RCSL No: 52-AA894

Edition: March 1979

Author: Bjarne Jørgensen

Title:

General Information
Hardwired Testprograms and Program Load to RC 3803
Revision 0

RCSL 42-11592

A/S REGNECENTRALEN
af 1979

Keywords:

RC 3803, CPU 720, Program Load, F10A

Abstract:

This paper describes hardwired testprograms for 3803 and program load bootstrap loaders to peripheral units as Paper Tape Reader, Card Reader, Flexible Disc, Mag. tape, Cartridge Disc and Disc Storage Module.

(28 pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL 42-11592

CONTENTS PAGE

1. SCOPE..... 1

2. OPERATING PROCEDURE..... 4

3. PROGRAM DESCRIPTION..... 6
3.1 Initialization program..... 6
3.2 Memory Test..... 6
3.3 Console Interface Exerciser..... 7
3.4 General Program Load..... 7
3.5 Card Reader Program Load..... 8
3.5.1 Operating procedure..... 8
3.5.2 Input card format..... 9
3.6 Flexible disc program load..... 9
3.6.1 Operating procedure..... 9
3.7 Cartridge disc program load.....10
3.7.1 Operating procedure.....10
3.7.2 Program description.....10
3.8 Disc Storage Modul program load.....11
3.8.1 Operating Procedure.....11
3.8.2 Programming considerations.....11

Appendix A: Program Listing - Initialization.....13
Appendix B: Program Listing - Memory test.....14
Appendix C: Program Listing - Console Exerciser.....15
Appendix D: Program Listing - General P. load.....16
Appendix E: Program Listing - Card Reader P. load...17
Appendix F: Program Listing - Flexible disc P. load.18
Appendix G: Program Listing - Cartridge disc P.load 19
Appendix H: Program Listing - Disc storage module P. load.....20

This paper describes a collection of hardwired diagnostic programs and program load programs for CPU 720.

Each program is 32 word and the programs are stacked in the ROM's (ROM 790 and ROM 791) placed in POS. 81 and 71. In a table-ROM (ROM 789) is established a relationship between the Device No. (set on the front panel of CPU 720 or on the TCP) and the program module stack.

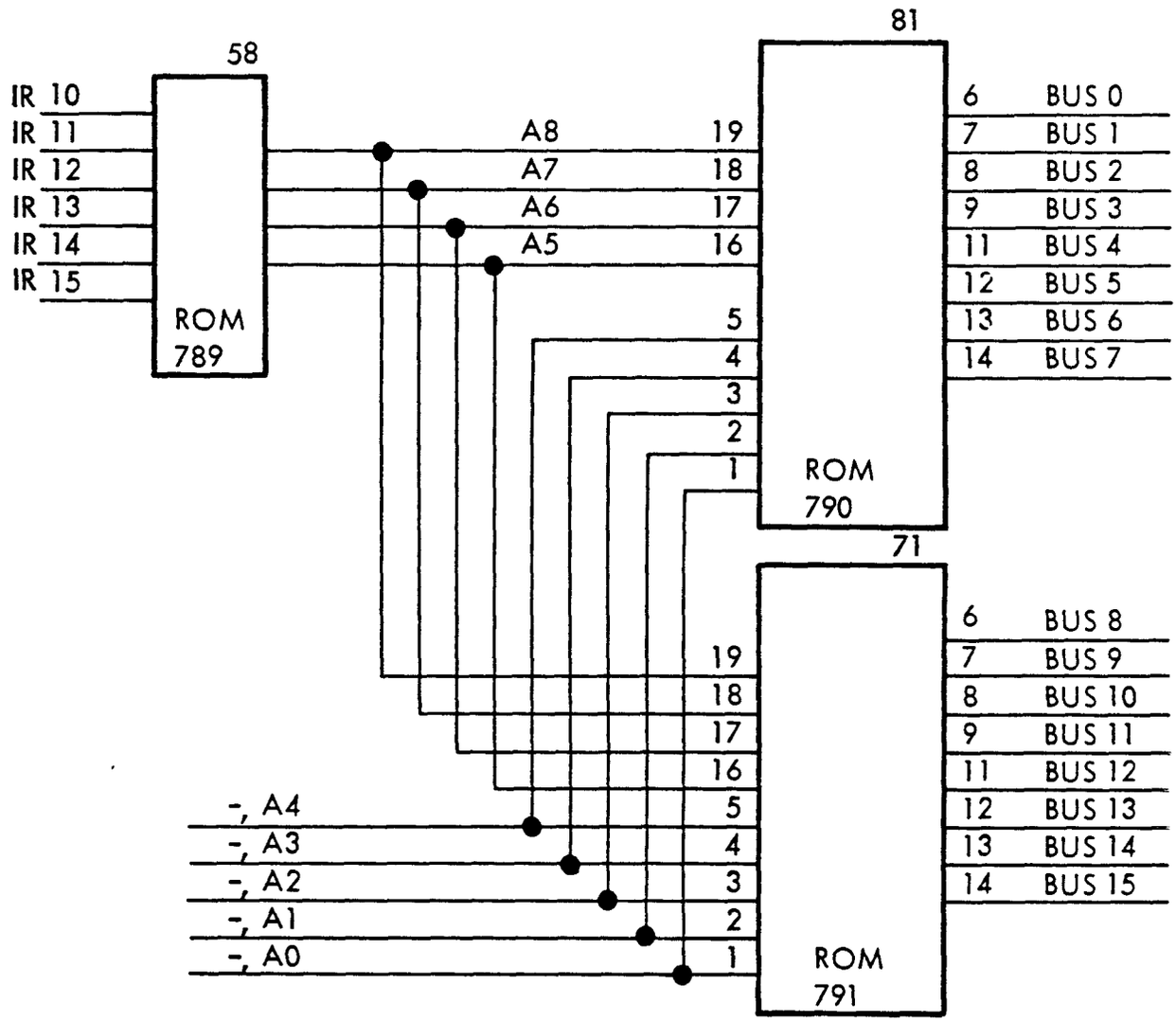
On fig. 1.0 is shown the layout of the autoload rom's.

