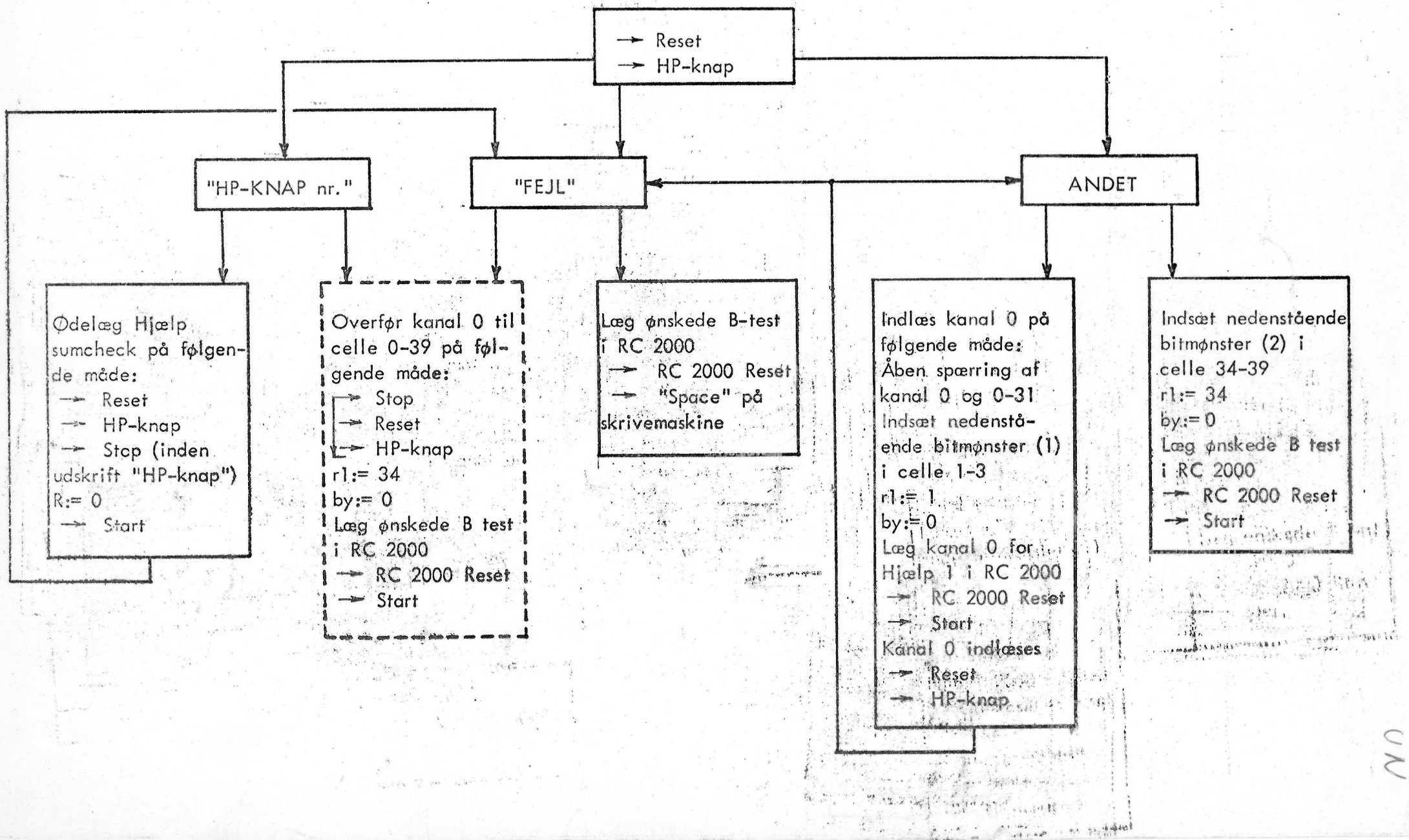
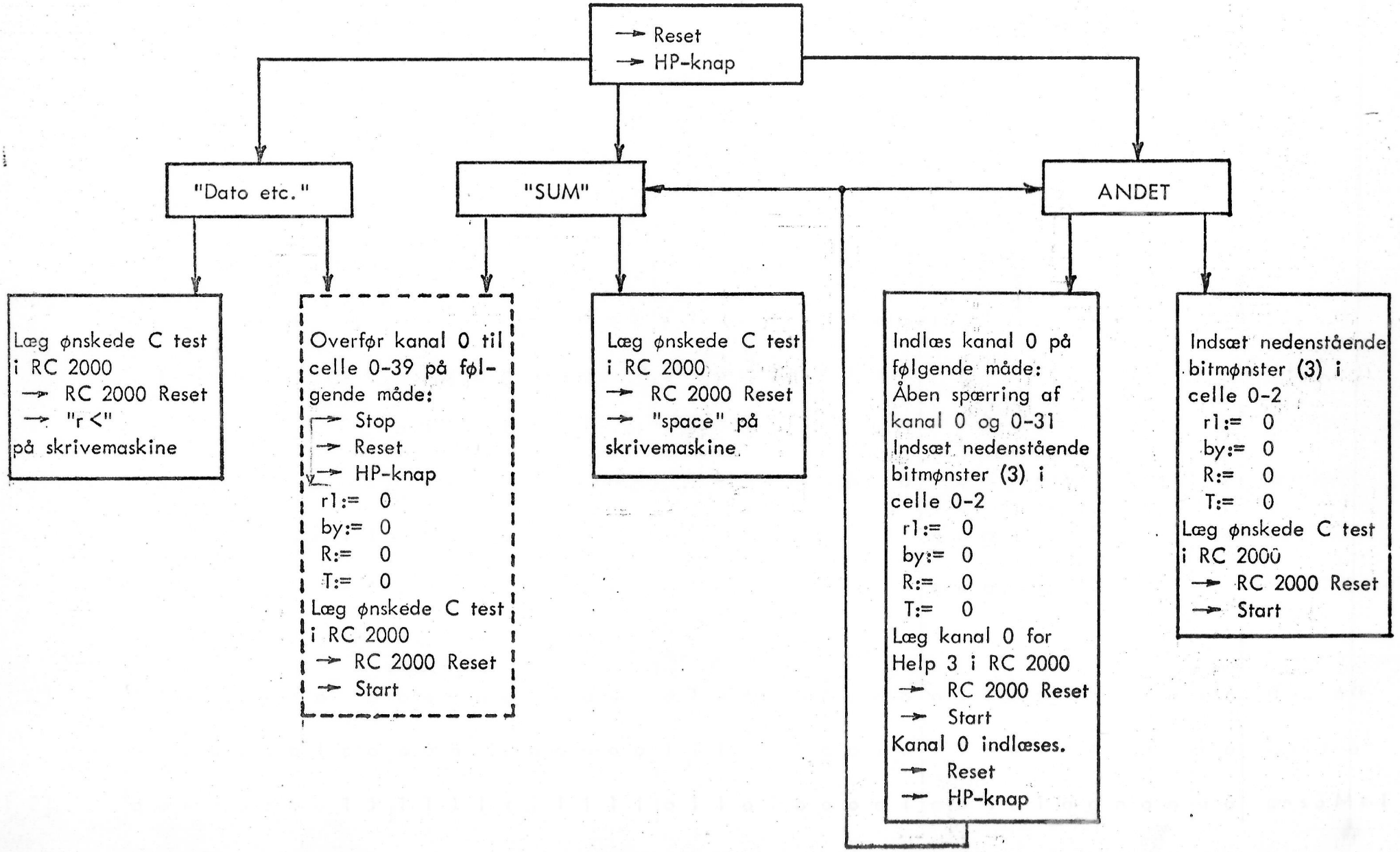
INDLÆSNING AF TESTPROGRAMMER
HJÆLP 1 (B Tests)



INDLÆSNING AF TESTPROGRAMMER
HELP 3 (C Tests)



INDLÆSNING AF TESTPROGRAMMER

	Pos.	o 9	10 19	20 29	30 39		
	[1]	o o o o o o o o o o	1 1 1 1 1 1 1 o 1 o	1 1 1 o 1 1 o o o o	o 1 o o 1 o o o o o	1 o	ly 0,tl -6
(1)	[2]	o o o o o o o o o o	o o o o o o o o o 1	o 1 1 o o 1 o o o o	1 1 1 o o o o o o o	1 o	ca 0,hv 1
	[3]	o o o o o o o 1 o 1	1 1 1 1 1 1 1 1 1 1	o 1 1 o 1 o o o o o	o o o o 1 o o o o o	o o	gm 5 M t-1
	[34]	o o o 1 o o o o o o	o o o o o o o o o o	o 1 1 o 1 1 1 o o o	1 o 1 o o o 1 o 1 o	o o	pmn 64 XDIZA
	[35]	1 1 1 1 1 1 1 o o 1	o o o o o o o o o 1	o 1 o o 1 o o o o o	1 1 1 o 1 1 o o 1 o	1 o	tl -7,ly r1
	[36]	o o o o o o o o o o	o 1 1 1 1 1 1 1 o o	1 o o o 1 1 o o o o	o o o 1 1 o 1 o 1 o	o o	pi 0 LZA t 508
(2)	[37]	o o o o o o o o o o	o o o o o o o o o o	o 1 1 1 o o o o o o	1 o o o o o 1 o o 1	o o	xr 0 XIZB
	[38]	o o o o 1 o o o 1 1	o o o o o o o o o o	1 1 1 o o o o o o o	o o o 1 1 o 1 o o 1	o o	hv 35 LZB
	[39]	o o o o 1 o 1 o o 1	1 1 1 1 1 1 1 1 1 1	o 1 o 1 o 1 o o o o	o o o o 1 1 1 1 1 1	o o	gr 41 MRC t-1
	[0]	1 1 1 1 1 1 1 o 1 o	o o o o o o o o o o	o 1 o o 1 o o o o o	o 1 1 o o 1 o o o o	1 o	tl -6,ca 0
(3)	[1]	o o o o o o o 1 o o	o o o o o o o o o o	1 1 1 o 1 1 o o 1 o	1 1 o o 1 o o o o o	1 o	ly r4,hs 0
	[2]	o o o o o o o o 1 1	1 1 1 1 1 1 1 1 1 1	o 1 1 o 1 o o o o 1	o o o o 1 o o o o o	o o	gm s 3 M t-1

UN

BEH/ilb
2.11.70

Test programmer

Røde strimler vil altid være C-tester, der kan indlæses på følgende måde:

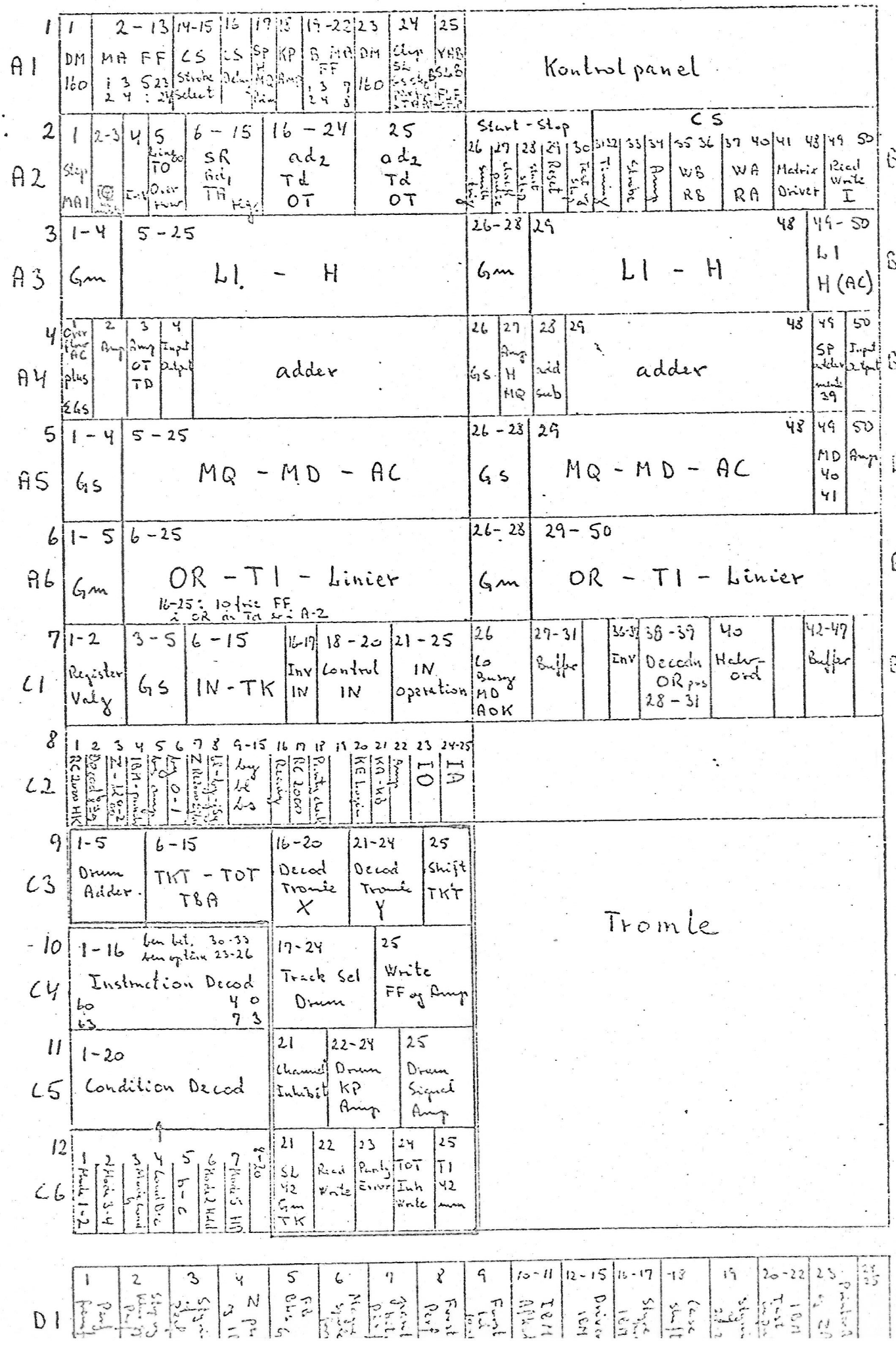
1. Skrivemaskinen skriver Hp knap (+ et tal):
   Man trykker på Hp knap og umiddelbart efter på normal stop, herefter nulstilles R-registeret og der trykkes normal start. Skrivemaskinen vil da udskrive "fejl".
2. Skrivemaskinen skriver "fejl":
   Det første stykke tape springes over, så man starter i gabet med feedhuller; man taster space på skrivemaskinen og testen indlæses.
   Skrivemaskinen er i UC; dette kan fjernes ved at trykke reset og normal start.
3. Skrivemaskinen skriver 07.25.10.70.828 testen kan indlæses med r<
4. Skrivemaskinen skriver SUM, testen indlæses med space.

Obs: Man kan ikke umiddelbart anvende skrivemaskinen eller hjælpesystemer, hvis dette ønskes, skal man nulstille celle 1023 (alle bits i r 1) (reset - 2 gange micro start - nulstil L - 1 gang micro start)

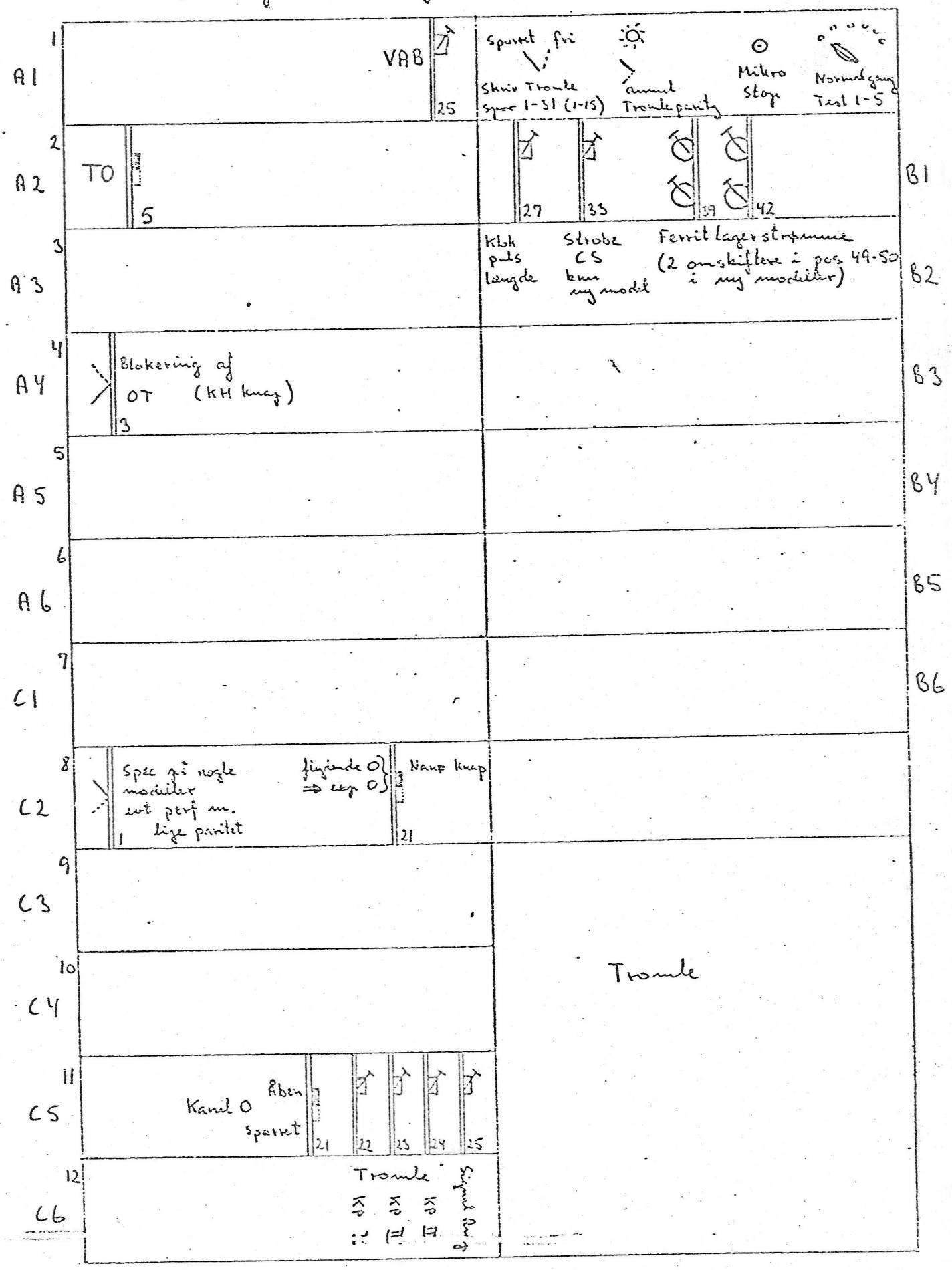
Grønne strimler vil altid være kompud strimler med et specielt help 3 hoved der medfører at man kan læse strimlerne ind i help 3 med r < og springe det første stykke over til man møder feedhuller, hvis man vil læse dem i hjælp 1 med 1.

Hvide strimler er ukurente og vil være beskrevet i hvert enkelt tilfælde.

Gier Kort Oversigt



Grov grille oversigt



Vippekontakters og skydekontakters stillinger vilkårlig fra model til model.

JEP

Unit: GIER G-1	Designed H.I.	GIER 6/ Standard	Drawing No
	Approved		Drawn by P.P. [illegible]
			Checked E.E. [illegible]

UN

by register værdier:

0	1	2	3	4	5	6	7	8	9
512	256	128	64	32	16	8	4	2	1

512 : ikke gyldig enhed, men hp - spænding

Input: Bit 7-8-9

0	0 0 0	RC 2000 med paritets stop
1	0 0 1	Skrivemaskine
2	0 1 0	Hulkortlæser, Bull Børstesorterer
3	0 1 1	RC 2000 uden paritets stop, men Ro og Roo gå "1" hvis fejl, kan i program selv sørge gå dette.
4	1 0 0	Hulkortlæser ?
7	1 1 1	Hulkortlæser ?

Output: Bit 3-4-5-6

8	0 0 0 1	Printer
16	0 0 1 0	Skrivemaskine
32	0 1 0 0	Perforator
64	1 0 0 0	Plotter

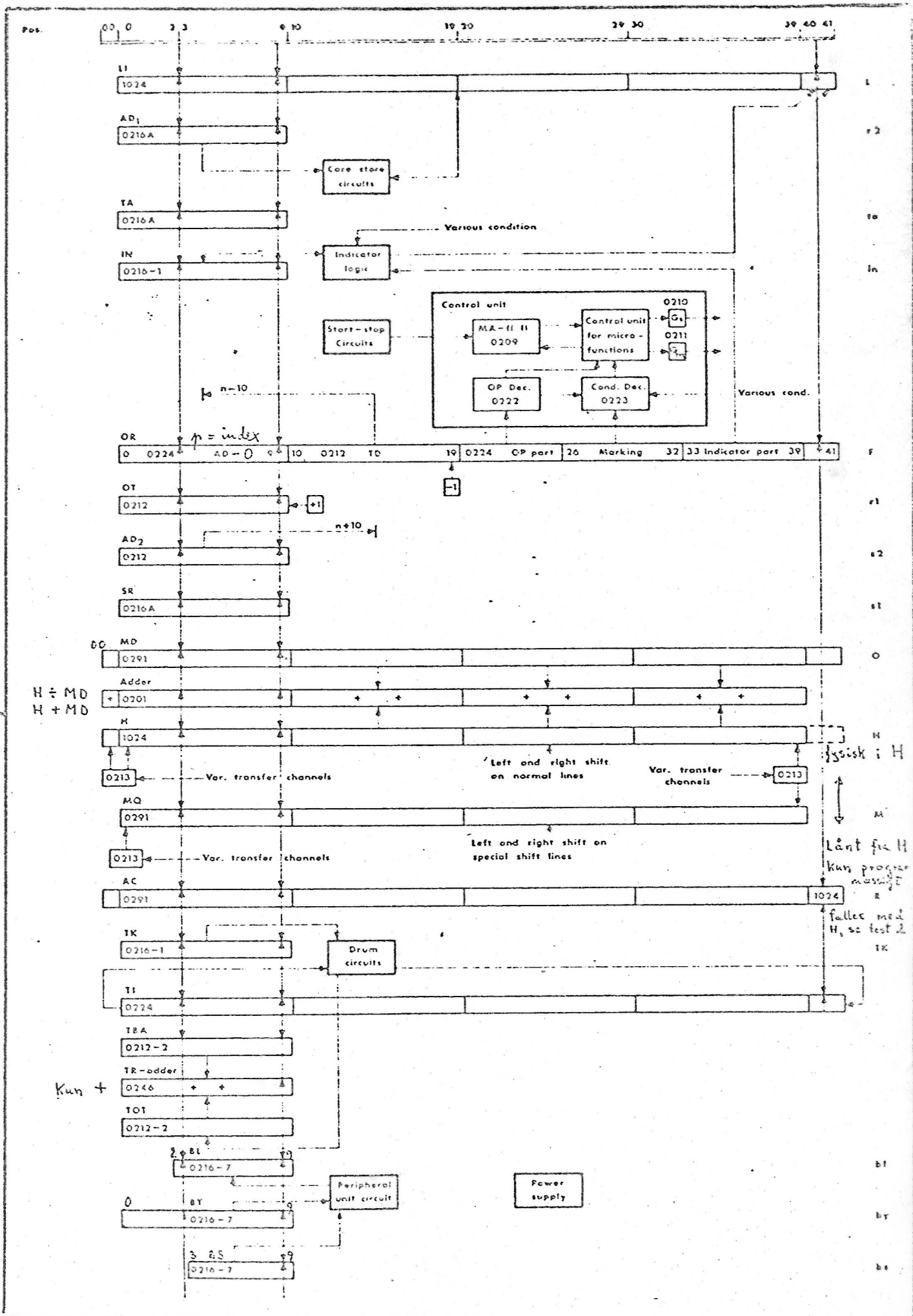
DM 160 MA lamper:

	MA - A											
Pos A1-1	12	11	10	9	8	7	6	5	4	3	2	1
Pos A1-1	24	23	22	21	20	19	18	17	16	15	14	13

					MA - B							
Pos A1-23	b FL.FL.	c FL.FL.	Hoo	Step	8	7	6	5	4	3	2	1

			TBA									
Pos A1-23	stub	TL	9	8	7	6	5	4	3	2	1	0

Unit: Gier	Designed	by register	Drawing No	
A/S REGNE	Approved	MA lamper	Drawn by	
	Checked		Checked	
			1 Sheets	Sheet 1



Unit GIER 6	Designed H.I.	Drawing No
		Drawn by

primitivt indlæseprogram PEH

[0]	lyn-1,tk-16	->	ly t1	->	t1-6,ca 0
[1]	gr 1 t1 M	->	gri27tlM	->	lyr4,hsr-1
[2]	hv 0	->	hv 3	->	gms3 t-1 M
[3]	tl 127				
[4]	ly 96				
[5]	cl 73				
[6]	ca 0				
[7]	hv 4				
[8]	cl 12				
[9]	gr 9	->	gri7 t-1 ->		hv 9t2 → hv 15 t 2
[10]	hv 3	->	hv 3		
[11]	gr 0 MA				
[12]	hv 3				
[13]	gr 1 MA				
[14]	hv 3				
[15]	gr 2 M				
[16]	hvh0				

Det første af øvrige ferritlager := zq 1

bitmønstre

	adresse	talle	gmnd	SOrS	gmnd	SOrS					
					xvDIo	IA	Tak	/	F		
[0]	xxxxxxxxxx	xxxxxx....	xxx.xx	x...	.xx.	..	..	..	x.		
[1]	x	x	.x.x.x	....	...	x.	..	..	..		

BY := 0

start i celle 0 ≈ r|: = 0

strimlen i læseren i det første gab,

Reset læser,

Start Gier.

pos.	0 1 2 3 4 5 6 7 8 9	10 11 12 13 14 15 16 17 18 19	20 21 22 23 24 25	26 27 28 29	30 31 32 33 34 35 36 37 38 39	40 41	Ordrer	Kan inkluderes binær strimmel, B-test indeholdt i HJÆLP kanal 0
1	X	Se bilag: startprocedure ved indlæsning med kanal 0	XX XX	X	X X X X		prin 64 XDIZA	
2	XXXXXXX . X	X	X X		XXX XX X	X	tl -7, ly r1	
5		XXXXXXXX	X XX		XX X X .		pi 0 LZA t508	
7			XXX		X X X		xr 0 X IZB	
8	X XX		XXX		XX X X		hr 35 LZB	
9	X X X	XXXXXXXXXXX	X X X		XXXXXX		gr 41 MRC t-1	

Prim. Input.
Tilsvarende finales i HJÆLP 3 samt 0 eller 23...

pos.	0 1 2 3 4 5 6 7 8 9	10 11 12 13 14 15 16 17 18 19	20 21 22 23 24 25	26 27 28 29	30 31 32 33 34 35 36 37 38 39	40 41	Ordrer	Primitiv læsning. Læser kanal 0 og andet med spec. start, ikke binær.
1		XXXXXXX X	XXX XX		X . X . .	X	ly 0, tl -6	
2		X	XX X		XXX .	X	ca 0, hv 1	
3	X X	XXXXXXXXXX	XX X		X		gm 5 M t-1	

HJÆLP

pos.	0 1 2 3 4 5 6 7 8 9	10 11 12 13 14 15 16 17 18 19	20 21 22 23 24 25	26 27 28 29	30 31 32 33 34 35 36 37 38 39	40 41	Ordrer	Primitiv læsning. Læser kanal 0, C-test og andet med spec. start, ikke binær
1	XXXXXXX X		X X		XX X	X	tl -6, ca 0	
2	X		XXX XX	X	XX X	X	ly r4, hs 0	
3	XX	XXXXXXXXXX	XX X	X	X		gm s3 M t-1	

HJÆLP 3

SE IØVRIGT: TEST PROGRAMS TYPE B/C (4 sider vejledning)

Testprograms Type B/C

The C tests on paper tape consists of a short header followed by the corresponding B program. Between header and program is a gap 1" long.

B test = The piece of tape after this gap including the proper test program in binary form starting in cell 41.

C test = The header + B test.

The old Help system called Hjælp uses the B test whereas the new system Help 3 reads the C test.

When the program types
t < number of program >
the header has been read and the remaining part of the tape (the B program) will be read immediately.

When the B program has been read, the program in question starts running.

The action of the header is:

a. A program (in bin 0 form) containing the primitive loader of Hjælp is read to the core store in cells 493-512 and entered.
b. This program types the test number and moves the loader to core 34-39.
c. The program jumps to core 34 and from this point the action is as if track 0 of Hjælp has been called.

Punching of C Programs

C programs are punched by the slip program A-224, "B to C Producer", and this program, and how to use it, is described in a separate paper.

Mentioned in method VI in the following there exists a primitive input program PEH, which only uses 2 cells to start.
For more information see page 8 , Application of PEH primitive input.

