CP1BF
CP2B

TEST PROGRAM FOR CP1BF, CP2B

TEST OF MEMORY MANAGEMENT PART

*****

MMU2B

Procedure 12NC : 5111 991 13081

Concerned PV :
PV250-164 (tape)
PV350-164 (cassette)
PV450-164 (floppy)

PAF2B includes :

Procedure	
+ Listing	12NC : 5111 991 13091
+ Program	12NC : 5111 199 56971

0. TABLE OF CONTENTS

1. GENERAL

2. MINIMUM CONFIGURATION REQUIRED

3. LOADING PROCEDURE

4. NORMAL RUNNING OF THE PROGRAM

4.1. Standard Initializations
4.2. Program Execution
4.3. Error Detection
4.4. Interference Interrupts
4.5. Power Failure - Automactic Restart
4.6. Restarting Address

5. INFORMATION CODES

6. ERROR CODES

7. PROGRAM MODIFICATIONS

7.1. Parameters relating to MMU operations
7.2. Parameters related to program processing

1. GENERAL

The test program MMU2B is used as a complementary test of CP1/2B and PAF2B.

The objective of this test program is to control the correct processing of the memory management unit part.

The test does not check the extension bits. That is done by the BBRAM or BBMEMO test program.

2. MINIMUM CONFIGURATION REQUIRED

- CP2B : 5 CPU's boards,

- CP1BF: CPU board ,

- An input peripheral to load the program (Tape, Cassette, Floppy or disc test resident),

- An operator console,

A minimum memory module of 32K is required.

3. LOADING PROCEDURE

Using the following procedure load the test program into the system :

- Load the test program media in the appropriate input peripheral,
- Using the operator's panel select the appropriate values to load the program from the peripheral using IPL routine,
- Press the MC button,
- Press the IPL button.

When the loading is complete, the program will stop at hexadecimal address /700 so that program initialization can be performed by the operator.

4. NORMAL RUNNING OF THE PROGRAM

4.1. Standard Initialization

- Page Fault address /7A.

- Program execution initialized - without "one shot" with "stop error", "STOP on END", and "Edition" execution check 0 to 14.

4.2. Program Execution

- After loading the program (stop in /700), depress the RUN button.

- The program automatically executes the checks 00 to 14.

The program execution consists in following the information provided by the information codes displayed in A1 register when stops in /5E0 (see chapter 5).

When IF $<$ 15, it is the beginning of a check, the information provided through these codes fall into the following items :

- details about tests executed during this check.
- errors capable of occuring during this test.

For IF $>$15 follow the information given through these codes :

- when all checks have been executed, the program stops in /700. If the operator wishes to restart the test, he may again depress the RUN button.

- the operator may at any time stop the program and restart at /700.

- in case of a modification of the standard operation refer to chapter 7.2.

Note : At any time it is possible toreturn to /700 by depressing the Control Panel interrupt button. To restart do MC then RUN.

4.3. Error Detection

In the case of an error, the program stops at address /5F0 in standard operation (otherwise see paragraph 7).

For the meaning of the error message, displayed in A1 refer to paragraph 6.

4.4. Interference Interrupts

Any interrupt is considred as an interference interrupt (except for PWF/AR and RTC).

In this case, the program stops on its starting address (/700). By depressing the control panel RST button, the operator will obtain the level of the interrupt having occurred. By MCand RUN, the program may restart normally.

4.5. Power Failure - Automatic Restart

After a power failure and if the control panel key was in Lock position, as soon as conditions go back to normal, the program is automatically started at the beginning of the check which was being executed. In the case where the current check was in Inhibit Mode, and the interrupt could not be treated within a delay than 2 ms, the correct operation cannot be guaranteed.

4.6. Restarting Address

After a power off, the registers A8, A9 and A10 can be re-initialized by restarting at address /6EE.

5. INFORMATION CODES

The codes permit the test program running to be followed. They also permit the dialog between the test program and the operator to be carried out.

Through each error code, the following information items are given :

- operator action,
- description of the tests carried out during the check IF 15,
- error codes that can be expressed under this IF code.

IF 0000 check 00 : TLR, TSR, TL, TS : system mode
IF 0001 check 01 : ELR, ESR, EL, ES : system mode
IF 0002 check 02 : MVSU, MVUS : system mode
IF 0003 check 03 : Privileged instructions : user mode
IF 0004 check 04 : User program in successive pages
IF 0005 check 05 : User program in random pages
IF 0006 check 06 : Test bit Modify
IF 0007 check 07 : Test bit Page Error
IF 0008 check 08 : Test bit Read Only
IF 0009 check 09 : System mode : non error performing
IF 0010 check 10 : Test extended system mode
IF 0011 check 11 . MVB, MVF, Page Fault resuming
IF 0012 check 12 : ML, Page Fault resuming
IF 0013 check 13 : Test bit read only if memory 32K
IF 0014 check 14 : Test of stack overflow

IF 0000

TLR, TSR, TL, TS : system mode

- Operator action :

None.

- Check description :

+ **Execution** of "Table Load" and "Table Store" instructions.

