N.V. ELECTROLOGICA
2e Boerhaavestraat 49
AMSTERDAM-O.

Telefoon 51660-56643

Korte algemene beschrijving van de X-1

Rapport EL-1-N

1957

N.V. ELECTROLOGICA
2e Boerhaavestraat 49
AMSTERDAM-O.

Telefoon 51660-56643

Korte algemene beschrijving van de X-1

Rapport EL-1-N

1957

Korte Algemene beschrijving van de X-1

I. De X-1 is een automatische rekenmachine welke zowel voor administratief als voor wetenschappelijk werk gebruikt kan worden. We onderscheiden in hoofdzaak drie onderdelen: de basismachine, het geheugen en de in- en uitvoerorganen.

a. De basismachine bevat het arithmetisch orgaan, de verschillende interne registers, de besturingsapparatuur etc. De physische gedaante is ongeveer die van een normaal schrijfbureau waarop zich tevens de verschillende bedieningsschakelaars en de indicatielampjes bevinden. De circuits welke in de electronische onderdelen worden gebruikt zijn alle volledig getransistoriseerd, zodat het totaal opgenomen vermogen voor een machine van deze omvang bijzonder laag is (enkele honderden Watts).

De X-1 werkt intern in het tweetallig stelsel, met vaste komma. Het optelorgaan is parallel, d.w.z. alle cijfers van de op te tellen getallen worden gelijktijdig aan de opteloperatie onderworpen.

Een "woord" van de machine bevat 27 bits en kan zowel een getal als een opdracht voorstellen. Als getal beschouwd is dit dus 26 bits + tekenbit. Als opdracht gezien kunnen we de 27 bits verdelen in 2 groepen; functionele cijfers en adrescijfers (de code van de machine is een één-adrescode).
Er zijn twee gelijkwaardige registers van 27 bits elk met accumulatorfunctie d.w.z. in deze registers kan worden opgeteld; alleen bij vermenigvuldiging, deling en handbediening hebben deze registers, resp. A- en S-register genaamd een verschillende functie. Nadere bijzonderheden zullen worden gegeven in verband met de bespreking van de opdrachtencode.
Behalve A- en S-register bevat de machine nog 3 registers, welke voor de programmeur een zekere betekenis hebben. In de eerste plaats het opdrachtregister (OR), evenals A en S met een capaciteit van 27 bits. Hierin blijft de opdracht gedurende de uitvoering ervan bewaard. Evenals dit bij A- en S-register het geval is, zal ook de inhoud van OR door middel van lampjes op het controlepaneel zichtbaar gemaakt kunnen worden, zodat men indien gewenst het programma ook stap voor stap

precies kan volgen.

De opdrachtteller (OT) is een register van 15 bits waarin steeds bewaard wordt het getal dat het adres aangeeft waar zich de volgende uit te voeren opdracht bevindt.

Tenslotte is er nog een register van 16 bits, B-register genaamd, waarvan de inhoud eventueel na extractie van een opdracht uit het geheugen doch voor de uitvoering ervan bij het adresgedeelte van deze opdracht kan worden opgeteld. De indicatie of deze optelling al dan niet dient plaats te vinden bevindt zich in de opdracht zelf. De capaciteit van 16 bits vloeit voort uit de omstandigheid dat dit adresincrement beide tekens kan hebben zodat er behalve de 15 adresbits nog een tekenbit aanwezig is.

b. Geheugen.
Het geheugen van de machine bestaat uit 2 gedeelten welke de naam "levend" en "dood" geheugen dragen. Beide zijn opgebouwd uit ferrietkernen en ze hebben dezelfde accestijd voor leesoperaties. Het verschil is dat terwijl in het levende geheugen door de machine ook geschreven kan worden dit bij het dode geheugen niet mogelijk is: hierin zijn de gegevens vastgelegd door een permanente bedrading. Het voordeel van het dode geheugen ligt in de in verhouding tot het levende geheugen bijzonder lage prijs: het is daardoor bij uitstek geschikt voor het bewaren van invoerprogramma's, subroutines van allerlei soort en eventueel voor standaard hoofdprogramma's.
De adreslengte der opdrachten maakt het mogelijk in principe 32768 woorden te accomoderen (levend en dood geheugen samen).
Het levend geheugen wordt afgeleverd in blokken van 512 woorden welke geassembleerd worden tot kasten van 4096 woorden: elke kast heeft zijn eigen volledig onafhankelijke selectieapparatuur zodat bij vergroting van het geheugen de betrouwbaarheidsmarges niet afnemen.

Het dode geheugen wordt geleverd in blokken van 64 woorden welke worden geassembleerd tot plugbare eenheden van 4096 woorden.

c. Voor in- en uitvoer bestaan eveneens verschillende mogelijkheden. In de eerste plaats kan invoer plaats vinden door middel van geponste telexband (150 symbolen/sec). Ponsband uitvoer (25 symbolen/sec) is eveneens mogelijk. Verder kan voor uitvoer gebruikt worden gemaakt van een schrijfmachine (10 symbolen/sec).