On fig. 1.1 is shown a blockdiagram of the hardware logic. The selected program is loaded into core by the procedure described in the next section. When the loading is finished, the program is started at location 0.

ADDRESS (OCTAL)	ROM BIT 0-7 / ROM BIT 8-15	MODULE
0 37	CONSOLE INITIALIZATION SECTION 3.1	0
40 77	MEMORY TEST SECTION 3.2	1
100 137	CONSOLE EXERCISER SECTION 3.3	2
140 177	GENERAL PROGRAM LOAD SECTION 3.4	3
200 237	CARD READER P. LOAD SECTION 3.5	4
240 277	FLEXIBLE DISC P. LOAD SECTION 3.6	5
300 337	CARTRIDGE DISC P. LOAD SECTION 3.7	6
340 377	DISC STORAGE MODULE P. LOAD SECTION 3.8	7
400 777	Spare area for 8 program modules of 32 words	

ROM 790 - 791 LAYOUT

Fig. 1.0



-, A0 least significant addr.

AUTOLOAD CIRCUIT

Fig. 1.1

1. Set device code in data switches 10-15 in accordance to the table on fig. 2.0.

2. Set data switch 0 in accordance to fig. 2.0.

3. Make the load medium ready, i.e. the tape on magnetic tape station must be on line.

4. Start the program load: Press Autoload or Autoload-load on the operator panel, depending on which one is used.

DEVICE NO. (OCTAL)	BIT 0	PROGRAM MODULE NO.	PROGRAM MODULE FUNCTION
0	x	0	CONSOLE INITIALIZATION (BAUD RATE - NO. OF STOP BITS AND MEMORY RESET
1	x	1	MEMORY TESTPROGRAM
2	0	2	CONSOLE ECHO PROGRAM
2	1	2	CONSOLE CHARACTER GENERATOR
3-15 17 21-55 57-60 62-72 74-77	0	3	STANDARD AUTOLOAD LOW SPEED DEVICE
3-15 17 21-55 57-60 62-72 74-77	1	3	STANDARD AUTOLOAD DATA CHANNEL PROGRAM LOAD
16	x	4	CARD READER PROGRAM LOAD
56	x	4	CARD READER PROGRAM LOAD
61	0	5	FLEXIBLE DISC PROGRAM LOAD
61	1	5	NO FUNCTION
73	0	6	DISC PROGRAM LOAD (incl. a Disc recalibration)
73	1	6	DISC PROGRAM LOAD (no recalibration)
20	0	7	Disc Storage Module PROGRAM LOAD

RELATION BETWEEN DEVICE NO.
(SET ON THE FRONT PANEL OF CPU 720 OR AT THE TCP)
AND THE SELECTED PROGRAM MODULE

ROM 789

Fig. 2.0

In the following sections each program is further described.

INITIALIZATION PROGRAM (MODULE 0) 3.1

This program writes correct parity into all locations of CPU 720 memory (C(ADDR):= ADDR).

This program can be loaded in two manners:

a) After power-up reset of CPU 720 (dataswitches 0, 10-15 are don't care).

b) As described in section 2 with dataswitches (0, 10-15):= (x,0).

The program listing is shown in Appendix A.

MEMORY TEST (MODULE 1) 3.2

This program is a maintenance program designed to detect malfunctions in the memory address selection logic.

The program fills memory with an address pattern (C(ADDRESS) = ADDRESS + CONSTANT). SUCCESSful read back of the pattern is proof that all locations exist.

After each run CONSTANT:= CONSTANT + 1.

If the TCP is connected, the type of error can be examined when the program halt.

This program is started as described in section 2 with dataswitches (0, 10-15):= (x,1).

The program listing is shown in Appendix B.

CONSOLE INTERFACE EXERCISER (MODULE 2) 3.3

This program module contains an ECHO program and a CHARACTER GENERATOR program.

The ECHO program transmits all characters, received on the console input (TTI), on the console output (TTO).

The CHARACTER GENERATOR program transmits to the console line by line the following sequence:

space, ! ------ CR NL

80 characters

The ECHO program is started as described in section 2 with:

dataswitches (0, 10-15):= (0,2).

