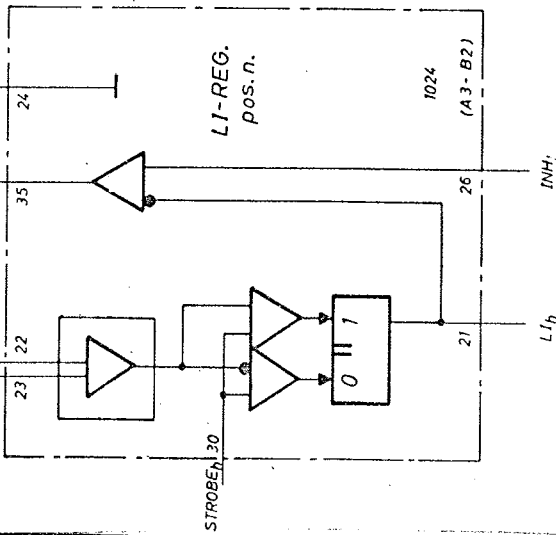
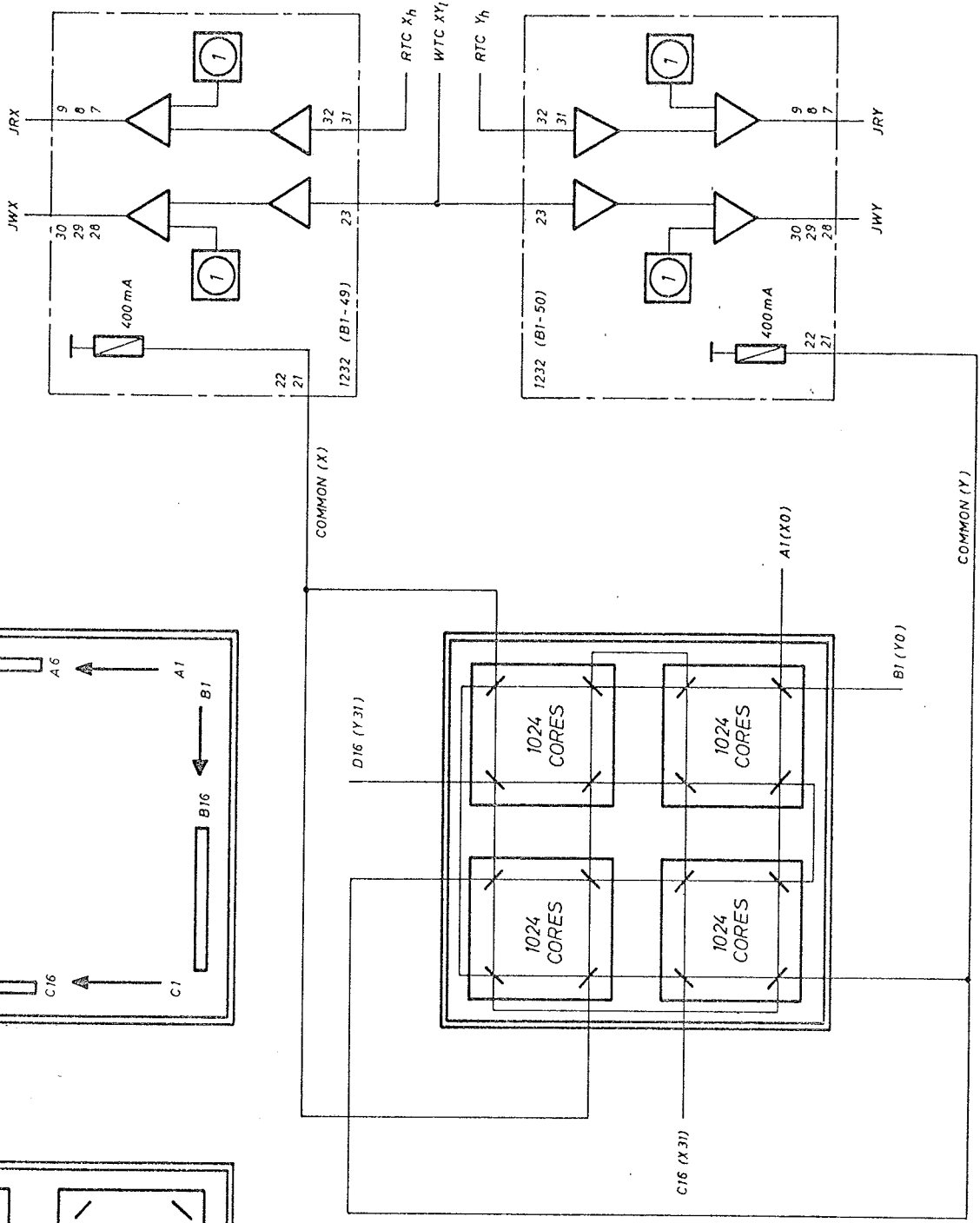
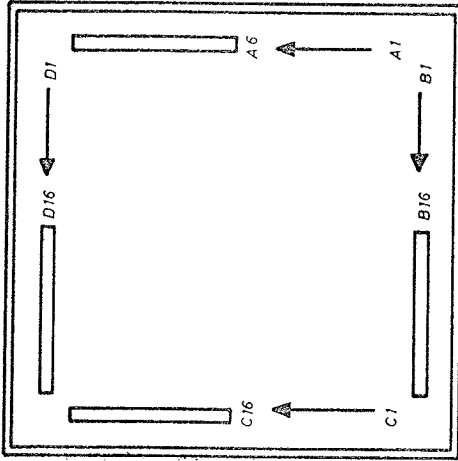
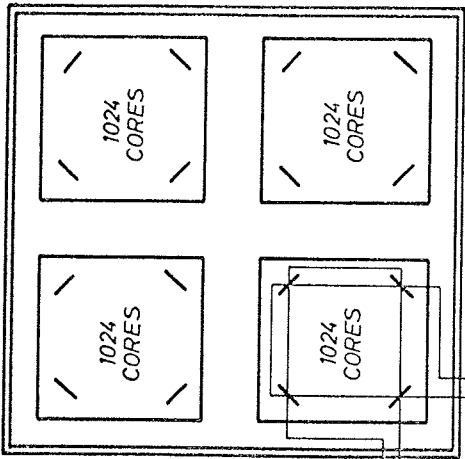


## 5. Core Store

-----

The timing-circuits for the core-store are controlled by means of 6 flip-flops (card 1022) one for each type of core-store functions, as seen from the microprograms. These signals are coded together to a central start-signal: Start CS, which will originate a number of pulses spaced in time ( RT, WT, Strobe prim. etc.) The timing is shown in the timing-schedule.

As seen from the CS-Decoding-System diagram ( Vol-1), the 10-bits register AD1 is divided in 4 groups: 2 groups controlling the y-wires and 2 groups controlling the x-wires. The cards 1002-1 are decoding the 3-bits-groups. The signals from the decoding-cards are wired to the matrix-drivers ( cards 1001-1 ), where the final decoding is found. The matrix-drivers gates the drive-current ( cards 1023 ) to the selected wire.



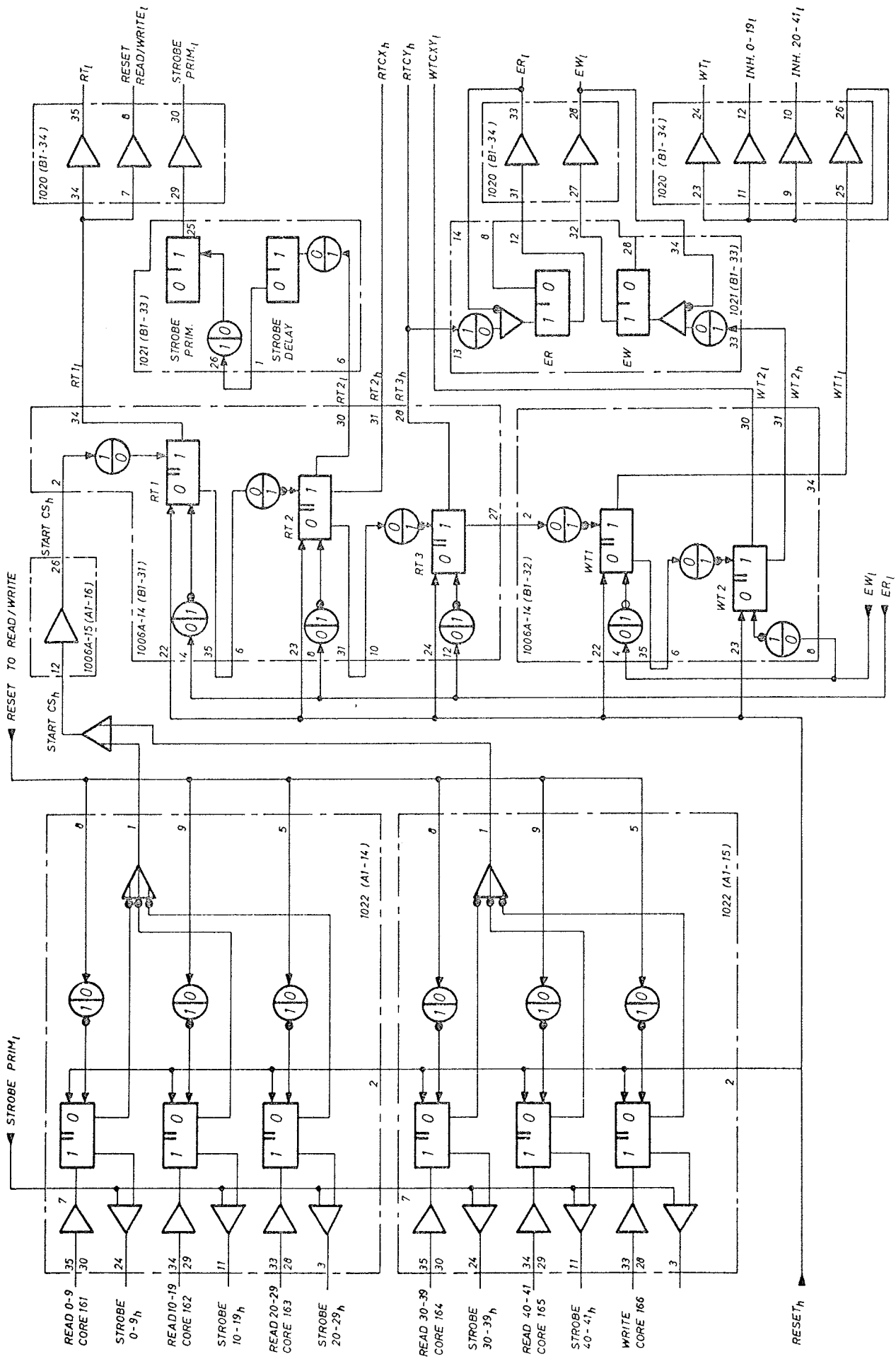
Unit	GIER 6	Designed	K.H.
		Approved	
		Checked	1. 2. 62.
		Last Revision	

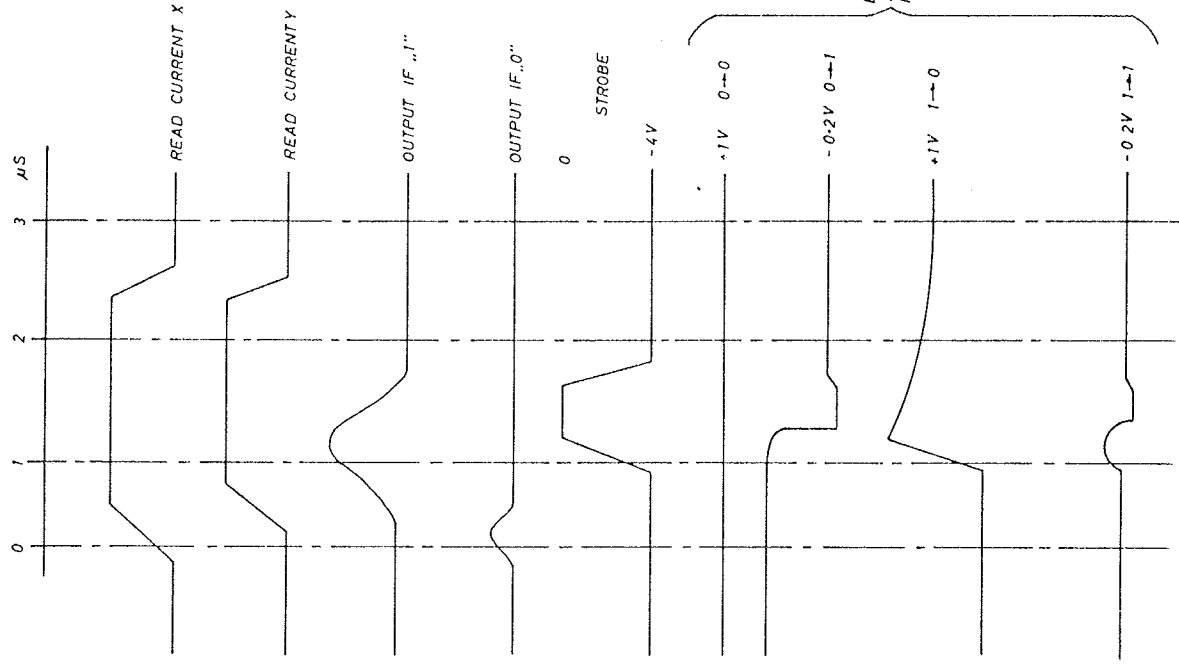
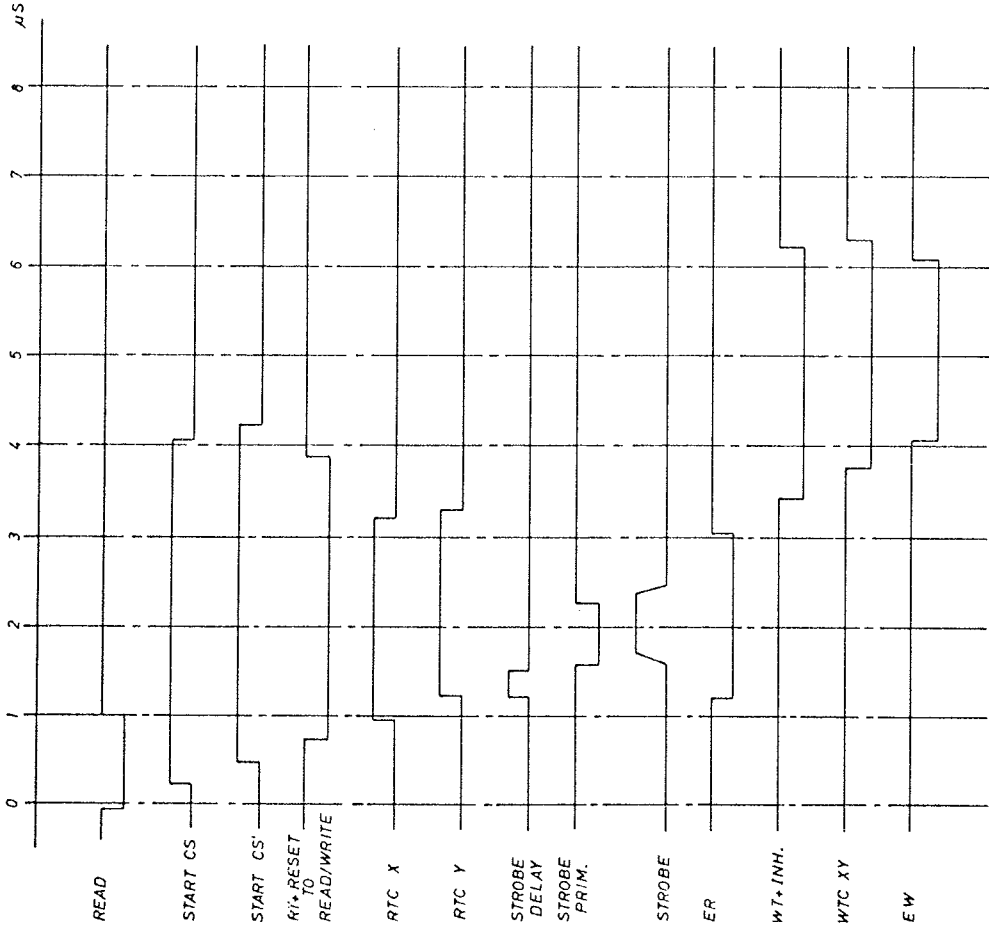
CORE STORE.	

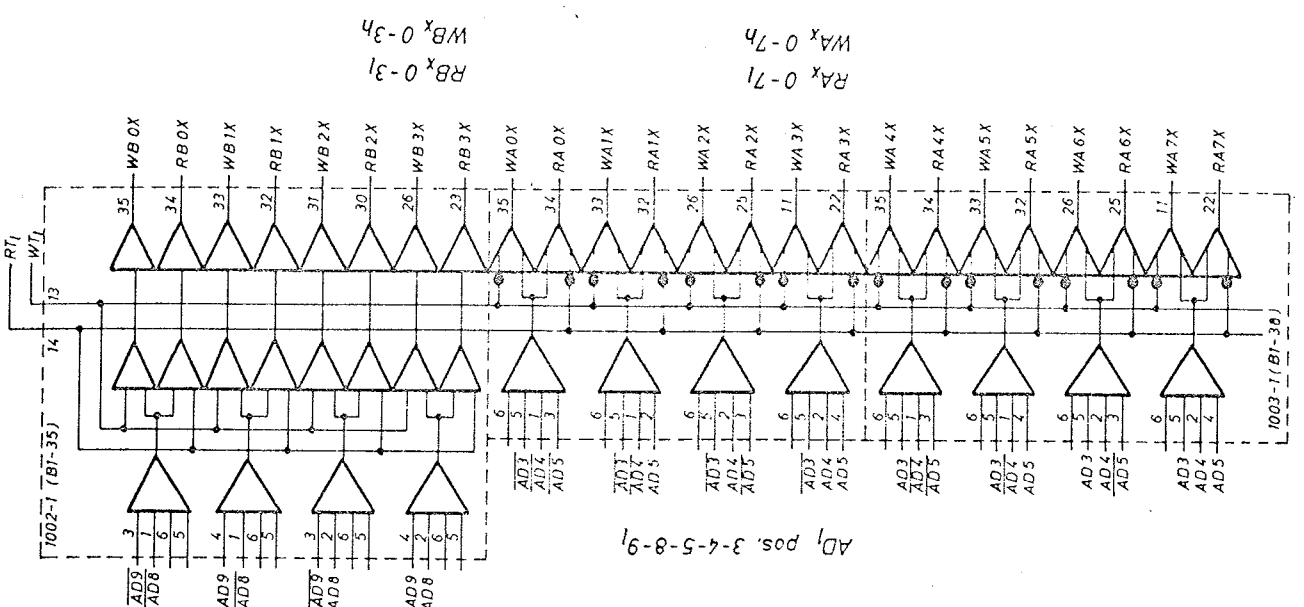
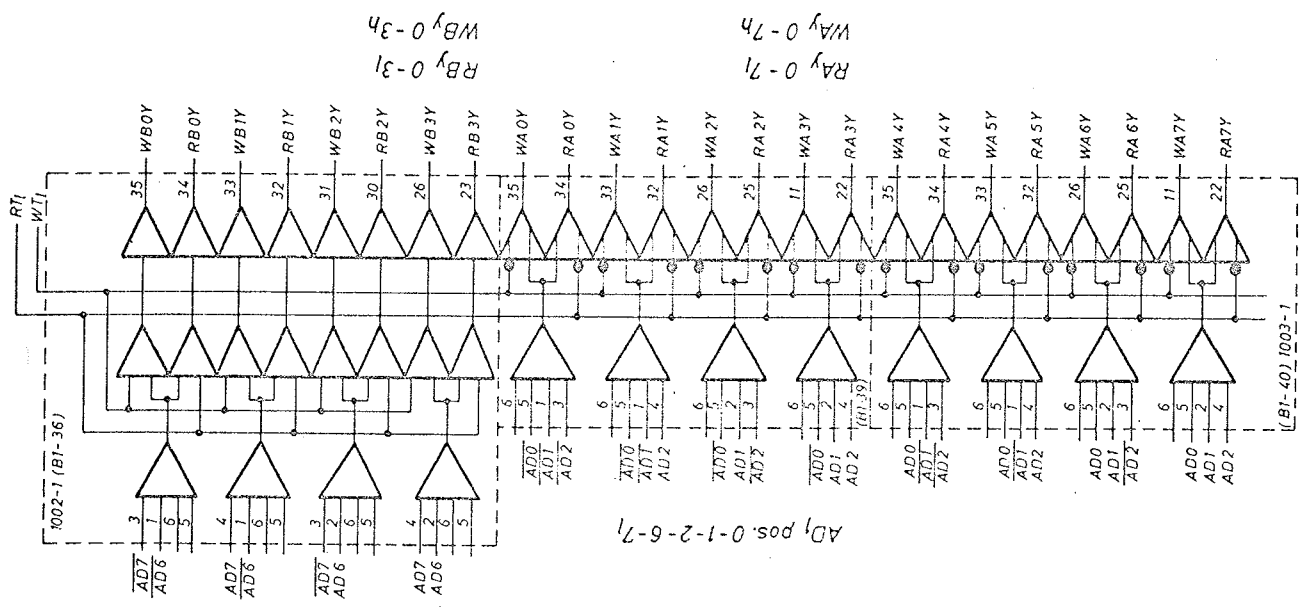
Drawing No	1
Drawn by	G.F. 10. 2. 67
Checked	F.E. 2. 2. 67
Sheet	1



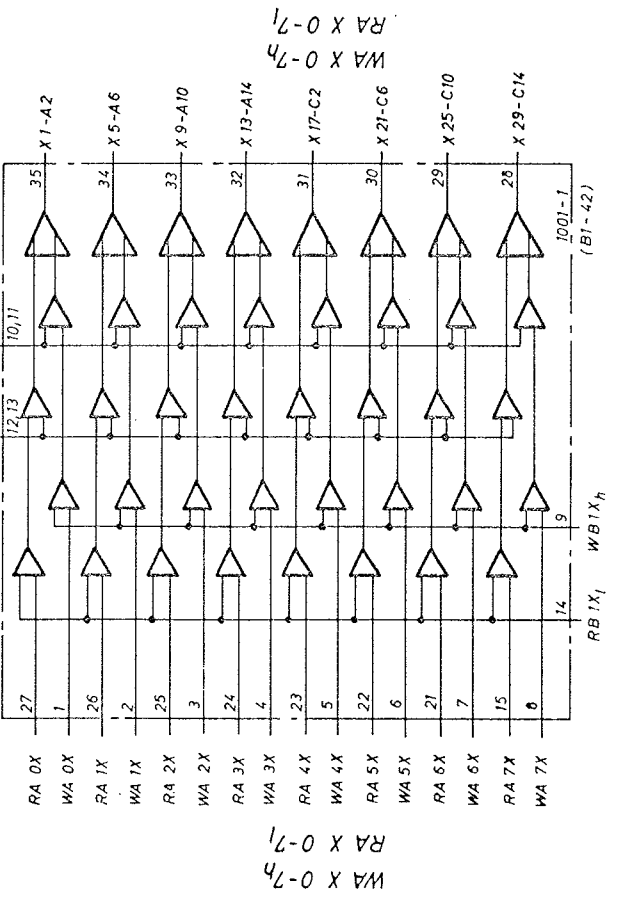
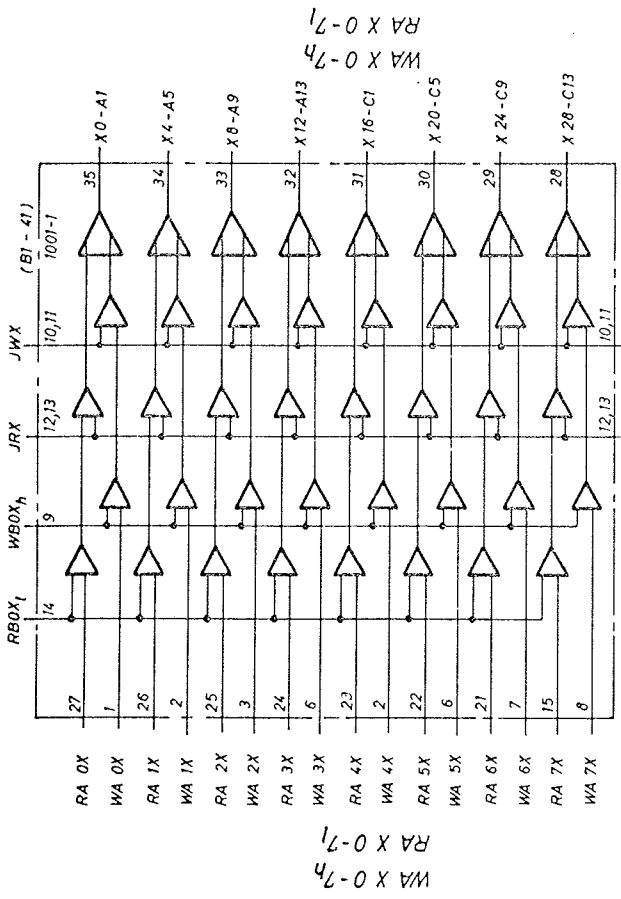
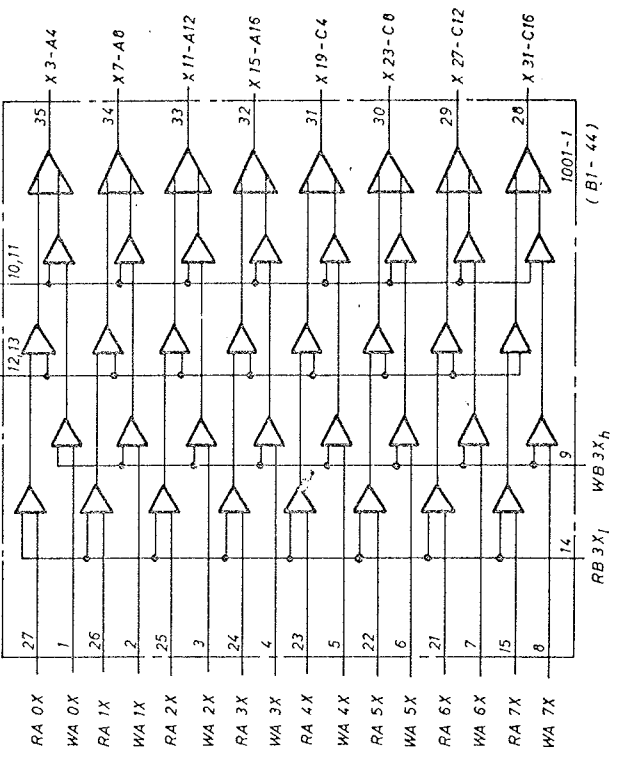
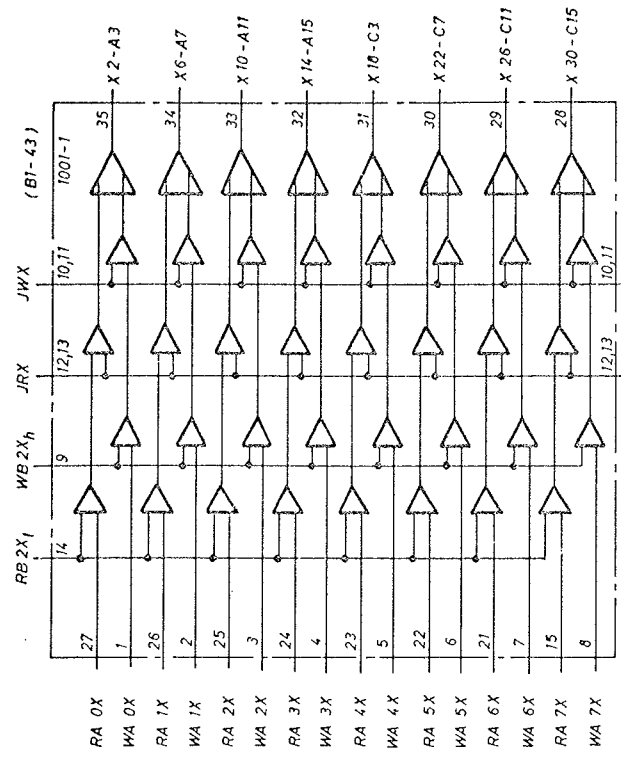
Unit	GIER 6
Designed	H.T.
Approved	
Checked	1.2.62.
Last Revision	
<b>CENTRALEN</b>	
CS TIMING	
Drawing No	Drawn By G.T. 26.1.67.
Checked	F.E. 26-2-67
Sheets	1
Sheet	1



Unit	GIER 6	Designed	K. H.	Drawing No.	67-102
		Approved		Drawn by	G. T. 10 2 67
		Checked	J. 2. 62.	Checked	F. E. 25-2-67
		Unit Position		Sheet	1
				CS TIMING SCHEDULE.	



Unit GIER 6	Designed K. H.	CS DECODING.
 CENTRALEN	Approved	
	Checked 1.2.62.	
	Last Revision	
		Drawing No. 6.7.2.67
		Drawn by G.7.2.67
		Checked F.E. 26-2-67
		I. Sheets Sheet I



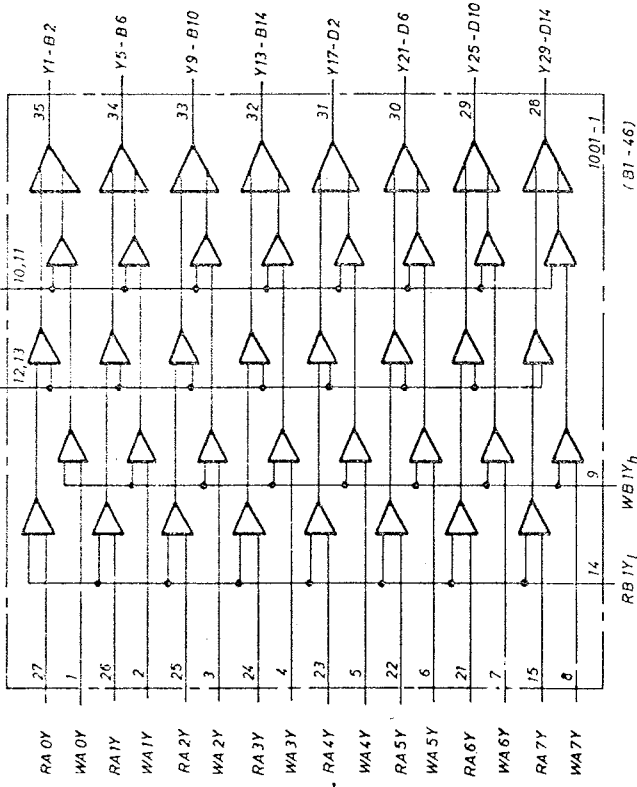
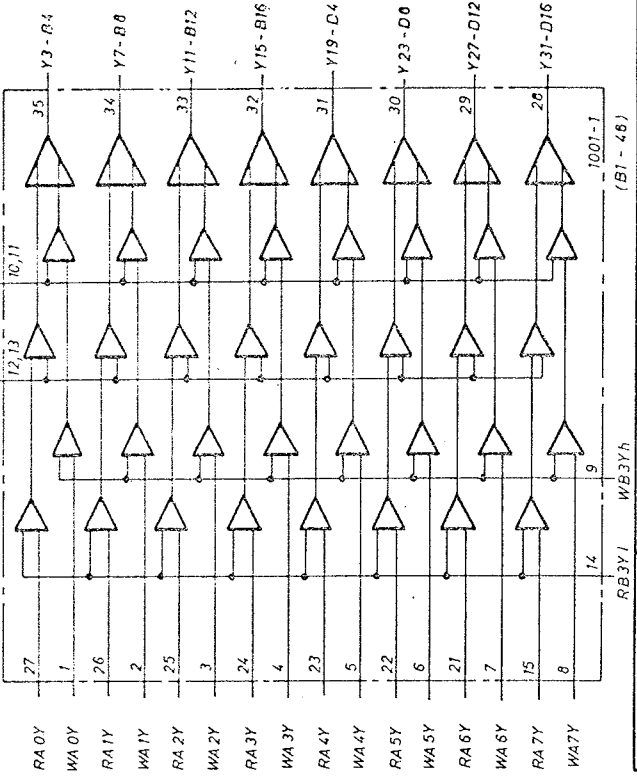
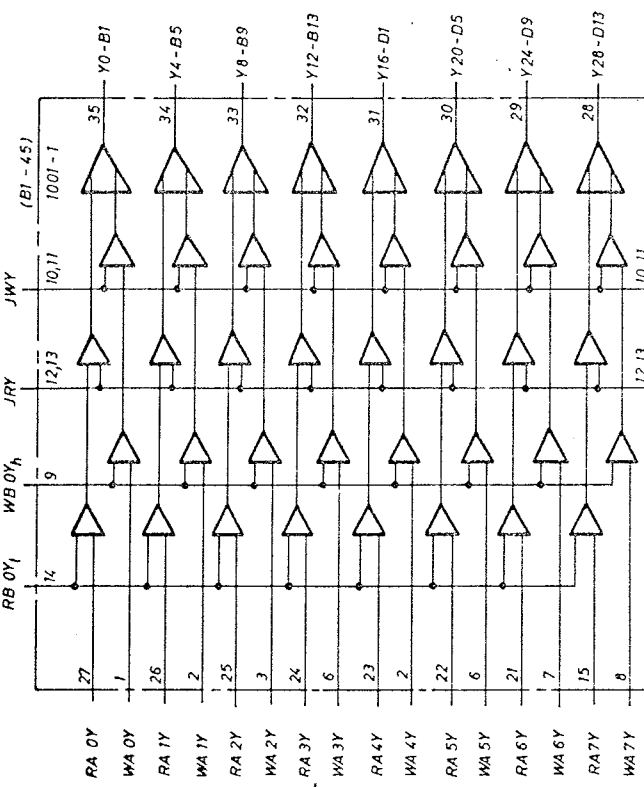
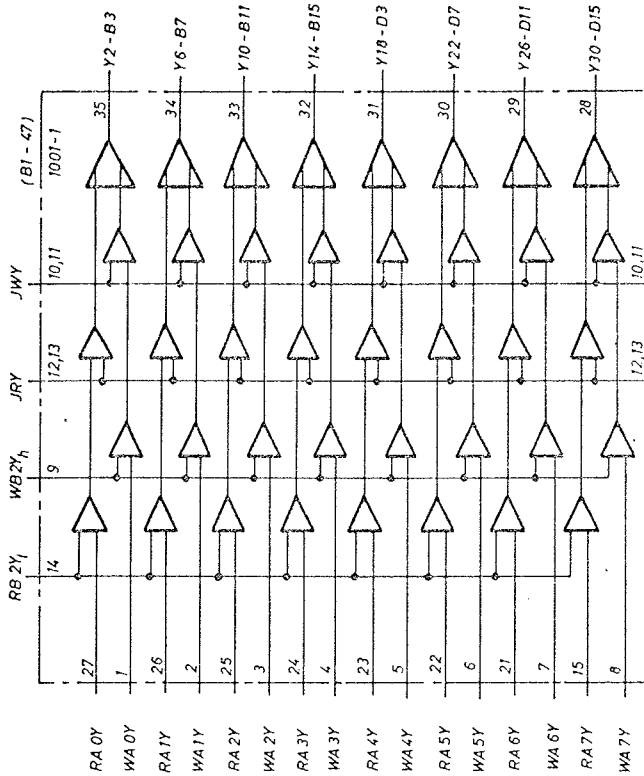
WA X 0-7h  
RA X 0-7l

WA X 0-7h  
RA X 0-7l

WA X 0-7h  
RA X 0-7l

WA X 0-7h  
RA X 0-7l

Unit GIER 6	 CENTRALEN	Designed K.H.	CS MATRIX DRIVER X.	Drawing No. 2.1.67.
		Approved		
		Checked 1.2.62.		Sheet 1
		Last Revision		



WAY 0-7h  
RAY 0-7l

WAY 0-7h  
RAY 0-7l

WAY 0-7h  
RAY 0-7l

WAY 0-7h  
RAY 0-7l

Unit: GIER 6

Designed: K. H.

Approved:

Checked: 1.2.62.

Last Revision:

CS MATRIX DRIVER Y.

Drawing No. 6.7.4.2.67

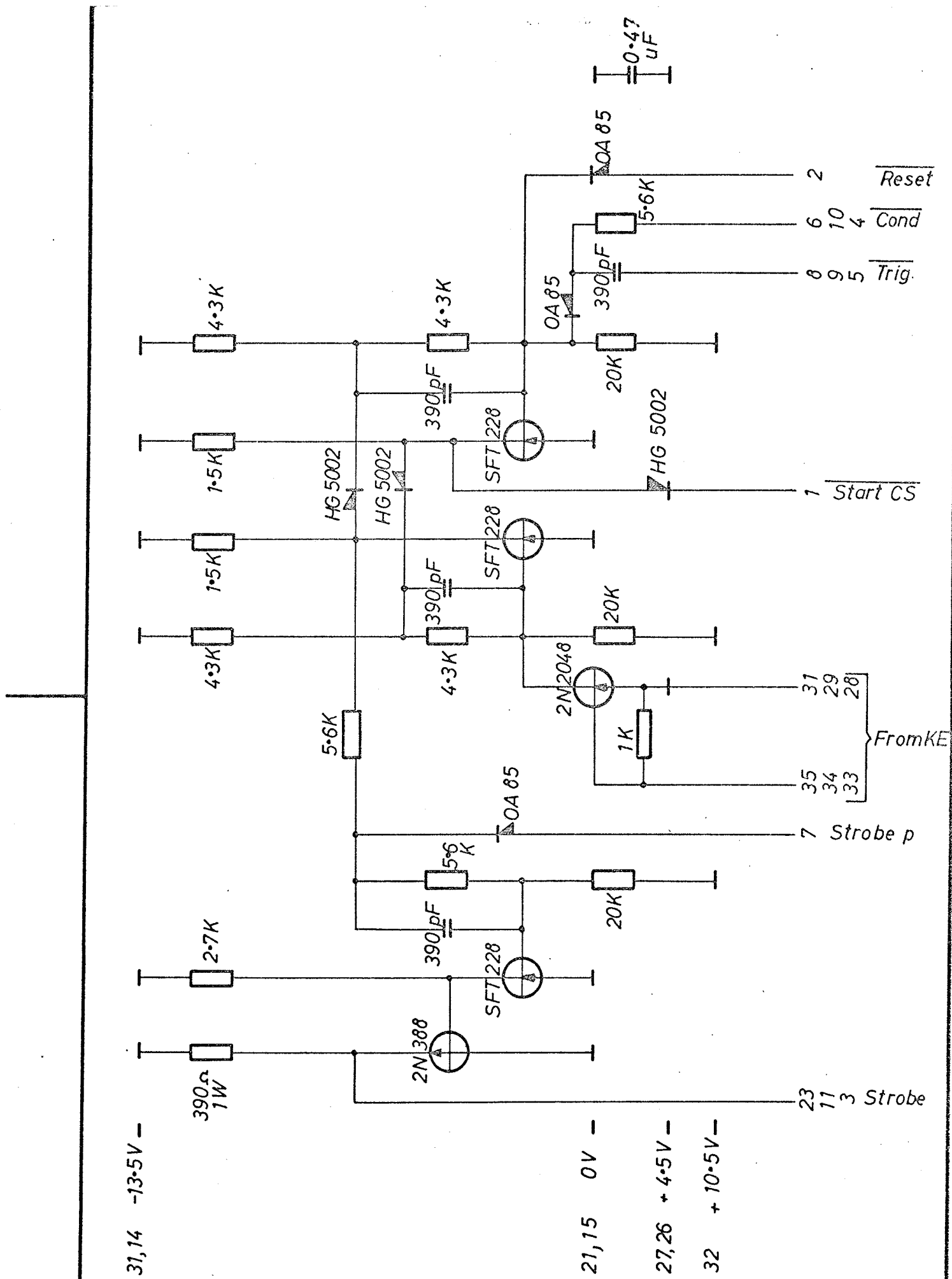
Drawn by: 6.7.4.2.67

Checked: F.E.6-2-67

J. Sheets

Sheet 1

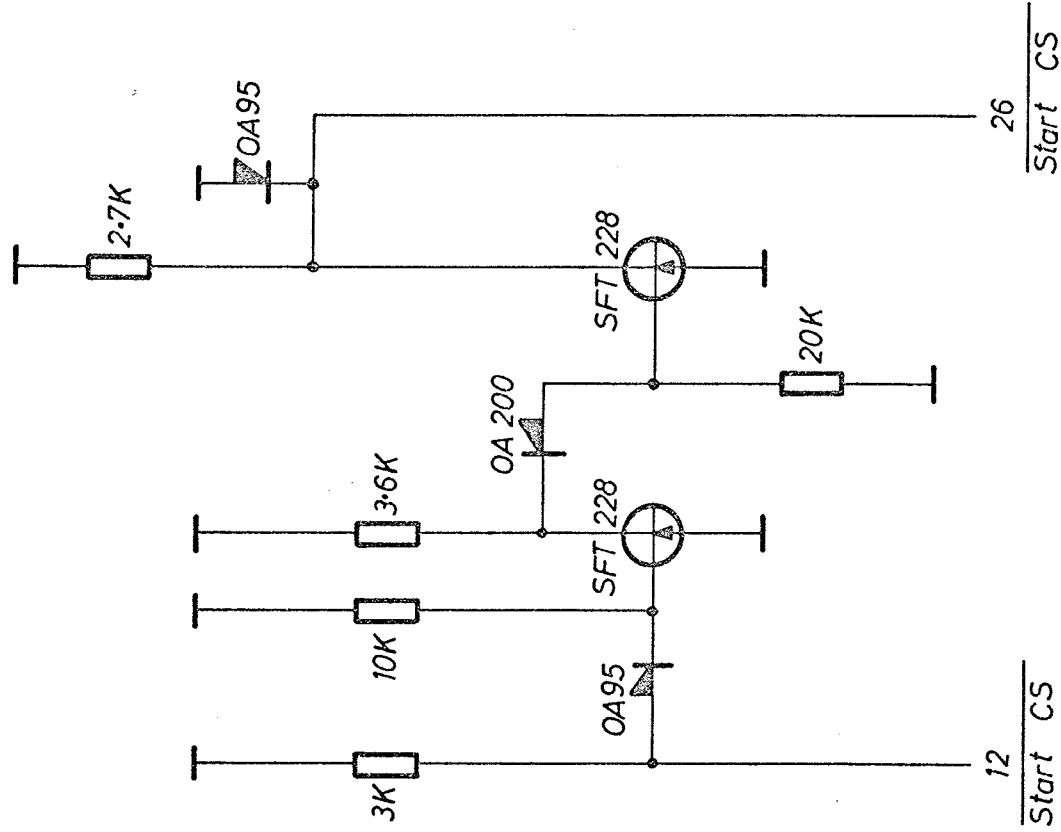
CENTRALEN



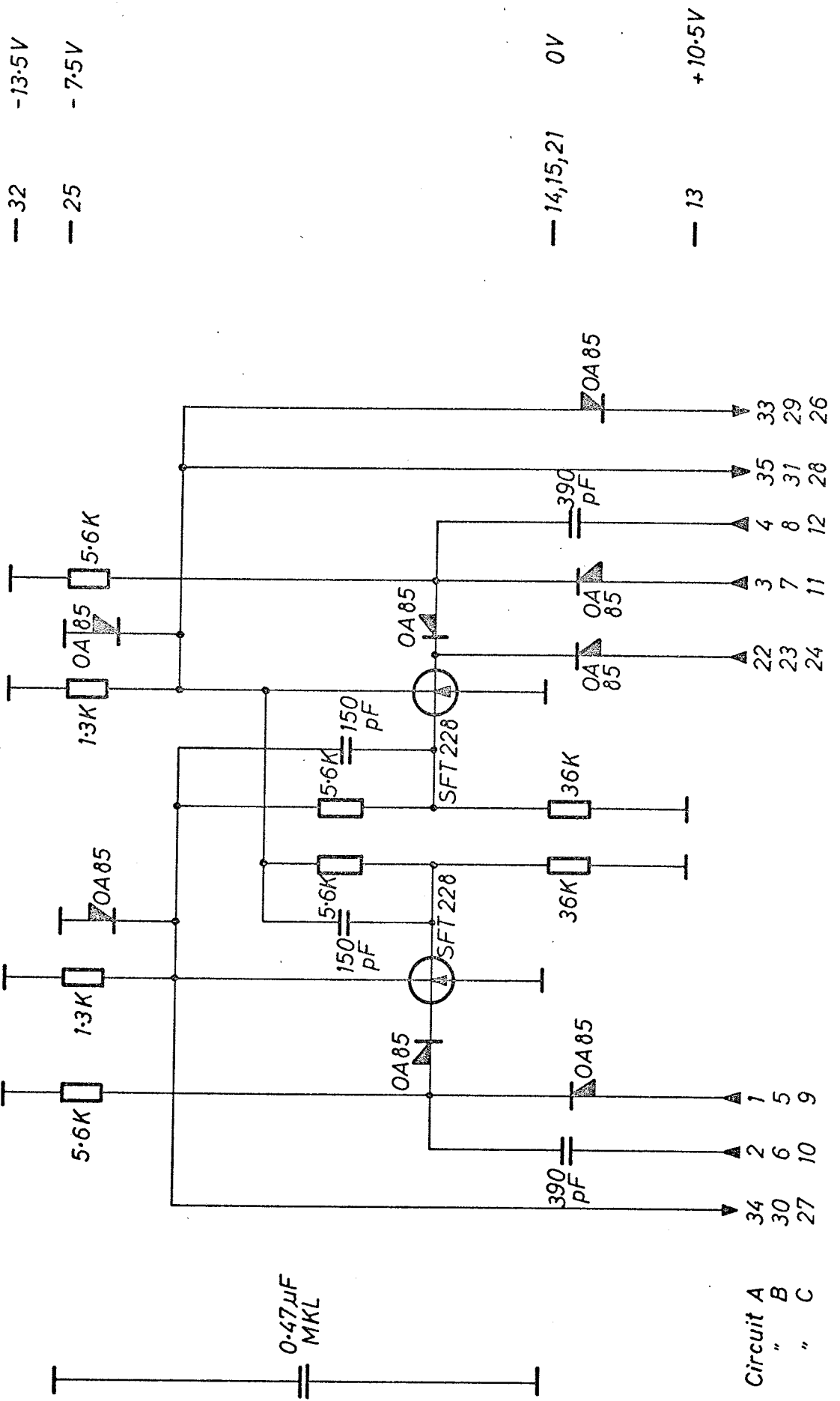
Unit: GIER 6  CENTRALEN	Designed H.I.	<b>STROBE-SELECTOR</b>	Drawing No	
	Approved		Drawn by B.R. 5. 10. 66	
	Checked 1. 2. 62.		Checked F.E. 22-2-67	
	Last Revision		1 Sheets	Sheet 1
			A1-14-15	1022



--- -31.5V 32  
 --- -13.5V 25  
 --- -1.5V 15  
 --- +10.5V 13

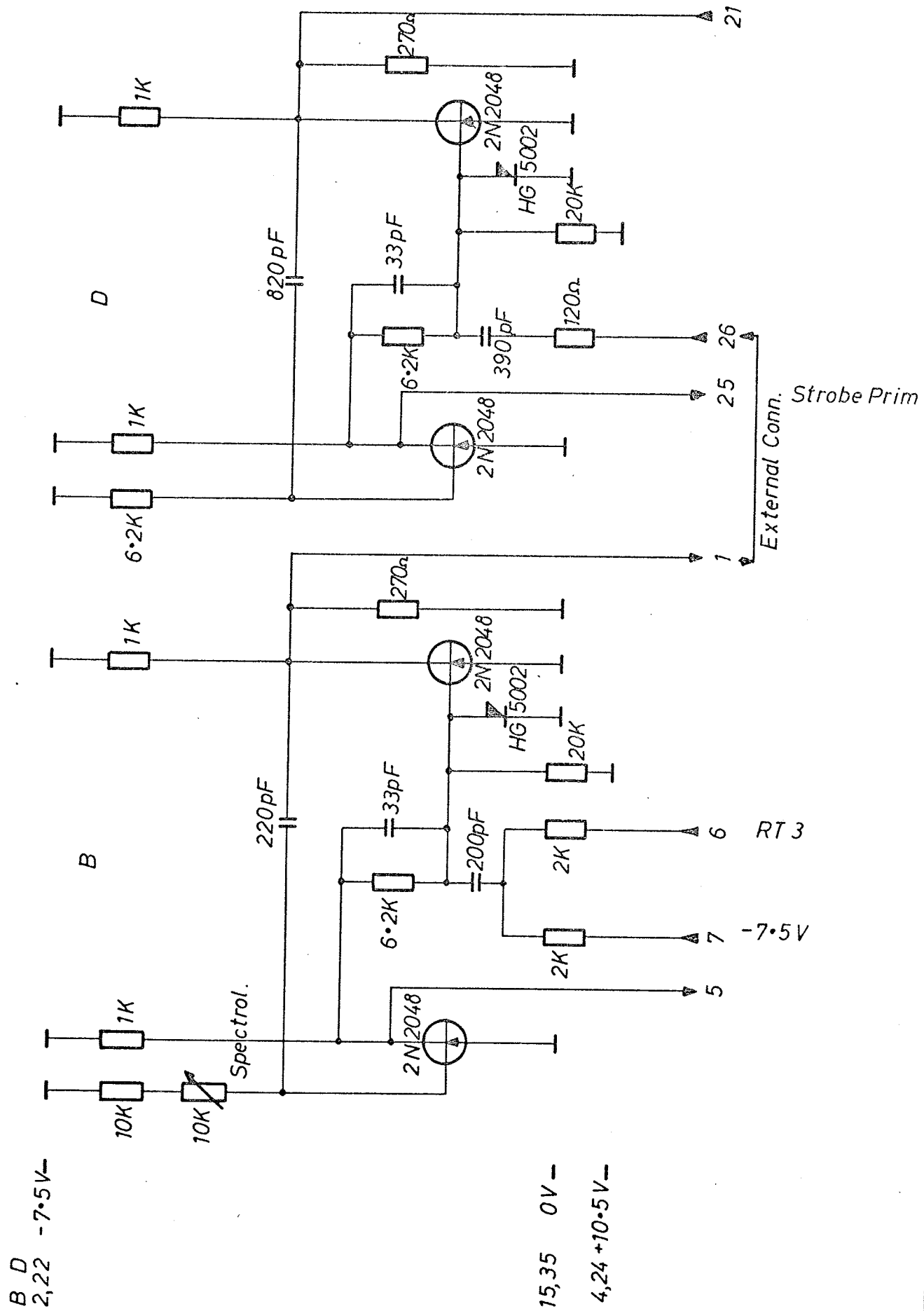



Unit: GIER 6 	Designed H. I.	DELAY	Drawing No	
	Approved		Drawn by G. T. 29.9.66	
	Checked 1.2.62.		Checked F.E. 22-2-67	
	Last Revision		1 Sheets	Sheet 1
			A1-16	1006A-15



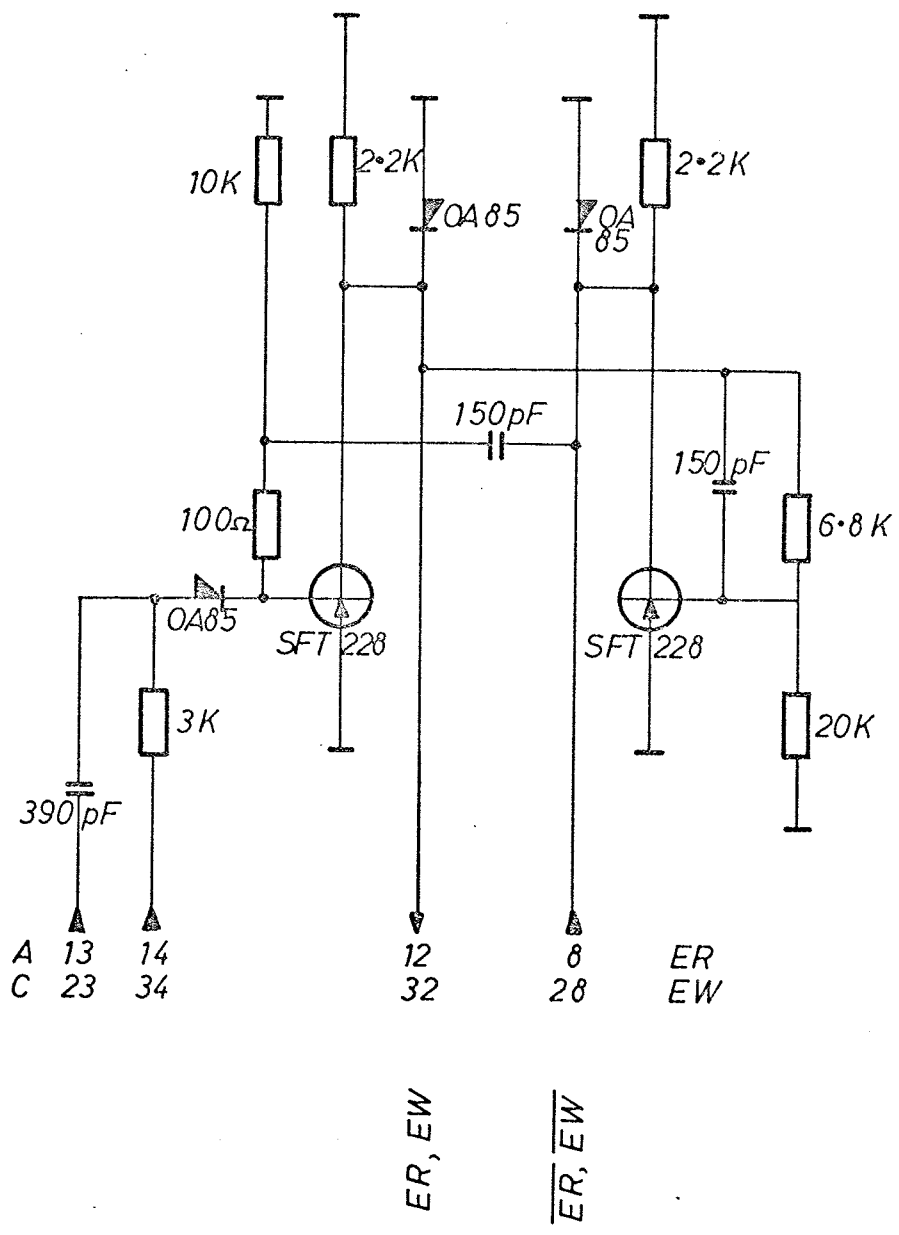
Circuit A  
" B  
" C

	Unit: GIER 6	Designed H. I.	Drawing No	
	Approved	Checked 1. 2. 62.	Drawn by G. T. 25. 7. 66	
	Last Revision		Checked F. E. 22-2-67	1 Sheets
				Sheet 1
CS TIMING FL. FL.			BI-31-32	1006A-14



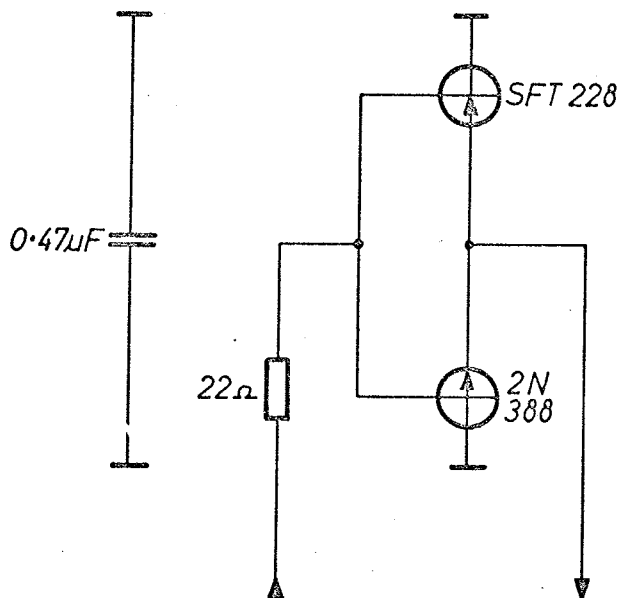
Unit: GIER 6	Designed H. I.	STROBE DELAY + STROBE PRIM.		Drawing No
	Approved			Drawn by B. R. 7. 10. 66
	Checked 1. 2. 62.			Checked F. E.
	Last Revision			2 Sheets    Sheet 1 B1-33        1021

A C  
 11, 31 -13.5 V -  
 10, 30 -7.5 V -



15,35 0V -  
 9,29 +10.5V -

Unit: GIER 6	Designed H. I.	<b>END READ +          END WRITE</b>	Drawing No	
	Approved		Drawn by B. R. 7. 10. 66	
	Checked 1. 2. 62.		Checked F. E.	
	Last Revision		2 Sheets	Sheet 2
		B1-33	1021	



- - 7.5V 14,22

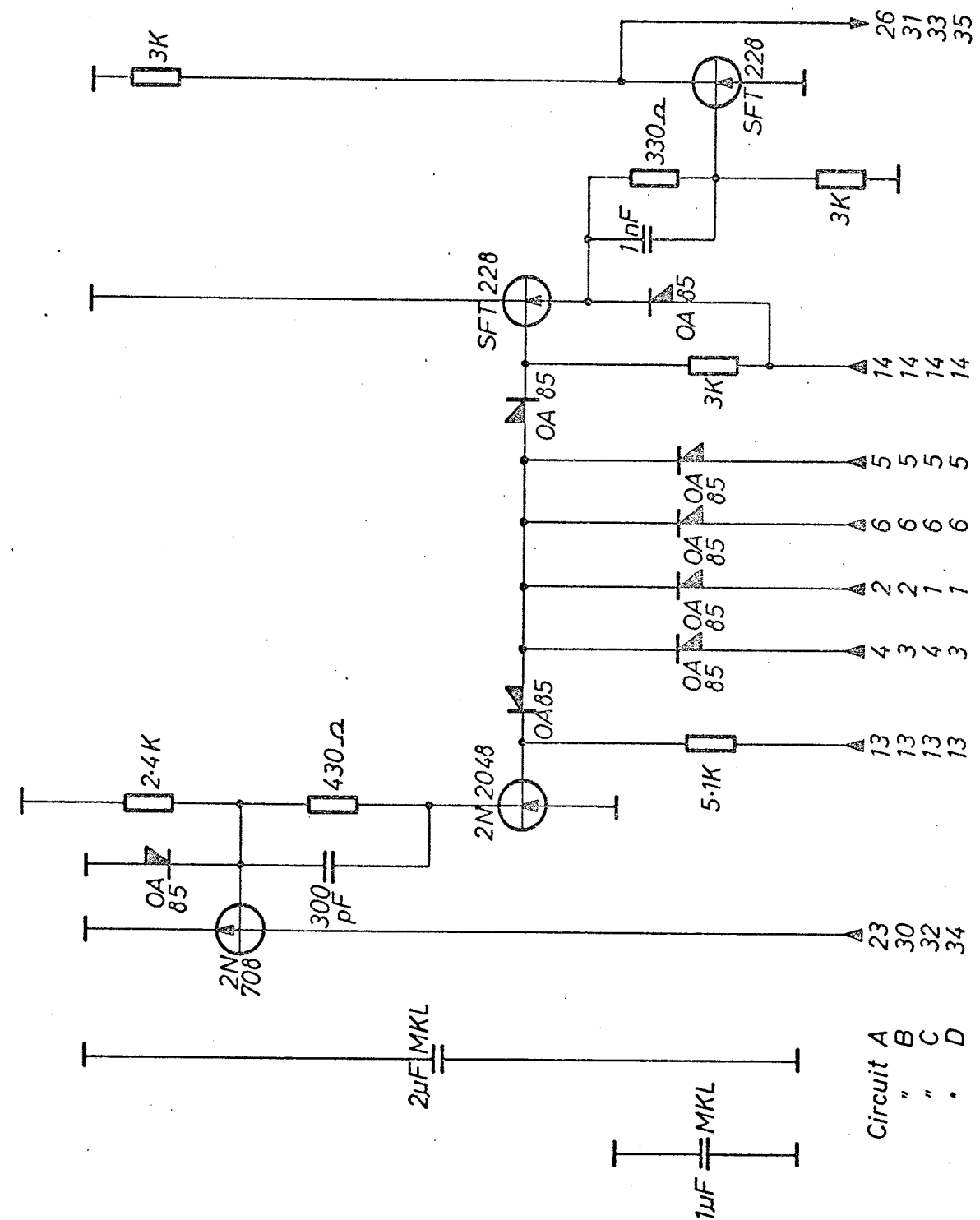
- 0V 15,21,32

Circuit	A	11	12
"	B	9	10
"	C	7	8
"	D	5	6
"	E	3	4
"	F	1	2
"	G	23	24
"	H	25	26
"	I	27	28
"	J	29	30
"	K	31	33
"	L	34	35

Unit <b>GIER 6</b>	Designed <i>H.I.</i>	<b>AMPLIFIERS</b>	Drawing No	
<b>REGNE CENTRALEN</b>	Approved		Drawn by <i>G.T. 30.6.66</i>	
	Checked <i>1.2.62.</i>		Checked <i>FE 22-2-67</i>	
	Last Revision		<u>1</u> Sheets	Sheet <u>1</u>
		<i>B1-34</i>	<i>1020</i>	

- -13.5V 27  
 - -7.5V 12,24,25,28

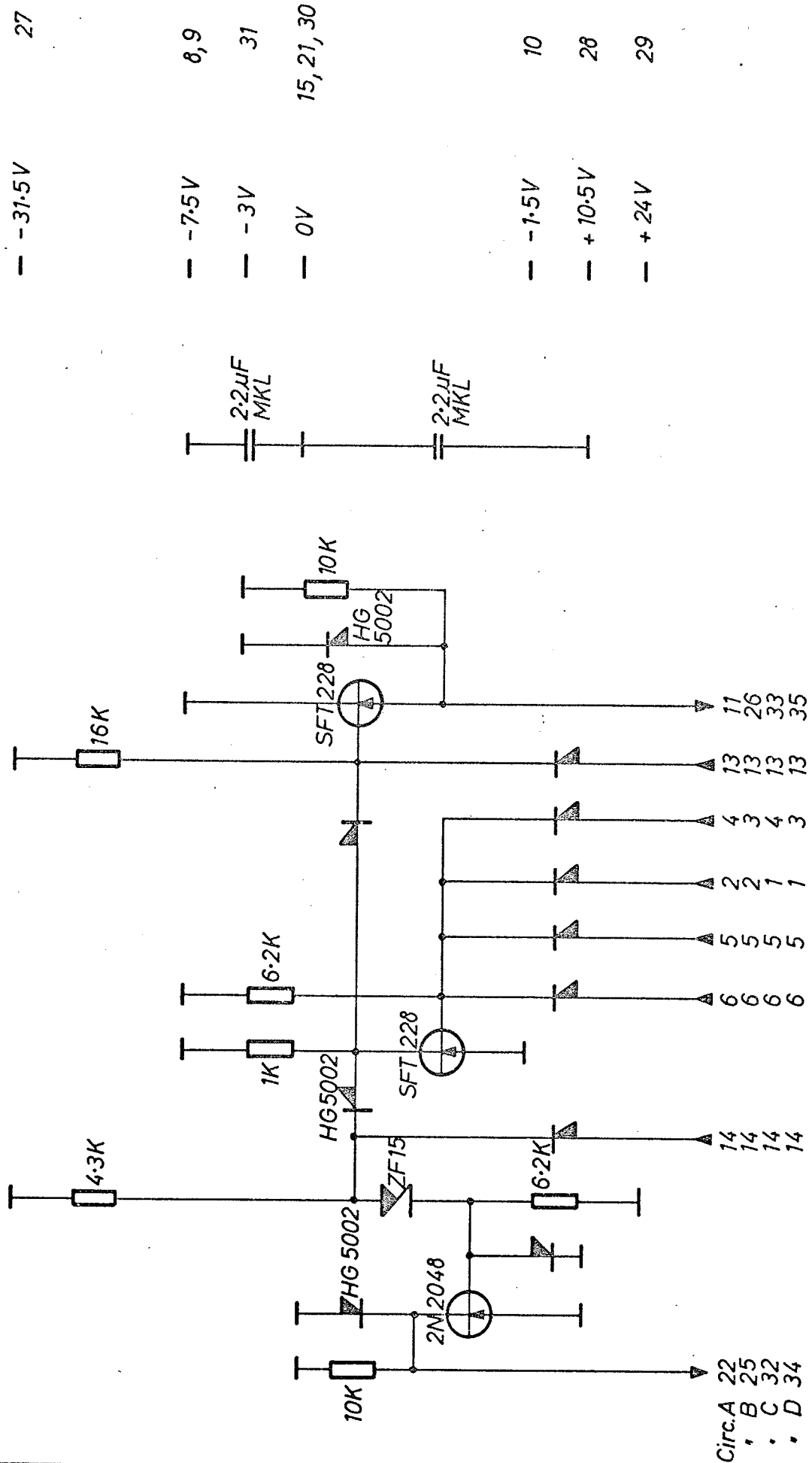
- -1.5V 10  
 - 0V 15,21  
 - +10.5V 29



Circuit A  
 " B  
 " C  
 " D

Unit: GIER 6 	Designed H.L	$\overline{WB} + RB$	Drawing No	
	Approved		Drawn by G.T. 4.7.66	
	Checked 1.2.62.		Checked F.E. 22-2-67	
	Last Revision		1 Sheets	Sheet 1
			B1-35-36	1002-1

Unmarked Diodes: OA 85



-- -31.5V

27

-- -7.5V

8,9

-- -3V

31

-- 0V

15, 21, 30

2.2uF  
MKL

2.2uF  
MKL

-- -1.5V

10

-- +10.5V

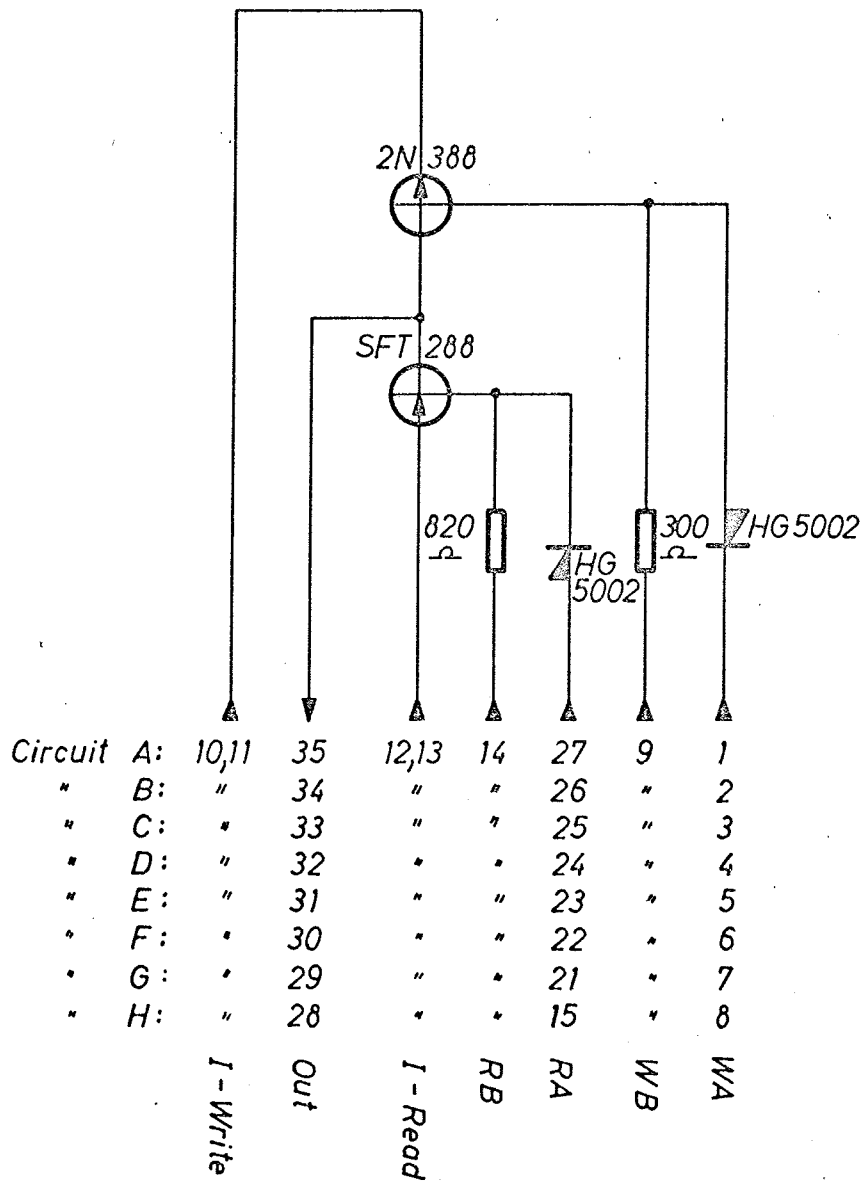
28

-- +24V

29

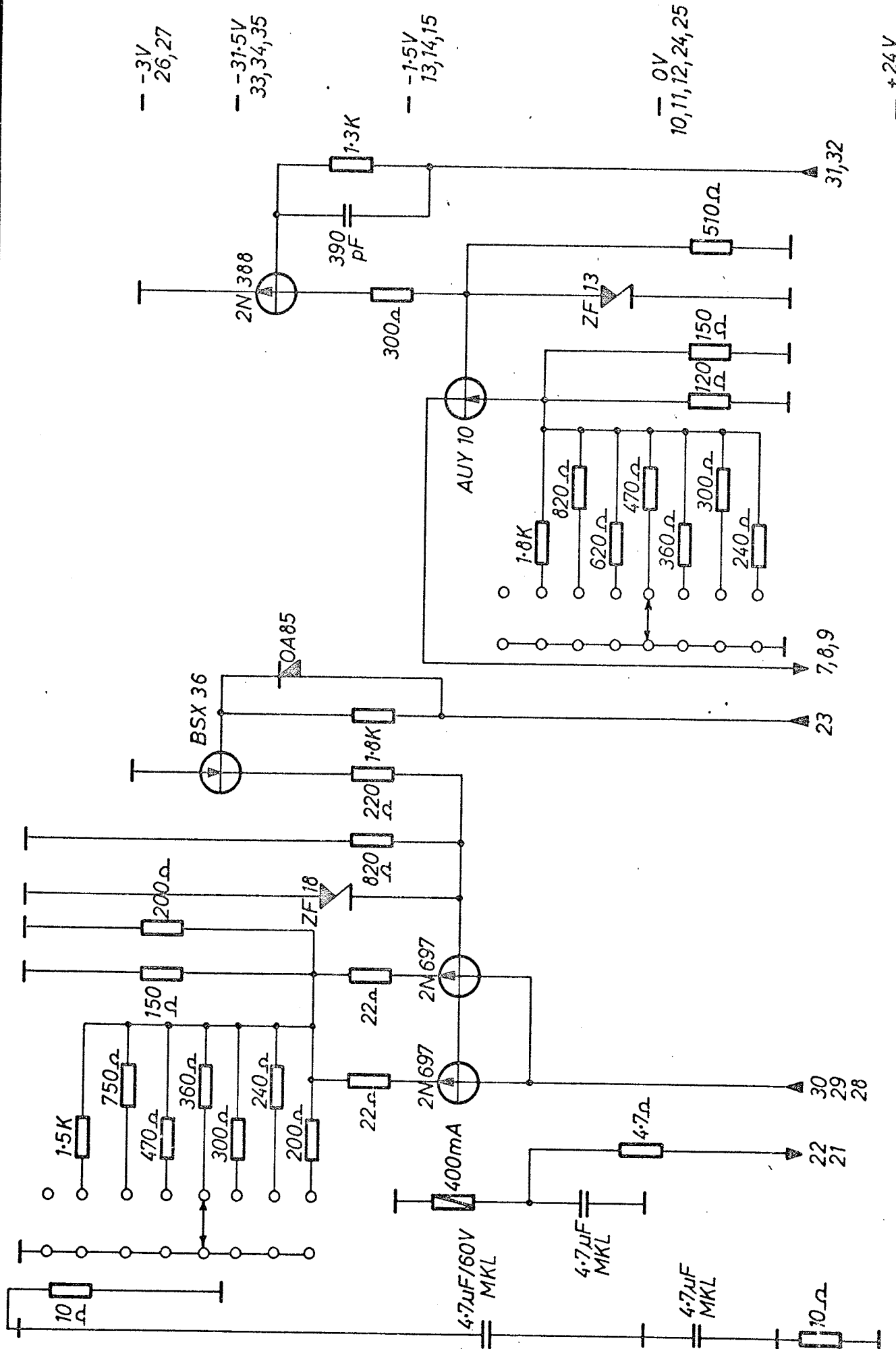
Circ. A 22  
B 25  
C 32  
D 34

	Unit: GIER 6	Designed H. I.	Drawing No	
		Approved	Drawn by G.T. 4.7.66	
		Checked 1.2.62.	Checked F.E. 2-2-67	
		Last Revision	1 Sheets Sheet 1	
WA + RA			B1-37-40	1003-1



Unit: GIER 6	Designed H. I.	MATRIX DRIVER	Drawing No	
	Approved		Drawn by G.T. 25.7.66	
	Checked 1. 2. 62.		Checked F.E 22-2-67	
	Last Revision		1 Sheets	Sheet 1
			B1-41-48	1001-1

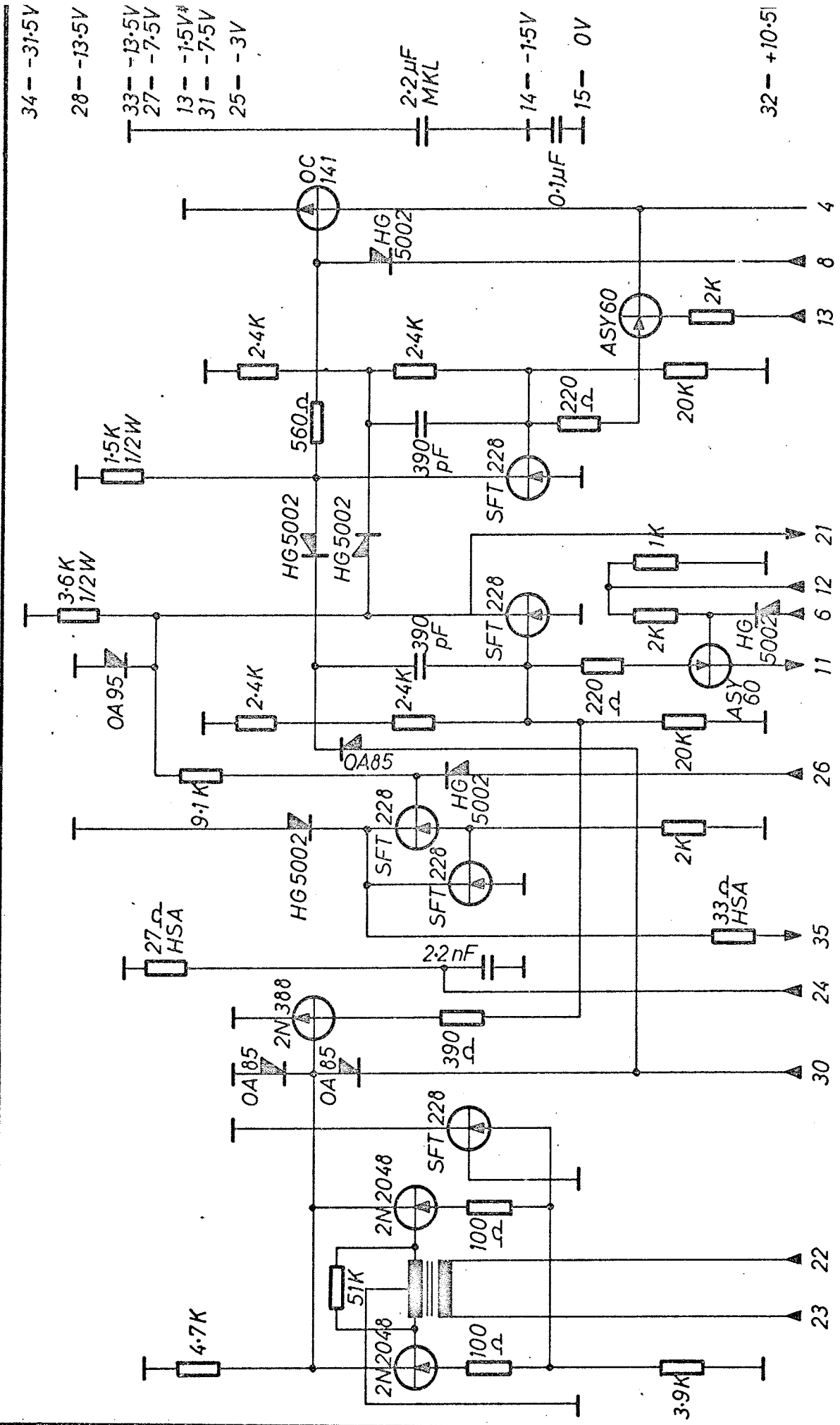




Unit <b>GIER 6</b>	Designed <b>H. I.</b>
	Approved
	Checked <b>17-4-68 wJ</b>
	Last Revision <b>28-10-67 d. d.</b>

**READ WRITE CURRENT**

Drawing No	
Drawn by <b>G.T. 1.7.66</b>	
Checked <b>FE 23-2-67</b>	
<b>1</b> Sheets	Sheet <b>1</b>
<b>B1-49-50</b>	<b>1232</b>



Unit: GIER 6 	Designed H.I.	CS READ AMPLIFIER + BUFFER + INH	Drawing No	
	Approved		Drawn by G.T. 25.7.66	
	Checked 1.2.62.		Checked F.E. 23-2-67	
	Last Revision		2 Sheets	Sheet 1
			B2-49-50	1024-1

## 6. Drumstore.

### Main principles of control.

Transfer of information between the drumstore and the core store is divided up in a drum-synchronous serialtransfer between the drum and the buffer-register TI and a GIER-synchronous paralleltransfer between TI and the core store. The drumsynchronous transfer is performed by the drumcircuits simultaneous with the execution of instructions not concerning the drum. The corestorageadress to or from which the drumtransfer shall be done, is also calculated by the drumcircuits simultaneous with normal operations. At the moment when the drumcircuits are ready for the parallel transport the execution of simultaneous operations is interrupted for about 26  $\mu$ S while the paralleltransfer to or from the core store is going on. This is done by a special microprogram which is activated by means of a special set of 8 microaddresses, MAB 1-8, and a special clockpulse, KPB. The time-interval between the 40 interruptions during a channel transfer is about 500  $\mu$ S.

### Channel selection.

Selection of the channel to or from which a transfer is wanted is done by execution of the operation VK by which the channel number modulo 512 is transferred to the register TK. Any channel-transferoperation is normally executed on condition that an existing channel has been selected. If a transfer to a nonexisting channel is executed (no. 320-511 or 832 - 1023) the drumcircuits will be occupied the normal time (about 21 ms). After this the situation is unchanged in the store. If a transfer from a nonexisting channel is executed the corresponding cells in the core-store will be set to zero and the computer will not finish the LK-operation untill the pushbutton "annul. of the TR-paritet" is activated.

The 320 channels in the drumstore are mechanically represented by 320 read-writeheads, which are individually wired to 40 plugs (A2-A11, B2-B11, C2-C11 and D2-D11), but electrically they are separated in 20 groups each containing 16 channels. The read-write heads are decoded from the TK-register. Bits 1-5 (numbers according to the buswire no.) are decoded to an X-value: 1 out of 20 possible (12 are not used) and bits 6-9 are decoded to an Y-value: 1 out of 16 possible. These X- and

Y-values are defining the selected track, and this last part of the decoding is performed by means of the diodes placed in the plugs sitting upon the drum. This part of the decoding is only active during the execution of the operations LK or SK.

Changing of the contents of the TK-register during a channel-transfer is impossible due to the signal "block Gm TK" from card 205-4.

#### Locking for writing.

Some of the drumtracks may be locked for writing. The locking can be done in two levels. The first level comprises normally track no. 0-31.. This lock is manually operated and is accessible to the user. It is possible to expand this locking to comprise up to 5 groups of 16 tracks by connecting the X-group in the decoding to one of the spare-inputs upon card 200-4. For instance a connection between the pins C3-20-15 and C5-21-12 will expand the lockingarea to tracks 0-47.

The second level of locking comprises normally only track no. 0, and is not accessible to the user. Unlocking can be done by removing the short circuit pin upon card 200-4. This locking may also be expanded so that it comprises up to 16 tracks, no. 0-15, by connecting the appropriate Y-group in the decoding to C3-24-1N.

The execution of an SK-instruction where the selected track has been locked will be finished without writing during 9  $\mu$ sec. for the basic-operation plus the normal executiontime for the modifications.

#### Clocktracks.

The drum is equipped with 3 clocktracks and associated readamplifiers. Clocktrack no. I contains one mark for detecting word no. 0, track no. II a marking for each of the 40 words, and track III a marking for each bit in all words.

Transfers to and from the drum.

The microprograms for the drum instructions LK and SK begin with the microoperation "Gm sæt TL" which will read the value of the signal TR" into the flip-flop TL upon card no. 266. If TL becomes 1 it means that the drumcircuits are unoccupied (TL means "Tromle ledig" or "drum ready"), and the microprogram will proceed with starting the drumcircuits by means of the microoperation "Gm start TR", which will set one of the flip-flops upon the card 264 dependent on whether it is a write-or a read-operation. This flip-flop will remain set during the drumaccess and will select the correct microprogram to be executed at the interruptions when words are transferred to or from the core store. The synchronizing signal to the start-stop circuits SL ("skriv-læs" means ready for write to or read from core memory) is derived from this flip-flop and clockpulse II. The delay of 12  $\mu$ sec. upon card 261 is necessary due to the parity check (see later).

As the drum access is initiated non-synchronous to the drum, the serial transfer of information from the drum to the TI-register may start in the middle of a word, so that the first word transferred to the corestore is not correct. The drumaccess is therefore always transferring 41 words so that the first word in the access is being transferred 2 times where the last one is always correct. The counting of the number of words transferred is done by means of the counting register TKT. This register is cleared (set to zero) at the beginning of a drumaccess (and when a parityfault occurs in a word). When the contents of TKT is 41, the drumaccess is finished by resetting the flip-flop upon the card 264.

If a SK-instruction is initiated during the word-time of word no. 39, the 41'st counting pulse to TKT will finish the drumaccess just before the second and correct writing of word no. 39 occurs. To prevent this, an extra clearing of TKT is being executed just before word no. 0 is written the first time in the above mentioned situation.

Addresscalculation.

The resultant address of a LK or SK-instruction is transferred to the TBA-register. The TOT-register always contains the word no. of the word that has just passed the read/writehead. The sum of TBA and TOT is calculated by means of the drumadder. This number is the corememoryaddress to where the word from the drum shall be transferred at the storing-interruption during execution of a LK-instruction. In case of the SK-instruction this sum from the drumadder has to be increased by one to give the address in the corememory wherefrom to fetch the word that shall be written upon the drum. This is done by setting the carryinput of the lowest position of the adder to the value 1. When word no. 0 is to be transferred the contents of TOT must be -1 this value is set into TOT during wordtime 39.

The contents of register TBA cannot be seen from the display in the operators panel, but by means of a DM 160 lampcard placed in A1-23.

During the B-microprograms which executes the transfers between the corestore and the drumstore the output from the drumadder is transferred to the TA-register and the inputs to the addressdecoding of the core store are connected to TA instead of AD1. This shift of them decoding is done by means of a flip-flop upon the card 266.

Wordtransfer.

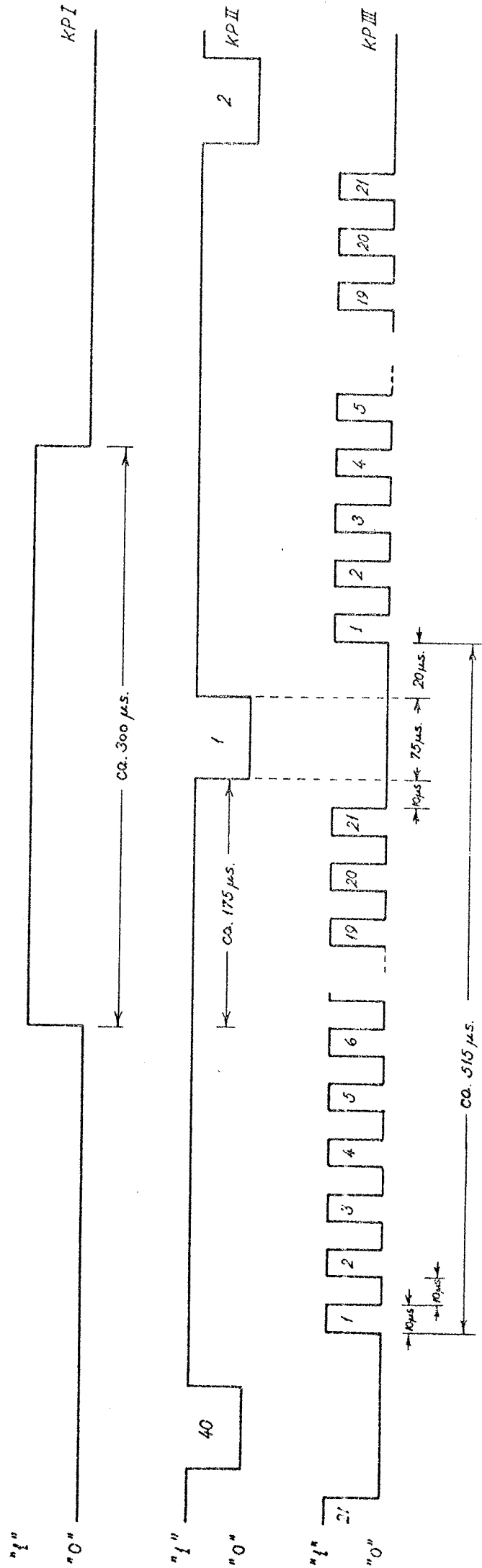
The serialtransfer between the drum and the bufferregister TI is performed one bit at a time, the contents of TI being shifted one position to the left for every bittime. During writing upon the drum the signal for controlling the write amplifier is taken from TI-position 0 while during reading the output from the readamplifier is connected to TI-position 41. The shiftpulse for TI is derived from clockpulse III and II, the shifts being performed at the times where the value of clockpulse III is changing from 1 to 0 and from 0 to 1 and where clockpulse II changes from 1 to 0. The principle of writing that is used is called NRZ, non-return-to-zero, because the current in the writehead is always running one or the other way. It is shifting at the beginning of words, at every bit = 1 in the words and after the finish of a word if the number of 1's was even.

The signal from the read amplifier is setting a flip-flop which again steers another flip-flop strobed by a signal derived from the clock-pulses. After this the shifting begins in a flip-flop called TI-pos.42. The last bit that is read in a word-time is the paritybit, which remains in TI-pos. 42 during the succeeding paralleltransfer to the corestore. At the shifting-in into TI of the next word from the drum, the proceeding word is shifted out through TI-0 and is lost.

#### Paritycontrol.

The paritycheck at reading claims that every word including paritybit contains an odd number of ones. The "ones" are counted by means of a flip-flop upon card 263. At the end of a word-time this flip-flop must contain a 0 indicating correct parity, at the beginning it contains 1. If this is not so the signal SL will via the logik cause a resetting of register TKT and the transfer will start from the beginning. If the fault continues the tracktransfer will never finish, unless the operator cancel the paritycheck by means of the switch "Annul.tr.parity" (The red lamp will light when the check is inactive). Operating the "Reset"-button will as usual stop every action.

Simultaneously with the resetting of TKT because of parityfault another flip-flop upon the card 263 will be set, and later it will be reset when TKT = 41. This flip-flop will light the lamp at the operatorspanel called "Tromlefejl" indicating error in the drum. Because of the non-synchronous start of reading from the drum, the first word read is often with parityfault. To prevent the lamp from lighting for this reason, there is a delay between the flip-flop and the lamp.




GIER drum circuits  
Clockpulses I, II and III



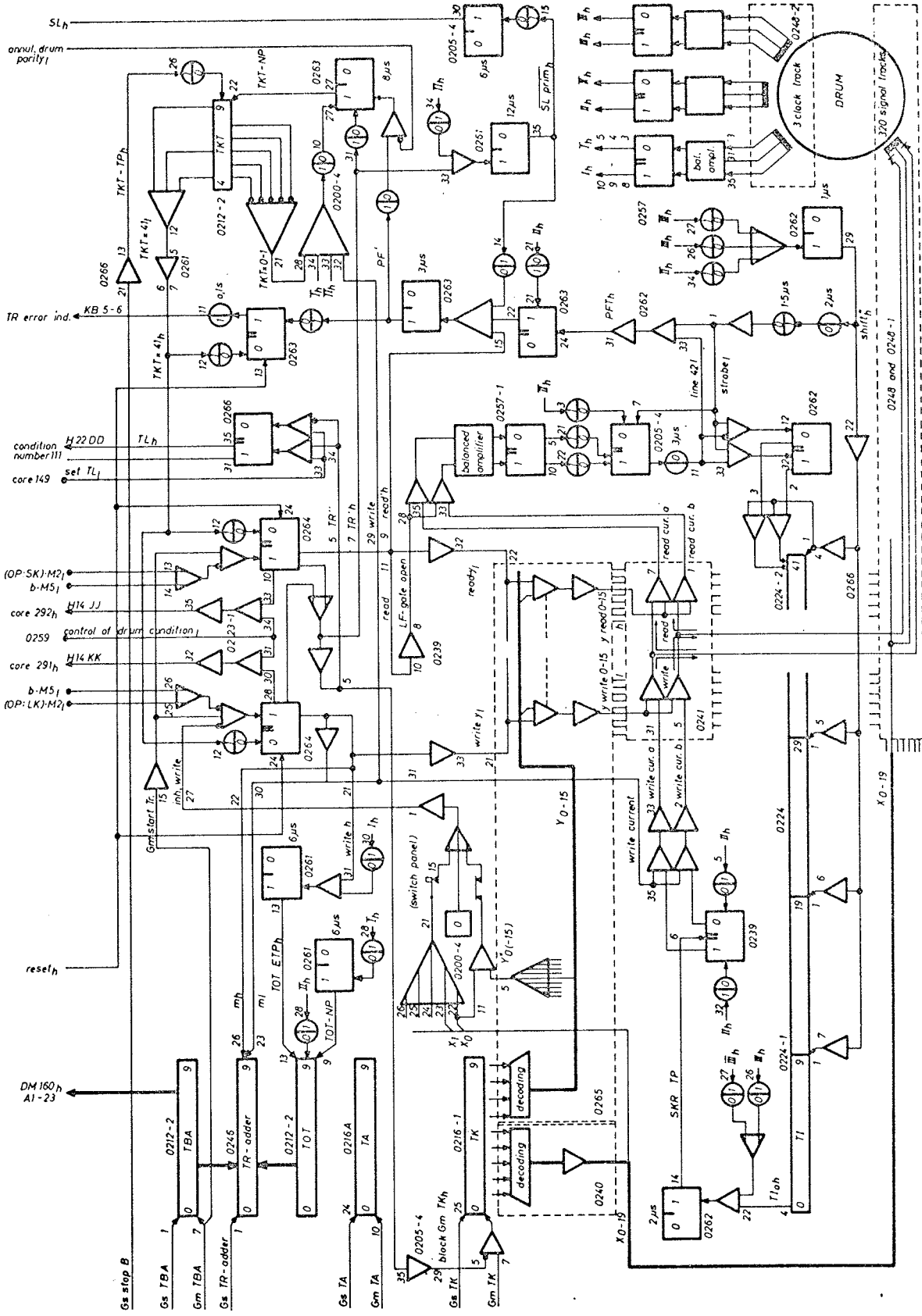
Operation no 52

Operation LK

MA	Gs	Gm	CS	Decoding	No	Jump
1		set TI, ob+c step				12
12	AD1	TBA start Drum		Drum ready TL	9a	13
12	Dummy			Drum busy TL	9b	1
13	Mode 4					2
1B	AD1	TA				
2B	TR-Add.	AD1				
3B	TI0-41	LI0-41	Write			
4B						
5B	Dummy					
6B						
7B	TA	AD1				
8B	Stop B-pulse	shift B--A				

Unit GIER 6	Designed H.I.	MICROPROGRAMS	Drawing No
	Approved		Drawn by L.L. 21.2.67
	Checked 1.2.62		Checked F.E. 23-2-67
	Last Revision	1 Sheets	Sheet 1

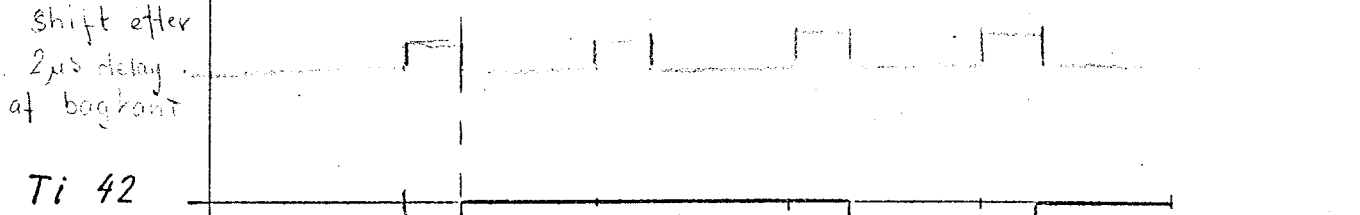
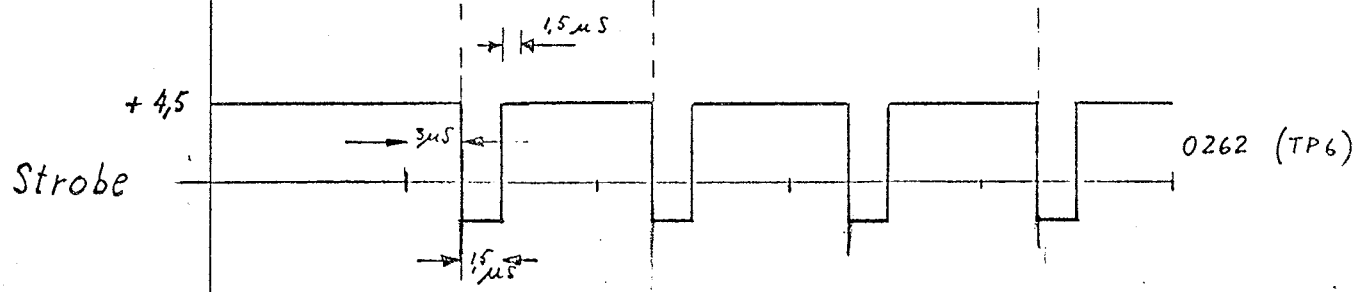
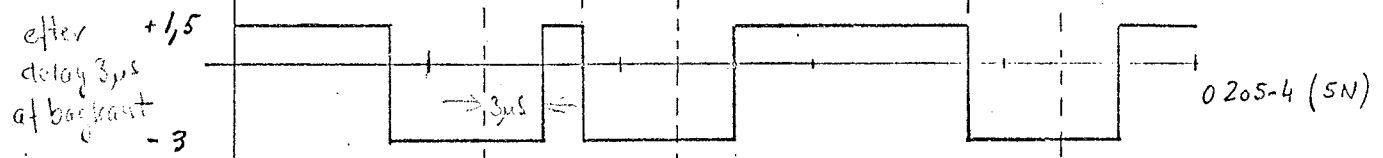
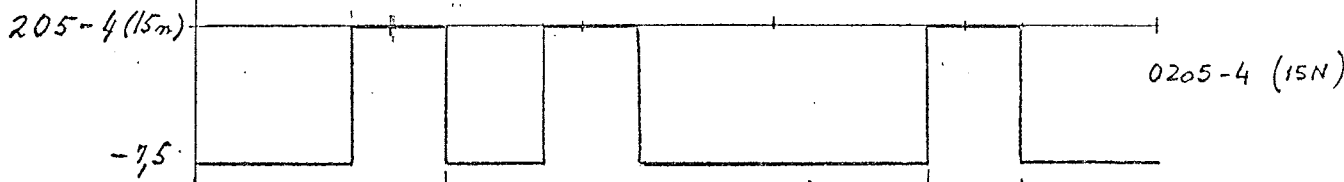
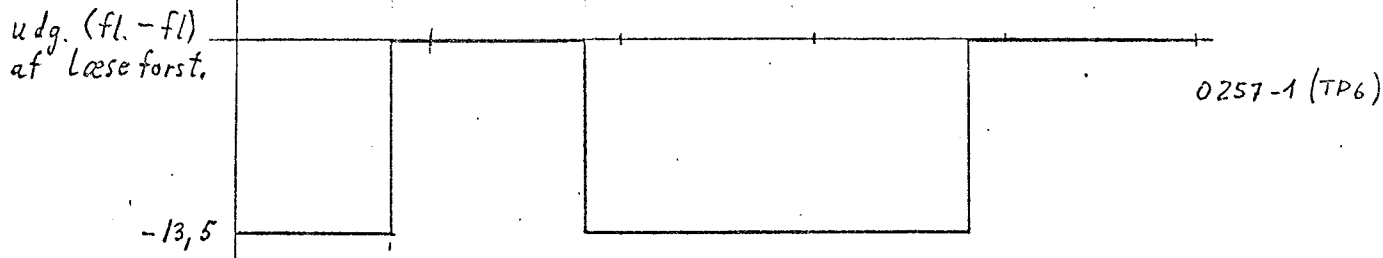
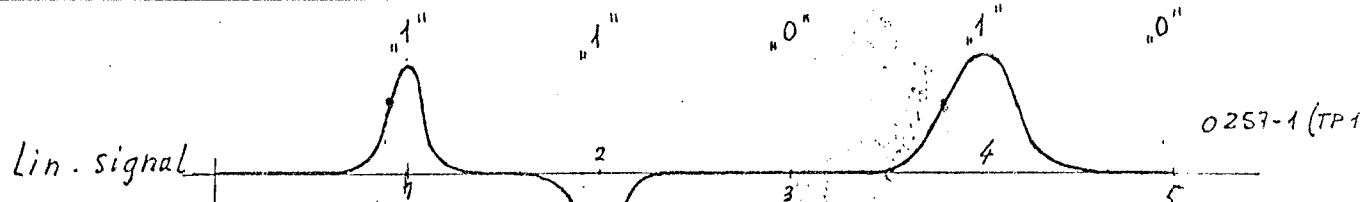




Unit: GIER 6	Designed P.E.P.	<b>BLOCKDIAGRAMS FOR DRUM CIRCUITS</b>	Drawing No
	Approved		Drawn by G.T. 3.9.66
	Checked 12.12.64.		Checked F.E. 22-2-67
	Last Revision		1 Sheets Sheet 1

Pulser i TR-kredsløb efter PEP

1 19



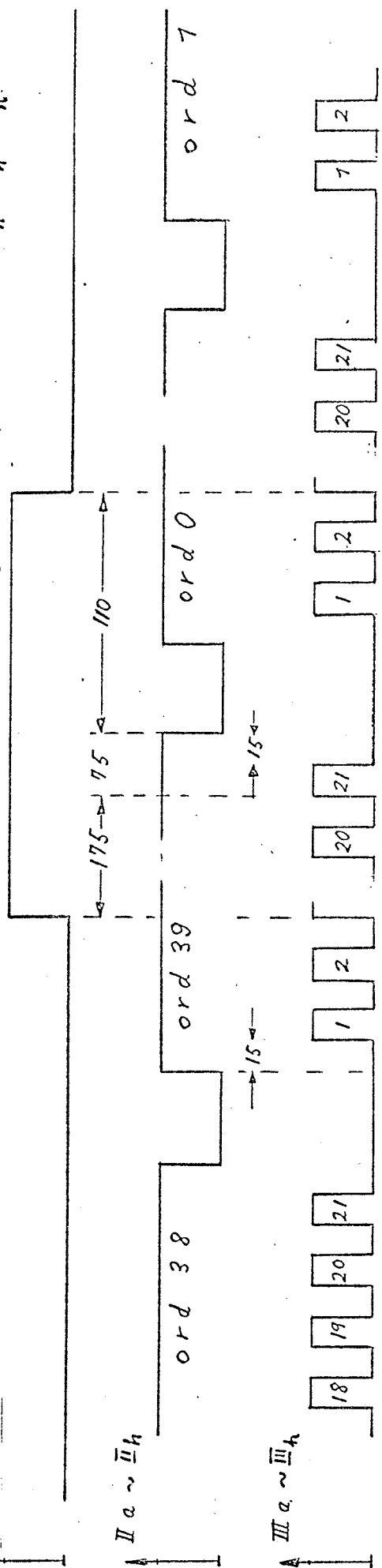
$$I_b \sim \overline{II}_b \sim \overline{III}_b \sim$$

$$\overline{I}_h \sim \overline{II}_h \sim \overline{III}_h$$

Ia ~ I<sub>h</sub>

IIa ~ II<sub>h</sub>

IIIa ~ III<sub>h</sub>



← indk. af skrivn. under KPI

41 A

40

39

38

TOT +  
evt. mente

2

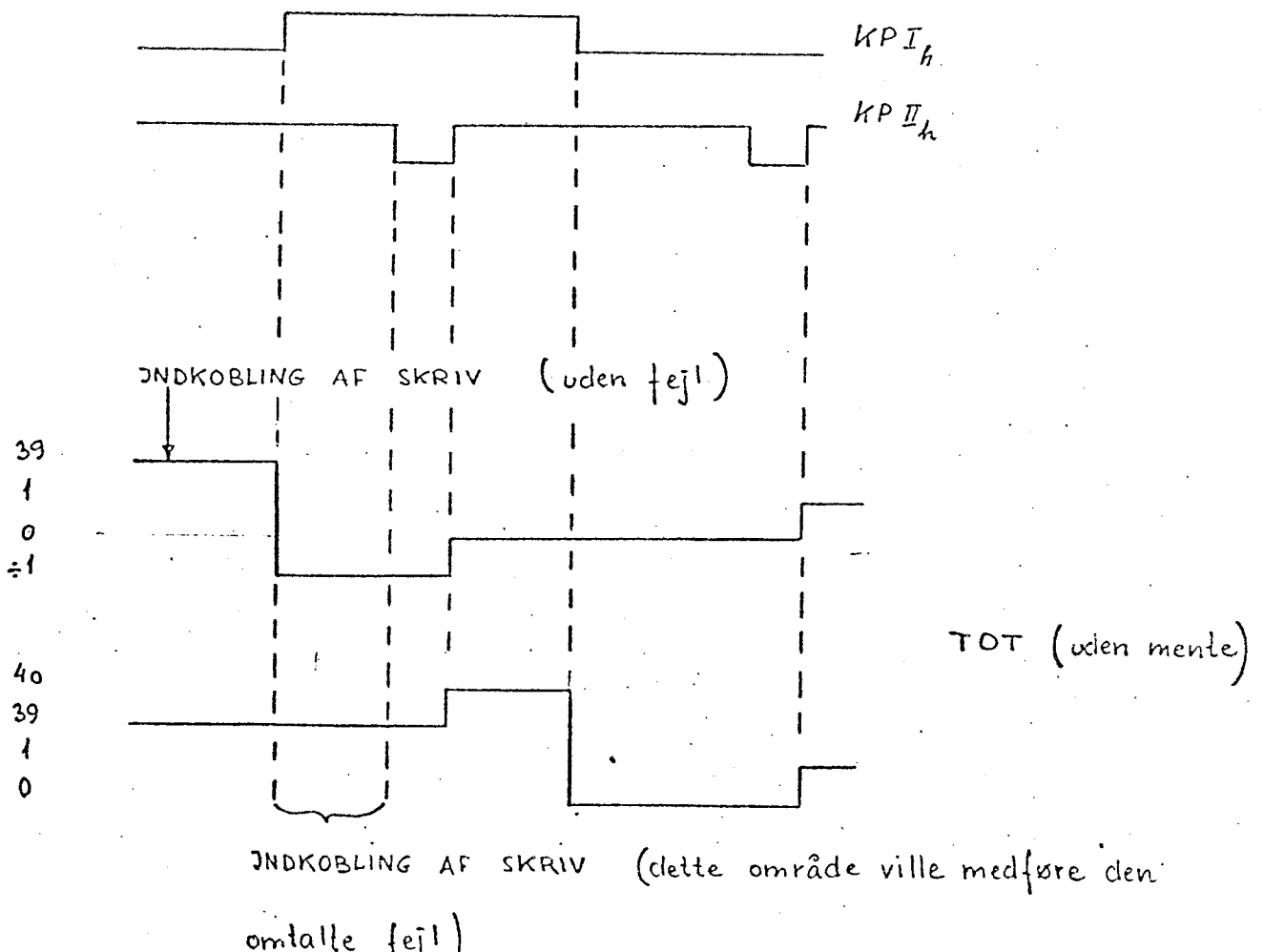
1

0

skrivning (+mente)

læsning

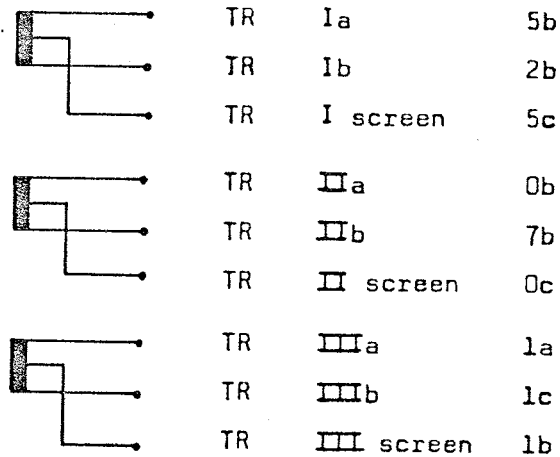
Ved skrivning skal transport af et ord foregå før skrivning, i modsætning til læsning, hvor transporten foregår efter læsningen. TOT tilføres derfor en mente i sidste pos. under skrivning. For at kunne transportere ord nr. 0 stilles TOT på  $\div 1$  på forkanten af  $KPI$ . Hvis indkoblingen sker som vist på tegningen, vil ord nr. 0 ikke blive transporteret, og operationen ville blive afsluttet netop som ord nr. 0 skulle skrives. For at hindre dette nulstilles TKT ved indkobling af skrivning under  $KPI_h \cdot KPI_{II}_h$



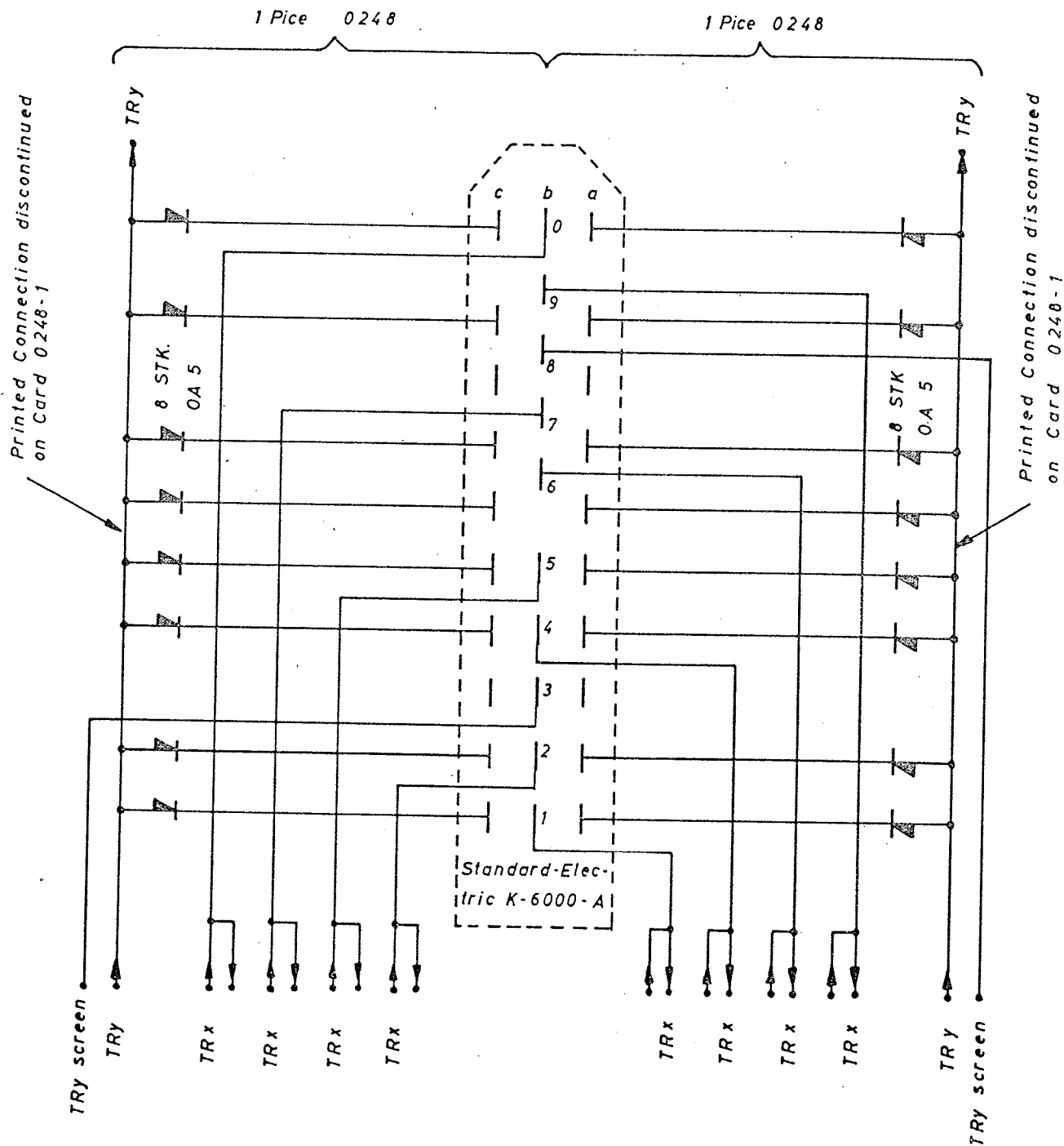
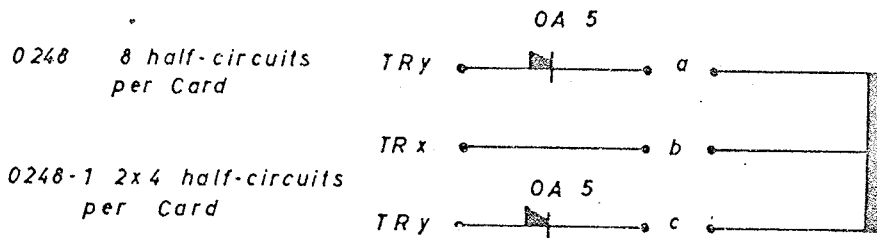
	Column D	Column A	Column B	Column C
1	0248-2			
2	0248	0248	0248	0248
3	0248-1	0248	0248-1	0248
4	0248	0248	0248	0248
5	0248	0248-1	0248	0248-1
6	0248	0248	0248	0248
7	0248	0248	0248	0248
8	0248-1	0248	0248-1	0248
9	0248	0248	0248	0248
10	0248	0248-1	0248	0248-1
11	0248	0248	0248	0248

2 Cards for each plug

The clock pulses are transferred from D1 (or possibly from A1, B1, or C1) via Cards 0248-2, which do not contain Diodes, but are Connected according to the following Diagram.



Unit: GIER 6	Designed H.I.	Drum Plug Distribution	Drawing No	
<b>REGNE</b> CENTRALEN	Approved		Drawn by R.N. 2.1.67	
	Checked 1.2.62		Checked F.E. 25-2-67	
	Last Revision		3 Sheets	Sheet 2
			0248-1-2	



Unit: GIER 6	Designed H.I.	Drawing No
	Approved	Drawn by R.N. 11.1.67.
	Checked 1. 2. 62.	Checked F.E. 23-2-67
	Last Revision	3 Sheets Sheet 1
		0248

Diodelogik, TR,

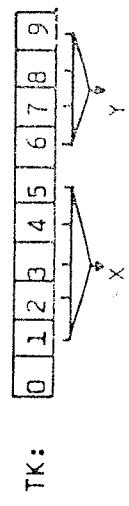




Associated track numbers and locations: Ex. Track no. 197: (X,Y) = (12.5). Location: Plug 09-5

X \ Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	A2-0	A3-1	A4-0	A6-1	A7-0	A8-1	A9-0	A11-1	C2-0	C3-1	C4-0	C6-1	C7-0	C8-1	C9-0	C11-1
1	A2-9	A3-2	A4-9	A6-2	A7-9	A8-2	A9-9	A11-2	C2-9	C3-2	C4-9	C6-2	C7-9	C8-2	C9-9	C11-2
2	A2-7	A3-4	A4-7	A6-4	A7-7	A8-4	A9-7	A11-4	C2-7	C3-4	C4-7	C6-4	C7-7	C8-4	C9-7	C11-4
3	A2-6	A3-5	A4-6	A6-5	A7-6	A8-5	A9-6	A11-5	C2-6	C3-5	C4-6	C6-5	C7-6	C8-5	C9-6	C11-5
4	A2-5	A3-6	A4-5	A6-6	A7-5	A8-6	A9-5	A11-6	C2-5	C3-6	C4-5	C6-6	C7-5	C8-6	C9-5	C11-6
5	A2-4	A3-7	A4-4	A6-7	A7-4	A8-7	A9-4	A11-7	C2-4	C3-7	C4-4	C6-7	C7-4	C8-7	C9-4	C11-7
6	A2-2	A3-9	A4-2	A6-9	A7-2	A8-9	A9-2	A11-9	C2-2	C3-9	C4-2	C6-9	C7-2	C8-9	C9-2	C11-9
7	A2-1	A3-0	A4-1	A6-0	A7-1	A8-0	A9-1	A11-0	C2-1	C3-0	C4-1	C6-0	C7-1	C8-0	C9-1	C11-0
8	D2-1	D4-0	D5-1	D6-0	D7-1	D9-0	D10-1	D11-0	B2-1	B4-0	B5-1	B6-0	B7-1	B9-0	B10-1	B11-0
9	D2-2	D4-9	D5-2	D6-9	D7-2	D9-9	D10-2	D11-9	B2-2	B4-9	B5-2	B6-9	B7-2	B9-9	B10-2	B11-9
10	D2-4	D4-7	D5-4	D6-7	D7-4	D9-7	D10-4	D11-7	B2-4	B4-7	B5-4	B6-7	B7-4	B9-7	B10-4	B11-7
11	D2-5	D4-6	D5-5	D6-6	D7-5	D9-6	D10-5	D11-6	B2-5	B4-6	B5-5	B6-6	B7-5	B9-6	B10-5	B11-6
12	D2-6	D4-5	D5-6	D6-5	D7-6	D9-5	D10-6	D11-5	B2-6	B4-5	B5-6	B6-5	B7-6	B9-5	B10-6	B11-5
13	D2-7	D4-4	D5-7	D6-4	D7-7	D9-4	D10-7	D11-4	B2-7	B4-4	B5-7	B6-4	B7-7	B9-4	B10-7	B11-4
14	D2-9	D4-2	D5-9	D6-2	D7-9	D9-2	D10-9	D11-2	B2-9	B4-2	B5-9	B6-2	B7-9	B9-2	B10-9	B11-2
15	D2-0	D4-1	D5-0	D6-1	D7-0	D9-1	D10-0	D11-1	B2-0	B4-1	B5-0	B6-1	B7-0	B9-1	B10-0	B11-1
16	D3-0	D3-5	A5-1	A5-6	D8-0	D8-5	A10-1	A10-6	B3-0	B3-5	C5-1	C5-6	B8-0	B8-5	C10-1	C10-6
17	D3-9	D3-4	A5-2	A5-7	D8-9	D8-4	A10-2	A10-7	B3-9	B3-4	C5-2	C5-7	B8-9	B8-4	C10-2	C10-7
18	D3-7	D3-2	A5-4	A5-9	D8-7	D8-2	A10-4	A10-9	B3-7	B3-2	C5-4	C5-9	B8-7	B8-2	C10-4	C10-9
19	D3-6	D3-1	A5-5	A5-0	D8-6	D8-1	A10-5	A10-0	B3-6	B3-1	C5-5	C5-0	B8-6	B8-1	C10-5	C10-0

Drum Tracks 0-319 are represented by bit 1-9, which are decoded in x (bit 1-5), and y (bit 6-9).



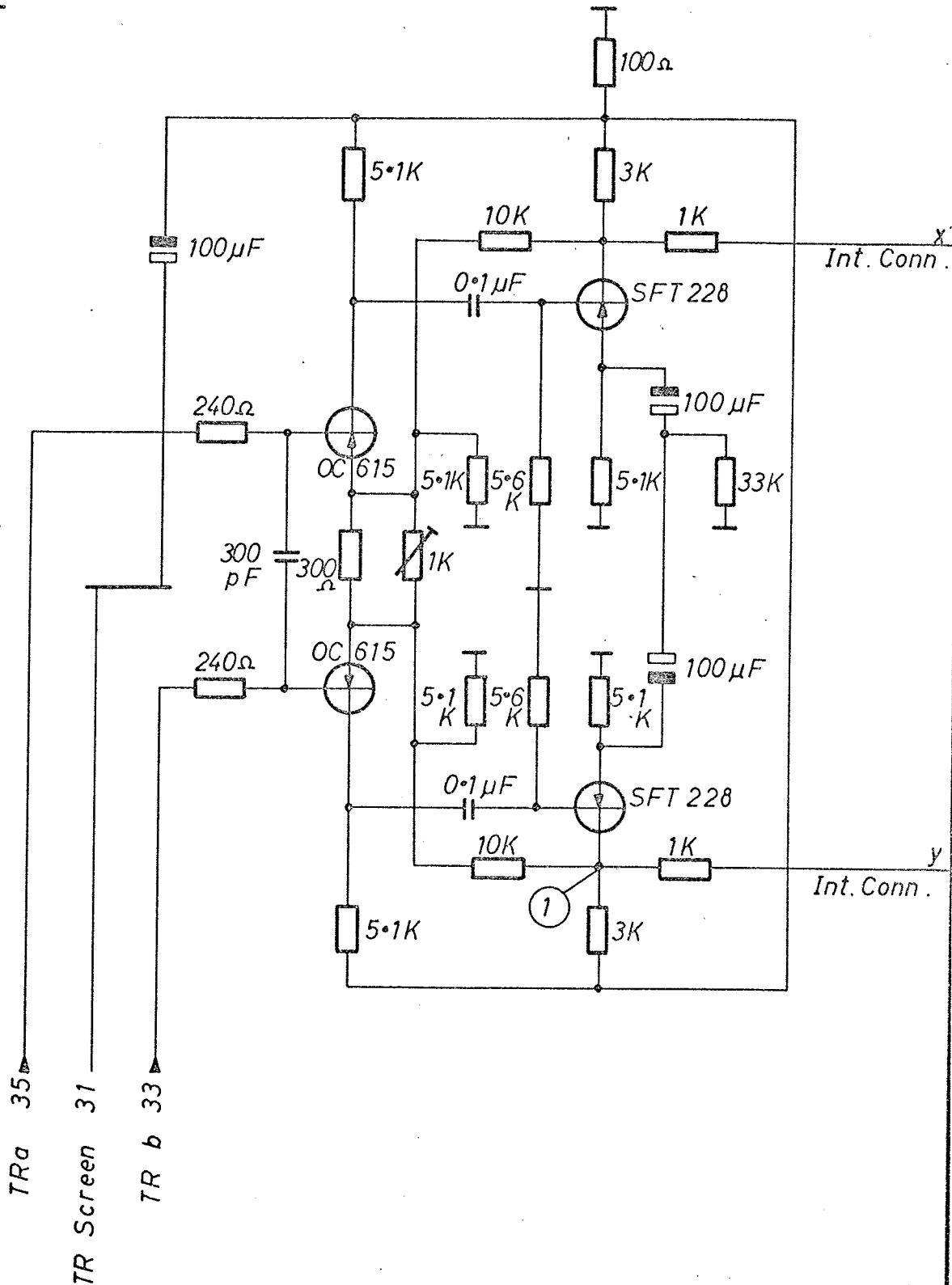
Unit: GIER 6  CENTRALEN	Designed H. I	Drawing No
	Approved	Drawn by R. N. S. 2.08
	Checked 1.2.62	Checked F. E. 20-2-67
Lost Revision	Numbering and Location of Drum Tracks	3 Sheets Sheet 3
		248-1-2

1 -13.5V -

21 +10.5V -

31,30,29 0V -

21 +10.5V -



Unit: GIER 6

Designed H. I.

Drawing No

Drawn by B. R. 23. 8. 66

§ **BEGNE**  
CENTRALEN

Approved

DRUM CLOCK PULSE  
AMPLIFIER

Checked F. E. 22-2-67

Checked 1. 2. 62.

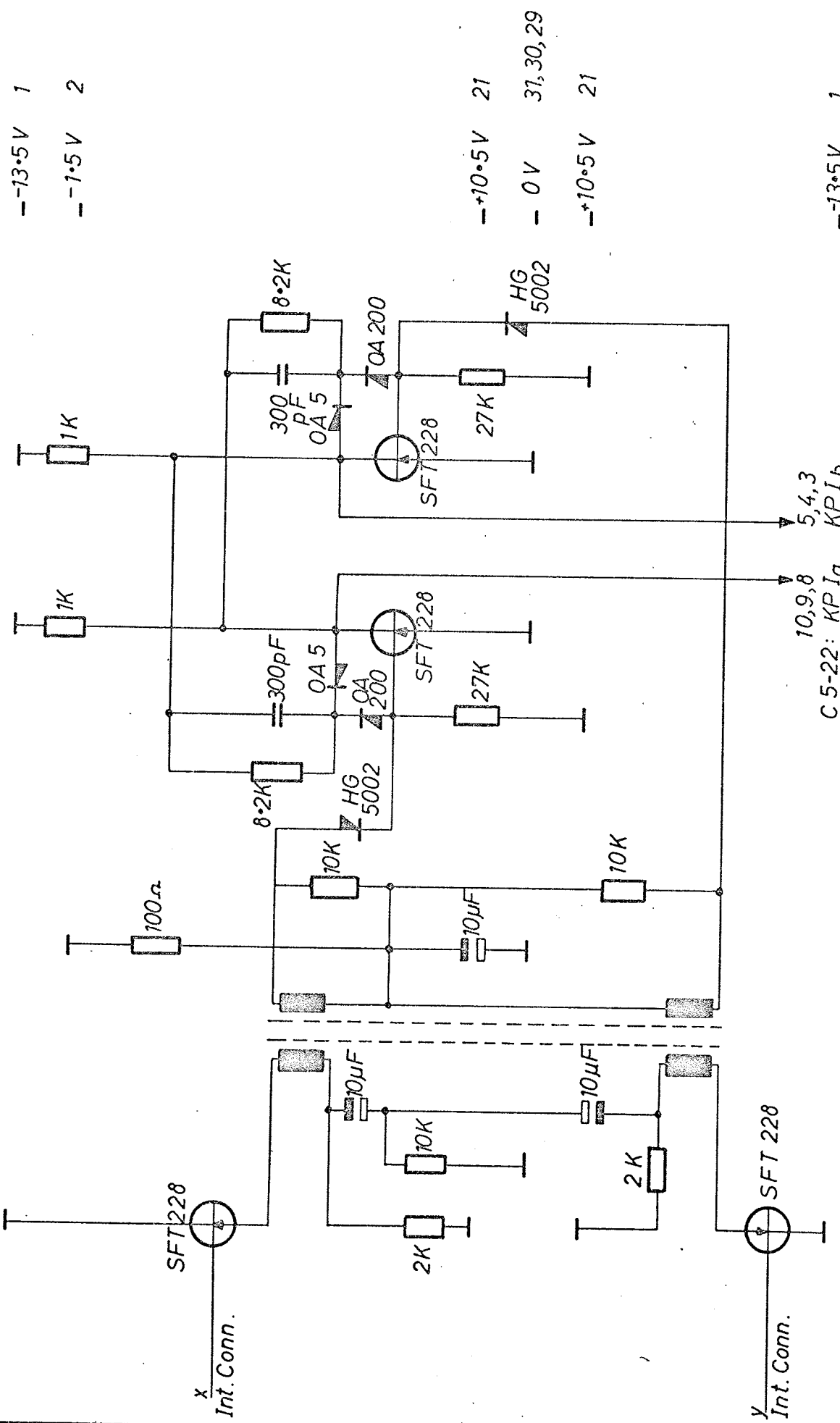
2 Sheets

Sheet 2

Last Revision

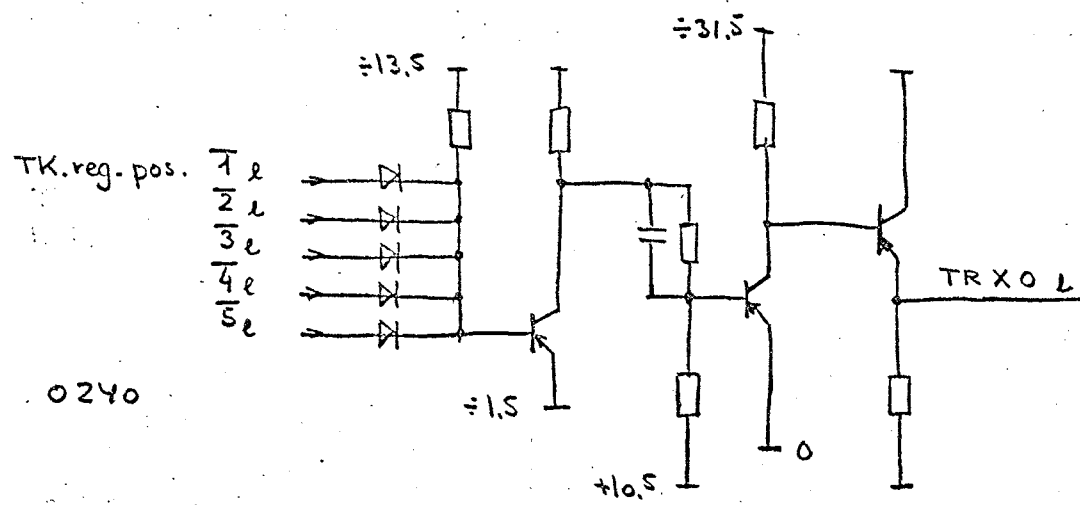
C 5

257

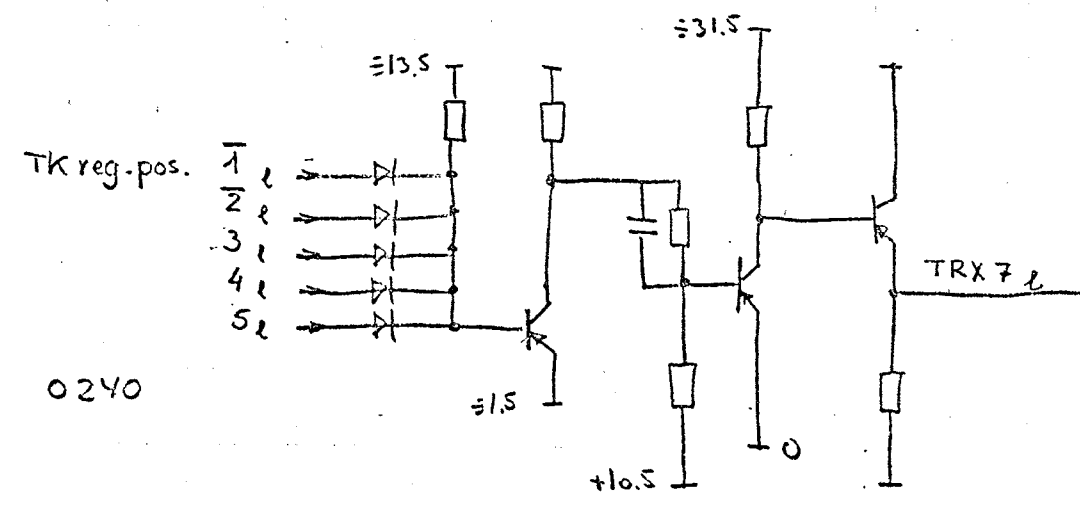


5,4,3  
KPI b  
C 5-22: KPI a  
10,9,8  
C 5-23: KPI a  
KPII b  
C 5-24: KPII a  
KPIII b

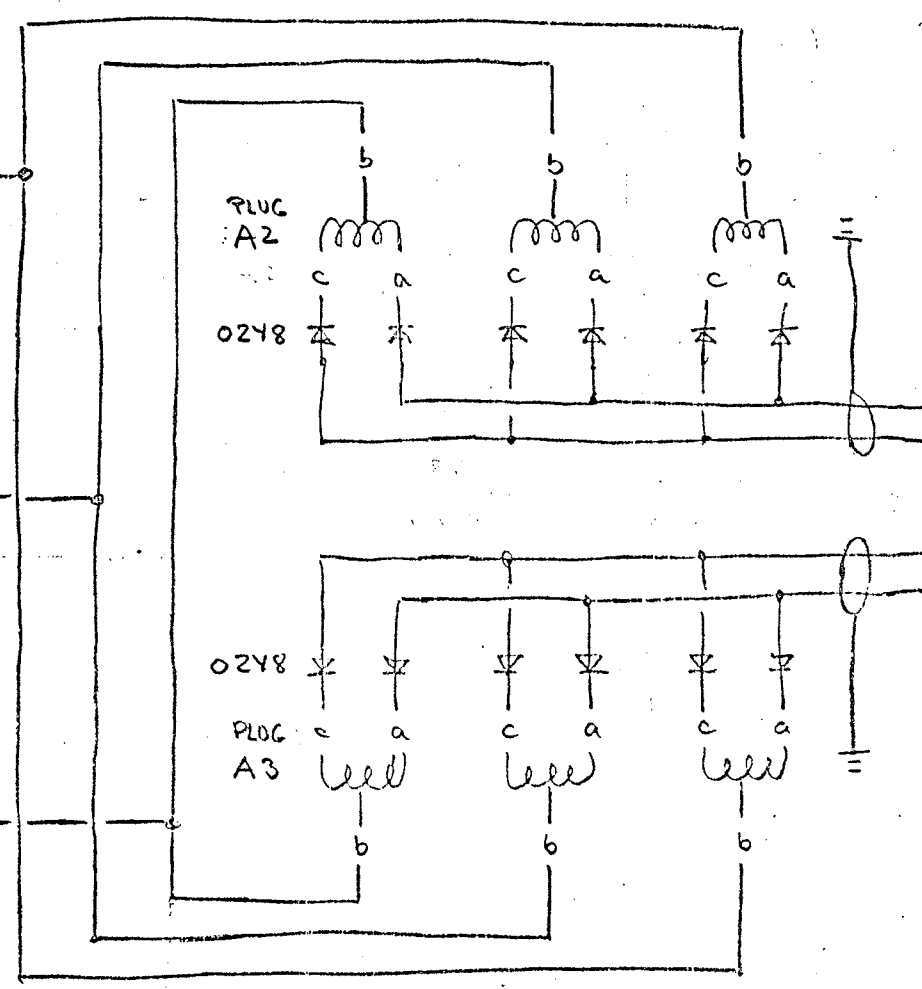
Unit: GIER 6 	Designed H. I.	DRUM CLOCK PULSE AMPLIFIER	Drawing No	
	Approved		Drawn by B. R. 23. 8. 66	
	Checked 1. 2. 62.		Checked F. E. 22-2-67	
	Last Revision		2 Sheets	Sheet 1
			C 5	257



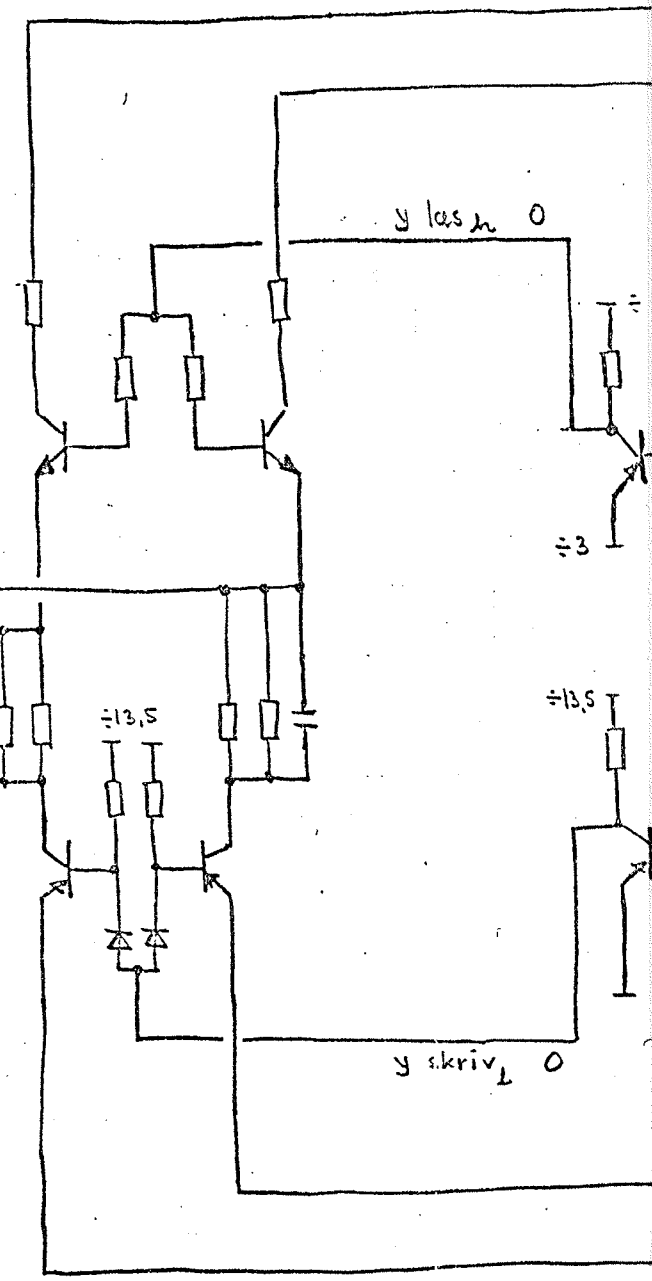
(only 2 of 20 x decodn. shown)



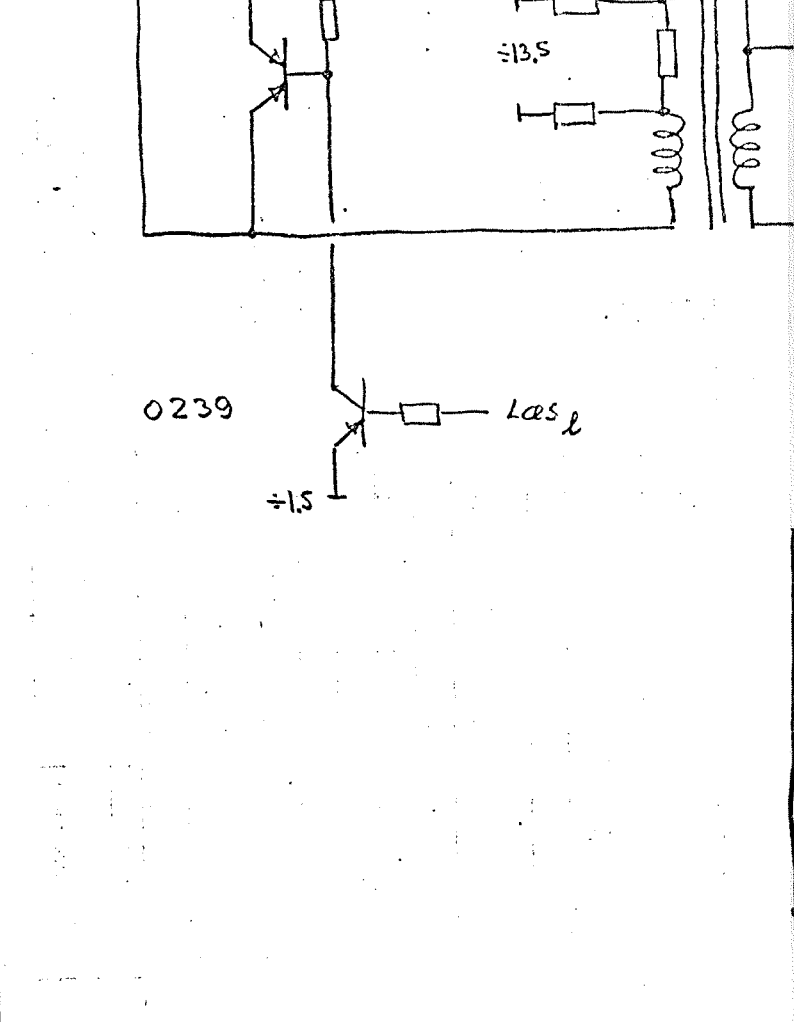
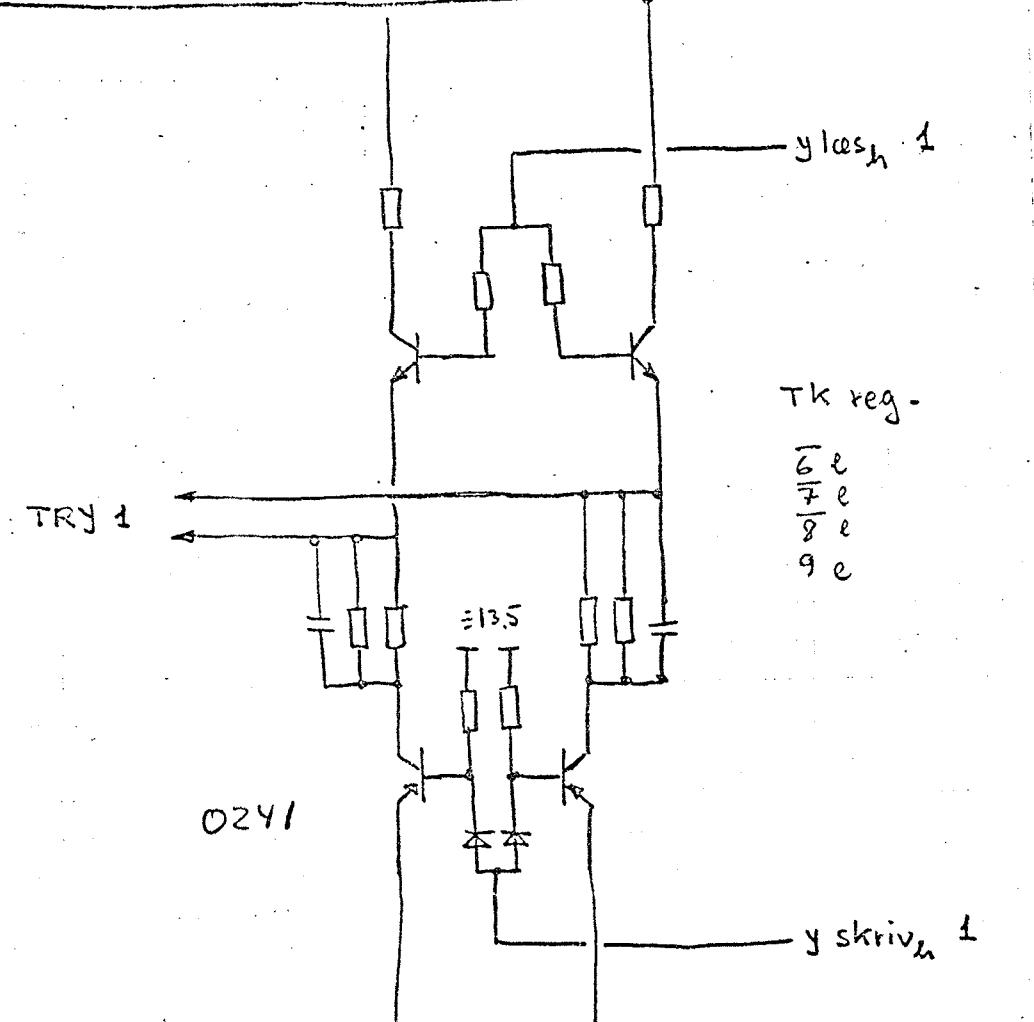
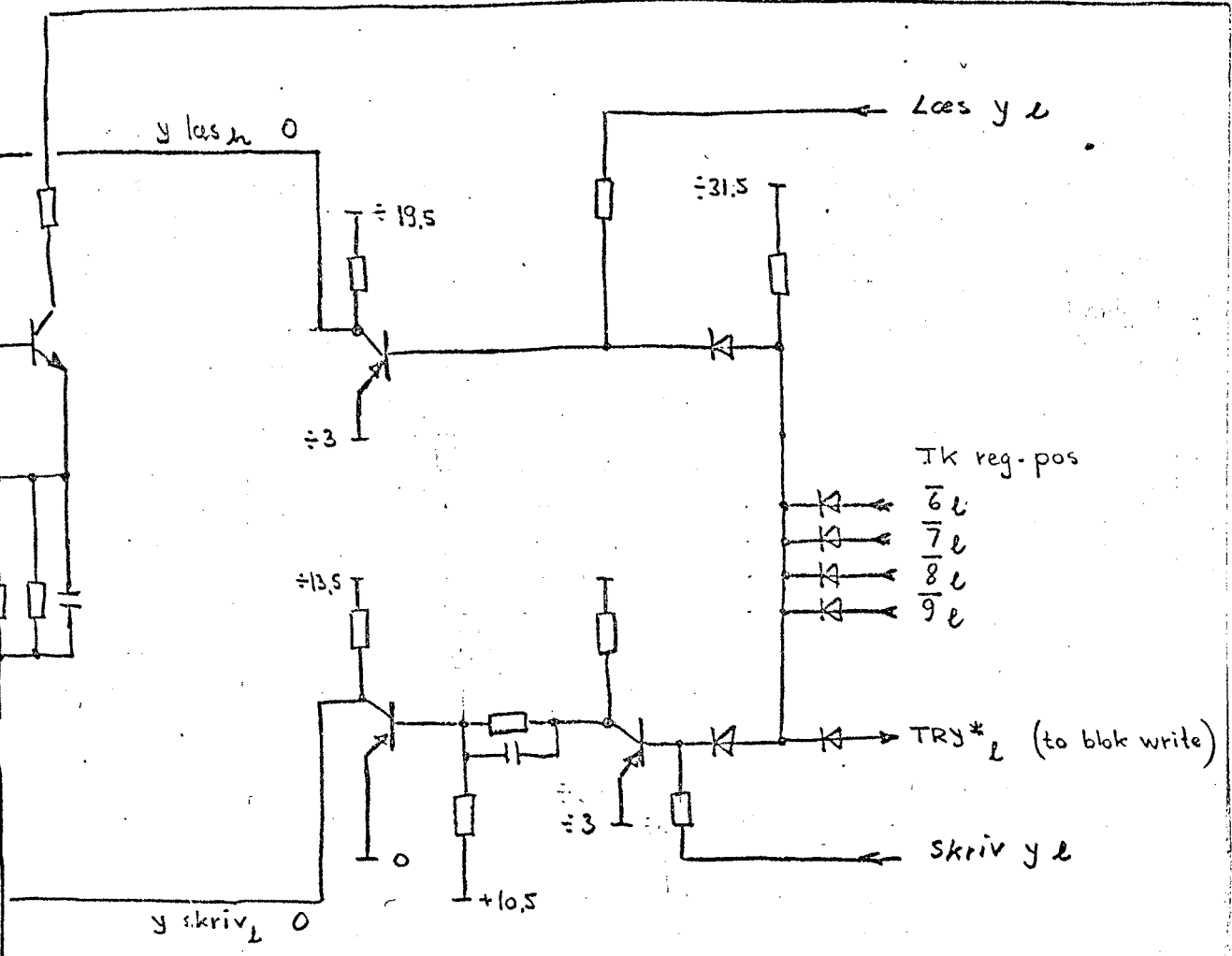
(only 3 of 20 x wires shown)



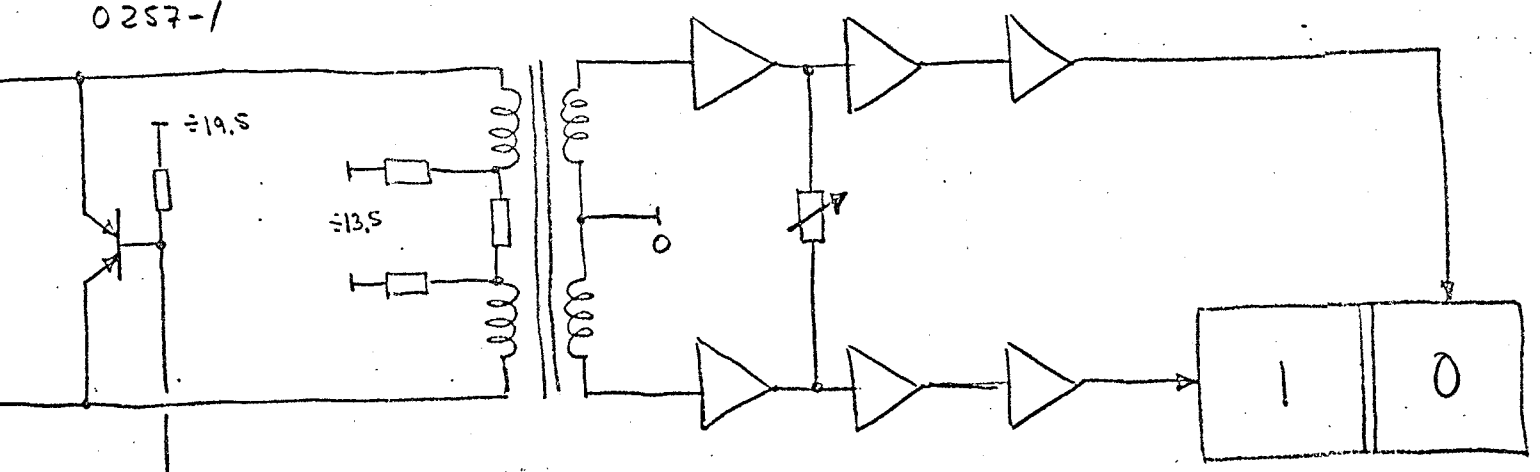
(only 2 of 6 y wires shown)



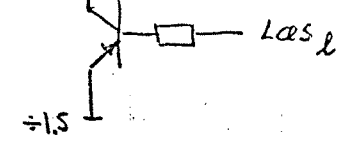
(only 3 of 8 heads shown on each plug)



0257-1

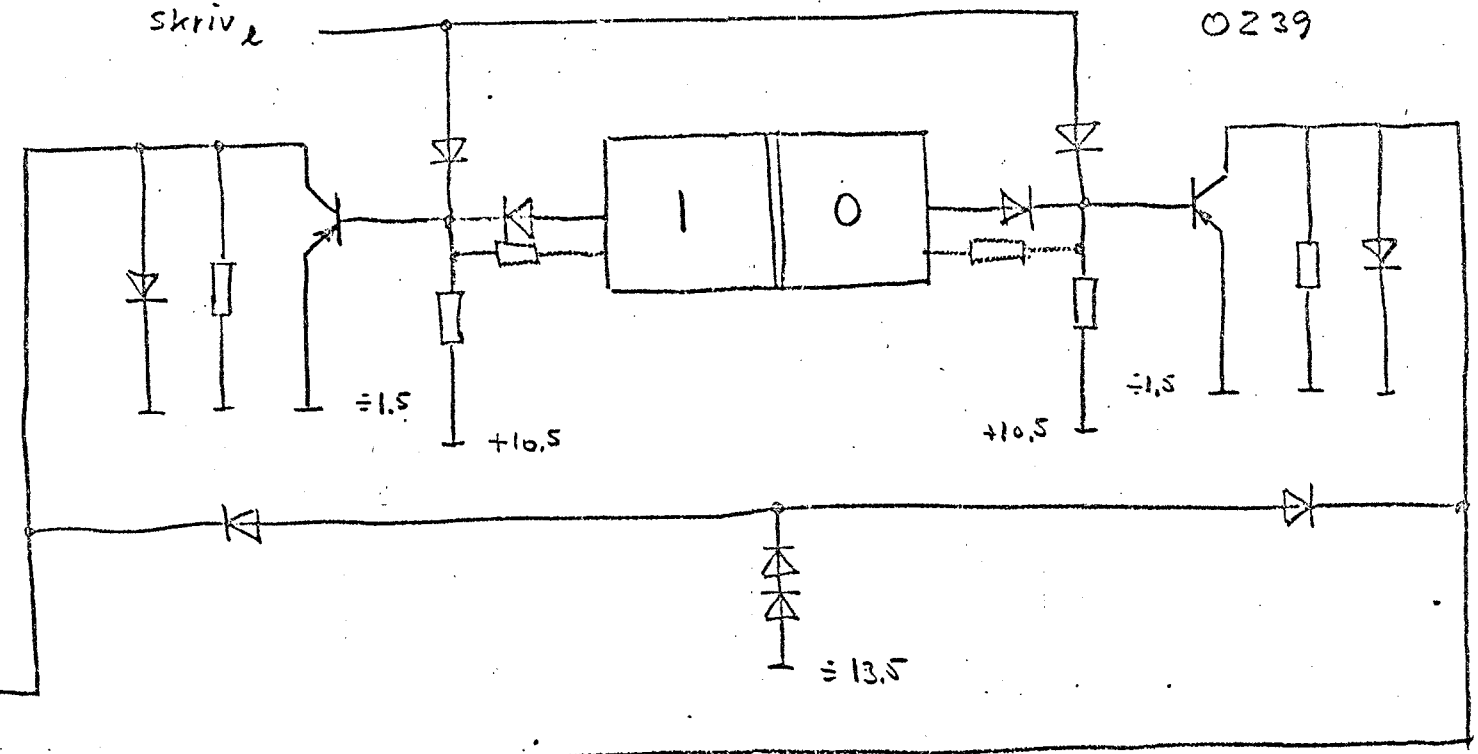


239



skriv

0239

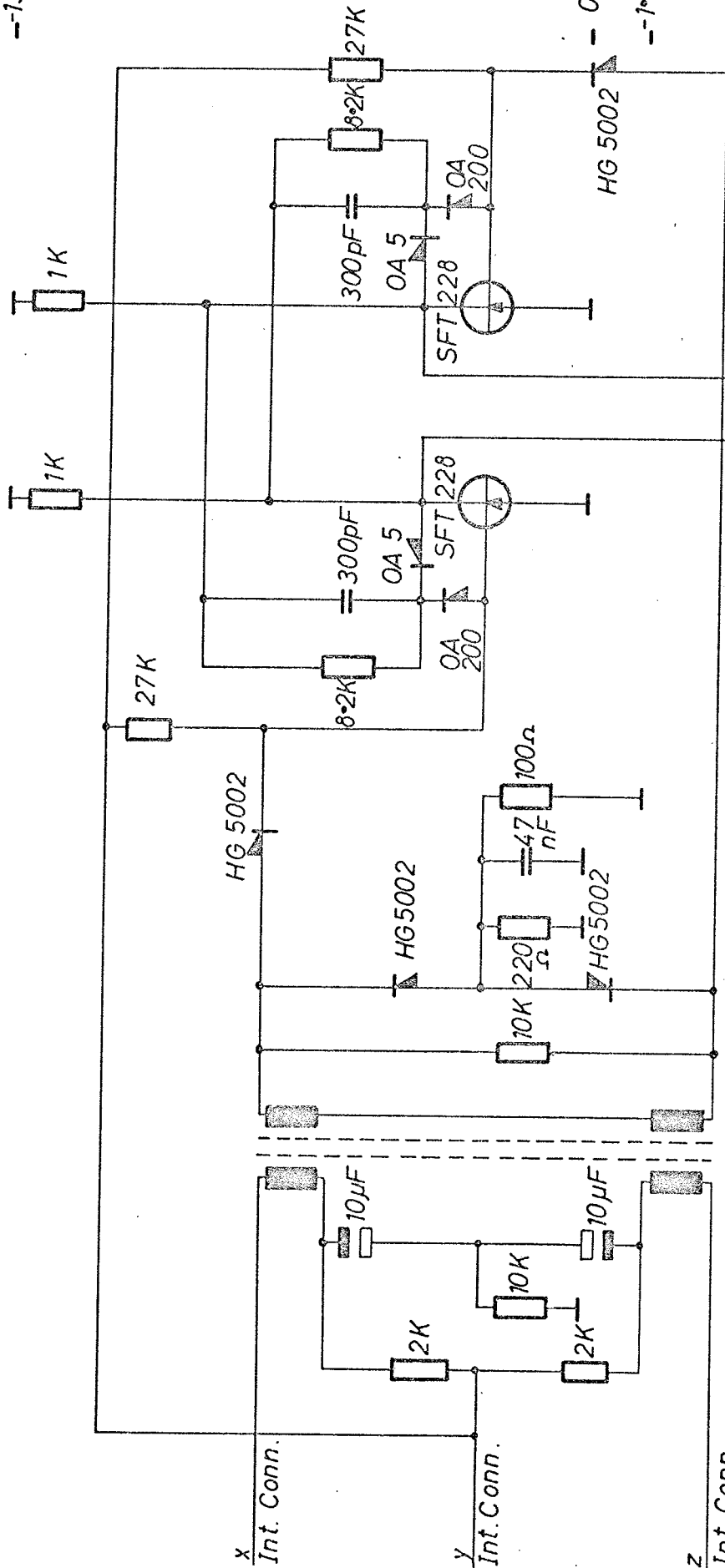


13/2-66

R. Ham

-13.5V 1

0V 31,30, 29,12  
-1.5V 2



Unit: GIER 6

REGNE CENTRALEN

Designed H.I.

Approved

Checked 1. 2. 62.

Last Revision

# DRUM SIGNAL AMPLIFIER

Drawing No	
Drawn by B.R. 22.9.66.	
Checked F.E. 22-2-67	
2 Sheets	Sheet 2
Pos. C5-25	0257-1



#### 4. Start-stop circuits

The clockpulses from the 450 KHz oscillator are distributed to different parts of the machine controlled by signals from the pushbuttons on the operators panel.

There are in principle 3 different output-clockpulses:

1. KPA and KP.A which drives the microaddress flip-flops and gates in the control unit during normal operation.
2. KPB and KP.B which drives the special microaddress flip-flops and gates in the control unit during a drumtransfer function.
3. Display and 'zero to all lines' which drives the register-display-system that is active when the machine is stopped.

Besides this there are some special pulses (Sp-adder, SpH and SpMC) which controls the shortterm-memories in the outputs of the adder and the registers H and MC.

The distribution of clockpulses is controlled by means of the flip-flops VAC and VAB.

The flip-flop VAT is only one when the special testmicroprograms are activated by means of the switch in the central processor.

When VAC = 1 the outputgates for the display-system are enabled.

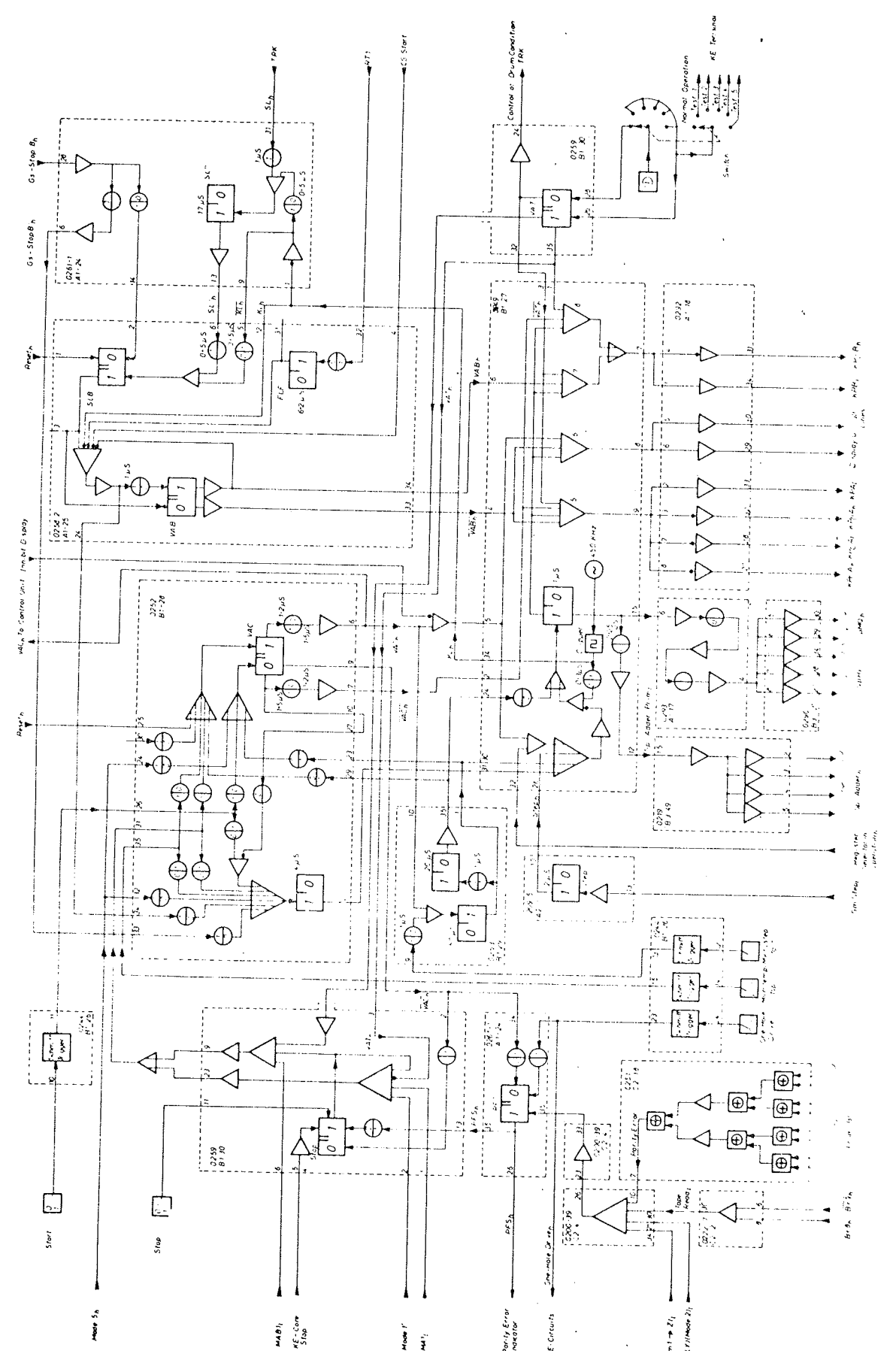
When VAC = 0 the machine is running and gate 5 on card 249 is enabled.

If VAB = 0 gate 5 will send outputpulses and KPA and KP.A will be active. If VAB = 1 gate 5 will be closed and gate 7 open so that KPB and KP.B will be active.

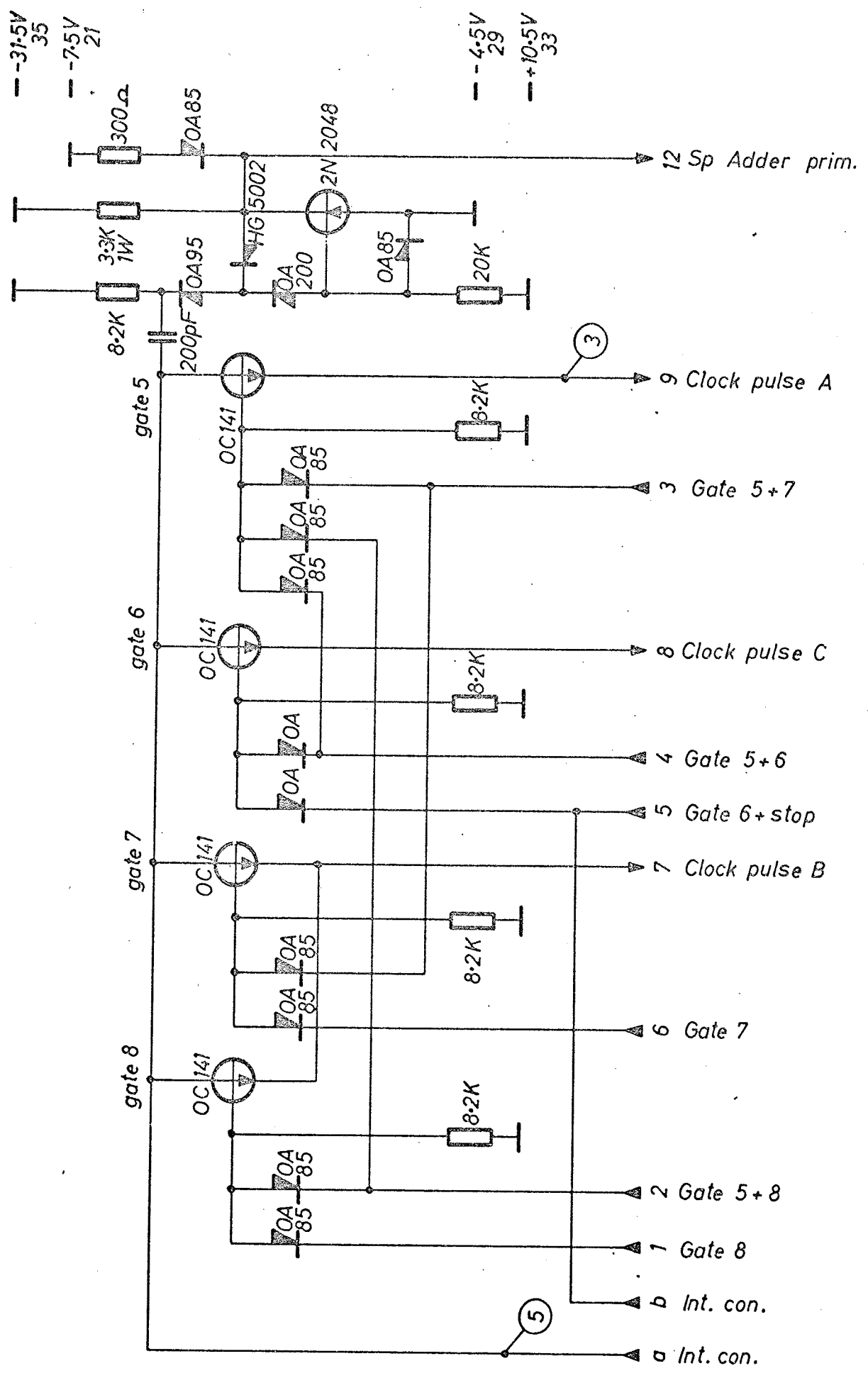
It is seen that if VAC = 1 and VAB = 1 gate 7 will still be active while display is inactive. This means that a drum transfer may be finished after VAC has been set to 1.

The primary clockpulse is interrupted in the following situations: after execution of microoperation Gmstep (1 clockpulse is removed), when VAC or VAB are changing or when pushbuttons are activated.

The logic around VAB is synchronizing the drumoperations and the clockpulse, and makes sure that a core-memory cycle is not interrupted by the drum.



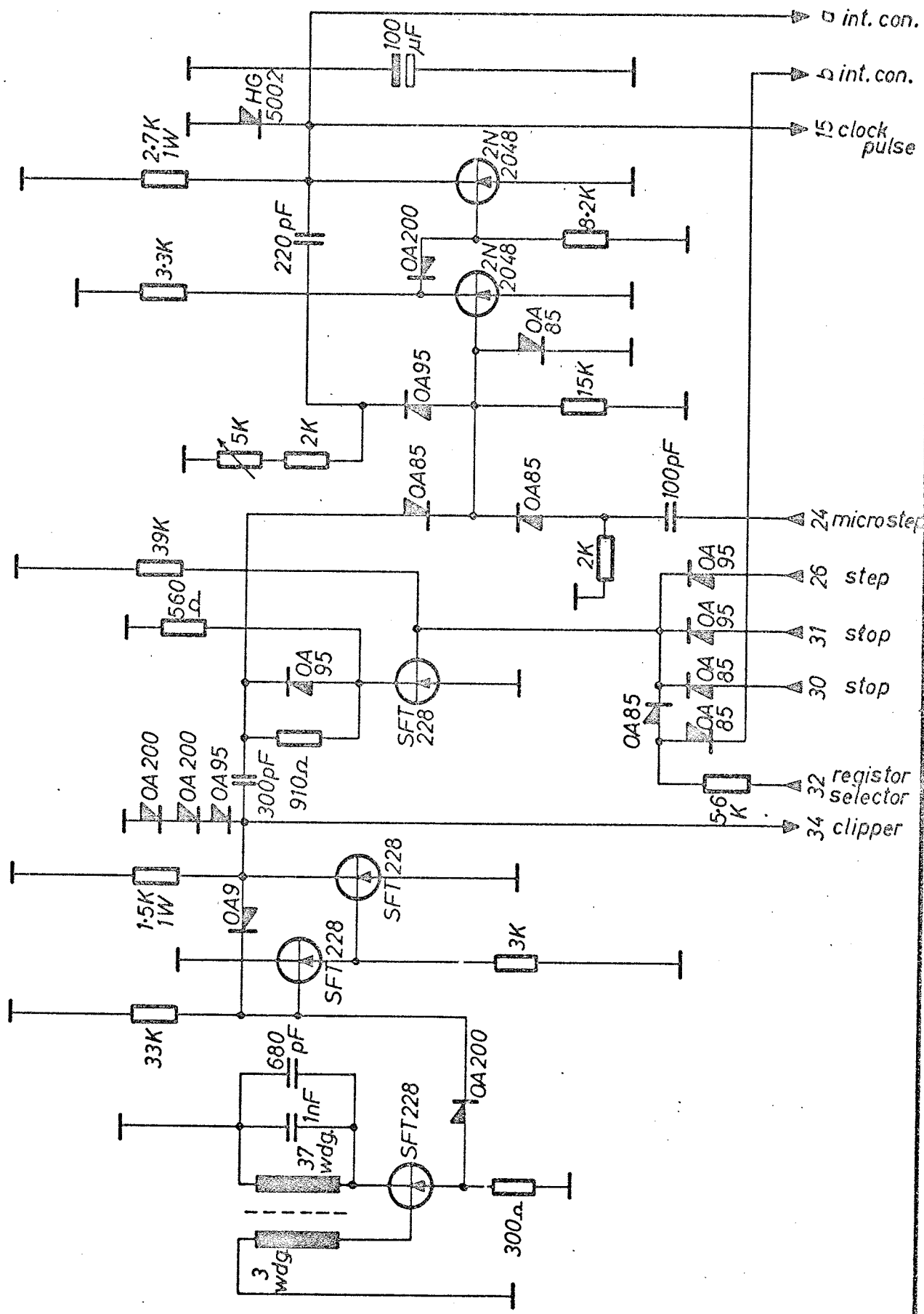
Drawing No Drawn by: <b>BR 1-12-66</b> Checked by: <b>FE 26-2-67</b> 1 Sheets Sheet 1	<b>START STOP CIRCUITS</b>	H.I. 1.262
<b>6</b>  <b>CENTRALEN</b>	H.I. 1.262	H.I. 1.262



Unit: GIER 6   BEGNE CENTRALEN	Designed H.I.	OSC. AND CLIPPER	Drawing No	
	Approved		Drawn by G.T. 23. 9. 66	
	Checked 1. 2. 62.		Checked F.E. 22-2-67	
	Last Revision		2 Sheets	Sheet 1
			Pos. B1 - 27	249

--- 31.5V 35  
 --- 13.5V 10  
 --- 7.5V 21  
 --- 4.5V 29

--- 1.5V 23  
 --- 0V 22  
 --- +1.5V 11  
 --- +10.5V 33



Unit: GIER 6  
**REGNE**  
 CENTRALEN

Designed H. I.  
 Approved  
 Checked 1. 2. 62.  
 Last Revision

OSC. AND CLIPPER

Drawing No  
 Drawn by G.T. 23. 9. 66.  
 Checked F.E. 20-8-67  
 2 Sheets Sheet 2  
 Pos. B1-27 249

### 8. Indicator circuits

The indicator operation is defined by means of the bits OR 33 and OR 34 in the following manner:

OR 33	OR 34	Name		Description
0	0	B <sub>0</sub>	I	Storing in indicator
0	1	B <sub>1</sub>	M	Marking of cell
1	0	B <sub>2</sub>	N	Conditioning of instruction
1	1	B <sub>3</sub>	L	- - -

It will be seen that each operation has two names. The names B<sub>0</sub>-B<sub>3</sub> are only used in the technical papers. The decoding of the indicator-operation is done by the 'I -afk' on the card 222-2.

The indicator address is defined by means of the bits OR35 - OR39. The first 3 bits are decoded by means of the 'IA-afk' on the cards 222-1 in the following manner:

OR 35	OR 36	OR 37	Name	Description
0	0	0	N	Zerokombination
0	0	1	K	Push buttons on control panel
0	1	0	W or Z	Zero status of the R-gister
0	1	1	O	Overflow
1	0	0	T	Sign
1	0	1	P	Marking bits
1	1	0	Q	- -
1	1	1	R	- -

The two last bits of the address are used in the following manner:

OR 38	OR 39	Name
0	0	Zerokombination
1	0	A
0	1	B
1	1	C

The Indicator-register IN consists of 12 flip-flops with the names:

OA, OB, TA, TB, PA, PB, QA, QB, RA, RB, KA and KB.

The 10 first are connected to the busline-system and are placed upon the circuit cards 216-1. KA and KB can only be set manually from the control panel and are placed upon the card 200-5.

The specific functions of the indicator-logic are described in full detail in 'A Manual of GIER Programming', vol 1, page 90-95.

Conditioning of an instruction by means of indicatoroperations N or L is performed by a group of logic circuits which gives the outputsignal B.

If  $B = 1$  the condition is fulfilled and the instruction will be executed.

This logic is placed upon the cards 233 and 200-9. The Boolean expression for B can be written in this way:

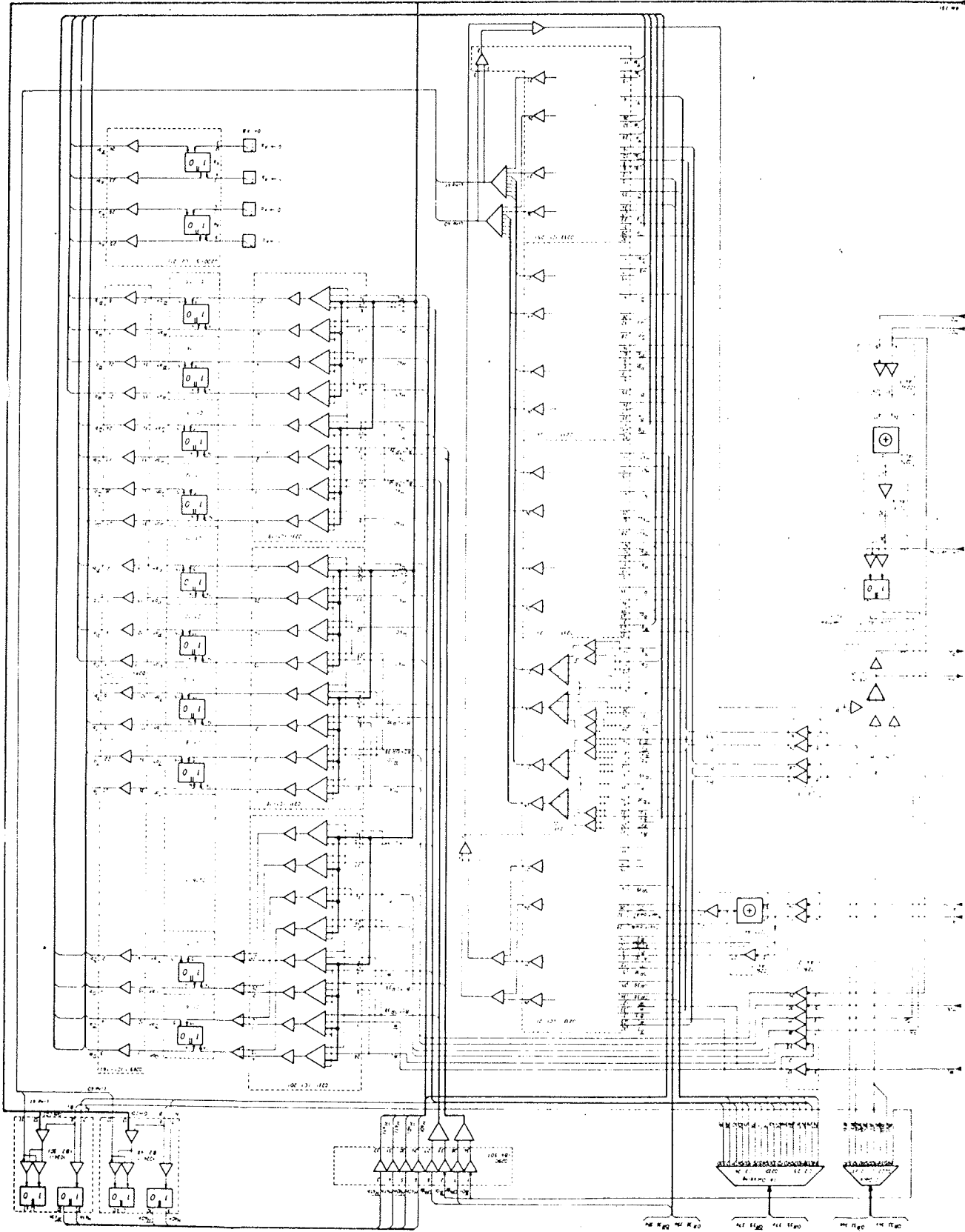
$$\begin{aligned}
 B = & (\bar{N} + \bar{B}_3 \cdot \bar{H}_{40} + B_3 \cdot H_{40} + \overline{OR}_{38}) (\bar{N} + \bar{B}_3 \cdot \bar{H}_{41} + B_3 \cdot H_{41} + \overline{OR}_{39}) \cdot \\
 & (\bar{K} + \bar{B}_3 \cdot \bar{K}_A + B_3 \cdot K_A + \overline{OR}_{38}) (\bar{K} + \bar{B}_3 \cdot \bar{K}_B + B_3 \cdot K_B + \overline{OR}_{39}) \cdot \\
 & (\bar{W} + \bar{B}_3 \cdot \bar{O}_A + B_3 \cdot O_A + \overline{OR}_{38}) (\bar{W} + \bar{B}_3 \cdot \bar{O}_B + B_3 \cdot O_B + \overline{OR}_{39}) \cdot \\
 & (\bar{W} + \bar{B}_3 \cdot \bar{w} + B_3 \cdot w + OR_{38} + OR_{39}) \cdot \\
 & (\bar{O} + \bar{B}_3 \cdot \bar{O}_A + B_3 \cdot O_A + \overline{OR}_{38}) (\bar{O} + \bar{B}_3 \cdot \bar{O}_B + B_3 \cdot O_B + \overline{OR}_{39}) \cdot \\
 & (\bar{O} + \bar{B}_3 \cdot \text{overflow in AC} + B_3 \cdot \text{overflow in AC} + OR_{38} + OR_{39}) \cdot \\
 & (\bar{T} + \bar{B}_3 \cdot \bar{T}_A + B_3 \cdot T_A + \overline{OR}_{38}) (\bar{T} + \bar{B}_3 \cdot \bar{T}_B + B_3 \cdot T_B + \overline{OR}_{39}) \cdot \\
 & (\bar{T} + \bar{B}_3 \cdot \overline{AC}_{00} + B_3 \cdot AC_{00} + \overline{OR}_{38} + OR_{39}) \cdot \\
 & (\bar{P} + \bar{B}_3 \cdot \bar{P}_A + B_3 \cdot P_A + \overline{OR}_{38}) (\bar{P} + \bar{B}_3 \cdot \bar{P}_B + B_3 \cdot P_B + \overline{OR}_{39}) \cdot \\
 & (\bar{Q} + \bar{B}_3 \cdot \bar{Q}_A + B_3 \cdot Q_A + \overline{OR}_{38}) (\bar{Q} + \bar{B}_3 \cdot \bar{Q}_B + B_3 \cdot Q_B + \overline{OR}_{39}) \cdot \\
 & (\bar{R} + \bar{B}_3 \cdot \bar{R}_A + B_3 \cdot R_A + \overline{OR}_{38}) (\bar{R} + \bar{B}_3 \cdot \bar{R}_B + B_3 \cdot R_B + \overline{OR}_{39}) \cdot
 \end{aligned}$$

In the case that indicator operation L is used then we have  $B_3 = 1$ . For operation N we have  $B_3 = 0$  in the expression. By selecting an address f.ex. P, we have  $P = 1$  and  $\bar{P} = 0$ . This means that only the paranthes containing  $\bar{P}$  is interesting, because all other addresses N, K, W, O, T, R and  $R = 0$  that is  $\bar{N}, \bar{K}, \bar{W}, \dots, \bar{R} = 1$  so that all the rest of the parantheses are equal to 1. The expression simplifies in this case

For  $B_3 = 1$  to  $B = (PA + \overline{OR}_{39}) (PB + \overline{OR}_{39})$  and

for  $B_2 = 1$  to  $B = (\overline{PA} + \overline{OR}_{38}) (\overline{PB} + \overline{OR}_{39})$ .

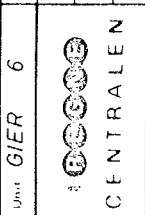




0-20-10  
 Drawn by DR 7-12-65  
 Checked by FE 26-2-67  
 1 Sheets Sheet 1

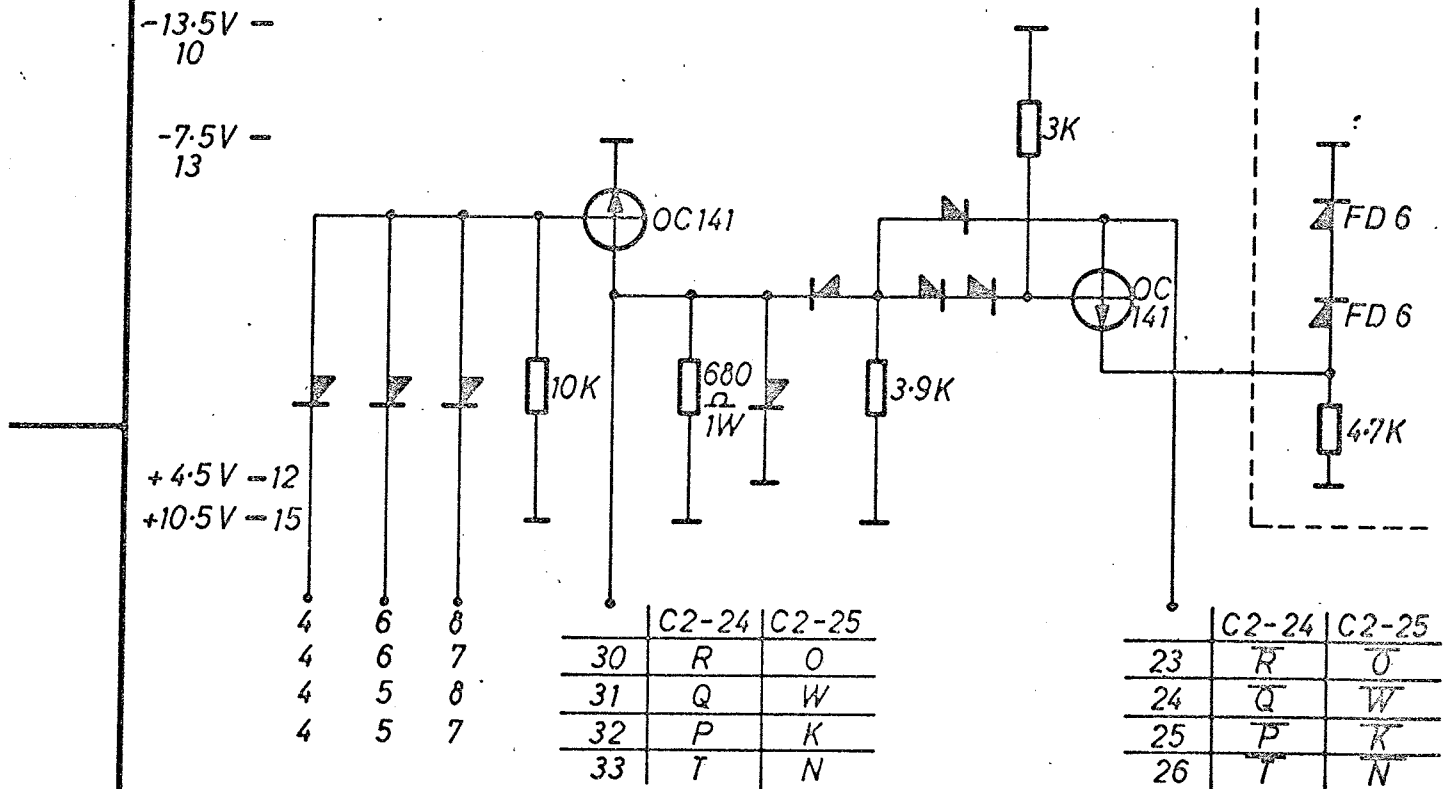
**BLOCK DIAGRAM  
 FOR  
 INDICATOR**

Unit GIER 6  
 Designed H. I.  
 Approved  
 Checked 1.2.62  
 Lost Revision




4 Circuits per card.

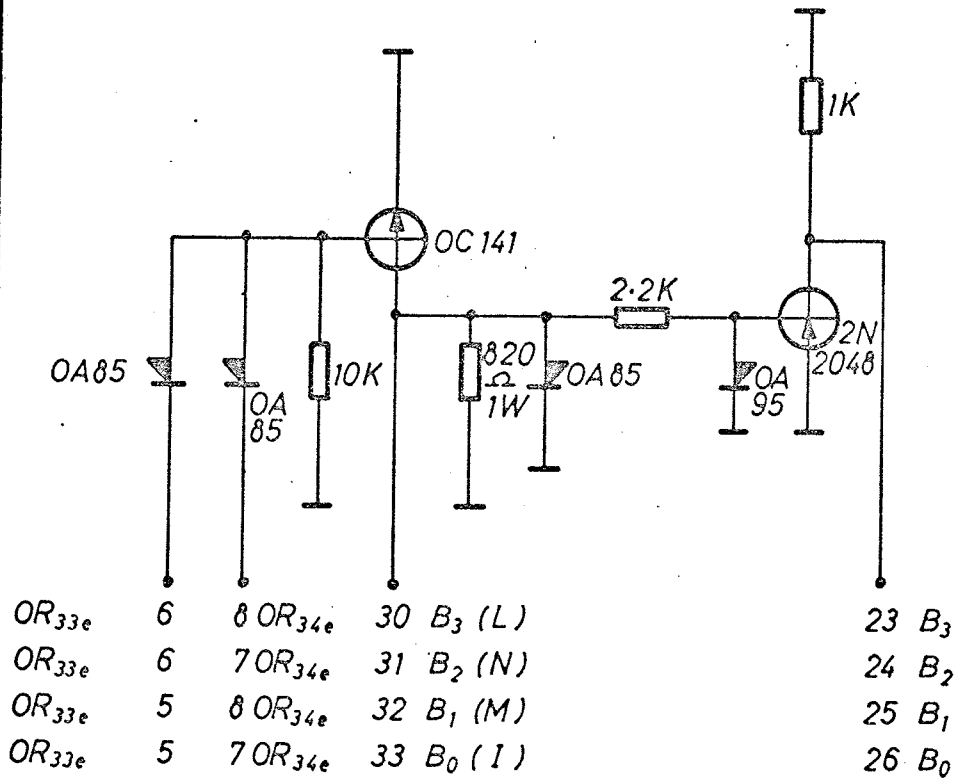
1 Circuit per card.



Unmarked Diodes : OA 85

Unit: <b>GIER 6</b>	Designed <b>H. I.</b>	<b>DECODING OF IA</b>	Drawing No	
 <b>CENTRALEN</b>	Approved		Drawn by <b>G.T. 22.5.66.</b>	
	Checked <b>1. 2. 62.</b>		Checked <b>F.E. 02-8-67</b>	
	Last Revision		<b>1</b> Sheets	Sheet <b>1</b>
		Pos. <b>C 2</b>	<b>0222-1</b>	

4 Circuits per card.

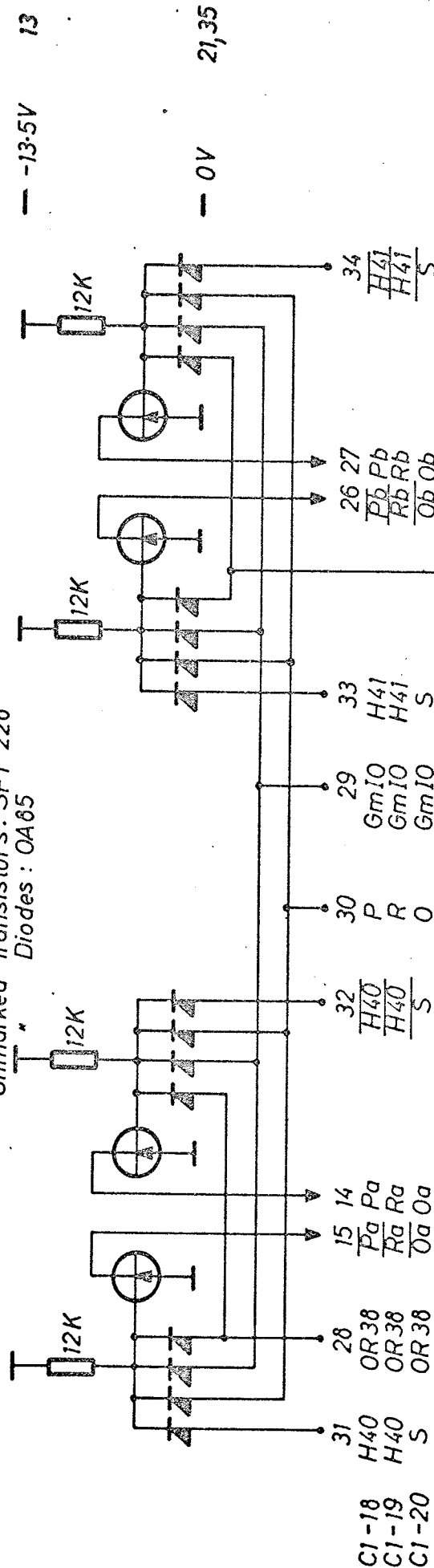


- -13.5V 35  
 - -7.5V 13

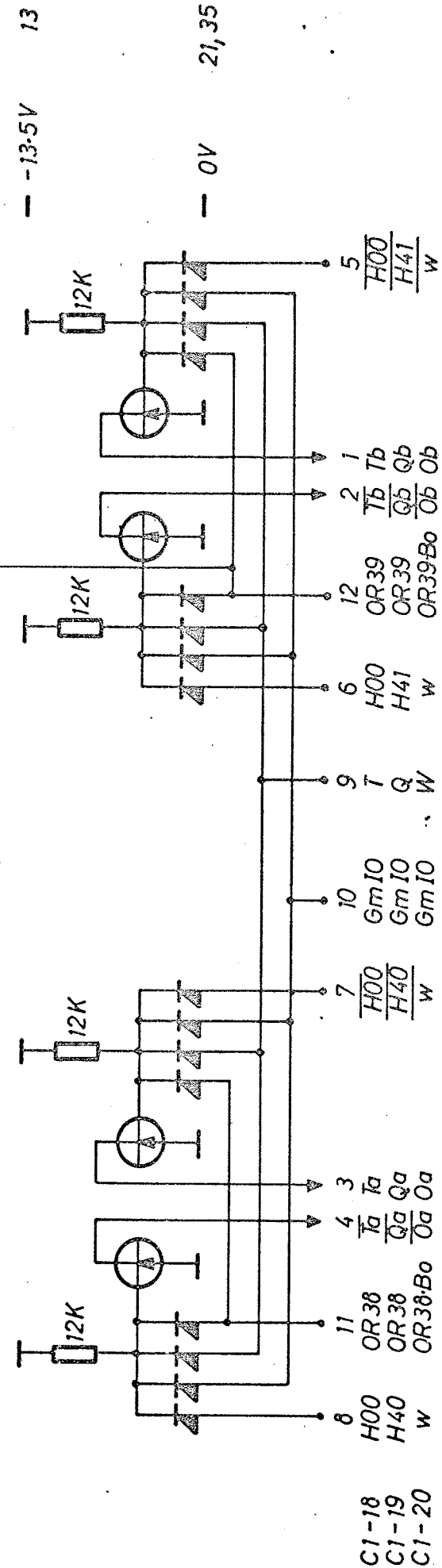
- 0V 14  
 - +4.5V 12  
 - +10.5V 15

Unit: GIER 6	Designed H. I.	DECODING OF 10	Drawing No	
	Approved		Drawn by G.T. 22.5.66	
	Checked 1.2.62.		Checked F.E. 22-2-67	
	Last Revision		1 Sheets	Sheet 1
		Pos. C2-23 0222-2		

Unmarked Transistors: SFT 220  
Diodes: OA65



C1-18  
C1-19  
C1-20



C1-18  
C1-19  
C1-20

S ~ overflow from overflow fl. fl., w = 0 for (AC) ≠ 0, W = 1 for (AC) = 0, Gm10 ~ set indicator.

Unit: GIER 6	Designed H.L.
	Approved
	Checked 1.2.62.
	Last Revision

### CONTROL OF INDICATOR

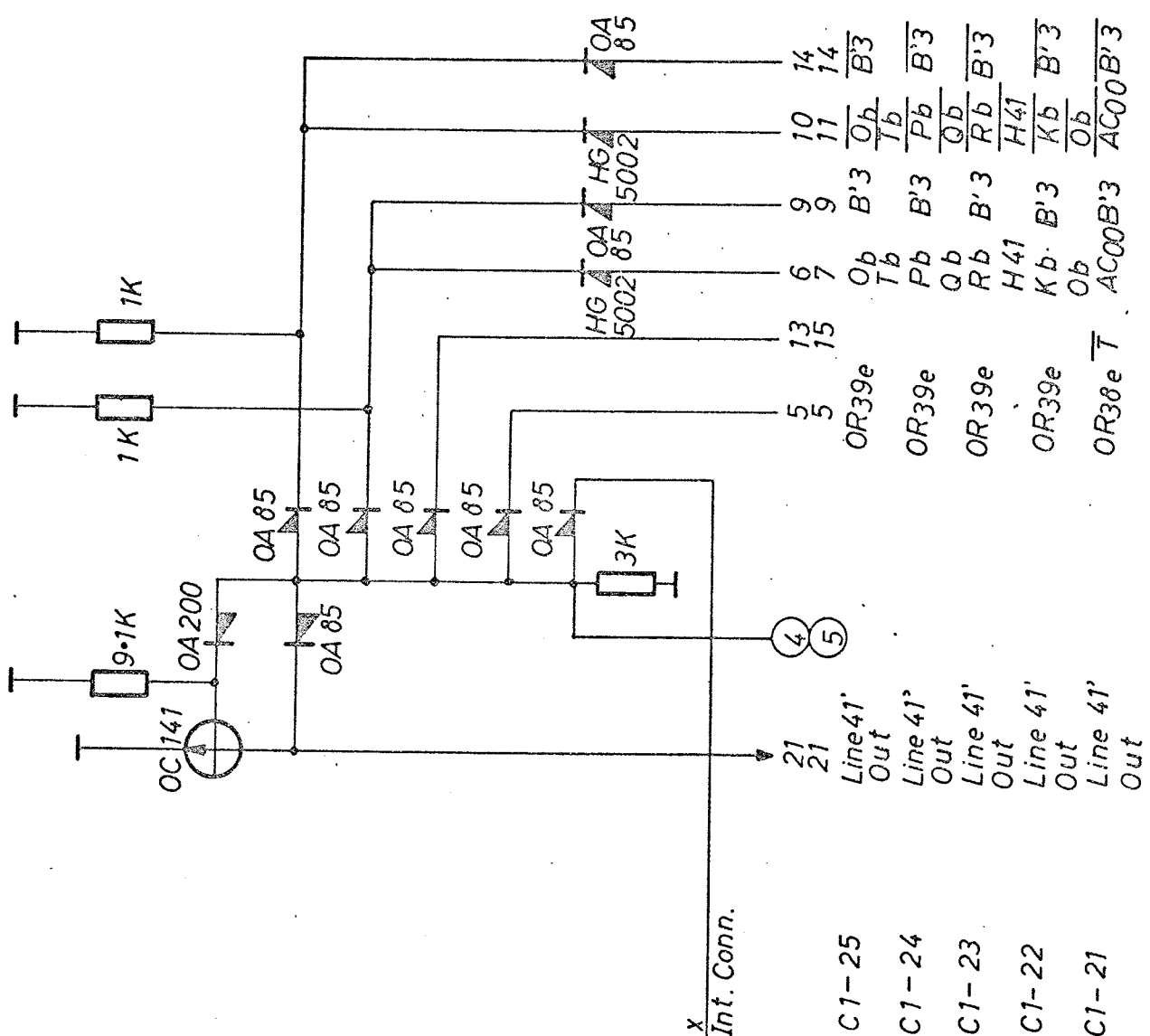
Drawing No	
Drawn by G.T. 13.6.66	
Checked F.E. 22-2-67	
1 Sheets	Sheet 1
P C1	0231

2 Circuits per Card

-13.5V 23

-3V 22

-10.5V 25



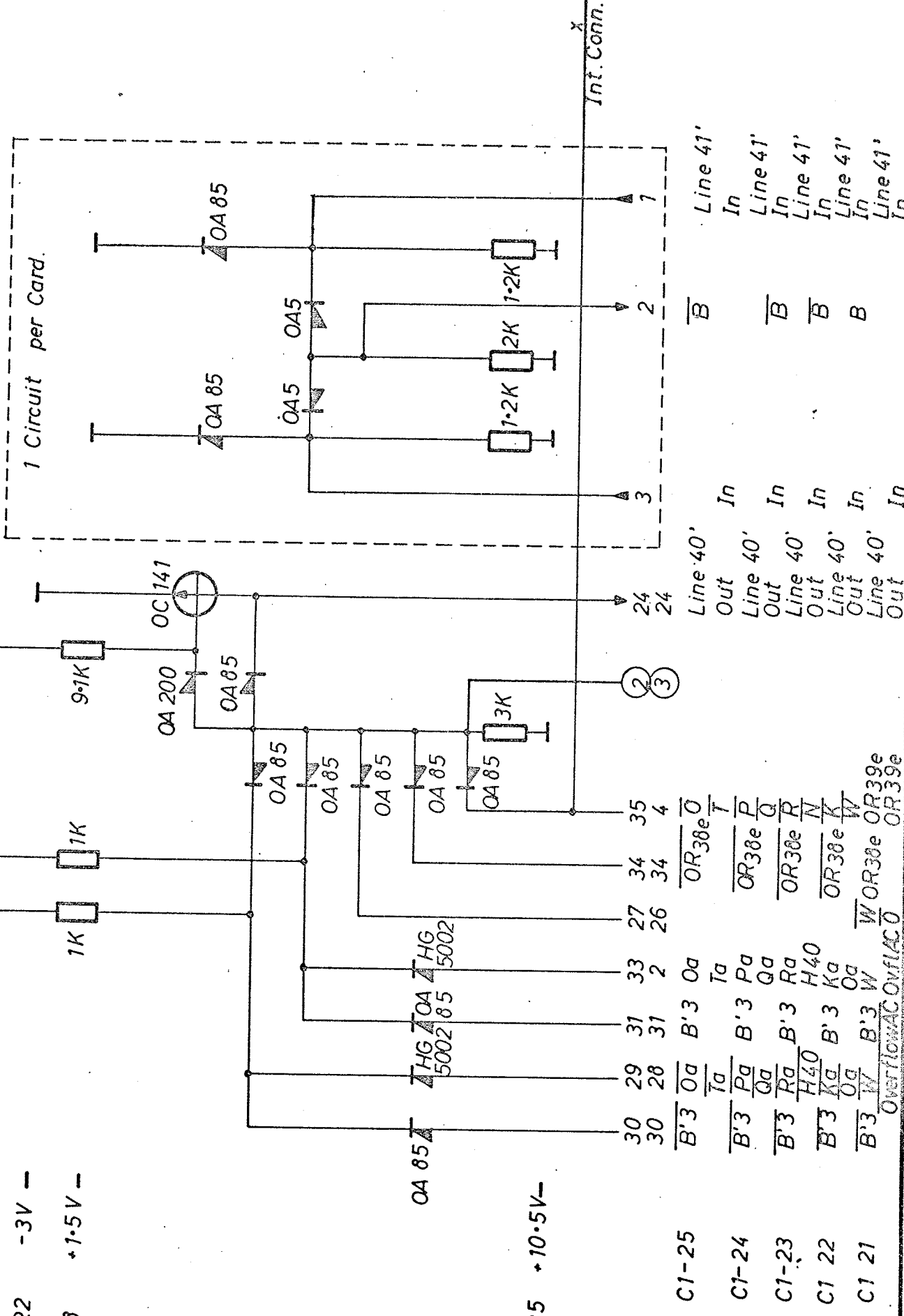
	Unit: GIER 6	Designed H.I.	Drawing No	
	Approved		Drawn by B.R. 6.10.66.	
	Checked 1. 2. 62.		Checked F.E. 22-2-67	
	Last Revision		2 Sheets	Sheet 1
<b>INDICATOR OPERATIONS</b>			Pos. C 1	0233

23 -13.5V-  
 22 -3V-  
 8 +1.5V-

25 +10.5V-

2 Circuits per Card

1 Circuit per Card



W = 0 For (AC) ≠ 0  
 W = 1 For (AC) = 0

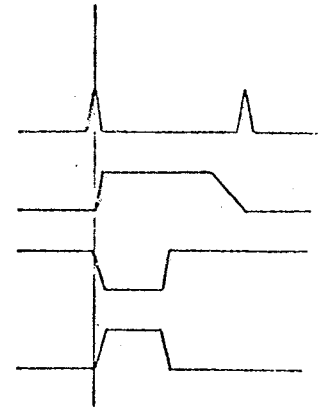
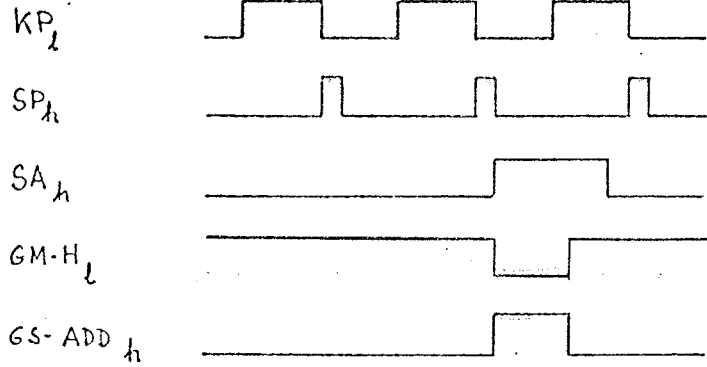
Unit: GIER 6  

 REGNE  
 CENTRALEN

Designed H. I.  
 Approved  
 Checked 1. 2. 62.  
 Last Revision

INDICATOR  
 OPERATIONS

Drawing No	
Drawn by B.R. 6. 10. 68	
Checked F.E. 22-2-67	
2 Sheets	Sheet 2
Pos. C1	0233



ADD.

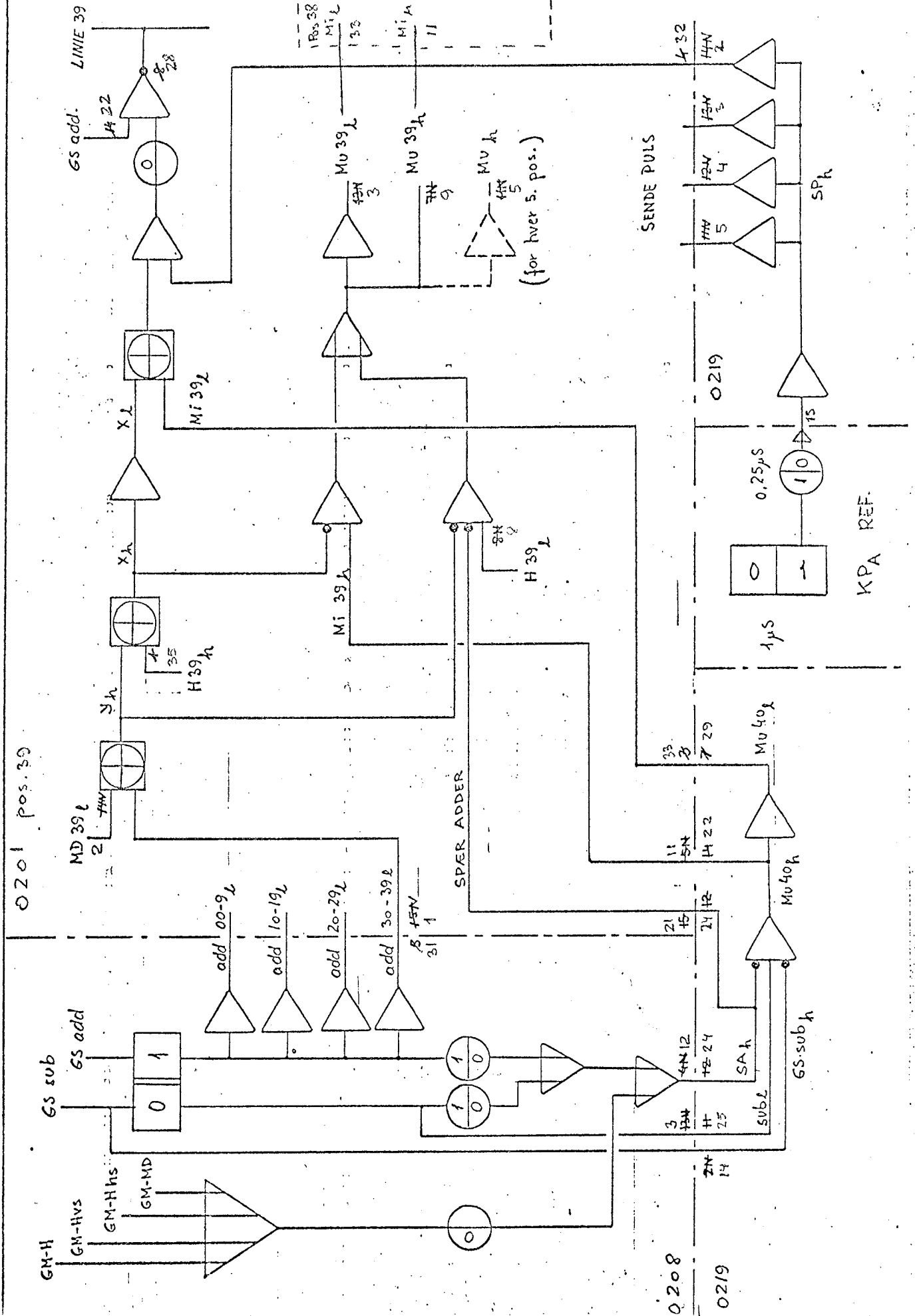
MD	0	1	0	0	1	1	0	1
H	0	0	1	0	1	0	1	1
Mi	0	0	0	1	0	1	1	1
y	1	0	1	1	0	0	1	0
X	1	0	0	1	1	0	0	1
S	0	1	1	1	0	0	0	1
Mu	0	0	0	0	1	1	1	1

$y \sim MD=0$        $\bar{y} \sim MD=1$   
 $X \sim MD=H$        $\bar{X} \sim MD \neq H$

SUB.

MD	0	1	0	0	1	1	0	1
H	0	0	1	0	1	0	1	1
Mi	0	0	0	1	0	1	1	1
y	0	1	0	0	1	1	0	1
X	0	1	1	0	0	1	1	0
S	1	0	0	0	1	1	1	0
Mu	0	0	1	1	0	0	1	1

$y \sim MD=1$        $\bar{y} \sim MD=0$   
 $X \sim MD \neq H$        $\bar{X} \sim MD=H$





ADDER

15/11 1965

LH

3/

Halvodderen bestående af de to emitterfølgere OC47 og  $750\Omega$ , vil såfremt  $x = M_i$ , have samme spænding på base og emitter og begge OC47 vil være blok.

Basen af den efterfølgende OC141 vil ligge på  $\approx 14$ , og denne vil være blokeret. Næste OC141 vil derfor lede, og basen af GS-transistoren vil være  $\approx 3$ , således at denne vil være blok. og linien vil forblive "1" under GS.

Såfremt  $x \neq M_i$  vil een af OC47 have  $\approx 7.5$  på basen og  $\approx 4.5$  på emittermodst. Transistoren vil lede og emitteren gå på  $\approx 7.5$  og over de  $750\Omega$  kan derfor leveres  $\frac{7.5 - 4.5}{0.75} = 4 \text{ mA}$ . Hvis SP er negativ ( $\approx 15$ ) skal disse 4mA leveres gennem  $12k \neq 3.3k \sim 2.5k$  og vil give et spænd. fald på  $2.5 \cdot 4 = 10 \text{ V}$ , hvilket ville give en spænd. på  $\approx 21.5$  på col. af disse OC47, men disse er klampt til  $\approx 13.5$  med en OA160, som leverer den resterende strøm, således at punktet holdes på  $\approx 14$ .

Herved er den efterfølgende OC141 spærret.

Når SP går pos. ( $\approx 7.5$ ) vil strømmen til de  $3.3k$  leveres af denne. De 4mA fra OC47 skal da kun leveres gennem de  $12k$ . Basen af OC141 kan ikke gå mere pos. end  $\approx 13$  før denne vil lede, og ved denne spænding vil der i  $12k$

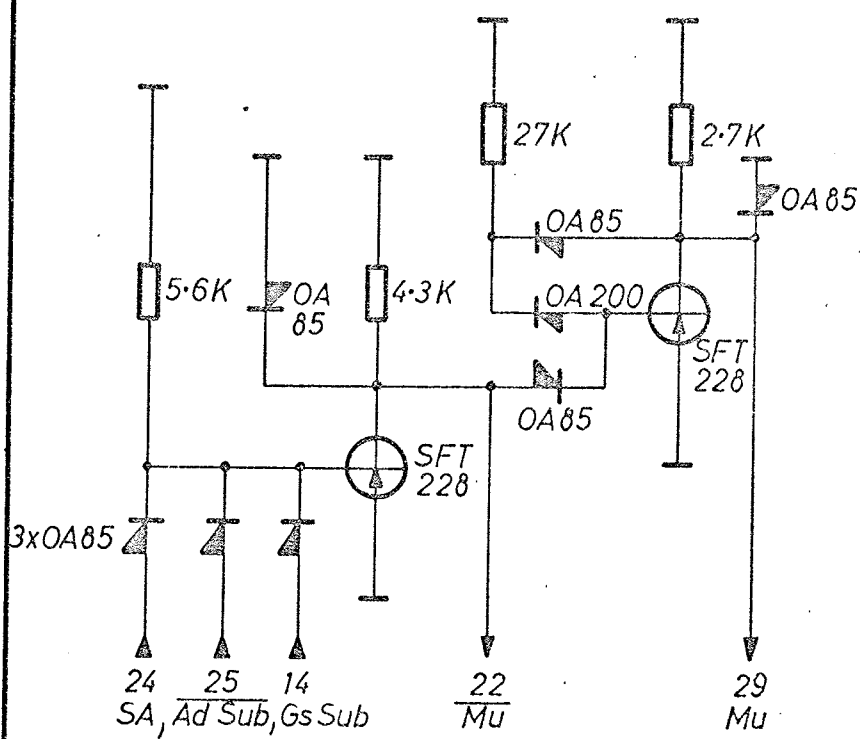
ADDER

15/11 1965

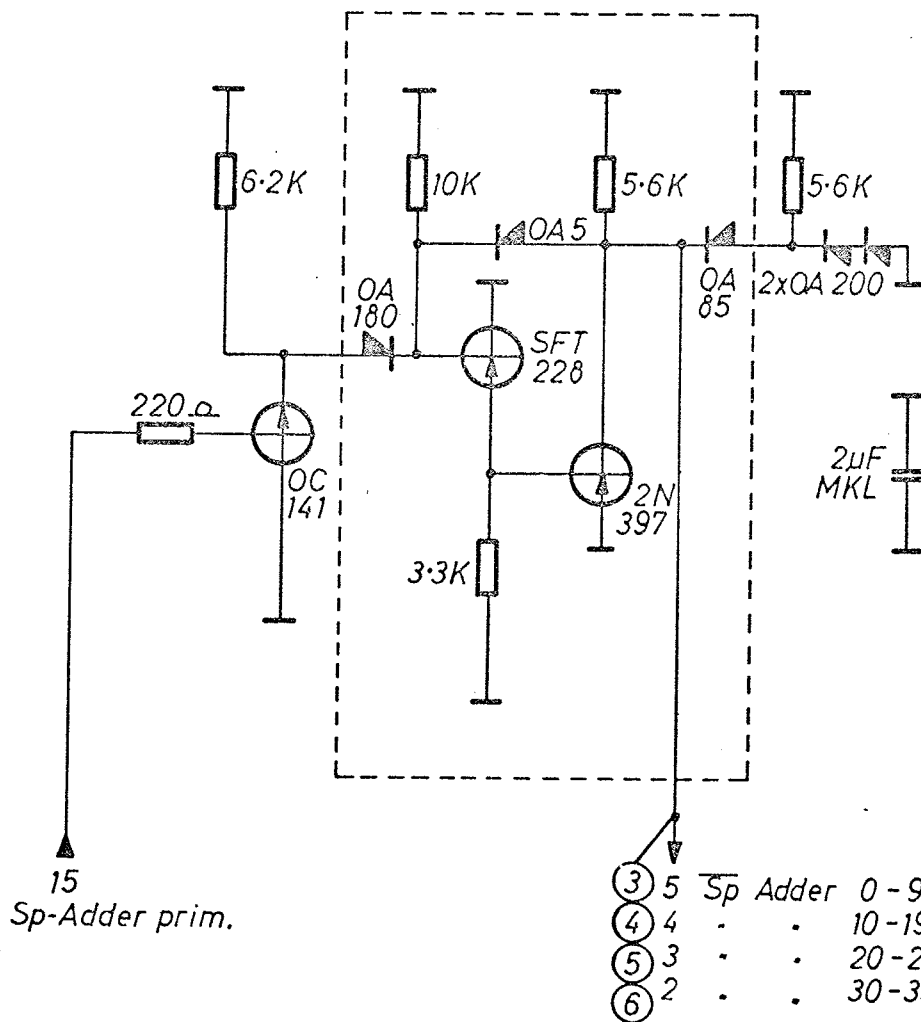
KH

4

gå  $\frac{31,5 \div 13}{12} = 1,5 \text{ mA}$ , de resterende  $4 \div 1,5 = 2,5 \text{ mA}$   
er da til rådighed for basen på OC141 som vil lede  
og trække basen på den efterfølgende OC141 på  
 $\approx 13,5$  og samtidig oplade kondensatoren på  
300 pF. Denne OC141 vil være blok. fra SP går  
pos. og indtil kondensatoren er afladet. (ca  $2 \mu\text{s}$ )  
I dette tidsrum vil GS-Adder gå pos. og basen af  
GS-transistoren følge med således at denne leder  
og linien bliver trukket på  $\approx 1,5$  d.v.s. "0"



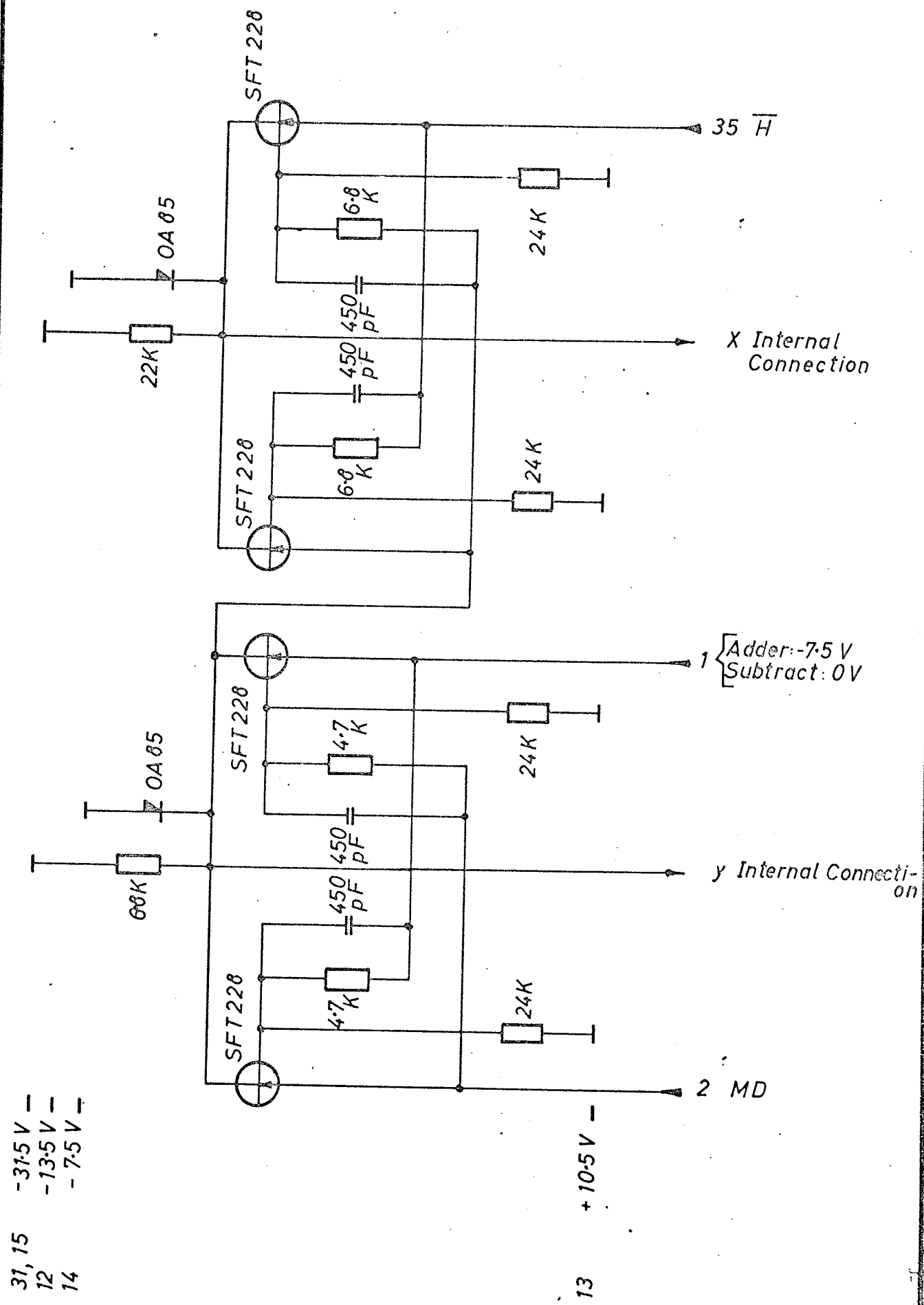
-31.5V	1,35
-13.5V	7
-7.5V	6
-4.5V	27
-1.5V	21



-31.5V	1,35
-13.5V	7
0V	34
-7.5V	6
-3V	9
+10.5V	8

③	5	Sp Adder	0-9
④	4	.	10-19
⑤	3	.	20-29
⑥	2	.	30-39

Unit: GIER 6	Designed H. I.	<b>SP - ADDER CARRIE 39</b>	Drawing No	
	Approved		Drawn by G.T. 20. 5. 66.	
	Checked 1. 2. 62.		Checked F.E. 23-2-67	
	Last Revision		1 Sheets	Sheet 1
		Pos. B3-49	0219	



31, 15 -31.5 V -  
 12 -13.5 V -  
 14 -7.5 V -

13 +10.5 V -

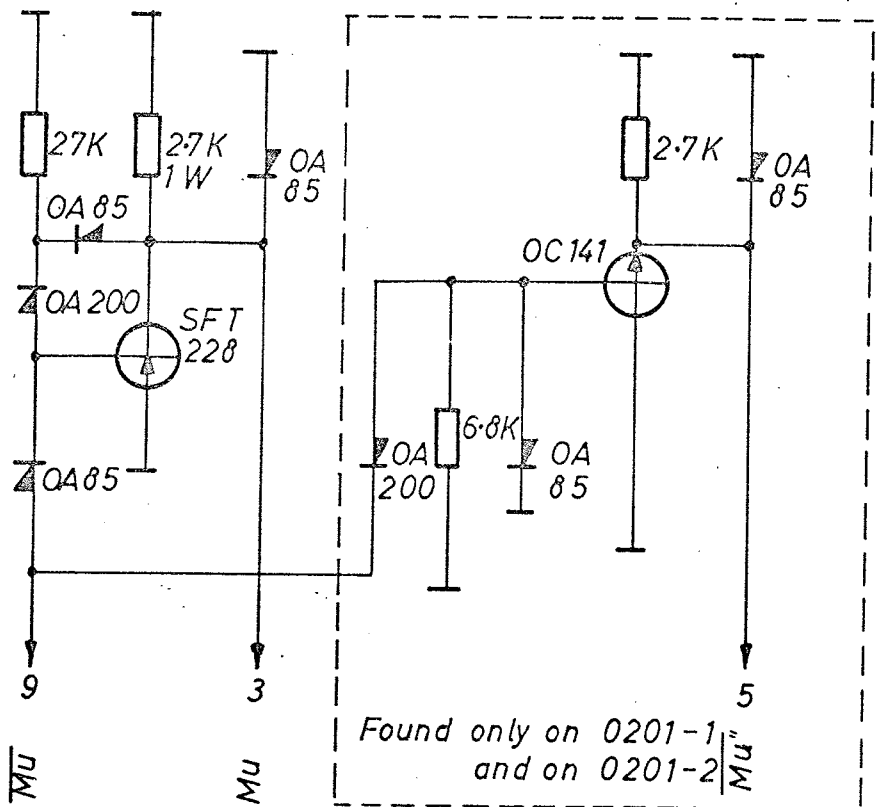
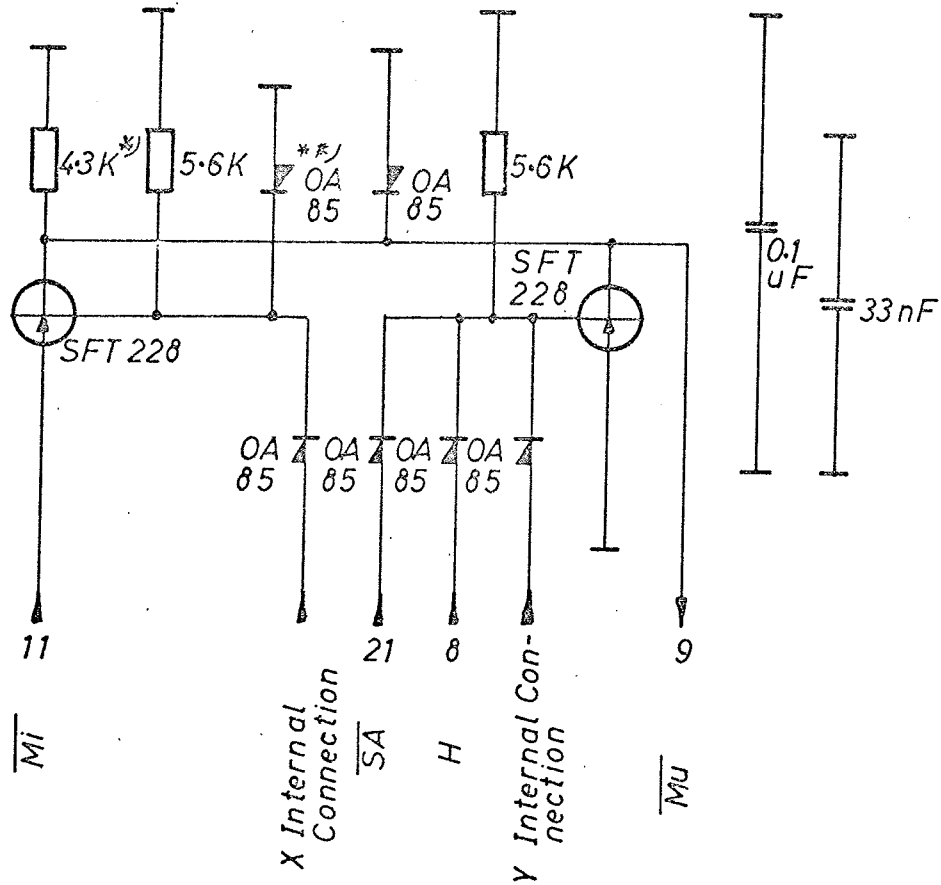
Unit: GIER 6 	Designed H.I.	<h1>ADDER</h1>	Drawing No	
	Approved		Drawn by B.R. 13. 1. 66.	
	Checked 1.2.62.		Checked F.E. 22-2-67	
	Last Revision		3 Sheets	Sheet 1
			A4-B3	02018-1-2

12 - 13.5 V ---  
 14 - 7.5 V ---  
 34, 4 - 4.5 V ---  
 30, 24 - 3 V ---

6 0 V ---  
 7 - 1.5 V ---

31, 15 - 31.5 V ---  
 14 - 7.5 V ---

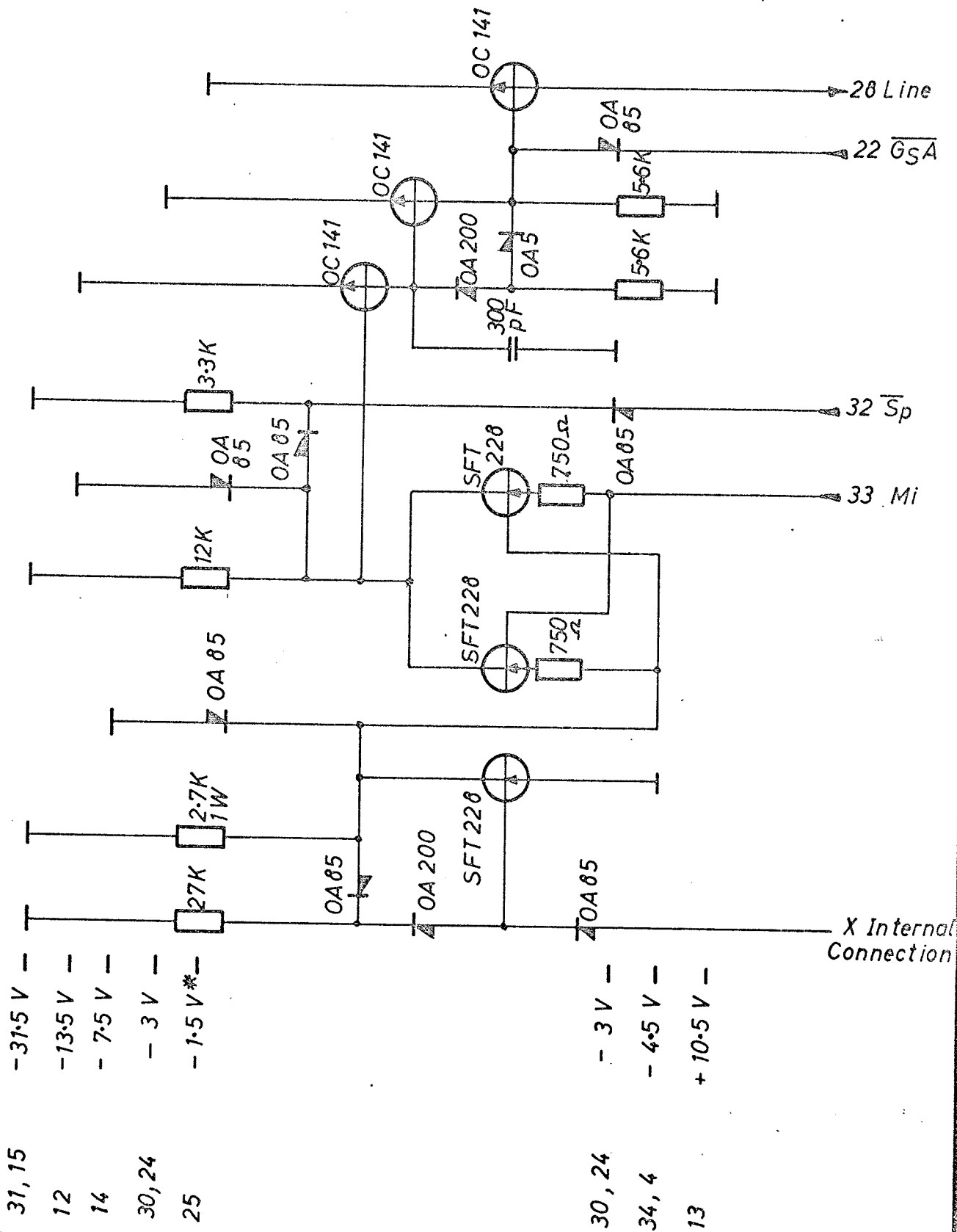
34, 4 - 4.5 V ---  
 7 - 1.5 V ---  
 6 0 V ---  
 13 + 10.5 V ---




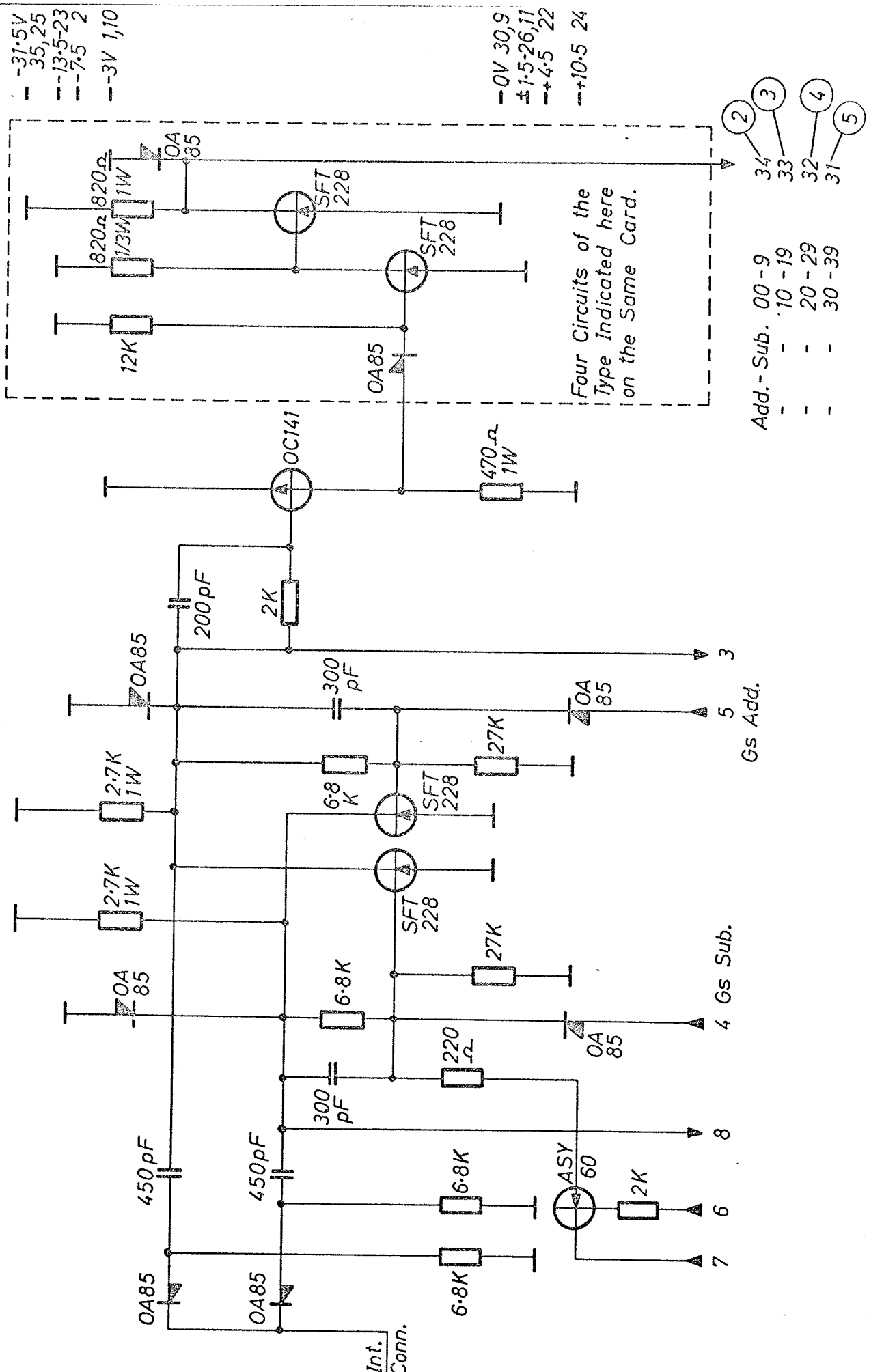
\*\*\*) Found only on 0201-2

\*) Value 5.1K and Connected to -31.5V on Cards 0201-1 and 0201-2

Unit: GIER 6	Designed H.I.	Drawing No
	Approved	Drawn by B.R. 13.1.66.
	Checked 1.2.62.	Checked F.E. 23-2-67
	Last Revision	3 Sheets Sheet 3
		Pos. A4-B3 0201&-1 -2
ADDER		



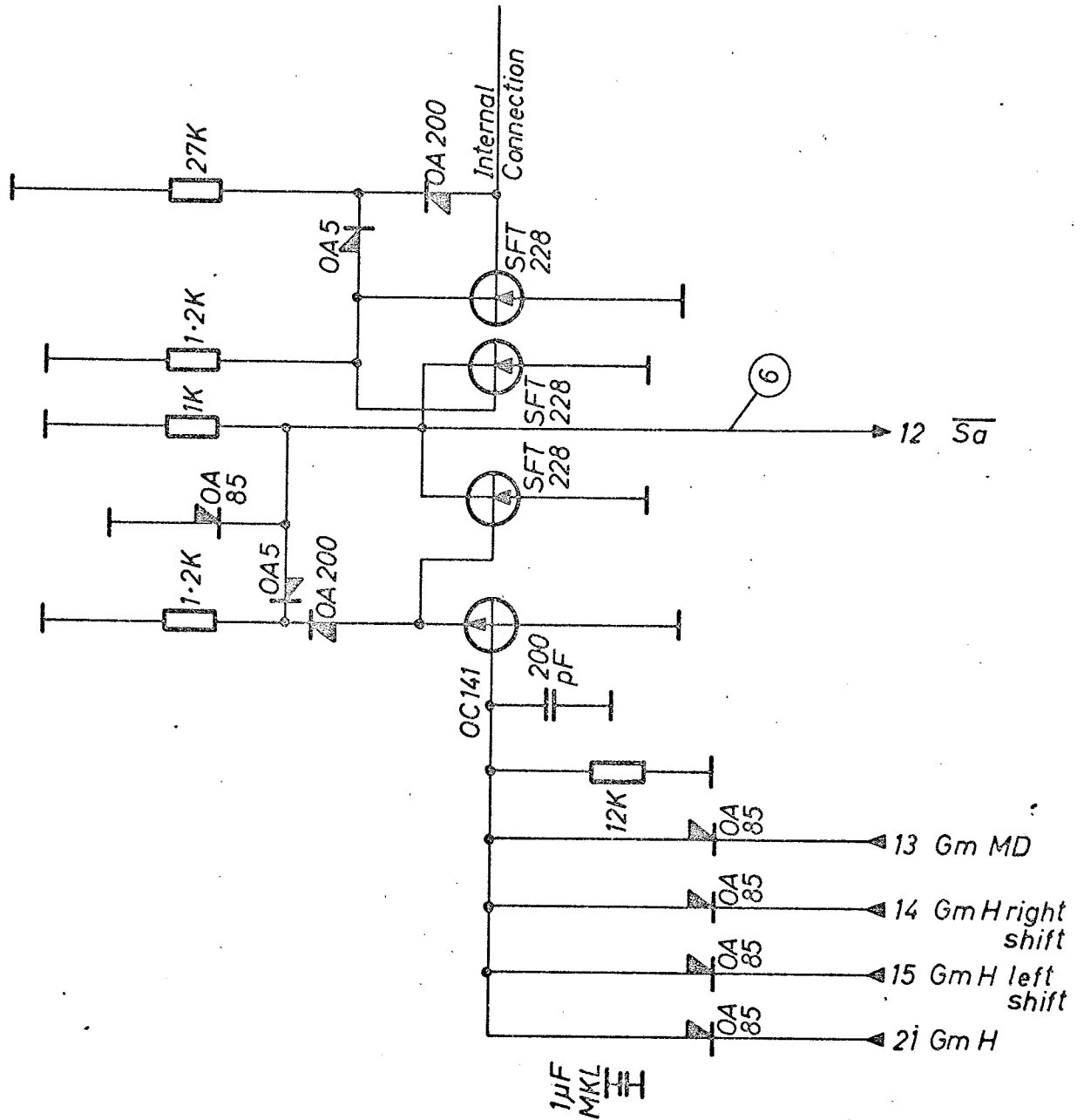
	Unit: GIER 6	Designed H.I.	<h1>ADDER</h1>	Drawing No	
		Approved		Drawn by B.R. 13.1.66.	
		Checked 1.2.62.		Checked F.E. 02-0-67	
		Last Revision		3 Sheets	Sheet 2
				Pos. A4-BJ 02018-1.2	



	Unit: GIER 6	Designed H. I.	Drawing No	
	Approved	Checked 1.2.62	Drawn by G.T. 16.1.67	
	Last Revision		Checked F.E. 22-2-67	2 Sheets
			B3-28	Sheet 2
<b>SA AND ADD. - SUB. - FL. - FL.</b>			Sheet 2 0208	

- -31.5V 35, 25  
 - -13.5V 23  
 - -3V 21, 10

- 0V 30, 9  
 - +1.5V 26  
 - +4.5V 22  
 - +10.5V 24



	Unit: GIER 6	Designed H I	Drawing No	
	Approved	Checked 1.2.62	Drawn by G.T. 17.1.67	
	Last Revision		Checked F.E. 23-8-67	
			2 Sheets	Sheet 1
SA AND ADD. - SUB. -FL. FL.			B3-20	0700




Unmarked Diodes: 0A 95

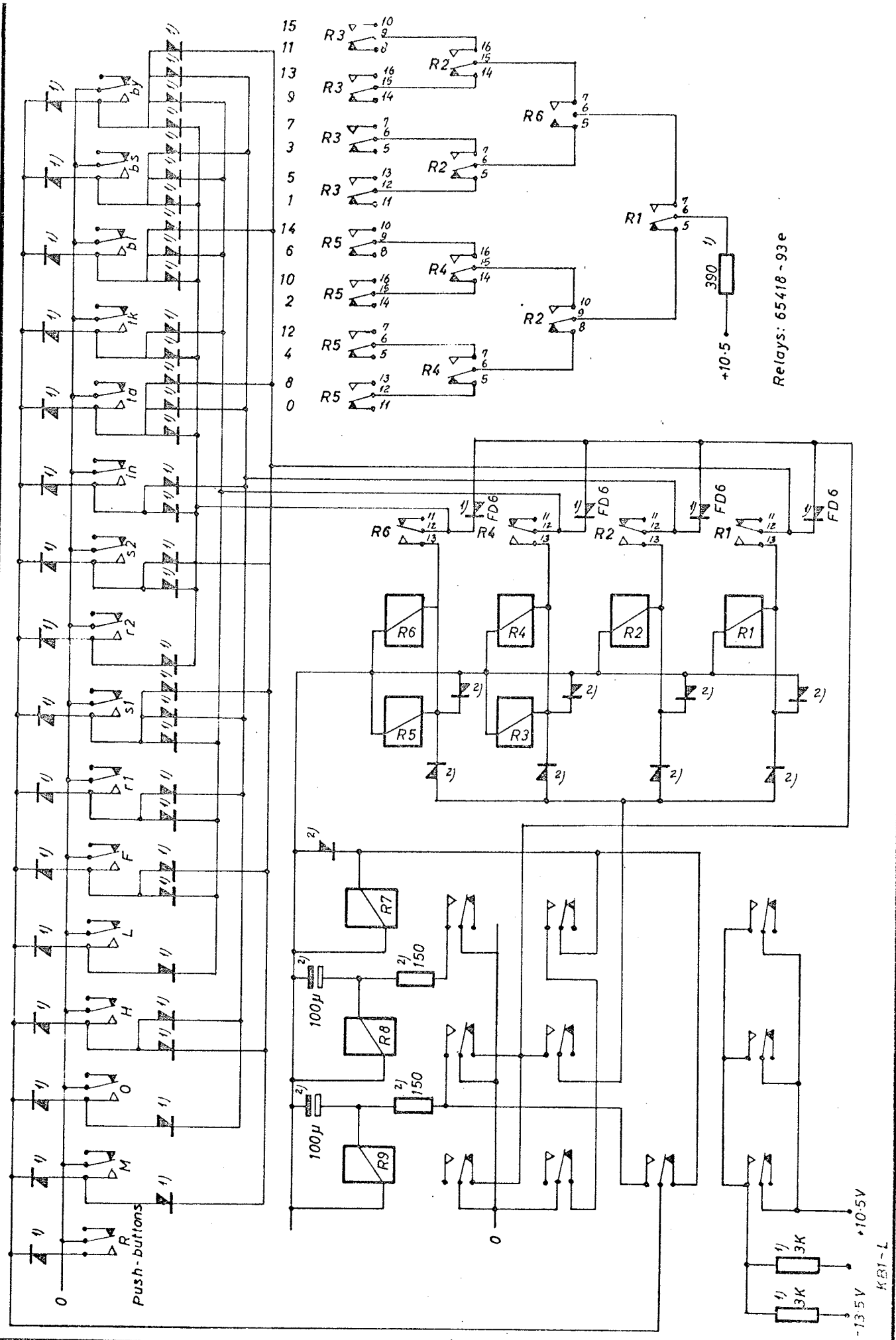
- 1) Mounted on Card 0255
- 2) Mounted on Relays


R correspond with input 0 on Card 0255

H -	-	-	1 -	-	-
O -	-	-	2 -	-	-
H -	-	-	3 -	-	-
L -	-	-	4 -	-	-
F -	-	-	5 -	-	-
r1 -	-	-	6 -	-	-
s1 -	-	-	7 -	-	-
r2 -	-	-	8 -	-	-
s2 -	-	-	9 -	-	-
in -	-	-	10 -	-	-
ta -	-	-	11 -	-	-
tk -	-	-	12 -	-	-
b1 -	-	-	13 -	-	-
bs -	-	-	14 -	-	-
by -	-	-	15 -	-	-

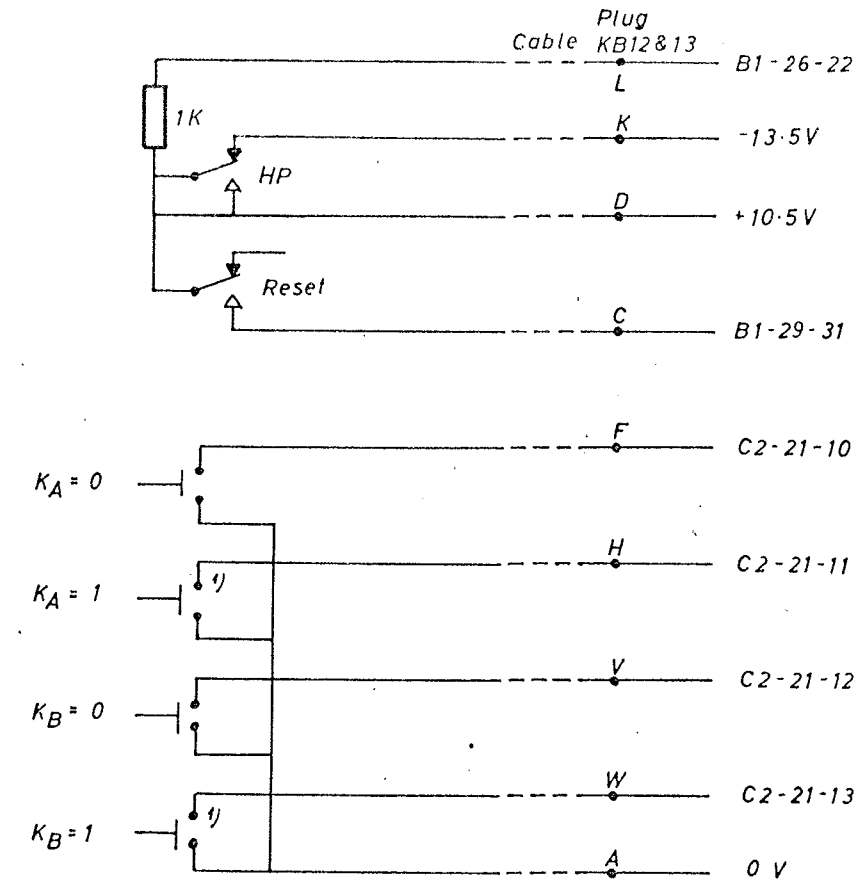
Relay 1-9, mounted in Display Panel  
beginning with Relay farthest from Hinge  
on Relayshef.

Unit: GIER 6	Designed H.I.	Register Selector in KB.	Drawing No		
	Approved		Drawn by 1.2.67	Checked F.E. 7-8-67	
	Checked 1.2.62		2 Sheets	Sheet 2	
	Last Revision				

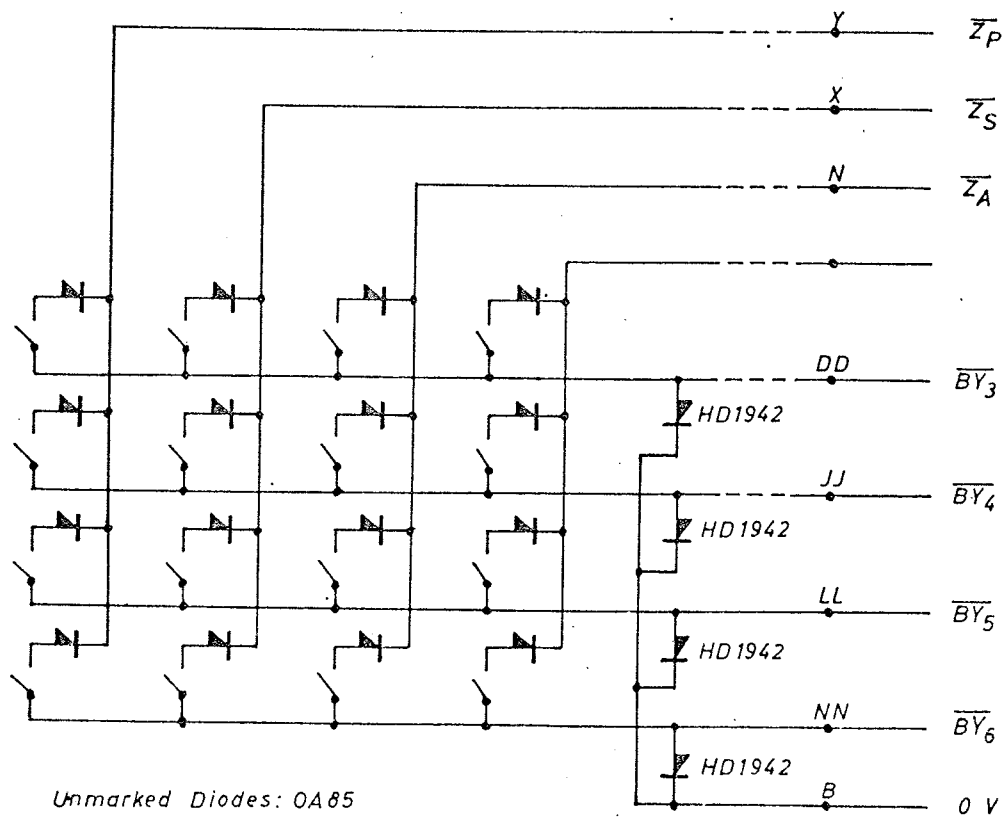


Unit: <b>GIER 6</b>	Designed <i>H.I.</i>	<b>Register - Selector in KB</b>	Drawing No
 <b>CENTRALEN</b>	Approved		Drawn by <i>I.K. 20.4.1967</i>
	Checked <i>1.2.1962</i>		Checked <i>FE. 7-8-67</i>
	Last Revision		2 Sheets Sheet <u>1</u>

DISPLAY PANEL II



⌋ Mounted to Lamp.



Unmarked Diodes: 0A85

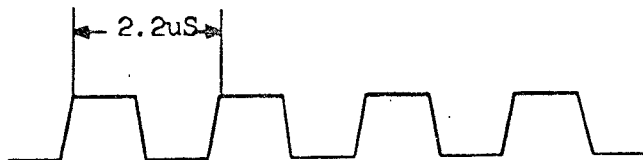
Unit: GIER 6 	Designed H.I.	Push-Buttons in KBII	Drawing No	
	Approved		Drawn by I. K. 28.4.1962	
	Checked 1.2.1962.		Checked F.E. 7-8-67	
	Last Revision		1 Sheets	Sheet 1

ADJUSTMENTS FOR GIER

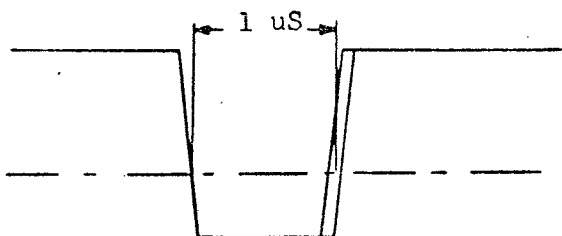
CLOCKPULSE

0249 (B1-27)

The oscillator frequency is adjusted by changing the capacitance (680 pF)



The clock pulse length can be adjusted by the potentiometer on card 0249 and should be 1 uS when measuring GM-H<sub>0-9</sub> as it passes 0 V.

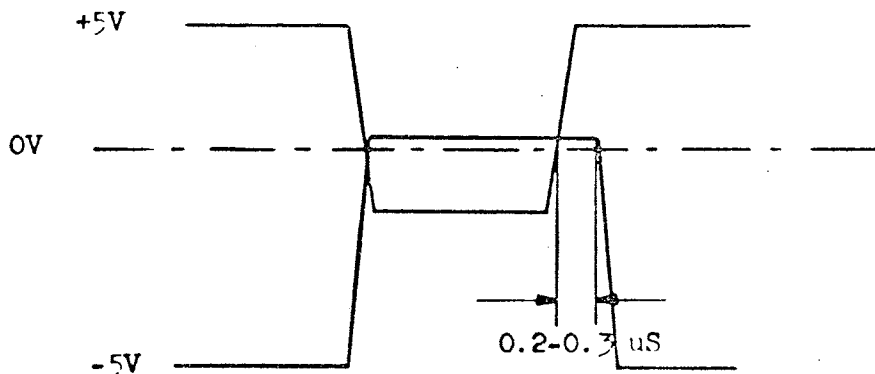


Marginal pulse length is 0,9 uS to 1,1 uS. ( $\pm 10\%$ )

SP-H and MQ

SP-H should start at the same time as GM-H and should lag behind 0.2-0.3 uS.

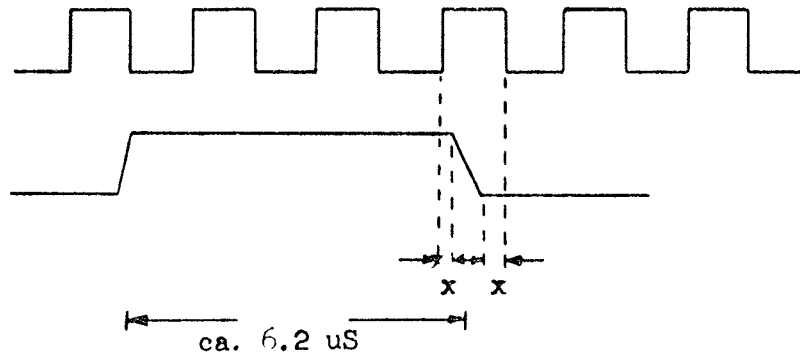
Adjustment is made on 0293 (A1-17) by changing the capacitance.



VAB:

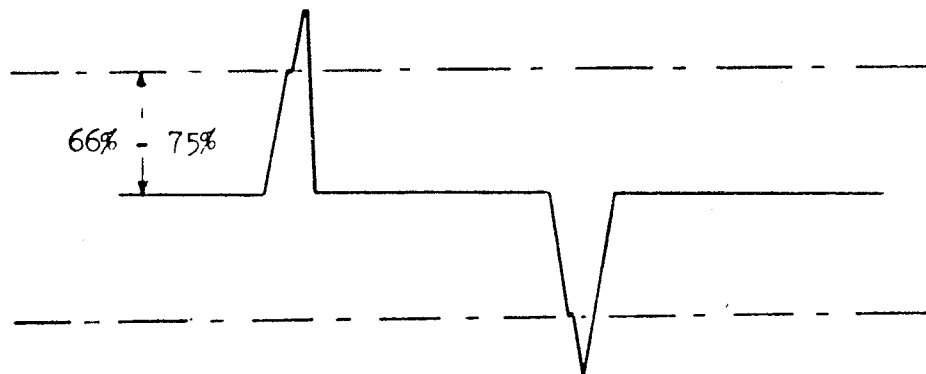
0258 (A1 - 25)

The shift from  $KP_A$  to  $KP_B$  should take place on the trailing edge of a clock pulse. FLF can be adjusted on the potentiometer on A1-25, and should return to zero in the middle of a clock pulse after approx 6.2uS i.e. in the third clock pulse after start.



DRUM

KP I, II and III amplifiers should be adjusted so that triggering takes place between 66% and 75% of the pulse amplitude. Adjustment is made with the potentiometer on 0257.



Signal amplifier is adjusted with the potentiometer on 0257-1. During drumtest (all tracks involved) the potentiometer at first is slowly turned clockwise until an error appears, then counter-clockwise until an error again appears. The potentiometer is now set in the middle of these two extremes.

FORWARD/BRAKE COILS:

Current in these coils is adjusted to 3 amp. with help of the variable resistor 4.7 60W, mounted under the console.

#### VOLTAGE

All voltages are adjusted with potentiometers on the control panel, to 100%, and should be measured for voltage drops across the relay contacts. Marginal voltages should be at least  $\pm 7\%$ .

#### THERMAL CUTOUT

Thermal cutout should take place between 33 and 37°, and adjusted with the three potentiometers on card 288 (E7).

Adjustment of Core Store Strobe

~~10000~~ [ GIER ]

The measuring is made in one pos. in ~~FR~~ [ LI ] reg. card ~~1007~~ [ 1024 ]  
to which "1" is read continously from the core store.

Use probe with ground connected on pin ~~15~~ [ ~~serial IV 1 V pin 1N,~~  
serial VI pin 15 ]

Trig: ~~R 1/2 (1007)~~ [ ~~serial IV 1 V R12 B1 31~~<sup>6</sup> / ~~serial VI B1-31-21~~<sup>30</sup> (1006A-14) ]

Ext. neg.

Time: 0,2 uS

Chopped

- A: 0,5V/cm
- B: 2 V/cm
- C: 2 V/cm

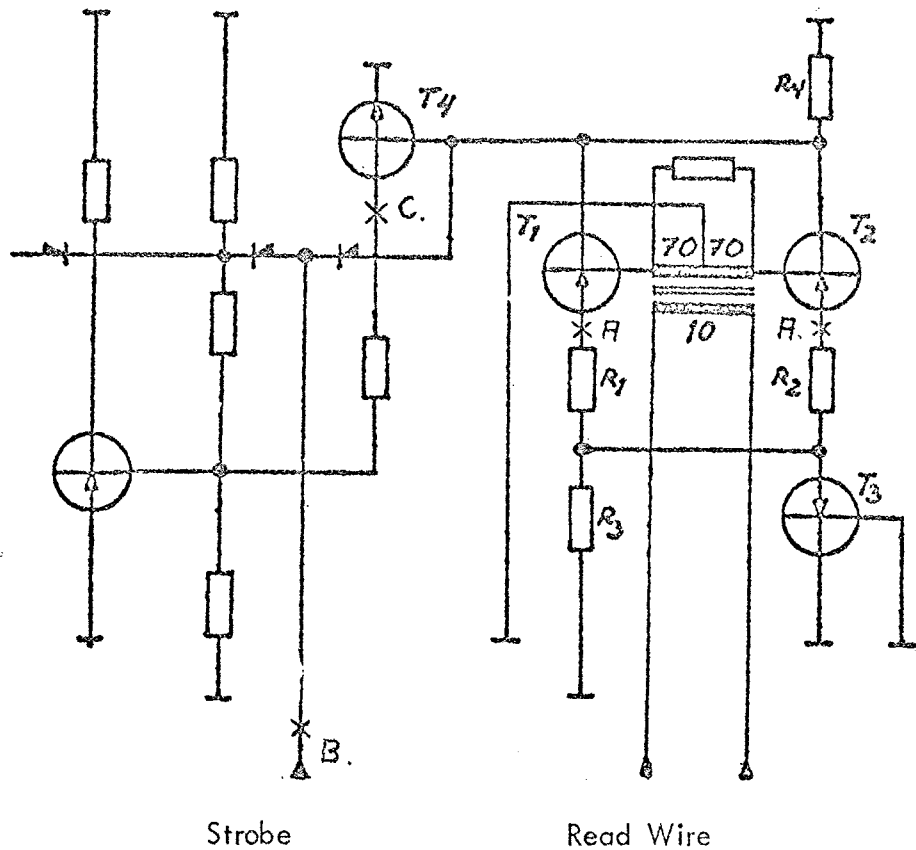


Fig. 1.

Fig. 2 shows the signals obtained in point A-B-C, on Fig. 1.

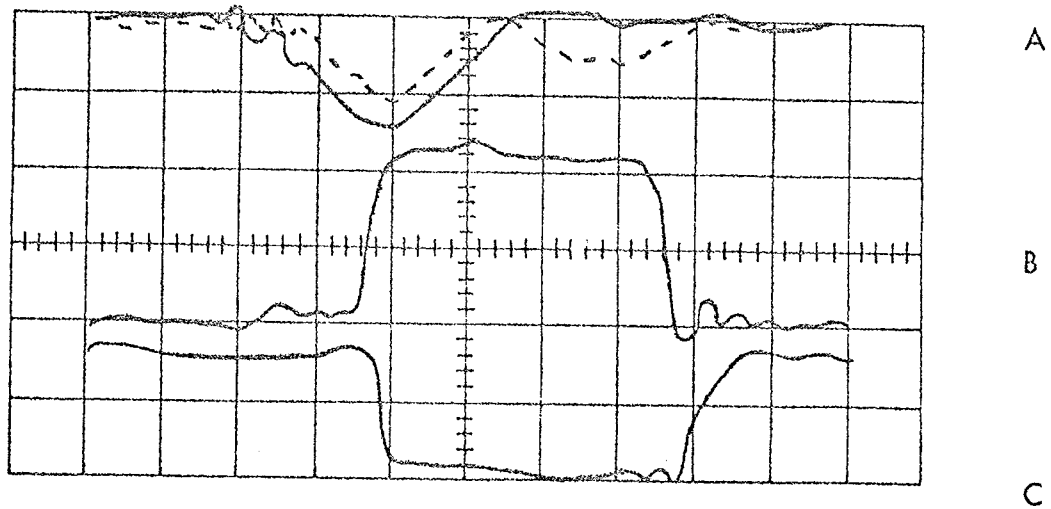


Fig. 2.

If  $\overline{OE}$  [ OT ] is blocked, only one of the two read-signals (A) will be present.

If it is the dotted signal, then change to the other point A.

Adjust the leading edge of the strobe (B) to be in the middle of the read pulse (A) as shown on fig. 2.

Check that the trailing edge of the strobe (B) is coming before the trailing edge of (C). If not, the strobe is too long and adjustment is necessary.

The strobe must be approximately ~~0.75  $\mu$ S~~ [ 1  $\mu$ S ] long.



[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 L0
[ 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. 2, typeout of bit-configuration or bit-pattern. Set the TYPEWRITER to the shortest distance between the lines.

Program is stored from location 41 to 89. The first instruction to be executed is zq LKB. The program will stop after 20 lines have been typed if KB=1. The typeout will be repeated by pressing START-button.]

```

i=41
[ 41] zq LKB ;stop if KB=1
[ 42] vy16 ,hs 50 ;choice of typewriter and jump
[ 43] qq 70 t1 ;constant and counter
[ 44] bt 19 t-1 ;is performed-
[ 45] hh r-3 ;20 times
[ 46] zq LKB ;stop if KB=1
[ 47] pa r-3 t19 ;set number of words
[ 48] pa r-5 t70 ;set start address
[ 49] hh r-7 ;go to repeat
[ 50] arn(s1) IPC ;typeout program. Fetch a word
[ 51] sy 64 , sy 60 ;type CR and upper case
[ 52] sy 6, tk 1 ;type [ and shift
[ 53] sy 57 V LT ;type I if sign is-
[ 54] sy ;else type space
[ 55] bt 39 t-1 ;is performed
[ 56] hh r-4 ;40 times
[ 57] pa r-2 t39 ;set number of bits
[ 58] sy 57 V LPA ;type I if A-mark
[ 59] sy ;else type space
[ 60] sy 57 V LPB ;type I if B-mark
[ 61] sy ;else type space
[ 62] sy 7,hr s1 ;type ] and jump back
[ 63] qq ;empty
[ 64] qq
[ 65] qq
[ 66] qq
[ 67] qq
[ 68] qq
[ 69] qq
[ 70] qq s1 IB t1 ;1 word in bitpattern
[ 71] qq p3 IC t3
[ 72] qq (p7) IKC t7
[ 73] qqn (p15) IOC t15
[ 74] zqn (p31) IRC t31
[ 75] srn (p63) MRC t63
[ 76] scn (p127) LRC t127
[ 77] nkn (p255) D LRC t255
[ 78] ppn (p511) VD LRC t511
[ 79] udn (p-1) XVD LRC t-1
[ 80] udn (p-1) XVD LRC t-1
[ 81] udn (r-2) XVD LRA t-2
[ 82] udn (-4) XVD LR t-4
[ 83] udn -8 XVD LQ t-8
[ 84] ud -16 XVD LT t-16
[ 85] zl -32 XVD L t-32
[ 86] hh -64 XVD N t-64
[ 87] hv -128 XVD t-128
[ 88] pc -256 XV t-256
[ 89] pa -512 X t-512 ;last word in bitpattern

```

e41

[Testprogram no. 3. Typeout of track no. 0 in bit-pattern. Set the TYPEWRITER to double distance between the lines.

The testprogram is stored in locations 41 to 63, it uses the locations 64 to 103 for storing. The first instruction to be performed is zq LKB. The run can be repeated by pressing the START-button.]

```

i=41
[ 41] zq LKB                ;stop if KB=1
[ 42] vk 0,lk 64           ;choose and read track no. 0
[ 43] vy 16,vk 0          ;choose typewriter and wait for the drum
[ 44] sy 62,hs 51         ;black ribbon and jump to typeoutprogram
[ 45] qq 64 t1            ;constant and counter
[ 46] bt 39 t-1           ;is performed-
[ 47] hh r-3              ;40 times
[ 48] pa r-3 t64          ;set start address
[ 49] pa r-3 t39          ;set number of words
[ 50] zq 0,hv r-7         ;stop and go to restart
[ 51] arn (s1) IPC        ;typeoutprogram.Fetch a word
[ 52] sy 64,sy 60         ;type CR and upper case
[ 53] sy 6,tk 1           ;type [ and shift
[ 54] sy 57 V LT         ;type I if sign is-
[ 55] sy 4                ;else type =
[ 56] bt 39 t-1           ;is performed-
[ 57] hh r-4              ;40 times
[ 58] pa r-2 t39          ;set number of bits
[ 59] sy 57 V LPA         ;type I if A-mark
[ 60] sy 4                ;else type =
[ 61] sy 57 V LPB         ;type I if B-mark
[ 62] sy 4                ;else type =
[ 63] sy 7 ,hr s1        ;type ] and jump back
[ 64] qq                  ;empty
e41

```

[Testprogram no. 4. Test of the fast memory or core-store. The content of M is stored in a location and is read back. If the read is equal to the written, the program will continue with the next location. The test is performed for the locations 60 - 1022. If KB=1 the program will stop and a new content can be placed in M. Position 40 and 41 in a location are not included in the test. By errors <f> is typed on the typewriter every time a content of a location is not hit right. If KA=1 GIER will stop while the number of the location that was hit wrong is placed in the in-register, and the difference between the read and the written is placed in the R-register. By restart or if KA=0 the program will go on testing the next location. The first instruction is zq LKB.]

i=41

[41] zq 0 LKB	;stop if KB=1
[42] gm 43 V	;store M in location no. 43
[43] qq 0	;empty
[44] gm 59 M t1	;store M in running location
[45] arn 43	;add content of location no. 43
[46] sr (44)	;subtract running location
[47] hv 52 NZ	;jump if R≠0
[48] arn 44	;add content of location no. 44
[49] ca 1022	;if number of running location is 1022
[50] pa 44 t59	;set 59 as address in location no. 44
[51] hv 41	;go to restart
[52] vy 16, sy 54	;choose typewriter and type f
[53] pi (44)	;place no. of location in in-register
[54] zq 0 LKA	;stop if KA=1
[55] bt 40 t-1	;until 40 times
[56] hv 48	;jump
[57] pa 55 t40	;else place 40 as address in location 55
[58] sy 64 , hv 48	;type CR and jump
[59] qq 0	;empty

e41

[Testprogram no. 5. Test of track no. 0. Help < 3  
 The program reads drum track no. 0 to the locations 128-167 in the core store and compares for each location with the correct content of track no. 0, that is read in to the locations 70-109 after the program. If there is no error in track no. 0 the message ok will be typed. If there is an error in one or more locations, the program will stop after having typed -i- for errors in positions 0-39 and -m- for errors in the mark-positions. The content of the wrong location can be observed in the R-register and the address of the wrong location in the s-register (read modulo 128). In the M-register is -by errors in pos. 0-39 - placed the difference from the subtraction and - by errors in the mark-bits - the correct mark-bits in position 8 and 9.]

i=41

[41]	zq LKB	
[42]	vy 16 , sy 58	;the typewriter in lower case
[43]	zq LKB	
[44]	vk 0, lk 128	;read track 0 to loc. 128 and on
[45]	vk ,pi	;wait for the drum
[46]	arn 69 IRC +1	;correct run cellcontent to R (markbits indic.)
[47]	sr 127 +1	;subtract next variable location
[48]	gi 66 , pp	;correct marks to location no. 66
[49]	hv 60 NZ	;jump if R ≠ 0
[50]	arn (47) IRC	;read marks indic.
[51]	arn 66 IK	;correct marks to R, read marks to p
[52]	ncp , hv 63	;jump if wrong marks
[53]	arn 47,nc 167	;timeout if 40 locations are tested
[54]	hh 45	;or jump back
[55]	sy 64	;type CR
[56]	sy 38, sy 34	;type ok
[57]	pa 46 +69	;replace address number
[58]	pa 47 +127	;replace address number
[59]	hv 43	
[60]	sy 57 X	;type -i- ,difference to M
[61]	arn (47) , ps(47)	;wrong cellcontent to R,address to s
[62]	zq ,hv 50	
[63]	sy 36 X	;type -m- ,correct marks to M pos. 8 and 9
[64]	arn(47), ps(47)	;wrong cellcontent to R,address to s
[65]	zq , hv 53	
[66]	qq	;location used for storing during run

[From location 70 to 109 is now placed the correct content of track no. 0. In this version it is assumed, that the HP-button is blocked by position 0 in the by-register. For a GIER, where the HP-button is blocked by pos. 3 in the by-register, the program can be used, if you before the run change the right half-word instructions in loc. 72 and 103 to vy r79 and vy 80 ]

i=70  
 [70] qq  
 [71] it 1,pa 10  
 [72] gk 1, vy r-497  
 [73] gi 2 IPC  
 [74] gm 3 MPC  
 [75] vk 32 IOB t-1  
 [76] gr 4 MOB  
 [77] tl -39 , pm 24  
 [78] gr 7, tln -19  
 [79] arn 2, cm -1  
 [80] pmfn r IRB  
 [81] pi 66 IZA t-511  
 [82] hh 17 NZA  
 [83] gp 8 , pp 38  
 [84] vk 319 t-1  
 [85] sk -24 t-40  
 [86] bt 23 t-1  
 [87] hv 14, pp 294  
 [88] gs 6 , gin 9  
 [89] vk 25 , lk -64  
 [90] vk p , lk -73  
 [91] vk(5) ,lk -124  
 [92] ps 41, ud 5  
 [93] ps s-1, ar s-125  
 [94] bs s-1,hv r-1  
 [95] bs (5) IOB t-1  
 [96] pp ( ) , hv 21  
 [97] sy 64 V NZ  
 [98] arn -25, hv -29  
 [99] sy 29, sy 60  
 [100] sy 54, sy 53  
 [101] sy 33, sy 35  
 [102] gr -512 MOB  
 [103] ly 33, vy -496  
 [104] pmm 64 XD IZA  
 [105] tl -7 ,ly r1  
 [106] pi LZA t508  
 [107] xr X IZB  
 [108] hv 35 LZB  
 [109] gr 41 MRC t-1  
 e41

;gk 1,vy r79 for HP blocked by pos. 3 in by-reg.

;ly 33,vy 80 for HP blocked by pos. 3 in by-reg.

[Testprogram no. 6. Test of the core-store including marks.

The program examines the whole core-store except 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  $\neq$  0, existing ones are wrong.

If stop in location 46 (r1=47): R40-41  $\neq$  0, existing ones are wrong.

If stop in location 50 (r1=51): R0-39  $\neq$  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  $\neq$  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 $\neq$ 0
[45] qq V NC	
[46] zq	;stop if R40.41 $\neq$ 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 $\neq$ 0
[51] qq V LC	
[52] zq	;stop if R40.41 $\neq$ 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

\*)  $\begin{matrix} \uparrow \\ \text{11111110000} \end{matrix}$



[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 35271407753020 36242706607000 66043725657530 15577777733744
6000 35271407753020 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. The 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                                ;parameter
[ 54] gcn s9 t320                         ;parameter
[ 55] pi t-65                               ;from 55 to 66 is placed -
[ 56] arn r-2,gr r-13                     ;a subroutine for printing -
[ 57] dln r-14,tk 20                     ;out the counter in location -
[ 58] pi 64 NZ t-65                        ;115 in decimal notation.
[ 59] hv r3 NZ
[ 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,t1 -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,t1 -10 ;fetch number , shift
[ 99] ga r17,sr r17 ;delete bits 0-9
[100] t1 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 ;go to 6000 new cycles

```

[Testprogram no. 8. Test of drum track transfer time.

The time for execution of an LK instruction is tested by the program. If KA is set to 1, the transfer time for all tracks is printed out. If KA=0 there will only be output for tracks with a transfer time different from 20 ms. The output is in binary notation for the transfer times as well as for the track numbers.

After the end of a test run ( output <slut> on the typewriter), the test can be repeated by pressing the NORMAL START-button. If the qq instruction in location 57 is changed to a zq instruction, the actual transfer time can after stop in location 57 be observed in the R-register in microseconds with unit in pos. 19.

First instruction to be executed is zq LKB in location 41.]

```

i=41
[41] zq LKB ;stop if KB=1
[42] vy 16, sy 64 ;select typewriter, CR
[43] sy 58, sy 36 ;lower case, m
[44] sy 18 ;s
[45] bt 10 t-1 ;write 10 SPACE
[46] sy , hv 45
[47] sy 19, sy 41 ;t, r
[48] sy 49, sy 51 ;a, c
[49] sy 34, sy 64 ;k, CR
[50] bt 320 t-1 ;select -
[51] vk -1 V t1 ;- tracks 0-319
[52] hv 75 ;go to 75 if track no. =319
[53] lkn -512 ;read track, zeroes to R
[54] hk 57 ;if transfer finished then go to 57 -
[55] ar 74 ;else add 113 microseconds
[56] hv 54 ;and go to 54
[57] qq ;empty
[58] nc 20 NKA ;output for all tracks if KA=1 -
[59] ps 66 V ;- else only output for tracks -
[60] hv 50 ; - with transfer time ≠ 20 ms
[61] tk 1 ;shift 1 position to the left
[62] sy 1 V LT ;print 1 if sign
[63] sy 38 ;else print 0
[64] bt 9 t-1 ;and repeat 9 times -
[65] hv 61 ;- from 61
[66] hv s1
[67] sy , sy ;2 SPACE
[68] pa 64 t9 ;reset address in loc. 64
[69] arn (51) D ;track no. to R
[70] hs 61 ;print track no.
[71] pa 64 t9 ;reset address in loc. 64
[72] sy 64 ;CR
[73] hv 50 ;go to new track
[74] qq t113 ;number of microseconds
[75] sy 18 ;s
[76] sy 35, sy 20 ;l, u
[77] sy 19, sy 64 ;t, CR
[78] zq ;stop
[79] pa 45 t10 ;reset loc. 45
[80] pa 50 t320 ;reset loc. 50
[81] pa 51 t-1 ;reset loc. 51
[82] hv 41 ;go to start
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

cells 0-3 for help 3

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 suncheck 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 suncheck 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:

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

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 RO-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 alter-nate 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 RO-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 10-39 and In[RA, RB] is stored in the cell, and the content of the cell is read to RO-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:=Mx2^(-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:=Mx2
[ 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 ;controlcell 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]
[ 99] udn (p-1)        XVDLRC t-1
[100] qq   -512
[101] pa   r3 [104]    t511
[102] pa   r1 [103]
[103] gm                MRC t-1
[104] bt   511         t-1
[105] hv   r-2 [103]
[106] pa   r12 [118]  t511
[107] pa   r7 [114]
[108] grn  r1 [109]   XV
[109] qq
[110] sc   r-1 [109]  X
[111] gi   r9 [120]
[112] pi                ,am r-3 [109]
[113] smn  r-4 [109]  LC
[114] ar                IRC t-1
[115] hv   r6 [121]   NZ
[116] gi   r1 [117]  ,am r4 [120]
[117] nc                ,hh r6 [123]
[118] bt                t-1
[119] hv   r-7 [112]
[120] pi                ,hr s1
[121] zq                LKA
[122] pp   (r-8)[114] ,am p
[123] hv   r2 [125]  ,pp (r-9)[114]
[124] zq                LKA
[125] pa   r-52 [73]  t8
[126] pi   (r-6)[120] ,hv r3 [129]
[127] gm   p          MRC
[128] am   p          IQC
[129] hv   r-2 [127]  LKB
[130] pi   (r-10)[120] ,hv r-12 [118]
[131] gmn  -319       X LP t488

```

```

;test finished
;constant 0-39 := all 1
;procedure test;begin
;entry to test
;for j:=1023 step -1 until 512 do
;cell j 0-39:= M cell j 40-41 :=RA, RB
.....
;for i:=1023 step -1 until 512 do begin
;controlcell 0-39 := 0 / R:=testdigits
;controlcell:=-testdigits / M:=testdigits
;c120adr:=In
;In:=0 / R:=-testdigits
;if R=-1 then R:=(-testdigits)
;R:=R+cell[i] / In[RA, RB]:=cell i 40,41
;if R≠0 then goto errorincell
;c117adr := In / R:= old In
;if newmarks≠oldmarks then goto errorinmarks
;one cell tested
.....
;In:=Original marks / end of procedure test;
;errorincell: if errorincellAKA then stop
;p:=adress of wrong cell / R:=wrong cell
;goto seterror / errorinmarks: p:=adress of wrong
;error in marks AKA then stop
;seterror: error:=true
;In:=original marks
;testdigits to cell
;cell to R
;if KB then cycle in 127-128
;In:=Original marks / continue the test
;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 RO-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 alter-nate 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 RO-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 MO-39 and In[RA,RB] is stored in the cell, and the content of the cell is read to RO-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 ;MO-39:= all 1 / test
[-489] xrm ,cl -1 ;R:= MX2^(-1)
[-488] hh r-2 [-490] X NZ ;if M#0 then begin M:=R;goto c534 end
[-487] pm r34 [-453] ,pi 2 ;MO-39:= all 1 / In:=2
[-486] hs -451 ;test
[-485] pi ,hs -451 ;In:= 0 / test
[-484] xr ,tk 1 ;R:= MX2
[-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		;controlcell10-39:= 0 / R:=testdigits
[ -443]	qq				
[ -442]	sc	r-1 [-443]	X		;controlcell:= -testdigits / 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

INDICATOR TEST

B 12 - 1

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<sub>1</sub>.

41	zq		LKB	
42	grn	300		
43	pi	-1		
44	gi	300		
45	arn	300		
46	sr	-1	D	
47	zq		NZ	
48	hv	43	LKA	loop A, test of flip-flops for "ones"
49	pi			
50	gi	300	MOC	
51	arn	300		
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	60	LRC	} proving "L" indicator orders
59	zq			
60	zq		LOA	
61	zq		LOB	
62	zq		LTA	
63	zq		LTB	
64	zq		LPA	
65	zq		LPB	
66	zq		LQA	
67	zq		LQB	
68	gi	300	,arn	300
69	sr	3	D	
70	zq		NZ	

71	qq		MOC	R <sub>40-41</sub> 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	R <sub>40-41</sub> are set to one
77	ck	10	IRB	dummy order, RB: = 1
78	ck	10	IRA	dummy order, RA: = 1
79	zq		NRA	
80	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	} proving "L and N" indicator orders
87	zq			
88	zq		LOA	
89	zq		LOB	
90	zq		LTA	
91	zq		LTB	
92	zq		LPA	
93	zq		LPB	
94	zq		NRA	
95	zq		NRB	
96	gi		,arn	300
97	sr	15	D	
98	zq		NZ	
99	gmn	400	IQB	dummy order, QB: = 0
100	gmn	400	IQA	dummy order, QA: = 0
101	zq		LQB	
102	zq		LQA	
103	qq		M	
104	tk	2	IQB	dummy order, QB: = 1
105	tk	2	IQA	dummy order, QA: = 1
106	zq		NQB	
107	zq		NQA	
108	hv	82	LKA	loop D, test of QA, QB

109	pi	-16		t-17 inhibit changing all except PB
110	zq		NPB	
111	pi	-32		t-33 inhibit changing all except PA
112	zq		NPA	
113	hv	115	LPC	} proving "L" and "N" indicator orders
114	zq			
115	zq		LOA	
116	zq		LOB	
117	zq		LTA	
118	zq		LTB	
119	zq		NQA	
120	zq		NQB	
121	zq		NRA	
122	zq		NRB	
123	gi	300		
124	arn	300		
125	sr	63 D		
126	zq		NZ	
127	gmn	400	IPB	dummy order, PB: = 0
128	gmn	400	IPA	dummy order, PA: = 0
129	zq		LPB	
130	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	109	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
140	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	
150	zq			NQB	
151	zq			NRA	
152	zq			NRB	
153	gi	300			
154	arn	300			
155	sr	255	D		
156	zq			NZ	
157	qqn			ITB	TB: = 0
158	qqn			ITA	TA: = 0
159	zq			LTB	
160	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
170	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	
180	zq			NQA	
181	zq			NQB	
182	zq			NRA	
183	zq			NRB	

184	gi	300			
185	arn	300			
186	sr	-1	D		
187	zq			NZ	
188	arn		IOB		dummy order, OB: = 0
189	qqn		IOA		dummy order, OA: = 0
190	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
200	hv	202	LZC	}	proving "L" indicator orders
201	zq				
202	hv	204	LOC		
203	zq				
204	hv	206	LTC		
205	zq				
206	hv	208	LPC		
207	zq				
208	hv	210	LQC		
209	zq				
210	hv	212	LRC		
211	zq				
212	arn	211	IZB	}	proving Z combinations
213	ar	211	IZA		
214	zq		LZB		
215	zq		LZA		
216	pmm		IZC		
217	zq		NZB		
218	zq		NZA		
219	pin				
220	qq		IZB		
221	ar	212			
222	zq		NZB		
223	pin				
224	qq		IZA		
225	ar	213			
226	zq		NZA		

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			
[ 78]	pm	68		[ Subroutine, produce random test number]
[ 79]	arn	68	X	[ From the numbers in cells 66, 67, and 68]
[ 80]	mk	66		[ 3 numbers are produced to be placed in the]
[ 81]	sc	67		[ same cells. The contents of R is on exit]
[ 82]	dl	68	X	[ depending on the contents of these cells]
[ 83]	ac	66	IOA	[ and used as a test number. Possibly]
[ 84]	ml	67		[ overflow is indicated in respective OA]
[ 85]	ac	68	IOB	[ and OB for later check of flag bits.]
[ 86]	hr	s1		
[ 87]	0			[ Subroutine, input of 3 numbers]
[ 88]	10			[ Working cell]
[ 89]	pa	106	t62	[ Constant]
[ 90]	pa	107	t2	[ Reset start cell]
[ 91]	vy	17	,sy 64	[ Reset counting]
[ 92]	sy	34	,sy 35	[ Select typewriter as I/O, write CR]
[ 93]	sy	49	,sy 41	[ Write k, write l]
[ 94]	sy	64		[ write a, write r]
[ 95]	pt	102	t106	[ Write CR]
[ 96]	xrm		,lyn 87	[ Set jump address for positive number]
[ 97]	nc	32	,hv 100	[ Clear M, input of 1 character]
[ 98]	pt	102	t105	[ If character ≠ 32 then jump]
[ 99]	hh	96		[ else set jump address for negative number]
[ 100]	ca	16	,hvn 103	[ Jump to input of next character]
[ 101]	ca	64		[ If character=0 then clear R and jump]
[ 102]	xr		,hv	[ If character=64]
[ 103]	tk	-30	,ml 88	[ then exchange M and R and jump to storing]
[ 104]	hh	96		[ else produce a decimal number in M]
[ 105]	mt	-1	D	[ Jump to input of next character]
[ 106]	gr	62	t1	[ If negative number then change sign]
[ 107]	bt	2	t-1	[ Store the decimal number in working cell]
[ 108]	hv	95		[ Counter for input of 3 numbers]
[ 109]	hr	s1		[ Jump within 3 numbers]
				[ else exit]
[ 110]	pa	121	t3	[ Subroutine, output of bit pattern from R]
[ 111]	tl	-10		[ Prepare output in 4 groups]
[ 112]	pa	118	t9	[ Prepare output of 1 bit at a time]
[ 113]	mb	71		[ Prepare 10 positions in each group]
				[ Clear R pos. 0-9]

[ 114]	t1	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]
[ 128]	sy		,sy	[ Subroutine, output of track no. and word no.]
[ 129]	sy	34	,sy 49	[ Write SP, write SP]
[ 130]	sy	37	,sy 49	[ Write k, write a]
[ 131]	sy	35	,sy	[ write n, write a]
[ 132]	arn	p	D	[ write l, write SP]
[ 133]	ck		,t1 -30	[ Prepare output]
[ 134]	hs	203		[ of track no.]
[ 135]	sy		,sy	[ Jump to output of track no.]
[ 136]	sy	38	,sy 41	[ Write SP, write SP]
[ 137]	sy	52	,sy 37	[ Write o, write r]
[ 138]	sy	41	,sy 59	[ write d, write n]
[ 139]	srn	314	D	[ write r, write .]
[ 140]	ar	170	,t1 -30	[ Prepare output]
[ 141]	hs	203		[ of word no.]
[ 142]	hr	s1		[ Jump to output of word no.]
[ 143]	arn	72	,ac 76	[ Subroutine, error output]
[ 144]	ac	77		[ Add 1 to number of errors totally]
[ 145]	hv	180	LQB	[ Add 1 to number of errors per track]
[ 146]	arn	69	IPC	[ If QB=1 then jump back without output]
[ 147]	vy	16	,sy 64	[ else prepare output of test number]
[ 148]	sy	64		[ Select typewriter, write CR]
[ 149]	hs	110		[ Write CR]
[ 150]	sy	64		[ Jump to output of bit pattern]
[ 151]	arn	(170)	IPC	[ Write CR]
[ 152]	hs	110		[ Prepare output of read test number]
[ 153]	hs	128		[ Jump to output of bit pattern]
[ 154]	hv	180		[ Jump to output of track no. and word no.]
[ 155]	pa	158	t215	[ Fill drum track]
[ 156]	pa	159	t39	[ Reset start address]
[ 157]	hs	78		[ Reset counting]
[ 158]	gr	215	MOC t1	[ Jump to produce random test number]
[ 159]	bt	39	t-1	[ Store result incl. flag bits]
[ 160]	hv	157		[ Counter for 40 words]
[ 161]	vk	p	,sk 216	[ Jump back within a track]
[ 162]	hv	56	NKB	[ else select actual track no., write on track]
[ 163]	vk	p	,hh 48	[ Interrupt if KB=0]
				[ Exit]

[ 164]	pa	180		t39	[ Read track and check]
[ 165]	pa	170		t313	[ Reset counting]
[ 166]	vk	p	,lk	314	[ Reset start address]
[ 167]	hv	57		NKB	[ Select actual track no., read track]
[ 168]	vk	p	,hs	78	[ Interrupt if KB=0]
[ 169]	gr	69		MOC	[ Jump to produce random test number]
[ 170]	sr	313		t1	[ Store test number incl. flag bits]
[ 171]	hv	143		NZ	[ Subtract read number]
[ 172]	hv	175		LOB	[ Jump to error output if R pos. 0-39≠0]
[ 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]
[ 187]	arn	63	,gr	66	[ Transfers of working cells]
[ 188]	arn	64	,gr	67	[ Storing of the 3 inputed numbers]
[ 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			
[ 203]	pa	210		t4	[ Output of number of errors]
[ 204]	dk	215			[ Adjustment for 5 digits]
[ 205]	ar	214	X		[ Transform to machine number]
[ 206]	mln	213			[ Possibly round off, exchange M and R]
[ 207]	tk	30	,ga	209	[ Prepare ]
[ 208]	sy	16	V LZ		[ output of 1 digit]
[ 209]	sy				[ If digit=0 then write 0]
[ 210]	bt	4		t-1	[ else write digit]
[ 211]	hv	206			[ Counter for output of 5 digits]
[ 212]	hr	s1			[ Jump within 5 digits]
					[ else exit]
[ 213]	10				[ Constants]
[ 214]	1				
[ 215]	100000				
[ 216]	qq				[ 40 cells for the instruction sk 216]

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		[ Transfers of test numbers]
[ 265]	grf	312		[ Adder test]
[ 266]	arfn	308		
[ 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]	t1	-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 Program No. 14. Reader Test. 15/1-1965.  
-----

The program is to be used with an infinite 8-hole punched paper tape containing sets of characters with odd parity and values ranging from 0-127. Between each set of 128 characters there is about 1 inch of blank tape. Such tapes can be produced with Test Program A 206.

Input of the test tape should be started with blank tape and the program stops in case of error. When the START button is pressed, the program proceeds with the next character.

In case of error the character read is to be found in the R register and the character which it ought to have been in the p register. The p register can be altered manually during error stop. ]

```
i=41
zq   LKB           ; stop if KB
zq   NKB           ; stop if not KB
vy                   ; choose reader
pm 127 D           ; set mask for bits 3-9
pp 1023            ; set first character to -1
lyn r-4, pp p+1    ; read, add 1 to p
cm p   D           ; compare
zq                   ; error stop
hv r-3             ; jump
e41
-
```

Should it be impossible to input the test program by means of the reader, it can be done manually from the console by insertion of the following in cells 0-3:

```
[0] lyn 2, pp p+1; bits No. 8,19-22,24,25,31-35,38-40
[1] cm p D       ; bits No. 20,24,25,28,29,32
[2] zq           ; bit No. 25
[3] hv          ; bits No. 20-22
```

Also the following must be inserted in the registers:

```
by = 0
M = 12710-9; bits No. 3-9
p = 1023 ; bits No. 0-9
```

The test is started by a jump to cell No. 0 and the test is performed as described above.

[Test Program No. A 206.

Production of Tape for Reader Test No. 14. 15/1-1965.

-----

Punches characters with the values 0-127 and stops. When the START button is pushed, the program is repeated. ]

```
i = 10
vy 32                ; choose punch
pa r1 t 1023        ; set first character to -1
sy t 1              ; print character +1
bs (r-1) t 126     ; next order if 127
zq                 ; stop, jump to first character
hv r-3             ; jump to next character
e10
```

Beschreibung vom Programm Roots, das beim Übernahmertest von GIER  
angewendet wird.

Das Programm testet die Prozedur Roots, die sämtliche Komplexwurzeln in einem n'ten Grad Polynom  $P_n(z)$  mit Komplexkoeffizienten findet.

Das geschieht, indem man Nullstellen für  $F(x,y) = |P_n(x+iy)|^2$  mit Hilfe einer Gradientenmethode sucht.

Das Testprogramm, das ein ALGOL Programm ist, wird eingelesen und übersetzt, darauf werden die n (n=75) komplexen Zahlen, die nach dem Programm auf demselben Streifen stehen, eingelesen.

Das Programm generiert nun die Koeffizienten in diesem Polynom, welches diese Zahlen als Wurzel hat und wendet diese Koeffizienten als Parameter für Roots an. Die Lösung kommt als Ausgabe auf den Lochstreifenlocher.

Danach beginnt automatisch das Aufsuchen der Wurzeln im gleichen Polynom, und die Wurzeln von jedem Durchlauf können nun verglichen werden. Ein Durchlauf dauert etwa 40 Minuten.

Will man untersuchen, wie weit die Maschine in einem Durchlauf gekommen ist, drückt man K B und jedesmal wenn die Maschine eine Wurzel gefunden hat, wird sie - solange K B gedrückt ist - schreiben, wieviele Wurzeln noch zu finden sind.