY TEST

DATE CREATED: MAY 1975

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: BRUCE BURGESS

CONTENTS

1. ABSTRACT

THIS PROGRAM WILL TEST PARITY ABORTS DURING CPU EXECUTION OF READ/RESTORE (DATI) AND READ/PAUSE (DATIP) MEMORY OPERATIONS. NORMAL PARITY IS GENERATED WHEN WRITING TO MEMORY (DATO) AND CHECKED FOR 'OTHER' PARITY WHEN READING FROM MEMORY (DATI OR DATIP). PARITY ABORTS ARE FORCED BY SETTING A PARITY CONTROL REGISTER FOR 'OTHER' PARITY (NOT NORMAL) BEFORE EXECUTION OF DATI OR DATIP INSTRUCTIONS.

THIS PROGRAM DOES NOT TEST MEMORY; IT TESTS THE PROCESSOR AND ASSUMES MEMORY TO BE FUNCTIONING PROPERLY. MAINDEC-11-DCMFA WILL TEST MEMORY AND SHOULD BE RUN IN CONJUNCTION WITH THIS PROGRAM TO PROVIDE A THOROUGH TEST OF PARITY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11/40 OR PDP-11/45 COMPUTER WITH CONSOLE TELETYPE, AND AN MF11 (CORE) OR MS11 (MOS) PARITY OPTION WITH ASSOCIATED PARITY MEMORY ANY WHERE WITHIN MACHINE BOUNDS

2.2 STORAGE

THIS PROGRAM REQUIRES APPROXIMATELY 3K STORAGE.

2.3 PRELIMINARY PROGRAMS

SINCE THIS PROGRAM ASSUMES MEMORY TO BE FUNCTIONING PROPERLY (AS MENTIONED IN THE ABSTRACT) IT WOULD BE WISE TO RUN MAINDEC-11-DCMFA BEFORE THIS PROGRAM.

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING .ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE PARAGRAPH 5.

4.2 STARTING ADDRESS

THE PROGRAM IS STARTED AT ADDRESS 200.

4.3 OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY USING .ABS LOADER
2. LOAD ADDRESS 200
3. SET SWITCHES, IF ANY (SEE PARAGRAPH 5.)
4. PRESS START
5. THE PROGRAM WILL LOOP AND THE TELETYPE BELL WILL RING EVERY PASS (IF SW<10>=0)

5. OPERATIONAL SWITCH SETTINGS

```
SW<15>=1...HALT ON ERROR
SW<14>=1...LOOP ON TEST
SW<13>=1...INHIBIT ERROR TYPEOUTS
SW<12>=1...ALLOW USER TO SELECT
          ...REGISTER HE DESIRES
SW<11>=1...INHIBIT ITERATIONS
            (NOT USED)
SW<10>=1...RING BELL ON ERROR
SW<10>=0...RING BELL ON PASS COMPLETE
SW<09>=1...LOOP ON ERROR
SW<08>=1...LOOP ON SPECIAL TEST SHOWN
            IN SWS<7 THRU 0>
SW<06>=1...DON'T ENABLE KT11 OPTION
            EVEN IF PRESENT
SWS<7 THRU 0>...USED IN CONJUNCTION WITH
                SW<08> DESCRIBED ABOVE
```

5.1 THE SWITCHES DEFINED ABOVE ARE SELF EXPLANATORY EXCEPT FOR THE SPECIAL COMBINATION OF SWS<06, 07 THRU 00, AND 12>. TWO (2) EXAMPLES ARE AS FOLLOWS:

(1) THE USER WISHES TO SELECT A PARTICULAR REGISTER TO UNDERGO TESTING, NOT USE THE KT11, AND LOOP ON TST37

(A) LOAD ADDRESS 200
(B) SET SWITCHES 6 AND 12
(C) HIT START
(D) THE TELETYPE WILL RESPOND BY ASKING THE USER TO 'TYPE THE REGISTER YOU DESIRE & HIT CARRIAGE RETURN' , AND WILL WAIT FOR THIS RESPONSE E.G. 172110 (NOT 772110)
(E) BEFORE TYPING A REPLY AND HITTING A CARRIAGE RETURN, PUT SW<06> AND SW<12> DOWN, SET SW<08>, AND PLACE THE VALUE 37 INTO SWS<07 THRU 00>
(F) TYPE THE RESPONSE AND HIT CARRIAGE RETURN
(G) YOU SHOULD BE LOOPING ON TST37 WHICH CAN BE EASILY VERIFIED BY EXAMINING THE CONTENTS OF SLPADR