The CHARACTER GENERATOR program is started as described in section 2 with:

dataswitches (0, 10-15):= (1,2).

The program listing is shown in Appendix C.

GENERAL PROGRAM LOAD (MODULE 3) 3.4

The bootstrap loader reads the data switches (0, 10-15), sets up its own I/O instructions with the specified device code, and then performs a program load procedure depending upon the state of data switch 0.

If the switch is a 1, the bootstrap loader starts the device for data channel storage beginning at location 0 and then loops at location 377_8 until a data channel transfer places a word into that location.

After a word has been placed in location 377_8, it is executed as an instruction. Typically, this word is a JUMP into the data that the data channel has placed in the first 377_8 memory locations.

If data switch 0 is a 0, the bootstrap loader reads the loader program via programmed I/O. The device must supply 8-bit data bytes, and each pair of bytes is stored as a single word in memory; wherein the first and second bytes read become the left and right halves of the word. To simplify the positioning of the tape in the reader, the bootstrap loader ignores leading null characters. It does not begin storing any words until it reads a non-zero synchronization byte. The first word following this synchronization byte must be the negative of the total number of words to be read, including the first word. The number of words to be read, including the first word may not be greater than 192_{10}. The bootstrap loader stores these words beginning at memory location 100_8. After storing the last word read, it transfers control to that location.

The program listing is shown in Appendix D.

CARD READER PROGRAM LOAD (MODULE 4) 3.5

OPERATING PROCEDURE 3.5.1

As described in section 2:

Press RESET on the Card Reader.

INPUT CARD FORMAT 3.5.2

The card must only contain one 8 bit byte in each column. The byte must be in row 2 through row 9. Row 2 contains the most significant bit.

Column 1 contains left byte word 1.
Column 2 contains right byte word 1.
Column 3 contains left byte word 2.
etc. etc.

Column 77 contains left byte of the 16 bit checksum of the card.
Column 78 contains right byte of the 16 bit checksum of the card.
Column 79 contains left byte of the card number.
Column 80 contains right byte of the card number.

The card number must be 1.

The program listing is shown in Appendix E. 3.5.3

FLEXIBLE DISC PROGRAM LOAD (MODULE 5) 3.6

OPERATING PROCEDURE 3.6.1

a. Set dataswitch (0) = 0 and switch (10:15) = 61_8.

b. Press LOAD-AUTOLOAD if F11 is present
   or AUTOLOAD if F19 is present.

Now the program will be loaded into the core and is started in location 0. The program waits until the device is ready and on-line.

The program listing is shown in Appendix F. 3.6.2

CARTRIDGE DISC PROGRAM LOAD (MODULE 6) 3.7

The Disc Program Load program is a 32-word program.

The program is loaded into core by the procedure described below. When the loading is finished, the program is started at location 0.

The Disc Program Load is capable of loading from various high-speed devices.

OPERATING PROCEDURE 3.7.1

1. Set device code in dataswitches 10-15.
2. Set dataswitch 0 if load medium is another high-speed device than Disc.
3. Dataswitches 1-9 are Don't-Cares.
4. Make the load medium ready, i.e. the tape on a magnetic tape station must be on line.
5. Start the Program Load: Press Autoload or Autoload-Load on the operator panel, depending on which one is used.

PROGRAM DESCRIPTION 3.7.2

If dataswitch 0 is reset, a Disc recalibration is made. The program starts the load medium and makes a jump to location octal 377, which contains a jump to location 377, i.e. when the load has overwritten location 377, the new instruction is executed.

The program listing is shown in Appendix G. 3.7.3

DISC STORAGE MODULE PROGRAM LOAD (MODULE 7) 3.8

The Disc Storage Module Program Load is a 32-word program.

The program is loaded into the core by the procedures given in section 2, and is, when loaded, started in location 0.

The program is designed to use the Disc Storage Module as load medium.

OPERATING PROCEDURE 3.8.1

a. If the load medium is the Disc Storage Module then set dataswitch (0) = 0, and dataswitches (1:15) = 20.

   Disc Storage Module need not be ready as the program will wait for ready state.

b. Press LOAD-AUTOLOAD if F11 is present
   or AUTOLOAD if F19 is present.

Now the program will be loaded into core and started in location 0.

PROGRAMMING CONSIDERATIONS 3.8.2

When autoload is done from Disc Storage Module, the ROM program starts transfer of 16 sectors to memory from location 0.

The read segments must contain a Channel-program stop command in word 25 (oct.). The stop command has the value 216B7 + 7 (dec.).

After the transfer status from the Disc Controller is placed in word (0:2), and the read code must therefore reserve these locations for status.

During transfer the CPU is looping in location 376-377 (octal) executing instructions:

```
DOA 1    20
JMP  .   .-1
```

with AC1 = 0.

until the disc is ready and later overwritten by the data.