There exists a test named t0 which in fact is not a test but a B/C test head. It gives the ability to find a given C test very easily.
t0 requires the number of each following C test punched (by hand) with even parity just before the beginning of the total C test.
A paper tape of B/C test ought to start with t0 for more convenience.
Eventually place PEH prim. input program at the very first on the tape.
Advantage: Read paper tape with r< or ! if a Help System is intact and you will get t0, or use PEH prim input procedure.
With gaps of suitable length among the parts of tape one could always start in a gap where one wishes.

Application of test t0 (head)

Input procedure: Exactly as any other B/C test (-since the t0 test also has a header followed by a B program), see later.

After input: Track 0 is changed. Thus pressing HP, Gier types on typewriter in red the sign:

t: type the wanted C test number followed by CR.

OR: Type space which gives you the nearest following C test.

Want another C test?:

a. Place the tape in the reader at an arbitrary spot before the wanted test number, press reader reset.
b. Press Reset - HP.
c. Type the wanted number as before.

Returning to a Help System:

a. Press Reset - HP (Gier types t).

Help 3 I

b. Place Basic Help 3 in the reader starting at the second row of spaces if old→new are included. Otherwise at the beginning after an eventual track 0. Press Reader Reset.
c. Type h

OR: II

b. Place track 0 in the reader starting at the first row of spaces, press Reader Reset.
c. Type h.
Gier now writes SUM (in red) meaning track 0 is OK. Continue at d.

OR: III

b. Place C test t0 in the reader, press Reader Reset.

c. Type space.
Gier now writes SUM (in red) meaning track 0 is OK.

d. In case of II and III: place Basic Help 3 in the reader starting at the second row of spaces if old→new are included. Otherwise at the beginning after an eventual track 0.
Press Reader Reset.

e. Type space
Gier now reads Help 3 and writes e. g.
511 1023.810.898 e0
The date is to be restored.

Hjælp (old Help):

a. Press Reset - HP (Gier types t).

b. Place C test t0 in the reader, press Reader Reset.

c. Type space
Gier now writes SUM (in red).

d. Type space
Gier now writes FEJL (in red).
0 is OK.

e. Place Hjælp system in the reader after track 0 or new→old if included, press Reader Reset.

f. Type space
Gier now reads Hjælp and writes e. g. hp-knap 422.

How to change Help system:

It is possible to change from old Help system (Hjælp) to new Help 3 system and vice versa by reading one of the programs old→new or new→old, which are changing track 0.
If one not has the programs present (usually in the head of Hjælp and Help 3 system paper tapes) one can use t0 instead:

a. Read t0, see input procedure.

b. Use 'Returning to a Help system', see above.

Input Procedures of B/C test:

Help 3 intact — writes e.g. 47 14.7.70 e 15 after pressing Reset - HP.

I
a. Place C-test in the reader, press Reader Reset
b. Type r< on typewriter.

OR: II — Read track 0 to the core store by:
a. pressing Reset or Microtempi Stop.
b. Hold Normal Stop permanently down while pressing HP.
Place C-test in the reader, press Reader Reset and start Gier in cell 23 by:
a. Setting r1=23
b. BY = 0
c. Pressing Normal Start

OR: III — Read PEH prim. input (in the head of t0 and the other B/C tests) by:
a. Place the tape in the reader starting at the very beginning, press Reader Reset.
b. Type r <
c. Gier now writes t if t0 is included.

Help 3 track 0 intact: writes SUM after pressing Reset - HP:

IV
a. Place C-test in the reader, press Reader Reset.
b. Type space on typewriter.

Help 3 not intact:

V
a. Insert Help 3 track 0's 3 cells used for primitive input into cell 0 to 2, see page 9 or A Manual of Help 3 page 24.
b. Place C test in the reader instead of the track 0 paper tape, and start as for track 0.

OR: VI
a. Insert PEH primitive input program, see page 8. If C test (or t0) are not on the same paper tape:
b. Place C test in the reader, press Reader Reset.

Hjælp (old Help system) intact, writes e.g. hp-knap 349 after pressing Reset-HP.

VII Destroy Hjælp sumcheck by:

a. Press Reset.

b. Press HP, interrupt quick by pressing normal stop.

c. Select R-register and clear.

d. Press Normal start

Gier now writes FEJL (or e.g. hp-knap 422). Do not touch Reset - HP, use procedure 'Hjælp track 0 intact' following, begin at a.

OR: VIII Read track 0 to the core store by:

a. Press Reset.

b. Hold Normal Stop permanently down while pressing HP

(track 0 is now placed in the core store).

c. Select by, clear.

d. Select r1, set r1:=34.

e. Place B test in the reader, press Reader Reset.

f. Press Normal Start.

Gier now writes t if t0, or the actual B test starts running.

OR: IX Read PEH prim. input (in the head of t0 and the other B/C tests) by:

a. Place the tape in the reader starting at the very beginning, press Reader Reset.

b. Type I

c. Gier now writes t if t0 is included.

Hjælp track 0 intact, writes FEJL after pressing Reset - HP.

X

a. Place B-test in the reader, press Reader Reset.

b. Type space on typewriter.

Hjælp not intact:

(V-VI)

a. See Help 3 not intact for use of Help 3 system.

OR:

XI

a. Insert Hjælp track 0's 3 cells used for primitive input into cell 1 to 3, see page 9 for principle.

b. Place Hjælp track 0 paper tape in the reader, press Reader Reset.

c. Press Normal start, Reset - HP.
Gier now writes FEJL.

d. See procedure Hjælp track 0 intact.

OR:

XII

Use primitive input program in cell 34-39 (part of Hjælp track 0) as follows: (or see the next possibility).

a. Insert the 6 cells 34-39, use the same principle as for track 0, see page 9.

BITS:

(34) 3,21-22, 24-26, 30, 32, 36, 38

(35) 0-6, 9, 19, 21, 24, 30-32, 34-35, 38, 40

(36) 11-17, 20, 24-25, 33-34, 36, 38

(37) 21-23, 30, 36, 39

(38) 4, 8-9, 20-22, 33-34, 36, 39

(39) 4, 6, 9-19, 21, 23, 25, 34-39

b. Set by: =0 and r1 : = 34

c. Place B-test in the reader, press Reader Reset.

d. Press Normal Start.

IF all the possibilities overleaf are not convenient:

XIII

Use B/C test 21 which is testing Mode 1 and 4 in 20 seconds. Just after input of the the program the primitive input program cell 34-39 is established and will remain during a normal program run.

Want prim. input program?:

j. Press Reset for stopping t21.

k. Select by, clear.

l. Select r1, set r1:=34.

You are now ready to use Hjælp prim. input program by:

m. Place B test in the reader, press Reader Reset.

n. Press Normal Start.

Emergency Input of t21:

a. Reset Gier.

b. Place B-test 21 in the reader starting a few inches up the tape in the middle of the 100 spaces, press Reader Reset.

Clear the entire core store using hardware test 1:

c. Select L register and clear.

d. Switch to test 1 in the Gier Cabinet.

e. Press Normal Start, wait a few seconds, press Normal Stop.

f. Switch to normal in the cabinet again.

g. Set R: = 0 , M: = 0 and by: = 0

h. Insert in cell 41-43 the following instructions: (see page 9 for principle).

(41)	ly 43	bit pos 4, 6, 8-9, 20-22, 24-25.
(42)	cl 76	3, 6-7, 21, 23
(43)	gm	21-22, 24

i. Set r1: = 41 and press Normal start, t21 is now running.

It is now possible to use cell 34-39 for input of the other B/C tests:

Continue with j. above.

Application of PEH primitive input.

Structure: Head followed by a gap again followed by the program.

Result: The 3 cells normally used for input of Help 3 track 0 are placed correctly into cell 0-2. These 3 cells are also used for input of C tests and tapes in bin-0 form .

Advantage: Easy to insert, only 2 simple cells. If known, easy to use for troubleshooting if Gier does not accept any input. (A complete description in Danish exists).

Input Procedure: a. Insert the 2 cells 0 and 1, use the same principle as for track 0, see page 9.

(0) lyn - 1, tk - 16

(1) gr 1 t1 M

Bit pattern:

(0) 0-15, 20-22, 24-26, 31-32, 40

(1) 9, 19, 21, 23, 25, 34

b. Set by: = 0

c. Set r1: = 0

d. Place the tape in the reader starting in the first gap, press reader Reset.

e. Press Normal Start.

OR If one of the Help systems are intact:

a. Press Reset - HP

b. Place the tape in the reader starting at the very beginning, press Reader Reset.

c. Type r< or r, slip< (Help 3)

! (Hjælp).

Input of track 0:

Reset ~ mikrostop (short: stop)
Step ~ mikrostart (short: start)

Help 3:

Reset Gier (Press reset or mikrostop)
Select r1, clear, stop, select L, 2 start (mikro)
clear, insert cell 0: (s-1)
(0) 0-6, 8, 21, 24, 31-32, 35, 40, start, stop, clear L, 2 start
select r1, insert 1 (bit 9), stop, select L, 2 start,
clear, insert cell 1: (s)
(1) 7, 20-22, 24-25, 28, 30-31, 34, 40, start, stop, clear L, 2 start
select r1, insert 2 (bit 8), stop, select L, 2 start
clear, insert cell 2: (s+1)
(2) 8-19, 21-22, 24, 29, 34, start, stop, clear L, 2 start, stop!
select by, clear,
select R, clear, pos. 00 (T) upper left too
select r1, clear.
Place track 0 tape in the reader, press Reader-Reset
Start Gier by pressing Normal Start, Gier now writes SUM.

Hjælp (Old Help): Are read in the same way (add 1 to all r1 values, clear:=1) but into cell 1-3, and the bits are different:
(1) 10-16, 18, 20-22, 24-25, 31, 34, 40
(2) 19, 21-22, 25, 30-32, 40
(3) 7, 9-19, 21-22, 24, 34
After normal start: Press Reset-HP, Gier now writes FEJL.

Generally: Every time L is selected, 2 starts after a stop will show the contents of the cell chosen in r1. Pressing the second start will cause L to change as a check (if the previous content was different) or you have lost one of the steps.

x) ved udskrift Fejl: rodt start i celle 34 i kanal 0.

Testprograms Type C
RIALTO, 19.8.1968
JFM

The usual testprograms of B type can not be read by the primitive input program on track 0 in HELP 3. This paper describes how the modified tapes, called C programs, may be read by HELP 3.

FORMAT OF C TAPES

The paper tapes containing C programs consists of a short header followed by the corresponding B program. The header can be read by HELP 3 and enables the user to read the B program.

INPUT PROCEDURE

The header is followed by a tape gap 1" long. The reading may start before the header or in this gap. If the should be read by HJÆLP one starts in the gap, and if HELP 3 is used for reading one starts before the header. In the last case the input procedure is as follows:

(1) Read track 0 to the core store by
   a. Pressing RESET or MICROTEMPI STOP.
   b. Pressing HP-BUTTON while NORMAL STOP is pressed.

(2) Place the tape in the reader and start GIER in cell 23 by
   a. Setting r1=23 x)
   b. BY=0
   c. Pressing NORMAL START

(3) When the program types
   t < number of program >
   the header hes been read and the remaining part of the tape (the B program) will be read immidiately.

(4) When the B program has been read, the program in question starts running.

The actual procedure between (2) and (3) above is:

(A) A program (in bin 0 form) containing the primitive loader of HJÆLP is read to the core store in cells 493-512 and entered.

(B) This program types the test number and moves the loader to core 34-39.

(C) The program jumps to core 34 and from this point the action is as if track 0 of HJÆLP has been called.

PUNCHING OF C PROGRAMS

C programs are punched by the slip program A-224, "B to C Producer", and this program and how to use it is described in a seperate paper.

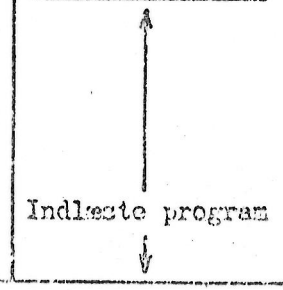
REGNECENTRALEN
Dansk Institut for Matematikmaskiner

Tegnet	Kontrol	Godk.
10.8.63		

Startprocedure ved indlæsning med kanal 0

Tid / celle	Før start	1. ord indlæst og lagret	2. ord indlæst og lagret	3. ord indlæst og lagret	4. ord indlæst og lagret
34	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA
35	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1
36	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508
37	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB
38	hv 35 LZB	hv 35 LZB	hv 35 LZB	hv 35 LZB	hv 35 LZB
39	gr 41 MRC t-1	gr 40 MRC t-1	gr 42 MRC t-1	gr 41 MRC t-1	gr 40 MRC t-1
40		hv 34	hv 34	hv 34	gr 46 MRC t-2
41				hv 34	hv 34
42					
43					
44					
45					
46					

Tid / celle	4. ord indlæst og lagret	5. ord indlæst og lagret	5. ord indlæst og lagret	6. ord indlæst og lagret	Sidste ord indlæst og lagret
34	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	pm n 64 DXIZA	
35	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	tl -7 , ly r+1	
36	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	pi 0 LZA t508	
37	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	xr 0 XIZB	
38	hv 35 LZB	hv 35 LZB	hv 35 LZB	hv 35 LZB	
39	gr 40 MRC t-1	tl 3	tl 3	tl 3	
40	gr 44 MRC t-2	gr 44 MRC t-2	gr 42 MRC t-2	gr 1 MRC t-1	gr 41 MRC t-1
41	hv 34	hv 34	hv 34	hv 34	
42			tl 3	tl 3	
43					
44	gr 46 MRC t-2	gr 46 MRC t-2	gr 46 MRC t-2	gr 46 MRC t-2	
45					
46					



DECIMAL	OPE.	BINARY VALUE 20	21	22	23	24	25
0	qq	.	.	.	.	.	.
1	zq	.	.	.	.	.	1
2	ar	.	.	.	.	1	.
3	sr	.	.	.	.	1	1
4	an	.	.	.	1	.	.
5	sn	.	.	.	1	.	1
6	ac	.	.	.	1	1	.
7	sc	.	.	.	1	1	1
8	mb	.	.	1	.	.	.
9	ab	.	.	1	.	.	1
10	mt	.	.	1	.	1	.
11	mk	.	.	1	.	1	1
12	ml	.	.	1	1	.	.
13	dk	.	.	1	1	.	1
14	dl	.	.	1	1	1	.
15	nk	.	.	1	1	1	1
16	nl	.	1	.	.	.	.
17	hr	.	1	.	.	.	1
18	tl	.	1	.	.	1	.
19	ck	.	1	.	.	1	1
20	cl	.	1	.	1	.	.
21	gr	.	1	.	1	.	1
22	ga	.	1	.	1	1	.
23	gt	.	1	.	1	1	1
24	tk	.	1	1	.	.	.
25	ca	.	1	1	.	.	1
26	gm	.	1	1	.	1	.
27	pm	.	1	1	.	1	1
28	xr	.	1	1	1	.	.
29	gi	.	1	1	1	.	1
30	ps	.	1	1	1	1	.
31	pp	.	1	1	1	1	1
32	pa	1	.	.	.	.	.
33	pt	1	.	.	.	.	1
34	hk	1	.	.	.	1	.
35	pi	1	.	.	.	1	1
36	is	1	.	.	1	.	.
37	it	1	.	.	1	.	1
38	cm	1	.	.	1	1	.
39	bt	1	.	.	1	1	1
40	ns	1	.	1	.	.	.
41	nt	1	.	1	.	.	1
42	gp	1	.	1	.	1	.
43	nc	1	.	1	.	1	1
44	il	1	.	1	1	.	.
45	us	1	.	1	1	.	1
46	gg	1	.	1	1	1	.
47	gc	1	.	1	1	1	1
48	pc	1	1	.	.	.	.
49	bs	1	1	.	.	.	1
50	hs	1	1	.	.	1	.
51	vy	1	1	.	.	1	1
52	lk	1	1	.	1	.	.
53	sk	1	1	.	1	.	1
54	gk	1	1	.	1	1	.
55	vk	1	1	.	1	1	1
56	hv	1	1	1	.	.	.
57	zj	1	1	1	.	.	1
58	sy	1	1	1	.	1	.
59	ly	1	1	1	.	1	1
60	hh	1	1	1	1	.	.
61	gs	1	1	1	1	.	1
62	zl	1	1	1	1	1	.
63	ud	1	1	1	1	1	1

OPE.	DECIMAL	BINARY VALUE 20	21	22	23	24	25
ab	9	.	.	1	.	.	1
ac	6	.	.	.	1	1	.
an	4	.	.	.	1	.	.
ar	2	.	.	.	.	1	.
bs	49	1	1	.	.	.	1
bt	39	1	.	.	1	1	1
ca	25	.	1	1	.	.	1
ck	19	.	1	.	.	1	1
cl	20	.	1	.	1	.	.
cm	38	1	.	.	1	1	.
dk	13	.	.	1	1	.	1
dl	14	.	.	1	1	1	.
ga	22	.	1	.	1	1	.
gc	47	1	.	1	1	1	1
gg	46	1	.	1	1	1	.
gi	29	.	1	1	1	.	1
gk	54	1	1	.	1	1	.
gm	26	.	1	1	.	1	.
gp	42	1	.	1	.	1	.
gr	21	.	1	.	1	.	1
gs	61	1	1	1	1	.	1
gt	23	.	1	.	1	1	1
hh	60	1	1	1	1	.	.
hk	34	1	.	.	.	1	.
hr	17	.	1	.	.	.	1
hs	50	1	1	.	.	1	.
hv	56	1	1	1	.	.	.
il	44	1	.	1	1	.	.
is	36	1	.	.	1	.	.
it	37	1	.	.	1	.	1
lk	52	1	1	.	1	.	.
ly	59	1	1	1	.	1	1
mb	8	.	.	1	.	.	.
mk	11	.	.	1	.	1	1
ml	12	.	.	1	1	.	.
mt	10	.	.	1	.	1	.
nc	43	1	.	1	.	1	1
nk	15	.	.	1	1	1	1
nl	16	.	1	.	.	.	.
ns	40	1	.	1	.	.	.
nt	41	1	.	1	.	.	1
pa	32	1	.	.	.	.	.
pc	48	1	1	.	.	.	.
pi	35	1	.	.	.	1	1
pm	27	.	1	1	.	1	1
pp	31	.	1	1	1	1	1
ps	30	.	1	1	1	1	.
pt	33	1	.	.	.	.	1
qq	0	.	.	.	.	.	.
sc	7	.	.	.	1	1	1
sk	53	1	1	.	1	.	1
sn	5	.	.	.	1	.	1
sr	3	.	.	.	.	1	1
sy	58	1	1	1	.	1	.
tk	24	.	1	1	.	.	.
tl	18	.	1	.	.	1	.
ud	63	1	1	1	1	1	1
us	45	1	.	1	1	.	1
vk	55	1	1	.	1	1	1
vy	51	1	1	.	.	1	1
xr	28	.	1	1	1	.	.
zj	57	1	1	1	.	.	1
zl	62	1	1	1	1	1	.
zq	1	.	.	.	.	.	1

apr 67
jfm

TABLE OF INDICATORPARTS
(inactive are marked with an ×)

33	34	35	36	37	38	39		33	34	35	36	37	38	39	
.	.	.	.	.	.	.	I ×	1	.	.	.	.	.	.	N ×
.	.	.	.	.	.	1	IB ×	1	.	.	.	.	.	1	NB
.	.	.	.	.	1	.	IA ×	1	.	.	.	.	1	.	NA
.	.	.	.	.	1	1	IC ×	1	.	.	.	.	1	1	NC
.	.	.	.	1	.	.	IK	1	.	.	.	1	.	.	NK ×
.	.	.	.	1	.	1	IKB	1	.	.	.	1	.	1	NKB
.	.	.	.	1	1	.	IKA	1	.	.	.	1	1	.	NKA
.	.	.	.	1	1	1	IKC	1	.	.	.	1	1	1	NKC
.	.	.	1	.	.	.	IZ ×	1	.	.	1	.	.	.	NZ
.	.	.	1	.	.	1	IZB	1	.	.	1	.	.	1	NZB
.	.	.	1	.	1	.	IZA	1	.	.	1	.	1	.	NZA
.	.	.	1	.	1	1	IZC	1	.	.	1	.	1	1	NZC
.	.	.	1	1	.	.	IO ×	1	.	.	1	1	.	.	NO
.	.	.	1	1	.	1	IOB	1	.	.	1	1	.	1	NOB
.	.	.	1	1	1	.	IOA	1	.	.	1	1	1	.	NOA
.	.	.	1	1	1	1	IOC	1	.	.	1	1	1	1	NOC
.	.	1	.	.	.	.	IT ×	1	.	1	.	.	.	.	NT
.	.	1	.	.	.	1	ITB	1	.	1	.	.	.	1	NTB
.	.	1	.	.	1	.	ITA	1	.	1	.	.	1	.	NTA
.	.	1	.	.	1	1	ITC	1	.	1	.	.	1	1	NTC
.	.	1	.	1	.	.	IP ×	1	.	1	.	1	.	.	NP ×
.	.	1	.	1	.	1	IPB	1	.	1	.	1	.	1	NPB
.	.	1	.	1	1	.	IPA	1	.	1	.	1	1	.	NPA
.	.	1	.	1	1	1	IPC	1	.	1	.	1	1	1	NPC
.	.	1	1	.	.	.	IQ ×	1	.	1	1	.	.	.	NQ ×
.	.	1	1	.	.	1	IQB	1	.	1	1	.	.	1	NQB
.	.	1	1	.	1	.	IQA	1	.	1	1	.	1	.	NQA
.	.	1	1	.	1	1	IQC	1	.	1	1	.	1	1	NQC
.	.	1	1	1	.	.	IR ×	1	.	1	1	1	.	.	NR ×
.	.	1	1	1	.	1	IRB	1	.	1	1	1	.	1	NRB
.	.	1	1	1	1	.	IRA	1	.	1	1	1	1	.	NRA
.	.	1	1	1	1	1	IRC	1	.	1	1	1	1	1	NRC
.	1	.	.	.	.	.	M	1	1	.	.	.	.	.	L ×
.	1	.	.	.	.	1	MB	1	1	.	.	.	.	1	LB
.	1	.	.	.	1	.	MA	1	1	.	.	.	1	.	LA
.	1	.	.	.	1	1	MC	1	1	.	.	.	1	1	LC
.	1	.	.	1	.	.	MK ×	1	1	.	.	1	.	.	LK ×
.	1	.	.	1	.	1	MKB ×	1	1	.	.	1	.	1	LKB
.	1	.	.	1	1	.	MKA ×	1	1	.	.	1	1	.	LKA
.	1	.	.	1	1	1	MKC ×	1	1	.	.	1	1	1	LKC
.	1	.	1	.	.	.	MZ ×	1	1	.	1	.	.	.	LZ
.	1	.	1	.	.	1	MZB ×	1	1	.	1	.	.	1	LZB
.	1	.	1	.	1	.	MZA ×	1	1	.	1	.	1	.	LZA
.	1	.	1	.	1	1	MZC ×	1	1	.	1	.	1	1	LZC
.	1	.	1	1	.	.	MO ×	1	1	.	1	1	.	.	LO
.	1	.	1	1	.	1	MOB	1	1	.	1	1	.	1	LOB
.	1	.	1	1	1	.	MOA	1	1	.	1	1	1	.	LOA
.	1	.	1	1	1	1	MOC	1	1	.	1	1	1	1	LOC
.	1	1	.	.	.	.	MT ×	1	1	1	.	.	.	.	LT
.	1	1	.	.	.	1	MTB	1	1	1	.	.	.	1	LTB
.	1	1	.	.	1	.	MTA	1	1	1	.	.	1	.	LTA
.	1	1	.	.	1	1	MTC	1	1	1	.	.	1	1	LTC
.	1	1	.	1	.	.	MP ×	1	1	1	.	1	.	.	LP ×
.	1	1	.	1	.	1	MPB	1	1	1	.	1	.	1	LPB
.	1	1	.	1	1	.	MPA	1	1	1	.	1	1	.	LPA
.	1	1	.	1	1	1	MPC	1	1	1	.	1	1	1	LPC
.	1	1	1	.	.	.	MQ ×	1	1	1	1	.	.	.	LQ ×
.	1	1	1	.	.	1	MQB	1	1	1	1	.	.	1	LQB
.	1	1	1	.	1	.	MQA	1	1	1	1	.	1	.	LQA
.	1	1	1	.	1	1	MQC	1	1	1	1	.	1	1	LQC
.	1	1	1	1	.	.	MR ×	1	1	1	1	1	.	.	LR ×
.	1	1	1	1	.	1	MRB	1	1	1	1	1	.	1	LRB
.	1	1	1	1	1	.	MRA	1	1	1	1	1	1	.	LRA
.	1	1	1	1	1	1	MRC	1	1	1	1	1	1	1	LRC

APR. 67
3 PM

[Testprogram no. 1.Topsøes test of the drum.
The program writes pseudo- random numbers on the tracks 1-319(the same number on all tracks) and compares the number that has been read with the written number.After each comparison for all the tracks is typed ok on the typewriter. By errors is typed track number,word number, the written word and the word that has been read.Bit 40 and 41 do not contain random numbers. KA=1 effects simultaneous computation by writing, KB=1 effects simultaneous computation by reading.Output on the typewriter can be removed by setting by=0 after the testprogram is read in.First instruction to be executed is zqLKB.
The drum parity check should be removed during run by the switch in GIER.]

```
i=41
[  41] zqLKB                    ;stop if KB=1
[  42] vy  16                   ;choice of typewriter
[  43] pa r8 t0                 ;choice of start track
[  44] pa r3 t-305              ;choice of start address for random  numbers
[  45] hs r31                   ;jump to sequ. for random  numbers
[  46] tk 1 ,it 1               ;shift
[  47] gr -305,hs r29           ;store random  numbers from 720 and jump
[  48] tk -30,ac (r-1)          ;shift and add
[  49] it(r-2),bs -265          ;is repeated 40-
[  50] hh r-5                   ;times
[  51] vk 0 t1                  ;choice of track
[  52] sk -304                  ;write track
[  53] hv r5 NKA                ;jump if KA=0
[  54] pa r2 t50                ;put in 50 as counting number
[  55] hs r21 t29               ;compute 50-
[  56] bt 50 t-1                ;random  numbers simultaneously-
[  57] hv r-2                   ;with writing
[  58] vk(r-7),lk -504          ;choose track and read until 520
[  59] hv r5 NKB                ;jump if KB=0
[  60] pa r2 t50                ;place 50
[  61] hs r15                   ;compute 50-
[  62] bt 50 t-1                ;random  numbers simultaneously-
[  63] hv r-2                   ;with reading
[  64] pa r3 t-305              ;choice of start address for random  numbers
[  65] pa r3 t-505              ;choice of start address for read numbers
[  66] vk (r-15)                ;wait for the drum
[  67] arn -305 t1              ;take the written
[  68] sr -505 t1               ;subtract the read
[  69] gr r15,arn r15           ;store the difference and fetch it again
[  70] hv r15 NZ                ;jump to typeout if not zero
[  71] it (r-3),bs -465         ;this is repeated-
[  72] hv r-5                   ;40 times
[  73] it (r-22),bs 319         ;if track number < 319
[  74] hv r-23                  ;jump back
[  75] hv r42                   ;else jump to typeout of ok
[  76] pm r7,mln r5             ;RAND-1,subroutine for random  numbers
[  77] tk 8,gm r6
[  78] ar r5
[  79] sr r5 LO
[  80] gr r3,hr s1              ;jump back
[  81] gtn p,dln r434           ;constant
[  82] udn (p511),gc -1         ;constant
[  83] sn p206 X t-228          ;constant
[  84] qq
[  85] sy 64,ud r39             ;type CR and count down in 120
```

```
[  86] arn(r-35) D          ;fetch track number
[  87] hs r17               ;jump to decimal typeout
[  88] qq
[  89] arn(r-21) D          ;fetch word number relatively


[  90] sr -504 D            ;find word number
[  91] hs r13               ;jump to decimal typeout
[  92] qq
[  93] sy 64                ;type CR
[  94] arn(r-27),tk 1       ;fetch the written word and-
[  95] sy 1 V LT            ;write-
[  96] sy                   ;it-
[  97] hh r-3 NZ            ;until the rest is only o-es
[  98] sy 64                ;write CR
[  99] arn(r-31),tk 1       ;fetch the read word and-
[ 100] sy 1V LT             ;write it-
[ 101] sy                   ;until the rest is only o-es
[ 102] hh r-3 NZ            ;
[ 103] hvr-32               ;jump to repeat
[ 104] ck 10,tk 30          ;decimal typeout
[ 105] ck -10,pt r9
[ 106] dk r9 X
[ 107] pp 4,mln r9
[ 108] pp p-1,tk 20
[ 109] ar r5 V LZ
[ 110] pt r4 t16
[ 111] gt r , sy
[ 112] bs p510,ud r-2
[ 113] bs p , hh r-6
[ 114] hr s1,qq 16          ;jump back
[ 115] qq9  t-241
[ 116] qq  IZA
[ 117] sy 38 ,sy 34         ;type ok
[ 118] sy ,sy               ;type two spaces
[ 119] sy ,sy               ;type two spaces
[ 120] bt 12 t-1            ;up until 12 times
[ 121] hv 43                ;jump to repeat
[ 122] pa r-2 t12           ;else place 12 as counting number
[ 123] sy 64 ,hv 43         ;write CR and jump to repeat
[ 124] pa r-4 t-1           ;count down to second to the last
e41
```

[Testprogram no. 6.Test of the core-store including marks.
The program examines the whole core-store exept for the locations that are occupied by the program itself.
At first zeroes are send to a location and the answer is checked, then ones are send to the same location and the answer is checked.