NOTE: LOOPING ON A PARTICULAR TEST CAPABILITY WILL ONLY WORK WHEN THE USER HAS SELECTED A PARTICULAR REGISTER USING THE SW<12> OPTION

(2) THE USER WISHES TO SELECT A PARTICULAR REGISTER TO UNDERGO TESTING, USE THE KT11, AND LOOP ON TST37

USE THE SAME PROCEDURE DESCRIBED UNDER (1) ABOVE EXCEPT ONLY SET SW<12> UNDER ITEM (B)

5.2 WHEN USING THE SW<12> OPTION THE RESPONSE EXPECTED IS A 6 - DIGIT OCTAL NUMBER E.G. 172100, 172120, ETC.

IF THE USER FOR SOME REASON DOES NOT TYPE A 6-DIGIT OCTAL NUMBER E.G. 172A.....THE TELETYPE WILL CARRIAGE RETURN, LINE FEED, AND TYPE A '?' (QUESTION MARK). IT WILL SIT HERE WAITING FOR THE NUMBER TO BE TYPED CORRECTLY FOLLOWED BY A CARRIAGE RETURN.

6. SUBROUTINE ABSTRACTS

6.1 ABORT

ONCE A REGISTER IS FOUND TO BE PRESENT, THIS ROUTINE WILL SEARCH MEMORY, PERFORMING A DATI, UNTIL THE CORRESPONDING PARITY MEMORY AREA IS FOUND. THIS ROUTINE IS ONLY USED DURING THE PROGRAM TABLE CREATION.

6.2 $ACCEPT

THIS ROUTINE IN CONJUNCTION WITH $READC WILL ACCEPT AN OCTAL NUMBER FROM THE TELETYPE. THESE 2 ROUTINES ARE SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. THEY ARE USED WHEN SW<12> IS SET TO A 1 BY THE USER.

6.3 $B2OCT

THIS ROUTINE HANDLES TYPING OF BINARY TO OCTAL (ASCII) NUMBERS. IT IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. IT IS USED FOR ERROR REPORTING.

6.4 CHECKLOC

AFTER A PARITY ABORT HAS BEEN FORCED BY THE PROGRAM THIS ROUTINE WILL LOOK FOR THE CORRECT HIGH ORDER ERROR ADDRESS BITS IN THE PARITY CONTROL REGISTER AS WELL AS THE PROPER PC PUSH ON THE STACK FROM THE ABORT. ANY DISCREPANCIES ARE STORED FOR ERROR PRINTOUT.

6.5 COMPUT

THIS ROUTINE IS INITIALLY USED TO DETERMINE (TOGETHER WITH THE ABORT ROUTINE) WHERE/IF PARITY MEMORY PRESIDES FOR A SPECIFIC PARITY CONTROL REGISTER. IT CREATES A 2 LOCATION MEMORY MAP AT THE HIGH END OF A 1K BANK. FOR EXAMPLE, IF THE ADDRESS 17776 WERE ADDRESSABLE, THEN THIS ROUTINE WOULD GIVE THE FOLLOWING LOCATIONS AND CONTENTS:

LOC.	CONTENTS*	*KT11 NOT TURNED ON
17400	17402	
17402	17402	

LOC.	CONTENTS	KT11 TURNED ON
17400	23402	
17402	23402	

THESE 2 LOCATIONS AND CONTENTS WOULD THEN BE USED BY THE ABORT ROUTINE. IF A PARITY ABORT OCCURRED THEN THESE LOCATIONS AND CONTENTS WITH THEIR ASSOCIATED PARITY CONTROL REGISTER WOULD BE USED FOR SUBSEQUENT TESTING. R1 WILL ALWAYS HOLD THE FIRST LOCATION OF THE 2 LOCATION MAP.

6.6 $EOP

THIS IS THE END OF PASS ROUTINE. BEFORE THE PROGRAM LOOPS BACK TO TEST THE NEXT TABLE ENTRY (OR ITERATE ON THE CURRENT ONE) THIS ROUTINE IS ENCOUNTERED. IT IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME.

6.7 FLAGSCLR

THIS ROUTINE IS USED TO CLEAR PERTINENT FLAGS BEFORE PASSING THRU THE PROGRAM WITH ANOTHER TABLE ENTRY OR ITERATING ON THE CURRENT ENTRY.