Tenslotte kan ten behoeve van ponskaarten in- en uitvoer een ponskaarten-reproductrice aan de X-1 worden gekoppeld. Deze machine verwerkt per gang ca. 7000 kaarten per uur. Van deze reproductrice kunnen zowel leesgang als ponsgang door de machine bediend worden. De machine kan het transportmechanisme van de reproductrice aan- en uitschakelen, terwijl bovendien de beide toevoermagazijnen elk afzonderlijk gestart en gestopt kunnen worden, zodat de kaarttoevoer in elke gang kan worden geregeld onafhankelijk van de andere gang. De communicatie tussen basismachine en ponskaartenmachine vindt plaats door tussenkomst van zogenaamde buffergeheugens. Deze maken het mogelijk dat de machine gedurende het kaartlezen normaal verder rekent.

Het kaartlezen wordt volkomen automatisch gecontroleerd, evenals het ponsen. Voor bijzonderheden verwijzen we naar hoofdstuk IV dat o.a. de ponskaart in- en uitvoer behandelt.

II. De opdrachtcode.

Zoals reeds werd gezegd kunnen de binaire cijfers waaruit een opdracht bestaat in twee groepen gesplitst worden: de functionele cijfers en de adrescijfers. Wanneer we de 27 opdrachtcijfers noemen OR0....OR26 (waarbij OR0 met het minst significante cijfer overeenkomt) dan beslaat het adres de cijfers OR0....OR14 en het functiegedeelte OR15....OR26.

Het tweede gedeelte kan men zich verder nog weer in twee stukken verdeeld denken: OR15....OR20 en OR21....OR26. Het laatste stuk specificeert het karakter van de opdracht (b.v. optelling, vermenigvuldiging enz.) terwijl het eerste aanwijzingen bevat omtrent enkele additionele mogelijkheden welke in principe bij elk type opdracht voorkomen: het betreffende stuk is in feite nog weer in drie secties van elk twee cijfers onderverdeeld. Daar er ter specificatie van het opdrachtkarakter 6 binaire cijfers beschikbaar zijn bevat de lijst der opdrachten dus 64 verschillende instructies, welke we nu achtereenvolgens in het kort zullen bespreken. De tabel geeft een volledig overzicht in symboliche notatie, gemakshalve genummerd van 0-63. (X) betekent "de inhoud van X". De cijfers in de 3 laatste kolommen hebben betrekking op de verschillende modificatiemogelijkheden welke in hoofdstuk III worden besproken.

		OR20 OR19	OR18 OR17	OR16 OR15
0	+ (n) + (A) → A	1	4	7
1	- (n) + (A) → A	1	4	7
2	+ (n) → A	1	4	7
3	- (n) → A	1	4	7
4	+ (A) + (n) → n	2	4	8
5	- (A) + (n) → n	2	4	8
6	+ (A) → n	2	4	8
7	- (A) → n	2	4	8
8	+ (n) + (S) → S	1	4	7
9	- (n) + (S) → S	1	4	7
10	+ (n) → S	1	4	7
11	- (n) → S	1	4	7
12	+ (S) + (n) → n	2	4	8
13	- (S) + (n) → n	2	4	8
14	+ (S) → n	2	4	8
15	- (S) → n	2	4	8
16	+ (n).(S) + (A) → AS	1	5	8
17	- (n).(S) + (A) → AS	1	5	8
18	+ (n).(S) → AS	1	5	8
19	- (n).(S) → AS	1	5	8
20	+ (n) ∔ (A) → A	1	4	7
21	- (n) ∔ (A) → A	1	4	7
22	+ (n) ∧ (A) → A	1	4	7
23	- (n) ∧ (A) → A	1	4	7
24	(AS): +(n) → A,S	1	5	8
25	(AS): -(n) → A,S	1	5	8
26	2^{26}(A): +(n) → A,S	1	5	8
27	2^{26}(A): -(n) → A,S	1	5	8
28	+ (n) ∔ (S) → S	1	4	7
29	- (n) ∔ (S) → S	1	4	7
30	+ (n) ∧ (S) → S	1	4	7
31	- (n) ∧ (S) → S	1	4	7

		OR20 OR19	OR18 OR17	OR16 OR15
32	+(n) + (B) → B	1	4	7
33	-(n) + (B) → B	1	4	7
34	+(n) → B	1	4	7
35	-(n) → B	1	4	7
36	+(B) + (n) → n	2	4	8
37	-(B) + (n) → n	2	4	8
38	+(B) → n	2	4	8
39	-(B) → n	2	4	8
40	+(n) + (OT) → OT	1	4	9
41	-(n) + (OT) → OT	1	4	9
42	+(n) → OT	1	4	9
43				
44	-1 + (τ_m) → τ_m n → OT	3	6	9
45	-1 + (τ_{m+4}) → τ_{m+4} n → OT	3	6	9
46	+ (OT) +1 → λ_m n → OT	3		9
47	+ (OT) +1 → λ_{m+4} n → OT	3		9

0,1 Optelling in

Het getal op geheugenplaats n wordt positief of negatief opgeteld bij het getal dat zich in het A-register bevindt. Het resultaat wordt in het A-register opgeborgen. De oorspronkelijke inhoud van A gaat dus verloren. De inhoud van geheugenplaats n blijft ongewijzigd.