The program listing is shown in Appendix H.

```
  0001   .MAIN
01
02
03                ;
04                ; 78.04.03 KNEH.
05                ; PROGRAM TO INITIALIZE CPU711 CONSOLE
06                ; INTERFACE AND TO WRITE CORRECT PARITY
07                ; IN ALL MEMORY LOCATIONS.
08
09        000000          .LOC    0
10
11 00000 060477           DIA     0,CPU     ; READ SWITCHES<BAUD RATE
12 00001 101220           MOVZR   0,0       ; AND NO. OF STOP BITS>.
13 00002 024033           LDA     1,MODE
14 00003 107000           ADD     0,1
15 00004 066011           DOB     1,TTO     ; SET MODE1
16 00005 101300           MOVS    0,0
17 00006 024034           LDA     1,C17
18 00007 107620           ANDZR   0,1
19 00010 030035           LDA     2,C8
20 00011 133000           ADD     1,2
21 00012 025000           LDA     1,0,2
22 00013 030033           LDA     2,MODE
23 00014 125002           MOV     1,1,SZC
24 00015 125300           MOVS    1,1
25 00016 147300           ADDS    2,1
26 00017 066011           DOB     1,TTO     ; SET MODE2
27 00020 024036           LDA     1,COMN
28 00021 067011           DOC     1,TTO     ; SET COMMAND
29 00022 030035           LDA     2,C40     ; RESET MEM
30 00023 051000           STA     2,0,2
31 00024 151404           INC     2,2,SZR
32 00025 000023           JMP     .-2
33 00026 062677           IORST
34 00027 007402 BUADT:    007402            ; 19200BPS,110BPS.
35 00030 002406           2406              ; 300BPS,600BPS.
36 00031 004012           4012              ; 1200BPS,2400BPS.
37 00032 006016           6016              ; 4800BPS,9600BPS.
38
39 00033 030116 MODE:     30116             ; MODE2,MODE1.
40 00034 000017 C17:      17
41 00035 000027 C8:       BUADT
42 00036 000047 COMN:     47                ; COMMAND.
43        000035 C40=     C8
44
45                        .END
  0002   .MAIN

BUADT    000027       1/34     1/41
C17      000034       1/17     1/40
C40      000035       1/29     1/43
C8       000035       1/19     1/41     1/43
COMN     000036       1/27     1/42
MODE     000033       1/13     1/22     1/39
```

```
  0001   .MAIN
01
02
03                ;
04                ; 78.04.05 KNEH.
05                ; MEMORY TEST.
06                ; THE PROGRAM RUN WITHOUT ERROR AS LONG AS
07                ; THE RUN INDICATOR ON THE FRONT PANEL IS LIT.
08                ; THIS IS ONLY A GO/NO GO TEST,I.E IT CAN ONLY
09                ; BE USED AS DIAGNOSTIC PROGRAM IF THE DEBUG UNIT
10                ; IS CONNECTED.
11
12        000000            .LOC    0
13 00000  000004            JMP     START
14 00001  000100  STADD:    100               ; TEST START ADDRESS.
15 00002  100000  STOPADD:  100000            ; TEST LAST+1 ADDRESS.
16 00003  000000  CONST:    0
17
18 00004  030002  START:    LDA     2,STOPADD
19 00005  034001            LDA     3,STADD ; WRITE SECTION.
20 00006  024003            LDA     1,CONST
21 00007  161000  E1:       MOV     3,0
22 00010  123000            ADD     1,0
23 00011  041400            STA     0,0,3
24 00012  175400            INC     3,3
25 00013  172414            SUB#    3,2,SZR
26 00014  000007            JMP     E1
27 00015  034001            LDA     3,STADD ; READ CHECK.
28 00016  024003  E2:       LDA     1,CONST
29 00017  161000            MOV     3,0
30 00020  123000            ADD     1,0
31 00021  025400            LDA     1,0,3
32 00022  122414            SUB#    1,0,SZP ; AC0=WANTED RESULT.
33 00023  063077            HALT              ; AC1=READ RESULT.
34 00024  175400            INC     3,3       ; AC3= ERROR ADDRESS.
35 00025  172414            SUB#    3,2,SZR
36 00026  000016            JMP     E2
37 00027  010003            ISZ     CONST
38 00030  000004            JMP     START
39 00031  000004            JMP     START
40
41                          .END
  0002   .MAIN

CONST   000003      1/16    1/20    1/28    1/37
E1      000007      1/21    1/26
E2      000016      1/28    1/36
STADD   000001      1/14    1/19    1/27
START   000004      1/13    1/18    1/38    1/39
STOPA   000002      1/15    1/18
```

```
                        ; 78.04.05 KEH
                        ; PROGRAM TO TEST THE CONSOLE INTERFACE.
                        ; SWITCH0= 1 SELECTS A PROGRAM GENERATING
                        ; 80CHARACTERS/LINE STARTING WITH "SPACE".
                        ; SWITCH0= 0 SELECTS AN ECHO PROGRAM,
                        ; TRANSMITTING ON THE TTO EACH CHARACTER RECEIVED
                        ; ON THE TTI.

        000000                  .LOC    0
00000   006477                  DIA     0,CPU           ; READ SWITCHES.
00001   101102                  MOVL    0,0,SZC         ; TEST THE STATE OF SWITCH0.
00002   000011                  JMP     PATT
00003   060110  ECHO:   NIOS    TTI
00004   063610          SKPDN   TTI
00005   000004          JMP     .-1
00006   060610          DIAC    0,TTI
00007   004025          JSR     OUT             ; OUTPUT THE RECEIVED
00010   000003          JMP     ECHO            ; CHARACTER.

00011   020031  PATT:   LDA     0,C80           ; SET CHAR COUNTER.
00012   040000          STA     0,0
00013   020034          LDA     0,CSPACE
00014   004025  LOOP:   JSR     OUT
00015   101400          INC     0,0
00016   014000          DSZ     0
00017   000014          JMP     LOOP
00020   020033          LDA     0,CCR
00021   004025          JSR     OUT             ; OUTPUT NEW LINE.
00022   020032          LDA     0,CNL
00023   004025          JSR     OUT             ; OUTPUT CARRIAGE RETURN.
00024   000011          JMP     PATT

00025   061111  OUT:    DOAS    0,TTO           ; ROUTINE TO OUTPUT
00026   063611          SKPDN   TTO             ; A CHARACTER ON TTO.
00027   000026          JMP     .-1
00030   001400          JMP     0,3

00031   000120  C80:    120
00032   000012  CNL:    12
00033   000015  CCR:    15

00034   000040  CSPACE: 40

                        .END
  0002  .MAIN

C80     000031      1/22    1/40
CCR     000033      1/29    1/42
CNL     000032      1/31    1/41
CSPAC   000034      1/24    1/44
ECHO    000003      1/15    1/20
LOOP    000014      1/25    1/28
OUT     000025      1/19    1/25    1/30    1/32    1/35
PATT    000011      1/14    1/22    1/33
```