- in system mode
- in the addressing mode : T3 to T7
- with various patterns of the segment table :
  - /0000
  - /5555
  - /AAAA
  - shift table : 1, 2, 4, 8 ...

+ **Verifications** .

- CR unchanged after each instruction
- Correct transmission through the scratchpad : comparison between the emitter buffer and the receiver buffer after the execution of Table Load and Table Store instructions.

- Possible errors

ER 0001
ER 0002
ER 0003
ER 0004
ER 0005
ER 0006
ER 0007
ER 0008
ER 0009
ER 0010
ER 0011
ER 0012

IF 0001

ELR, ESR, EL, ES : system mode

- Operator action :

None.

- Check description :

The check 01 is executed in system mode.

1)**execution** of Extended Load and Extended Store instructions :

+ in addressing mode T3 to T7
+ through a segment table addressing the memory 16K to 32K
+ with various values of the loaded and stored data : 0, /5555, /8888, /FFFF

2)**Verifications** :

+ correct value of the CR - successively positionned to : 0, < 0, > 0 by Entended Load instructions,
  - unchanged by Extended Store instructions.

- Possible errors :

ER 0013
ER 0014
" "
" "
" "
" "
ER 0041

IF 0002

MVSU, MVUS : system mode

- Operator action :

None.

- Check description :

The check 02 is executed in system mode.

1. **MVSU test :**

- Execution of a "Move Table from System to user" of one word and of two K words through a segment table addressing the memory 16K to 32K.

2. **MVUS test :**

- Execution of a Move Table from user to system of one word and two K words through a segment table addressing the memory 16K to 32K.

- Verifications after execution :
  + CR unchanged
  + A1 and A2 point on the first address after the buffer
  + R2 = 3 is set to zero.

- Possible errors :

ER 0042
ER 0043
" "
" "
" "
" "
ER 0057

IF 0003

Privileged instructions : user mode

- Operator action :

None.

- Check description :

The privileged instructions TLR, TSR, ELR, ESR, TL, TS, EL, ES, MVSU (withR2 = 15) and MVUS (with R2 = 15) are programmed successively in user mode. It is verified each instruction is trapped in user mode.

- Possible errors :

ER 0058
ER 0059
" "
" "
" "
" "
ER 0069

IF 0004

User program in successive pages

- Operator action :

None.

- Check description :

The check 04 tests the correct execution in user mode of a "user program" stored in successive pages. It consists of :

- loading in system mode of a segment table addressing the memory from 16K to 32K words.
- storing the user program in system mode.

```
16K          ADK          A5,1
 "            "
 "            "
 "            "
32KW-2       LKM
32K
```

- program execution in user mode through the segment table translation.
- return to system mode by LKM for error checking.
- test of the result in A5.

- Possible errors :

ER 0070

IF 0005

User program in random pages

- Operator action :

None.

- Check description :

The check 05 verifies the correct execution in user mode of a "user program" stored in random pages. It consists of 4 parts :

ADS (label C05 ADS)
CMP (label C05 CMP)
EA
CONST

situated in various pages :

Logical addresses	Physical addresses	User program
/0C00 : /0FFE	16K : 18K (/2000)	ADS
/1000 : /13FE	20K : 22K (/2800)	CMP
/2C00 : /2FFE	18K : 20K (/2400)	EA
/3000 : /33FE	22K : 24K (/2C00)	CONST
/4C00 : /4FFE	20K : 22K (/2800)	ADS
/5000 : /53FE	24K : 26K (/3000)	CMP
/6C00 : /6FFE	22K : 24K (/2C00)	EA
/7000 : /73FE	26K : 28K (/3400)	CONST
/8C00 : /8FFE	24K : 26K (/3000)	ADS
/9000 : /93FE	28K : 30K (/3800)	CMP
/AC00 : /AFFE	26K : 28K (/3400)	EA
/B000 : /B3FE	30K : 32K (/3C00)	CONST
/CC00 : /CFFE	28K : 30K (/3800)	ADS
/D000 : /DC00	16K : 18K (/2000)	CMP
/EC00 : /EFFE	30K : 32K (/3C00)	EA
/F000 : /F3FE	18K : 20K (/2400)	CONST

User program execution : . addition (programmed in C05ADS) of the A7 register contents with the memory effective address EA contents.
. result comparison (programmed in C05CMP) with a memory buffer CONST (correct result).

- Possible error :

ER 0071

IF 0006

Test bit "Modify"

- Operator action :

  None.

- Check description :

  - **Execution** , in user mode, through a segment table with bit Modify initialized to 0 of clear Memory of one memory case in each page.

  - **Verification** of the bit Modify (Bit 08) set to 1 in each word of the segment table.

- Possible error :

  ER 0072

IF 0007

Test page error

- Operator action :

None.

- Check description :

- **Execution :**
- Test 07A : Attempt to execute in user mode an instruction in each page protected by the bit Page Error.

- Test 07B : Attempt to execute, in user mode, a Clear Memory in a memory case of each page protected by the bit Page Error.
Each time a Page Fault is generated in the routine (P a f 07B :
1) it is verified that + The Clear Memory was not executed :

bit Modify = 0 in the appropriate word of the segment table.
+ the stack corresponding to the Page Faut is correctly loaded.

