P854
P871
P874
P876

FPPE TEST PROGRAM

E F P P 1

Applicable to Floating Point Processor

Procedure 12NC : 5111 991 12233

Concerned PV :
PV250-156 (tape)
PV450-156 (floppy)

EFFP1 includes :
Procedure
+ Listing 12NC : 5111 991 12243
+ Program 12NC : 5111 199 59933

0. TABLE OF CONTENTS

1. GENERAL

2. SYSTEM CONFIGURATION NECESSARY TO PERFORM THE TEST

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 - Automatic Restart

5. INFORMATION CODES

6. ERROR CODES

7. PROGRAM MODIFICATION

7.1. Parameters related to program processing

8. FLOW CHARTS

1. GENERAL

EFPP1 - is a stand alone program
- tests the Floating Point Processor in 12 checks.
- a service routine is provided in the check 12.

2. SYSTEM CONFIGURATION REQUIRED

- One paper tape reader or one floppy drive to load the program.
- One C.P.U. P857EB /CP18F/CP2B and its back panel.
- One TRPZ circuit board equipped with PTR controller, or a F1MZ controller with its floppy drive.
- One FPPE (FPE1, FPE2) and its back panel.
- A 32K memory is sufficient.

REMARK :

The operator's work will be made easier if the configuration possesses a peripheral control (of P817 type for instance).

3. LOADING PROCEDURE

Using the following procedure load the Test Program into the system:

- Load the Test Program medium in the appropriate input peripheral
- Using the Operator's Panel select the appropriate values to load the program from the peripheral using the IPL routine
- Push the MC button
- Push 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 initializations

- Program execution initialized - without "one shot" with " stop on error", "STOP on END", and "Edition"
  . to execute the check 00 to 11
- The program is initialised for a CP2B CPU (A9=/8000).
  These parameters are automatically loaded in the program.
  To change these initializations, refer to chapter 7.1 .
  To restart with the standard initializations, load the address / 06EE.

4.2 Program execution

- After loading the program (stop in/700), depress the RUN button.
- The program automatically executes the checks 00 to 11.

The program execution consits 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 occurring 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 te 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 7.1.

NOTE : At any time it is possible to return 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 /5FC 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

An interrupt is considered 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 or the interrupt having occurred. By MC and 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. If the current check was in Inhibit Mode, and Interrupt could not be treated within a delay <than 2 ms, correct operation cannot be guaranteed.

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 $\leqslant$ 15.
- Error codes that can be expressed under this IF code.

IF 0000 check 00 FPCK00 : FPPE present
IF 0001 check 01 FPCK01 : Data path
IF 0002 check 02 FPCK02 : FLD check
IF 0003 check 03 FPCK03 : FST check
IF 0004 check 04 FPCK04 : FFL check
IF 0005 check 05 FPCK05 : FFX check
IF 0006 check 06 FPCK06 : FAD check
IF 0007 check 07 FPCK07 : FSU check
IF 0008 check 08 FPCK08 : FMU check
IF 0009 check 09 FPCK09 : FDV check
IF 0010 check 10 FPCK10 : ROM address control check
IF 0011 check 11 FPCK11 : Abnormal conditions
IF 0012 check 12 FPCK12 : Service Routine

IF 000 FPPE presence.

Check description :

. Initialization of trap /7E (FPPE absence)
. FFL

. Verifications :

If FPPE is present the FFL instruction is executed.

If not, the instruction is trapped (Trap /7E).

. Possible error :

ER C001

IF 0001 Test of data path

- Check description : The test consists of two parts :
  - various floating point data are loaded in the FPP accumulator by FLD and copied into memory by FST.
  - a double precision integer is transferred to the FPP and converted in a floating point data with exponent = 30 by FFL. The floating point data is copied into memory by FST.

- Verifications :

  First part : The three floating point words are correctly transferred :
  result stored into memory = initial point data

  Second part: During FFL, the exponent value "30" is correctly loaded.

- Possible errors

  ER 0010
  ER 0012

IF 0002 Test of Floating Load instruction

- Check description : It consists of 2 parts :
  - execution of the Floating Load instruction in addressing mode T4, T5, T6 and T7 : for each mode, after an FSTR execution, the result is verified.
  - execution of the Floating load instruction with a floating point operand zero to verify CR positioning to 0
    > 0 to verify CR positioning to 1
    < 0 to verify CR positioning to 2

- Possible errors

ER 0023
"
"
ER 0029

IF 0003 Test of the Floating store instruction

- Check description : It consists of 2 parts :
  - execution of a Floating Store instruction with a floating point operand zero to verify CR positioning to 0
    >0 to verify CR positioning to 1
    <0 to verify CR positioning to 2
  - execution of a Floating Store instruction in addressing mode T4, T5, T6, T7 : for each mode, the result in memory is compared with the initial floating data loaded by an FLD.

- Possible errors

  ER 0031
  "
  "
  ER 0037

IF 0004 Test of the Floating Float instruction

- Check description : It consists of 2 parts :

  . execution of a Floating Float instruction with a double precision integer zero to verify CR positioning to 0
  > 0 to verify CR positioning to 1
  < 0 to verify CR positioning to 2

  . execution of an FSTR after an FFL to verify the correct transfer of a double precision integer to the FPP and the correct conversion in a floating point data.

1) double precision integer —FFL→ floating point data in FPP —FSTR→ floating point data sent to memory for verification
   (/0000, /0001) (/4000, /0000, /0001)
2) double precision integer FFL floating point data in FPP FSTR floating point data sent to memory for verification
   (/2000, /0000) (/4000, /0000, /1D)

- Possible errors

ER 0041
"
"
ER 0044
ER 0045

IF 0005 Test of the Floating FIX instruction

- Check description : It consists of the three following verifications :

  • the FFX execution does not destroy the FPP accumulator contents : a reference floating point data is + sent to the FPP accumulator by FLD,

  + converted

  in a double precision integer and sent into A1, A2 by FFX,

  + transferred

  from FPP accumulator to memory by FSTR to verify as the initial value.

  • Correct FFX execution :
  After FFX, the A1, A2 registers contain the correct double precision integer.

  • Correct CR positioning
  It depends on the FFX result. FFX is executed successively with an appropriate FPP floating point data to obtain a result = 0 for CR positioning to 0
  > 0 for CR positioning to 1
  < 0 for CR positioning to 2

- Possible errors :

  ER 0051
  "
  "
  ER 0057

. Check description : It is verified that :

. The operand is not destroyed during the floating point addition.

. The alignment is correct depending on the exponent :

+ operand alignment : operand exponent : /0001

FPP accumulator contents exponent = /001E

the operand is modified to obtain the exponent /001E before the addition.

+ there is no operand and FPP accumulator contents modification before the addition if their exponents are identical.

+ accumulator contents alignment :

Operand exponent : /002E

FPP accumulator contents exponent : /0010

The FPP accumulator is modified to obtain the exponent /002E before the addition.

. The floating point addition is correct in addressing mode T4 to T7 for two modes :

+ result in the FPP accumulator,

+ result in memory : in this case it is also verified that the FPP accumulator contents is not modified during the addition.

. Possible errors :

ER 0061 to ER 0064
ER 0604 to ER 0607
ER 0613 to ER 0617
ER 0623

IF 0007 Test of the Floating Point Subtraction

- Check description :
  The check 07 consists of the execution of 2 particular subtractions :
  - with operand alignment
    FPP accumulator contents exponent = /0004
    operand exponent = /FFFC
    The operand mantissa is modified (shifted right) to obtain the exponent /0004 before the subtraction.
  - with FPP accumulator contents alignment
    FPP accumulator contents exponent = /3FFC
    operand exponent = /4000
    The FPP accumulator contents is modified (shifted right) to obtain the exponent /4000 before the subtraction.

- Verification of the subtraction result.

- Possible errors :
  ER 0071
  ER 0072

IF 0008 Test of the Floating Point Multiplication

- Check description : The check 08 consists of 4 tests. Each test is the execution of one particular multiplication :

  - Test 1 : The result is the smallest mantissa in absolute value.
  - Test 2 : The result is obtained after normalization.
  - Test 3 : Normal execution of a multiplication.
  - Test 4 : The result is the greatest mantissa.

- Verification of the multiplication result in each test

- Possible errors :

  ER 0081 to ER 0084

IF 0009 Test of Floating Point Division

Check description :

The check 09 consists of 5 tests. Each test is the execution of one particular division :

- Test 1 : smallest mantissa quotient (in absolute value) : $\frac{1}{2} / -1 = -\frac{1}{2}$.
- Test 2 : $\frac{1}{2} <$ mantissa quotient < 1
- Test 3 : $1 <$ mantissa quotient < 2
- Test 4 : greatest mantissa quotient (in absolute value) $(-1)/\frac{1}{2} = -2$
- Test 5 : division with the least significant bit of the dividend.

Verification : of the division result in each test.

Possible errors :

ER 0091 to ER 0095

IF 0010 ROM next address test

Check description : It consists of 4 tests :

- FFX branch : The next address depends on the exponent and mantissa values : check for next address to be correct depending on the case :

Different possibilities	Next address	Floating Point Data	double precision integer A1	A2
Exponent < 0 or nul mantissa	RESNU (result = 0)	/0000,/0000,/0020 /0000,/0000,/0010 /0000,/0000,/001E /4000,/0000,/801C /4000,/0000,/8020 /0000,/0000,/801C /0000,/0000,/8020	0	0
exponent = 30	RESFA - no denormalization - result = mantissa loaded in the accumulator	/4000,/0000,/001E	/4000	/0000
0 ≤ exponent < 30	DENORM for denormalization	Verified in the FFX test (see check 05)		
exponent > 30 (mantissa is not 0)	OVXDV (overflow)	Verified in the check 11		

- operation branch : Some possibilities are tested on others checks. The check 10 consists of :

1. C 100 PU : a non normalized operand is used to execute
   - FAD, FMU and FDV, the FPP accumulator contents being ≠ 0.
   - FAD, FMU, FDV and FLD, the FPP accumulator contents being = 0.

   In each case, the trap activation and the "non normalized operand" CR are verified.

2. C 100 PZ :

Accumulator contents	Memory Operand	Floating Point Instruction	Rom next address	Result
/4000, /0000, /001E	0	FAD	RESFA	accumulator contents
0	0	FAD	RESNU	nul result
/4000, /0000, /001E	0	FMU	RESNU	nul result
0	0	FMU	RESNU	nul result
0	/4000, /0000, /001E	FLD	REQM	memory operand
/4000, /0000, /001E	0	FLD	RESNU	nul result
0	0	FLD	RESNU	nul result
0	0	FDV	OVXDV	status : division by 0
/4000, /0000, /001E	0	FDV	OVXDV	status : division by 0
0	/8000, /0000, /0000	FSU	REQPMM	/4000, /0000, /0001
0	/4000, /0000, /001E	FMU	RESNU	nul result
0	/4000, /0000, /001E	FDV	RESNU	nul result

The trap activation and the "division by zero" CR is verified.

- Alignment branch : For addition and subtraction, the next address depends on the exponent values :
  - accumulator contents exponent > memory operand exponent > +30 : the result is the accumulator contents (RESFA).
  - accumulator contents exponent < memory operand exponent < -30 : the result is the memory operand. (REQPMM).
  - others cases are verified in other checks.

In the operations executed in the check 10, the alignment is not realized, one operand being negligible with regard to the other.

. Result Branch :

Four next addresses are possible depending on the result :

- . correct result
- . nul mantissa

} Tested in precedent checks

- . result with overflow : it is modified by a right shift
- . unormalized result : it is modified by a left shift.