0,3 Transport in

Het getal op geheugenplaats n vervangt het getal in register A (al of niet met tekenwijziging). De oorspronkelijke inhoud van A gaat verloren, de inhoud van geheugenplaats n blijft ongewijzigd.

4,5 Optelling uit

Het getal in register A wordt positief of negatief opgeteld bij het getal op geheugenplaats n, het resultaat wordt opgeborgen op geheugenplaats n. De oorspronkelijke inhoud van geheugenplaats n gaat verloren, de inhoud van A blijft onaangetast.

6,7 Transport uit

De inhoud van register A vervangt, (al of niet met tekenwisse-

ling) de inhoud van geheugenplaats n. De oorspronkelijke inhoud van geheugenplaats n gaat verloren, de inhoud van A blijft ongewijzigd.

8....15
Aequivalent aan 0....7 waarbij het A-register door het S-register vervangen is.

16,17 Vermenigvuldiging
De inhoud van geheugenplaats n wordt (met positief of negatief teken) vermenigvuldigd met de inhoud van het S-register. Het dubbele-lengte product, aan de minst significante zijde vermeerderd met de inhoud van het A-register komt in de registers A en S; waarbij A het meest significante en S het minst significante stuk bevat. De tekens van A en S zijn beide gelijk aan het teken van het resultaat. De oorspronkelijke inhoud van A en S gaat verloren; de inhoud van geheugenplaats n blijft ongewijzigd.

18,19 Vermenigvuldiging Als 16,17 doch zonder vermeerdering van het product met de inhoud van A. De oorspronkelijke inhoud van A gaat ook hier verloren.

20, Logische optelling
Uit de inhoud van geheugenplaats n en die van A wordt gevormd een getal dat een 1 heeft in alle cijferposities waarin (n) en (A) verschillende cijfers hebben (dus (n) een 0 en (A) een 1 of omgekeerd). In alle andere posities heeft het resultaat een nul. Dit resultaat vervangt tenslotte de inhoud van A, waarbij dus de oorspronkelijke inhoud van A weer verloren gaat doch de inhoud van geheugenplaats n niet verandert.

21.
Als 20, doch met de getallen (A) en - (n).

22 Logische vermenigvuldiging.
Uit de inhoud van geheugenplaats n en die van A wordt gevormd een getal dat een 1 heeft in alle cijferposities waarin beide oorspronkelijke getallen een 1 hebben. In alle andere cijferposities heeft het resultaat een 0. Dit resultaat vervangt de oorspronkelijke inhoud van A, welke verloren gaat. De inhoud

van geheugenpositie n blijft onaangetast.

23
Als 22, doch met de getallen (A) en - (n).

24,25 Deling.
Het dubbele-lengte getal in A en S (A meest significante gedeelte, S minst significante gedeelte; teken A en teken S moeten gelijk zijn) wordt gedeeld door de inhoud van geheugenplaats n. (met positief of negatief teken). Het quotient komt in S en de rest blijft in A achter.(De rest is van de resten met het teken van het deeltal die met de kleinste absolute waarde), De oorspronkelijke inhoud van A en S gaat verloren, de inhoud van geheugenplaats n verandert niet.

26,27 Deling.
Als 24,25 doch voor het begin van de deel-operatie wordt de inhoud van S vervangen door nul. Bij deze opdrachten geldt de eis dat teken A en teken S gelijk moeten zijn niet.

28....31
Aequivalent aan 20-23, waarbij het A-register door het S-register vervangen is.

32....39
Aequivalent aan 0-7 waarbij het A-register door het B-register vervangen is. Er zij nog aan herinnerd dat het B-register slechts 16 cijfers heeft.

40,41 Additieve sprongopdrachten.
De inhoud van geheugenplaats n wordt opgeteld bij, of afgetrokken van de inhoud van de opdrachtteller en het resultaat gaat terug naar de opdrachtteller: de volgende instructie wordt nu dus gehaald van het adres dat de OT nu bevat.

42 Gewone sprongopdracht.
De inhoud van de OT wordt vervangen door de inhoud van geheugenplaats n; deze zelf blijft ongewijzigd.

44,45 Tellende sprongopdracht. De inhoud van de OT wordt vervangen door het adresgedeelte van deze opdracht (dus door het getal n, niet door de inhoud van geheugenplaats n) Bovendien wordt de inhoud van de (vaste) geheugenplaats τ_m(m = 0....3)

resp. τ_{m+4} als geheel getal geinterpreteerd, met 1 verminderd. De ware betekenis van deze opdracht komt pas tot uiting wanneer ook gebruik gemaakt wordt van de faciliteiten welke in de resterende functiecijfers omschreven worden.

46,47 Subroutine sprongopdrachten.

De inhoud van de OT wordt vervangen door het adresgedeelte van deze opdracht. Bovendien wordt de oorspronkelijke inhoud van OT, nadat de normale ophoging met 1 der OT heeft plaats gevonden, overgebracht naar de vaste geheugenplaats λ_m resp. λ_{m+4} (m = 0....3). De oorspronkelijke inhoud van λ_m gaat hierbij verloren.