If stop in location 44 (r1=45): R0-39 ≠ 0, existing ones are wrong.

If stop in location 46 (r2=47): R40-41 ≠ 0, existing ones are wrong.

If stop in location 50 (r1=51): R0-39 ≠ 0, to position(s) containing a one and having a zero on the right side a zero has been send from the store instead of a one.

If stop in location 52 (r1=53): R=40.41 ≠ 1.1 , existing zeroes are wrong.

With LKB=1 the program will stop after having examined the core store from location 58 to 1023 and further on from location 0 to 40.In the indicator, the address of the location that is tested, can be observed.]

```
i=41

[41] zq LKB
[42] grn 57  M +1              ;zero to variable location
[43] arn(42) ,pi (42)
[44] zq NZ                     ;stop if R0-39 ≠ 0
[45] qq  V NC
[46] zq                        ;stop if R40.41 ≠ 0.0
[47] srn 57                    ;constant is subtracted
[48] gr (42) MC                ;ones to variable location
[49] arn 57 , ar (42)          ;constant to R,content of location is added
[50] zq NZ                     ;stop if R0-39 ≠ 0
[51] qq  V LC
[52] zq                        ;stop if R40.41 ≠ 1.1
[53] arn 42 ,nc 40
[54] hv 42                     ;examine the next (variable) location
[55] pa 42 +57                 ;replace the address 57 in location 42
[56] hv 41                     ;restart with location no. 58
[57] qq IB                     ;constant( 1 bit in position 39)

e41
```

[Testprogram no. 7. Topsøes test of the floating point operations: AR F,SR F, MK F and DK F.
The testprogram operates in 6000 cycles.In each cycle two random,floating point numbers,a and b, are generated by means of the subroutines RAND-1 and RAND-2.The four floating point operations are then carried out on the numbers a and b, and the four results are added(as fixed points numbers) to four sum cells.
After 6000 cycles a line is printed on the typewriter,containing the number 6000 and the four sum cells in octal notation.The four sum cells are then cleared and 6000 new cycles are calculated,etc.If KB=1 the program will stop after 6000 cycles, and can be repeated by pressing the NORMAL START-button.

The correct output is (only the numbers):

6000 352714077753020 36242706607000 66043725657530 15577777733744
6000 352714077753020 36242706607000 66043725657530 15577777733744
etc.
test of: add. sub. mult. div.

Calculation of 6000 cycles takes about 35 sec.The value of 6000 is stored in cell 114 as an integer in pos. 39.It may be changed if required.
If KA is set to 1, output is obtained after each cycle.

The program is stored from cell 41 to 136.he first instruction to be executed is zq LKB.]

```
i=41

[ 41]  zq LKB                  ;stop if KB=1
[ 42]  vy 16,hv r4             ;typewriter output,go to 46
[ 43]  qq 0
[ 44]  qq 0
[ 45]  zq LKB                  ;stop if KB=1
[ 46]  grn r71 [117]           ;sum:=0
[ 47]  grn r71 [118]           ;sum:=0
[ 48]  grn r71 [119]           ;sum:=0
[ 49]  grn r71 [120]           ;sum:=0
[ 50]  hv r19                  ;go to 69
[ 51]  qq 0
[ 52]  qq 0
[ 53]  qq IZA
[ 54]  gcn s9  t320            ;parameter
[ 55]  pi      t-65            ;parameter
[ 56]  arn r-2,gr r-13         ;from 55 to 66 is placed -
[ 57]  dln r-14,tk 20          ;a subroutine for printing -
[ 58]  pi  64  NZ t-65         ;out the counter in location -
[ 59]  hv  r3   NZ             ;115 in decimal notation.
[ 60]  hv  r2   NTB
[ 61]  sy  16,hv  r2
[ 62]  gt  r ,sy
[ 63]  gm  r-11,pm  r-20
[ 64]  dln  r-11,gr  r-21
[ 65]  hr s2   LZ
[ 66]  pm  r-14,hv  r-9
[ 67]  qq        t256          ;parameter
[ 68]  qq        IB            ;parameter
[ 69]  grn  r-25               ;zeroes to cell 44
[ 70]  arn  r39,gr  r65        ;mge start-random from 109 to 135
[ 71]  grn  r44                ;counter:=0
[ 72]  pp  2,pp  p-1           ;p:=2 , p:=p-1
[ 73]  grn  p121,hs  r51       ;cell:=0 , go to RAND-2
[ 74]  qq  20                  ;parameter
```

```
[ 75]  sr  10 D                ;-10
[ 76]  ga  p121,hs  r52        ;set exp. ,go to RAND-1
[ 77]  tk  2    ITA            ;set sign
[ 78]  ck  -6,tl  -6           ;Roo:=0
[ 79]  ac  p121,arn  r-12      ;store , fetch 1
[ 80]  tk  1   LTA             ;if neg. then -2
[ 81]  ac  p121                ;add to cell
[ 82]  bs  p,hh  r-10          ;repeat
[ 83]  pp  4,pp  p-1           ;p:=4 , p:=p-1
[ 84]  arfn  r37,udf  p110     ;fetch b , execute
[ 85]  grf  r38                ;store in 123
[ 86]  arn  r37,ac  p117       ;add to cell
[ 87]  bs  p,hh  r-4           ;repeat
[ 88]  hsn  r5 X LKA t52       ;check output
[ 89]  arn  r-21,ar  r26       ;count
[ 90]  gr  r25,sr  r24         ;- limit
[ 91]  hv  r-19   LT           ;repeat
[ 92]  psn  r43                ;final output
[ 93]  sy  64,arn  r22         ;CAR RET,counter to R-reg.
[ 94]  hs  r-39 X  t59         ;print counter
[ 95]  qq  257  t-144          ;parameters
[ 96]  sy  ,sy                 ;2 SPACES
[ 97]  pp  4,pp  p-1           ;p:=4 , p:=p-1
[ 98]  arn  p117,tl  -10       ;fetch number , shift
[ 99]  ga  r17,sr  r17         ;delete bits 0-9
[100]  tl  3,ca                ;shift , if zero
[101]  sy  16,hv  r2           ;then print zero
[102]  ga  r14,sy (r14)        ;store and print
[103]  bt  13  t-1             ;repeat 13 times
[104]  hv  r-5
[105]  pan  r-2  X  t13        ;reset address in bt-instruction
[106]  sy  ,sy                 ;2 SPACES
[107]  bs  p,hh  r-10          ;repeat
[108]  hr  s1                  ;exit ( to 136)
[109]  bs  p104  V  t-333      ;start random
[110]  dkf  r11                ;/b
[111]  mkf  r11                ;Xa
[112]  srf  r10                ;-a
[113]  arf  r9                 ;+a
[114]  qq  (s)   XV  LT        ;6000 = limit
[115]  qq                      ;counter
[116]  qq                      ;storage of address
[117]  qq                      ;storage of div.
[118]  qq                      ;storage of mult.
[119]  qq                      ;storage of subtr.
[120]  qq                      ;storage of add.
[121]  qq                      ;storage of b
[122]  qq                      ;storage of a
[123]  qq                      ;storage of result
[124]  hs  r4  t44             ;RAND-2
[125]  xr  ,mkn  s1
[126]  mb  511 D
[127]  hr  s2
[128]  pm  r7,mln  r5          ;RAND-1
[129]  tk  8,gm  r6
[130]  ar  r5
[131]  sr  r3  LO
[132]  gr  r3,hr  s1
[133]  gtn  p,dln  r434
[134]  udn (p511),pc  -1
[135]  qq                      ;age of start random
[136]  hv  45  S<>             ;go to 6000 new cycles
e41
```

The program test reading and writing in cells 512-1023 in the fast memory. If a cell in this interval is mistaken for another cell in the same interval there will be no errorreaction , because the program use the same bitpattern in all cells during one testcycle.The program is stored from cell 41 to 131.

INPUT OF PROGRAM

The program is published in a condensed version (a B-tape) to be read into the machine by the basic inputprogram on channel 0,cells 34-39.After the input a sumcheck of the program is performed.If this check is ok. the program writes <CR,t 10 > on the typewriter with black ribbon, otherwise the programname will be typed in red,and the program stops in cell 49.

TESTDIGITS

After reading in and sumcheck the program will stop in cell 51 if KB=1.During this stop the registers R and In contains zeroes.If no bits are inserted the program itself will generate the testdigits for the run.If bits are inserted in R or In these bits will serve as testdigits during the run.In In only RA and RB gives sense.

BUILD IN TESTDIGITS

When the build in testdigits are used , 42 consecutive bits is taken from a group of bits containing:

$$\underbrace{1,1,1,1,\ldots,1}_{42\text{ bits}},\underbrace{0,0,0,0,\ldots,0}_{42\text{ bits}},\underbrace{1,1,1,1,\ldots,1}_{41\text{ bits}}$$

This means that each cell is automatically tested with 84 different (but known) bitpatterns.

OPERATORS TESTDIGITS

If the operator choose the testdigits each cell will be tested 81 times with the choosen bitcombination.The operator must insert the bits in R0-39 and the marks in In[RA,RB].

ERROR REACTIONS

During the running of the program , the normal condition of KA and KB is KA=KB=0.In this case there will be no error-output before one complete run is terminated.A complete run means writing and reading in cells 512-1023 of either the 84 generated bitpatterns or 81 times with the operators bits.

NO ERROR: If no error is registrated the run will terminate with the typed message <CR,ok> and a hoot at 1000 c/s in 5 seconds.

ERROR: If an error is registrated the run will terminate with the typed message <CR,-> and the frequency of the hoot will alternate between 950 c/s and 1100 c/s.

KA=1 will cause the program to stop when an error is detected in the bits 0-39 or the marks in the cell in question.There is two possible stoppoints in the program:

1) Stop with r1=122 [0001111010] indicates error in bits 0-39. The testbits used will be found in M.Bits in R set to 1 are errors.Depressing the START-button two times while the STOP-button is depressed will transfer the adress of the cell to p and the cellcontent to R0-39.Depressing START will continue the test.
2) Stop with r1=125 [0001111101] indicates error in the marks. The testmarks is stored in R8-9 and the marks of the cell in In[RA,RB].The adress of the cell is transferred to p. Depressing START will continue the test.

Setting KB=1 when the machine is stopped in an error , and depressing START , will cause cycling in a loop as long as KB=1.In this loop th content of M0-39 and In[RA,RB] is stored in the cell , and the content of the cell is read to R0-39 and In[QA,QB].

]

```
[Test of the fast memory,cell 512-1023]
i=  41
[  41] vy   16          ,psn  132
[  42] ps   s-1         ,ar   s
[  43] bs   s-41        ,hv   r-1 [42]   ;form checksum of the program
[  44] sy   62          ,tk   1
[  45] sy   29          NZ               ;if errorsum then redshift
[  46] sy   64          ,sy   19
[  47] sy               ,sy   1
[  48] sy   16          ,sy   62         ;outtext(outcr,«t 10»,blackshift)
[  49] zq               NZ               ;if errorsum then stop
[  50] pin                               ;R:=In:=0
[  51] zq               LKB              ;if KB then stop for manual testdigits
[  52] gr   r57 [109]   M                ;controlcell 40,41 := 0,0
[  53] pa   r35 [88]        t1           ;manualdigits:=true
[  54] gi   r2 [56]     V  LZ            ;if R=0 then c56adr:=In / goto c56
[  55] pa   r18 [73]    ,hv   r36 [91]   ;error:=false / goto operators testdigits
[  56] nc               ,hv   r-1 [55]   ;if In≠0 then goto c55
[  57] pa   r31 [88]    ,pa   r16 [73]   ;manualdigits:=false / error:=false
[  58] pm    D                           ;M:=0
[  59] pi               ,hs   101        ;In:=0 / test
[  60] pi   1           ,hs   101        ;In:=1 / test
[  61] pi   3           ,hs   101        ;In:=3 / test
[  62] pm   r37 [99]    ,hs   101        ;M 0-39 := all 1 / test
[  63] xrn              ,cl   -1         ;R:=M×2↑(-1)
[  64] hh   r-2 [62]    X  NZ           ;if M≠0 then begin M:=R;goto c62 end
[  65] pm   r34 [99]    ,pi   2          ;M 0-39 := all 1 / In:=2
[  66] hs   101                          ;test
[  67] pi               ,hs   101        ;In:=0 / test
[  68] xr               ,tk   1          ;R:=M×2
[  69] pm   r30 [99]    ,cm   r31 [100]  ;if R≠-1 then begin M:=R;goto c71 end
[  70] hh   r-3 [67]    X                ;else goto c67
[  71] gr   r38 [109]   XV MRC           ;controlcell 40,41 := 1,1 / M:=R
[  72] hs   101                          ;test
[  73] ps               ,sy   64         ;s:=error / outcr
[  74] can  s           ,hv   r2 [76]    ;if -,error then goto c76
[  75] sy   32          ,hv   r2 [77]    ;else writetext(«<-») / goto c77
[  76] sy   38          ,sy   34         ;writetext(«<ok»)
[  77] pa   r9 [86]         t9
[  78] pa   r5 [83]         t511         ;initialize hootduration
[  79] pa   r2 [81]         t7           ;initialize frequency-cycle
[  80] ar   s128        D                ;Radr := (if error then 8 else 0)+128+Radr
[  81] bt   7               t-1          ;for i:=7 step -1 until 1 do
[  82] hv   r-2 [80]                     ;goto c80
[  83] bt   511             t-1          ;for j:=511 step -1 until 1 do
[  84] hv   r-5 [79]                     ;goto c79
[  85] ns   s           ,ps   s          ;s:=-s
[  86] bt   9               t-1          ;for k:=9 step -1 until 1 do
[  87] hv   r-9 [78]                     ;goto  c78
[  88] bs                                ;if manualdigits then
[  89] hv   r-38 [51]   X                ;goto newtest(Operatorsdigits)
[  90] hv   r-40 [50]                    ;else goto newtest(buildindigits)
[  91] pm   r8 [99]     ,cm   r9 [100]   ;Operatorsdigits:
[  92] hv   r2 [94]                      ;if R≠-1 then goto c94
[  93] gr   r16 [109]      MRC           ;controllcell 0-39 := R ,40,41:=RA,RB
[  94] pa   r2 [96]     X      t80       ;M:=R / for i:=1 step 1 until 81 do
[  95] hs   101                          ;test
```

```
[  96] bt                          t-1
[  97] hv   r-2 [95]
[  98] hv   r-25 [73]                        ;test finished
[  99] udn (p-1)          XVDLRC t-1         ;constant 0-39 := all 1
[ 100] qq   -512                             ;procedure test;begin
[ 101] pa   r3 [104]               t511      ;entry to test
[ 102] pa   r1 [103]                         ;for j:=1023 step -1 until 512 do
[ 103] gm                     MRC  t-1       ;cell j 0-39:= M cellj 40-41 :=RA,RB
[ 104] bt   511                    t-1
[ 105] hv   r-2 [103]
[ 106] pa   r12 [118]              t511      ;for i:=1023 step -1 until 512 do begin
[ 107] pa   r7 [114]
[ 108] grn  r1 [109]      XV                 ;controlcell 0-39 := 0 / R:=testdigits
[ 109] qq
[ 110] sc   r-1 [109]     X                  ;controlcell:=-testdigits / M:=testdigits
[ 111] gi   r9 [120]                         ;c120adr:=In
[ 112] pi                 ,arn  r-3 [109]    ;In:=0 / R:=-testdigits
[ 113] srn  r-4 [109]        LC              ;if R=-1 then R:=-(-testdigits)
[ 114] ar                    IRC t-1         ;R:=R+cell[i] / In[RA,RB]:=cell i 40,41
[ 115] hv   r6 [121]         NZ              ;if R≠0 then goto errorincell
[ 116] gi   r1 [117]      ,arn  r4 [120]     ;c117adr := In / R:= old In
[ 117] nc                 ,hh   r6 [123]     ;if newmarks≠oldmarks then goto errorinmarks
[ 118] bt                          t-1       ;one cell tested
[ 119] hv   r-7 [112]
[ 120] pi                 ,hr   s1           ;In:=Original marks / end of procedure test;
[ 121] zq                    LKA             ;errorincell: if errorincell∧KA then stop
[ 122] pp  (r-8)[114]     ,arn  p            ;p:=adress of wrong cell / R:=wrong cell
[ 123] hv   r2 [125]      ,pp  (r-9)[114]    ;goto seterror / errorinmarks: p:=adress of wrong
[ 124] zq                    LKA             ;if error in marks ∧KA then stop
[ 125] pa   r-52 [73]              t8        ;seterror: error:=true
[ 126] pi  (r-6)[120]     ,hv   r3 [129]     ;In:=original marks
[ 127] gm   p                MRC             ;testdigits to cell
[ 128] arn  p                IQC             ;cell to R
[ 129] hv   r-2 [127]        LKB             ;if KB then cycle in 127-128
[ 130] pi  (r-10)[120]    ,hv   r-12 [118]   ;In:=Original marks / continue the test
[ 131] gmn  -319          X  LP  t488        ;checksum of program
s
```

[Testprogram t11 (ferrittest)

The program test reading and writing in cells 0-511 in the fast memory. If a cell in this interval is mistaken for another cell in the same interval there will be no errorreaction , because the program use the same bitpattern in all cells during one testcycle.The program is stored from cell 513 to 603 .

INPUT OF PROGRAM

The program is published in a condensed version (a B-tape) to be read into the machine by the basic inputprogram on channel 0,cells 34-39.After the input a sumcheck of the program is performed.If this check is ok. the program writes <CR,t 10 > on the typewriter with black ribbon, otherwise the programname will be typed in red,and the program stops in cell 521.

TESTDIGITS

After reading in and sumcheck the program will stop in cell 523 if KB=1.During this stop the registers R and In contains zeroes.If no bits are inserted the program itself will generate the testdigits for the run.If bits are inserted in R or In these bits will serve as testdigits during the run.In In only RA and RB gives sense.

BUILD IN TESTDIGITS

When the build in testdigits are used , 42 consecutive bits is taken from a group of bits containing:

1,1,1,1,....,1,0,0,0,0,....,0,1,1,1,1,....,1

42 bits 42 bits 41 bits

This means that each cell is automatically tested with 84 different (but known) bitpatterns.

OPERATORS TESTDIGITS

If the operator choose the testdigits each cell will be tested 81 times with the choosen bitcombination.The operator must insert the bits in R0-39 and the marks in In[RA,RB].

ERROR REACTIONS

During the running of the program , the normal condition of KA and KB is KA=KB=0.In this case there will be no error-output before one complete run is terminated.A complete run means writing and reading in cells 0-511 of either the 84 generated bitpatterns or 81 times with the operators bits.

NO ERROR: If no error is registrated the run will terminate with the typed message <CR,ok> and a hoot at 1000 c/s in 5 seconds.

ERROR: If an error is registrated the run will terminate with the typed message <CR,-> and the frequency of the hoot will alternate between 950 c/s and 1100 c/s.

KA=1 will cause the program to stop when an error is detected in the bits 0-39 or the marks in the cell in question.There is two possible stoppoints in the program:

1) Stop with r1=594[1001010010] indicates error in bits 0-39. The testbits used will be found in M.Bits in R set to 1 are errors.Depressing the START-button two times while the STOP-button is depressed will transfer the adress of the cell to p and the cellcontent to R0-39.Depressing START will continue the test.
2) Stop with r1=597[1001010101] indicates error in the marks. The testmarks is stored in R8-9 and the marks of the cell in In[RA,RB].The adress of the cell is transferred to p. Depressing START will continue the test.

Setting KB=1 when the machine is stopped in an error , and depressing START , will cause cycling in a loop as long as KB=1.In this loop th content of M0-39 and In[RA,RB] is stored in the cell , and the content of the cell is read to R0-39 and In[QA,QB].

```
[Test of the fast memory,cells 0-511]
i= -511
[-511] vy   16          ,psn  -420
[-510] ps   s-1         ,ar   s
[-509] bs   s511        ,hv   r-1 [-510]   ;form checksum of the program
[-508] sy   62          ,tk   1
[-507] sy   29            NZ               ;if errorsum then redshift
[-506] sy   64          ,sy   19
[-505] sy               ,sy   1
[-504] sy   1           ,sy   62           ;typetext(CR,<<t 11>>,blackshift)
[-503] zq                 NZ               ;if errorsum then STOP
[-502] pin                                 ;R:=In:=0
[-501] zq                 LKB              ;if KB=1 then STOP for manual testdigits
[-500] gr   r57 [-443]    M                ;controlcell 40,41:= 0,0
[-499] pa   r35 [-464]      t1             ;manualdigits:=true
[-498] gi   r2 [-496]   V  LZ              ;if R=0 then c528adr:=In / goto c528
[-497] pa   r18 [-479]  ,hv   r36 [-461]   ;error:=false / goto operators testdigits
[-496] nc               ,hv   r-1 [-497]   ;if In≠0 then goto c527
[-495] pa   r31 [-464]  ,pa   r16 [-479]   ;manualdigits:= false / error:= false
[-494] pm    D                             ;M:= 0
[-493] pi               ,hs   -451         ;In:= 0 / test
[-492] pi   1           ,hs   -451         ;In:= 1 / test
[-491] pi   3           ,hs   -451         ;In:= 3 / test
[-490] pm   r37 [-453]  ,hs   -451         ;M0-39:= all 1 / test
[-489] xrn              ,cl   -1           ;R:= M×2^(-1)
[-488] hh   r-2 [-490]  X  NZ              ;if M≠0 then begin M:=R;goto c534 end
[-487] pm   r34 [-453]  ,pi   2            ;M0-39:= all 1 / In:=2
[-486] hs   -451                           ;test
[-485] pi               ,hs   -451         ;In:= 0 / test
[-484] xr               ,tk   1            ;R:= M×2
[-483] pm   r30 [-453]  ,cm   r31 [-452]   ;if R≠-1 then begin M:=R;goto c543 end
[-482] hh   r-3 [-485]  X                  ;else goto c539
[-481] gr   r38 [-443]  XV MRC             ;controlcell 40,41:= 1,1 / M:=R
[-480] hs   -451                           ;test
[-479] ps               ,sy   64           ;s:= error / typetext(CR)
[-478] can  s           ,hv   r2 [-476]    ;if -,error then goto c548
[-477] sy   32          ,hv   r2 [-475]    ;else typetext(<<->>) / goto c549
[-476] sy   38          ,sy   34           ;typetext(<<ok>>)
[-475] pa   r9 [-466]       t9
[-474] pa   r5 [-469]       t511           ;initialize hootduration
[-473] pa   r2 [-471]       t7             ;initialize frequencycycle
[-472] ar   s128        D                  ;Radr:= Radr+128+(if error then 8 else 1)
[-471] bt   7               t-1            ;for i:= 7 step -1 until 1 do
[-470] hv   r-2 [-472]                     ;goto c552
[-469] bt   511             t-1            ;for j:= 511 step -1 until 1 do
[-468] hv   r-5 [-473]                     ;goto c551
[-467] ns   s           ,ps   s            ;s:= -s
[-466] bt   9               t-1            ;for k:= 9 step -1 until 1 do
[-465] hv   r-9 [-474]                     ;goto c550
[-464] bs                                  ;if manualdigits then
[-463] hv   r-38 [-501] X                  ;newtest(operators testdigits)
[-462] hv   r-40 [-502]                    ;else newtest(buildin testdigits)
[-461] pm   r8 [-453]   ,cm   r9 [-452]    ;operators testdigits:
[-460] hv   r2 [-458]                      ;if R≠ -1 then goto c566
[-459] gr   r16 [-443]    MRC              ;controlcell0-39:= R , 40-41:=RA-RB
[-458] pa   r2 [-456]   X     t80          ;M:=R / for i:= 1 step 1 until 81 do
[-457] hs   -451                           ;test
[-456] bt                   t-1
[-455] hv   r-2 [-457]
[-454] hv   r-25 [-479]                    ;test finished
[-453] udn (p-1)        XVDLRC t-1         ;constant0-39= all 1
```

```
[-452] qq   -512                                ;procedure test;begin
[-451] pa   r3 [-448]          t511             ;entry to test
[-450] pa   r1 [-449]          t-512            ;for j:= 511 step -1 until 0 do
[-449] gm                   MRC t-1             ;cellj0-39:= M, 40-41:=RA,RB
[-448] bt   511                t-1
[-447] hv   r-2 [-449]
[-446] pa   r12 [-434]         t511             ;for i:= 511 step -1 until 0 do begin
[-445] pa   r7 [-438]          t-512
[-444] grn  r1 [-443]       XV                  ;controlcell0-39:= 0 / R:=testdigits
[-443] qq
[-442] sc   r-1 [-443]      X                   ;controlcell:= -testdogits / M:= testdigits
[-441] gi   r9 [-432]                           ;c592adr:= In
[-440] pi                   ,arn  r-3 [-443]    ;In:= 0 / R:=-testdigits
[-439] srn  r-4 [-443]         LC               ;if R= -1 then R:=-(-testdigits)
[-438] ar                      IRC t-1          ;R:=R+cell i / In[RA,RB]:= cell i 40,41
[-437] hv   r6 [-431]          NZ               ;if R≠ 0 then goto error in cell
[-436] gi   r1 [-435]       ,arn  r4 [-432]     ;c589adr:= In / Radr:= old In
[-435] nc                   ,hh   r6 [-429]     ;if newmarks≠ old marks then goto error in marks
[-434] bt                          t-1          ;one cell tested
[-433] hv   r-7 [-440]
[-432] pi                   ,hr   s1            ;In:= original marks / end of procedure test;
[-431] zq                      LKA              ;error in cell: if error∧KA=1 then STOP
[-430] pp  (r-8)[-438]      ,arn  p             ;p:=adress of error / R:=errorcell
[-429] hv   r2 [-427]       ,pp  (r-9)[-438]    ;goto set error / error in marks: p:=erroradress
[-428] zq                      LKA              ;if error∧KA=1 then STOP
[-427] pa   r-52 [-479]            t8           ;set error: error:=true
[-426] pi  (r-6)[-432]      ,hv   r3 [-423]     ;In:= original marks
[-425] gm   p                  MRC              ;send testdigits to cell
[-424] arn  p                  IQC              ;read cell to R
[-423] hv   r-2 [-425]         LKB              ;if KB=1 then cycle in cells 599-600
[-422] pi  (r-10)[-432]     ,hv   r-12 [-434]   ;In:= original marks / continue with next cell
[-421] gmn  -228            X  LP  t-296        ;checksum of the program
s
```

INDICATOR TEST

Before start KA, KB are set to 0, 1. The test is supplied with a number of loops, so if any error occurs, KA is set to one, and the program will run in the loop, where the error appeared. All stops indicate errors. Cellnumber is found by subtraction of one from the content of r_1.

```
41   zq          LKB
42   grn  3oo
43   pi   -1
44   gi   3oo
45   arn  3oo
46   sr   -1  D
47   zq          NZ
48   hv   43     LKA   loop A, test of flip-flops for "ones"
49   pi
5o   gi   3oo    MOC
51   arn  3oo
52   zq          NZ
53   hv   49     LKA   loop B, test of flip-flops for "zeroes"
54   pi   -1           t-2 inhibit changing all except RB
55   zq          NRB
56   pi   -2           t-3 inhibit changing all except RA
57   zq          NRA
58   hv   6o     LRC   }
59   zq                }
6o   zq          LOA   }
61   zq          LOB   }
62   zq          LTA   }
63   zq          LTB   }  proving "L" indicator orders
64   zq          LPA   }
65   zq          LPB   }
66   zq          LQA   }
67   zq          LQB   }
68   gi   3oo ,arn  3oo
69   sr   3   D
7o   zq          NZ
```

```
71   qq          MOC   R40-41 are set to zero
72   ppn         IRB   dummy order , RB: = 0
73   ppn         IRA   dummy order , RA: = 0
74   zq          LRB
75   zq          LRA
76   qq          M     R40-41 are set to one
77   ck   1o     IRB   dummy order, RB: = 1
78   ck   1o     IRA   dummy order, RA: = 1
79   zq          NRA
8o   zq          NRB
81   hv   54     LKA   loop C, test of RA, RB
82   pi   -4           t-5 inhibit changing all except QB
83   zq          NQB
84   pi   -8           t-9 inhibit changing all except QA
85   zq          NQA
86   hv   88     LQC  }
87   zq                }
88   zq          LOA   }
89   zq          LOB   }
9o   zq          LTA   } proving "L and N" indicator orders
91   zq          LTB   }
92   zq          LPA   }
93   zq          LPB   }
94   zq          NRA   }
95   zq          NRB  }
96   gi       ,arn     3oo
97   sr   15  D
98   zq          NZ
99   gmn  4oo    IQB   dummy order, QB: = 0
1oo  gmn  4oo    IQA   dummy order, QA: = 0
1o1  zq          LQB
1o2  zq          LQA
1o3  qq          M
1o4  tk   2      IQB   dummy order, QB: = 1
1o5  tk   2      IQA   dummy order, QA: = 1
1o6  zq          NQB
1o7  zq          NQA
1o8  hv   82     LKA   loop D, test of QA, QB
```

```
1o9   pi    -16          t-17 inhibit changing all except PB
11o   zq         NPB
111   pi    -32          t-33 inhibit changing all except PA
112   zq         NPA
113   hv    115  LPC
114   zq
115   zq         LOA
116   zq         LOB
117   zq         LTA      proving "L" and "N" indicator orders
118   zq         LTB
119   zq         NQA
12o   zq         NQB
121   zq         NRA
122   zq         NRB
123   gi    3oo
124   arn   3oo
125   sr    63 D
126   zq         NZ
127   gmn   4oo  IPB      dummy order, PB: = 0
128   gmn   4oo  IPA      dummy order, PA: = 0
129   zq         LPB
13o   zq         LPA
131   qq         M
132   tl    4    IPB      dummy order, PB: = 1
133   tl    4    IPA      dummy order, PA: = 1
134   zq         NPB
135   zq         NPA
136   hv    1o9  LKA      loop E, test of PA and PB
137   pi    -64          t-65 inhibit changing all except TB
138   zq         NTB
139   pi    -128         t-129 inhibit changing all except TA
14o   zq         NTA
141   hv    143  LTC
142   zq
```

```
143   zq          LOA
144   zq          LOB
145   zq          NTA
146   zq          NTB
147   zq          NPA   } proving "L" and "N" indicator orders
148   zq          NPB
149   zq          NQA
15o   zq          NQB
151   zq          NRA
152   zq          NRB
153   gi   3oo
154   arn  3oo
155   sr   255  D
156   zq          NZ
157   qqn         ITB   TB: = 0
158   qqn         ITA   TA: = 0
159   zq          LTB
16o   zq          LTA
161   arn  -1   D ITB   TB: = 1
162   arn  -1   D ITA   TA: = 1
163   zq          NTB
164   zq          NTA
165   hv   137    LKA   loop F, test of TA and TB
166   pi   -256         t-257 inhibit changing all except OB
167   zq          NOB
168   zq          NZB
169   pi   -512         t-511 inhibit changing all except OA
17o   zq          NOA
171   zq          NZA
172   hv   174    LOC
173   zq
174   hv   176    LZC
175   zq
176   zq          NTA
177   zq          NTB   proving "L" and "N" indicator orders
178   zq          NPA
179   zq          NPB
18o   zq          NQA
181   zq          NQB
182   zq          NRA
183   zq          NRB
```

```
184   gi   3oo
185   arn  3oo
186   sr   -1    D
187   zq             NZ
188   arn            IOB   dummy order, OB: = 0
189   qqn            IOA   dummy order, OA: = 0
19o   zq             LOB
191   zq             LOA
192   arn  -1    D
193   ar   -512  D IOB                  OB: = 1
194   ar   -512  D IOA                  OA: = 1
195   zq             NOB
196   zq             NOA
197   hv   166       LKA   loop G, test of OA, OB
198   pi   -1
199   pi                   t-1
2oo   hv   2o2       LZC
2o1   zq
2o2   hv   2o4       LOC
2o3   zq
2o4   hv   2o6       LTC
2o5   zq                   } proving "L" indicator orders
2o6   hv   2o8       LPC
2o7   zq
2o8   hv   21o       LQC
2o9   zq
21o   hv   212       LRC
211   zq
212   arn  211       IZB
213   ar   211       IZA
214   zq             LZB
215   zq             LZA
216   pmn            IZC
217   zq             NZB
218   zq             NZA   } proving Z combinations
219   pin
22o   qq             IZB
221   ar   212
222   zq             NZB
223   pin
224   qq             IZA
225   ar   213
```

```
227   pi
228   hv   198      LKA   loop H, test of Z
229   hv   41
```

[Tromletest 4. This program writes random numbers on previous selected drum tracks, reads and compares the contents word by word incl. flag bits. Calculations may be performed simultaneously with the drum test by means of an adder test.

