Title:

CPU810 MICROPROGRAMMED TEST
USERS' MANUAL
REVISION 1.0

A/S REGNECENTRALEN

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 30-M167
Edition: 78.11.29
Author: Flemming Kam

Keywords:

CPU810, Test, Microprogrammed test, CPU test, Memory test, RC8000, RC8002, RC8003, RC8004.

Abstract:

This manual is a users' guide to CPU810 Microprogrammed test.

Printed by A/S Regnecentralen, Copenhagen

0. Index. Page 0.

1. Revision Status..... 2
2. Abstract..... 2
3. Operating Procedure..... 2
   3.1 Operating via OCP..... 2
   3.2 Operating via TCP..... 3
4. Messages..... 4
   4.1 Error Messages..... 4
      4.1.1 Operating via OCP..... 4
      4.1.2 Operating via TCP..... 4
      4.1.3 Error Types..... 5
      4.1.4 Action after Error..... 6
   4.2 Test Synchronization..... 6
5. Program Description..... 7
   5.1 CPU Test..... 7
   5.2 Memory Test..... 7
      5.2.1 Normal Mode..... 7
      5.2.2 Short Mode..... 8
6. Functionality..... 8
   6.1 CPU Test..... 8
   6.2 Memory Test..... 9
7. Algorithm, Flow Diagram..... 9
   7.1 The Memory Algorithm, Normal Mode..... 9
   7.2 Flow Diagram..... 11
8. Appendix..... 12

1. Revision Status.

1.

0. Preliminary Edition 780619/FK
1. First edition 781129/FK.

2. Abstract.

2.

The test consists of two parts, a CPU test and a memory test. To insure correct operation of the CPU810 before loading the system, the test is automatically run before autoload. In this case the memory test is run in short mode, in order to reduce the time consumption (App. 5 sec/32 Kwords). In normal mode the test is able to make a thorough test of the memory. The time consumption for a pass i about 130 sec/32 Kwords.

3. Operating Procedure.

3.

3.1 Operating via OCP.

3.1

The test is started when the autoload button at OCP is activated. Depending of a switch at the front edge of the CPU810 the test is either run in short mode or in normal mode. In the latter case the test is performed continuously.

The test is broken and restarted if the autoload button is activated.

In some error situations it is desirable to dump the core-image. For this purpose there is a second switch at the front edge in order to disable the test.

3.2 Operating via TCP.

The test is started with following commands typed at TCP:

T 0 : RUN ALL, i.e. both CPU and memory test, normal mode.

T 1 : RUN ONLY CPU TEST.

T 2 : RUN MEMORY TEST, normal mode.

T 3 : RUN MEMORY TEST, short mode.

T 4 : INITIALIZATE REGISTERS, TURN THE AUTOLOAD LAMP ON, etc. (This action is also performed in T 0-3).

T 5 : CLEAR AND SIZE MEMORY, W3: = last loc in MEM.

R ADDRESS (AENTRY): RUN TEST AS BY AUTOLOAD.

0 <MICROADDRESS> : START TESTLOOP IN LOOPMODE.
The selected testloop between SETPX (LABEL) and the following call of the loop subroutine is performed continuously.
This facility is useful both to develop a specific error and to adjust the delays in the bus-interface:

```
T  4        ; INIT REG
0  ADDRESS (SIZEM)
                ; START IN LOOPMODE:
```

The microprogram, between SETP1 and the call of the loop (LABEL: LOOP) is now performed.
In the loop is a zero written in address 8.

Pass Indication

In both the CPU and in the memory test the TCP bell is activated once.

Break

The test is breaked if any command is typed at the TCP.

4. Messages.

4.1 Error Messages.

4.1.1 Operating via OCP.

In case of an error, the autoload lamp will begin to gleam with a period of 0.5 sec.

4.1.2 Operating via TCP.

In case of an error the following text will be displayed:

```
ERR
C(WRK0)                    ; error type, see 4.1.3
C(WRK1)
C(WRK2)
C(PC)
```

4.1.3 Error Types.

WRK0	TYPE	WRK1	WRK2	PC
-1	illegalt interrupt	ILEVEL	./.	./.
-2	buserror	0 TIMEOUT 1 NACK 2 PARITY	./.	ADDRESS
0	error in 1. access to memory	expected data	READ DATA	memory address
1	error in MEM size routine	do.	do.	do.
2	error in MEM test	do.	do.	do.
10	test of IMOPREG.	the cont. of IMOP	expect- ed data	the U-address in IMOP was loaded
100 - 234	test of ALU and SCRATCH-PAD REGS. Look in the U-program listing for further in-formation.	the cont. of the SEL. register	expect- ed data	./.
	Test of EXT interrupt	./.	the simu- lated in- terrupt level	./.
300	Timeout when accessing CPU, ILEV 0-7		do.	