. Possible errors :

ER 0104 to ER 0136
ER 0013 to ER 0019
ER 0100
ER 0101

IF 0011 Abnormal condition test

Check description : It contains three parts :

- C 11B : "underflow" and "overflow" detection during FMU, FDV, FAD and FSU instructions - "overflow" detection during FFX instruction.
  The trap activation and the CR are checked. It is verified that the memory is not destroyed by FMUS with a trap /7E.

- C 11C : execution of FAD with memory operand M1, M2, 2^E successively implemented in the logical addresses :
  - /2000 : only the segment 2 is protected : page Fault generated on M1.
  - /1FFE : only the segment 2 is protected : Page Fault generated on M2.
  - /1FFC : only the segment 2 is protected : Page Fault generated on the exponent.

  It is verified that the FPP accumulator contents is not destroyed by the FAD with Page Fault (trap /7A).

- C 11US : see C 11C - same checks but with FST instruction.

- Possible errors :

  ER 0202 to ER 0208
  ER 0220 to ER 0227
  ER 0233 to ER 0244

Check description :

This check provides a loop on any FPPE instruction. It asks the instruction, the contents of the accumulator and operand.
("EX" = exit of check 12)
Bit 12 of A10 must be at one to execute this check.

Example :

```
CHECK 12 : LOOP ON FPP INST.
SPECIFY INSTRUCTION (LD,ST,AD,SU, MU,DV,FX,FL,EX) : AD
ACCU :5F4567348675
OPER :800000004000
LOOP IS ENTERED , STOP BY "CR" :
END OF LOOP. M1=8000, M2= 0000, E=4000, ST= 0000
SPECIFY INSTRUCTION (LD,ST,AD,SU, MU,DV,FX,FL,EX) : EX
END OF CHECK 12
```

6. ERROR CODES

ER 0001 : (Check 0) Trap error
FPP is absent or signal FPP ABS is not wired.

ER 0010 : (Check 1) Data Not Correctly Transferred.
. Check the data path during FLDR and FSTR instructions.

[A5] , [A6] , [A7] : reference floating point data (operand during FLD)

[A2] , [A3] , [A4] : result after FLD and FST instructions.

ER 0012 : (Check 1) The exponent is not "30"
during FFL, "30" should have to be loaded in the exponent floating point data register EM and transferred successively to EALU, EF and on the BIO lines.

[A2] = incorrect exponent
[A3] = 30

ER 0013 to ER 0019 : (Check 10) Incorrect floating point instruction.
execution with a non normalized operand.

Incorrect instruction	Error	Accumulator contents
FAD	13	
FMU	14	≠ 0
FDV	15	
FAD	16	
FMU	17	
FDV	18	= 0
FLD	19	

[A2] = incorrect status
[A3] = reference status /200 (CR = 2)

ER 0023 to ER 0026 : (Check 2) Wrong FLD Execution

A floating point data is transferred to the FPP by FLD and from the FPP to memory by FSTR to check the result of the FLD execution.
There is an error if the result is different from the initial floating point data :

ER 0023 occurs after FLD (T4)
ER 0024 occurs after FLD (T5)
ER 0025 occurs after FLD (T6)
ER 0026 occurs after FLD (T7)

[A5] , [A6] , [A7] reference floating point data (operand for FLD)
[A2] , [A3] , [A4] incorrect result into memory.

ER 0027 to ER 0029 : (Check 2) CR not correctly positioned by an FLD execution.

- the CR depends on the result value.
- to verify the FLD result is correct, execute an FSTR.

ER 0027 CR ≠ 0
ER 0028 CR ≠ 1
ER 0029 CR ≠ 2

ER 0031 to ER 0033 : (Check 3) CR not correctly positioned by a FST execution.

The CR depends on the transferred data :

ER 0031 when CR ≠ 0 for a floating point data = 0

ER 0032 when CR ≠ 1 for a floating point data > 0

ER 0033 when CR ≠ 2 for a floating point data < 0

ER 0034 to ER 0037 : (Check 2) Wrong FST Execution

The floating point data transferred from FPP to the memory is not correct. It should have the value of the floating point operand previously loaded in the FPP by a FLDR.

ER 0034 occurs after FST (T4)

ER 0035 occurs after FST (T5)

ER 0036 occurs after FST (T6)

ER 0037 occurs after FST (T7)

[A5] , [A6] , [A7] reference floating point data loaded into FPP.

[A2] , [A3] , [A4] incorrect result into memory after FST.

ER 0041 to ER 0043 : (Check 4) CR not correctly positioned by a FFL execution.

The CR depends on the result :

ER 0041 when CR ≠ 0 for a floating point data = 0
ER 0042 when CR ≠ 1 for a floating point data > 0
ER 0043 when CR ≠ 2 for a floating point data < 0

It is possible to check the result after a FST/FSTR execution.

ER 0044 : (Check 4) Wrong FFL Execution

The floating point data copied into memory by FST should have the reference value : /4000, /0000, /0001.

Verify the FFL execution :

- transfer of the double precision integer :
  A1 = 0
  A2 = 1.
- conversion into a floating point data :
  exponent loaded to "30" in EM.
  left shifts executed for normalization.

[A5] , [A6] , [A7] reference floating point data.

[A2] , [A3] , [A4] result after FFL execution.

ER 0045 : (Check 4) idem to above but FFL
A1 = /2000 A2 = 0
Result must be on FSTR
A1 = /4000 A2 = /0000 A3 = /001D

ER 0051 : (Check 5) FPP Accumulator Contents Destroyed by FFX

After a FFX the FPP accumulator contents checked after a FSTR should have the initial value-loaded by FLDR.

- Verify : - the FPP accumulator loading during FLD,
  - the contents is not modified during FFX,
  - it is correctly transferred into memory during FSTR.

[A5] , [A6] , [A7] reference floating point data.

[A2] , [A3] , [A4] result after FSTR execution.

ER 0052 : (Check 5) FFX Result Incorrect

The FPP floating point data is /4000, /0000, /0001.
The double precision integer stored in memory by FFX should have the value /0000, /0001.

The FPP floating point exponent is "1" :
Check - the right shift executions to obtain the exponent "30".

- the double precision integer transmission during FFX.
  0 → A1
  1 → A2

[A2] = 0)
[A3] = 1) reference value

[A4] , [A5] incorrect result.

ER 0053 to ER 0055 : (Check 5) CR not correctly postioned by an FFX execution.

FFX is executed with different floating point data :

- /0000, /0000, /0000 to obtain the integer double precision /00C0, /0000 0
- /4000, /0000, /0001 to obtain the integer double precision /0000, /0001
- /8000, /0000, /0000 to obtain the integer double precision /FFFF, /EFFF

ER 0053 if CR ≠ 0 for result = 0 [A2] , [A3] result of FFX
ER 0054 if CR ≠ 1 for result 0 " " " "
ER 0055 if CR ≠ 2 for result 0 " " " "

ER 0056 - ER 0057 : (Check 5) FFX Result Incorrect

Floating Point Data FFP accumulator contents	Double Precision Integer (FFX result)	Error Number
/7FFF, /7FFF, /0000	/0000, /0000	56
/8000, /0000, /0000	/FFFF, /7FFF	57
/8000, /0001, /001D	/C000, /0001	
/A000, /0000, /0001	/FFFF, /7FFF	
/8000, /0001, /0000	/0000, /0000	

Verify : - right shift executions to obtain the exponent "30".
- the correction (+1) for the negative data when a bit "1" is lost.
- the double precision integer transmission during FFX.

[A2] , [A3] reference value.

[A4] , [A5] incorrect result.

ER 0061 : (Check 06) Operand Destroyed During a Floating Point Addition.

[A5] , [A6] , [A7] = /4000, /0000, /0001 reference operand floating point data.

[A2] , [A3] , [A4] = modified floating point data.

ER 0062 to ER 0064 : (Check 06) FADR Result Incorrect

Floating Point Data		Operations to be Checked	Result	ER
Operand	FPP Accumulator Contents			
/4000, /0000, /0001	/5FFF, /7FFF, /001E	- operand alignment - mantissa addition	≠ /6000, /0000, /001E	0062
/AAAA, /2AAB, /5555	/5555, /5555, /5555	- mantissa addition	≠ 0	0063
/BFFF, /7FFF, /002E	/8010, /0010, /0010	- FPP accumulator contents - mantissa addition	= /BFFF, /7FFE, /002E	0064

[A5] , [A6] , [A7] reference result

[A2] , [A3] , [A4] incorrect FADR result

ER 0071 - ER 0072 : (Check 07) Incorrect Floating Point Subtraction Result

Floating Point Data		Operations to be Checked	Result	ER
FPP Accumulator Contents	Operand			
/9876, /2190, /0004	/4567, /4480, /FFFC	- operand alignment - mantissa subtraction	/9830, /6DCC, /0004	71
/4210, /4210, /3FFC	/8765, /2190, /4000	- FPP accumulator alignment - mantissa subtraction	/7CBB, /6291, /4000	72

[A5], [A6], [A7] reference result.

[A2], [A3], [A4] incorrect FSU result.

ER 0081 to ER 0084 : (Check 08) Incorrect Floating Point Multiplication Result

For each error, Check the following operations :

- correct loading of the multiplicator.
- successive right shifts during the multiplication.
- exponent addition.
- correct FSTR execution.

For ER 0081 and ER 0082, Check the left shift and exponent modification (-1) for result normalization.

For ER 0084, Check the right shift and the exponent modification (+1) to suppress the overflow.

ER 0091 to ER 0095 : (Check 09) Incorrect Floating Point Division Result

For each error, Check the following operations :

- correct loading of the dividend.
- the shift right of FA to have the dividend mantissa inferior to the divisor mantissa.
- successive right shifts of FA and Q during the division.
- exponent subtraction.
- correct FSTR execution.

For ER 0093, ER 0094, ER 0095, check the right shift and the exponent modification (+1) to suppress the overflow.

ER 0100 (Check 10) Incorrect Result after FFX with exponent = 30

In this case, the result in [A1] , [A2] should have the value of the mantissa. During FFX, the FPP accumulator contents is directly loaded in A1, A2.

After an error, [A2] and [A3] have the incorrect value.
[A4] = /4000)
[A5] = /0000) reference value.

ER 0101 (Check 10) Incorrect Result after FFX with nul mantissa or with negative exponent.

In these cases, during FFX, the FPP accumulator contents and the result in A1 and A2 should have the value "0".
[A2] , [A3] incorrect value
[A4] , [A5] reference value (correct FFX result).

ER 0104 : (Check 10) see ER 0107 but accumulator contents = 0.

ER 0105 : (Check 10) Incorrect FAD result

The memory operand "0" is added to the accumulator.
The result should have the value of the initial accumulator contents.

[A5] , [A6] , [A7] = /4000, /0000, /001E
[A2] , [A3] , [A4] = incorrect result.

ER 0106 : (Check 10) Incorrect FAD result

The memory operand "0" is added to the accumulator contents "0".
The result should have the value "0".

[A5] , [A6] , [A7] = /0000, /0000, /0000
[A2] , [A3] , [A4] = incorrect result.

ER 0107 : (Check 10) Incorrect FMU result (accumulator contents ≠ 0)

The memory operand "0" is multiplied by the accumulator contents /4000, /0000, /001E and "0". The result should have the value "0" in the two cases :

[A5] , [A6] , [A7] = /0000, /0000, /0000
[A2] , [A3] , [A4] = incorrect result.

ER 0108 : (Check 10) Incorrect FLD result

Any accumulator contents whatever, after FLD, should be replaced by the memory operand.