After the message ,klar, the typewriter is ready for input of 3 arbitrary numbers (each terminated by CR) intended for the production of the random test numbers.

The program has several adjustment variations. The p-register contains the track no. to be tested. The five last bits of the in-register have the following meaning:

KB=0 : Adder test simultaneously with the drum test.
KB=1 : Either adder test or drum test depending on when KB is set equal to 1.
KA=0 : No stop if adder error.
KA=1 : Stop if adder error.
The typewriter writes ,1, if adder test is ok, and ,3, if adder error.
RB=0 : No counting in the p-register. Test of the same track.
RB=1 : Counting in the p-register. Track no. is increased with 1.
RA=0 : Output of number of errors per track.
RA=1 : Output of number of errors totally only.
QB=0 : Output of the read and written bit pattern, track no. and word no. if error.
QB=1 : No output of the read and written bit pattern, track no. and word no.

By entry a normal adjustment is set intended for a routine test of tracks 1-319 and output only of the number of errors totally (cf. the instructions in cell 42). By installations with Three Drum Cabinet the instruction in cell 51 should be changed to: arn 959 D.

The only way to change the normal adjustment is to change the contents of the in- and p-registers manually and jump to cell 43. The program starts with the instruction ,zq 0 LKB, and stops only if the operator interferes.]

```
i= 41
[  41] zq          LKB        [ Stop if KB=1]
[  42] pi   7     ,pp         [ Normal adjustment. Start on track 1]
[  43] gp  53     ,hs   89    [ Jump to input of 3 numbers]
[  44] pt  61           t45   [ Initialising return jump to drum test]
[  45] hv 257     ,grn  76    [ Start adder test, clear error counter tot.]
[  46] pp  p1       LRB       [ Add 1 to track no.]
[  47] hs  187                [ Store inputed numbers]
[  48] hv  155    ,hv   191   [ Fill track, store numbers]
[  49] grn  77    ,hs   187   [ Clear error counter per track, store numbers]
[  50] hv  164    ,hv   199   [ Read track, transfer numbers]
[  51] arn 319    D           [ If more tracks]
[  52] ncp        ,hv   46    [ then jump to new test]
[  53] pp         ,sy   64    [ else reset start track, write CR]
[  54] arn  76    ,hs   203   [ Jump to output of number of errors totally]
[  55] sy   64    ,hh   45    [ Write CR, jump to new test]
[  56] pt   61          V163  [ Entry interrupt drum test]
[  57] pt   61          t168  [ Entry interrupt drum test]
[  58] pm  195    ,arn  196   [ Reset R and M before]
[  59] gs   61    ,hv   290   [ jump to adder test]
[  60] gm  195    ,gr   196   [ Entry interrupt adder test]
[  61] ps         ,hh         [ Jump to drum test]
[  62] qq
```

```
[  63] qq                          [ Working cells]
[  64] qq
[  65] qq
[  66] qq
[  67] qq
[  68] qq
[  69] qq
[  70] qq
[  71] 0/1023/1023/1023            [ Constant]
[  72]     1                       [ Constant]
[  73] qq
[  74] qq
[  75] qq
[  76] qq
[  77] qq


                                   [ Subroutine, produce random test number]
[  78] pm   68                     [ From the numbers in cells 66, 67, and 68]
[  79] arn  68    X                [ 3 numbers are produced to be placed in the]
[  80] mk   66                     [ same cells. The contents of R is on exit]
[  81] sc   67                     [ depending on the contents of these cells]
[  82] dl   68    X                [ and used as a test number. Possibly]
[  83] ac   66      IOA            [ overflow is indicated in respective OA]
[  84] ml   67                     [ and OB for later check of flag bits.]
[  85] ac   68      IOB
[  86] hr   s1


                                   [ Subroutine, input of 3 numbers]
[  87]   0                         [ Working cell]
[  88]  10                         [ Constant]
[  89] pa   106           t62      [ Reset start cell]
[  90] pa   107           t2       [ Reset counting]
[  91] vy   17    ,sy     64       [ Select typewriter as I/O, write CR]
[  92] sy   34    ,sy     35       [ Write k, write 1]
[  93] sy   49    ,sy     41       [ write a, write r]
[  94] sy   64                     [ Write CR]
[  95] pt   102           t106     [ Set jump address for positive number]
[  96] xrn        ,lyn    87       [ Clear M, input of 1 character]
[  97] nc   32    ,hv     100      [ If character ≠32 then jump]
[  98] pt   102           t105     [ else set jump address for negative number]
[  99] hh   96                     [ Jump to input of next character]
[ 100] ca   16    ,hvn    103      [ If character=0 then clear R and jump]
[ 101] ca   64                     [ If character=64]
[ 102] xr         ,hv              [ then exchange M and R and jump to storing]
[ 103] tk   -30   ,ml     88       [ else produce a decimal number in M]
[ 104] hh   96                     [ Jump to input of next character]
[ 105] mt   -1    D                [ If negative number then change sign]
[ 106] gr   62            t1       [ Store the decimal number in working cell]
[ 107] bt   2             t-1      [ Counter for input of 3 numbers]
[ 108] hv   95                     [ Jump within 3 numbers]
[ 109] hr   s1                     [ else exit]


                                   [ Subroutine, output of bit pattern from R]
[ 110] pa   121           t3       [ Prepare output in 4 groups]
[ 111] tl   -10                    [ Prepare output of 1 bit at a time]
[ 112] pa   118           t9       [ Prepare 10 positions in each group]
[ 113] mb   71                     [ Clear R pos. 0-9]
```

```
[ 114] tl   1                    [ Produce 1 bit]
[ 115] ca                        [ If it is 0]
[ 116] sy   16   V               [ then write 0]
[ 117] sy   1                    [ else write 1]
[ 118] bt   9          t-1       [ Counter for output of 10 pos.]
[ 119] hv   113                  [ Jump back within a group]
[ 120] sy                        [ else write SP]
[ 121] bt   3          t-1       [ Counter for output of 4 groups]
[ 122] hv   112                  [ Jump back within 4 groups]
[ 123] sy   1    V  LPA          [ else write ,1, or ,0, according to PA]
[ 124] sy   16                   [ which corresponds to bit 40]
[ 125] sy   1    V  LPB          [ Write ,1, or ,0, according to PB]
[ 126] sy   16                   [ which corresponds to bit 41]
[ 127] hr   s1                   [ Exit]


                                 [ Subroutine, output of track no. and word no.]
[ 128] sy        ,sy             [ Write SP, write SP]
[ 129] sy   34   ,sy   49        [ Write k, write a]
[ 130] sy   37   ,sy   49        [ write n, write a]
[ 131] sy   35   ,sy             [ write l, write SP]
[ 132] arn  p    D               [ Prepare output]
[ 133] ck        ,tl   -30       [ of track no.]
[ 134] hs   203                  [ Jump to output of track no.]
[ 135] sy        ,sy             [ Write SP, write SP]
[ 136] sy   38   ,sy   41        [ Write o, write r]
[ 137] sy   52   ,sy   37        [ write d, write n]
[ 138] sy   41   ,sy   59        [ write r, write .]
[ 139] srn  314  D               [ Prepare output]
[ 140] ar   170  ,tl   -30       [ of word no.]
[ 141] hs   203                  [ Jump to output of word no.]
[ 142] hr   s1                   [ Jump back to error output]


                                 [ Subroutine, error output]
[ 143] arn  72   ,ac   76        [ Add 1 to number of errors totally]
[ 144] ac   77                   [ Add 1 to number of errors per track]
[ 145] hv   180     LQB          [ If QB=1 then jump back without output]
[ 146] arn  69      IPC          [ else prepare output of test number]
[ 147] vy   16   ,sy   64        [ Select typewriter, write CR]
[ 148] sy   64                   [ Write CR]
[ 149] hs   110                  [ Jump to output of bit pattern]
[ 150] sy   64                   [ Write CR]
[ 151] arn (170)    IPC          [ Prepare output of read test number]
[ 152] hs   110                  [ Jump to output of bit pattern]
[ 153] hs   128                  [ Jump to output of track no. and word no.]
[ 154] hv   180                  [ Jump back to next test]


                                 [ Fill drum track]
[ 155] pa   158         t215     [ Reset start address]
[ 156] pa   159         t39      [ Reset counting]
[ 157] hs   78                   [ Jump to produce random test number]
[ 158] gr   215     MOC t1       [ Store result incl. flag bits]
[ 159] bt   39          t-1      [ Counter for 40 words]
[ 160] hv   157                  [ Jump back within a track]
[ 161] vk   p    ,sk   216       [ else select actual track no., write on track]
[ 162] hv   56      NKB          [ Interrupt if KB=0]
[ 163] vk   p    ,hh   48        [ Exit]
```

```
                                        [ Read track and check]
[ 164] pa   180          t39            [ Reset counting]
[ 165] pa   170          t313           [ Reset start address]
[ 166] vk   p      ,lk   314            [ Select actual track no., read track]
[ 167] hv   57       NKB                [ Interrupt if KB=0]
[ 168] vk   p      ,hs   78             [ Jump to produce random test number]
[ 169] gr   69       MOC                [ Store test number incl. flag bits]
[ 170] sr   313          t1             [ Subtract read number]
[ 171] hv   143      NZ                 [ Jump to error output if R pos. 0-39≠0]
[ 172] hv   175      LOB
[ 173] hv   143      LB                 [ Jump to error output if OB=0∧b=1]
[ 174] hv   176
[ 175] hv   143      NB                 [ Jump to error output if OB=1∧b=0]
[ 176] hv   179      LOA
[ 177] hv   143      LA                 [ Jump to error output if OA=0∧a=1]
[ 178] hv   180
[ 179] hv   143      NA                 [ Jump to error output if OA=1∧a=0]
[ 180] bt   39           t-1            [ Counter for 40 words]
[ 181] hv   168                         [ Jump back within a track]
[ 182] hh   50       LRA                [ else if RA=1 then return jump]
[ 183] vy   16     ,sy   64             [ else select typewriter, write CR]
[ 184] arn  77                          [ Prepare output of number of errors per track]
[ 185] hs   203                         [ Jump to output of number of errors]
[ 186] sy   64     ,hh   50             [ Write CR, exit]


                                        [ Transfers of working cells]
[ 187] arn  63     ,gr   66             [ Storing of the 3 inputed numbers]
[ 188] arn  64     ,gr   67
[ 189] arn  65     ,gr   68
[ 190] hr   s1
[ 191] arn  66     ,gr   73             [ Storing of the 3 numbers produced by]
[ 192] arn  67     ,gr   74             [ passage of Produce random test number]
[ 193] arn  68     ,gr   75
[ 194] hv   49
[ 195] qq                               [ Working cells]
[ 196] qq
[ 197] qq
[ 198] qq
[ 199] arn  73     ,gr   63             [ Results for later use as start numbers]
[ 200] arn  74     ,gr   64
[ 201] arn  75     ,gr   65
[ 202] hv   51


                                        [ Output of number of errors]
[ 203] pa   210          t4             [ Adjustment for 5 digits]
[ 204] dk   215                         [ Transform to machine number]
[ 205] ar   214    X                    [ Possibly round off, exchange M and R]
[ 206] mln  213                         [ Prepare ]
[ 207] tk   30     ,ga   209            [ output of 1 digit]
[ 208] sy   16     V  LZ                [ If digit=0 then write 0]
[ 209] sy                               [ else write digit]
[ 210] bt   4            t-1            [ Counter for output of 5 digits]
[ 211] hv   206                         [ Jump within 5 digits]
[ 212] hr   s1                          [ else exit]


[ 213]  10                              [ Constants]
[ 214]  1
[ 215]  100000
[ 216] qq                               [ 40 cells for the instruction sk 216]
```

Kan brugesaltså som addertest.

```
i= 256                                  [ Simultaneous calculating by adder test 5]
[ 256] gr   309                         [ Only intended for corrections]
[ 257] arn  304   ,vy   16              [ Entry by start of adder test]
[ 258] gr           M                   [ Set address for floating-point overflow]
[ 259] pan  296   X       t1
[ 260] arfn 305
[ 261] grf  310
[ 262] arfn 306
[ 263] grf  311
[ 264] arfn 307
[ 265] grf  312                         [ Transfers of test numbers]
[ 266] arfn 308                         [ Adder test]
[ 267] grf  313
[ 268] arf  310
[ 269] mkf  311
[ 270] acf  312
[ 271] anf  313
[ 272] mlf  310   X
[ 273] scf  311
[ 274] srf  312
[ 275] mtf  313
[ 276] grf  310
[ 277] snf  311
[ 278] dkf  312
[ 279] gmf  313
[ 280] dlf  310
[ 281] ar   310
[ 282] mk   311
[ 283] ac   312
[ 284] an   313
[ 285] ml   310   X
[ 286] sc   311
[ 287] sr   312
[ 288] tl   -1
[ 289] hk   60        NKB               [ Jump to drum test if transfer is finished]
[ 290] mt   313
[ 291] gr   310
[ 292] sn   311
[ 293] dk   312
[ 294] gm   313
[ 295] dl   310
[ 296] bt   1               t1          [ Counter for 512 passages]
[ 297] hv   268                         [ Jump back within 512 passages]
[ 298] sr   309   ,ck                   [ else subtract result]
[ 299] sy   1         LZ                [ Write ,1, if adder is ok]
[ 300] hv   259       LZ                [ Jump to new test if adder is ok]
[ 301] zq             LKA               [ Stop if adder error^KA=1]
[ 302] sy   3                           [ Write ,3, if adder error]
[ 303] hv   259                         [ Jump to new test]
[ 304] hv   269
[ 305]  1234567890                      [ Test number]
[ 306]  2345678901
[ 307]  3456789012
[ 308]  4567890123
[ 309]  275/446/57/624                  [ Result of adder test]
[ 310]  0
[ 311]  0
[ 312]  0
[ 313]  0
[ 314]  0                               [ 40 cells for the instruction lk 314]
```

Test C-16

GIER-CP

Technical Center, 26.9.1968
JFM

The program is used for test of Mode 5 in GIER. Mode 5 is executed whenever the HP-button is pressed. The program is published as a B-C program only.

Method

After start (in Core[41]) the following happens:

1. Track 0 is saved in the core store during the test. Core[128:167]:=Track[0];
2. A special track 0 (part of the program) is written.
3. Track 38 is destroyed with special information.
4. The instructions

   [512] vy 0, vk 87
   [513] hv 512

   are moved to Core[512:513] and entered.

The program will now loop in 512 and 513 until the HP-button is pressed. When this happens the following test sequence is executed:

1. Core[0:39] is stored on track 38 (by Mode 5)
2. Track 0 is read to the core store (Core[0:39]:=Track[0]). (by Mode 5)
3. The instruction counter CT (r1) is stored in the address part of Core[0] and the remaining bits cleared. (by Mode 5);
4. Core[1] is enterdd in either rigth halfword or in left halfword. The entry depends on the state of the h-bit when HP was pressed. (by Mode 5);
5. Now the functions 1-4 above are checked by the program on track 0.
6. A jump to the instructions in Core[512:513] is performed and the test continue.

In case of errors, messages are given on the typewriter.

Use of KA,KB

KB=1	The program stops in Core[41] when the tape has been read.
KA=0	The normal state of KA during the test.
KA=1	After press on HP-button and the following test, the old track 0 is reestablished. One more press on HP will then activate the normal function of track 0 and the test is finished.

Error Messages

The following five messages are possible during the test, and the meaning is considered evident:

1. wrong tk stored
2. wrong r1 stored
3. -entry in Core[1]
4. Core[0:39] not stored on track 38
5. track 0 not transferred to core store

```
;slip<
;Test of HP function and mode 5, 1.8.68.4

b i=41, a40

[Core part of test, executed after track 0]

        zq          LKB          ; if kbon then STOP
        vy    512, grn    -1 ; set HP-inhibit; sum:=0;
        pa     a7   t    207 ; setsum of track 38:
        pa     a8   t     38 ;
a7:     arn   207   t      1 ;
        ar    1.8   DLA       ; include marks in sum
        ar    1.9   DLB       ;
        sc     -1             ; sum:=sum-R;
a8:     bt     38   t     -1 ;
        hv     a7             ;
        pm     -1, gm    228 ;
        vk      0   lk   128 ; save present track 0 in core;
        sk    168, vk    38 ; set new track zero;
        sk    208, lk     0
        sk    288, vk     0 ; noise to track 38 after setting core;
        pm     a9, arn   a10 ;
        gm    512   HA       ; set fixed program in core;
        gr    513   H        ; program used for interrupt by HP;
        hv    512            ; goto program;

a9:
[512]   vy      0  vk      87 ; release HP-inhibit; set tk to 87;
a10:
[513]   hv    512              ; goto Core[512];


a1:     gr    a12              ; saveR:=R; error in tk:
        hs    a11              ; writetext(<<
        qq    a13              ; wrong tk stored>);
        arn   a12, hr     s1 ; R:=saveR; return;

a2:     gr    a12              ; saveR:=R; error in 0T:
        hs    a11              ; writetext(<<
        qq    a14              ; wrong r1 stored>);
        arn   a12, hr     s1 ; R:=saveR; return;

a3:     hs    a11              ; improper entry to track 0: writetext(<<
        qq    a15              ; -entry in Core[1]>);
        hr     s1              ; return;

a6:     hs    a11              ; errror track 38: writetext(<<
        qq    a16              ; Core[0:39] not stored on track 38>);
        hr     s1              ; return;

a13:    k6b 58twrong tk stored;58.
a14:    k6b 58twrong r1 stored;58.
a15:    k6b 58t-entry in Core[1];58.
a16:    k6b 58tCore[0:39] not stored on track 38;58.
```

```
a11:  [textprint of HELP 3]
      gs     r5, an      s ;
      is     s1   MA      ;
      ps    (s), arn   r-2 ;
      pm      s, ps     s1 ;
      cl     -6; ca    10.5 ;
      ps     -1  hr     s1 ; return;
      ca   15.5, hh    r-h ;
      tk     -h, ga     r2 ;
      ca     63, it      1 ;
      sy      0, hvm   r-5 ;

a12:  qq      0, qq      0 ; saveR;
```

d i=168

[New track 0, replacing the existing in the machine.
Handles and checks all actions taken by the HP-button
and mode 5]

```
[ 0]  qq      0   t    517 ; used for save CT;
      it    311, pt      0 ; normal entry: set bits(10-19) of Core[0];
      gk     -1, arn    -1 ; save by and tk; Rcount:=tk;
      vy    529, tk     10 ; set HP-inhibit;
      nc     87, hs     a1 ; if tk≠87 then call error;
      arn     0             ; R:=Core[0];
      ca    512, hv   ra17 ;
      ca    513  hv   ra17 ;
      ca      1, hv   ra17 ;
      ca      2, hv   ra17 ;
      ca      3, hv   ra17 ;
      hs     a2            ;
a17:  tk     10            ;
      nc    311, ca      0 ;
      hv   ra18            ;
      hs     a3            ;
a18:  vk     38, lk    288 ; read stored track 38 to core;
      vk     38, grn    -2 ; sum:=0;
      pa    ra4   t    287 ; setsum of track 38 in core[-2];
      pa    ra5   t     39 ;
a4:   arn   287   t      1 ;
      ar    1.8   DLA      ;
      ar    1.9   DLB      ;
[20]  sc     -2             ;
a5:   bt     39   t     -1 ;
      hv    ra4             ;
      arn    -2             ; R:=computed sum;
      hv   ra19   NZ        ; if R ≠ 0 then call error;
a20:
[27]  sk    208, lk      0 ; switch to new track 38;
[28]  hk     r1, hv      r ;
a19:  arn   289, nc    311 ;
      hs     a6            ;
      hv   ra20            ;
      qq                   ;
      qq                   ;
      qq                   ;
      qq                   ;
[34]  prn   1.3   DMIZA     ; Primitive input of HELP:
      tl     -7, ly     r1 ;
      pi      0   LZA t508 ;
      xr      0   XIZB     ;
      hv     35   LZB      ;
[39]  cr     41   MRC  t-1 ;
```

```
d i=208
[Core 0:39 during the test. Stored on track 38 after HP.
The storing is performed by mode 5.]

[ 0] qq                          ;
     vy    529, sy     64 ; writetext(<<
     sy     58, sy     19 ; track 0 not transferred to core store>);
     sy     41, sy     49 ;
     sy     51, sy     34 ;
     sy      0, sy     16 ;
     sy      0  sy     37 ;
     sy     38, sy     19 ;
     sy      0  sy     10 ;
     sy     41, sy     49 ;
[10] sy     37, sy     18 ;
     sy     54  sy     53 ;
      sy     41, sy     41 ;
     sy     53  sy     52 ;
     hv     26              ;
     qq                     ;
     qq                     ;
     qq                     ;
     qq                     ;
     qq                     ;
[20] qq                     ; sumbalance for track;
     qq                     ;
     qq                     ;
     qq                     ;
     qq                     ;
     qq                     ;
     qq                     ;
     qq                     ;
[28] hk      r1, hv      r  ; wait for transfer from drum;
     vk     38, sk    288  ; noise to track 38;
     hv    512   MZA        ; if - keen then exit to core;
     vk      0, sk    128  ; regenerate track 0 of machine;
[30] vk      0, hv    512  ; exit to core;
     qq                     ;
[34] prn   1.3   DMIZA      ; Primitive input of MJELP:
     tl     -7, lv     r1  ;
     pi      0   LZA t508   ;
     xr      0   XIZB       ;
     hv     35    LZB       ;
[39] gr     41   MRC t-1    ;

s
e41
```

```
 25   1.8.68  849
start,image<
exit,10<
p26   1.8.68  e11
r<

1
?9h  h1   2h8  300
a1    62
a2    66
a3    70
a4    188
a5    192
a6    73
a7    h5
a8    h9
a9    60
a10   61
a11   95
a12   105
a13   76
a1h   79
a15   82
a16   87
a17   180
a18   18h
a19   198
a20   196

 27   1.8.68  e42
binout,0,h1..512<
print,p1..75,t75..9h,p95..105,p168..2h7<
```

```
41. zq                        LKB
42. vy   512            ,grn  -1
43. ra   45                    t207
44. ra   49                    t39
45. grn  207                   t1
46. gr   2          D   LA
47. ar   1          D   LB
48. gc   -1
49. bt   39                    t-1
50. hv   45
51. pm   -1             ,gm    228
52. vk                  ,lk    128
53. sk   168            ,vk    38
54. sk   208            ,lk
55. sk   288            ,vk
56. pm   60             ,grn   61
57. gm   512               MA
58. gr   513               M
59. hv   512
60. vy                  ,vk    87
61. hv   512
62. gr   105
63. hs   95
64. qq   76
65. grn  105            ,hr    s1
66. gr   105
67. hs   95
68. qq   79
69. grn  105            ,hr    s1
70. hs   95
71. qq   82
72. hr   s1
73. hs   95
74. qq   87
75. hr   s1

 76.
wron
 77. g tk s
 78. tored
 79.
wron
 80. g r1 s
 81. tored
 82.
-ent
 83. ry in
 84. Core
 85. [1]
 86.
 87.
Co
 88. ra[0
 89. :39
 90. ] not
 91.  store
 92. d on t
 93. rack 3
 94. 8

 95. gs   r5 [100]     ,an    s
 96. is   s1              MA
```

```
 98. pm   s              ,ps   s1
 99. cl   -6             ,ca   160
100. ps   -1             ,hr   s1
101. ca   210            ,hh   r-4 [97]
102. tk   -4             ,ca   r2 [101]
103. ca   63             ,it   1
104. sy                  ,hvn  r-5 [99]
105. qq                  ,qq

168. qq                        t517
169. it   311            ,pt
170. gk   -1             ,crn  -1
171. vy   529            ,tk   10
172. nc   87             ,hs   62
173. crn
174. ca   512            ,hv   r6 [180]
175. ca   513            ,hv   r5 [180]
176. ca   1              ,hv   r4 [180]
177. ca   2              ,hv   r3 [180]
178. ca   3              ,hv   r2 [180]
179. hs   66
180. tk   10
181. nc   311            ,ca
182. hv   r2 [184]
183. hs   70
184. vk   38             ,lk   288
185. vk   38             ,crn  -2
186. pa   r2 [188]             t287
187. pa   r5 [192]             t39
188. crn  287                  t1
189. ar   2              D  LA
190. ar   1              D  LB
191. sc   -2
192. bt   39                   t-1
193. hv   r-5 [188]
194. crn  -2
195. hv   r3 [198]         MZ
196. gk   208            ,lk
197. hk   r1 [198]       ,hv   r [197]
198. crn  ≤ ε89          ,nc   311
199. hs   73
200. hv   r-4 [196]
201. qq
202. qq
203. qq
204. qq
205. pmn  64             XD IZA
206. tl   -7             ,ly   r1 [207]
207. pi                    LZA t503
208. qq
209. vy   529            ,sy   64
210. sy   58             ,sy   19
211. sy   41             ,sy   40
212. sy   51             ,sy   34
213. sy                  ,sy   16
214. sy                  ,sy   37
215. sy   38             ,sy   19
216. sy                  ,sy   19
217. sy   41             ,sy   40
218. sy   37             ,sy   18
219. sy   54             ,sy   53
220. sy   41             ,sy   41
221. sy   53             ,sy   52
222. hv   26
223. qq
224. qq
```

```
227. qa
228. qq
229. qq
230. qq
231. qq
232. qq
233. qq
234. qq
235. qq
236. hk   r1 [237]      .hv   r [236]
237. vk   38            .sk   288
238. hv   512              IKA
239. vk                 .sk   128
240. vk                 .hv   512
241. qq
242. rmn  64            XD IZA
243. tl   -7            .ly   r1 [244]
244. pi                    LZA t508
245. xr                 X  IZB
246. hv   35               LZB
247. gr   h1               MRC t-1
```

; slip<

[Diagnostic program t 21 test of mode 1 and 4 oct.68 PEH

The program tests all possible combinations of the conditions in mode 1 and 4. If an error is detected output is given on typewriter according to the following table

output	type of non-successful function
g0, g1, g2	basic operation fullword, LH, RH halfword
a0, a1, a2	adress part of - - - - - -
n0, n1, n2	n-mark (S-mark) - - - - - -
i0, i1, i2	indirect adressing - - - - -
r0, r1, r2	relative-mark - - - - - -
s0, s1, s2	subroutinemark - - - - - -
p0, p1, p2	p-index - - - - - -
B	indicator condition
D	D-mark
f	floating-point mark
H	halfword mark added
h	- - - deleted
IK	IKC operation
K	K-part of indicator (bit pos.37)
T	T-part of indicator (bit pos.35)
Z	Z-part of indicator (bit pos.36)
L	LA or LB operation (bit pos.33, 34, 38, 39)
X	X-modification
V	V-modification
q	incremental operation
qs	incremental operation is erroneously treated as static or vice-versa

The letters may be combined, thus the message

isB

means that a mode 1 or 4 error is detected during execution of an instruction which is indirect adressed, s-marked and furnished with some indicator condition.