2) The segment table is modified to suppress the Page Error bit in the appropriate word of the segment table.

3) The Clear Memory is effectively executed.

- Test 07C : In user mode :
- execution of one short instruction located inan unprotected page,
- attempt to execute an instruction in a protected page to verify the corresponding P' contents in the page fault stack.

- **Verifications :**
Each time a Page Fault is generated, the page fault stack is verified :

+ first word : P' is the program counter contents saved at the beginning of each instruction.

+ third word : S' is the logical page address which generated Page Fault.

- Possible errors :

ER 0073
" "
" "
ER 0077
ER 0121
ER 0122
ER 0124
ER 0125
ER 0126

IF 0008

Test Read Only for 32K memory
(for 16K memory use check 13)

- Operator action :

None.

- Check description :

- **Test** of the execution of a Clear Memory in one memory case of each page : + in user mode,
+ through a segment table (16K to 32K words) with the bit Modify initialized to 0 and the bit Read Only initialized to 1 in each word of the segment table.

- **Verifications** : Each time a Page Fault generated :

+ the page fault stack is verified :
  - first word : P' is the program counter contents saved at the beginning of each instruction,
  - third word : S' is the logical page address which generated the page fault.

+ it is verified that the Clear Memory was not executed : bit Modify = 0 in the appropriate segment table word.

- Possible error :

ER 0078
ER 0079
ER 0080
ER 0123

IF 0009

System Mode : nonerror performing

- Operator action :

None.

- Check description :

- **Execution** in system mode of an Extended Load and an Extended Store instructions through a segment table (addressing the memory 16K to 32K words) with the bits Page Error and Read Only initialized to 1 and the bit Modify initialized to 0 in each segment table word.

- **Verifications** :

  - the instructions are correctly executed ;
  - there is no Page Fault ;
  - the bit Modify is set to 1 in the appropriate segment table word after an Extended Store execution.

- Possible error :

ER 0082
ER 0083
ER 0084

IF 0010

Extended System Mode

- Operator action :

None.

- Check description :

. It tests the correct execution in extended system mode of a program.

. It also executes in extended system mode the MVUS, MVSU instruction to check that in extended mode MVSU = MVF, MVUS = MVB.

- **Verifications** :

- the instructions are correctly executed in extended system mode,
- the memory protection mechanism is inhibited
- page fault not generated in extended system mode
- bit modify is positionned if CFU is CP1BF
- there is translation in extended system mode.

- Possible error :

ER 0126 to ER 0135

IF 0011

MVB, MVF, Page Fault resuming

- Operator action :

None.

- Check description :

Test or the intructions Move Table Backward (MVB) and Move Table Forward (MVF) broken in user mode by Page Fault (either reading the source buffer, either writting the destination buffer).

The check 11 consists of :

- Storage in system mode of : +a source buffer (/B000 to /F000 : user program with /1FFF instructions ADK A4,1 and LKM to return to system mode),
  +a Move Table instruction in /4FFE.

- **Execution** in user mode of the Move Table instruction. Each time a Page Fault is generated, the Move Table goes on after modification of the appropriate segment table word. After the complete execution of the Move Table instruction, the user program (contents of the buffer moved in the physical address /5000 to /9000) is executed.

- **Verifications** in system mode after execution :

+ A4 register : result controlling the correctness of the Move execution.

+ The registers A1,A2, and R2 = 3 defining the transfer parameters for the Move Table :
  - A3 = 0,
  - after MVF, A1 and A2 contains their initial value,
  - after MVB, A1 and A2 points on the first addresses after the buffer.

- Possible error :

ER 0095
"
ER 0109
ER 0118

IF 0012

ML, Page Fault resuming

- Operator action :

None.

- Check description :

The check 12 tests the correct execution of the Mutiple Load instruction broken by Page Fault.
It consists of :

- The **execution** of the Multiple Load,
  + in user mode,
  + in addressing mode T3, T5, T7,
  + with the registers A1 to A7,
  + in different memory cases from 16K to 32K words.

The buffers are stored in successive addresses of two different pages.

Each time there is a Page Fault, the corresponding segment table word is modified and the ML instruction is resumed.

- **Verifications** : + CR = 1 (A1 being $>$ 0)
  + A1 to A7, contents are correct.

- Possible error :

ER 0110
"
"
ER 0116
ER 0120

IF 0013 See IF 0008.

IF 0014

Test of stack overflow

- Operator action .

None.

- Check description :

. Test or the generation of interrupt stak overflow on the following instructions : ST format T3B,
MSR rormat T3B,
CF formats T2,T1, T4.

A15 has been initialized at /7E.

- **Verifications** : Each time an interrupt stack overflow is generated a flag is set and checked in the main routine.

- Possible error :

ER 0140
ER 0142
ER 0144 to 0146

IF 0016

"One shot" function selected

- Operator action :

The program being stopped in /5E0 at the end of each check, depress the RUN button to execute the subsequent test.

6. ERROR CODES

ER 0001 : (Check 00) CR modified by a TLR execution.
(A2) = pattern loaded in the segment table.
(0, /FFFF, /5555, or /AAAA).