6.8 $HLT

THIS ROUTINE CALLED (IN NUMEROUS PLACES THRU OUT THE PROGRAM) BY THE 'EMT' INSTRUCTION IS USED WHENEVER AN ERROR HAS BEEN DETECTED. THIS ROUTINE RELIES ON SWS<9,10,14,15> FOR FUNCTIONING AND IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME. THE TYPERR ROUTINE WHICH TYPES OUT THE ERROR MESSAGES AND DATA HEADERS IS CALLED WITHIN THIS ROUTINE.

6.9 INITIALIZE

THIS ROUTINE WILL COMPLETELY REINITIALIZE PROGRAM FLAGS, ETC. BEFORE RESTARTING THE PROGRAM OVER AT THE BEGINNING OF THE TABLE.

6.10 PARTST

ONCE A PARITY CONTROL REGISTER HAS BEEN FOUND TO BE PRESENT THEN THIS ROUTINE IS USED TO CHECK IF THE REGISTER IN GOOD OPERATION BEFORE TESTING IS CONDUCTED.

6.11 $PWRDN

THIS ROUTINE IN CONJUNCTION WITH $PWRUP COMPRISE THE 'POWER FAIL' ROUTINES. IF THE SYSTEM GOES DOWN WHILE THE PROGRAM IS EXECUTING, GENERAL PURPOSE REGISTERS 0 THRU 5 ARE SAVED. WHEN THE SYSTEM POWERS BACK UP THE MESSAGE 'POWER' WILL BE TYPED ON THE CONSOLE TELETYPE. GENERAL PURPOSE REGISTERS 0 THRU 5 ARE RESTORED, AND THE PROGRAM WILL AUTOMATICALLY RESTART FROM THE BEGINNING. THESE 2 ROUTINES ARE SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME.

6.12 $SCOPE

THIS ROUTINE CALLED (AT THE BEGINNING AND END OF EVERY TEST) BY THE 'IOT' INSTRUCTION IS USED FOR TEST LOOPING PURPOSES. IT DEPENDS UPON SW5<8,9,11,14> FOR FUNCTIONING AND RECORDS THE STARTING ADDRESS OF EACH TEST IN '$LPADR' AS IT IS BEING ENTERED. 'LPADR' (IN THE COMMON TAG SECTION OF THE PROGRAM) MAY BE EXAMINED TO DETERMINE THE LAST TEST SUCCESSFULLY COMPLETED. THIS ROUTINE IS SUPPLIED BY AN EXTERNAL PACKAGE (SYSMAC.SML) AT ASSEMBLY TIME.

6.13 TRAPCATCHER

A '.+2' AND 'HALT' SEQUENCE IS REPEATED FROM LOCATION 0 TO LOCATION 776 TO CATCH ANY UNEXPECTED DEVICE TRAPS. THUS, ANY UNEXPECTED TRAPS WILL HALT AT THE DEVICE TRAP VECTOR +2. WHEN/IF THIS OCCURS EXAMINATION OF THE STACK SHOULD BE THE STARTING POINT TO FIND WHERE IN THE PROGRAM YOU WERE BEFORE THE UNEXPECTED TRAP OCCURRED.

6.14 TYPERR

THIS ROUTINE CALLED WITHIN THE $HLT ROUTINE HANDLES THE ERROR MESSAGE AND DATA HEADER PRINTOUTS.

6.15 VECSET

THIS ROUTINE IS ACCESSED AT THE BEGINNING OF EVERY TEST TO SET UP THE ADDRESS OF THE SERVICE ROUTINE FOR THE PARITY ABORT VECTOR 114.

7. ERROR PRINTOUTS

*** SPECIAL NOTE ***
****

BE AWARE THAT WHEN THE PROGRAM IS BEING EXECUTED WITH MEMORY MANAGEMENT ENABLED, THE 'ACTUAL' AND 'EXPECTED' ABORT PC VALUES GREATER THAN THE LAST ADDRESS OF THE PROGRAM ARE VIRTUAL ADDRESSES. TO FIND THE PHYSICAL (OR IN REALITY) ADDRESS PULL THE OFFSET VALUE FROM THE PROGRAM TABLE DESCRIBED IN PARAGRAPH 9.2, AND DO THE ADDITION PROCEDURE OUTLINED UNDER ITEM (2), PARAGRAPH 9.3