The Program Load Program

The program for the automatic program load is listed below.

```
        000000  .LOC    0

000000  060477  BEG:    READS   0       ;READ SWITCHES INTO AC0
000001  105120          MOVZL   0,1     ;ISOLATE DEVICE CODE
000002  124240          COMCR   1,1     ;-DEVICE CODE - 1

000003  010011  LOOP:   ISZ     OP1     ;COUNT DEVICE CODE INTO ALL
000004  010031          ISZ     OP2     ;IO INSTRUCTIONS
000005  010033          ISZ     OP3
000006  010014          ISZ     OP4
000007  125404          INC     1,1,SZR ;DONE?
000010  000003          JMP     LOOP    ;NO INCREMENT AGAIN

000011  060077  OP1:    060077          ;START DEVICE; (NIOS 0) - 1
000012  030017          LDA     2,C377  ;YES, PUT JMP 377 INTO LOCATION 377
000013  050377          STA     2,377
000014  063377  OP4:    063377          ;BUSY?: (SKPBN 0) - 1
000015  000011          JMP     OP1     ;NO, GO TO OP1
000016  101102          MOVL    0,0,SZC ;LOW SPEED DEVICE? (TEST SWITCH 0)
000017  000377  C377:   JMP     377     ;NO, GO TO 377 AND WAIT FOR CHANNEL

000020  004031  LOOP2:  JSR     GET+1   ;GET A FRAME
000021  101065          MOVC    0,0,SNR ;IS IT NONZERO?
000022  000020          JMP     LOOP2   ;NO, IGNORE AND GET ANOTHER

000023  004030  LOOP4:  JSR     GET     ;YES, GET FULL WORD
000024  046027          STA     1,@C77  ;STORE STARTING AT 100
000025  010100          ISZ     100     ;COUNT WORD - DONE?
000026  000023          JMP     LOOP4   ;NO, GET ANOTHER
000027  000077  C77:    JMP     77      ;YES - LOCATION CUONTER AND
                                        ;JUMP TO LAST WORD
000030  126420  GET:    SUBZ    1,1     ;CLEAR AC1, SET CARRY
        000000  OP2:
000031  063577  LOOP3:  063577          ;DONE?: (SKPDN 0) - 1
000032  000031          JMP     LOOP3   ;NO, WAIT
000033  060477  OP3:    060477          ;YES, READ INTO AC0: (DIAS 0,0) - 1
000034  107363          ADDCS   0,1,SNC ;ADD 2 FRAMES SWAPPED - GOT SECOND?
000035  000031          JMP     LOOP3   ;NO, GO BACK AFTER IT
000036  125300          MOVS    1,1     ;YES, SWAP AC1
000037  001400          JMP     0,3     ;RETURN WITH FULL WORD

        .END
```