De opdrachten met nummer boven 47 worden "communicatieopdrachten" genoemd. Zij dragen een bijzonder karakter en worden later besproken. Hierbij heeft het adresgedeelte n.l. geen betekenis als adres, doch wordt gebruikt voor een nadere omschrijving van de functie der opdrachten.

III. We geven nu nog een uiteenzetting van de betekenis van de
a functiecijfers OR15....OR20. Zoals reeds werd gezegd vallen deze weer in drie groepen van twee uiteen. De verschillende mogelijkheden per groep nummeren we in overeenstemming met de numerieke waarde der bijbehorende cijfers.

1.2 OR19 en OR20

Voor opdrachten 0-42 en 48-63

0 Geen modificatie
1 "Absoluut" (dit geldt niet voor opdr. 4-7, 12-15, 36-39)
2 B-gecorrigeerd, ongemodificeerd teruggeschreven
3 B-gecorrigeerd, gemodificeerd teruggeschreven.

Bij geval 1 betekent "Absoluut" dat in de betreffende opdracht het adresgedeelte niet als adres doch als getal moet worden opgevat: dit getal denke men zich daartoe tot een normaal getal van 27 cijfers uitgebreid door toevoeging van nullen aan de meest significante kant. Men heeft dus steeds wanneer in de opdrachtspecificatie staat "de inhoud van geheugenplaats n" te lezen "het getal n". Bij de gevallen 2 en 3 is sprake van de reeds in het voorgaande vermelde B-correctie. Hierbij wordt dus voor het uitvoeren van de betreffende opdracht de inhoud van het B-register bij het adresgedeelte ervan opgeteld en wordt vervolgens dus het getal gebruikt dat zich bevindt op het aldus verkregen nieuwe adres. Het onderscheid tussen de gevallen 2 en 3 bestaat dan verder hierin dat bij 2 de opdracht zoals die oorspronkelijk in het geheugen stond ongewijzigd blijft terwijl in geval 3 ook in het geheugen het nieuw gevormde adres wordt overgenomen.

Voor opdrachten 44-47 is de interpretatie van OR19 en OR20 anders. Bij deze opdrachten hebben de bovengenoemde functies geen reele betekenis en worden deze cijfers daarom gebruikt om de adressen der in deze opdrachten optredende vaste registers aan te geven:

3. 0 m = 0
1 m = 1
2 m = 2
3 m = 3

Tenslotte kan in dit verband nog worden opgemerkt, dat bij sommige communicatieopdrachten het gebruik van deze cijfers tot bijzonder vreemde resultaten kan leiden aangezien hier het adres in feite functiebetekenis heeft, zodat door B-correctie een opdracht kan worden veranderd in één met totaal ander karakter.

b. OR17 en OR18

Deze twee cijfers hebben over het algemeen betrekking op de z.g. "conditie". Dit is een afzonderlijk intern geheugenelement waarin een binair cijfer kan worden bewaard. Het kan door opdrachten van bepaalde gedaante in de nul- of één-stand gezet worden op grond van bepaalde criteria en kan door andere opdrachten worden "gehoorzaamd". Dit laatste houdt dan gewoonlijk in dat de betreffende opdracht geheel of ten dele onuitgevoerd blijft wanneer de conditie in een bepaalde stand verkeert. De cijfers OR17 en OR18 beinvloeden speciaal het zetten van de conditie. Er doen zich hierbij 4 verschillende gevallen voor.

Voor opdrachten 0-15, 20-23, 28-42, 48-63

4. 0 Geen modificatie
   1 Zet conditie overeenkomstig het teken van het resultaat (S → C)
   2 Zet de conditie op nul als het resultaat nul is en op een als het resultaat niet gelijk aan nul is.
   3 Zet de conditie op nul als het teken van het resultaat gelijk is aan het teken van het resultaat van de vorige conditie zettende opdracht.

Bij de hier genoemde opdrachten is steeds voldoende duidelijk wat met "resultaat" bedoeld wordt: dit is het getal dat als antwoord van de arithmetische bewerking wordt verkregen. Bij sommige communicatieopdrachten is van een antwoord in eigenlijke zin geen sprake (b.v. startopdrachten reproductrice) en hier heeft het gebruik van deze cijfers dus geen zin. Bij vermenigvuldiging en deling wordt een resultaat van dubbele lengte verkregen zodat hier een nadere omschrijving gewenst is.

5. Voor opdrachten 16-19, 24-27
   0 Geen modificatie
   1 Zet conditie overeenkomstig het teken van het A-register.
   2 Als 4-2 met gebruikmaking van het resultaat in A
   3 Als 4-3 met gebruikmaking voor de vergelijking van het teken in A.

   Dit betekent dus dat bij de deling steeds wordt uitgegaan van de rest.