*** END OF SPECIAL NOTE ***
****

7.1 HLT +1

```
TEST DIDN'T ABORT
PROGRAM   REGISTER     EXPECTED
   PC     UNDER TEST   ABORT PC
**      APPROPRIATE  VALUES  **
```

7.2 HLT +2

```
FATAL ERROR TO PROGRAM
PROGRAM   REGISTER
   PC     UNDER TEST
**      APPROPRIATE VALUES  **
```

NOTE: THIS ERROR REPORT WILL COME FROM 1 OF 3 TESTS IN THE 'PARTST' ROUTINE. SOMETHING WILL BE WRONG WITH BIT00 OR BIT02 OF THE PARITY CONTROL REGISTER

7.3 HLT +3

```
ABORTED INCORRECTLY
PROGRAM REGISTER        EXPECTED        ACTUAL  EXPECTED        ACTUAL
   PC  UNDER TEST       ADDR.BITS     ADDR.BITS ABORT PC      ABORT PC
**              APPROPRIATE  VALUES                                  **
```

NOTE: THIS ERROR REPORT WILL COVER A NUMBER OF OCCURRENCES:

(1) THE EXPECTED HIGH ORDER ADDRESS BITS AND THE EXPECTED ABORT PC PUSHED ON THE STACK WERE BOTH WRONG.
   A) IN THE CASE OF AN OLD MOS DESIGN WITH NO ADDRESS BITS ZEROS (0'S) WILL APPEAR UNDER THE ADDR. BITS COLUMNS.
(2) THE EXPECTED HIGH ORDER ADDRESS BITS WERE CORRECT BUT THE WRONG ABORT PC WAS PUSHED ON THE STACK.
   B) IN THIS CASE THE VALUES APPEARING UNDER THE ADDR. BITS COLUMNS WOULD BE THE SAME
(3) THE EXPECTED HIGH ORDER ADDRESS BITS WERE INCURRECT BUT THE CORRECT ABORT PC WAS PUSHED ON THE STACK
   C) IN THIS CASE THE VALUES APPEARING UNDER THE ABORT PC COLUMNS WOULD BE THE SAME

7.4 HLT +4

```
NO PARITY MEMORY FOUND BELOW 28K
REGISTER
UNDER TEST
**  APPROPRIATE VALUE  **
```

NOTE: THIS PRINTOUT WILL OCCUR FOR 1 OF 2 REASONS:

(1) WITH NO KT11 OPTION ON THE SYSTEM, A PARITY CONTROL REGISTER WAS FOUND BUT THE CORRESPONDING PARITY MEMORY WAS NOT FOUND IN LOOKING ALL THE WAY UP TO 28K, OR
(2) A KT11 OPTION IS ON THE SYSTEM WITH THE PARITY CONTROL REGISTERS' CORRESPONDING PARITY MEMORY AREA ABOVE 28K BUT THE USER DISABLED THE KT11 (DID NOT ALLOW USE) BY SETTING SW<06>.

7.5 HLT +5

```
RESET DOESN'T WORK
PROGRAM  REGISTER
   PC    UNDER TEST
**      APPROPRIATE VALUES     **
```

NOTE: IF A KT11 OPTION IS PRESENT, AND NOT DISABLED BY SETTING SW<06>, THEN THE TEST (TEST #4) INCURRING THIS PRINTOUT WILL NOT BE EXECUTED.

7.6 HLT +6

```
USER SELECTED REGISTER NOT PRESENT
PROGRAM
   PC
**  APPROPRIATE VALUE  **
```

NOTE: THIS PRINTOUT WILL COME ABOUT AS A RESULT OF USING THE SW<12> OPTION. IF IN RESPONSE TO THE MESSAGE "TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN" THE USER TYPES A NON-EXISTANT REGISTER ADDRESS THEN THE ABOVE PRINTOUT WILL OCCUR AND THE USER RESPONSE MESSAGE WILL BE REITERATED.