If KA is set during run the error message will be followed by a three-digit number which gives the adress of the cell from which the alarm was called.

Pressing of KB causes the machine to stop.

If no errors are detected an ok message is typed every 20 seconds.

Input of program.

The program may be read in like any other B or C test. However, if this does not work the program may be read in using as simple instructions as possible in this way:

1. RESET GIER.
2. place the tape in the reader starting in the middle of the 100 spaces found a few inches after the starting point used for old track 0.
3. clear the entire core store using hardware test 1.
4. clear R, M and by.
5. insert in cell 41 - 43 the following instructions:

41.	ly 43	bit pos. 4, 6,8, 9, 20, 21, 22, 24, 25
42	cl 76	- - 3, 6, 7, 21, 23
43.	gm	- - 21, 22, 24

6. start GIER in cell 41

When the program has been read in the primitive input program in cell 34 - 39 is re-established.

i=27

```
[  27]      pm i+5      ;
[  28]      gm -2, M    ;
[  29]      pm i+4      ;
[  30]      gm 0   M    ;
[  31]      hv 40       ;
[  32]      ar a1       ;
[  33]      tk 42       ;
[  34]      pmm64XDIZA; primitive input
[  35]      tl -7,lyr1;
[  36]      pi LZAt508;
[  37]      xr X IZB    ;
[  38]      hv 35 LZB   ;
[  39]      gr41MRCt-1;
[  40]      tk 42       ;
[  41]      qq          ;
[  42]      vy 16       ;
[  43]      sy 29       ; red ribbon
[  44]      sy 58       ; LC
[  45]      sy 64       ;
[  46]      hv a0       ; goto start


;     alarm print

[  47]b0:   qq          ; work
[  48]      qq          ; constant
[  49]      sy 55       ; g
[  50]      qq          ;
[  51]      sy 49       ; a
[  52]      hv b1       ;
[  53]      sy 37       ; n
[  54]      hv b1       ;
[  55]      sy 57       ; i
[  56]      hv b1       ;
[  57]      sy 41       ; r
[  58]      hv b1       ;
[  59]      sy 18       ; s
[  60]      hv b1       ;
[  61]      sy 39       ; p
[  62]b1:   sy 16       ; [full-word]
[  63]      hv a2       ; goto s-print
[  64]      sy 55       ; g
[  65]      hv b2       ;
[  66]      sy 49       ; a
[  67]      hv b2       ;
[  68]      sy 37       ; n
[  69]      hv b2       ;
[  70]      sy 57       ; i
[  71]      hv b2       ;
[  72]      sy 41       ; r
[  73]      hv b2       ;
[  74]      sy 18       ; s
[  75]      hv b2       ;
[  76]      sy 39       ; p
[  77]b2:   sy  1       ; 1 [LH half-word]
[  78]      hv a2       ; goto s-print
[  79]      sy 55       ; g
```

```
[  80]      hv b3      ;
[  81]      sy 49      ; a
[  82]      hv b3      ;
[  83]      sy 37      ; n
[  84]      hv b3      ;
[  85]      sy 57      ; i
[  86]      hv b3      ;
[  87]      sy 41      ; r
[  88]      hv b3      ;
[  89]      sy 18      ; s
[  90]      hv b3      ;
[  91]      sy 39      ; p
[  92]b3:   sy  2      ; 2 [RH half-word]
[  93]      hv a2      ; goto s-print
[  94]      sy 60      ; B [indicator condition]
[  95]      sy 50      ;
[  96]      hv a2      ;
[  97]      sy 60      ; D
[  98]      sy 52      ;
[  99]      hv a2      ;
[ 100]      sy 54      ; f [floating-point mark]
[ 101]      hv a2      ;
[ 102]b4:   sy 60      ; H
[ 103]      sy 56      ; h
[ 104]      hv a2      ;
[ 105]      sy 60      ; IK [IKC operation]
[ 106]      sy 57      ;
[ 107]      sy 60      ; K
[ 108]      sy 34      ;
[ 109]      hv a2      ;
[ 110]      sy 60      ; L
[ 111]      sy 35      ;
[ 112]      hv a2      ;
[ 113]b5:   sy 40      ; q
[ 114]      hv a2      ;
[ 115]      sy 40      ; qs
[ 116]      sy 18      ; [static op treated as incremental
[ 117]      hv a2      ; or vice-versa]
[ 118]      sy 60      ; T
[ 119]      sy 19      ;
[ 120]      hv a2      ;
[ 121]      sy 60      ; X
[ 122]      sy 23      ;
[ 123]      hv a2      ;
[ 124]b6:   sy 60      ; XV
[ 125]      sy 23      ;
[ 126]      sy 60      ; V
[ 127]      sy 21      ;
[ 128]      hv a2      ;
[ 129]      sy 60      ; Z
[ 130]      sy 25      ;

;     s-print

[ 131]a2:   sy 58      ;
[ 132]      hv a3 NKA  ;
[ 133]      sy  0      ;
[ 134]      tk 42      ;
[ 135]      ar c2      ;
[ 136]      gr c3      ;
[ 137]      tl 80      ; R:=M:=0
[ 138]      gs i+1     ;
```

```
[ 139]      qq [s]     ;
[ 140]      ar i-1     ;
[ 141]      cl 10      ;
[ 142]      ml c       ;
[ 143]c4:   ck -10     ;
[ 144]      ca  0      ;
[ 145]      hv i+2     ;
[ 146]      hv i+3     ;
[ 147]      arn i+1    ;
[ 148]      qq 16      ;
[ 149]      ga i+1     ;
[ 150]      sy         ;
[ 151]      tk 42      ;
[ 152]      ar c3      ;
[ 153]      ar a1      ;
[ 154]      gr c3      ;
[ 155]      ca  0      ;
[ 156]      hv i+3     ;
[ 157]      ac c6      ;
[ 158]      hv a3      ;
[ 159]      ml c1      ;
[ 160]      hv c4      ;
[ 161]c0:   m 10-2     ;
[ 162]c1:   qq 10.39   ;
[ 163]c2:   qq 2.39    ;
[ 164]c3:   qq         ; work

;     message counter

[ 165]a3:   sy  0      ;
[ 166]      tk 42      ;
[ 167]      ar c6      ;
[ 168]      ar a1      ;
[ 169]      gr c6      ;
[ 170]      ca  0      ;
[ 171]      hv a0      ;
[ 172]      sy 64      ;
[ 173]      ar c5      ;
[ 174]      gr c6      ;
[ 175]      hv a0      ;
[ 176]c5:   qq 20.39   ;
[ 177]c6:   qq 19.39   ;

;     reset and start

[ 178]a0:   tk 42      ;
[ 179]      ar c7      ;
[ 180]      gr c8      ; reset ok-counter
[ 181]      pp  0      ;
[ 182]      ps  0      ;
[ 183]      hv i+2     ;
[ 184]a1:   qq -1.39   ; constant
[ 185]      tk 42      ;
[ 186]      hv i+2 NKB;
[ 187]      zq  LKB    ;
[ 188]      ud -2      ; test of adress, fullword
[ 189]      sr a1      ;
[ 190]      ca  0      ;
[ 191]      hv i+2     ;
[ 192]      hs 2b      ; print(ga0)
[ 193]      qqtb4XVD   ; [qq  ,hv b4], test of +half-word mark, print(H)
```

```
[ 194]      qq          ; dummy
[ 195]a4:   ar a1       ; test of LA, LB
[ 196]      hs 8b4 LA   ; print(L)
[ 197]      hs 8b4 LB   ; print(L)
[ 198]      pi -1       ; test of T, Z, K
[ 199]      hs 5b5NTC   ; print(T)
[ 200]      hs 5b6 LZ   ; print(Z)
[ 201]      tk 42       ;
[ 202]      hs 5b6 NZ   ; print(Z)
[ 203]      hs 5b4 LC   ; print(K)
[ 204]      tk 42       ; R:=0
[ 205]a5:   pm a1       ; test of X, M:=-1.39
[ 206]      cl 40  X    ;
[ 207]      hv i+2 LZ   ;
[ 208]      hs 8b5 NZ   ; print(X)
[ 209]      pm a1  X    ; R:=-1.39, M:=0
[ 210]      sr a1       ; R:=0
[ 211]      hv i+2 LZ   ;
[ 212]      hs 8b5 NZ   ; print(X)
[ 213]      tk 42  X    ; M:=0
[ 214]      xr          ; R:=0
[ 215]a6:   hv i+2 LZ   ;
[ 216]      hs 8b5 NZ   ; print(X)
[ 217]      ar a1  X    ; M:=-1.39
[ 218]      xr          ; R:=-1.39
[ 219]      sr a1       ; R:=o
[ 220]      hv i+2 LZ   ;
[ 221]      hs 8b5 NZ   ; print(X)
[ 222]      qq     V    ; test of V
[ 223]      hs 2b6      ; print(V)
[ 224]      pm a1  XV   ; test of XV
[ 225]a7:   hs b6       ; print(XV)
[ 226]      sr a1       ;
[ 227]      hv i+2 LZ   ;
[ 228]      hs b6  NZ   ; print(XV)
[ 229]      arf a1  X   ; test of floating-point mark
[ 230]      hv i+2 NZ   ;
[ 231]      hs 8b3      ; print(f)
[ 232]      tk 42       ;
[ 233]      ar a1  D    ; test of D
[ 234]      ca a1       ;
[ 235]      hv i+2      ;
[ 236]      hs 5b3      ; print(D)
[ 237]      tk 42       ;
[ 238]      ar a1       ; test of n, full-word
[ 239]a8:   qqn         ;
[ 240]      hv i+2 LZ   ;
[ 241]      hs 6b  NZ   ; print(n0)
[ 242]      ca (1b)     ; test of i, full-word
[ 243]      hv i+2      ;
[ 244]      hs 8b       ; print(i0)
[ 245]      tk 42       ;
[ 246]      ar ra1      ; test of r, full-word
[ 247]      sr a1       ;
[ 248]      hv i+2 LZ   ;
[ 249]a9:   hs 10b NZ   ; print(r0)
[ 250]      ps a1       ; test of s, full-word
[ 251]      ar s        ;
[ 252]      sr ai       ;
[ 253]      ps  0       ;
[ 254]      hv i+2 LZ   ;
[ 255]      hs 12b NZ   ; print(s0)
[ 256]      pp a1       ; test of p, full-word
```

```
[ 257]      ar  p      ;
[ 258]      sr a1      ;
[ 259]a10:  pp  0      ;
[ 260]      hv i+2 LZ  ;
[ 261]      hs 14b NZ  ; print(p0)
[ 262]      ar a1      ; test of LH hailf-word
[ 263]      ps 42      ;
[ 264]      ud   ,     ; LH, basic op
[ 265]      ps  0      ;
[ 266]      hv i+2 LZ  ;
[ 267]      hs 2b1 NZ  ; print(g1)
[ 268]      ps -1      ; test of LH adress
[ 269]a11:  ud -2,     ;
[ 270]      ps 0       ;
[ 271]      sr a1      ;
[ 272]      hv i+2 LZ  ;
[ 273]      hs 4b1 NZ  ; print(a1)
[ 274]      ar a1      ; n, LH
[ 275]      qqn  ,     ;
[ 276]      hv i+2 LZ  ;
[ 277]      hs 6b1 NZ  ; print(n1)
[ 278]      pi i-4     ;
[ 279]      gi i+1     ;
[ 280]      ar (i-6),  ; i, LH
[ 281]a12:  sr a1      ;
[ 282]      hv i+2 LZ  ;
[ 283]      hs 8b1 NZ  ; print(i1)
[ 284]      pi a1-i-2  ;
[ 285]      gi i+1     ;
[ 286]      ar ra1,    ; r, LH
[ 287]      sr a1      ;
[ 288]      hv i+2 LZ  ;
[ 289]      hs 10b1NZ  ; print(r1)
[ 290]      ps a1      ;
[ 291]      ar  s,     ;
[ 292]      sr a1      ;
[ 293]a13:  ps. 0      ;
[ 294]      hv i+2 LZ  ;
[ 295]      hs 12b1NZ  ; print(s1)
[ 296]      pp a1      ;
[ 297]      ar  p,     ;
[ 298]      sr a1      ;
[ 299]      pp  0      ;
[ 300]      hv i+2 LZ  ;
[ 301]      hs 14b1NZ  ; print(p1)
[ 302]      ,tl [VM]   ; test of -half-word
[ 303]a14:  hv i+2     ;
[ 304]      hs 1b4     ; print(h)
[ 305]      tk 42      ;
[ 306]      ar a1      ; test of RH half-word
[ 307]      ps 42      ;
[ 308]      ,ud        ; basic op
[ 309]      qq         ; dummy
[ 310]      ps  0      ;
[ 311]      hv i+2 LZ  ;
[ 312]      hs 2b2 NZ  ; print(g2)
[ 313]a15:  ps -1      ; RH adress
[ 314]      ,ud -2     ;
[ 315]      sr a1      ;
[ 316]      ps  0      ;
[ 317]      hv i+2 LZ  ;
[ 318]      hs 4b2 NZ  ; print(a2)
[ 319]      ar a1      ; n, RH
```

```
[ 320]      ,qqn       ;
[ 321]      hv i+2 LZ  ;
[ 322]      hs 6b2 NZ  ; print(r2)
[ 323]a16:  ,ca (1b)   ; i, RH
[ 324]      hv i+2     ;
[ 325]      hs 8b2     ; print(i2)
[ 326]      ,ar ra1    ; r, RH
[ 327]      sr a1      ;
[ 328]      hv i+2 LZ  ;
[ 329]      hs 10b2NZ  ; print(r2)
[ 330]      ps a1      ; s, RH
[ 331]      ,ar  s     ;
[ 332]      sr a1      ;
[ 333]a17:  ps  0      ;
[ 334]      hv i+2 LZ  ;
[ 335]      hs 12b2NZ  ; print(s2)
[ 336]      pp a1      ; p, RH
[ 337]      ,ar  p     ;
[ 338]      sr a1      ;
[ 339]      pp  0      ;
[ 340]      hv i+2 LZ  ;
[ 341]      hs 14b2NZ  ; print(p2)
[ 342]      ,t1  [WM]  ; test of -half-word mark
[ 343]a18:  hv i+2     ;
[ 344]      hs 1b4     ; print(h).
[ 345]      pa i+1     ; test of increment
[ 346]      qq  ,qq 1  ;
[ 347]      ar i-1     ;
[ 348]      ca  0      ;
[ 349]      hv i+2     ;
[ 350]      hs b5      ; print(q)
[ 351]      tk 42      ;
[ 352]      pa i+1     ; test of -,B∧q
[ 353]a19:  ar ta1 NZ  ;
[ 354]      hv i+3 LZ  ;
[ 355]      ud b5  NZ  ; print(qB)
[ 356]      hs 2b3 NZ  ;
[ 357]      ar a1      ; test of q
[ 358]      pa i+1     ;
[ 359]      sr  ta1    ;
[ 360]      hv i+2 LZ  ;
[ 361]      hs b5  NZ  ; print(q)
[ 362]      qq tb4 XWM; [qq  ,hs b4]
[ 363]a20:  qq         ; test of +half-word mark, print(H)
[ 364]      tk 42      ;
[ 365]      ca (1b)    ; test of i, full-word
[ 366]      hv i+2     ;
[ 367]      hs 8b      ; print(i0)
[ 368]      tk 42      ; test of increment
[ 369]      ga i+12    ; decoding of TD≠0
[ 370]      ar a1      ;
[ 371]      qq(i+10)t512
[ 372]      qq(i+9)t256
[ 373]      qq(i+8)t128
[ 374]a21:  qq(i+7)t64;
[ 375]      qq(i+6)t32;
[ 376]      qq(i+5)t16;
[ 377]      qq(i+4)t8 ;
[ 378]      qq(i+3)t4 ;
[ 379]      qq(i+2)t2 ;
[ 380]      qq(i+1)t1 ;
[ 381]      ca -1      ;
[ 382]      hv i+2     ;
```

```
[ 383]      hs b5      ; print(q)
[ 384]a22:  ga i+1     ; decoding of
[ 385]      pi -1 t1   ; static op
[ 386]      tk 42      ;
[ 387]      ar i-2     ;
[ 388]      ca -1      ;
[ 389]      hv i+2     ;
[ 390]      hs 2b5     ; print(qs)
[ 391]      pi -500    ;
[ 392]      gi i+1     ; decoding of non-static op
[ 393]      bt-500t-100
[ 394]a23:  hv i+2     ;
[ 395]      hs 2b5     ; print(qs)
[ 396]      tk 42      ;
[ 397]      pa i+1     ; decoding of
[ 398]      nc 0 t-1   ; non-static op
[ 399]      hv i+2     ;
[ 400]      hs 2b5     ; print(qs)
[ 401]      qq tb4 XVM; test of +h
[ 402]      qq         ;
[ 403]      tk 42      ;
[ 404]      pa i+2     ; test of step 12
[ 405]      nt i+1     ; increment^r
[ 406]a24:  ca  r      ;
[ 407]      hv i+3     ;
[ 408]      ud 10b     ; print(rq)
[ 409]      hs b5      ;
[ 410]      pa i+3     ;
[ 411]      ps i+2     ;
[ 412]      nt i+1     ; increment^s
[ 413]      ca   s     ;
[ 414]      hv i+4     ;
[ 415]      ps  0      ;
[ 416]      ud 12b     ; print(sq)
[ 417]a25:  hs b5      ;
[ 418]      ps  0      ;
[ 419]      pa i+3     ;
[ 420]      pp i+2     ; increment^p
[ 421]      nt i+1     ;
[ 422]      ca  p      ;
[ 423]      hv i+3     ;
[ 424]      ud 14b     ; print(pq)
[ 425]      hs b5      ;
[ 426]      pp  0      ;
[ 427]      tk 42      ;
[ 428]      hv 1a1 NZ  ;
[ 429]a26:  ar a1      ;
[ 430]      ncn   LZ   ; test of -,B^n
[ 431]      hv i+3     ;
[ 432]      ud 6b      ; print(nB)
[ 433]      hs 2b3     ;
[ 434]      tk 42      ;
[ 435]      ar a1      ; test of -,B^(s)
[ 436]      ga i+1     ;
[ 437]      qq         ;
[ 438]      ps i-1     ;
[ 439]      nc (s) LZ  ;
[ 440]      hv i+5     ;
[ 441]a27:  ps  0      ;
[ 442]      ud 8b      ; print(isB)
[ 443]      ud 12b     ;
[ 444]      hs 2b3     ;
[ 445]      ga i+1     ; test of (r) and (s)
```

```
[ 446]      qq          ;
[ 447]      ps i-1      ;
[ 448]      ca (s)      ;
[ 449]      hv i+4      ;
[ 450]      ps  0       ;
[ 451]a28:  ud 8b       ; print(is0)
[ 452]      hs 12b      ;
[ 453]      ps  0       ;
[ 454]      ca (r-8)    ;
[ 455]      hv i+3      ;
[ 456]      ud 8b       ; print(ir0)
[ 457]      hs 10b      ;
[ 458]a29:  pp -1       ; test of IK (mode 4)
[ 459]      pi  0       ;
[ 460]      tk 42       ;
[ 461]      nc  0 IKC   ; p:=0, in:=-1
[ 462]      hs b4       ; if +h then print(H)
[ 463]      gp i+1      ;
[ 464]      ca          ;
[ 465]      hv i+2      ;
[ 466]a30:  hs 3b4      ; print(IK)
[ 467]      gi i+1      ;
[ 468]      qq          ;
[ 469]      nc  0 IKC   ; p:=-1, in:=0
[ 470]      hs b4       ; if +h then print(H)
[ 471]      ar i-3      ;
[ 472]      ca -1       ;
[ 473]      hv i+2      ;
[ 474]      hs 3b4      ; print(IK)
[ 475]      gp i+1      ;
[ 476]      ca          ;
[ 477]a31:  hv i+2      ;
[ 478]      hs 3b4      ; print(IK)
[ 479]      gi i+2      ;
[ 480]      tk 42       ;
[ 481]      ca          ;
[ 482]      hv i+2      ;
[ 483]      hs 3b4      ; print(IK)
[ 484]      qq    V     ; test of V
[ 485]      hs 2b6      ; print(V)
[ 486]      pp -1       ; test of VIK
[ 487]      pi  0 VIKC; p:=0, in:=-1
[ 488]      hs i+2      ; print(VIK)
[ 489]a32:  hv i+4      ;
[ 490]      ud 2b6      ;
[ 491]      ud 3b6      ;
[ 492]      hv 3b4      ;
[ 493]      tk 42       ;
[ 494]      gp i+1      ;
[ 495]      ca          ;
[ 496]      hv i+2      ;
[ 497]      hs i-7      ; print(VIK)
[ 498]      hv 1a1 NZ   ; skip if +h
[ 499]      pm a1  X    ; test of X
[ 500]      hs 8b5 LZ   ; print(X)
[ 501]      pm a1 XIKC; test of XIK
[ 502]      gp i+1      ; R:=-1, M:=0, p:=-1, in:=0
[ 503]a33:  ca          ;
[ 504]      hv i+4 NZ   ;
[ 505]      ud 8b5      ; print (XIK)
[ 506]      ud 9b5      ;
[ 507]      hs 3b4      ;
[ 508]c7:   qq 2000.39; Ok counter
```

```
[ 509]c8:   qq          ; work
[ 510]      tk 42       ;
[ 511]      ar i-2      ;
[ 512]      ar a1       ;
[ 513]a34:  gr i-4      ;
[ 514]      hv 1a1 NZ   ; if 2000 successful runs then
[ 515]      sy 62       ; print(ok)
[ 516]      sy 38       ;
[ 517]      sy 34       ;
[ 518]      sy 29       ;
[ 519]a35:  hv a3       ; goto start and reset
```

B HJ DRUMTEST 5

Programmet skriver paa tromlen fra første kanal til sidste kanal, *)
og sammenligner det skrevne med det læste (mærkebits testes).
Programmet er placeret i celle 41 - 270 , kanal 0 i 271 - 310
og 2 lageromraader i 512 - 551 og 768 - 807.
Ved første indhop til celle 41 læses kanal 0, tælleren og indikatoren
nulstilles (ved senere indhop undlades dette).
Programmet stopper (zq) og nu kan man vælge første og sidste kanal,
idet de indsættes i R-reg adressedel og tælledel. 50 og 319 er valgt
ved indlæsningen.
Ønsker man at retablere den gamle kanal 0, hoppes til celle 256,
checksummen testes og hvis den er ok skrives kanal 0 paa tromlen,
første og sidste kanal sættes til 1 og 319, paa by bit 5 skrives : nul
og der hoppes til stop i programstarten (husk at laase kanal 0).
Indikatorens sidste 6 bit bruges saaledes:

KA = 0	efter endt læsning læses igen.
= 1	efter endt læsning skrives paany.
KB = 0	simultankørsel (tromletransport og testning samtidig).
= 1	ingen simultankørsel.
RA = 0	kanal og cellenr udskrives ved fejl.
= 1	kanal og cellenr udskrives ikke ved fejl.
RB = 0	bitmønster udskrives ved fejl.
= 1	bitmønster udskrives ikke ved fejl.

Mærkebits gemmes i PA og PB.
Efter endt læsning skrives ok og antal fejl hvis ≠ 0 over by bit 4.

Eksempel paa fejludskrift:

```
f  kanal   200  celle    24
s 0000110001 0101010111 1111111111 0000010101 01      (skrevet celle)
l 0000110001 0101010111 0000000000 0000011111 11      ( læst   celle)

ok  120
```

*) den skrevne information er: n.ord = n.19 + n.39 (incl. markebits)

```
b i=41,a100,b100
grm  a6                ;  tæller:=0
     vk   0
     lk   a20                    ;   a20:=kanal 0
     vk   0
     pm   b34
     gm    41                    ;  [41]:=  hv   b41
     hs   b31                    ;   R:=sum af kanal 0
     gr   a19                    ;   a19:=R
b41:
     pi     0                    ;   nulstil indikatoren
     arm   a1                    ;   R:=føste kanal.9+sidste kanal.19
     zq                          ;   stop
     gr    a1                    ;   gem R
b10: pa   a3      t 512          ;skrivning: plads:=512
     arm  a1                    ;   hent første og sidste kanal
     ga   a2                    ;   tr:=første kanal
     gt   b14                   ;   sidste kanal i b14
     gt   b16                   ;     -      -   - b16
     ar   a4
     gt   b15                   ;   sidste kanal + 1 i b15
     pm    a6
     gm    a0                   ;   gem tæller
b0:  pa   b1      t -39          ;skriv40:
     it   (a3)    ,pa   b2      ;   p:=plads
b1:  bt           t 1           ;skrivcelle:
     hv   b3                    ;   hop til skrivkanal
     arm   a6                   ;   dan ny tæller
     ar    a4                   ;   + step
     gr    a6
b2:  gr   [p]
     mb   a7                    ;   vælg mærke (= bit 38-39)
     ar   a8
     gr   r1      M
     qq [acn   (b2)   M mærke ]
     qq   (b2)    t 1           ;   p:=p+1
     hv   b1                    ;   hop til skrivcelle
b3:  vk   (a2)                  ;skrivkanal: vælg kanal tr
     sk   (a3)                  ;   skriv
     vk   (a2)    LKB           ;   if kbon then vent
     qq   (a2)    t 1           ;   tr:=tr+1
14:  bs   (a2)    t 319         ;   if tr>sidste kanal then
     hv   b4                    ;   hop til læsning
     arm  a3                    ;   skift plads(512,768)
     ca   512     ,it  768
     pa   a3      t    512
     hv   b0                    ;   hop til skriv40
     [
        arbejdsceller           ]
a0:  qq                         ; gem tællerens startværdi
a1:  qq   50      t 319         ;  første kanal.9  sidste kanal.19
a2:  qq                         ; tr       aktuel kanal
a3:  qq                         ; plads    aktuel lagerplads
a4:  qq         1.19  + 1.39    ; step
a5:  qq                         ; antal fejl pr læsning
a6:  qq                         ; tæller
a7:              3              ; bit 38-39
a8:  acn  (b2)        M         ;  (b2) mærkes
a9:  qq                         ; skrevet mærkebit i .39
a14: acn   a6     M             ; a6 mærkes
     [    læsning                 ]
b4:  vk   (a1)                  ;læsning
     grm  a5                    ;   antal fejl:=0
     sy   58
     pm   a0                    ;   retabler start i a6
     gm   a6
```

```
        it  (a1)    ,pa   a2      ;   tr:=første kanal
        vk  (a2)                  ;   vælg kanal tr
        lk  (a3)                  ;   læs til plads
        pa   a3     t 768         ;   plads:=768
        qq  (a2)    t 1           ;   tr:=tr+1
9:      vk  (a2)                  ;test40: vælg kanal tr
        lk  (a3)                  ;   læs til plads
b12:    vk  (a2)    LKB           ;byt:  if kbon then vent
        arn  a3                   ;   skift plads(512,768)
        ca   512    ,it 768
        pa   a3     t   512
        it   (a3)   ,pa   b6      ;   p:=plads
        pa   b5     t -39         ;   tæl til 40
b5:     bt          t 1           ;test1: tæl
        hv    b7                  ;   hop til sluttest
        arn   a4                  ;   tæller+step
        ac    a6
        arn   a6
b6:     sr   [p]
        hv   b11    NZ            ;   if R≠0 then goto fejl
122     arn   a6                  ;   test mærkebit
        mb    a7
        gr    a9
        pmn    (b6)               ;   R40-41:=læst mærkning
        ar 1        DLB
        ar 2        DLA
        ck    10                  ;   læst mærkning -> .39
        sr    a9                  ;   - skrevet mærkning
        hv   b11    NZ            ;   if R≠0 then goto fejl
        qq   (b6)   t 1           ;   p:=p+1
        hv    b5                  ;   hop til test1
b11:    vy   32                   ;fejl :  skriv(f)
        sy   54                   ;   vælg bit 4 til udskrift
        hs   b25    NRA           ;   if RA=0 then skriv kanalnr og cellenr
        hv   b13    LRB           ;   if RB=1 then goto fejl slut
        sy   64
        arn  a6
        mb   a7                   ;   dan skrevet mærkning
        ar   a14
        gr   r1     M             ;  skrevet mærkning til a6
        qq   [acn  a6   M mærke  ]
        sy   18                   ,sy    0    ;   skriv(s)
        arn  a6     IPC           ;   R:=skrevet celle
        hs   b20                  ;   tryk(bitmønster)
        sy   64     ,sy   35      ;   skriv(1 )
        sy    0
        arn  (b6)   IPC           ;   R:=fejllæst celle
        hs   b20
b13:    arn  a11                  ;   1
        ac   a5     ,sy   64              ;   antalfejl:=antalfejl+1
        qq   (b6)   t 1           ;   p:=p+1
        hv    b5                  ;   hop til test1
b7:     qq  (a2)    t 1           ;   tr:=tr+1
b15:    bs  (a2)    t 320         ;   if tr>sidste kanal+1 then goto finis
        hv   b8
b16:    bs  (a2)    t 319         ;   if tr>sidste kanal then goto byt
          hv   b12
          hv   b9                 ;   hop til test 40
 b8:      vy   32                 ;finis:  skriv(ok)
          sy   38     ,sy   34
          arn  a5                 ;   R:=antal fejl
          hs   b26    NZ          ;   if R≠0 then  tryk(antal fejl)
          sy   64
          hv   b10    LKA         ;   if kaon then goto skrivning
          hv   b4                 ;            else goto læsning
          [
          udskrift af r-reg vha binær form]
```

```
        tl   -10                ;   flyt væk fra Radr
b21:    pa   b23    t 9         ;ti: klar til at tælle til 10
b22:    mb   a13                ;næst: Radr:=0
        tl   1                  ;   næste bit hentes
        ca   0                  ;   if Radr=0
        sy   16     V           ;   then skriv(0)
        sy   1                  ;   else skriv(1)
b23:    bt   9      t -1
        hv   b22
        sy   0                  ;   space
b24:    bt   3      t -1        ;tæl4:
        hv   b21
        sy   1      V LPA       ;   bit 40
        sy   16
        sy   1      V LPB       ;   bit 41
        sy   16
        sy   0
        hr   s 1                ;   hop retur
a13:    0/1023/1023/1023
        [
        udskrift af kanalnr og cellenr]
b25:    sy   0      ,sy  0
        sy   34     ,sy  49     ;   skriv(  kanal )
        sy   37     ,sy  49
        sy   35     ,sy  0
        arn  (a2)   D
        sr    1     D
        tl   -30                ;   R:=kanalnr
        hs   b26                ;   tryk(kanalnr)
        sy   0      ,sy  0
        sy   51     ,sy  53     ;   skriv(  celle )
        sy   35     ,sy  35
        sy   53     ,sy  0
        arn  (b6)   D           ;   R:=p
        sr   (a3)   D           ;     -plads
        tl   -30
        hs   b26                ;   tryk(cellenr)
        hr   s 1                ;  hop retur
        [udskrift af r-reg paa decimal form]
b26:    gs   b30    V NZ        ;tryk: if R#0 then gem s og tryk
        sy   16     ,hr  s1     ;   else skriv(0) , hop retur
        pa   b29    t 4         ;   trim til 5 cifre
        ps    0                 ;   s:=space
        dk   a12                ;   :100000
        ar   a11    X           ;   evt forhøjelse
b27:    mln  a10                ;næst: hent næste ciffer
        tk   30     ,ga  b28    ;   gem ciffer
        sy    s     V LZ        ;   if R=0 then skriv(s)
b28:    sy [ciffer]             ,ps  16   ;   else skriv(ciffer) , s:=0
b29:    bt   4      t -1        ;  tæl til 5
        hv   b27                ;   hop til næst
b30:    ps   [gl s]                 ;   retabler s
        hr   s 1                ;   hop retur
a10:    10
a11:    1
a12:    100000
b31:    pa   b33    t  a20
        pan   r1    t -39
b32:    bt          t  1
        hr   s 1
b33:    ar   a20
        qq   (b33)  t  1
        hv   b32
i=256
b40:    hs   b31
        sr   a19
        hv   b41
```

```
        vx
        vy    32
        sy    37
        sy    20
        sy    35
        pm    b35
        gm    a1
b34:    hv    b41
b35:    qq    1     t  319
a19:    qq                      ;   checksum af kanal 0
a20:    qq
s
```

PEM

```
'BUFFE TEST 2 - SKRIVER TILFÆLDIGE TAL I ALLE BUFFERCELLER, LÆSER DISSE IGEN
 OG SAMMENLIGNER DET LÆSTE MED DET SKREVNE.  PÅ SKRIVEMASKINEN TASTES TRE
 TAL (SKILLETEGN CR) EFTER UDSKRIFT AF KLAR. DISSE TAL ANVENDES TIL DANNELSE
 AF DE TILFÆLDIGE TAL, DER BRUGES TIL TEST.  MÆRKEBITS TESTES.  PROGRAMMET
 STARTER I CELLE 41    ZQ  LKB.   FEJLUDSKRIFTER SKER OVER BY-8.
 IPROGRAMMET ER FORSKELLIGE TRIMMEMULIGHEDER:

     KA := 1     TEST MED SAMME TAL

     KB := 1      TEST MED ANDRE TAL

     KC := 1       TEST MED NYLÆSTE TAL )

I=41


                                                  ;STYRESEKVENS
[  41] ZQ                      LKB
[  42] HS    86                                   ;LÆS 3 TAL
[  43] HS    198                                  ;GEM INDLÆSTE TAL
[  44] HS    155                                  ;OVERFØR INDLÆSTE TAL
   45] HS    202                                  ;OVERFØR INDLÆSTE TAL
[  46] HS    198                                  ;OVERFØR INDLÆSTE TAL
   47] GRN   74
[  48] HS    170                                  ;BUFFER LÆSES TILBAG
[  49] VY    72              ,SY   64
[  50] ARN   74                                   ;ANTAL FEJL PR. GENNEMLØB
[  51] HS    214
[  52] HV    42                LKC               ;INDLÆS 3 NYE TAL
[  53] HV    43                LKA               ;KØR OM MED SAMME TAL
[  54] HS    210               LKB               ;KØR OM MED ANDRE TAL
[  55] HSF   2               ,HVF  42
[  56] HV    43
                                                 ;ARBEJDSCELLER
[  57] QQ
[  58] QQ
[  59] QQ
[  60] QQ
[  61] QQ
   62] QQ
   63] QQ                                         ;FL.ADR. FOR AKTUELLE CELLE I POS 39
[  64] QQ                                         ;BLOKNUMMER I BUFFER (VÆRDI 0-7 I PO
   65] QQ                                         ;TILFÆLDIGT TESTTAL
[  66] QQ    500                    T-512
[  67] QQ                  X
[  68] QQ
[  69] UDN (P)             XVDLRC T-1
[  70] QQ                     IB
[  71] QQ
[  72] QQ
[  73] QQ
[  74] QQ                                         ;ANTAL FEJL PR. BUFFERGENNEMLØB
                                                  ;DAN TILFÆLDIGT TAL
[  75] PM    62
[  76] ARN   62            X
[  77] MK    60
[  78] SC    61
[  79] DL    62            X
[  80] AC    60               IOA
   81] ML    61
[  82] AC    62               IOB
   83] HR    S1
                                                  ;LÆS 3 TAL FRA SKRIVEMASKINEN
[  84] QQ
```

```
[  85] QQ                    IZA
   86] PA    103                    T56
   87] PA    104                    T2
[  88] VY    17          ,SY    64
   89] SY    34          ,SY    35
   90] SY    49          ,SY    41          ;KLAR
[  91] SY    64
[  92] PT    99                     T103
[  93] XRN               ,LYN   84
[  94] NC    32          ,HV    97          ;MINUS
[  95] PT    99                     T102
[  96] HH    93
[  97] CA    16          ,HVN   100
[  98] CA    64
[  99] XR                ,HV
[ 100] TK    -30         ,ML    85
[ 101] HH    93
[ 102] MT    -1          D
[ 103] GR    56                     T1
[ 104] BT    2                      T-1
   105] HV    92
   106] HR    S1

                                                ;SEKVENS TIL UDSKRIFT AF ROG 40-41.
   107] PA    118                    T3
[ 108] TL    -10
[ 109] PA    115                    T9
[ 110] MB    69                     ;R <-9 RENSES
[ 111] TL    1                      ;1 BIT HENTES FREM
[ 112] CA
[ 113] SY    16          V
[ 114] SY    1                       ;UDSKRIV BIT
[ 115] BT    9                      T-1
[ 116] HV    110
[ 117] SY    23                     ;UDSKRIV SKILLETEGN
[ 118] BT    3                      T-1
[ 119] HV    109
[ 120] SY    1           V  LPA     ;SKRIV BIT 40
[ 121] SY    16
[ 122] SY    1           V  LPB     ;SKRIV BIT 41
   123] SY    16
   124] HR    S1

                                                ;SEKVENS TIL UDSKRIFT AF BUFFERADR.
   125] SY                ,SY
[ 126] SY    50          ,SY    54
[ 127] SY                ,SY    49
[ 128] SY    52          ,SY    41
[ 129] SY
[ 130] PA    141                    T3
[ 131] TK    -11         X          ;BLOKNR. TIL M POS. 20
[ 132] MKN   66
[ 133] AR    63                     ;ADDER FL.ADR
[ 134] TK    18                     ;BUFFERADR. I 10-22
[ 135] MB    69                     ;RENS 0-9
[ 136] TK    3                      ;HENT CIFFER
[ 137] GA    R3 [140]
[ 138] CA
[ 139] SY    16          V
[ 140] SY
[ 141] BT    3                      T-1
   142] HV    135
[ 143] HR    S1

                                                ;FEJLUDSKRIFTSSEKVENS
[ 144] ARN   70          ,AC    74
[ 145] ARN   65             IPC                 ;TESTTAL TIL R
```

```
[ 146] VY   8          ,SY   64
  147] SY   64
  148] HS   107                      ;TRYK TESTTAL BINÆRT
[ 149] SY   64
  150] ARN  (179)          IPC
[ 151] HS   107                      ;TRYK FORKERTE TAL BINÆRT
[ 152] ARN  64
[ 153] HS   125                    ;TRYK BUFFERADRESSEN
                                     ;FYLD BUFFER
[ 154] HV   189
[ 155] PA   166              T7      ;RETABLER STARTADRESSE
[ 156] ARN  66         ,GR   169
[ 157] PA   160              T499
[ 158] PA   161              T511
[ 159] HS   75                       ;DAN TILFÆLDIGT TAL
[ 160] GR   499          MOC T1
[ 161] BT   511              T-1
[ 162] HV   159
[ 163] ARN  169
[ 164] US                            ;512 ORD AD GAN GEN TIL BUFFER
  165] ARN  67         ,AC   169     ;TÆL STARTPARAMETER OP
  166] BT   7                T-1
[ 167] HV   157
  168] HR   S1
[ 169] QQ
                                     ;LÆS BUFFER TILBAGE OG KONTROLLER
[ 170] GRN  64                       ;BLOKNR. := 0
[ 171] ARN  66         ,GR   197
[ 172] PA   194              T7
[ 173] ARN  197
[ 174] IL              ,GRN  63      ;HENT 512 ORD FRA BUFFER
[ 175] PA   190              T511
[ 176] PA   179              T499
[ 177] HS   75                       ;DAN TILFÆLDIGT TAL
[ 178] GR   65           MOC         ;GEM TESTTAL
[ 179] SR   499              T1      ;R := TESTTAL - TILBAGELÆST TAL
[ 180] HV   144          NZ          ;HOP UD HVIS R 0-39 FORKERT
[ 181] HV   184          LOB
[ 182] HV   144          LB          ; -  -  -    B     -
  183] HV   185
  184] HV   144          NB          ; -  -  -    B     -
[ 185] HV   188          LOA
  186] HV   144          LA          ; -  -  -    A     -a
[ 187] HV   189
[ 188] HV   144          NA          ; -  -  -    A     -
[ 189] ARN  70         ,AC   63      ;ORDNR. := ORDNR + 1
[ 190] BT   511              T-1
[ 191] HV   177
[ 192] QQ   (64)             T1
[ 193] ARN  67         ,AC   197     ;BLOKNR. := BLOKNR. + 1
[ 194] BT   7                T-1     ;BUFFERADR. := BUFFERADR. + 1
[ 195] HV   173
[ 196] HR   S1
[ 197] QQ
                                     ;OVERFØRSLER AF ARBEJDSCELLER
[ 198] ARN  57         ,GR   60
[ 199] ARN  58         ,GR   61
[ 200] ARN  59         ,GR   62
  201] HR   S1
  202] ARN  60         ,GR   71
[ 203] ARN  61         ,GR   72
  204] ARN  62         ,GR   73
[ 205] HR   S1
[ 206] ARN  71         ,GR   60
```

```
 207] ARN   72          ,GR   61
 208] ARN   73          ,GR   62
 209] HR    S1
 210] ARN   71          ,GR   57
 211] ARN   72          ,GR   58
 212] ARN   73          ,GR   59
 213] HR    S2

 214] PA    221                T3          ;TRYKNING AF ANTAL FEJL
 215] DK    226
 216] AR    225         X
 217] MLN   224
 218] TK    30          ,GA   220
 219] SY    16          V  LZ
 220] SY
 221] BT    3                  T-1
 222] HV    217
 223] HR    S1
 224] QQ                   IZA
 225] QQ                   IB
 226] QQN   S           XV IT
```

```
[Testprogram til buffer
 PVV  9/2-67

 Testord: bit 0 - 27:  0101...01 eller 1010...10
          bit 28 - 39: bufferadresse
 Fejludskrift:
     1. linie:  Bufferadressen oktalt, det afsendte (korrekte) bitmønster
     2. linie:  Det modtagne (fejlramte) bitmønster fra bufferen

 Programmet skriver testord i bufferen med given afstand (63 cellers afstand,
 slipnavnet d), hvorefter resten af bufferen fyldes med vilkaarligt indhold
 et vist antal gange, bestemt af KA og KB:
    KA=0, KB=0    1 gang
    KA=L, KB=0    5 gange (slipnavn d2)
    KA=0, KB=L   10 gange (slipnavn d3)
    KA=L, KB=L   30 gange (slipnavn d4)
 hvorefter testordene i bufferen sammenlignes med de afsendte ord.  Mærkebits
 testes ogsaa.

 Programmet startes i celle 41.]

b i=10, a15, b19, d5

d d=63                                  ;spring i bufferadresse mellem
                                        ;testceller
d d2=5, d3=10, d4=30                    ;antal buffergennemløb
                                        ;mellem test ved LKA,LKB og LKC
d b3=41                                 ;programstart
s                                       ; redefiner evt. d, d2, d3, d4 eller b3

d d1=d-1, d5=d4-d3

i=b3.
41      vy m17                          ;programstart:
        gm a                            ;startadr:=0
        pa b

b7:     pa b1                           ;dan testord: s:=0
        pa b15
        pp m0
        pm a
        gm a1                           ;lbadr:=startadr; p:=0

b1:     ps m0           t 1             ;igen: s:=s+1
        pm a4       X                   ;testord:=integer
b15:    bt m0           t 512           ;   (alternate bits shift
        tk m1                           ;     (if s:2x2=s then 1 else 0)
        ab a                            ;    V boolean lb adr)
        pp p1                       IK  ; p:=p+1
        gr a3                       MRC ; testmarks:=p mod 4
        gr sa7                      MRC ;buf[s]:=testord; marks[s]:=testmarks
        pm a1       X               IK  ;if lbadr >
        sr a6                           ;   maxadr then
        hv b                        NT  ;  go to test adr
        pm a1       X
        ar a2
        uc m0                           ;us(testord)
        pm a5       X                   ;lbadr:=lbadr+spring
        ac a1
        hv b1                           ;go to igen
```

```
b:      pp m0          t 1          ;test adr:
        pm p         X D            ;startadr:=startadr+1
        ck m10
        gr a
        bs p-d1                     ;if startadr > d1
        hv b3                       ;  then go to programstart

        grn a10                     ;tælle:=0

b6:     pm a                        ;til BF:
        gm a1                       ;lbadr:=startadr
b14:    pm a1          X            ;udfyld: if lbadr
        sr a6                       ;    ≥ maxadr
        ar   a5                     ;      - spring
        hv b4                  NT   ;    then go to tæl
        pm a1          X
        ar a9
        us m0                       ;us (udfyldning)
        pm a5          X            ;lbadr:=lbadr+spring
        ac a1
        hv b14                      ;go to udfyld

b4:     pm a10         X            ;tæl:
        ar a7
        gr a10                      ;tælle:=tælle+1
        sr a7           V      NKC  ;if tælle<(if KA=0∧KB=0 then 1
        sr a11          V      LKC  ;         else if KA=1∧KB=1 then d4
        sr a12          V      LKA  ;         else if KA=1 then d2 else d3)
        sr a13                 LKB
        hv b6                  LT   ;    then go to til BF

        pm a           X
        sr a7
        gr a1                       ;lbadr:=startadr-1
        pa b8                       ;s:=0

b8:     ps m0            t 1        ;test: s:=s+1
        pm a1          X            ;if lbadr
        sr a6                       ;    > maxadr
        hv b7                  NT   ;    then go to dan testord
        pm a1          X
        ar a2
        il m0                       ;il (testord)
        pm a3          X       IRC
        sr sa7                 IZA
        qqn                    IPC
        ar   m1          D     LRB
        ar   m2            D     LRA
        sr   m1          D     LPB
        sr   m2          D     LPA
        pm a1          X       IZB
        hv b9                  LZC  ;if testord ≠ buf[s] ∨
```

```
                sy  m64                          ; testmarks≠marks[s] then begin writecr
                pp  m4                           ;
                tl  m-12
b11:            pp  p-1                          ;for p:=3 step -1 until 0 do
                tln m3                           ;begin
                ck  m-10                         ;   b10:=integer(((boolean lbadr)
                ga  b10                          ;            shift(-3xp)) ∧ 40 7)
b10:            sy  m0          V          NZ    ;  writechar(if b10≠0 then b10 else 16)
                sy  m16
                bs  p
                hvn b11                          ;end

                pm  sa7         X          IRC   ;boo:=boolean buf[s]
                pa   b2             t -30
                sy  m0                           ;writechar(0)
                pp  m40                          ;for p:=40 step-1 until 1 do
b12:            ck  m0                           ;begin
                sy  m1          V          LO    ;  writechar(if boo then 1 else 16)
                sy  m16
                ck  m1                           ;  boo:=boo shift 1
                pp  p-1         X
b2:             can  p-30                        ;  if (p-1):10X10=p-1
                sy   m0        ,it   m10         ;    then writechar(0)
b5:             qq  (b2)
                bs  p
                hv  b12         X                ; end

                sy   m1         V          LRA   ; writechar(if marks[s]:2=1 then 1
                sy   m16                         ;                 else 16)
                sy   m1         V          LRB   ; writechar(if marks[s] mod 2=1 then 1
                sy   m16                         ;                 else 16)

                sy  m64                          ;writecr
                pp  m5                           ;for p:=5 step-1 until 1 do
b16:            sy  m0                           ;writechar(0)
                pp  p-1
                bs  p
                hv  b16

                pm  a3          X          IRC   ;boo:=boolean testord
                pa   b18            t -30
                pp  m40                          ;for p:=40 step-1 until 1 do
b13:            ck  m0                           ;begin
                sy  m1          V          LO    ;  writechar(if boo then 1 else 16)
                sy  m16
                ck  m1                           ;  boo:=boo shift 1
                pp  p-1         X
b18:            can  p-30                        ; if (p-1):10X10 = p-1
                sy   m0        ,it   m10         ;   then writechar(0)
                qq  (b18)
                bs  p
                hv  b13         X                ;end

                sy   m1         V          LRA   ; writechar(if testmarks:2=1 then 1
                sy   m16                         ;                 else 16)
                sy   m1         V          LRB   ; writechar(if testmarks mod 2=1 then 1
                sy   m16                         ;                 else 16)
                                                 ; end
```

```
b9:     pm a5          X
        ac a1          ~~~~
        hv b8                          ;lbadr:=lbadr+spring
                                       ;go to test

a:      0                              ;startadr
a5:     qq md.39                       ;spring
a1:     0                              ;lbadr
a2:     qq ma3         ,qq m1          ;stamparameter testord
a9:     q ma8          ,qq md1         ;stamparameter udfyldning
a6:     4096                           ;maxadr
a10:    0                              ;tælle
a12:    qq md2.39                      ;maxtælle LKA
a13:    qq md3.39                      ;maxtælle LKB
a11:    qq md5.39                      ;maxtælle LKC-maxtælle LKB
a4:     341/341/340/0                  ;alternate bits
a3:     0                              ;testord
a7:     1
a8=1a7                                 ;buf[1:(4096-startadr):d+1]

e
s
```

P EM

MB-TEST 1 - SKRIVER TILFÆLDIGE TAL PAA FORUDVALGT BAANDSTATIONSNUMMER, LÆSER DISSE TAL IGEN. OG SAMMENLIGNER DET LÆSTE MED DET SKREVNE.
PAA SKRIVEMASKINEN TASTES BAANDSTATIONSNUMMER EFTER UDSKRIFT AF ST.NR., OG TRE TAL (SKILLETEGN CR) EFTER UDSKRIFT AF KLAR . DE TRE TAL ANVENDES TIL DANNELSE AF TILFÆLDIGE TAL, DER BRUGES TIL MB-TEST! MÆRKEBITS TESTES.
I PROGREMMET FINDES FORSKELLIGE TRIMMEMULIGHEDER:

KA := 1 TEST MED SAMME TAL.

KB := 0 TEST MED ANDRE TAL.
1 TEST MED NYLÆSTE TAL.

KC := 0 TESTER BAAND VED NORMAL SKRIVNING AF BLOK OG EFTERFØLG LÆSNING.
1 DER FORSØGES NORMAL SKRIVNING TRE GANGE, HEREFTER FORS SKRIVNING MED OVERSPRINGNING.
HVIS SKRIVNING UMULIG UDHOP HSF 2 312, ELLERS FORSØ LÆSNING 6 GANGE, OG HVIS DETTE ER UMULIGT UDHOP HSF 2

RB := 0 VED FEJL UDSKRIFT AF DET SKREVNE OG DET LÆSTE BITMØNST SAMT BUFFERADRESSE.
1 INGEN UDSKRIFT VED FEJL.

P-REGISTRET INDEHOLDER AKTUELT BAANDSTATIONSNUMMER.

PROGRAMMET STARTER MED ORDREN ZQ LKB I CELLE 41, OG STOP AF PROGRAMMET SKER VED MANUELT AT STOPPE GIER)

I=41

```
                                                    ;STYPESEKVENS
[  41] ZQ              LKB
[  42] VY    17
[  43] SY    64      ,SY   18
[  44] SY    19      ,SY   59
[  45] SY    37      ,SY   41
   46] SY    59
[  47] PI    1         T-2
[  48] LYN   86      ,GA   R1 [49]          ;LÆS BAANDSTATIONSNUMMER
   49] PP                                   ;P:=ST.NR.
[  50] HS    98                             ;LÆS TRE TAL
[  51] HS    235                            ;OVERFØR TAL
[  52] HS    173                            ;FYLD BUFFER
[  53] HS    239                            ;OVERFØR TAL
[  54] HS    235                            ;OVERFØR TAL
[  55] GRN   83
[  56] HS    282       LKC                  ;SKRIV BLOK BINÆRT,OVERSPRING
[  57] HS    265                            ;SKRIV OG LÆS BLOK
[  58] HS    202                            ;LÆS BUFFER TILBAGE OG KONTRO
[  59] SY    64
[  60] ARN   83
[  61] HS    251                            ;SKRIV ANTAL FEJL PR. GENNEML
[  62] HV    50        NKC                  ;INDLÆS 3 NYE TAL
[  63] HV    51        LKA                  ;KØR OM MED SAMME TAL
[  64] HV    247       LKB                        ;OVERFØR TAL
   65] ZQ

                                             ;ARBEJDSCELLER
   66] QQ
   67] QQ
[  68] QQ
[  69] QQ
```

```
[  70] QQ
   71] QQ
   72] QQ                                          ;FL-ADR. FOR AKTF  CELLE I PO
   73] QQ                                          ;BLOKNR. I BUFFER (VÆRDI 0-7
   74] QQ                                          ;TILFÆLDIGT TESTTAL
[  75] QQ    500                  T-512
[  76] QQ                 VD LRC
[  77] QQ
[  78] UDN (P)            XVDLRC T-1
[  79] QQ                    IB
[  80] QQ
[  81] QQ
[  82] QQ
[  83] QQ                                        ;ANTAL FEJL PR. BUFFERGENNEML
[  84] QQ    3               IB  T-1
[  85] UDN (P-8)          XVDLRC T-1
[  86] QQ
                                                 ;DAN TILFÆLDIGT TAL
[  87] PM    71
[  88] ARN   71           X
   89] MK    69
   90] SC    70
   91] DL    71           X
   92] AC    69              IOA
[  93] ML    70
[  94] AC    71              IOB
[  95] HR    S1
                                                 ;LÆS TRE TAL FRA SKRIVEMASKIN
[  96] QQ
[  97] QQ                    IZA
[  98] PA    115                 T65
[  99] PA    116                 T2
[ 100] SY    64
[ 101] SY    34           ,SY   35
[ 102] SY    49           ,SY   41
[ 103] SY    64                                  ;KLAR
[ 104] PT    111                 T115
[ 105] XRN                ,LYN  96
[ 106] NC    32           ,HV   109              ;MINUS
   107] PT    111                 T114
   108] HH    105
   109] CA    16           ,HVN  112
   110] CA    64
[ 111] XR                 ,HV
[ 112] TK    -30          ,ML   97
[ 113] HH    105
[ 114] MT    -1           D
[ 115] GR    65                  T1
[ 116] BT    2                   T-1
[ 117] HV    104
[ 118] HR    S1
                                                 ;BINÆR UDSKRIFT AF R OG 40-41
[ 119] PA    131                 T5
[ 120] TL    -10
[ 121] PA    127                 T5
[ 122] MB    78                                  ;R 0-9 RENSES
[ 123] TL    1
[ 124] CA
   125] SY    16           V                      ;UDSKRIFT AF BIT
   126] SY    1
   127] BT    5                   T-1
[ 128] HV    122
[ 129] HV    130
[ 130] SY                                        ;UDSKRIV SKILLETEGN
```

```
[ 131] BT   5                    T-1
[ 132] HV   121
[ 133] PA   127                  T3
[ 134] PA   129                  T136
[ 135] HV   122
[ 136] SY   1            V  LPA                  ;UDSKRIV BIT 40
[ 137] SY   16
[ 138] SY   1            V  LPB                  ;UDSKRIV BIT 41
[ 139] SY   16
[ 140] PA   129                  T130
[ 141] HR   S1

                                                 ;UDSKRIFT AF BUFFERADR. 4 CI
[ 142] SY                ,SY
[ 143] SY   50           ,SY   54
[ 144] SY                ,SY   49
[ 145] SY   52           ,SY   41
[ 146] SY
[ 147] PA   158                  T3
[ 148] TK   -11          X                      ;BLOKNUMMER TIL M POS. 20
[ 149] MKN  75
[ 150] AR   72                                  ;ADDER FL-ADR.
[ 151] TK   18                                  ;BUFFERADR
[ 152] MB   78                                  ;RETTES 0-9
[ 153] TK   3                                   ;HENT CIFFER
[ 154] GA   R3 (157)
[ 155] CA
[ 156] SY   16           V
[ 157] SY
[ 158] BT   3                    T-1
[ 159] HV   152
[ 160] HR   S1

                                                 ;FEJLUDSKRIFTSSEKVENS
[ 161] ARN  79           ,AC   83
[ 162] HV   226             LRB                  ;EJ UDSKRIFT HVIS RB = 1
[ 163] ARN  74              IPC                  ;TESTTAL TIL R
[ 164] VY   32           ,SY   64
[ 165] SY   64
[ 166] HS   119                                  ;TRYK TESTTAL BINÆRT
[ 167] SY   64
[ 168] ARN (216)            IPC
[ 169] HS   119                                  ;TRYK FORKERTE TAL BINÆRT
[ 170] ARN  73
[ 171] HS   142                                  ;TRYK BUFFERADRESSE
[ 172] HV   226             IK
                                                 ;FYLD BUFFER
[ 173] PA   184 T7
[ 174] ARN  75           ,GR   187               ;STARTPARAMETER
[ 175] PA   178                 T499
[ 176] PA   179                 T510
[ 177] HS   87                                   ;DAN TILFÆLDIGT TAL
[ 178] GR   499             MOC T1
[ 179] BT   510                 T-1
[ 180] HV   177
[ 181] ARN  187
[ 182] US                                        ;500 ORD TIL BUFFER
[ 183] ARN  76           ,AC   187               ;OPTÆL PARAMETER
[ 184] BT   7                   T-1
[ 185] HV   175
[ 186] HR   S1
[ 187] QQ

                                                 ;NULSTIL BUFFER
[ 188] PA   198                 T7
[ 189] ARN  75           ,GR   201               ;STARTPARAMETER
[ 190] PA   192                 T499
```

```
[ 191] PA   193                 T511
  192] GRN  499                 T1
[ 193] BT   511                 T-1
  194] HV   192
[ 195] ARN  201
[ 196] US                                   ;500 ORD TIL BUFFER
[ 197] ARN  76          ,AC     201         ;OPTAL PARAMETER
[ 198] BT   7                   T-1
[ 199] HV   190
[ 200] HR   S1
[ 201] QQ

                                            ;LAS BUFFER TILBAGE OG KONTRO
[ 202] PA   231                 T7
[ 203] ARN  75          ,GR     234         ;STARTPARAMETER
[ 204] GRN  73          ,ARN    -4
[ 205] HV   210             NT              ;HOP HVIS EJ PARITETSFEJL
[ 206] SY   64          ,SY     39
[ 207] SY   59          ,SY
[ 208] SY               ,ARN    79
[ 209] HS   251                             ;PARITETSFEJL
  210] ARN  234
  211] IL               ,GRN    72          ;500 ORD TIL FL. 512
  212] PA   227                 T510
[ 213] PA   216                 T499
[ 214] HS   87
[ 215] GR   74              MOC             ;DAN TILFALDIGT TAL
[ 216] SR   499                 T1          ;GEM TESTTAL
[ 217] HV   161             NZ              ;R := TESTTAL - TILBAGELAST T
[ 218] HV   221 LOB                         ;HOP UD HVIS R 0-39 FORKERT
[ 219] HV   161             LB              ; -  -  -   B   -
[ 220] HV   222
[ 221] HV   161             NB              ; -  -  -   B   -
[ 222] HV   225             LOA
[ 223] HV   161             LA              ; -  -  -   A   -
[ 224] HV   226
[ 225] HV   161             NA              ; -  -  -   A   -
[ 226] ARN  79          ,AC     72
[ 227] BT   510                 T-1
[ 228] HV   214
  229] QQ   (73)                T1          ;OPTAL BLOKNR.
  230] ARN  76          ,AC     234         ;OPTAL BUFFERADR.
  231] BT   7                   T-1         ;IALT 8 BLOKKE A 500 ORD
[ 232] VY   17          ,HV     210
[ 233] HR   S1
[ 234] QQ

                                            ;OVERFØRSEL AF ARBEJDSCELLER
[ 235] ARN  66          ,GR     69
[ 236] ARN  67          ,GR     70
[ 237] ARN  68          ,GR     71
[ 238] HR   S1
[ 239] ARN  69          ,GR     80
[ 240] ARN  70          ,GR     81
[ 241] ARN  71          ,GR     82
[ 242] HR   S1
[ 243] ARN  80          ,GR     69
[ 244] ARN  81          ,GR     70
[ 245] ARN  82          ,GR     71
[ 246] HR   S1
  247] ARN  80          ,GR     66
  248] ARN  81          ,GR     67
  249] ARN  82          ,GR     68
[ 250] HV   51

                                            ;TRYKNING AF ANTAL FEJL
[ 251] PA   258                 T3
```

```
[ 252] DK    263
  253] AR    262          X
  254] MLN   261
  255] TK    30           ,GA   257
  256] SY    16           V  LZ
[ 257] SY
[ 258] BT    3                  T-1
[ 259] HV    254
[ 260] HR    S1
[ 261] QQ                    IZA
[ 262] QQ     IB
[ 263] QQN   S            XV IT
                                          ;FYLD OG LÆS BAAND
[ 264] QQ    3                  T-8       ;PARAMETER
[ 265] ARN   264          ,US   P          ;SKRIV BLOK
[ 266] ILN   P32                          ;KØR EN BLOK TILBAGE
[ 267] HS    188                          ;NULSTIL BUFFER
[ 268] ARN   264          ,IL   P          ;LÆS EN BLOK
[ 269] ARN   281          ,IL   P160       ;LÆS STATUSORD, KØR TILBAGE H
[ 270] ARN   280          ,IL
  271] ARN   -4
  272] SY    64
  273] SY    18           ,SY   19
  274] SY    49           ,SY   19
[ 275] SY    20           ,SY   18
[ 276] SY                 ,SY
[ 277] HS    119                          ;UDSKRIV STATUSORD
[ 278] SY    64
[ 279] HR    S1
[ 280] QQ    P-4          XVDLRC T1        ;PARAMETER
                                          ;SKRIVNING AF BLOK MED OVERSP
[ 281] QQ    P            XVDLRC T1        ;PARAMETER
[ 282] PA    300                T2
[ 283] PA    325                T5
[ 284] ARN   264          ,US   P          ;SKRIV BLOK
[ 285] ARN   281          ,IL   P160       ;LÆS STATUS,KØR BLOK TILBAGE
[ 286] ARN   280          ,IL
[ 287] ARN   -4
[ 288] HV    290             LT
  289] HV    312
  290] SY    64           ,SY   18
  291] SY    59           ,SY   39
  292] SY    54           ,SY   59
[ 293] SY                 ,SY   33
[ 294] SY    53           ,SY   55
[ 295] SY                 ,SY   34
[ 296] SY    43           ,SY   41
[ 297] SY    53           ,SY   41
[ 298] SY                 ,SY   38
[ 299] SY    36           ,SY
[ 300] BT    2                  T-1
[ 301] HV    284
[ 302] ARN   264          ,US   P32        ;SKRIV BLOK MED OVERSPRINGNIN
[ 303] ARN   281          ,IL   P160        ;LÆS STATUSORD,KØR TILBAGE H
[ 304] ARN   280          ,IL
[ 305] ARN   -4
[ 306] HV    308             LT
[ 307] HV    312
  308] BT    5                  T-1       ;SKRIV MED OVERSPRINGNING 6 G
  309] HV    302
  310] PA    308                T5
[ 311] HSF   2
[ 312] HS    188                          ;NULSTIL BUFFER
[ 313] IL    P32                          ;KØR EN BLOK TILBAGE
```

```
[ 314] ARN   264          ,IL   P                ;LÆS BLOK
  315] ARN   281          ,IL   P160             ;LÆS STATUS, KØR BLOK TILBAGE
[ 316] ARN   280          ,IL
[ 317] ARN   -4
[ 318] HV    R2 [320]       LT
[ 319] HR    S2
[ 320] SY    64           ,SY   34
[ 321] SY    49           ,SY   37
[ 322] SY                 ,SY   57
[ 323] SY    34           ,SY   34
[ 324] SY    53           ,SY   64
[ 325] BT    5                   T-1
[ 326] HV    312
[ 327] PA    325                 T5
[ 328] HSF   2

*E41
```

MB test 4 (filemarktest)

Testen skriver først 10 filemarks med mellemliggende blokke.
Rewindes til Lp, læser 10 filemarks forward dernæst 10 reverse.
Checker om den er ved Lp og udskriver LP hvis dette er tilfældet (ellers ej LP)
Derefter køres frem igen til det 10. filemark og man checker information - hvis det går godt skrives OK (ellers fejl).

MB test 4 (Filemarktest)

```
41. ZO                      LKB
42. HS   102
43. PA   51                  T9
44. US   P64          ,ARN  99
45. AR   113          D
46. US                ,US   P
47. ARN  98
48. US   P            ,ARN  99
49. AR   100          D
50. US                ,US   P128
51. BT   9                   T-1
52. HV   47
53. US   P64
54. PA   56                  T9
55. IL   P64
56. BT   9                   T-1
57. HV   55
58. PA   60                  T9
59. IL   P96
60. BT   9                   T-1
61. HV   59
62. IL   P32          ,IL   P32
63. ARN  99           ,IL   P
64. ILN  P160         ,ARN  99
65. AR   101          D
66. IL
67. ARN  101
68. SR   113
69. HV   75                LZ
70. VY   16           ,SY   64
71. SY   53           ,SY   33
72. SY                ,SY   35
73. SY   39
74. HV   41
75. VY   16           ,SY   64
76. SY   35           ,SY   39
77. PA   78                  T10
78. BT   10                  T-1
79. IL   P64          ,HV   R-1 [78]
80. PA   84           ,IL   P32
81. ARN  114          ,IL   P160
82. ILN  P160         ,ARN  114
83. IL                ,PI   (101)
84. QO                   LOB T1
85. ARN  84           ,NC   10
86. HH   80
87. IL   P32          ,IL   P32
88. ARN  114          ,IL   P
89. ILN  P160         ,ARN  114
90. IL                ,SY   64
```

```
 91. ARN   101              ,SR    113
 92. HV    95                 NZ
 93. SY    38               ,SY    34
 94. HV    41
 95. SY    54               ,SY    53
 96. SY    33               ,SY    35
 97. HV    41
 98. QQ    3                  IZA  T928  (3/ 928/   0/    10,)
 99. QQ                       IB   T1    (0/   1/   0/     1,)
100. HHFN(P243)             ,NKF  P828   (243/  828/  975/  243C)
101. QQ                                   (0/   0/   0/     0,)
102. VY    17               ,SY    64
103. SY    58               ,SY    18
104. SY    19               ,SY    59
105. SY    37               ,SY    41
106. SY    59               ,SY    60
107. SY    4                ,SY    58
108. LY    112
109. CA    1                ,HV    112
110. CA    2                ,HV    112
111. NC    3                ,QQ
112. PP                     ,HR    S1
113. GMN   R426 [539]       VD MZA T426  (426/  426/  426/  426,)
114. QQ    101                IB   T1    (101/    1/    0/    1,)
                                          ]
```

UN

Test A 241

Båndtest TT

Glostrup, den 18.2.1971

JEP.

Skriver og læser blokke på 4000 ord på båndstation nr. 1. Statusord og data testes.

Indlæsning:

Hjælp: 1 1

Help 3: r< r< r<
(efterhånden som klarlampe lyser)

Er det første indlæsning skriver Gier med rødt:

name
name
1
1

mellem 2. og 3. r< . Ved følgende indlæsninger udelades de to "name".

Reservering:
(Kun Help 3)

Testen opretter to arealer:

a (≈ 16 tromlekanaler)
a241 (≈ 1 tromlekanal)

Disse kan efter endt kørsel fjernes således:
Reset - HP
clear, a<
clear, a 241<

Start:

Af sig selv i celle 10.

Skriver:

A241 Båndtest TT på båndstation nr. 1

og fortsætter:

Med 4 spørgsmål, der alle besvares med j eller n for ja eller nej.

1. Er printeren tilsluttet? ellers bruges TW.
2. Ønskes afkortet ordlængde? ellers alm.
3. Ønskes lige paritet? ellers ulige.
4. Skal der skrives på bånd? ellers læsning.

Programmet kører nu:

Momentanstop: Normal stop/start

(Båndstation afslutter båndtransport)

Afbrydes: Tryk (Reset) HP

Kan fortsættes: Type exit< (Help 3)

e CR (Hjælp)

Kan genstartes: Type e 10 CR (Hjælp)

exit, 10< eller

a 241< (Help 3)

Hvis reservering (Help 3) af testen ikke gav andre udskrifter end ovenfor nævnt, kan andre programmer køres ind imellem, og denne test hentes frem igen med

a 241<

Går det ikke godt, må strimlen ligesom i Hjælp, læses ind igen.

Nyt gennemløb: KA:=1 ved skrivning:
Det skrevne kontrollæses.

KA:=1 ved læsning:
Programmet starter med 4. spørgsmål, (skriv).

Ved at trykke (Reset) HP og skrive

exit, xx< (Help 3)

e yy CR (Hjælp)

label	xx	yy	vil programmet starte således:
e15	429	416	4. spørgsmål (skriv?)
e20	416	403	3. og 4. spørgsmål (ny paritet, alm. ordlængde)
a5	400	387	2.-3. og 4. spørgsmål (ny ordlængde, paritet, skriv?)
a0	366	353	A 241 alle 4 spørgsmål (printer, ny ordl., paritet, skriv?)
c30	10	10	Unødigt, helt forfra.

Tekst:

Hvis udskriften "system er ikke identificeret" fås, er det fordi programmet ikke ved, hvilket system teksterne er pakket efter. Operatøren kan hjælpe ved at ændre indholdet i celle d 11 således: 3.9 ændres til 2.9 hvis Hjælp eller 1.9 hvis Help 3.

Label adr.:

d 11 og andre label adresser fås ved at indlæse programstrimmel (igen) med KB:= 1. (Evt. printer som output enhed på bit 5)
d 11 = 621 (Help 3) 607 (Hjælp)

Bitmønster:

Vælges i R efter besked på skrivemaskinen.

Forslag: 111111.001000.111111.101010.010101.010100.
101010.

Eller: 111111.000000.111111.000000.111111.000000.
111111.

Ved lige paritet må en 6 bit gruppe ikke indeholde ene 0'er.

Ved afkortet ord bruges bit 24-41 ikke, må ikke ændres.

Lagerområder:

Programmet starter i celle 10 og slutter i celle d12: 608 Hjælp, 622 Help 3. Celle 623 til celle 1022 incl. bruges til at danne og opbevare 1/10 bloks bitmønster i. Bufferen opbevarer 1 blok fra celle 0 til celle 3999 incl.

Statusord passerer bufferen i celle 4095 og gemmes i Gier celle b1:
76 Hjælp, 80 Help 3

Programudskrift: Strimlen med programmet kan på printer direkte udskrives som på en flexowriter:

Help 3: Tryk (reset) HP:

Læg strimlen i læseren og skriv:

I. edit, o 1, l 13, n, l 11, r w< å

alle stopkoder udskrives som w

II. copy, 8< hvor 8 kan ændres til anden outputenhed. copy< giver output på perforator.

Det oversatte program udskrives således:

Tryk (Reset) HP.

Originalprogram, skriv:

1, print, a, p 0.10.15.22<

Programmet som det så ud v. HP tryk, skriv:

1, print, p 10.. 622<

Hjælp: Originalprogram: Ingen båndstationer må have nr.

1. Oversæt program. Når det stopper (venter på båndstation nr. 1):

Reset - HP, skriv:

h tryk
gpr 10 t 608 } alle 3 linier afsluttet med CR.
e

Det kørende program: Tryk (Reset) HP.

Skriv det samme som ovenfor.

Fejludskrift: Der er 2 afsnit for fejludskrift. Et for læs og et for skriv. Udskriftens form styres af indikatorbit. RA-RB-PA-PB må ikke røres midt i en programløkke, de bruges internt i programmet.

Nærmere se følgende afsnit.

Sekvenser:

a27 : 1-tryk
Fejludskriftkontrol ved læsning.

b0 : S/1 tryk
Når fejl, så a27 eller c13.

c0 : tal tryk
Udskriver 12 bit som 4 cifre decimaltal.

c2 : bloknr.tryk
Udskriver bloknr. ved hver fejludskrift.

c5 : statustryk
Udskriver statusord ved fejl.

c6 : bit-7-tryk
udskriver R i 7 - 6bit grupper

c8 : svar
Tager mod besked efter spørgsmål ja/nej

c10 : checkskriv
Hvis skrivning $\wedge$ (KB $\vee$ p-fejl), så læs-kontrol blokken med fejludskrifter.

c13 : s-tryk
Fejludskrift kontrol ved skrivning

c19 : checkstatus
Kontrollerer statusord efter hver blok.

c25 : bit-4-tryk
Udskriver R i 4 grupper

c7, d3 : teksttryk
Udskriver tekster både i Help 3 og Hjælp, pakket med t i følgende celler:

a1, a2, a3, a4, a7, a18
b5, b6, b20, b24, b29
c3, c4, c24, c26, c27
d8
e0, e6, e7, e8

Konstanter:

Parameter ord til skrivning og læsning på båndstationen, andre konstanter og bitmønstre er indsat/kan gemmes i følgende celler:

a6 : bloktæller. 39

a8 : bitmønster ønsket

a9 : paramord CS → buffer (tæller i R reg.)

a10 : paramord buffer ↔ TM-7

a11 : tælletal. 39 til a9 og a13

a12 : tælletal til a6 og b23

a13 : tællende paramord buffer → CS

a17 : bitmønster til "clear" buffer

a30 : antal skrevne blokke. 39

b1 : statusord

b2 : paramord hent statusord

b22 : paramord clear buffer

b23 : tællende paramord clear buffer

b26 : Konstant til beregning af bufferadresse

b30 : antal fejlord. 39 i en blok

d9 : gem M

d10 : gem R

d11 : system identifikation

d12 : kanal 0 start

Handlingsforløb:

Testen starter med at stille skrivemaskine og en eventuel printer i startsituation. Undersøger så, hvilket system den er indlæst med af hensyn til teksttryk sekvensen.
Nu overspringes alle tekster, sekvenser og konstanter, som fylder ca. 300 celler. Teksttryk sekvensen er dog placeret sidst i programmet.
Derpå følger:

a0 : Start afsnit
der indstiller testen til med/uden printer.

a5 : Paritet/ordlængde afsnit
der indstiller skrivning/læsning efter ønske.

e15 : Rewind/skriv/læs afsnit
vælger skriv eller læs - program.

Skriv afsnit:

e22 : Gem ønsket bitmønster
Skriv ønsket bitmønster
Dan inverteret bitmønster til "clear" buffer
Gem inverteret bitmønster
d13 : Dan blok i buffer, hvis den ikke er intakt
b4 : Skriv blok på båndstation
Kontroller statusord
Skriv/kontroller 1 gang mere, hvis p-fejl ∧ QA=0
Kontrollæs blok med c10, hvis KB ∨ p-fejl
b3 : Tæl bloktæller op
Skriv antal blokke og gå til 'læs afsnit', hvis KA ∨ ECT ellers fortsæt fra d13 med ny blok.

Læs afsnit:

a20 : Rewind, 0 stil bloktæller
a23 : clear buffer, (fyldes med inverteret bitmønster)
e3 : Læs blok fra båndstation
Kontroller statusord
Læs/kontroller 1 gang mere, hvis p-fejl ∧ QA=0
c28 : Kontrollæs blok
a21 : Gentag samme blok hvis TA (fra a 23)
Tæl bloktæller op
Gå til e15, når alle blokke er læst ∨ KA
ellers fortsæt fra a23 med ny blok.

For hver blok skrevet eller læst, udfører tw et farvebåndsskift. Det er en kontrol på, at programmet stadig kører.

Ligeledes kan man tydelig skelne de enkelte blokke fra hinanden, og derved ved man, at TM-7 ikke er "stukket af".

Fejludskrift - skriv: På t.w. skrives besked om eventuelle p-fejl efterfulgt af bloknr. Derpå følger eventuelle andre fejludskrifter afhængig af indikatorbit:

RA - RB - PA - PB bruges internt, bør ikke røres.

Indikatorer ændres således nårsomhelst:

KA - KB på deres knapper.

Resten: Normal stop, vælg in-register
stil bit 0 - 9, normal start.

Normalstilling: alle bit = 0

Styring af testen under skrivning ved hjælp af indikatorbit:

KA:=1 Stop skrivning når blokken er færdig, rewind, gå til læseprogram.

KB:=1 Læs og kontroller hver skreven blivende blok med det samme.

(eller p-fejl) Uden KB er der kun p-fejl-check af det skrevne, men i tilfælde af p-fejl reagerer programmet som om KB var 1. Køres der med omskrivning, læs/kontrolleres den 1. defekte blok ikke.

QA:=1 Ingen omskrivning (læsning) v. p-fejl
(Normalt 1 omskrivning)

QB:=1 Bloklængden i statusordet udskrives også binært.
QB sættes =1 af programmet, hvis der er bloklængdefejl.

Datafejl: Udskrift af datafejl på printer: (eller skrivemaskine afhængigt af svaret på spørgsmålet: printer (JXN):)

OA=OB=TA=TB=0 bloknr.
Fejlbit 7-6 bitgrupper, bufferadresse decimalt
(rødt) (sort)
(på printer)
De 4 ovennævnte bit er prioriteret.
Sættes en eller flere =1 er det kun højeste prioritet der udføres.

1. TB:=1 bloknr. på printer df på t.w.
df (rødt)

df (datafejl), blot 1 fejl giver ovenstående udskrift, skipper resten af kontrollæsningen og begynder på næste blok.

2. OB:=1 bloknr xxx fejlramte ord i blokken
bloknr på printer (med rødt)
bloknr på t.w.: eller
bloknr 0 fejlramte ord i blokken
(med sort)

ingen bitmønstre eller adresser, kun antallet af fejlord i blokken.

3. OA:=1 bloknr
f f f f f ---- f f f på printer
f f f f ----
50 pr. linie, 1 f for hvert fejlord
3 gange bliver linien skåret over af bloknr udskrift, men stadig 50 f'er pr. linie.

4. TA:=1 bloknr
bufferadresse (i rødt) på printer
bufferadressen for hver fejlord udskrives.

Fejludskrift - læs: På t.w. skrives besked om eventuelle p-fejl efterfulgt af bloknr.
Derpå følger eventuelle andre fejludskrifter afhængig af indikator bit:

RA - RB - PA - PB bruges internt, bør ikke røres.

Indikatorer ændres således nårsomhelst:

KA - KB på deres knapper

Resten: Normal stop, vælg in-register
stil bit 0 - 9, normal start.

Normalstilling: alle bit =0

Styring af testen under læsning ved hjælp af indikatorbit:

KA:=1 Stop læsning når blokken er færdig,
rewind, gå til skriv (JXN): spørgsmål

QA:=1 Ingen omlæsning ved p-fejl, normalt 1 omlæsning.

QB:=1 Bloklængden i statusordet udskrives også binært.
QB sættes af programmet hvis der er bloklængdefejl.

TA:=1 Gentag samme blok.

TA kan sættes under en bloklæsning eller fejludskrift. Sålænge TA er sat, testes samme blok. Når teksten "blokken er færdiglæst" fremkommer, tager man pånyt stilling til bl.a. TA. Fjernes den fås samme blok en gang mere, hvorpå testen kører som uden TA.

KB:=0 bloknr.
Fejlbit 7-6 bitgrupper, bufferadresse decimalt
(rødt) (sort)
(på printer)

KB:=1
og De 3 nedennævnte bit er priotetet.
Sættes 1 eller flere =1 er det kun højeste priotet der udføres.
Der stoppes for valg af indikator efter hver blokkontrol sålænge KB=1

1. TB:=1 bloknr på printer df på t.w.
df (rødt)

df (datafejl), blot 1 fejl giver ovenstående udskrift, skipper resten af kontrollæsningen og begynder på næste blok.

2. OB:=1 bloknr xxx fejlramte ord i blokken
bloknr på printer (med rødt)
bloknr på t.w.: eller
bloknr 0 fejlramte ord i blokken
(med sort)

ingen bitmønstre eller adresser, kun antallet af fejlord i blokken.

3. OA:=1 bloknr
fffff---fff på printer
fffff----

50 pr. linie, 1 f for hvert fejlord
3 gange bliver linien skåret over af bloknr. udskrift, men stadig 50 f'er pr. linie.

4. OA=OB=TB=0 bloknr
bufferadresse (i rødt) på printer
(og stadig KB=1)

bufferadressen for hvert fejlord udskrives.

Praktiske bemærkninger:

1. Har man et bånd der ved gentagne skrivninger (forfra) kun giver fejl i nogle bestemte bloknr, er dette bånd dårligt og båndstationen OK.
Kasser båndet eller fraklip det dårlige stykke eller mærk båndet således:

Båndfejl i blok 27 og 62
ved 4000 ord/blok, 556 bpi

2. Spørgsmålet skriv (JxN): kommer først, når båndet er rewindet (står i BOT) og båndstationen er i remote.

3. p-fejl skriv betyder ikke nødvendigvis at båndet ikke kan læses uden p-fejl og datafejl. F.eks. vil dobbelt skew register kunne redde svage pletter eller skævt bånd??

4. Hvis der er bloklængde-fejl vil statusord fortælle hvor mange ord der er læst. Det der mangler vil stadig henstå i bufferen som det inverterede bitmønster og udskrives sådan ved fejludskriften.

5. En enkelt installation har en anden ordlængde i afkortet ord.

6. Indlæsning under job-system:

Skriv:		svar:	
Space			job
103 , r<			xxxx
r <			(name)
			(name)
			1
			1
			endjob xxx
Space			job
103 , a241<			xxxxx

Forkortelser: CR betyder vognretur

Fejl eks.:

1.

	p-fejl skriv
Blok x	bloknr x
	p-fejl læs
	bloknr x
	2900 fejlramte ord i blokken
Blok y	bloknr y
	df
Blok z	bloknr z
	df

2. Eksempel på fejl oveni EOT mærket:
(stopper selv efter EOT).

```
bits i R, tryk start
111111 111111 111111 111111 ..... ..... .....    ønsket bitmønster

 bloknummer: 968
111111 111111 .11111 .11111 ..... ..... .....  1997
.11111 .11111 .11111 .11111 ..... ..... .....  1998
.11111 .11111 .11111 .11111 ..... ..... .....  1999
.11111 .11111 .11111 .11111 ..... ..... .....  2000
.11111 .11111 .11111 .11111 ..... ..... .....  2001
.11111 .11111 111111 111111 ..... ..... .....  2002
 antal skrevne blokke: 968
læsning, vælg indikator, normalt 0
KA=1: restart
tryk start
```

3. Eksempel på stikfejl i adapter, der gav forkert bit i statusord, rutineprogrammer kørte ikke, da de gik ud fra, at ubrugte bit i statusord skulle være 0.

```
bits i R, tryk start
111111 111111 111111 111111 111111 111111 111111    ønsket bitmønster
uønsket bit i statusord: ....11 .. 11111.1..... 1.....................
uønsket bit i statusord: ....11 .. 11111.1..... 1.....................
   0 fejlramte ord i blokken
uønsket bit i statusord: ....11 .. 11111.1..... 1.....................
uønsket bit i statusord: ....11 .. 11111.1..... 1.....................
   0 fejlramte ord i blokken
uønsket bit i statusord: ....11 .. 11111.1..... 1.....................
uønsket bit i statusord: ....11 .. 11111.1..... 1.....................
   0 fejlramte ord i blokken
 antal skrevne blokke:  3
læsning, vælg indikator, normalt 0
```

Notater:

Kopi Kopi

OK

```
[ addertest 1.Tester adderen ved gennemløb af nedenstaaende operationer.
              KA = 1  stop hvis adderfejl
              hvis adderen regner rigtigt skrives 1 over bit 5
                -     -       -     forkert    -    3   -   -   -
              programmet starter med ordren zq LKB]

i=41
[  41] zq                 LKB      ; stop hvis  kb = 1
[  42] vy   16    V
[  43] sy   1                      ; skriv 1 hvis ok
[  44] pan  58    X       t1       ; retablering
[  45] arn  65                     ; overførsel af testtal
[  46] gr   69
[  47] arn  66
[  48] gr   70
[  49] arn  67
[  50] gr   71
[  51] mk   69                     ; start test
[  52] sc   70
[  53] dl   71    X
[  54] ac   69
[  55] ml   70
[  56] ac   71
[  57] zq                 LKB
[  58] bt   1                t1    ; kør igennem 512 gange
[  59] hv   50
[  60] sr   68                     ; subtraher resultatet
[  61] hv   43            LZ       ; hop hcis ok
[  62] zq                 LKA      ; stop ved fejl
[  63] sy   3                      ; skriv 3 ved fejl
[  64] hv   44
[  65]        123 456 789,
[  66]        234 567 891,
[  67]        345 678 912,
[  68]      2 134 743 605,
[  69]                  0,
[  70]                  0,
[  71]                  0,
e41
```

Kopi Kopi

OK

```
[ addertest 2.Tester adderen ved gennemløb af nedenstaaende operationer.
              KA = 1  stop hvis adderfejl
              hvis adderen regner rigtigt skrives 1 over bit 5
                -    -      -    forkert   -   3   -   -  -
              programmet starter med ordren zq LKB]

i=41
[  41] zq            LKB          ; stop hvis kb = 1
[  42] vy   16   V
[  43] sy   1                     ; skriv 1 hvis ok
[  44] pan  73   X      t1        ; retablering
[  45] arn  80                    ; overførsel af testtal
[  46] gr   84
[  47] arn  81
[  48] gr   85
[  49] arn  82
[  50] gr   86
[  51] ar   84                    ; start test
[  52] nk   -1
[  53] ac   85
[  54] tk   7
[  55] sr   86
[  56] tl   -7
[  57] sc   84
[  58] ck   7
[  59] ar   85
[  60] cl   -7
[  61] ac   86
[  62] nl   -1
[  63] sr   84
[  64] tk   -7
[  65] sc   85
[  66] tl   7
[  67] ar   86
[  68] ck   -7
[  69] ac   84
[  70] cl   7
[  71] sr   85
[  72] sc   86
[  73] bt   1          t1         ; kør igennem 512 gange
[  74] hv   51
[  75] sr   83    ,ck              ; subtraher resultatet
[  76] hv   43      LZ             ; hop hvis ok
[  77] zq           LKA            ; STOP VED FEJL
[  78] sy   3                      ; skriv 3 ved fejl
[  79] hv   44
[  80]     1 234 567 890,
[  81]     2 345 678 901,
[  82]     3 456 789 012,
[  83]  336/  34/ 406/ 528,
[  84]                 0,
[  85]                 0,
[  86]                 0,
e42
```