6. Voor opdrachten 44 en 45
   Daar deze opdrachten geen arithmetisch resultaat hebben zijn de boven gegeven omschrijvingen hier niet van toepassing. De cijfers OR17 en OR18 worden nu gebruikt om aan te geven of, en zo ja, onder welke nadere voorwaarde de betreffende sprong uitgevoerd dient te worden.
   0 Geen modificatie
   1 Voer de sprong uit als het nieuwe teken van τ_m positief is.
   2 Voer de sprong uit als de nieuwe inhoud van τ_m gelijk is aan nul.
   3 Voer de sprong uit als de nieuwe inhoud van $\tau_m \geq 0$ is.

   Het zal nu duidelijk zijn dat met behulp van een tellende sprong-opdracht op eenvoudige wijze een bepaalde cyclus in het programma een gegeven aantal malen doorlopen kan worden door van de cijfers OR17 en OR18 gebruik te maken. In geval 1 wordt de cyclus precies zo vaak doorlopen als door de oorspronkelijke "telling" in τ_m wordt aangegeven, in geval 3 één maal vaker, terwijl geval 2 in zekere zin als het inverse van geval 1 beschouwd kan worden.
   Bij opdrachten 46 en 47 kan men ook niet van een arithmetisch resultaat spreken. De cijfers OR17 en OR18 worden hier genegeerd.

c. OR15 en OR16
   Deze cijfers worden in hoofdzaak gebruikt om aan te geven dat de conditie gehoorzaamd moet worden en op welke wijze dit heeft te geschieden.
   Voor opdrachten 0-3, 8-11, 20-23, 28-35, 48-51, 56-59 (d.w.z. voor alle in-opdrachten behalve vermenigvuldiging en deling)

7. 0 Geen modificatie
   1 Geen registerwijziging
   2 Voer de opdracht uit als de conditie = 0, sla anders deze opdracht over (Skip).
   3 Voer de opdracht uit als de conditie = 1, sla anders deze opdracht over (Skip).

"Geen registerwijziging" in geval 1 betekent dat het arithmetisch resultaat van de in de opdracht gespecificeerde bewerking wel in de machine wordt gevormd, doch dat dit niet vervolgens in één der registers wordt opgeborgen. Het resultaat van de bewerking gaat dus volledig verloren terwijl de oorspronkelijke gegevens alle bewaard blijven. Het is duidelijk dat deze modificatie dus op zichzelf zinloos is: ze krijgt betekenis in combinatie met de cijfers OR17 en OR18 waardoor het wel mogelijk is op grond van het resultaat de conditie te beinvloeden: men kan dus b.v. de conditie zetten op grond van de vraag of het getal in A groter is dat het getal op een bepaalde geheugenplaats zonder daarbij A of die geheugenplaats te verstoren.
De gevallen 2 en 3 behoeven weinig toelichting: ze stellen ons in staat van elke opdracht een conditionele opdracht te maken.

Bij uit-opdrachten, vermenigvuldiging en deling vervalt de bovengenoemde mogelijkheid 1. Dus voor opdrachten 4-7, 12-19, 24-27, 36-39, 52-55, 60-63.

8. 0 Geen modificatie
   1 Wordt genegeerd
   2, 3 } als onder 7

Bij opdrachten 40 t/m 47 geldt de volgende interpretatie

9. 0 Geen modificatie
   1 Voer de opdracht uit als overloopindicatie = 1 en herzet overloopindicatie, sla anders deze opdracht over. (Skip)
   2, 3 } als onder 7

De overloopindicatie is een apart geheugenelement dat 1 bit kan bevatten (vergelijkbaar met de conditie) In dit element wordt een 1 gelezen wanneer door optelling van 2 positieve

getallen in de machine (tengevolge van capaciteitsoverschrijding) een negatief antwoord ontstaat of wanneer door optelling van 2 negatieve getallen een positief antwoord ontstaat. In sommige gevallen kan dit geen kwaad zodat het geen zin heeft de machine bij iedere capaciteitsoverschrijding te laten stoppen. Soms zal men echter in dit geval speciale maatregelen willen nemen waartoe het hiergenoemde geval 1 de mogelijkheid schept.
Een enigszins merkwaardige situatie doet zich voor met betrekking tot de tellende sprongen. Deze worden n.l. wanneer we zowel van OR15 en OR16 als ook van OR17 en OR18 gebruik maken dubbel conditioneel: hierbij heeft de conditionaliteit op OR15 en OR16 preferentie in die zin dat wanneer volgens OR15 en OR16 een skip moet plaats vinden de opdracht genegeerd wordt. Wanneer op grond van OR15 en OR16 de opdracht wel voor uitvoering in aanmerking komt wordt indien volgens OR17 en OR18 de sprong niet moet worden gedaan alleen de telling uitgevoerd.

N.B.: Onder in-opdrachten worden verstaan opdrachten waarvan het arithmetisch resultaat of wat daarmee overeenstemt in een der interne registers wordt geborgen. Bij uit-opdrachten gaat het resultaat naar het geheugen of een der uitvoerorganen.
Tenslotte laten we nog een tabellarisch overzicht van de verschillende mogelijkheden volgen.