7.7 HLT +7

```
NO PARITY MEMORY FOUND AT ALL
REGISTER
UNDER TEST
**  APPROPRIATE VALUE  **
```

NOTE: THIS ERROR PRINTOUT COULD OCCUR FOR 1 OF 2 REASONS:

(1) THE KT11 OPTION IS PRESENT AND NOT DISABLED (USING SW<06>) INDICATING NOWHERE WAS A CORRESPONDING PARITY MEMORY AREA FOUND, OR
(2) A POSSIBLE HOLE IN MEMORY EXISTS BECAUSE WE TIMED OUT BEFORE REACHING THE SUPPOSED SYSTEM MAXIMUM CORE LOCATION

7.8 HLT +10

DIDN'T ABORT OR RECOGNIZE STACK VIOLATION
PROGRAM REGISTER EXPECTED
PC UNDER TEST ABORT PC
** APPROPRIATE VALUES **

7.9 HLT +11

ABORTED BUT STACK VIOLATION NOT RECOGNIZED
PROGRAM REGISTER
PC UNDER TEST
** APPROPRIATE VALUES **

7.10 HLT +12

STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED
PROGRAM REGISTER
PC UNDER TEST
** APPROPRIATE VALUES **

8. RESTRICTIONS

AS MENTIONED IN PARAGRAPHS 1 AND 2.3, THIS PROGRAM DOES NOT TEST MEMORY, IT TESTS THE PROCESSOR. IF PARITY MEMORY CHECKING IS WHAT YOU ARE AFTER THEN RUN MAINDEC-11-DCMFA

9. MISCELLANEOUS

9.1 EXECUTION TIME

ERROR FREE PASSES ARE ON THE ORDER OF 1 OR 2 SECONDS

9.2 PROGRAM TABLE LOCATIONS

WHEN THE SW<12> OPTION IS NOT USED THE PROGRAM WILL FIND ALL PARITY CONTROL REGISTERS AND A CORRESPONDING PARITY MEMORY LOCATION AND STORE THESE VALUES INTO A MAXIMUM 10 WORD, 4 COLUMN TABLE TO BE USED BY THE PROGRAM FOR TESTING. IF, FOR EXAMPLE, 2 PARITY CONTROL REGISTERS AND PARITY MEMORY AREAS ARE FOUND THEN PASS 1 OF THE PROGRAM WILL USE THE 1ST TABLE ENTRY INFORMATION; PASS 2 THE 2ND TABLE ENTRY INFROMATION; PASS 3 BACK TO THE 1ST TABLE ENTRY INFORMATION, ETC.

THE ABSOLUTE CORE LOCATIONS FOR TABLE ENTRYS ARE AS FOLLOWS:

$REG0(LOCATION 1340) WILL CONTAIN THE 1ST PARITY REGISTER

LOCATION 1342 UP TO 1364 WILL CONTAIN ANYMORE REGISTERS FOUND

$TMP0(LOCATION 1366) WILL CONTAIN A PARITY MEMORY LOCATION CORRESPONDING TO THE REGISTER IN $REG0

LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS

$SET0(LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED WITH THE CORRESPONDING VALUE IN $TMP0

LOCATION 1422 UP TO 1444 WILL CONTAIN THE CORRESPONDING OFFSET VALUES FOR THE OTHER REGISTERS.

NTER0(LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED WITH THE PARITY REGISTER IN $REG0

LOCATION 1452 UP TO 1474 WILL CONTAIN THE CORRESPONDING INTERLEAVE FACTORS FOR THE OTHER REGISTERS

9.3 PROGRAM TABLE SET UP WITH KT11 ENABLED

IF A KT11 OPTION IS PRESENT AND IS NOT DISABLED THRU USER SETTING OF SW<06> (SEE PARAGRAPH 5.), THE PROGRAM TABLE LOCATIONS AND CONTENTS WILL APPEAR AS DESCRIBED AND SHOWN IN THE EXAMPLE BELOW.

GIVEN: A) 172100 GOVERNING 0-8K MOS MEMORY
B) 172102 GOVERNING 8-16K CORE MEMORY
C) 172112 GOVERNING 40-48K CORE MEMORY

LOC.	REGISTER COLUMN	LOC.	MEMORY COLUMN	LOC.	OFFSET COLUMN	ILEAVE COLUMN
1340	172100	1366	23700	1420	140	2
1342	172102	1370	23700	1422	200	2
1344	172112	1372	23700	1424	2500	1
1346	0					

NOTES: (1) WHEN THE KT11 IS ENABLED THE MEMORY COLUMN CONTENTS WILL ALWAYS BE THE SAME BASE ADDRESS. (UNLESS WE HAVE MEMORY INTERLEAVING)
(2) 23700 IS A PAGE 1 ADDRESS AS SEEN BY THE KT11.