ER 0002 : (Check 00) CR modified by a TSR execution.
(A2) = pattern loaded in the segment table.
(0, /FFFF, /5555, or /AAAA).

ER 0003 : (Check 00) pattern error or MMU not present.
After a TLR and a TLS executions, the result is not correct in the receiver buffer SCRTCH.
(A2) correct pattern (0, /FFFF, /5555, or /AAAA).
(A3) number of the segment table word with incorrect contents.
2

ER 0004 : (Check 00) CR modified by TL executed in addressing mode T4.

ER 0005 : (Check 00) CR modified by TS execution in addressing mode T4.

ER 0006 : (Check 00) pattern error.
After a TL and a TS executions, the result is not correct in the receiver buffer SCRTCH.
(A2) = correct word (word 0 + A6)
(A3) = (A7) : incorrect word (SCRTCH + A6)
(A4) = 5, 6 or 7 : addressing mode (T5, T6 or T7).

ER 0007 to ER 0012 : (Check 00)CR modified.
ER 0007 : by TL executed in addressing mode T5
ER 0008 : by TS executed in addressing mode T5
ER 0009 : by TL executed in addressing mode T6
ER 0010 : by TS executed in addressing mode T6
ER 0011 : by TL executed in addressing mode T7
ER 0012 : by TS executed in addressing mode T7.

ER 0013 : (Check 01) CR not correctly positionned by the ELR execution :
correct value = 2 (loaded pattern being : A2 = /8888).

ER 0014 : (Check 01) pattern error : Data not correctly loaded in A3 by the ELR.
(A2) = /8888 : correct value
(A3) = incorrect value loaded by the ELR.

ER 0015 : (Check 01) CR not correctly positionned by the ELR execution.
correct value = 0 (loaded pattern being 0).

ER 0016 : (Check 01) pattern error : the data loaded in A3 by the ELR is not correct :
(A2) = 0 correct value
(A3) = incorrect value loaded by the ELR.

ER 0017 : (Check 01) CR error :
CR is not positionned to "1" by an ELR execution :
(A2) = /5555 (constant loaded during ELR).
ER 0017 is also indicated when the system memory is less than 32K.

ER 0018 : (Check 01) pattern error :
The data is not correctly loaded in A3 by the ELR.
(A2) = /5555 correct value
(A3) = incorrect value loaded by the ELR.

ER 0019 : (Check 01) CR error :
CR is not positionned to "2" by an EL execution (addressing mode T4).
(A2) = /8888 (constant loaded during EL).

ER 0020 : (Check 01) pattern error :
The data is not correctly loaded in A3 by the EL (T4)
(A2) = /8888 correct value
(A3) = incorrect value loaded during the EL execution.

ER 0021 : (Check 01) pattern error :
The data is not correctly loaded by the EL (T5)
(A2) = /5555 correct value
(A3) = incorrect value loaded during the EL execution.

ER 0022 : (Check 01) Pattern error :
The data is not correctly loaded by the EL (T5)
(A2) = /5555 (correct value)
(A3) = incorrect value loaded during the EL execution.

ER 0023 : (Check 01) CR error :
CR not positionned to "0" by EL execution (T6) while loaded constant = 0.

ER 0024 : (Check 01) pattern error :
The data is not correctly loaded by the EL (T6).
(A2) = 0 correct value
(A3) = incorrect value loaded during the EL execution.

ER 0025 : (Check 01) CR error :
CR not positionned to "0" by EL execution (T7) while the loaded constant = 0.

ER 0026 : (Check 01) pattern error :
The data is not correctly loaded by the EL (T7) :
(A2) = 0 (correct value)
(A3) = incorrect value loaded during the EL execution.

ER 0027 : (Check 01) CR error :
CR not positionned to "1" by the execution (T7) while the loaded constant = /5555.

ER 0028 : (Check 01) pattern error :
Data loaded by the EL (T7) is incorrect.
(A2) = /5555 (correct value)
(A3) = incorrect value loaded during the EL.

ER 0029 : (Check 01) CR error :
CR modified by the ESR execution.
It was positionned to 0 before the instruction.

ER 0030 : (Check 01) pattern error :
The data is not correctly stored by the ESR.
(A2) = /FFFF correct value to store.
(A3) = incorrect value stored by ESR in the physical address /97FE.

ER 0031 : (Check 01) CR modified by the ESR execution :
It was positionned to 1 before the instruction.

ER 0032 : (Check 01) pattern error :
The data is not correctly stored by the ESR.
(A2) = 0 correct value to store
(A3) = incorrect value stored by ESR in the physical address /B7FE.

ER 0033 : (Check 01) CR modified by the ESR
It was positionned to 2 before the instruction execution.

ER 0034 : (Check 01) pattern error.
The data is not correctly stored by the ESR.
(A2) = /5555 : correct value to store.
(A3) = incorrect value stored by ESR in the physical address /F7FE.

ER 0035 : (Check 01) CR modified by the ES (T4).
It was positionned to 1 before the ES execution.

ER 0036 - 37 - 38 : (Check 01) pattern error :
The data is not correctly stored by the ES.