	OR20 OR19	
	0	Normaal
1	1	Absoluut
	2	met B-correctie, opdracht wordt niet gemodificeerd
	3	met B-correctie, opdracht wordt gemodificeerd

	OR20 OR19	
	0	Normaal
2	1	Niet van toepassing
	2	met B-correctie, opdracht wordt niet gemodificeerd
	3	met B-correctie, opdracht wordt gemodificeerd

OR20 OR19

	0	m = 0
3	1	m = 1
	2	m = 2
	3	m = 3

OR18 OR17

	0	Normaal
4	1	zet conditie op teken van het resultaat (T→C)
	2	zet conditie op het nul zijn van het resultaat (=0?)
	3	zet conditie op tekenwisseling (Is teken arithm. resultaat gelijk aan dat van de laatste conditiezettende opdracht ?)

OR18 OR17

	0	Normaal
5	1	zet conditie op het teken van de nieuwe inhoud van A.
	2	zet conditie op het nul zijn van de nieuwe inhoud van A.
	3	zet conditie op tekenwisseling van A.

OR18 OR17

	0	Normaal
6	1	Spring indien nieuwe teken $\tau_m = 0$.
	2	Spring indien nieuwe $(\tau_m) = 0$
	3	Spring indien nieuwe $(\tau_m) \geqslant - 0$

OR16 OR15

	0	Normaal
7	1	Geen registerwijziging
	2	Voer opdracht uit als C = 0, anders skip.
	3	Voer opdracht uit als C = 1, anders skip.

OR16 OR15

	0	Normaal
8	1	Niet van toepassing
	2	Voer opdracht uit als C = 0, anders skip
	3	Voer opdracht uit als C = 1, anders skip

			OR16 OR15
	0	Normaal	
9	1	Voer opdracht uit als OF = 1, anders skip	
	2	Voer opdracht uit als C = 0, anders skip	
	3	Voer opdracht uit als C = 1, anders skip	

III. Schuif- en normeeropdrachten

De preciese codering van de schuifopdrachten volgt uit onderstaand schema.

Schuifopdrachten

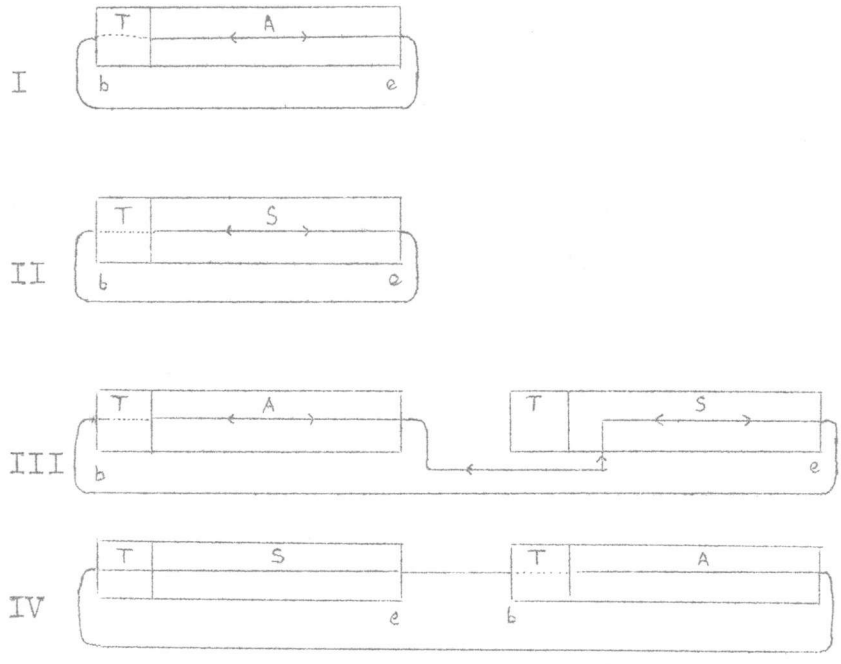
functie / adres		links 54	rechts 55	links 62	rechts 63
0	n	I rond	I rond	II rond	II rond
1	n	I schoon	I schoon	II schoon	II schoon
2	n	III rond	III rond	IV rond	IV rond
3	n	III schoon	III schoon	IV schoon	IV schoon

n = 0....31 tijdsduur: 40 + 8n μ s

De bovenste rij vermeldt de functiecijfers, de eerste twee kolommen het adresgedeelte van de opdracht (van een adres in de eigenlijke zin van het woord is natuurlijk geen sprake) en wel in die zin dat het getal in de eerste kolom, vermenigvuldigd met 32 bij het getal uit de tweede kolom moet worden opgeteld om de numerieke waarde van het adresgedeelte te verkrijgen. In overeenstemming hiermee mag het getal in de tweede kolom hoogstens gelijk zijn aan 31. Aangezien bij alle communicatie-opdrachten het getal in de eerste kolom hoogstens gelijk is aan64 zijn de digits OR12, OR13 en OR14 bij deze opdrachten gelijk aan nul.

We zullen nu het schema van de schuifopdrachten iets meer gedetailleerd bespreken.