770222 ERC

TAPE PROGRAM LOAD

```
                    ; CARD READER PROGRAM LOAD II
                    ;
                    ; FOR LOADING OF PROGRAMS FROM A CARD READER CONNECTED
                    ; TO CRC 705 OR EQUIVALENT.
                    ;
           000000   .LOC    0
                    ;
00000 020006        SA:     LDA     0,SA    ; GET ADDRESS FOR CARD BUFFER
00001 004007                JSR     GETCD   ; GET ONE CARD
00002 004022                JSR     CONV    ; CONVERT ONE CARD TO WORDS
00003 020110                LDA     0,110   ; GET CARD NUMBER
00004 142004                ADC     2,0,SZR ; CHECK FOR CARD NUMBER 1
00005 063077                HALT            ; CARD NUMBER ERROR
00006 000041                JMP     41      ; GO TO PRE-LOADER
                    ;
00007 062016        GETCD:  DOB     0,CDR   ; OUTPUT BUFFER ADDRESS
00010 061116                DOAS    0,CDR   ; OUTPUT READ COMMAND
00011 063416                SKPBN   CDR     ; CHECK FOR THE READER IS STARTED
00012 000010                JMP     .-2     ; NO, TRY AGAIN
00013 063516                SKPBZ   CDR     ; WAIT FOR COMPLETATION OF
00014 000013                JMP     .-1     ; READING A CARD
00015 001400                JMP     0,3     ; RETURN
                    ;
                    ; VARIABLES
                    ;
00016 000000        COUNT:  0
00017 177730        M50:    -50
00020 000040        ADDR1:  40
00021 000040        ADDR2:  40
                    ;
                    ;CONVERTING SUBROUTINE
                    ;
00022 152400        CONV:   SUB     2,2     ; CLEAR AC2
00023 020017                LDA     0,M50   ; GET COUNT FOR NUMBER OF WORDS
00024 040016                STA     0,COUNT ; STORE COUNT
00025 022020        LOOP:   LDA     0,@ADDR1; GET LEFT BYTE
00026 101300                MOVS    0,0     ; SWAP BYTE
00027 026020                LDA     1,@ADDR1; GET RIGHT BYTE
00030 107000                ADD     0,1     ; COMPUTE WORD
00031 046021                STA     1,@ADDR2; STORE WORD
00032 133000                ADD     1,2     ; COMPUTE CHECKSUM
00033 010016                ISZ     COUNT   ; CHECK FOR MORE WORDS
00034 000025                JMP     LOOP    ; YES, GET THEM
00035 151004                MOV     2,2,SZR ; CHECK FOR CHECKSUM ERROR
00036 063077                HALT            ; CHECKSUM ERROR
00037 001400                JMP     0,3     ; RETURN

                    .END
```

```
  0001   .MAIN
                    ; PROGRAM LOAD, FLEXIBLE DISC, HKM 75.11.01
                    ;  THIS PROGRAM LOAD RESIDES IN 32*16 ROM.
                    ;  IT IS DESIGNED FOR FLEXIBLE DISC AS PRIMARY LOAD MEDIUM
                    ;   . AND USES MOVING HEAD DISC OR MAGTAPE AS SECONDARY
                    ;     LOAD MEDIUM.
                    ;
                    ;  FLEXIBLE DISC: SWITCH(0) = 0, SWITCH(1:15) = NOT USED,
                    ;                 THE DISC IS RECALIBRATED BY THE PROGRAM.
                    ;  MAGTAPE:
                    ;  MOVING HEAD DISC: SWITCH(0) = 1, SWITCH(1:9) = 0,
                    ;                 SWITCH(10:15) = DEVICE NUMBER,
                    ;                 BOTH DISC AND MAGTAPE MUST BE RECALIBRATED
                    ;                 BEFORE ACTIVATING THE PROGRAM LOAD.
                    ;
                    ;  IN CASE OF MAGTAPE OR FLEXIBLE DISC THE LOAD WAITS UNTIL
                    ;     THE SELECTED DEVICE IS READY FOR COMMANDS.
          000000    .LOC    0
          000061    FLEX=   61                  ; FLEXIBLE DISC
00000 070477        READS   2                   ;00; READ SWITCHES(S);
00001 150122        COMZL   2,2 SZC             ;01; IF S(0) = 0 THEN
00002 000026        JMP         FD              ;02;    CARRY:= TRUE AND GOTO FLOP
00003 151240        MOVOR   2,2                 ;03; NOT FLOPPY: DEVICE:= OCT(77);
00004 010010 LOOP:  ISZ         OP1             ;04; FOR DEVICE INDEX:= -S(1:15)-1
00005 010013        ISZ         OP2             ;05;      STEP 1 UNTIL 0 DO
00006 151404        INC     2,2 SZR             ;06;      DEVICE:= DEVICE + 1;
00007 000004        JMP         LOOP            ;07; IFOR FURTHER COMMENTS SEE OP1,
00010 071077 OP1:               071077          ;10; DOAS 2 <DEV> - 1: INCREMENTS :
                                                ;    189 (S(TART)) + S(10:15);
00011 024015        LDA     1   .377           ;11; LOAD "JMP  .+0" INTO LAST WOR
00012 044377        STA     1   377             ;12;   OF PAGE ZERO;
00013 063377 OP2:               063377          ;13; SKPBN  <DEV> - 1: INCREMENTS :
                                                ;    489            + S(10:15);