	A2 correct value to store	A3 = incorrect value stored by ES in the physical addresses	Addressing mode Tx
ER 0036	/8888	/E7FE	4
ER 0037	/FFFF	/BFFE	5
ER 0038	/5555	/9FFE	6

ER 0039 : (Check 01) CR modified by the ES (T7) execution :
It was positionned to 2 (LDKL A7,/87FE) before the ES execution.

ER 0040 - 41 : (Check 01) pattern error :
The data is not correctly stored by the ES (T7).

			ER 0040	ER 0041
(A2)	=	corect value to store :	0	/8888
(A3)	=	incorrect value stored by ES in the physical addresses :	/C7FE	/FFFE

ER 0042 : (Check 02) CR modified by the MVSU (one word).
It was positionned to 1 (LDKL A2,/4000) before the MVSU execution.

ER 0043 : (Check 02) data error during MVSU (one word).
The data moved by MVSU from the physical address /7000 to the physical address /A000 is incorrect.
(A2) = 0 correct value to move
(A3) = incorrect value moved by MVSU.

ER 0044 : (Check 02) : (A1) incorrect after MVSU execution.
(A1) should have the value /7000 : starting physical address of the emitter buffer.

ER 0045 : (Check 02) : (A2) incorrect after MVSU execution.
(A2) should have the value /4000 : starting logical address of the receiver buffer.

ER 0046 - 47 : (Check 02) : Incorrect transfer during MVSU execution :
After MVSU execution, (A3) should have the value "0".
Before the MVSU execution, (A3) was loaded by the double of the number of transfers :
either (A3) = 2 (ER 0046) transfer of 1 word
either (A3) = /1000 (ER 0047) transfer of 2K words.

Result :

(A2) = correct value = 0
(A3) = incorrect value : residual length of transfers.

ER 0048 : (Check 02) : Data error during MVSU (2K words) :
The data moved by MVSU from the physical address /7000 to the physical address /A000 is incorrect.
(A2) = correct value
(A3) = incorrect value moved.

ER 0049 : (Check 02) CR modified by the MVUS (one word) execution :
It was positionned to 1 (LDKL A2, /7000) before the MVUS execution.

ER 0050 : (Check 02) : (A1) incorrect after MVUS (one word) execution :
(A1) should have the value /4002 (first logical address after the emitter buffer).

ER 0051 : (Check 02) :
(A1) incorrect after MVUS (one word) execution :
(A2) should have the value /7002 (first physical address after the receiver buffer).

ER 0052 : (Check 02) : incorrect transfer during MVUS (one word) :
After MVUS execution, (A3) should have the value "0".
Before the MVUS execution, (A3) was loaded by the double of the number of transfers : 2.

(A2) = correct value = 0
(A3) = incorrect value = residual length of transfers.

ER 0053 : (Check 02) : Data error during MVUS (one word) :
The data moved by MVUS from the physical address /A000 to the physical address /7000 is incorrect.

(A2) = correct value
(A3) = incorrect value moved.

ER 0054 : (Check 02) : (A1) incorrect after MVUS (2K words) execution :
(A1) should have the value /5000 (first logical address after the receiver buffer).

ER 0055 : (Check 02) : (A2) incorrect after MVUS (2K words) execution :
(A2) should have the value /8000 (first physical address after the receiver buffer).

ER 0056 : (Check 02) : incorrect transfer during MVUS (1K word) :
After MVUS execution, (A3) should have the value "0".
Before the MVUS execution, (A3) was loaded by the double of the number of transfers : /1000.

(A2) = correct value
(A3) = incorrect value : residual length of transfers.

ER 0057 : (Check 02) : Data error during MVUS (1K word) :
The data moved by MVUS from the physical address /A000 to the physical address /7000 is incorrect.

(A2) = correct value
(A3) = incorrect value moved.

ER 0058 to ER 0067 : (Check 03) : privileged instructions are not trapped in user mode.
Verify (A7) : (A7) is set to 1 before the privileged instruction execution and reset to 0 during the trap routine.

ER 0058 : error TLR
ER 0059 : error TSR
ER 0060 : error ELR
ER 0061 : error ESR
ER 0062 : error TL
ER 0063 : error TS
ER 0064 : error EL
ER 0065 : error ES
ER 0066 : error MVSU with R2 = 15
ER 0067 : error MVUS with R2 = 15

ER 0068 - ER 0069 : (Check 03) : trap error
The program goes to the trap routine.
- See the check number in CCHEC4 (/EA).
The bit that is "1" indicates the running check.
- Verify the trap routine address was correctly loaded.

ER 0070 : (Check 04) : incorrect result of the user program execution.
The user program consists of the execution of /3FFF instructions ADK A5, 1 stored in successive pages in the physical addresses from /8000 to /FFFC. The result should have the value /3FFF.

(A5) = incorrect result.

ER 0071 : (Check 05) : error during the execution of a user program stored in random pages.
The user program consists of the addition of A7 (programmed in C05ADS) with the memory effective address EA and comparison (by C05CMP) with a memory buffer CONST.

Verify ° the good storage of C05ADS, C05CMP, EA and CONST ;

° the correct passage between the different buffers;

° for a value of the flage A4, the result of C05ADS addition in EA must have the value of the appropriate memory case of CONST.