WRK0	TYPE	WRK1	WRK2	PC
301	Interrupt occurred with ILEV 0-7.		do.	
310	Timeout when accessing CPU, ILEV 8-15.		do.	
311	Interrupt not occurred when ILEV 8-15.	Received interrupt level	do.	
312	The rec ILEV and simulated ILEV are not equal.	do.	do.	
313	The interrupt simulated can not be cleared.	do.	do.	

4.1.4 Action after error. 4.1.4

It is possible to proceed in 3 ways:

1) Type P at the TCP: Proceed the test.
2) Type N at the TCP: Loop between the previous label SETPX and the call of loop. The subsequent error messages are suppressed.
3) Type a TCP command : The test is breaked and the command is executed.

4.2 Test Synchronization. 4.2

By means of the test sync signal (pos. 120, pin 19), it is possible to synchronize e.g. a scope.

The signal is set at the label SETPX and reset in loop subroutine.

5. Program Description. 5.

5.1 CPU Test. 5.1

The CPU test consists of a number of independent test loops. In every loop it is possible to loop.

A. Test of IMOP register.
The immediate operand register is loaded with a shifting 'zero', and checked.

B. Test of the ALU registers and scratchpad files.
A one is shifted through both register files. The ALU reg stack is read both from the A- and B-files.

Note: WRK0 and STAT registers are not tested.

C. Test of external interrupt.
The test checks that interrupt level 0-7 do not set interrupt. It is verified that interrupt level 8-15 set the proper interrupt level. The test also checks that the proper interrupt level is able to clear the simulated external interrupt.

5.2 Memory Test. 5.2

5.2.1 Normal Mode. 5.2.1

The total number of tests performed by the memory test is 12 * B * N * MEMSIZE, where N = number of address bits, and B the number of data bits in each word.

After the program is started the memory is loaded with 0's. The N address bit indicates up to 2 ** N = MEMSIZE words in the memory, each address is read and verified to be all 0's; then a single 1 is substituted

in one bit position, and the altered word is written back in the same location; and finally the word is re-read to verify that the newly entered 1 is still there. This procedure is repeated for each of the 24 data bits per word, accounting thus far for 3 * MEMSIZE * 24 cycles, leaving the memory filled with 1's. This again is repeated, plugging 0's back into the memory; and re-peated twice more with 0's and 1's, but reading the words in reverse order from size down to 8. At this point the number of cycles is 12 * MEMSIZE * 24. The remaining factor of N is accounted for by repeating this whole series of tests N times, using a different bit for incrementing through all possible addresses. This has the effect of incrementing through all the addresses by 2's, by 4's, by 8's (address bit 23 is a parity bit), and so on.

5.2.2 Short Mode. 5.2.2

In this version the 48 data patterns is reduced to two alternative patterns:

101010.....10, and
010101.....01.

6. Functionality. 6.

6.1 CPU Test. 6.1

The CPU test for stuck-at and bridge failures in the IMOP, ALU and scratchpad registers.

The ability of the hardware to handle external inter-rupts is also tested.

Note: The main part of the hardware has to function properly before it is possible to communicate with the TCP.

6.2 Memory Test. 6.2

The microprogrammed test can test for address selection failures caused by fixed address bit, uni- or bidirectional coupling between address bits, or an address bit that does not work in one state. Is also tests for fixed memory bits, uni- or bidirectional coupling between different bits, and uni- or bidirectional coupling between rows or columns in the memory matrix.

7. Algorithm, Flow Diagram. 7.

7.1 The Memory Algorithm, Normal Mode. 7.1

```
PC:=8;  DATA: = 8;
WHILE -, BUSERROR DO
BEGIN          COMMENT  SIZE THE MEMORY;
   WRITE (PC,DATA);
   READ (PC,DATI);
   IF DATA <> DATI THEN GOTO MEMEND;
   PC: = DATA : = PC+2;
END;

MEMEND:
   WRK1: = PC;
   WHILE WRK1 > 8 DO
   BEGIN         COMMENT  CLEAR MEMORY;
      WRK1: = WRK1 - 2;
      WRITE (WRK1,0);
   END;

   W3: = PC - 2;   COMMENT  LAST LOC IN MEMORY;

   OLDDATA: = 0;  NEWDATA: = 0;  Q: = 2;
   START: = 8;    STOP: = SIZE;
```