00014 000010        JMP         OP1             ;14; READ FIRST BLOCK, WAIT UNTIL
                                                ;   COMMAND IS ACCEPTED;
00015 000377 .377:  JMP         377             ;15; GOTO WAIT BLOCK TRANSFERED;

00016 126420 READN: SUBZ    1,1                 ;16; GETWORDS: WORD:=0; CARRY:=TRU
00017 061461        DIB     0   FLEX            ;17; READ(CHAR);
00020 107363        ADDCS   0,1 SNC             ;20; WORD:= WORD SHIFT 8 + CHAR;
                                                ;    CARRY:= -,CARRY;
00021 000017        JMP         READN+1         ;21; IF CARRY = FALSE THEN READ CH
00022 046025        STA@    1   ADR             ;22; INCR(ADR); CORE(ADR):= WORD;
00023 010100        ISZ         100             ;23; IF INCR(CORE(100)) <> 0 THEN
00024 000016        JMP         READN           ;24;    READ NEXT WORD ELSE
00025 000077 ADR:   JMP         77              ;25;    GOTO ADR;

                    ; FLEXIBLE DISC: AT ENTRY, CARRY == TRUE!!
00026 030037 FD:    LDA     2   COMM            ;26; FLOPPY: COMMAND:= RECALIBRATE,
00027 071161 EXE:   DOAS    2   FLEX            ;27; EXECUTE: EXECUTE(COMMAND);
00030 063461        SKPBN       FLEX            ;30; ! COMMAND(0:7) = DONT CARE !
00031 000027        JMP         EXE             ;31; WAIT UNTIL COMMAND IS ACCEPTE
00032 063661        SKPDN       FLEX            ;32; WAIT UNTIL COMMAND IS EXECUTE
00033 000032        JMP         .-1             ;33;
00034 151102        MOVL    2,2 SZC             ;34; IF NEXT COMMAND = READ BLOCK
00035 000027        JMP         EXE             ;35;    THEN GOTO EXECUTE ELSE
00036 000016        JMP         READN           ;36;    GOTO GETWORDS;
00037 101000 COMM:              1B0+1B6         ;37; COMMAND BITS;
                    .END
```

```
                ;    PROGRAM LOAD FROM DISK
                ;    AND OTHER HIGH SPEED DEVICES


        000000  .LOC    0

        000000  DEV=    0

00000 006477  BEG:    READS  1              ; READ SWITCHES
00001 020037          LDA    0    C77       ; ISOLATE DEVICE CODE
00002 123400          AND    1,0
00003 100404          NEG    0,0 SZR
00004 010031  LOOP:   ISZ           OP1     ; COUNT DEVICE CODE
00005 010032          ISZ           OP2     ; INTO ALL IN/OUT
00006 010022          ISZ           OP3     ; INSTRUCTIONS
00007 010025          ISZ           OP4
00010 101404          INC    0,0 SZR        ; DONE
00011 000004          JMP           LOOP    ; NO, INCREMENT AGAIN
00012 125102          MOVL   1,1 SZC        ; DISK
00013 000022          JMP           OP3     ; NO
00014 004030          JSR           SPEC    ; SEEK WITH CLEAR
00015 175000                        175000  ; SEEK INSTRUCTION
00016 004030          JSR           SPEC    ; RECALIBRATE WITH CLEAR
00017 175400                        175400  ; RECALIBRATE INSTRUCTION
00020 004030          JSR           SPEC    ; DISK READY
00021 175000                        175000
00022 006100  OP3:    DOAS   0      DEV     ; START DEVICE WITH
                                              ; RESET INSTRUCTION REGISTER
00023 030027          LDA    2      C377    ; SETUP JMP 377 IN
00024 050377          STA    2      377     ; LOCATION 377
00025 006400  OP4:    SKPBN         DEV     ; BUSY
00026 000022          JMP           OP3     ; NO, START AGAIN
00027 000377  C377:   JMP           377     ; YES, WAIT PAGE ZERO
                                              ; OVERWRITTEN


00030 025401  SPEC:   LDA    1      0,3     ; DISK ROUTINE
00031 006500  OP1:    DOAP   1      DEV     ; RECALIBRATE DISK
00032 006440  OP2:    DIA    1      DEV     ; READ STATUS
00033 151000          MOVS   1,2
00034 133405          AND    1,2 SNR        ; DONE
00035 000032          JMP           OP2     ; NO, WAIT
00036 001401          JMP           1,3     ; RETURN


00037 000077  C77:    77


                .END
```

```
                    ; PROGRAM LOAD, CDC DISC, HKM 77.01.04, REV.77.05.23/1;
                    ;   THIS PROGRAM RESIDES IN 32*2 ROM.
                    ;   IT IS DESIGNED FOR CDC MASS DISC AS PRIMARY LOAD MEDIUM
                    ;   AND USES DIABLO DISC AND MAGTAPE AS SECONDARY LOADS.
                    ;
                    ;   CDC DISC:  SWITCH(0) = 0, SWITCH(1:15) = NOT USED,
                    ;              THE DISC NEED NOT TO BE CONNECTED AND ON-LINE
                    ;              BEFORE LOADING.
                    ;              THE READ DATA MUST CONTAIN AN EXPLICIT STOP COMM
                    ;              IN WORD 25 (OCTAL) (REF. CHANNEL PROGRAM BELOW).
                    ;              MOREOVER THE STATUS IS PLACED IN WORD(0:2), I.E.
                    ;              THE READ SEGMENT SHOULD NOT CONTAIN INFORMATION