(A2) = reference CONST (A3 should have be that correct value).
(A3) = incorrect result (stored in EA) of the last addition.
(A4) = flag incremented of "34" from an addition to the next to execute a new loop.
(C05ADS, C05CMP).

ER 0072 : (Check 06) : bit modify incorrect.
The bit modify is not set to "1" when a Clear Memory is executed in the appropriate page.

Verify ° the segment table ;
° the cleared memory cases.

$\frac{(A2)}{2}$ = word number of the segment table in which the bit modify is not correct.

ER 0073 to ER 0076 : (Check 07) : stack error during Page Fault :
The Page Fault stack verification consists of :

° Verification of the first word P' : program counter contents saved at the beginning of each instruction,
° Verification of the third word S' : logical page address which generated the Page Fault.

ER 0073 : S' error during Test 07A
ER 0074 : P' error during Test 07A
ER 0075 : S' error during TST 07B
ER 0076 : P' error during TST 07B.

(A2) = correct logical address forER 0073 and 75.
$\frac{(A2)}{2}$ = protected segment number for ER 0074 and 76
(A3) = incorrect value.

ER 0077 : (Check 07) : a protected page is modified in user mode :
The instruction which generated Page Fault is executed in user mode.
° Verify the Clear Memory (user program C678) execution.
$\frac{(A2)}{2}$ = protected segment number.

ER 0078 : (Check 08) : S' (third word of the page fault stack)
(Check 13)
Each page is protected by the bit Read Only.
A Page Fault occured for each Clear Memory of C678 user program.
$\frac{(A2)}{2}$ = protected segment number
(A3) = incorrect S'.
If during check 13 $\frac{A2}{2}$ = 0018 the test program is destroyed.

ER 0079 : (Check 08) : P' (first word of the page fault stack)
(Check 13) error.
Each page is protected by the bit Read Only.
A Page Fault occured for each Clear Memory of C678 user program.
(A2) = reference value (P')
(A3) = incorrect P'.

ER 0080 : (Check 08) : bit modify error.
(Check 13)
It is set to "1" during the testing of Clear Memory execution in page protected by the bit Read Only.
$\frac{(A2)}{2}$ = number of the segment which generates a Page Fault.

ER 0081 : (Check 09) : Page Fault occurs in system mode
(A2) = /9A : Page Fault occurs during the Extended Load execution.
(A2) = /9B : Page Fault occurs during the Extended Store execution.
Verify the system is in system mode.

ER 0082 : (Check 09) : Extended Load not correctly executed :
(A2) = /9A : correct value to load in A3 by EL.
(A3) = incorrect value loaded during EL execution.

ER 0083 : (Check 09) : Extended Store not correctly executed :
(A2) = /9B correct value stored by ES.
(A3) = incorrect value stored by ES.

ER 0084 : (Check 09) : bit modify error after Extended Store execution in a protected page :
Verify in the word 1 of the segment table SCRTCH that :
° the bit 6 (Page Error) = 1
° the bit 7 (Read Only) = 1

ER 0095 : (Check 11) :
(A1) incorrect after MVB execution broken by Page Fault in reading the source buffer.
Verify the (A1) evolution. During Page Fault, A1 points on the first location on which the execution must be resumed.
After MVB execution, (A1) should have the value of the first address after the emitting buffer (logical address : /9000).
(A1) = incorrect value.

ER 0096 : (Check 11) :
(A2) incorrect after MVB execution broken by page Fault in reading the source buffer.
Verify the (A2) evolution. During Page Fault, A2 points on the first location on which the execution must be resumed.
After MVB execution, (A2) should have the value of the first address after the receiving buffer (logical address /5000).
(A2) = incorrect value

ER 0097 : (Check 11) : Residual length is not 0 in (A3) after MVB execution with Page Fault in reading the source buffer A3 was initialized to /4000 (double of the number of transfers).
Verify the evolution of (A3). When a Page Fault occurs, the contents of A3 is updated with the residual length of transfers.
After MOVE execution, (A3) = incorrect value of the length of transfers.

ER 0098 : (Check 11) . incorrect result of the user program execution after MVB with Page Fault in writting the destination buffer :
The user program consists of the execution of /1FFF instructions ADK A4, 1 stored from 16K to 24K words by the MOVE instruction.
(A4) should have the value /1FFF
(A4) = incorrect result.
Verify the MOVE execution : before that instruction, the user program is loaded from 24K to 32K words and (A4) = 0.

ER 0099 : (Check 11) :
(A1) incorrect after MVB execution with Page Fault in writting the destination buffer.
See ER 0095.

ER 0100 : (Check 11) :
(A2) incorrect after MVB execution with Page Fault in writting the destination buffer.
See ER 0096.

ER 0101 : (Check 11) : residual length is not 0 in (A3) after MVB execution with Page Fault in writting the destination buffer.
See ER 0097.

ER 0102 : (Check 11) : incorrect result of the moved user program execution after MVB with Page Fault in writting the destination buffer.
See ER 0098.