Om te weten welke bewerking een bepaalde schuifopdracht uitvoert, beginnen we te kijken naar de in de hokjes aangegeven romeinse cijfers (I t/m IV). Dit cijfer geeft het z.g. "circuit" aan. Er zijn 4 mogelijkheden



Elk dezer circuits heeft een begin en een einde, in de figuur aangegeven resp. met b en e. Het teken dat gevonden wordt aan het begin van het circuit zal in het volgende worden aangeduid als het teken van het circuit. In al deze circuits kan links of rechts geschoven worden en bovendien nog z.g. "rond" of "schoon". Het aantal plaatsen waarover geschoven wordt, wordt aangegeven door n.

Rondschuifopdrachten

Deze behoeven ternauwernood nadere explicatie. De gehele inhoud van het circuit wordt cyclisch rondgeschoven over n plaatsen hetzij linksom, hetzij rechtsom.

"Schone" schuifopdrachten

Hierbij wordt het circuit a.h.w. opengeknipt en wel in het geval van rechts schuiven vlak voor het teken van het circuit, in het geval van links schuiven vlak daarachter. In beide gevallen heeft dit tot gevolg dat het teken van het circuit ongewijzigd blijft terwijl het circuit aan de meest resp. minst significante zijde wordt aangevuld met het teken van het circuit.

Opmerking: Bij deze opdrachten verliest men dus informatie.
Tot slot nog enige voorbeelden ter illustratie van het voorgaande.
54/0/1 Romeins cijfer I, heeft dus betrekking op het A-register afzonderlijk en wel 1 plaats naar links rond. Het oorspronkelijke tekencijfer van A komt dus nu te staan op de minst significante plaats, terwijl de meest significante bit verschuift naar het tekencijfer. Arithmetisch gezien is dit verdubbeling modulo $2^{27}-1$.
54/1/1 Als boven, doch nu schoon. Teken van A blijft dus ongewijzigd. Arithmetisch is dit verdubbeling modulo 2^{26}.
63/1/2 Heeft betrekking op het S-register en 2 plaatsen naar rechts, schoon. Aan de meest significante zijde wordt tweemaal aangevuld met teken S. Dit komt neer op bepaling van het aantal gehelen van $\frac{(S)}{4}$.
54/2/3 Circuit no. III, 3 plaatsen naar links schoon. S wordt aan de minst significante kant aangevuld met 3 maal teken A. Teken A blijft ongewijzigd i.v.m. de keuze van de "schoon" versie. Teken S blijft bij dit circuit altijd ongewijzigd.
Bij schuifopdrachten wordt in de gevallen III en IV de conditie gezet op grond van de inhoud van het register aan het begin van het circuit.
De niet-register wijzigende uitvoering van de opdracht bestaat bij schuifopdrachten niet.
We bespreken nu de normeeropdrachten, zie onderstaand schema.

Normeeropdrachten

Adres		functie	
		54	62
5	0	Normeer (A) n ≠ B	Normeer (S) n ≠ B
7	0	Normeer (AS) n ≠ B	

n = aantal geschoven plaatsen
tijdsduur $40\,\mu s + 8.n\,\mu s$

54/5/0
62/5/0 A, resp. S wordt genormeerd, d.w.z. de inhoud van het register wordt zo vaak naar links geschoven tot de meest significante bit van het teken van het register verschilt. Het aantal plaatsen wat te dien einde geschoven moest worden, wordt genoteerd in B.

54/7/0 Idem, doch nu wordt het lange getal AS genormeerd (circuit III) Wanneer de inhoud van de te normeren registers gelijk aan nul is wordt het normeren in "t geval van een enkele-register-normering na 26 schuifslagen beeindigd, in 't geval van een dubbele-register-normering na 52 schuifslagen.

IV. Overige communicatieopdrachten

De nog resterende communicatie-opdrachten worden in het volgende schema samengevat.

* = het andere register

** = alleen 62

Communicatie-opdrachten.

functie R = A / S	48 / 56	49 / 57	50 / 58	51 / 59	54 / 62	55 / 63
Adres						
OR20 OR19 | OR18 OR17 | OR16 OR15						
2 | 4 | 7						
8 | 0					+(R) ≠ R*	-(R) ≠ R*
16 | 1	2 4 7 | (R) + (BL) ≠ R	2 4 7 | (R) - (BL) ≠ R	2 4 7 | + (BL) ≠ R	2 4 7 | - (BL) ≠ R	2 | 8 | +(R) ≠ TP pons (TP)	2 | 8 | -(R) ≠ TP pons (TP)
16 | 2	2 4 7 | (R) + (TR) ≠ R	2 4 7 | (R) - (TR) ≠ R	2 4 7 | + (TR) ≠ R	2 4 7 | - (TR) ≠ R	2 | 8 | +(R) ≠ TP type (TP)	2 | 8 | -(R) ≠ TP type (TP)
2 | 4 | 8						
32 | 0					(S) x 10 ≠ (AS) **	
32 | 1					(S) x 10 ≠ (S) **	
2 | 8						
64 | 4	Start reproductrice zonder kaarttoevoer					
64 | 5	Start reproductrice met kaarttoevoer leesgang					
64 | 6	Start reproductrice met kaarttoevoer ponsgang					
64 | 7	Start reproductrice met kaartoevoer beide leesgangen					
64 | 8	Zet conditie: zijn we in intercyclus?					
64 | 9	Zet conditie: zijn we tussen rij 0 en rij 11?					
64 | 10	Zet conditie: was er een fout in de leesgang?					
64 | 11	Zet conditie: was er een fout in de ponsgang?					
64 | 16	STOP					

Van al deze opdrachten is een A-register versie (bovenste rij functiecijfers) en een S-register versie (onderste rij cijfers) beschikbaar (R betekent hier dus A of S). Voorzover in de beschrijving van de opdracht de registers niet optreden betekent dit dat beide versies dezelfde operatie uitvoeren. In de nu volgende beschrijving zullen we volstaan met een nadere explicatie van de A-register versie van deze opdrachten.