THIS VIRTUAL ADDRESS AND ITS' CORRESPONDING OFFSET VALUE WILL GIVE THE PHYSICAL ADDRESS AS FOLLOWS:

```
VIRTUAL ADDRESS =       2  3  7  0  0
+ OFFSET VALUE  =       1  4  0
                        ----------
PHYSICAL ADDRESS=       1  7  7  0  0
```

NOTICE THAT THE OFFSET VALUE IS TO BE SHIFTED TWICE TO THE LEFT AND THE LEFTMOST DIGIT OF THE VIRTUAL ADDRESS TO BE IGNORED BEFORE ADDING.

(3) THE PHYSICAL ADDRESS VALUE FROM ABOVE IS THE VALUE USED BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5) WHICH WILL DROP THE PHYSICAL ADDRESS DOWN SO AS NOT TO DESTROY THE .ABS LOADER (I.E. - 376 IS SUBTRACTED) THUS GIVING A PHYSICAL ADDRESS FOR THE 2 LOCATION MAP CREATION AND TESTING. THIS VALUE IS ALWAYS PRESENT IN R1. (GENERAL PURPOSE REGISTER 1)

(4) THE ZERO IN THE LAST REGISTER COLUMN LOCATION IS THE PROGRAM TABLE TERMINATION INDICATOR

(5) A '1' IN THE ILEAVE COLUMN MEANS NO INTERLEAVING
A '2' IN THE ILEAVE COLUMN MEANS 2-WAY INTERLEAVING
.
.
ETC. (UP TO 8-WAY)

9.4 PROGRAM TABLE SETUP WITH KT11 DISABLED

IF A KT11 OPTION IS PRESENT AND IS DISABLED THRU USER SETTING OF SW<06> (SEE PARAGRAPH 5.) OR NO KT11 OPTION IS PRESENT THEN, THE PROGRAM TABLE LOCATIONS AND CONTENTS WILL APPEAR AS DESCRIBED AND SHOWN IN THE EXAMPLE BELOW.

GIVEN: A) 172100 GOVERNING 0-8K MOS MEMORY
B) 172102 GOVERNING 8-16K CORE MEMORY

LOC.	REGISTER COLUMN	LOC.	MEMORY COLUMN	LOC.	OFFSET COLUMN	ILEAVE COLUMN
1340	172100	1366	1700	1420	0	1
1342	172102	1370	25700	1422	0	1
1344	0					

NOTES:

(1) THE MEMORY COLUMN LOCATION CONTENTS ARE THE ACTUAL VALUES USED BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5)

(2) THE OFFSET COLUMN CONTENTS ARE NOT IN AFFECT UNLESS THE KT11 IS ENABLED (SEE PARAGRAPH 9.3)

(3) THE ZERO IN LOCATION 1344 WOULD BE THE PROGRAM TABLE TERMINATION INDICATOR.

(4) A '1' IN THE ILEAVE COLUMN MEANS NO INTERLEAVING
A '2' IN THE ILEAVE COLUMN MEANS 2-WAY INTERLEAVING
.
.
ETC. (UP TO 8-WAY)

9.5 STACK POINTER

THE STACK IS INITIALLY SET TO 1100. IT WILL REMAIN THIS VALUE FOR ALL TESTS NOT DEPENDENT ON THE STACK BEING IN PARITY MEMORY AREA. FOR EXAMPLE, A TEST CHECKING FOR A PARITY ABORT ON THE 1ST 'POP' FROM AN 'RTI' INSTRUCTION WOULD REQUIRE THE STACK TO BE IN THE PARITY MEMORY AREA CONTROLLED BY THE REGISTER UNDER TEST. IN THIS CASE THE STACK POINTER IS REPOSITIONED AND INITIALIZED TO THE 1ST ADDRESS OF THE 2 LOCATION MAP SET UP BY THE 'COMPUT' ROUTINE (SEE PARAGRAPH 6.5).

FOR EXAMPLE, CONSIDERING THE 2ND TABLE ENTRY GIVEN IN PARAGRAPH 9.4, THE 'COMPUT' ROUTINE WOULD SET UP A 2 LOCATION MAP STARTING AT LOCATION 23302. THE STACK POINTER, FOR PERTINENT TESTS MENTIONED ABOVE, WOULD THEN BE REINITIALIZED TO 23302.