ER 0103 : (Check 11) :
(A1) incorrect after MVF execution with Page Fault in reading the source buffer :
When Page Fault occurs and after the MVF execution,
(A1) should have the value of the starting address of the emitting buffer : (logical address : /5000).
(A1) = incorrect value.

ER 0104 : (Check 11) :
(A2) incorrect after MVF execution with Page Fault in reading the source buffer :
When Page Fault occurs and after the MVF execution,
(A2) should have the value of the starting address of the receiving buffer : (logical address : /1000).
(A2) = incorrect value.

ER 0105 : (Check 11) : residual length is not 0 in (A3) after MVF execution with Page Fault in reading the source buffer :
See ER 0097.

ER 0106 : (Check 11) : incorrect result of the moved user program execution after MVF with Page Fault in reading the source buffer :
See ER 0098.

ER 0107 : (Check 11) :
(A1) incorrect after MVF execution with Page Fault in writting the destination buffer :
See ER 0103.

ER 0108 : (Check 11) :
(A2) incorrect after MVF execution with Page Fault in writting the destination buffer :
See ER 0104.

ER 0109 : (Check 11) : incorrect result of the moved user program execution after MVF with Page Fault in writting the destination buffer :
See ER 0098.

ER 0110 : (Check 12) : CR error after Multiple Load execution :
The CR should have the value "1" |(A1) being /1FFE|
$\frac{(A13)-4}{2}$ = segment number in which the Page Fault occurs during the last Multiple Load execution.

ER 0111 to ER 0116 : (Check 12) : Loaded register incorrect after Multiple Load executed in user mode with Page Fault :

ER 0111 : (A1) is not /1111	(incorrect value of	A1	—	A11)
ER 0112 : (A2) is not /2222	"	"	"	"
ER 0113 : (A3) is not /3333	"	"	"	"
ER 0114 : (A4) is not /4444	"	"	"	"
ER 0115 : (A5) is not /5555	"	"	"	"
ER 0116 : (A6) is not /6666	"	"	"	"

$\frac{(A13)-4}{2}$ = segment number in which the Page Fault occurs during the last Multiple Load execution.
Verify that the Multiple Load is integrally executed after a Page Fault.

ER 0118 : (Check 11) : residual length is not 0 in (A3) after MVF execution with Page Fault in writting the destination buffer :
See ER 0097.

ER 0120 : (Check 12) :
(A7) incorrect after a Multiple Load execution in user mode with Page Fault :
(A7) should have the value /7777.
Verify that the Multiple Load is executed from the beginning after a Page Fault.

(A7) = incorrect value
$\frac{(A3)-4}{2}$ = segment number in which the Page Fault occurs during the last Multiple Load execution.

ER 0121 : (Check 07) : incorrect test page error during Test 07B. If (A2) is not "32", the second part of the check 07 is not correctly executed. Normally the segment table is entirely modified without bit page error after the execution of the user program C678.

Verify
- the segment table : SCRTCH
- the memories cleared by the Clear Memory of the user program C678
- if the PAF07B routine is correctly executed.

$\frac{(A2)-2}{2}$ = protected segment number.

ER 0122 : (Check 07) : incorrect test Page Error during Test 07A. During the first part of the check 07 (Test 07A), each page is protected. The user program consists of instructions LKM stored in each page and they have not to be executed.

Verify
- the bit PE in each word of the segment table
- the execution of the PAF07A routine.

ER 0123 : (Check 08-13) : incorrect test Read Only. If (A2) is not "32", the check 08 is not correctly executed. Normally, (A2) is incremented of 2 for each Page Fault.

Verify
- the segment table has the bit Read Only = 1 in each word
- that the instructions Clear Memory of the user program C678 were not executed
- if the PAF08 routine is correctly executed.

$\frac{(A2)-2}{2}$ = number of the segment in which the Page Fault occurs.

ER 0124 : (Check 07) : S' error in the Page Fault stack during Test 07C :

(A3) = incorrect value of S'
(A3) should have the value /200 (bits 4 to 7 corresponds to the logical address which generated Page Fault).

ER 0125 : (Check 07) : P' error in the Page Fault stack during Test 07C.

(A3) = incorrect value of P'
(A3) should have the value /2000 (program counter value saved at the beginning of the last instruction).

ER 0126 : (Check 07) : incorrect test page error during Test 07C. In Test 07C, the LKM instruction stored in a protected page has not to be executed.

Verify
- the bit Page Error in the segment 2 after a Table Store execution
- the execution of the PAF07C routine.

ER 0127 : (Check 10) no translation in extended system mode. To see if it is during MVUS, MVSU test, check A7 = 7100.
Verify if the TL has been correctly executed with the table TESM1.

ER 0128 : (Check 10) write protect is active in extended system mode. A Store at physical address 8B30 is impossible.
Verify if /8B30 is equal to 1.

ER 0129 : (Check 10) The bit modify has not been positionned when a store in memory has been executed.
Verify that TL and TS instructions work correctly.

ER 0130 : (Check 10) MVSU instruction has not been correctly executed in extended system mode.
Verify physical address /A000.

/A000	AAAA
/A002	"
/A004	"
/A006	"

ER 0131 : (Check 10) MVUS instruction has not been correctly executed in extended system mode.
Verify physical address /9000.