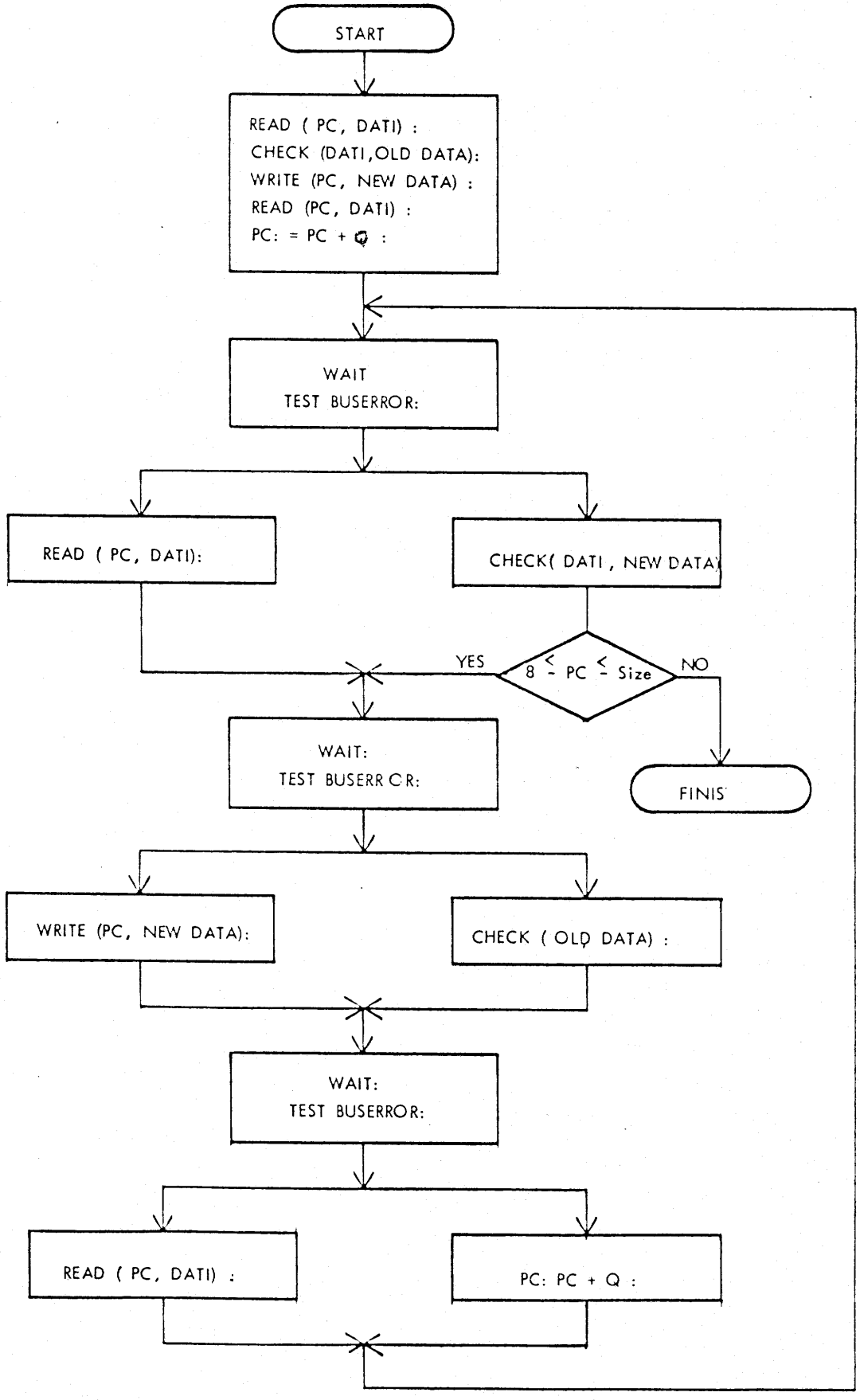
```
A01:    WHILE Q < SIZE DO
A02:    BEGIN
           FOR K: = 1,2 DO
           BEGIN
A03:          FOR J: = 0 STEP 1 UNTIL 47 DO
              BEGIN
A04:             NEWDATA: = NEWDATA SHIFT 1 ADD (IF
                 J < 24 THEN 1 ELSE 0);
A10:             FOR START: = START STEP  IF Q > 0
                 THEN 2 ELSE -2 UNTIL IF Q > 0 THEN
                 (START-8) > Q ELSE (SIZE-START) > Q
                 DO
                 FOR PC: = START STEP Q
                 UNTIL STOP DO
                 BEGIN
A20:                READ (PC,DATAIN);
                    IF DATAIN <> OLDDATA
                    THEN ERROR;
                    WRITE (PC,NEWDATA);
                    READ (PC,DATAIN);
                    IF DATAIN <> NEWDATA
                    THEN ERROR;
                 END PC;
FINIS:           OLDDATA: = NEWDATA;
              END J;
              START: = SIZE;  STOP: = 8;   Q: = -Q;
           END K;
           Q: = (-Q)  SHIFT 1;
           START: = 8;   STOP: = SIZE;
        END WHILE;
```

see flow diagram

7.2 FLOW DIAGRAM

A 10 :

A20 :



8. Appendix. 8.

The execution time for the memory test is proportional to $N*\log_2 N$, where N is the memory size.

In the table below the approximate execution time for the test run in a core-memory is shown:

Approximate execution time for the memory test

Memory size (words)	Short mode (min:sec)	Normal mode (min:sec)
32Kw	0:05	2:10
64Kw	0:11	5:37
96Kw	0:17	7:11
128Kw	0:23	10:49
160Kw	0:29	13:31
192Kw	1:35	15:14
224Kw	1:42	18:00
256Kw	1:48	21:48