NOTE: BEWARE! IF A KT11 OPTION IS PRESENT AND ENABLED AND YOU WISH TO EXAMINE THE CONTENTS OF THE STACK (AFTER A TEST REQUIRING THE STACK TO BE REPOSITIONED ABOVE 8K HAS BEEN EXECUTED) THE STACK WOULD NOT-NOT-NOT BE AT 23302 USING THIS EXAMPLE. IT WOULD BE AT 17302 BECAUSE OF AN OFFSET VALUE. SEE THE PHYSICAL ADDRESS CALCULATION EXPLANATION UNDER PARAGRAPH 9.3.

9.6 MAINTENANCE HINT

THE FOLLOWING SHOULD BE USEFUL INFORMATION FOR 11/45 USERS WHO WISH TO EXAMINE A TEST TO ASCERTAIN STEP BY STEP WHAT THE TEST DID. THE FOLLOWING INFORMATION PRESUMES THAT THE USER HAS ACCESS TO A MAINTENANCE BOARD.

(1) MAKING SURE THAT THE PARITY REGISTER CONTROLLING THE LOWER 4K DOES NOT HAVE BIT02 SET, PROCEED TO DEPOSIT A 0 INTO THE CORE LOCATION OF THE 'SCOPE' STATEMENT AT THE BEGINNING OF THE TEST.
(2) LOAD ADDRESS 200 (SETTING SW<12> IF DESIRED) AND HIT START
(3) THE PROGRAM WILL HALT AT THE CORE LOCATION USED IN (1) ABOVE
(4) PUT THE 'SINGLE INSTRUCTION' AND 'SINGLE BUS CYCLE' SWITCHES ON THE PROCESSOR CONSOLE DOWN
(5) HIT THE CONTINUE SWITCH REPEATEDLY UNTIL THE ADDRESS OF THE INSTRUCTION THAT WAS TO CAUSE THE PARITY ABORT APPEARS IN THE ADDRESS LIGHTS.
(6) SET THE DATA DISPLAY SELECT KNOB TO DISPLAY THE CPU MICROSTATE IN BITS 7-0.
(7) LOOKING AT THE MAINTENANCE BOARD, RIGHT SIDE UP, AND TOGGLE SWITCHES ON THE RIGHT; PRESS THE BOTTOM RIGHTMOST SWITCH TO THE RIGHT.
(8) THEN JUST LIGHTLY TAP THE BOTTOM LEFTMOST SWITCH (JUST ENOUGH FOR IT TO BOUNCE BACK) REPEATEDLY. THE MICROSTATES WILL BE DISPLAYED IN BITS 7-0 OF THE CONSOLE DATA REGISTER
(9) THE MICROSTATE VALUE THAT WAS IN THE CONSOLE DATA REGISTER JUST BEFORE IT TURNED 0 WAS THE ABORT MICROSTATE.

10. PROGRAM DESCRIPTION

THE MAIN FUNCTION OF THIS PROGRAM IS TO TEST THE ABILITY OF A PARITY CONTROL REGISTER TO INTERFACE PROPERLY WITH ITS CORRESPONDING MEMORY PARITY AREA THUS ALLOWING PARITY ABORTS ON CPU EXECUTION OF DATI AND DATIP INSTRUCTIONS SET UP WITH 'NOT NORMAL' (BAD) PARITY. BASIC COMBINATIONS OF SOURCE AND DESTINATION MODES ARE TESTED TO PICK UP ALL POSSIBLE MICROSTATES AT WHICH PARITY ABORTS CAN OCCUR. ALSO TESTED ARE SUCH THINGS AS:

(A) 1ST AND 2ND 'POP' ON A MARK INSTRUCTION
(B) THE SUB INSTRUCTION
(C) A 'MOV SM0,DM0' INSTRUCTION
(D) THE 'POP' ON AN RTS INSTRUCTION
(E) 1ST AND 2ND 'POP' ON AN RTI INSTRUCTION
(F) PS AN PC FETCH INSTRUCTIONS
(G) INDEXED WORD INSTRUCTIONS (DM6,DM7,SM6)
(H) CONDITIONAL BRANCH NOT OK INSTRUCTIONS
(I) STACK VIOLATIONS IN 'RED' AND 'YELLOW' ZONES

THIS PROGRAM USED IN CONJUNCTION WITH MAINDEC-11-DCMFA SHOULD PROVIDE A PRETTY THOROUGH TEST OF PARITY.

10.1 PROGRAM FLOW DIAGRAM