Accumulator Contents	Memory Operand	Result
0 /4000, /0000, /001E	/4000, /0000, /001E 0	/4000, /0000, /001E 0

[A5] , [A6] , [A7] = correct result
[A2] , [A3] , [A4] = incorrect result

ER 0109 : (Check 10) Incorrect status after a division by zero.

Whatever the acculmulator contents may be a division by zero must be rejected and generates a trap /7E.

Verify the floating point status :

[A3] = /0300 = correct status
division by zero
[A2] = incorrect status

ER 0110 : (Check 10) Incorrect FSU result

The subtraction result should have the value :
/4000, /0000, /0001
[A5] , [A6] , [A7] reference result.
[A2] , [A3] , [A4] incorrect FSU result.

ER 0111 : (Check 10) Incorrect FMU result

The accumulator contents being nul, the FMU result should have the value 0.
[A5] , [A6] , [A7] = 0
[A2] , [A3] , [A4] = incorrect FMU result

ER 0112 : (Check 10) Incorrect FDV result

The dividend being "0", the FDV result should have the value 0.
[A5] , [A6] , [A7] = 0
[A2] , [A3] , [A4] = incorrect FDV result.

ER 0113 to 0119 : (Check 10) Incorrect alignment test

Accumulator Contents	Memory Operand	FPP Instruction	Result	ER
/4000, /0000, /7FFF	/5555, /5555, /5555	FAD	accumulator contents	113
/5555, /5555, /5555	/4000, /0000, /7FFF	FSU	/8000, /0000, /7FFE - (memory operand)	114
/AAAA, /2AAA, /AAAA	/4000, /0000, /7FFF	FAD	memory operand	115
/4000, /4000, /8000	/7FFF, /7FFF, /7FFF	FAD	memory operand	116
/4000, /4000, /8000	/4000, /4000, /4000	FAD	memory operand	117
/AAAA, /2AAA, /AAAA	/4000, /0000, /001E	FAD	memory operand	118
/4000, /4000, /8000	/AAAA, /2AAA, /AAAA	FAD	memory operand	119

[A5] , [A6] , [A7] reference result.

[A2] , [A3] , [A4] incorrect result.

ER 0120 to 0123 : (Check 10) Result with overflow incorrectly rectified

These errors occur after a FAD. The accumulator contents and the memory operand have the same value. For each error, verify the correct execution of the right shift to rectify the overflow.

Memory Operand Accumulator contents	Result	ER
/4000, /4000, /EEEF	/4000, /4000, /EEF0	120
/7FFF, /7FFF, /EEFF	/7FFF, /7FFF, /EF00	121
/8000, /0000, /EFFF	/8000, /0000, /F000	122
/AAAA, /0000, /FFFF	/AAAA, /0000, /0000	123

[A5], [A6], [A7] reference result.

[A2], [A3], [A4] incorrect result.

ER 0124 : (Check 10) Non normalized result incorrectly rectified

/AAAA, /5555, /000F is added to /5555, /2AAA, /000F.
The result is not normalized. Check for correct execution of the left shift to obtain a correct result /8000, /0000, /FFF1.

[A5] , [A6] , [A7]= correct result.

[A2] , [A3] , [A4]= incorrect result.

ER 0125 : (check 10) see ER 0108
but initial accumulator value = 0
and memory operand value = 0

ER 0126 : (check 10) see ER 0109
but accumulator value ≠ 0

ER 0127 to ER 135:(Check 10) trap 7E not detected during the following instructions :

ER 127 : FAD with operand unnormalized , accumulator value ≠ 0
ER 128 : FMU " " " " " "
ER 129 : FDV " " " " " "
ER 130 : see ER 0127 but accumulator value = 0
ER 131 : see ER 0128 " " "
ER 132 : see ER 0129 " " "
ER 133 : FLD with operand unnormalized
ER 134 : FDV with an operand = 0 and accumulator = 0
ER 135 : FDV with an operand = 0 and accumulator ≠ 0

ER 136:(Check 10) accumulator destroyed by a FAD with an operand unnormalized.

[A5] , [A6] , [A7] = correct result
[A2] , [A3] , [A4] = incorrect result

ER 0202 to ER 0208 : (Check 11) Trap /7E not detected during the following instruction :

ER 0202 : FDV with overflow
ER 0203 : FDV with underflow
ER 0204 : FMU with overflow
ER 0205 : FMU with underflow
ER 0206 : FFX with overflow
ER 0207 : FAD with overflow
ER 0208 : FAD with underflow

ER 0220 to ER 0223 : (Check 11) Overflow and underflow not detected during FMU and FDV.

FPP Accumulator Contents	Memory Operand	FPP Instruction	Waited Status	ERROR
/7FFF, /7FFF, /7FFF	/4000, /4000, /8000	FDV	/0100 CR=1	220 overflow not detected
/4000, /4000, /8000	/7FFF, /7FFF, /7FFF	FDV	0 CR=0	221 underflow not detected
/7FFF, /7FFF, /7FFF	/7FFF, /7FFF, /7FFF	FMU	/0100 CR=1	222 overflow not detected
/4000, /4000, /8000	/4000, /4000, /8000	FMU	0 CR=0	223 underflow not detected

[A2] incorrect status.

[A3] reference status.

ER 0224 : (Check 11) Memory operand has been modified during FMU/S with trap underflow.

[A5] , [A6] , [A7] = /4000, /4000, /8000
[A2] , [A3] , [A4] = result

ER 0225 to 227 : (Check 11) Overflow and underflow not detected during FFX and FAD.

FPP accumulator contents	Memory operand	FPP instruction	Wait status	Error
/4000, /0000, /7FFF	/	FFX	/0100 CR=1	225 overflow not detected
/7FFF, /7FFF, /7FFF	/4000, /0000, /7FFF	FAD	/0100 CR=1	226 overflow not detected
/5555, /2AAA, /8000	/AAAA, /5555, /8000	FAD	0 CR=0	227 underflow not detected

ER 0233 to ER 0235 : (Check 11) FPP accumulator contents destroyed during FAD with a page fault. (Take care of the CPU type in A9!). FAD should not be executed. The accumulator contents should have the precedent value.

ER 0233 : page fault has occurred on the first part of the mantissa.
ER 0234 : page fault has occurred on the second part of the mantissa.
ER 0235 = page fault has occurred on the exponent.

[A5] , [A6] , [A7] = /AAAA, /2AAA, /AAAA
[A2] , [A3] , [A4] incorrect FPP accumulator contents.

ER 0236 to ER 0238 : (Check 11) Page fault not detected during FAD with memory operand M1, M2 2^E in a protected segment.

Verify : . the segment 2 in the segment table. The bit 06 should be 1.
. the page fault generation in MIOP.

ER 0236 : PAF should have occurred on M1.
ER 0237 : PAF should have occurred on M2.
ER 0238 : PAF should have occurred on E.

ER 0239 to ER 023B : (Check 11) FPP accumulator contents destroyed during FST with a page fault (trap /7A). (Take care of the CPU type in A9!)

FST should not be executed. The accumulator contents should have the preceoent value.

ER 0239 : page fault has occurred on the first part of the mantissa.

ER 023A : page fault has occurred on the second part of the mantissa.

ER 023B : page fault has occurred on the exponent.

[A5] , [A6] , [A7] = /AAAA, /2AAA, /AAAA
[A2] , [A3] , [A4] incorrect FPP accumulator contents.

ER 0242 to ER 0244 : (Check 11) Page fault not detected during FST with memory operand M1, M2, 2^E in a protected segment.

Verify : . the segment 2 in the segment table. The bit 06 should be 1.
. the page fault generation in MIOP.

ER 0242 : PAF should have occurred on M1.

ER 0243 : PAF should have occurred on M2.

ER 0244 : PAF should have occurred on E.

ER 0604 to ER 0607 : (Check 06) Incorrect Floating point addition result.

Floating Point Data		Addressing Mode	Operations to be Checked	Result	ER
Operand	FPP accumulator contents				
/7654, /3210, /FFF6	/7654, /3210, 0	T4	- correct addressing - operand alignment - mantissa addition	≠ /7671, /7C9C, 0	0604
/8000, /0001, /0014	/AAAA, /5555, /FFF6	T5	- correct addressing - FPP accumulator alignment - mantissa addition	≠ /8000, /0000, /0014	0605
/A222, /1112, /FF7F	/5111, /0889 /FF80	T6	- correct addressing - operand alignment - mantissa addition - result normalization	≠ /4444, /2224, /FF7F	0606
/4321, /2AAA, /801E	/AAAA, /5555, /800E	T7	- correct addressing - FPP accumulator alignment - mantissa addition	≠ /4320, /7FFF, /801E	0607

[A5] , [A6] , [A7] reference result.
[A2] , [A3] , [A4] incorrect FAD result.

ER 0613 to ER 0617 : (Check 06)

Incorrect result after a floating point addition with result into memory.

Floating Point Data		Addressing Mode	Operations to be Checked	Result in memory	ER
Operand	FPP accumulator contents				
/4777, /3BB9, /FFFF	/B888, /4446, /FFFF	T3	- correct addressing - mantissa addition - result normalization - memory transfer	≠ /8000, /0000, /FFE1	0613
/AAAA, /2AAA, /FEE2	/5555, /5555, /FF00	T4		≠ /5555, /5555, /FF00	0614
/9999, /1999, /0000	/6666, /6666, /4000	T5	- correct addressing - Result = FPP accumulator contents. - Memory transfer	≠ /6666, /6666, /4000	0615
/8888; /0888, /8000	/7777, /7777, /7FE2	T6		≠ /7777, /7777, /7FE2	0616
/BBBB, /3BBB, /F000	/4444, /4444, /0100	T7		≠ /4444, /4444, /0100	0617

[A5], [A6], [A7] reference result.

[A2], [A3], [A4] incorrect FAD result.

ER 0623 : (Check 06) FPP accumulator contents destroyed by FAD with result storage into memory.

If there is no ER 0613, the precedent FAD is correctly executed.
The correct result is in memory.
Verify the FPP accumulator evolution during FAD execution.

[A5], [A6], [A7] = /B888, /4446, /FFFF
correct FFP accumulator contents.
[A2], [A3], [A4] incorrect FFP accumulator contents.

7. PROGRAM MODIFICATION

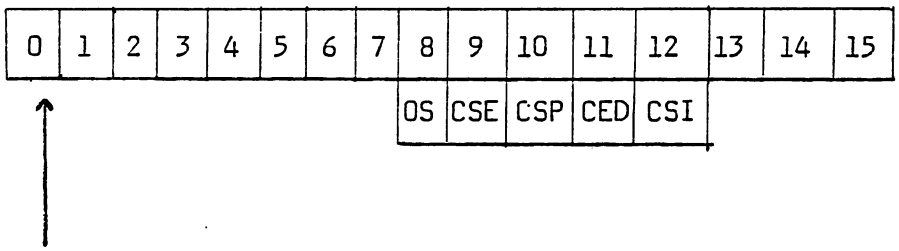
After laoding, the program stops in /700. For a non standard initialization (see paragraph 4-1), the operator should load A9 and A10 to introduce the parameters relating to FPPE operation and porgram processing.

7.1. Parameters Related to Program Processing

- "One Shot" OS command.
- "Clear stop on error" CSE command.
- "Clear stop on program" CSP command.
- "Clear edition" CED command.
- "Clear stop information" CSI command.
- "Checks" to be executed command.