                    ;              IN THESE 3 WORDS.
                    ;
                    ;   DIABLO DISC: SWITCH(0) = 1, SWITCH(1:15) = DEVICE NUMBER,
                    ;              THE DEVICE MUST BE RECALIBRATED BEFORE LOADING.
                    ;
                    ;   MAGTAPE:   AS FOR DIABLO DISC.
                    ;
       000000       .LOC    0                  ;
                    ;
       071000       .OP1=   DOA     2,0        ; LOAD REGISTER;
       060100       .OP2=   NIOS    0          ; START HIGH SPEED;
       000020       MDISC=  20                 ; CDC DEVICE NUMBER;
                    ;
00000  070477               READS   2          ;00; Z:= -(READ SWITCHES) - 1;
00001  150123               COMZL   2,2,SNC    ;01; IF Z(0) = 1 !S(0) = 0! THEN
00002  000026               JMP     SECUN      ;02;BEGIN !CDC LOAD!
00003  120400               SUB     1,1        ;03;  CLEAR(DISC DRIVE0);
00004  065020       .DOA:   DOA     1,MDISC    ;04;
00005  024013               LDA     1,AHEAD    ;05;
00006  066020               DOB     1,MDISC    ;06;  SEND(CHANNEL PROG, HEAD ADDRESS);
00007  024004               LDA     1,.DOA     ;07;  WORD(376):=
00010  046037               STA@    1,.377     ;10;              DOA  1,MDISC;
00011  024025               LDA     1,.START   ;11;  ! CONST(377):= 376 !
00012  000035               JMP     N377       ;12;  GOTO SET 'WAIT-JMP';

       000012       .RDX    10                 ;
                    ; CHANNEL PROGRAM AREA FOR CDC PROGRAM LOAD;
00013  000020       AHEAD:  A00                ;13; HEAD OF CHANNEL PROGRAM;
00014  000000               0                  ;14; NOT USED;
00015  000000       ASEEK:  0          ;DUMMY  ;15; CYLINDER = 0;
00016  000000               0          ;DUMMY  ;16; HEAD<8+SECTOR = 0;
00017  154002       A01:    21687+2            ;17; SEEK;
00020  000017       A00:    A01        ;DUMMY  ;20; ADDRESS OF CHANNEL PROGRAM(DRIVE 0
00021  000015               ASEEK              ;21;  ADDR(SEEK PARAMETERS);
00022  154001               21687+1            ;22; READ DATA;
00023  020000               512*16             ;23;  BYTECOUNT = 512*16 = 16 SECTORS;
00024  000000               0                  ;24;  WORD ADDRESS = 0; ADDR(STATUS);
                                                ;   ! AFTER INPUT, THE DISC CONTROLLER
                                                ;     WILL DELIVER THE STATUS INFORMATI
                                                ;     INTO AN AREA OF 3 WORDS WHICH
                                                ;     ADDRESS IS IN WORD(24): WORD(0:2)
                    ;          21687+7          ;25; STOP !OPERATION MUST BE CONTAINED
                                                ;     IN THE READ SEGMENT AT EXACTLY
                                                ;     THIS POSITION!
00025  000004       .START: 1B13               ;25; COMMAND(START);
                                                ;   END !CDC LOAD! ELSE
```

```
.TTY
        000014  .RDX    5                       ;
        000000  SECOND:                         ;   BEGIN !SECONDARY LOAD DEVICES!
00026 151240            MOVOR   2,2             ;26;  REPEAT
  027 010033  LOOP:     ISZ     OP1             ;27;    INCR(DOA);
00030 010034            ISZ     OP2             ;30;    INCR(DIOS);
00031 151404            INC     2,2,SZR         ;31;    INCR(Z);
00032 000027            JMP     LOOP            ;32;  UNTIL Z = 0;
00033 070777  OP1:      .OP1-1                  ;33;  "DOA 2 0" - 1;
00034 060077  OP2:      .OP2-1                  ;34;  "DIOS  0" - 1;
00035 030037  N377:     LDA     2,.377          ;35;  LAST WORD OF SEGMENT AREA:=
00036 050377            STA     2,377           ;36;    JMP  .+0 !JMP  .-1 IN CASE
00037 000377  .377:     JMP     377             ;37;    OF CDC DISC LOAD!
                                                ;     GOTO WAIT DATA;
                                                ;   END !SECONDARY LOAD DEVICES!
                .END                            ;
```

RETURN LETTER

General Information
Title: Hardwired testprograms and program Load to RC 3803

RCSL No.: 52-AA894

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

_____

_____

_____

_____

Do you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

How can this manual be improved?

_____

_____

_____

_____

Other comments?

_____

_____

_____

_____

_____

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

42-i 1288

. . . . . Fold here . . . . .

. . . . . Do not tear - Fold here and staple . . . . .

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark