

MATHEMATISCH CENTRUM

2e BOERHAAVESTRAAT 49

AMSTERDAM

REKENAFDELING

EEN SCHAKELSCHEMA VOOR KAARTPROGRAMMERING VAN DE
I.B.M. REKENENDE PONSMACHINE TYPE 602-A

door

M.L. Potters

R 224 II

1 9 5 5

EEN SCHAKELSCHEMA VOOR KAARTPROGRAMMERING VAN DE
I.B.M. REKENENDE PONSMACHINE TYPE 602-A

1. Inleiding.

In rapport R 224 I 1) is de functionele beschrijving gegeven van een systeem voor kaartprogrammering van de 602-A. De interne programmering van dit systeem, alsmede de belangrijkste details van het schakelschema zijn vastgelegd in dit tweede deel.

Het systeem, door ons gebruikt, is gebaseerd op een door Schrage 2) vereenvoudigde schakeling van Hankam 3). Wij hebben hierop de volgende uitbreidingen aangebracht:

- (a) Bij vermenigvuldiging of deling kan een getal naar believen als geheel getal of als breuk beschouwd worden.
- (b) Bij vermenigvuldiging van breuken en bij deling wordt het resultaat automatisch afgerond.
- (c) Elk resultaat kan één positie naar rechts of naar links opgeschoven worden zonder nieuwe bewerking.
- (d) Er is een opdracht toegevoegd, waardoor een iteratie van het proces van Newton voor de vierkantsworteltrekking gemaakt wordt.

Referenties:

- 1) J. Berghuis, Handleiding voor de kaartprogrammering op de I.B.M. rekenende ponsmachine type 602-A.
Rapport R 224 I, Rekenafdeling Mathematisch Centrum, 1954.
- 2) R.W. Schrage, The I.B.M. 602-A Calculating Punch as a General Purpose Calculator.
Techn. Newsletter no. 6, I.B.M. Applied Science Division, 1953.
- 3) E.V. Hankam, A Card Programmed 602-A Board.
9 pp mimeographed, New York, I.B.M. Nov. 25, 1951.

2. Telwerken en storages.

De programmeur heeft de beschikking over de Storages (2) t/m (8), waarin getallen van zeven cijfers geplaatst kunnen worden. Negatieve getallen komen als negen-complement hierin. In Storage (1R) wordt, door interne programmering, de deler of vermenigvuldiger in absolute waarde geplaatst.

De telwerken worden als volgt tot twee groepen, elk van zestien posities, gekoppeld:

$$\begin{aligned} [I] &= [1, 2, 3] \\ [II] &= [6, 7, 8] \end{aligned}$$

Deze telwerken staan uitsluitend ten dienste van de interne programmering. Negatieve getallen komen als negen-complement in de telwerken. Deeltal of vermenigvuldigtal worden in [I] opgenomen in absolute waarde. Het resultaat van de bewerking verschijnt in [II]. De in- en uitgangen van deze storages en telwerken worden verbonden door kanalen. Storage (1L) en telwerk [4] dienen als code-opbergruimte; telwerk [5] wordt niet gebruikt.

3. De code.

31 Algemene gegevens.

In dit systeem wordt een drie-adrescode toegepast, d.w.z. op elke programmakaart worden drie adressen A, B en C gegeven. Verder komt op de kaart nog een operatiecode voor en een code met betrekking tot de in- en uitvoer.

De machine haalt de getallen van de adressen A en B, voert er de verlangde operatie op uit en vervangt de laatste inhoud van C, die daardoor verloren gaat, door het resultaat van die operatie. Symbolisch geschreven: $A \text{ op. } B \rightarrow C$.

32 Adressen.

Als adres kan elk der storages (2) t/m (8) dienen. Storage 0, mer, adres en codepossing in de kaart komen overeen. Formeel kan ook storage (0) gebruikt worden, welks inhoud steeds = 0 is. De inhoud van de adressen A en B blijft intact bij elke operatie, tenzij een van beide identiek is met het adres C. Een getal in de kaart kan zonder verdere operatie op het A-adres geplaatst worden.

33 Operaties.

De operatiecode kan uit drie ponsingen bestaan, waardoor bepaald wordt:

- 1^o de te verrichten elementaire bewerking.
- 2^o de interpretatie van adres-inhouden bij vermenigvuldiging en deling als gehele getallen of als breuken.
- 3^o vermenigvuldiging met 10 of deling door 10 van het resultaat. De tweede en derde code komen in één kolom, zodat daarin dubbelponsing mogelijk is.

331 De eerste operatiecode indiceert de volgende operaties:

ponsing voor 1 ^e op. code	bewerking A op B → C
1	$A + B \rightarrow C$
2	$A - B \rightarrow C$
3	$A \times B \rightarrow C$
4	$\frac{B}{A} \rightarrow C$
6	$\frac{1}{2} \left(\frac{B}{A} + C \right) \rightarrow C$

Operatie 6 kan gebruikt worden voor iteratie van een vierkantswortel volgens Newton. Hiertoe moet $C = A$ gekozen worden.

Ook kan men met operatie 6 de inhoud van adres C halveren door $B=0$ te nemen.

Opm. De code 0 (of blanco) heeft dezelfde uitwerking als 4. Ter wille van de overzichtelijkheid gebruike men echter steeds de code 4 voor deling.

332 De tweede operatiecode heeft betrekking op de plaats van het decimaalteken. De inhoud van een adres kan n.l. op twee wijzen geïnterpreteerd worden in dit schema:

- 1^o als geheel getal.
- 2^o als breuk met de komma vooraan.

Voor optelling en aftrekking is de plaats van de komma irrelevant, maar bij vermenigvuldiging, deling en worteltrekking dient deze als volgt aangegeven te worden.

ponsing voor 2 ^e op. code	uitwerking
0	altijd bij optelling en aftrekking bij verm., deling en worteltrekking: inhoud van adres A, B en C als ge- heel getal beschouwd.
5	nooit bij optelling en aftrekking bij verm., deling en worteltrekking inhoud van adres A, B en C als breuk beschouwd (door automatische shift van 7 posities).

333 Door middel van de derde operatiecode kan het resultaat van een bewerking, vóór het op adres C geplaatst wordt, nog een vermenigvuldiging met of een deling door 10 ondergaan. De codering is als volgt

ponsing voor 3 ^e op. code	uitwerking
9	het resultaat wordt een positie naar rechts geschoven d.w.z. ver- menigvuldiging met 0,1. het cijfer in de laagste posi- tie gaat verloren.
0 (of blanco)	het resultaat wordt niet geschoven
X	het resultaat wordt een positie naar links geschoven d.w.z. ver- menigvuldiging met 10. het cijfer in de laagste positi- e wordt 0.

De afronding.

Alle afrondingen worden automatisch uitgevoerd en wel op de volgende wijze:

Bij opdracht 10, 20, 30, 60 wordt niet afgerond.

Bij opdracht 35, 40, 45, 65 wordt het cijfer in de laagste positie afgerond. Wanneer schuifopdrachten aanwezig zijn (9 of X) wordt eerst afgerond, daarna geschoven.

34 In- en uitvoer.

341 Een gegeven getal G wordt in absolute waarde in de kaart geponst. Negatieve getallen worden aangegeven door een X-ponsing in de hoogste positie.

Met deze gegeven getallen kan op twee wijzen gehandeld worden:

(1) Door de non-operatiecode wordt de inhoud van adres A vervangen door G. Symbolisch: $G \rightarrow A$.

De A-code mag in dit geval meervoudig zijn.

De B- en C-code en de operatiecode moeten achterwege blijven.

(11) Wordt geen non-operatiecode gegeven, doch op de gewone wijze geprogrammeerd, dan wordt G eerst bij het getal op adres A opgeteld en daarna de opdracht uitgevoerd.

Symbolisch: $(A + G) \text{ op } B \rightarrow C$.

Is speciaal $A=0$ dan wordt dus uitgevoerd

$G \text{ op } B \rightarrow C$

342 Een resultaat R kan door de machine geponst worden telkens wanneer een getal in storage (8) geplaatst wordt. Het resultaat wordt in dezelfde kaart geponst, waarin deze inleesopdracht gecodeerd is, en wel in absolute waarde met eventueel een X-ponsing in de hoogste positie als minteken.

Er kan dus in de volgende gevallen geponst worden:

(1) Als er een non-operatiecode is als $A=8$.

Symbolisch: $G \rightarrow A(=8) \rightarrow R$.

(11) Als er geen non-operatiecode is als $C=8$.

Symbolisch: $(A + G) \text{ op } B \rightarrow C(=8) \rightarrow R$.

Als er geen resultaat geponst moet worden, dient een skip-out code aangebracht te worden.

35 Kaartindeling.

De kaartindeling voor zover deze op de kaartprogramma's betrekking heeft, is als volgt vastgesteld:

kolom	ponsing
11-13	kaartno.
14	A-code
15	1 ^e operatiecode
16	2 ^e operatiecode
	3 ^e operatiecode
17	B-code
18	C-code
19	skip-out-code
20	non-operatie-code
21-27	gegeven getal G
28-30	gegeven no.
31-37	resultaat R
38-40	resultaat no.

4. Tijdsduur der opdrachten.

In onderstaande tabel is voor elke opdracht de benodigde machinetijd in seconden aangegeven. Hierbij is aangenomen, dat de cijfers 0 t/m 9 in elk van de zeven posities van de vermenigvuldiger of het quotient gemiddeld even vaak voorkomen.

Operatie	Tijdsduur	
	met ponsen	zonder ponsen
Non-op		
10	1,5	1,2
20		
30	5,7	
35		
40		
45		
60	6,6	
65		

Code 9 of X heeft geen invloed

5. Kanalen.

51 Algemeen.

Uit de volgende schakelschema's blijkt, hoe de in- en uitgangen van de verschillende storages en telwerken onderling verbonden zijn.

[illegible]

De exit (1R) is terwille van de duidelijkheid in viervoud getekend, hoewel de contacten in feite slechts dubbel zijn uitgevoerd.

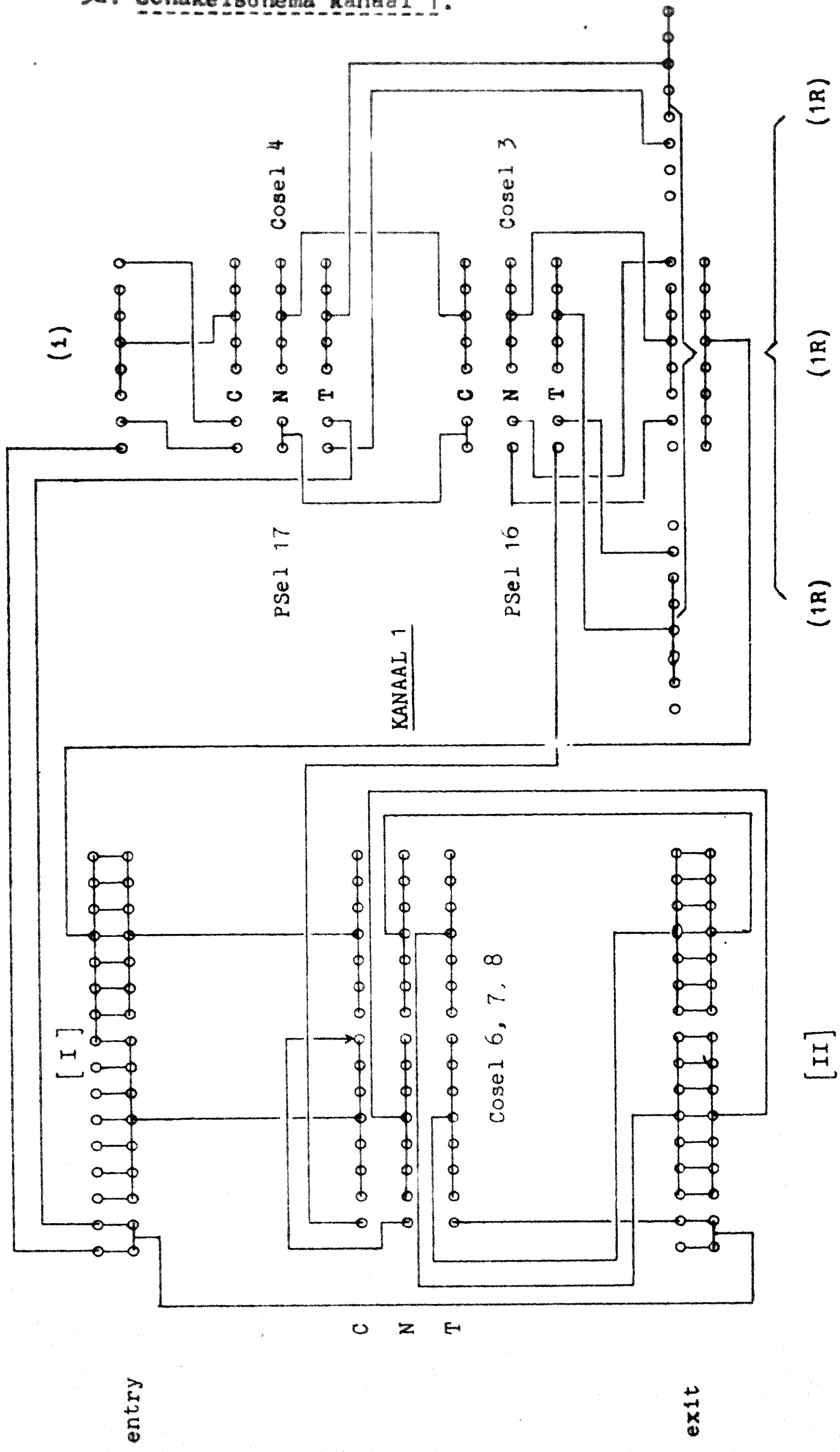
Cosel 6, 7 8 bewerkt waar nodig, de verschuiving over zeven posities bij operaties met breken.

Psel 16 en 17 resp. gekoppeld met Cosel 3 en 4 dienen voor een eventuele verschuiving van het resultaat over één positie naar rechts resp. links.

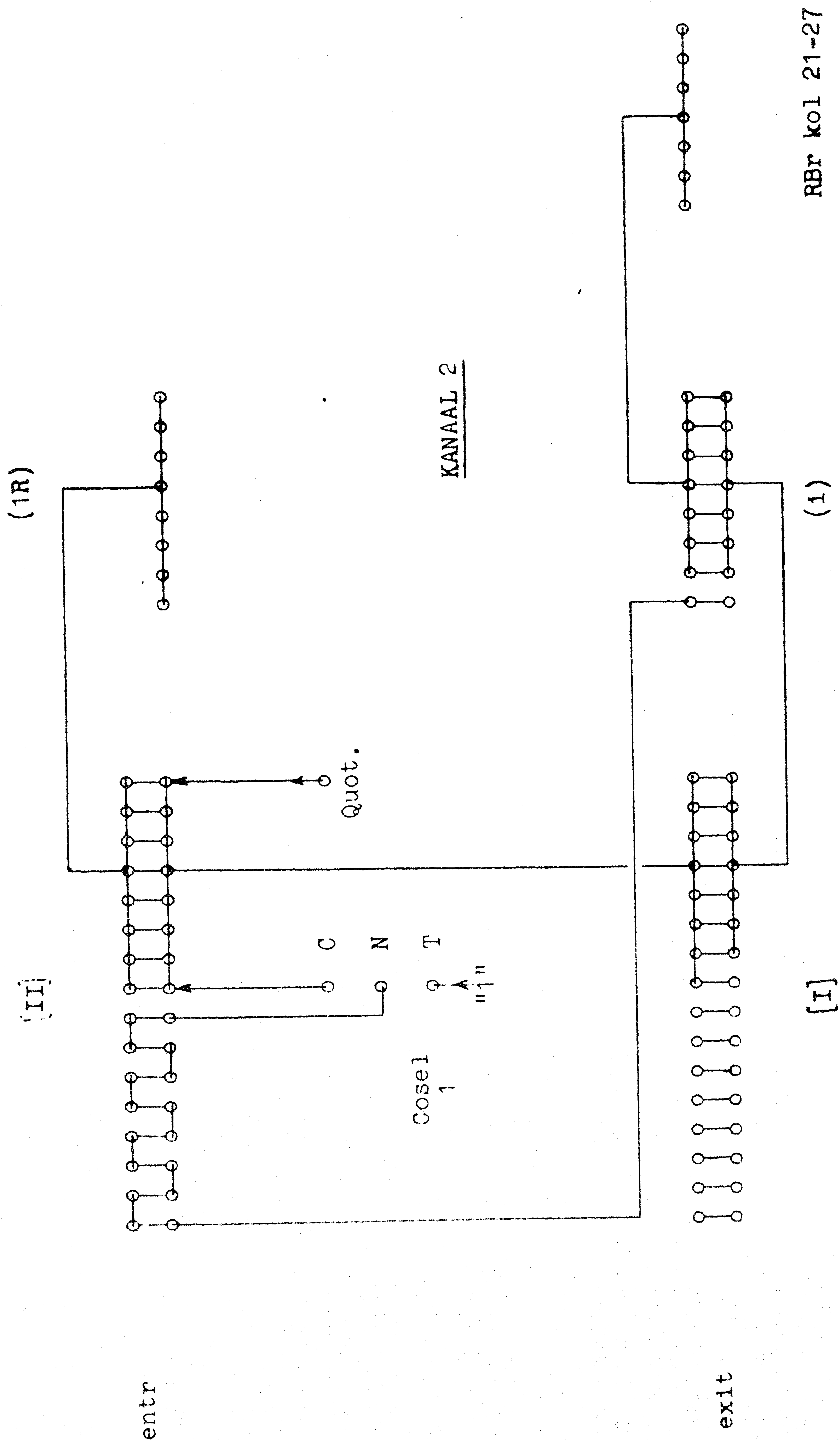
Kanaal 2 verbindt: exit [I], exit (1) 1 = 2 ... 7.
entry [II], entry (1R)
reading brushes, output quotient.

Cosel 1 dient voor toevoer van een "1" in verband met de afronding. Het ponskanaal verbindt de punch exit van storage (8) met de ponsmagneten, met conversie van negatieve getallen en een X-ponsing als minteken.

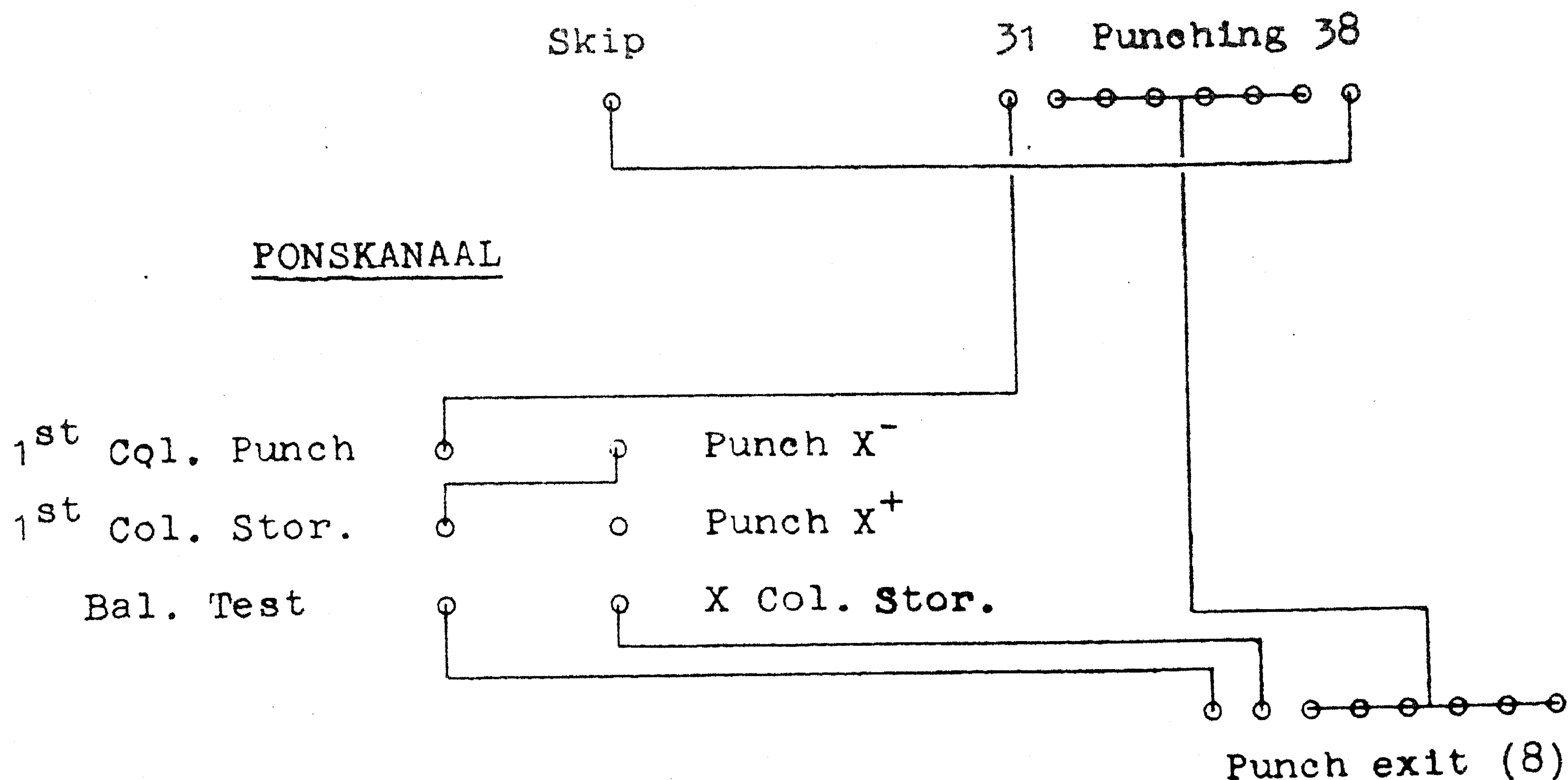
52. Schakelschema kanaal 1.



53 Schakelschema kanaal 2.



54 Schakelschema_ponskanaal.



6. Interne programmering.

61 Algemeen.

In de volgende secties vindt men het programma, dat de machine uitvoert bij elk van de mogelijke operaties en bij non-operatie. De derde operatiecode heeft geen invloed op het programma en is daarom buiten beschouwing gelaten.

In 62 is schematisch het verloop van de rekenkundige bewerkingen, welke tot het verlangde resultaat voeren, gegeven.

De stuurimpulsen, die deze bewerkingen realiseren, zijn in het schema van 63 beschreven. Waar de soort van de impuls niet is vermeld is de programimpuls bedoeld.

De keuze van het vereiste programma geschiedt door het gebruiken van selectors. De weg van de stuurimpulsen over de selectors, voor zover afhankelijk van de codering, is in 63 niet aangegeven. In Hoofdstuk 7 is de toepassing van selectors in detail beschreven.

Program	Operatie code									
	non-op.	10	20	30	35	40	45	60	65	
R	Gegeven $\rightarrow$ [II]									
1	skip	(A) $\rightarrow$ [II]								
2	skip		$\pm$ [II] $\rightarrow$ [I]				[II]	schoon		
3	skip		[I] $\rightarrow$ (1R)				[I]	schoon		
4	skip	(B) $\rightarrow$ [II]								
5	skip		$\pm$ [II] $\rightarrow$ [I]			[II]	schoon			
			inlezen in [I]: normaal			7 posities naar links	normaal	7 posities naar links		
6	skip		$\pm 5 \cdot 10^6$ [II]		$\frac{1}{2}(1R) \rightarrow$ [I]					
7	skip		$\pm$ [I] x (1R) $\rightarrow$ [II]			$\pm$ [I] / (1R) $\rightarrow$ [II]				
8	skip				(C) $\rightarrow$ [II], [I] schoon					
9	skip				$\frac{1}{2}$ [II] $\rightarrow$ [I]					
10	skip				- [I] $\rightarrow$ [II]					
11	[II] $\rightarrow$ (A)			[II] $\rightarrow$ (C)						
	uitlezen uit [II]: normaal ,			7 posities naar links			normaal			
	[I] schoon, [II] schoon									

63. Schema besturing.

Operatie code									
non-op.		10	20	30	35	40	45	60	65
Program	CBR	X21 → PUPSel 1					CplR → PU Cosel 9		
	RBr	14 (A-code) → PUPSel 2...8 via DSel 2				R1(1)			
	15,16 (Op-code) → PUPSel 11...17 via DSel 1					R1+ [4]			
	17 (B-code) → (1L)					R1+ [II] tken via Psel 1			
	18 (C-code) → [4]					RDO → Psel 1,9,12...17			
R	19 skipout code) → skip-out en → PUPSel 10					RDO → Psel 2...8			
	20 (non-op. code) → skip naar P 11					RDO → Psel 11			
	skip	Cpl P1 → PU Cosel 10		RO(A)	NB[II] → PU Psel 9	"9" → DOPSel 2...8			
		skip naar P4		R1+ [II]	RO(1L)	(1L) → PUPSel 2...8 via DSel 2			
		skip		RO [II]	RE [II]	R1+ [I] token via Psel 9			
2		skip		RO [I]	RE [I]	R1(1)			
	3	skip							
			Cpl P4 → PU Cosel 11		RO(B)	NB[II] → PUPSel 11	"9" → DOPSel 2...8		
4	skip	skip naar P11		R1+ [II]	RR[4]	"9" → DOPSel 11			
				teken via Psel 12		[4] → PUPSel 2...8 via DSel 2			
		skip		RO [II]	RE [II]	R1+ [I] token via Psel 11			
5						Cpl P5 →			Cpl P5 →
						PUCosel 16,78			PUCosel 16-8