/9000	AAAA
/9002	"
/9004	"
/9006	"

ER 0132 to ER 0135 : (Check 10) : an error has occured during execution of various instruction when in extended system mode.

ER 0140 : (Check 14) an interrupt stack overflow has not been generated on ST instruction.
Verify physical address /1768 = 1.

ER 0141 : (Check 10) a trap page fault has been generated in extended system mode.
Verify that the program was in extended system mode.

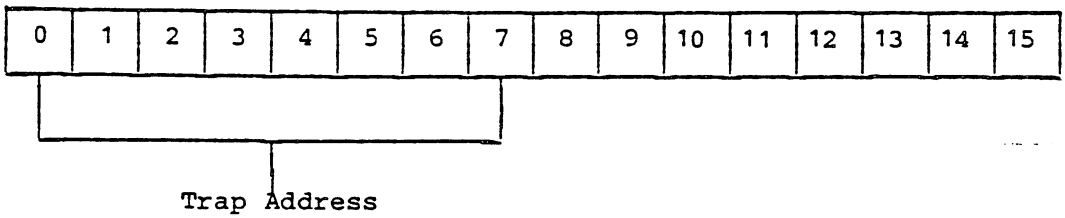
ER 0142 to ER 0146 : (Check 14) an interrupt stack overflow has not been generated on following instruqctions :

ER 0142	MSR	T3B
ER 0144	CF	T2
ER 0145	CFR	T1
ER 0146	CFI	T4

7. PROGRAM MODIFICATION

After loading, the program stops in /700. For a non standard initialization (see § 4.1.), the operator should load A8, A9 and A10 to introduce the parameters relating to MMU operation and program processing.

7.1. Parameters relating to MMU operation : A8 register

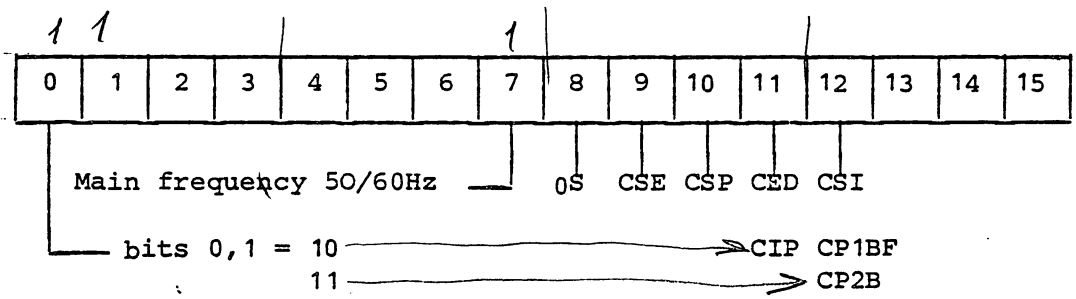


7.2. Parameters related to program processing :

- "one shoot" OS command
- "clear stop on error" CSE command
- "clear stop on program" CSP command
- "clear edition" CED command
- "clear stop information" CSI comand
- "checks" to be executed command

7.2.1. A9 register :

In standard operation (ss § 4.1.), the bits 8 to 12 are initialized at 0.



- "one shoot" (OS) :

  Bit 08 = 1 : Stop in /5E0 (hexa) after the running of each check, before going to the next one.
  Depress the run button to carry on.

- "clear stop on error" (CSE) :

  Bit 09 = 1 : Suppress the in /5F0 in case of error.

- Main frequency :

  Bit 07 = 0 : 50Hz
  = 1 : 60Hz

  In case of 50Hz, it is possible to set bit 7 to "1" that will increase by 20 % all delays. This permits to determine if problem is due to a critical timing or real fault.

- "Clear stop on program" (CPS) :

  Bit 10 = 1 : Suppress the stop in "700" (hexa) after the program has been executed (loop again on test).

- "Clear Edition" (CED) :

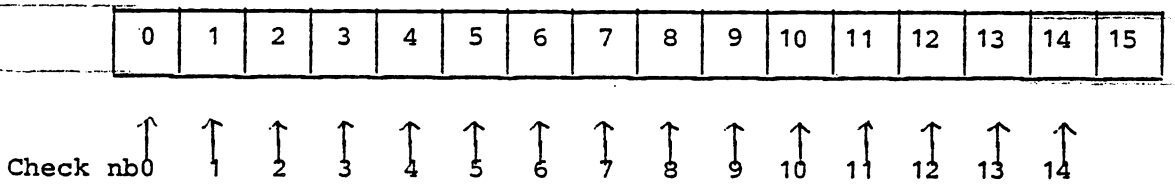
  Bit 11 = 1 : Suppress the printing out of error and information Messages :

  CSI at 0 : Stop in /5EO : information Code in A1
  CSE at 0 : Stop in /5F0 : error code in A1.

- "Clear stop on information" (CSI) :

  Bit 12 = 1 : Supress the stop in /5E0 (hexa).

7.2.2. A10 register :



(running in normal operation)

In non standard operation,load A10 before starting the program.
- checks to be run : bits corresponding of A10 initialized at 1.
- checks absents : bits corresponding of A10 initialized at 0.