7.1.1. A9 Register

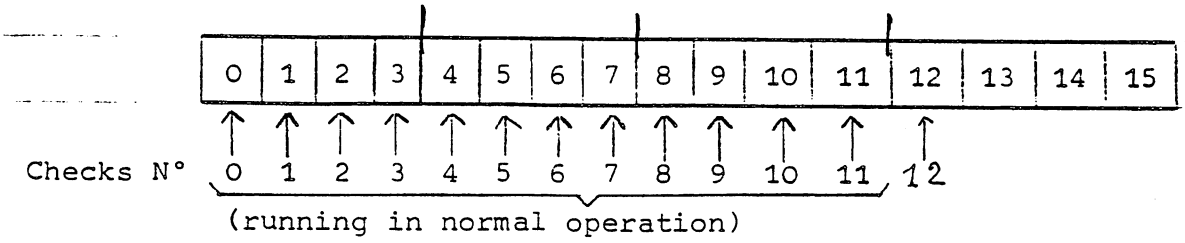
In standard operation (see paragraph 4-1), the bits 8 to 12 are initialized at 0. Bit 0 is set to "1" to indicate a CP2B.



0 = CP1BF/P857EB

1 = CP2B

7.1.2. A10 Register

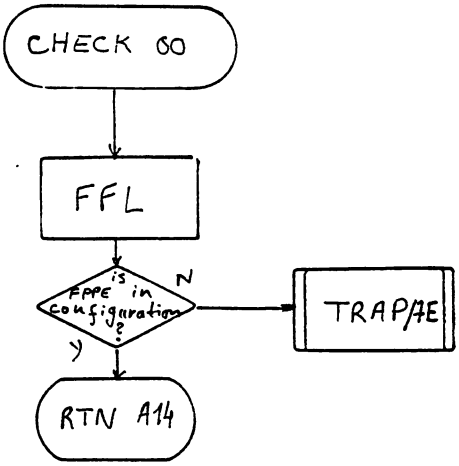


In non standard operation, load A10 before starting the program :

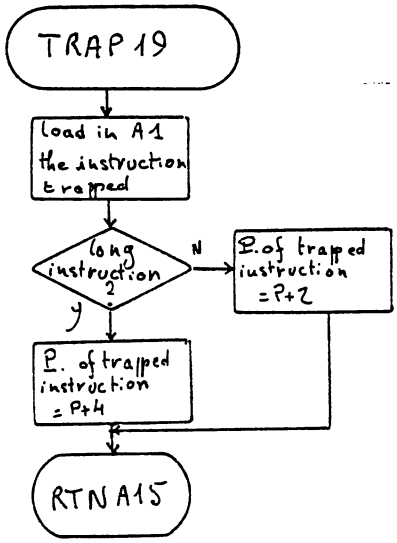
- checks to be run : corresponding bits of A10 initialized at 1.
- checks absent : corresponding bits of A10 initialized at 0.

8. FLOW CHARTS

FPPE present

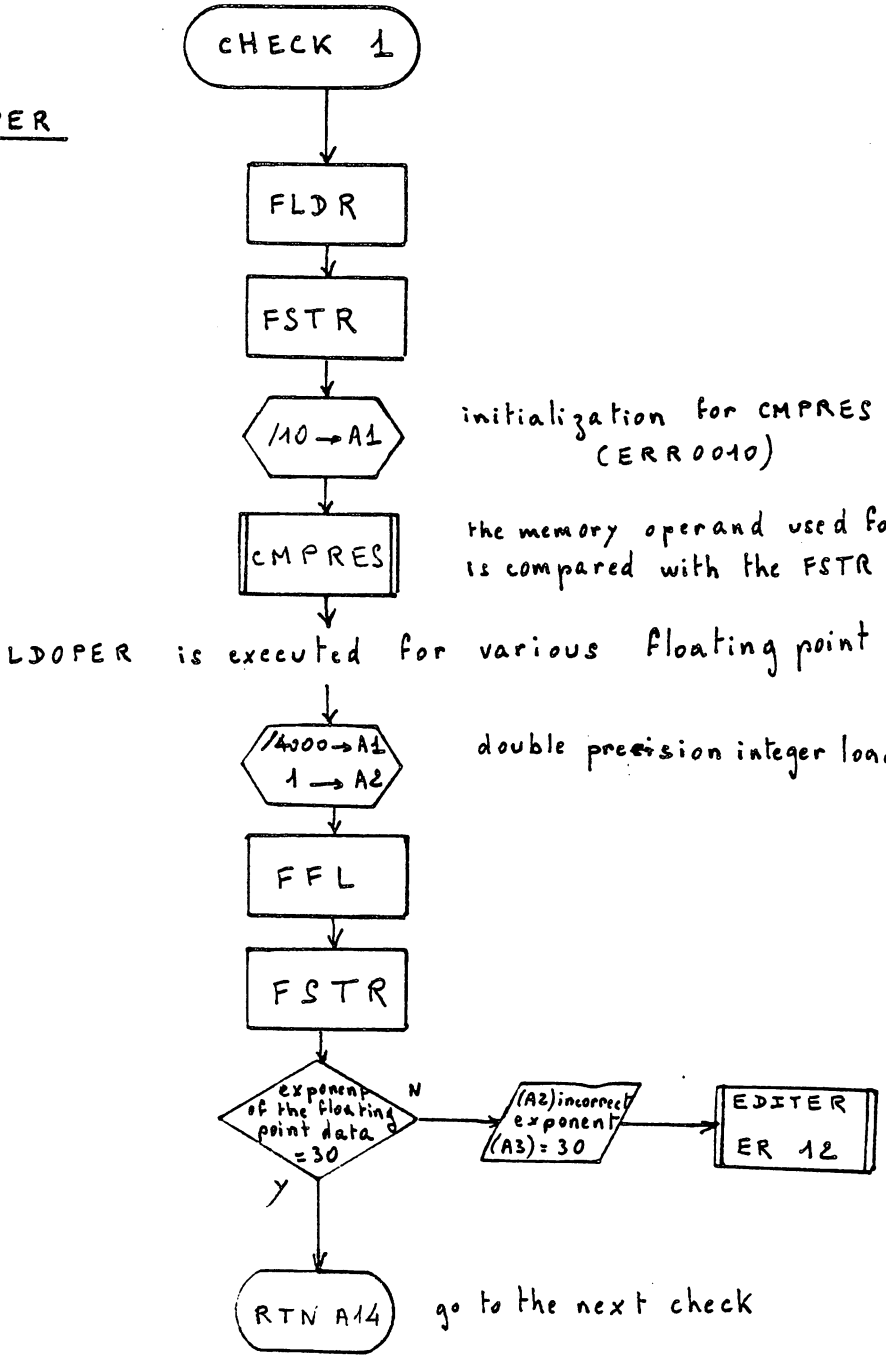


Trap 19 routine

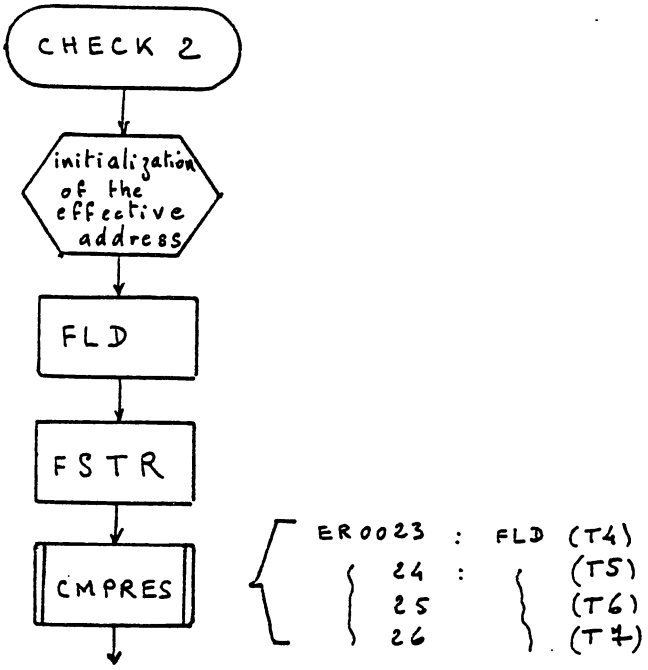


Test of data path :

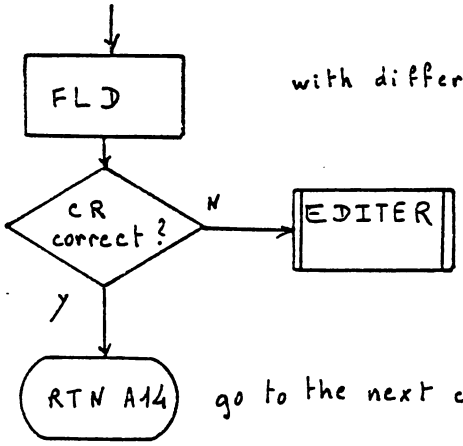
LDOPER



Floating load instruction test



The precedent part is executed successively for FLD in addressing mode T4, T5, T6, T7



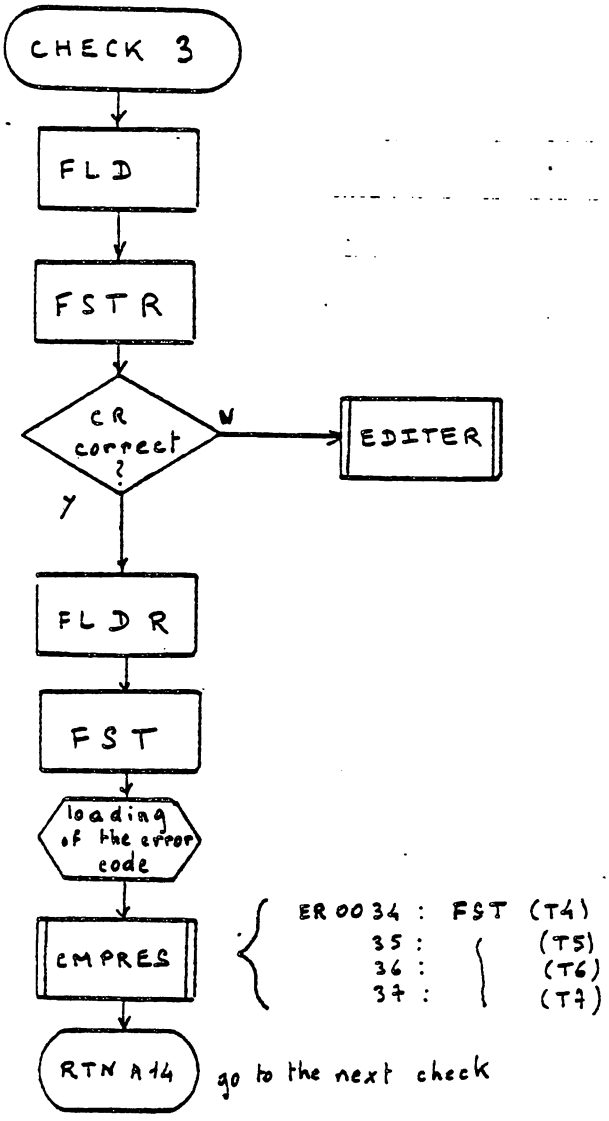
with different floating point operand

go to the next check

	Floating point operand	CR	ER
OPZERO	/0000, /0000, /0000	≠ 0	27
OPE30	/4000, /0000, /001E	≠ 1	28
OPDATP	/8000, /0000, /0000	≠ 2	29

Floating Store instruction test

Floating point operand		CR	ER
OPZERO	(0)	≠0	31
OPE30	(>0)	≠1	32
OPDATP	(<0)	≠2	33

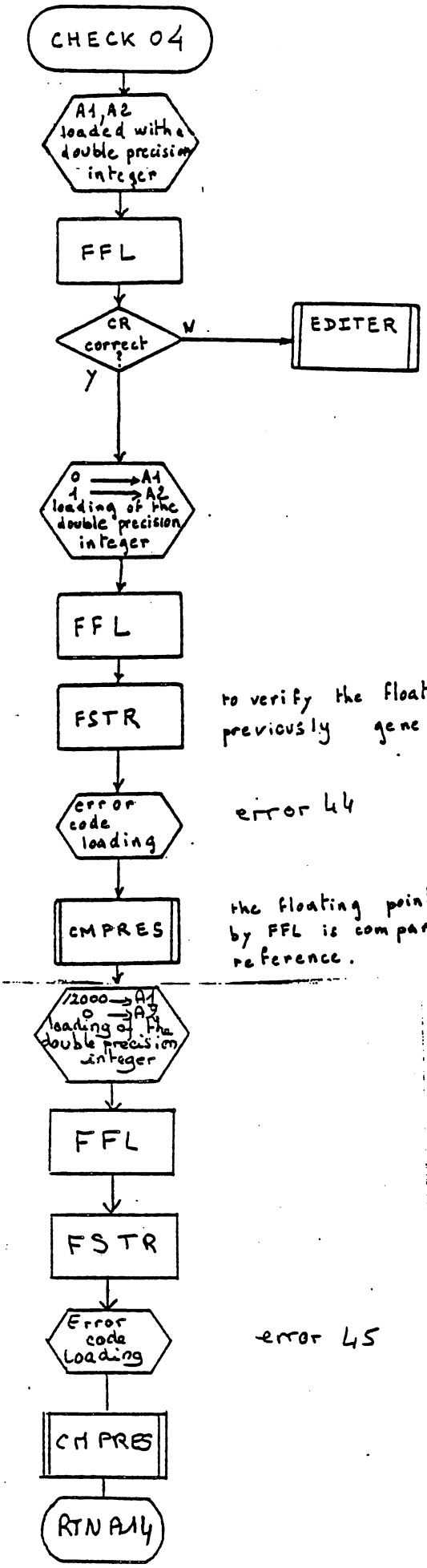


C301

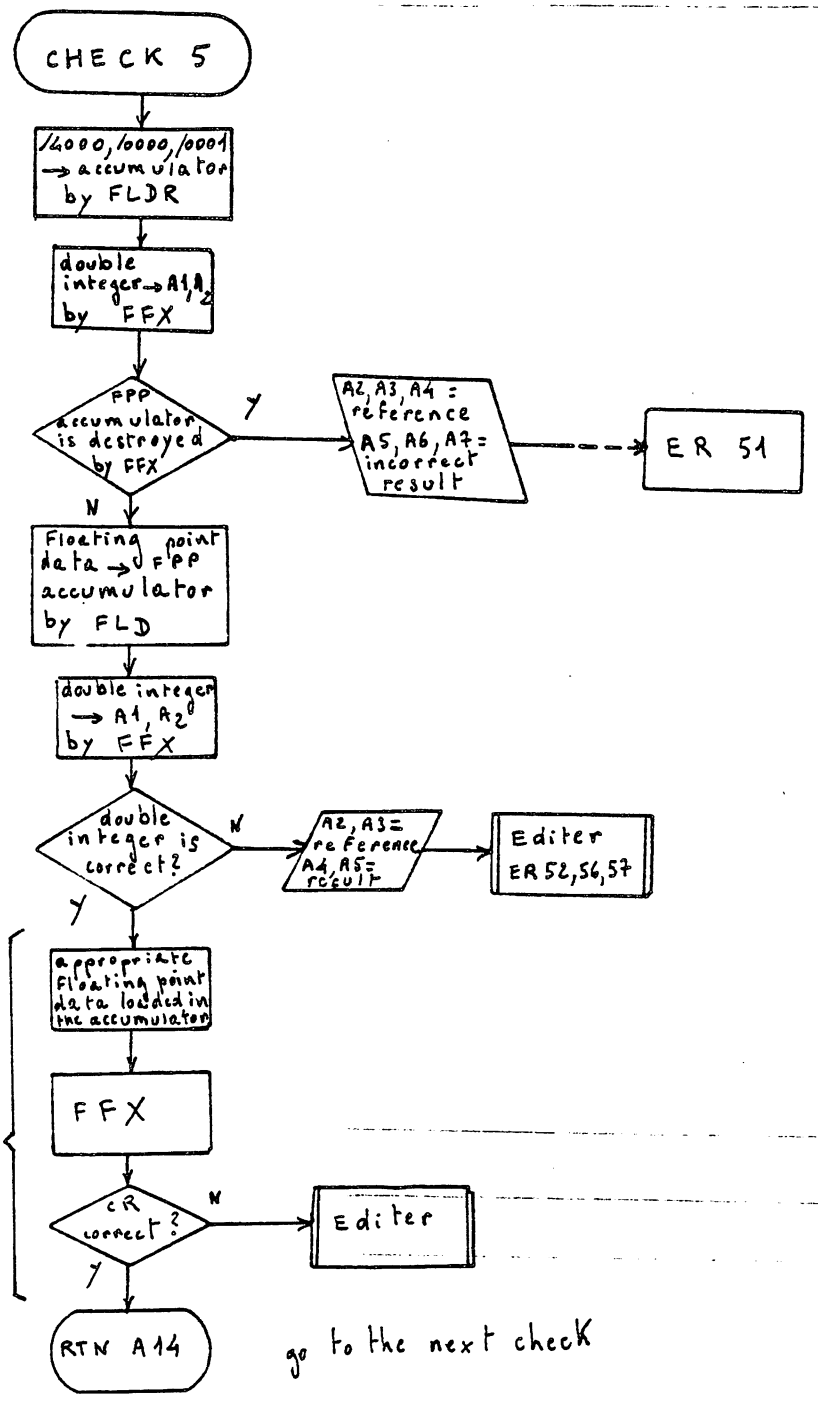
C301 is executed successively for FST in addressing mode T4 to T7

Floating Float instruction test :

double precision integer	CR	ER
/0000, /0000	≠0	41
/4000, /0000	≠1	42
/8000, /0000	≠2	43



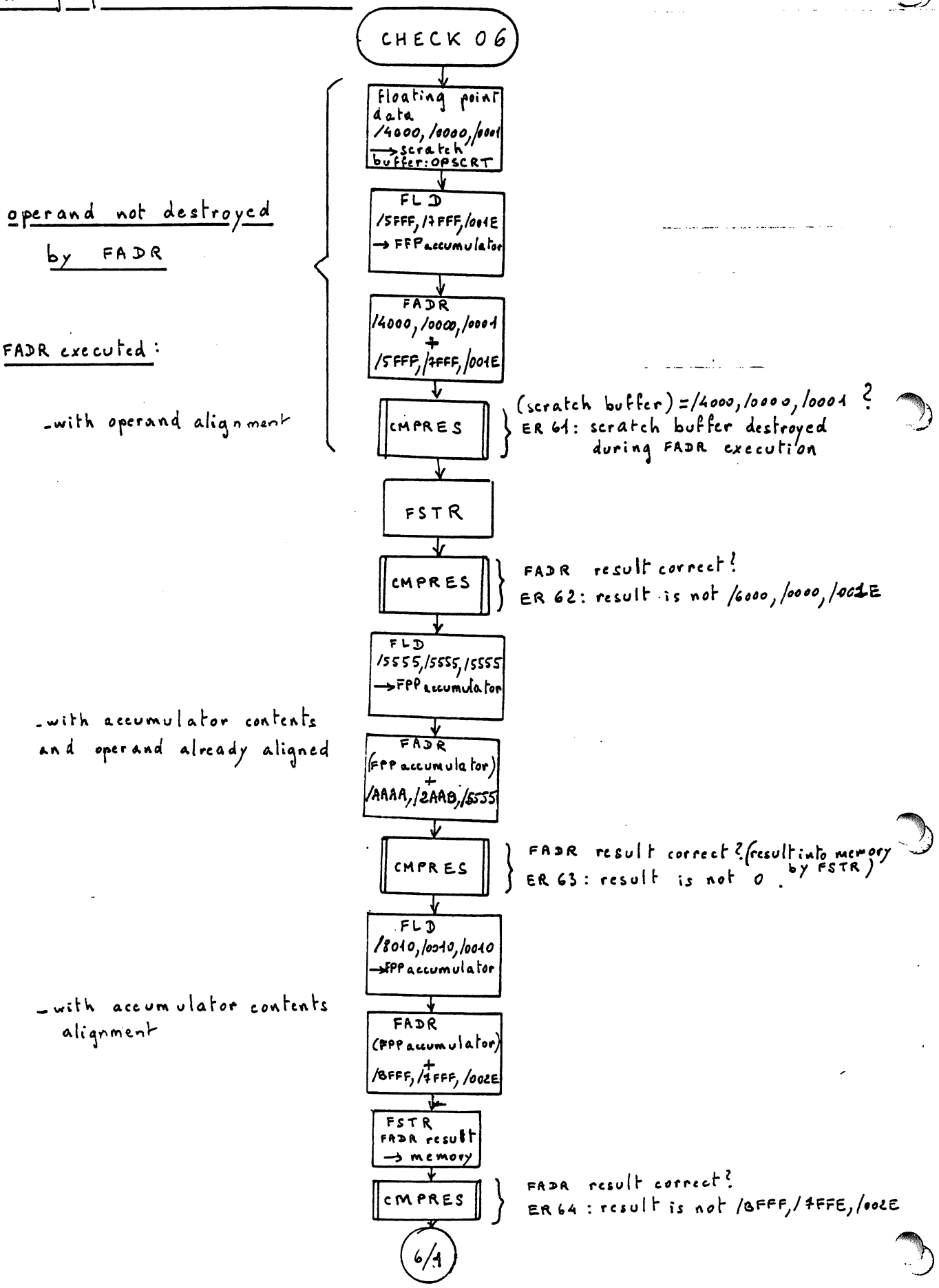
Floating Fix instruction test:



executed for different values

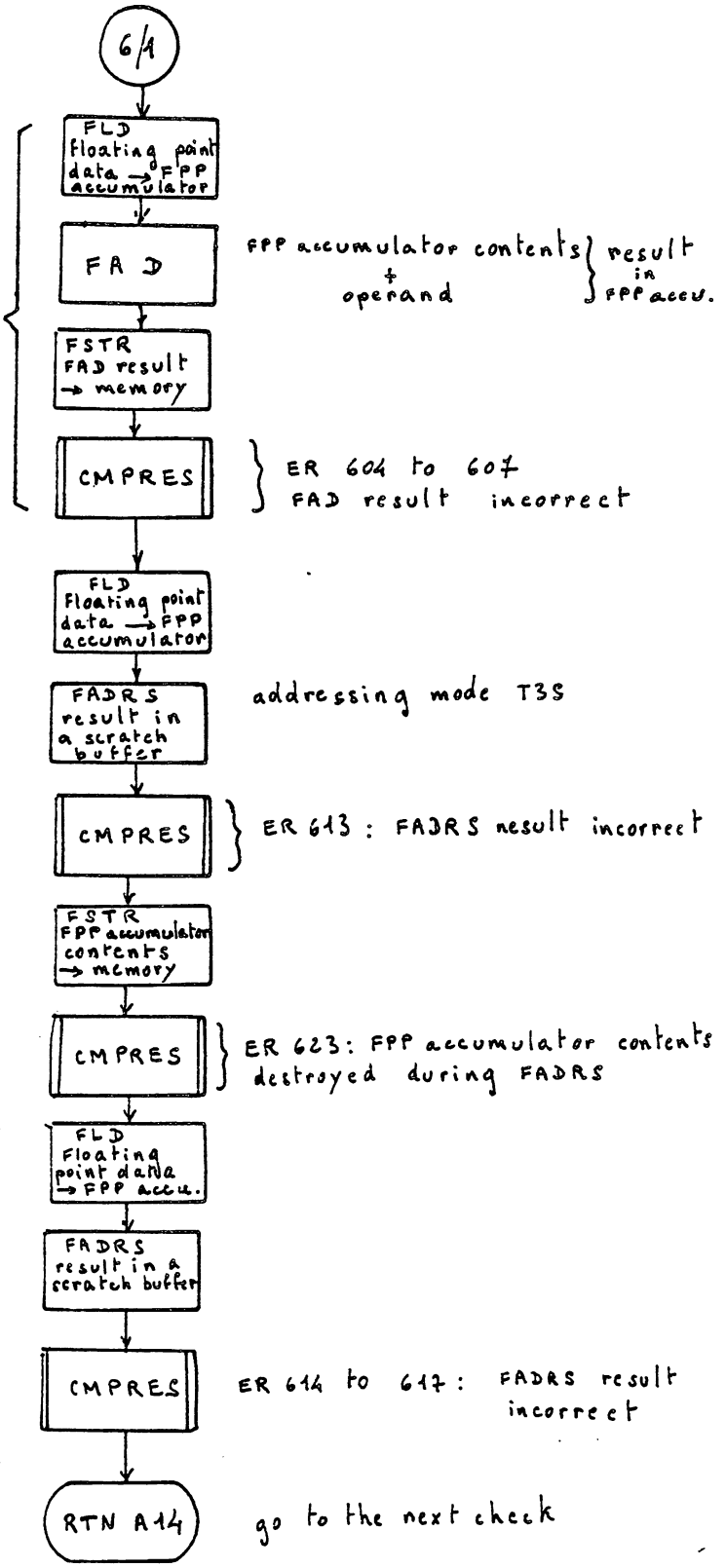
FPP floating point data	FFX result	CR	ER
/0000, /0000, /0000	0	≠0	53
/4000, /0000, /0001	>0	≠1	54
/8000, /0000, /0000	<0	≠2	55

Floating point addition test:



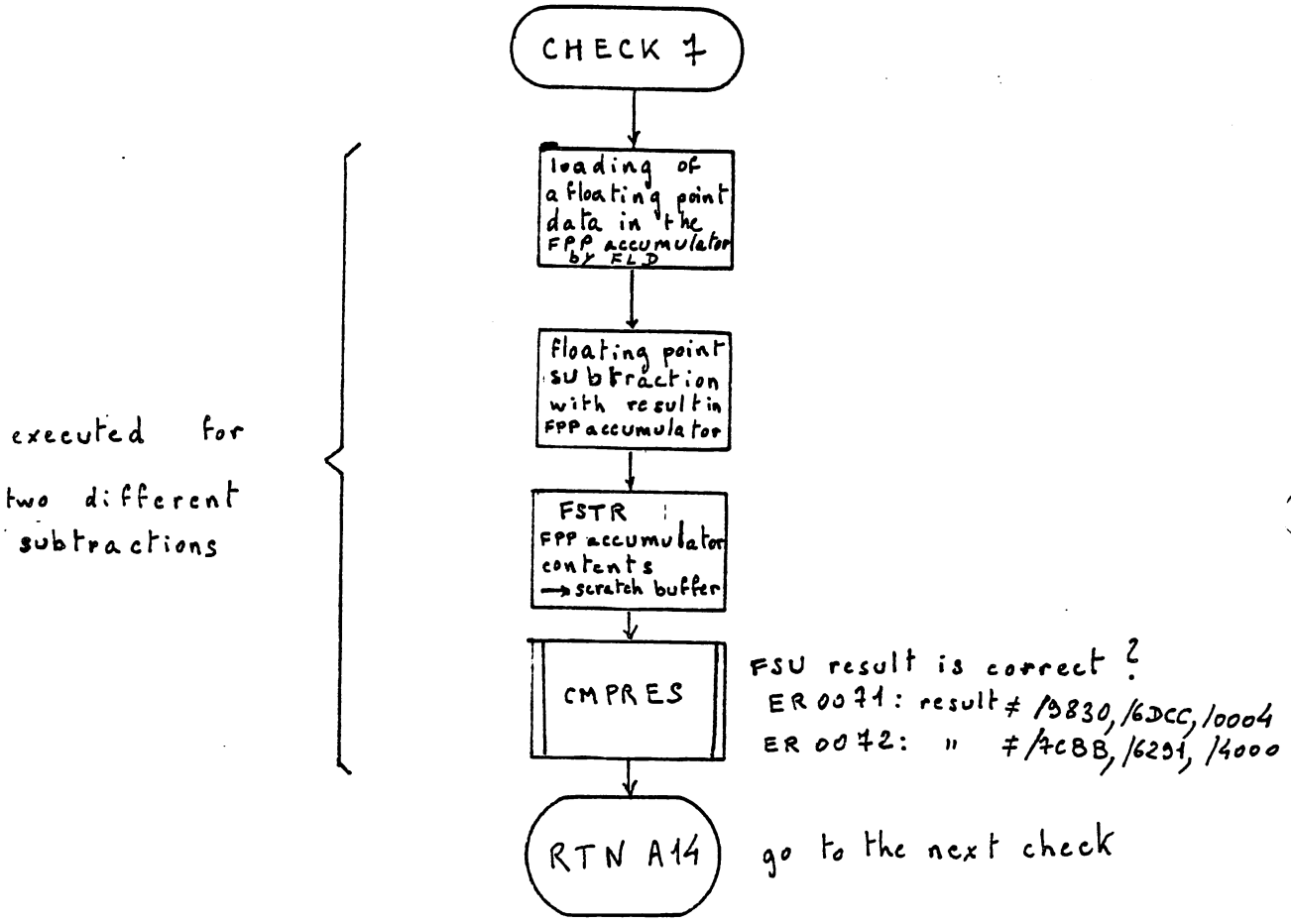
C6 AT4

executed with FAD in addressing mode T4 to T7 and with various values

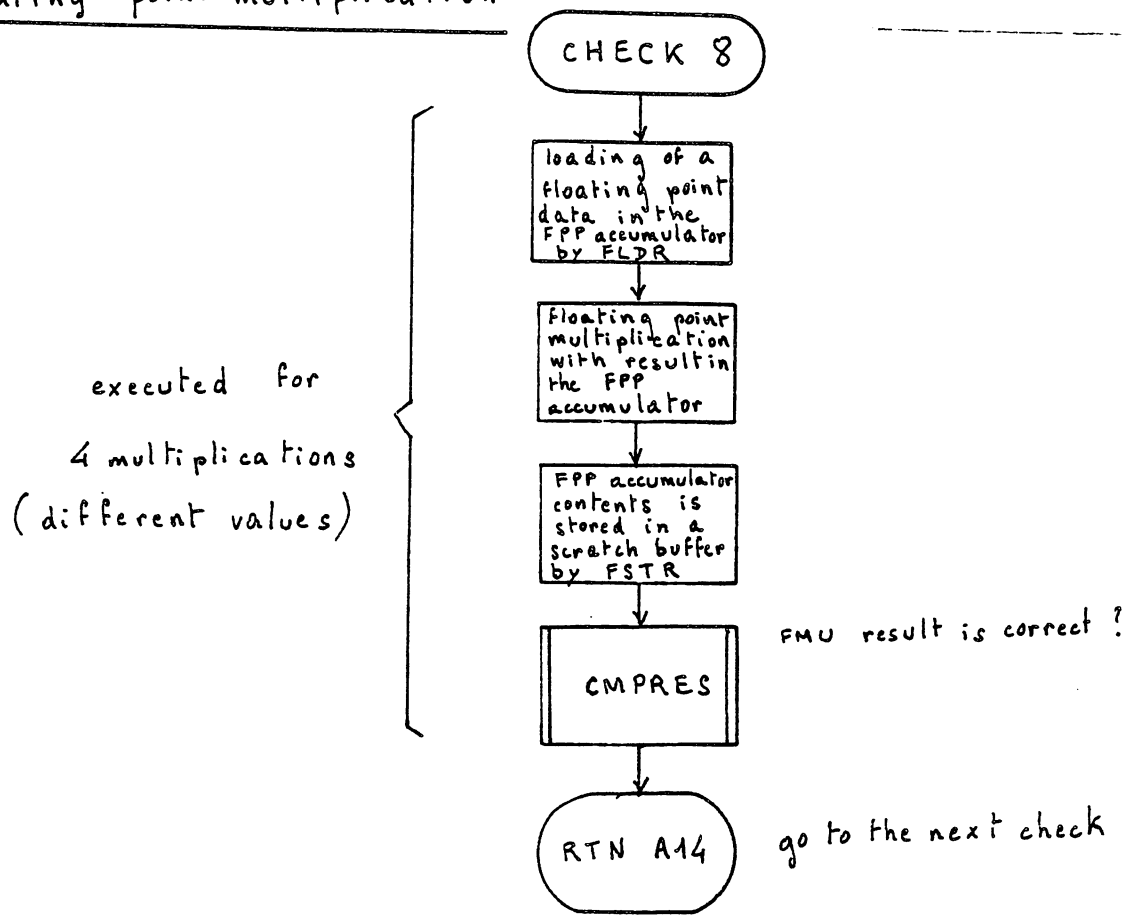


executed with FADR in addressing mode T4S to T7S with various values

Floating Point subtraction test:

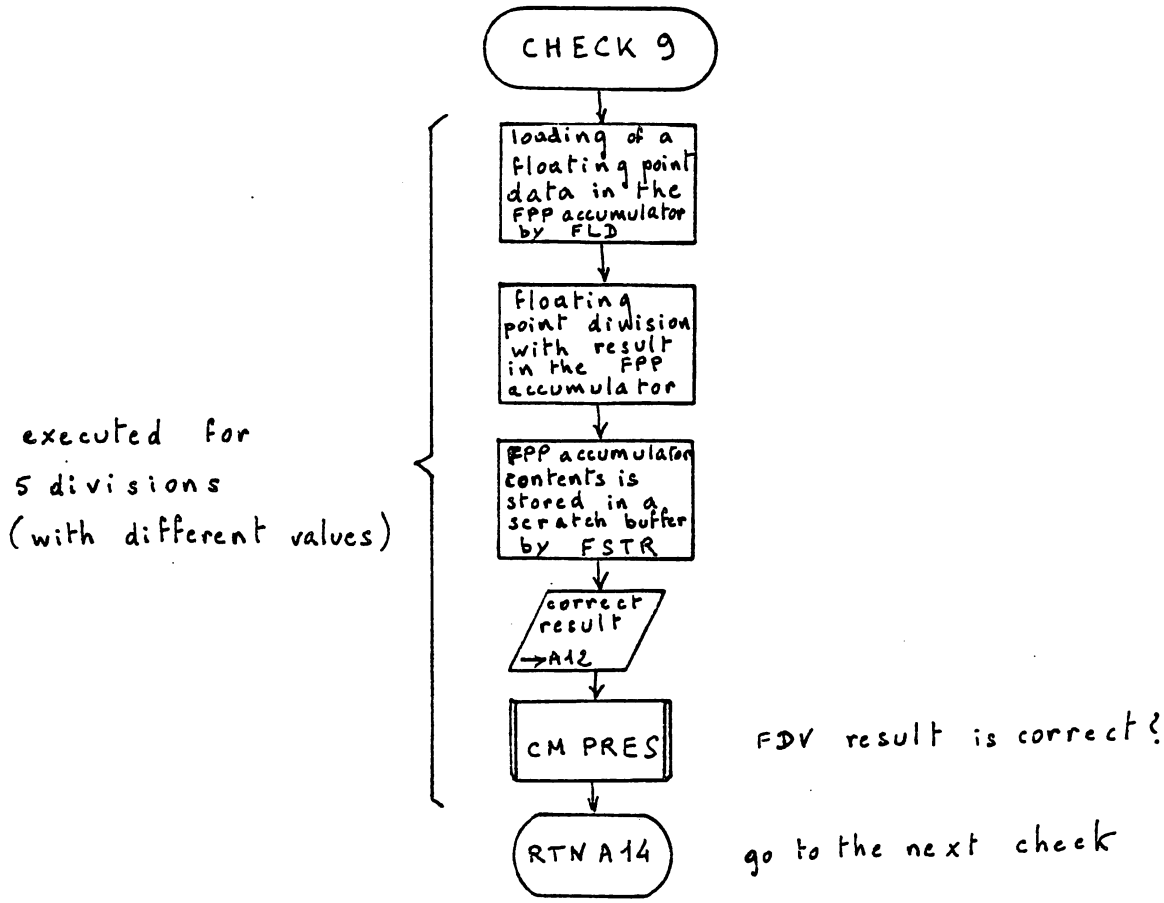


Floating point multiplication



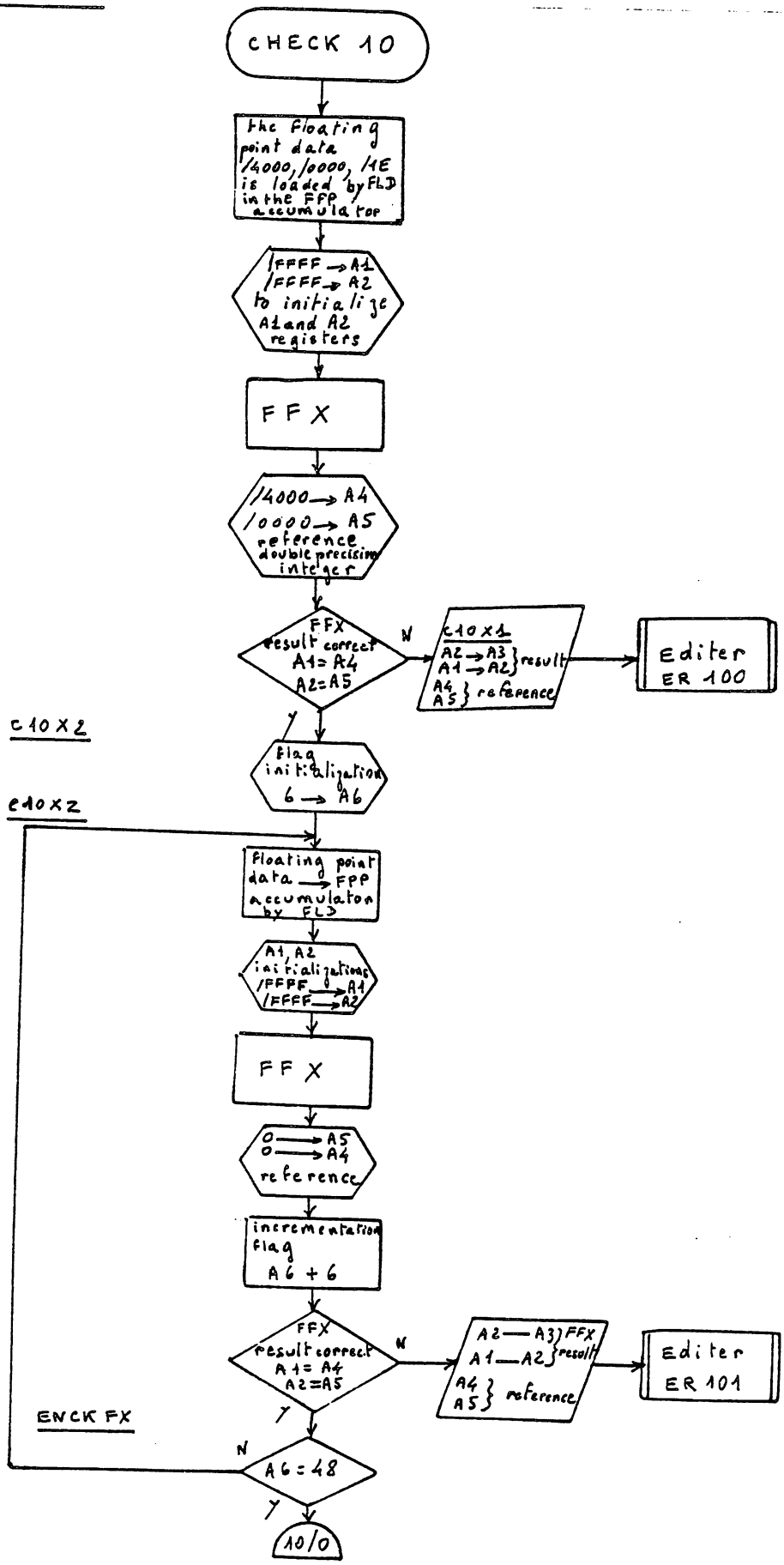
	memory operand	multiplicator (accumulator contents)	result ≠	ER
Test 1	/4000,/0000, /0001	/4000,/0000,/EEEF	/4000,/0000,/EEEF	81
Test 2	/BBBB,/5DDD,/EEFF	/6666, /3333, /0001	/92C5,/7C95,/EEFF	82
Test 3	/7777,/3BBB,/0001	/AAAA,/5555,/EFFF	/B05B,/02D8,/F000	83
Test 4	/8000,/0000,/FFFF	/8000,/0000,/0001	/4000,/0000,/0001	84

Floating point division



	memory operand (divisor)	accumulator contents (dividend)	result ≠	ER
Test 1	/8000,/0000,/0001	/4000,/0000,/1110	/8000,/0000,/110E	91
Test 2	/AAAA,/5555,/0001	/B05B,/02D8,/1100	/7777,/3BBB,/10FF	92
Test 3	/6666,/3333,/0001	/92C5,/7C96,/1000	/BBBB,/5DDE,/1000	93
Test 4	/4000,/0000,/0001	/8000,/0000,/0000	/8000,/0000,/0000	94
Test 5	/4000,/0000,/0001	/4000,/0001,/0000	/4000,/0000,/0001	95

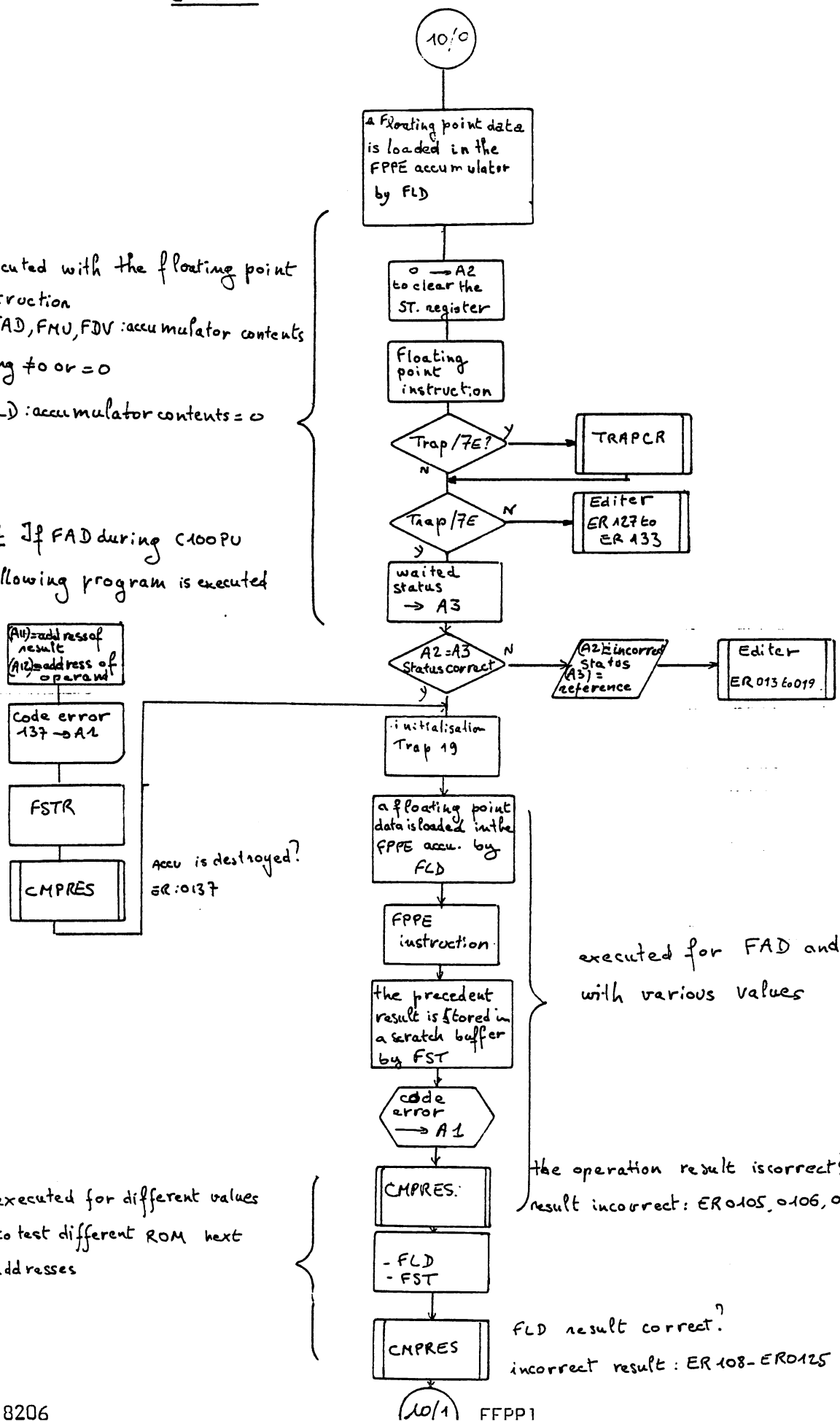
ROM next address test:

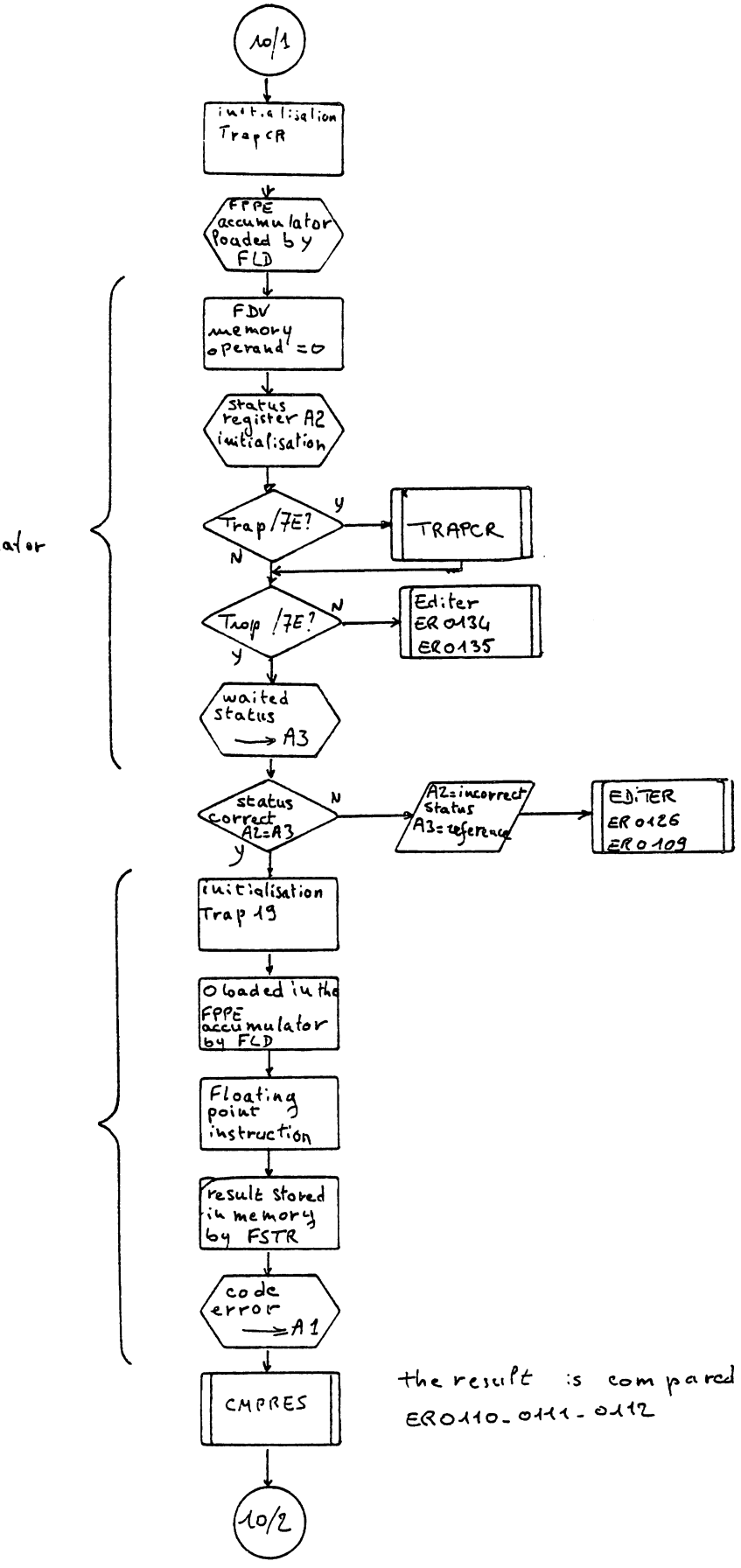


executed for different data values

"Operation test"

C100 PU





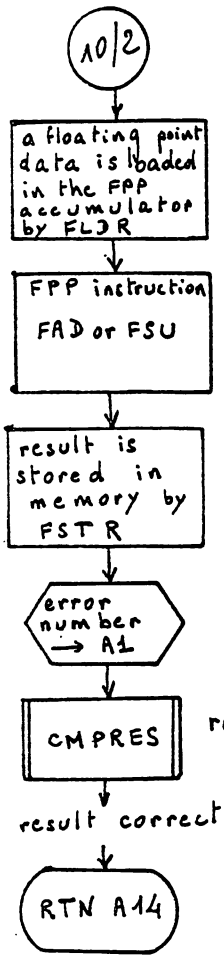
executed with the initial values of the FPPE accumulator

- 0
- /4000, /0000, /001E

executed with the floating point instruction FMU
FSU
FDV

the result is compared with a reference ER0110-0111-0112

alignment and result tests:



executed for the tests	possible errors
C10 ALG	ER0110 to ER0119
C10 ROV	ER0120 - ER0121
C10 RUD	ER0122 - ER0123
C10 RUN	ER0124

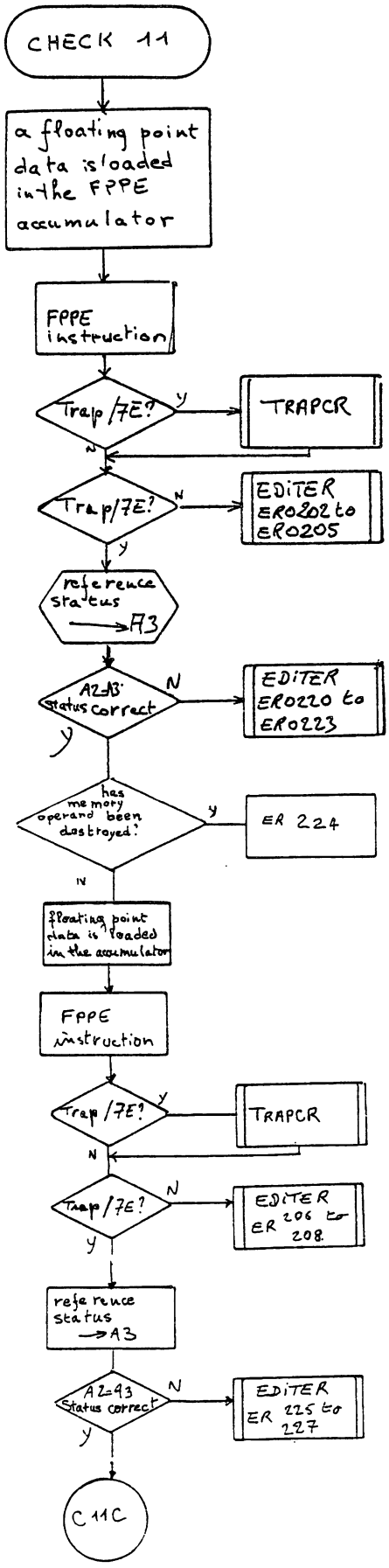
Abnormal conditions :

C11B

executed with FDV and FMU with different operands to obtain "overflow" and "underflow" status

only with FMUS instruction

executed with FFX, and FAD instruction with different operands to obtain "overflow" and "underflow" status

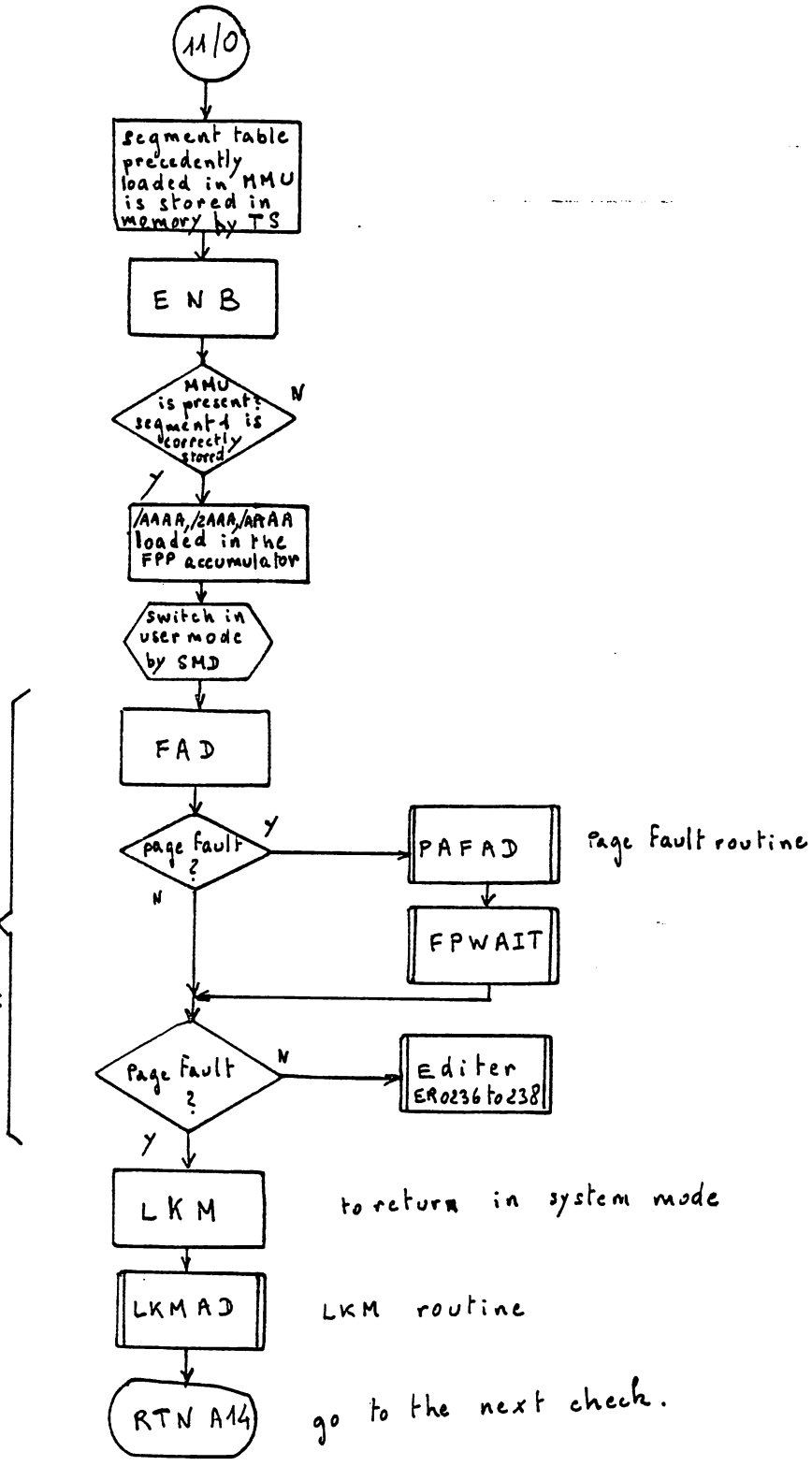


Page fault check:

C11 C

segment table

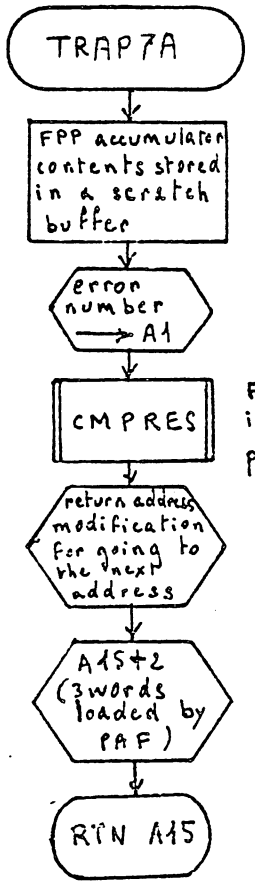
/0000
/0400
/0A00
/0C00
/1000
/1400
/1800
/1C00
/2000
/2400
/2800
/2C00
/3000
/3400



executed 3 times for generating a page fault during the addition successively on the mantissas and on the exponent.

ENCK

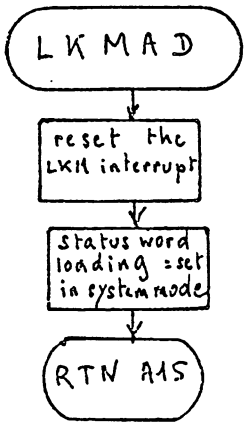
Page fault routine



FPP accumulator contents is compared with a reference:
possible errors : 233 - 234 - 235
239 - 23A - 23B

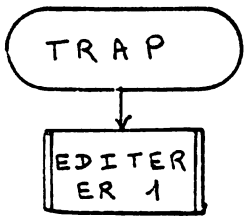
go to the check 11

LKM routine:



go to the check 11

TRAP ROUTINE



comparison between two floating point data