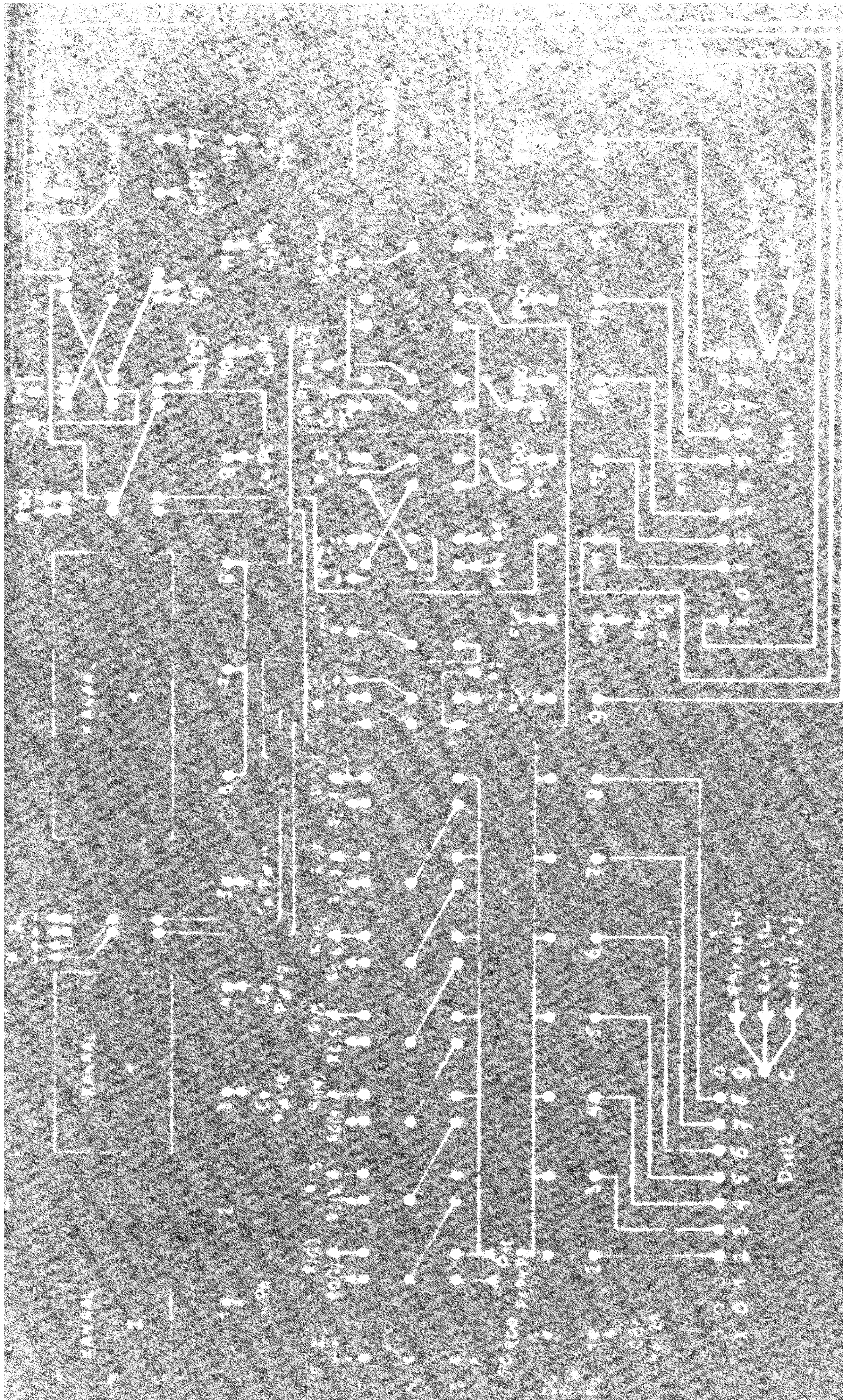
Program	Operatie-code										
	non-op.	10	20	30	35	40	45	60	65		
6	skip			Cpl P6 PUCosel 1 RO(1R) X5, tenths			R1+[I]				
				R1+ [II] teken via Psel 9 en Cosel 5							
7	skip			RO(1R) R1+[II]	teken via Psel 9 en Cosel 5 (Psel 11)						
				RO[I]	MPL	R1 - [I]		DI			
				skip naar P11							
8				skip			RO(C) R1+[II] RE[I]				
9				skip			RO[II] R1+[I] x0,5				
10				skip			RO[I] R1-[II]				
11	R1(A)	R1(C)									
	Voor A=8 resp. C=8 indien geen skip-out: punch (8) via Psel 10										
				Cpl P11 PUCosel 678		RO[II] RE[II]		RE[I] READ			

All Cycles → Punch Interlock

7. Selectors.

71 Overzicht gebruik der selectors.

No	Pilot Selectors			Coselectors Pick Up
	Pick Up	Drop out		
1	X21 van Control Brushes (teken van G)	X RDO		Cpl P6
2	2	D		—
3	3	D		Cpl Psel 16
4	4	D	RDO	Cpl Psel 17
5	5	D	"9" op P1	Cpl Psel 11
6	6	D	"9" op P4	Cpl P5 voor ⁽⁴⁵⁾ 65 op. Cpl P11 voor 35 op.
7	7	D		
8	8	D		
9	NB[II] op P1	X RDO		Cpl P0
10	skip-out code op P0	RDO		Cpl P1
11	NB[II] op P4	X ^D	RDO , "9" op P4	Cpl P4
12		D		Cpl Psel 13
13		D		
14	op code op P0	D	RDO	
15		D		
16		D		



INHOUD

	B12
1. INLEIDING	1
2. TELWERKEN EN STORAGES	2
3. DE CODE	2
31. Algemene gegevens	2
32. Adressen	2
33. Operaties	3
34. In- en uitvoer	4
35. Kaartindeling	5
4. TIJDSDUUR DER OPDRACHTEN	6
5. KANALEN	6
51. Algemeen	6
52. Schakelschema kanaal 1	7
53. Schakelschema kanaal 2	8
54. Schakelschema ponskanaal	9
6. INTERNE PROGRAMMERING	9
61. Algemeen	9
62. Schema rekenkundige bewerkingen	10
63. Schema besturing	11
7. SELECTORS	13
71. Overzicht gebruik der selectors	13
72. Schakelschema	14