8K

```
[ addertest 4.Tester adderen ved gennemløb af nedenstaaende operationer.
              KA = 1  stop hvis adderfejl
              hvis adderen regner rigtigt skrives 1 over bit 5
                -     -      -     forkert    -    3    -   -  -
              programmet starter med ordren zq LKB]

i=41
[  41] zq              LKB          ; stop hvis kb = 1
[  42] vy   16
[  43] arn  75
[  44] gr          V   M            ; tilbagehop ved fl.overløb
[  45] sy   1                       ; skriv 1 hvis ok
[  46] pan  68      X       t1      ; retablering
[  47] arfn 76                      ; overfør testtal
[  48] grf  81
[  49] arfn 77
[  50] grf  82
[  51] arfn 78
[  52] grf  83
[  53] arfn 79
[  54] grf  84
[  55] arf  81                      ; start test
[  56] mkf  82
[  57] acf  83
[  58] anf  84
[  59] mlf  81      X
[  60] scf  82
[  61] srf  83
[  62] mtf  84
[  63] grf  81
[  64] snf  82
[  65] dkf  83
[  66] gmf  84
[  67] dlf  81
[  68] bt   1               t1      ; kør igennem 512 gange
[  69] hv   55
[  70] sr   80                      ; subtraher resultatet
[  71] hv   45          LZ          ; hop hvis ok
[  72] zq               LKA         ; stop ved fejl
[  73] sy   3                       ; skriv 3 ved fejl
[  74] hv   46
[  75] hv   56
[  76]      1 234 567 890,
[  77]      2 345 678 901,
[  78]      3 456 789 012,
[  79]      4 567 890 123,
[  80]  848/ 532/ 585/ 312,
[  81]                 0,
[  82]                 0,
[  83]                 0,
[  84]                 0,
e41
```

```
[ ADDERTEST 5. TESTER ADDEREN VED GENNEMLØB AF SAMTLIGE ARITMETISKE
  OPERATIONER PAA FAST OG FLYDENDE FORM.
          KA = 1  STOP HVIS ADDERFEJL
  HVIS ADDEREN REGNER RIGTIGT SKRIVES 1 OVER BIT 5
  -      -      -     FORKERT    -      3    -    -    -
  PROGRAMMET STARTER MED ORDREN ZQ LKB ]

I = 41

[  41] ZQ              LKB          ; STOP HVIS KB = 1
[  42] ARN  89     ,VY   16
[  43] GR              M            ; PLACER TILBAGEHOP VED FLYDENDE OVERLØB
[  44] PAN  81     X       T1       ; RETABLER TÆLLING - NULSTIL M
[  45] ARFN 90                       ; OVERFØRSEL AF TESTTAL
[  46] GRF  95
[  47] ARFN 91
[  48] GRF  96
[  49] ARFN 92
[  50] GRF  97
[  51] ARFN 93
   52] GRF  98
[  53] ARF  95                       ; START TEST
[  54] MKF  96
[  55] ACF  97
[  56] ANF  98
[  57] MLF  95     X
[  58] SCF  96
[  59] SRF  97
[  60] MTF  98
[  61] GRF  95
[  62] SNF  96
[  63] DKF  97
[  64] GMF  98
[  65] DLF  95
[  66] AR   95
[  67] MK   96
[  68] AC   97
[  69] AN   98
[  70] ML   95     X
   71] SC   96
[  72] SR   97
[  73] TL   =1
[  74] QQ
[  75] MT   98
[  76] GR   95
[  77] SN   96
[  78] DK   97
[  79] GM   98
[  80] DL   95
[  81] BT   1            T1
[  82] HV   53                       ; KØR IGENNEM 512 GANGE
[  83] SR   94     ,CK               ; SUBTRAHER RESULTATET
[  84] SY   1          LZ            ; SKRIV 1 HVIS ADDER OK
[  85] HV   44         LZ            ; HOP TIL NY TEST
[  86] ZQ              LKA           ; STOP HVIS ADDERFEJL
[  87] SY   3                        ; SKRIV 3 HVIS ADDERFEJL
[  88] HV   44                       ; HOP TIL NY TEST
[  89] HV   54                       ; TILBAGEHOP VED FLYDENDE OVERLØB
[  90] 1234567890
[  91] 2345678901
[  92] 3456789012                    ; TESTTAL
[  93] 4567890123
[  94] 275/446/57/624                ; RESULTATET AF ADDERTEST
```

Fi Test

TEST A-211
Diskfile on Buffer
RIALTO 21/6/66
JFM

FAST INFORMATIONTEST OF DISKFILE ON BUFFER

The program writes some, more or less, random numbers on the entire file and check this information. The check is performed as a sumcheck, and the program types 'sum-error' in red whenever this summation fails. At the start-point the filenumber (0, 1, 2 or 3) should be typed.

Jan Flemming Madsen

e41

3-11-85 Pikat I

```
;SLIP<

_B I=41,A8,D20

        ZQ          LKB
        VY     17, SY      64
        SY     60, SY      54
        SY     57, SY      58
        SY     32, SY      19
        SY     53, SY      18
        SY     19, SY      64
A7:     SY     64
        SY     54, SY      57
        SY     35, SY      53
        SY      0, LYN     D6
        CA     16, GRN     D6
        ARN  (D6)   D
        SR      9   DITA
        HH     A7   NTA
        PP   (D6)
        GRN    D0, GRN      D1
A3:     PA     A0   T    199
        ARN    D1, UD       A0
        HV     A2   NZ
        ARN    42, GR       D1
A2:     PMN  (A0)   D
        MK     D1, GR    (A0)
        AC     D0
A0:     QQ      _0   T        1
A1:     ARN   (A0)   D
      NC   820, HV       A2
        ARN    D1, SR      42
        GRN    D1   LZ
D12:    ARN    D2
        US      0, ARN     D1
        CK     18, AR      D3
        US    P24
        ARN    A8, IL      P24
        IL      0, ARN     512
        HS     D7   LT
        QQ    D12
        ARN   512, CK       10
        NC     620
        HS    D7
        QQ    D12
        ARN    D5, AC      D1
        ARN    D1
        SR     D4   ITA
        HV     A3   LTA
SY64
ARND0
TK1
SY1VLT
SY
VR-3NZ
GRN    D6, GRN    D1
```

```
        AR      D5
        IL      P8
        ARN     R1    V
A8:   QQ 512.9+1.19+4095.39
        IL     P24, IL       0
        ARN    512
        HS      D7    LT
        QQ      A6
        ARN    512
        CK      10, NC     620
        HS     D7
        QQ      A6
        ARN     D2, IL       0
        PA      A4    T     199
A4:     ARN      _0    T       1
        AC      D6
A5:     ARN    (A4)  D
        NC    819, HV      A4
        ARN     D5, AC      D1
        ARN     D1
        SR      D4    ITA
        HV      A6    LTA
SY64
ARND6
TK1
SY1VLT
SY
HVR-3NZ
SY64
ARN     D0, SR      D6
      HV     D11    NZ
      SY      64, SY      38
      SY      34, HV A3-1
D11:  SY      64, SY      29
       SY      18, SY      20
       SY      36, SY      32
       SY      53, SY      41
       SY      41, SY      38
       SY      41, SY      62
       HV    A3-1
D0:     QQ
D1:     QQ
D2:     QQ 200.9 +620.19 +0.39
D3:     QQ  620.9 +_0.21 +0.39
D4:     QQ 2030.39
D5:     QQ 1.39
D6:     QQ
D7:     SY      64, SY      58
        PA      D8    T       7
D9:     TK       1
        SY       1    LTV
        SY      32
D8:     BT       _7    T      -1
        HV      D9
        SY       0, ARN    512
        TK      10
        CK      10    X
        HS     D10
        QQ       3, SY       0
        PM      D1
        HS     D10
        QQ       4
        ZQ             LKA
        HR    (S1)    LKB
        HR      S2
```

```
_D D10=I
[NUMBERPRINT
THE ROUTINE PRINT A POSITIVE  INTEGER IN M.
THE INTEGER MUST BE WITH UNIT IN POSITION 39.
ENTRY:      HS <NUMBERPRINT>
            QQ <NUMBER OF FIGURES TO BE PRINTED>
RETURN:     HR S+1
<NUMBER OF FIGURES TO BE PRINTED> MUST BE GREATER THAN, OR EQUAL TO
THE NUMBER OF SIGNIFICANT DIGITS IN THE NUMBER IN M.IF NOT, THE PRINTED
NUMBER WILL BE INCOMPLETE.
]
_B C8
        PT      RC3    T       C2
        GR      RC2, ARN     RC7        ; SAVE R, RADR:= PRINTPLACES
        SR     (S1)    D
        CK      -10, AC      RC3        ; C3COUNT:DDDRESS OF FIRST DIVISOR - 1
        GAN     RC4                     ; CHAR:= SPACE, NOPRINT:= _T_R_U_E
C3:     PA      RC6    T       C2       ; SET DIVISOR ADDRESS
C6:     DL        _0    T        1       ; R:= CIFFER IN POSITION 39
        CK      -10, BS   (RC4)         ; RADR:= CIFFER
C5:     AR       16    DLZ              ; _I_F CIFFER=0 ^ -,NOPRINT _T_H_E_N CIFFER:
        GA      RC4, ARN     RC6        ; CHAR := CIFFER, RADR := DIVISOR ADDRESS =1
        NC       C0, HV      RC4        ; _I_F DIVISOR |= -1 _T_H_E_N PRINT(CIFFER)
        BS    (RC4), HV      RC4        ; _I_F CHAR |= SPACE _T_H__E_N PRINT(CIFFER)
        ARN     RC5, GA      RC4        ; _I_F NOPRINT ^ LAST DIGIT =0 _T_H_E_N CIFFE
C4:     SY        _0, ARN    RC6        ; PRINT(CIFFER ~ CHAR)                  ↳ :=16
        NC       C0, HVN     RC6        ; _I_F DIVISOR |= 1 _T_H_E_N TAKE NEW DIVISOR
        ARN     RC2, HR       S1        ; _E_L_S_E RESTORE R AND RETURN
C2:     QQ                              ; USED FOR SAVE R
_M
C1:     1 000 000 000 00
        1 000 000 000 0
        1 000 000 000
        1 000 000 00
        1 000 000 0
        1 000 000
        1 000 00
        1 000 0
        1 000
        1 00
        1 0
C0:     1
 D C8=C0-C2
C7:     QQ    C8.9
_E
_S
_E41
```

Disk test 7

Programmet skriver og læser fra disken.

Alle ord bliver ens.

Ordenes indhold kan indsættes i R ved start.

Alle læste ord bliver checket for korrekt indhold; hvis fejl, fremkommer den fejllæste information i R , og programmet stopper. I M ses samtidig nr. af det fejlramte ord(enh. i pos.9) og nr. af det fejlramte spor(enh.i pos.21).

Hvis det ønskes kan zq fjernes ved at ændre celle 74 til hv124.

Hvis statusord siger fejl, om-læses/skrives op til 3 gange, derefter fortsættes med næste spor.

Programmet udskriver statusord over bit 5, hvis fejl.

HVIS KA: skrives 1 gang, derpaa gentages læsning

HVIS KB: stop for ny information efter afsluttet læsning.

Udskrifter:

bit nr. i statusord efterfulgt af s eller l, eftersom fejlen er fremkommet under henholdsvis us24 eller il8.
Hvis der forekommer informationsfejl skrives f.

NB: CR er skilletegn ved indlæsning.

19/5 JdJ.

D I S K T E S T 7A

b d
t i s udsat: d=unit, d1=checklæs (0, 16)

```
     i=10
10:  vy17
     sy64
     sy64                indlæsning af ny information
     hs162
     hv166
15:  arn128
     gr97
     arn1000
     zq        ----------------------------------------------------
     gr1000
20:  arn1000
     pa22t299
     gr299t1
     bt400t-1                 dan blok i buffer
     hv22
25:  bt225t-1
     hv22
     pa23t400
     pa25t225
     arnr1V
30:  qq300.9+625.19+100.39
     us          ------------------------------------------------
     arnr1V
->   qq620.9+100.39
     arr1V
->35: qq100.21
     us24 d+fd1
     hs100
     hv41LDA
     pa41t2              skriv paa n antal spor, udskriv
40:  hv45                 statusord hvis fejl
     bt2t-1
     hv118
     pa41t2
     sy18,sy0
45:  arr1V
     qq1.21
     hs136
     hv36
     hv86LKB     ----------------------------------------------------
50:  arn 219
     ar35
     il8 d
     hs100
     hv57LDA
55:  pa57t2
     hv61
     bt2t-1              læs et spor og anbring blokken
     hv121               i gier (celle 300 ->)
     pa57t2
60:  sy35,sy0
```

```
     ar46
     gr1001
     arn220
     il
 65: pa67t299
     arn1000
     sr299t1
     hv75LZ
     arn1001,sr33
 70: gr90,arn67
     sr91,ar90
     gr92,pm92
     arn(67)
     zq,hv124    : kan ændres til hv124
 75: bt400t-1
     hv66
     bt218t-1
     hv66
     pa75t400
 80: pa77t224
     arn1001
     hs136
     hv52NKB
     hv86LKB
 85: hv50LKA
     arn1000LKB
     hv10LKB
     arn1000
     hv20
 90: qq
     qq300.9+1.19+1.21+3.25
     qq
     hsf2
     hsf2
 95: hsf2
     qq
     qq
     hsf2
     hsf2
     i=100
100: gr1002
     arnr1V
     qq1003.9+1.19+4095.39
     il24
     il
105: pi0,pi(1003)
     arn1002
     hv109LCA
     hrs1
     sy1LCB
110: sy2LTA
     sy3LTB
     sy4LPA
     sy5LPB
     sy6LQA
115: sy7LQB
     hs131
     hv108
```

undersøg blokken for fejl

statusordstest

```
118: sy18,sy
     hs131              skriv s
     hv36
     ------------------------------------------------------------

121: sy35,sy
     hs131              skriv l
     hv52
     ------------------------------------------------------------

124: sy54,sy
     hs131              skriv f
     hv75
     ------------------------------------------------------------

     qq1.21
128: qq64.21
     hv11
     hsf2
     ------------------------------------------------------------

131: bt8t-1
     hrs1
     pa131t8            cr. efter udskrift af 9 tegn
     sy64,sy64
     hv132
     ------------------------------------------------------------

136: gr96
     arn97
     sr127
     hv143LZ            nedtælling af sporantal
140: gr97
     arn96
     hrs1
     arn128
     gr97
145: arn96
     hrs2
     hsf2
     ------------------------------------------------------------

148: arn218,gr214
     gr216,ly216
150: arn216,sr215
     hh159LZ
     arn216,sr213       sekvens for indlæsning af variable
     hv155LZ             paa decimalform med enh. i pos.21
     arn216,ck-3o
155: gr216,pm217
     mln214X
     ar216,gr214
     arn218
     hv149,arn214
160: ck18,gr214
     hrs1
     hsf2
     ------------------------------------------------------------
```

```
      i=162
162 : pa163t6
      bt6t-1
      sy0,hv163                    7 mellemslag
165 : hrs1
      ------------------------------------------------
166 : sy18,sy35
      sy20,sy34
      sy0,sy60
      sy34,sy50
170 : sy58,sy64
      sy64
      hs162
      sy50,sy53          udskrifter og indlæsning af variable
      sy55,sy59
175 : sy0,sy18
      sy39,sy38
      sy41,sy60
      sy59,sy58
      hs150      148
180 : arn214,gr35
      sy64,sy64
      hs162
      sy49,sy37
      sy19,sy49
185 : sy35,sy0
      sy18,sy39
      sy38,sy41
      sy60,sy59
      sy58
190 : hs150        148
      arn214,gr128
      sy64,sy64
      hs162
      sy50,sy35
195 : sy38,sy34
      sy0,sy35
      sy37,sy55
      sy60,sy59
      sy58
200 : hs150        148
      arn214,ck12
      ar212,gr33
      sy64,sy64
      hs162
205 : sy18,sy53
      sy19,sy0
      sy60,sy41
      sy58
      sy64,sy64
210 : hv15
      ------------------------------------------------
211 : hsf2
      qq100.39
      qq16.9                    konstanter.
      qq
      qq64.9
      qq
      qq10.21  10.39
      qq
219qq 620.9 + 1000.39
```

T 23 Indicator test

```
;SLIP<

_B I=41, A60
  : ZQ       0   LKB

A1:   ARN   1,9   D          ; INDICATOR TEST: BIT:= 1;
      GA    RA3              ; NEXT BIT:
A3:   PP      _0, PS      P  ; P:= S:= BIT;
      ABN     P   DX         ;
      GA   RA4               ;
      PI  (RA4), IT      S   ; IN:= -,P;
      PI      P, GI   RA5    ;
      PMN (RA5)   DX         ;
      NC      0, ZQ      1   ; MASK ERROR SET;
      PI  (RA4), IT      0   ;
      PI      P, GI   RA5    ;
      PMN (RA5)   DX         ;
      NC      P, ZQ  2       ; IN ERROR SET;
PIP,IT S ;
      PI  (RA4), GI   RA5    ;
      PMN (RA5)   DX         ;
      NC   1023, ZQ      3   ; MASK ERROR RESET;
      PI      P, IT      0   ;
      PI  (RA4), GI   RA5    ;
      PMN (RA5)   DX         ;
      NC  (RA4), ZQ      4   ; IN ERROR RESET;
      PMN     P   DX         ;
      CK      1, NC      0   ; BIT:= BIT _S_H_I_F_T 1;
      GA    RA3, HV   RA3    ; _I_F BIT |= 0 _T_H_W_N _G_O_T_O NEXT BIT;

A6:   PP      0, PI      P   ; P:= 0; NEXT: IN:=P;
      GI    RA5, ARN   RA5   ; RADDR:= SELECTED IN;
      NC      P, ZQ      5   ; _I_F P |= SELECTED IN _T_H_E_N ERROR 5;
      PP     P1, NCN     P   ; P:= P+1;
      HH    RA6              ; _I_F P < 1024 _T_H_E_N _G_O_T_O NEXT;
      HV    RA7              ;

A4:   QQ      _0             ; WORK FOR GI-INSTRUCTION;
A5:   QQ      _0             ; INVERTED P;
A13:  0,75

/     GRN   RA4   IZA        ;
      QQ          IZB        ;
      HV    RA8   NZ         ;
      HV    RA8   NZC        ;
      HV    RA8   NZA        ;
      HV    RA8   NZB        ;
      HH    RA8   LZC        ;
A8:   ZQ      6, ARN   RA8   ;
      HV    RA9   IZC        ;
A9:   HV   RA10   LZ         ;
      HV   RA10   LZA        ;
      HV   RA10   LZB        ;
      HV   RA10   LZC        ; TESTED OPERATIONS: IZA IZB IZC NZA NZB NZC
      HH   RA10   NZC        ; LZA LZB LZC LZ  AND NZ ;
```

```
A10:  ZQ      7, ARN   RA13 ;
      GR     RA4     MC     ;
      ARN    RA13           ;
      AR     RA4     IOA    ;
      ARN    RA13           ;
      AR     RA4     IOB    ;
      HV     RA11    NO     ;
      HV     RA11    NOA    ;
      HV     RA11    NOB    ;
      HV     RA11    NOC    ;
      HH     RA11    LOC    ;
A11:  ZQ      8, ARN     R  ;
      GRN    RA4     M      ;
      HV     RA12    NA     ;
      HV     RA12    NB     ;
      HV     RA12    NC     ;
      ARN    RA4     IOC    ;
      HV     RA12    LO     ;
      HV     RA12    LOA    ;
      HV     RA12    LOB    ;
      HH     RA12           ; TESTED OPERATIONS: IOA IOB IOC NOA NOB NOC
A12:  ZQ      9, QQ          ; LOA LOB LOC M MC LO AND NO ;
      ARN     -1    DITA   ;
      ARN     -1    DITB   ;
      HV     RA14    NT     ;
      HV     RA14    NTA    ;
      HV     RA14    NTB    ;
      HV     RA14    NTC    ;
      ARN      0    DITC   ;
      HV     RA14    LT     ;
      HV     RA14    LTA    ;
      HV     RA14    LTB    ;
      HV     RA14    LTC    ;
      HV     RA15           ; TESTED OPERATIONS: ITA ITB ITC NTA NTB NTC
A14:  ZQ      10             ; LTA LTB LTC LT NT NA NB NC;
```

```
A15:  PI      0, PI      63 ; PA:=PB:=QA:=QB:=RA:=RB:= 1;
      HV    RA16    NPA     ;
      HV    RA16    NPB     ;
      HV    RA16    NPC     ;
      HV    RA16    NQA     ;
      HV    RA16    NQB     ;
      HV    RA16    NQC     ;
      HV    RA16    NRA     ;
      HV    RA16    NRB     ;
      HV    RA16    NRC     ;
      PI       0            ; PA:=PB:=QA:=QB:=RA:=RB:= 0;
      HV    RA16    LPA     ;
      HV    RA16    LPB     ;
      HV    RA16    LPC     ;
      HV    RA16    LQA     ;
      HV    RA16    LQB     ;
      HV    RA16    LQC     ;
      HV    RA16    LRA     ;
      HV    RA16    LRB     ;
      HV    RA16    LRC
      HV    RA17            ; TESTED OPERATIONS: REMAINING L AND N CONDITIONS;
A16:  ZQ      11            ;
A17:  PI      3, PP      2  ;
      QQ       0    IK      ;
      QQ       0    IKA     ;
      QQ       0    IKB     ;
      QQ       0    IKC     ;
      NCN   P-2, ZQ     12 ; TESTED: SWOP OPERATIONS IK,IKA,IKB AND IKC;
      GR     RA4    MB      ;
      ARN    RA4            ;
      HV    RA18    LA      ;
      HV    RA18    LC      ;
      GR     RA4    MA      ;
      ARN    RA4, PI     0 ;
      HV    RA18    LB      ;
      HV    RA19            ;
A18:  ZQ      13            ; TESTED OPERATIONS: MB LA LB LC MA;
```

```
A19:  GR     RA4    MC          ;
      ARN    RA4    IPA         ;
      ARN    RA4    IPB         ;
      ARN    RA4    IQA         ;
      ARN    RA4    IQB         ;
      ARN    RA4    IRA         ;
      ARN    RA4    IRB         ;
      GI   RA20                 ;
A20:  ARN      0    D           ;
      NC      63, ZQ       14  ;
      PI       0               ;
      ARN  RA4      IPC        ;
      ARN    RA4    IQC         ;
      ARN    RA4    IRC         ;
      GI   RA20, ARN  RA20      ;
      NC      63, ZQ       15  ; TESTED : IPA IPB IPC IQA IQB IQC IRA IRB IRC;
      ARN    RA4, PI     1023  ;
      GR     RA4    M           ;
      GR     RA4    MQA         ;
      GR     RA4    MQB         ;
      ARN    RA4                ;
      HV     RA21   NC          ;
      GR     RA4    MTA         ;
      GR     RA4    MTB         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MPA         ;
      GR     RA4    MPB         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MQA         ;
      GR     RA4    MQB         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MRA         ;
      GR     RA4    MRB         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MQC         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MTC         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MPC         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MQC         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      GR     RA4    MRC         ;
      ARN    RA4                ;
      HV    RA21    NC          ;
      HV     RA0                ;
A21:  ZQ      16                ; TESTED: MQA MQB MQC MRA MRB MRC NC;
      HV      RA0               ; MQA MQB MQC MRA MRB MRC NC;

                                ; END OF TEST 23, INDICATOR TEST;
_E41
```

UN

17.2.71
BEH

JUM Disktest

Velegnet til rutinetest af disk på Gier, da den ikke ødelægger noget.
tester i det frie område:
Udskriver:

Seperation: (et tal mellem ca. 20 og 150)
error trace: (j el n
j: alle fejl bliver udskrevet
n: man får kun udskrevet at der er fejl)

testen fortsætter i al evighed (med slukket KA og KB) og udskriver for hvert gennemløb loop nr.

tændes KA standses efter et spor
tændes KB standses efter endt gennemløb

Testen er binout og indlæses med r <, genopstart kan ske ved at taste run < (UNI dog y <)

25.2.1??

Routine test of Disk - Sam's Disktest

```
ALGOL,_N<
_B_E_G_I_N _M_E_S_S_A_G_EDISKTEST 1 / JUM;
   _I_N_T_E_G_E_RI,J,K,L,M,PLACE,DIF,DIF1, LOOP;
   _B_O_O_L_E_A_NBOO,BOO1,ERRTRACE;
   _I_N_T_E_G_E_R _A_R_R_A_YA,B[1:1200];
   _P_R_O_C_E_D_U_R_EGETT(A,B,C); _V_A_L_U_EB,C; _I_N_T_E_G_E_R_A_R_R_A_YA; _I_N_T_E_G_E_RB,C;
   _B_E_G_I_N _I_N_T_E_G_E_RI; I:=0; REP: _I_F GET(A,B,C)<0 _T_H_E_N
      _B_E_G_I_N I:=I+1; _I_F I<4 _T_H_E_N
         _B_E_G_I_N WRITETEXT(|<<READ PARITY ERROR - ABS REL|>);
            WRITE(|< -DDDD|>,C-1+_I_N_T_E_G_E_R(_B_O_O_L_E_A_NB^_2_90_1_1M),C); WRITECR; _G_O_T_O REP
         _E_N_D _E_L_S_E
         _B_E_G_I_N WRITETEXT(|<<FAULT|>); _G_O_T_O SLUT
         _E_N_D
      _E_N_D _E_N_D GETT;
   _P_R_O_C_E_D_U_R_E PUTT(A,B,C); _V_A_L_U_EB,C; _I_N_T_E_G_E_R_A_R_R_A_YA; _I_N_T_E_G_E_RB,C;
   _B_E_G_I_N _I_N_T_E_G_E_RI; I:=0; REP: _I_F PUT(A,B,C)<0 _T_H_E_N
      _B_E_G_I_N I:=I+1; _I_F I<4 _T_H_E_N
         _B_E_G_I_N WRITETEXT(|<<WRITE PARITY ERROR - ABS REL|>);
            WRITE(|< -DDDD|>,C-1+_I_N_T_E_G_E_R(_B_O_O_L_E_A_NB^_2_90_1_1M),C); WRITECR; _G_O_T_O REP
         _E_N_D _E_L_S_E
         _B_E_G_I_N WRITETEXT(|<<FAULT|>); _G_O_T_OSLUT
         _E_N_D _E_N_D _E_N_D PUTT;
   WHERE(|<<FREE|>,PLACE); WRITECR; WRITETEXT(|<<SEPERATION: |>);
   DIF:=READINTEGER; _I_FCHAR|=64 _T_H_E_N WRITECR; WRITETEXT(|<<ERROR TRACE |>);
   ERRTRACE:=LYN|=24; WRITECR; DIF1:=DIF-2; LOOP:=0;
   _F_O_R I:=1 _S_T_E_P 2 _U_N_T_I_L DIF1 _D_O
   _B_E_G_I_N _F_O_R J:=1 _S_T_E_P 1 _U_N_T_I_L 1200 _D_O A[J]:=I;
      PUTT(A,PLACE,I); PUTT(A,PLACE,I+DIF) _E_N_D _F_O_R;
   NEXT:
   _F_O_R I:=1 _S_T_E_P 2 _U_N_T_I_L DIF1 _D_O
   _B_E_G_I_N _I_F KBON _T_H_E_N _G_O_T_O SLUT; GETT(A,PLACE,I); GETT(B,PLACE,I+DIF);
      BOO:=BOO1:=_T_R_U_E; _F_O_R J:=1 _S_T_E_P 1 _U_N_T_I_L 1200 _D_O
      _B_E_G_I_N _I_F A[J]|=I^BOO _T_H_E_N
         _B_E_G_I_N WRITE(|< -DDDD|>,I,J,A[J]); WRITECR; BOO:=_F_A_L_S_E _E_N_D;
         _I_F B[J]|=I^BOO1 _T_H_E_N
         _B_E_G_I_N WRITE(|< -DDDD|>,I+DIF,J,B[J]); WRITECR; BOO1:=_F_A_L_S_E _E_N_D _E_N_D _F_O_R;
      _I_F ERRTRACE _T_H_E_N BOO:=BOO1:=_T_R_U_E;
      _I_F (-,BOO)v(-,BOO1) _T_H_E_N
      _B_E_G_I_N SELECT(8); _F_O_R J:=1 _S_T_E_P 4 _U_N_T_I_L 1197 _D_O
         _B_E_G_I_N WRITE(|<-DDDD|>,J,A[J],A[J+1],A[J+2],A[J+3]); WRITECHAR(0); WRITECHAR(0);
            WRITE(|<-DDDD|>,B[J],B[J+1],B[J+2],B[J+3]); WRITECR _E_N_D;
         SELECT(17) _E_N_D _E_N_D _F_O_R;
   LOOP:=LOOP+1; WRITEINTEGER(|<-DDD|>,LOOP); WRITETEXT(|<< LOOP|>); WRITECR;
   _I_F-,KAON _T_H_E_N _G_O_T_ONEXT;
   SLUT: _E_N_D T<
```