54/8/0
55/8/0 Deze opdrachten transporteren de inhoud van het A-register positief of negatief naar het S-register. (A) blijft ongewijzigd. (de S-register versies transporteren van het S-register naar het A-register) Tijdsduur 36 μ sec.

48 t/m 51/16/1 Bandleesopdrachten

De gebruikte ponsband heeft 5 gaatjes per rij; de 2 meest significante cijfers aan de ene zijde van het transport-gaatje en de 3 minst significante aan de andere. De bandleesopdrachten beschouwen de uitgang van de bandlezer als een getal met 5 significante cijfers aan de linkerzijde aangevuld gedacht met 22 nullen. Dit getal kan bij A of S worden opgeteld: (R) + (BL) $\Rightarrow$ R of ervan afgetrokken: (R) - (BL) $\Rightarrow$ R. Ook kan dit getal naar A of S getransporteerd worden (met positief of negatief teken) onder gelijktijdige vernietiging van de oorspronkelijke inhoud van A of S: (BL) $\Rightarrow$ R resp. - (BL) $\Rightarrow$ R. Wanneer een bandleesopdracht optreedt wordt, indien het mechanisch bandleesmechanisme inmiddels tot rust is gekomen allereerst de op dat moment onder het leesstation gelegen rij gaatjes afgelezen waarna het opstepmechanisme in werking wordt gezet. Daarna kan de machine verder gewoon doorrekenen: de voltooiing van de mechanische operatie geschiedt onder controle van aparte apparatuur. Indien een nieuwe bandleesopdracht gevonden wordt voordat de stepoperatie van de vorige opdracht voltooid is wacht de machine met de uitvoering van deze tweede opdracht totdat de vorige volledig voltooid is. De programmeur hoeft zich dus bij het gebruik van bandleesopdrachten niet te bekommeren om de relatieve timing van bandlezer en machine. Zoals bekend kunnen ongeveer 150 bandleesopdrachten per seconde plaats vinden.

54,55/16/1 Bandponsopdrachten

Deze opdracht zorgt ervoor dat de 5 minst significante cijfers van R (A of S) worden overgebracht op een rij van de band, met of zonder inversie. De regeling van de timing is hierbij net als bij de bandleesopdrachten. Daar de te ponsen informatie eerst in een apart bufferregister wordt overgebracht kan ook in dit geval onmiddellijk verder gerekend worden indien de vorige pons-of typeopdracht inmiddels volledig is afgelopen.

N.B. Bandponsopdrachten kunnen eerst uitgevoerd worden wanneer de vorige bandpons- of typeopdracht volledig klaar is: bandlees- en typeinstructies maken n.l. van dezelfde controleapparatuur gebruik.

48 t/m 51/16/2 Terugleesopdrachten

Deze opdrachten hebben ten doel een zo volledig mogelijke controle op het uittypen van resultaten mogelijk te maken. De decimaal of letter welke getypt wordt kan, in gedecodeerde vorm, van de typerelais worden teruggevoerd naar A- of S-register. Door tijdens het typen uit de teruggelezen symbolen het oorspronkelijke woord opnieuw op te bouwen controleert men niet alleen het typen zelf doch ook de eventuele deconversie van binair naar tientallig stelsel. Het symbool TR in het schema staat hier voor Type-Relais. Evenals bij de bandleesopdracht beschouwt de machine het nu 6 binaire cijfers bevattende symbool als een getal dat aan de meest significante zijde met 21 nullen aangevuld gedacht wordt.

De verschillende modificaties van deze opdracht komen ook overigens geheel overeen met die van de bandleesopdracht. De terugleesopdracht wordt direct uitgevoerd indien de relais welke het te lezen symbool bepalen (op grond van de vorige typeopdracht), reeds tot rust zijn gekomen. Is dit niet het geval dan wordt ook hier automatisch dit moment afgewacht alvorens tot uitvoering van de opdracht wordt overgegaan.

N.B. Er wordt dus niet gewacht tot het typen is afgelopen doch slechts tot de voor het typen benodigde relais hun definitieve stand bereikt hebben.

54,55/16/2 Typeopdrachten

De werking van de type-opdracht is volledig vergelijkbaar met die van de bandponsopdracht. Het symbool gevormd door de 6 minst significante cijfers van R (A of S) wordt, al dan niet na inversie, getypt.
Het gebruik van 6 binaire cijfers maakt het mogelijk zowel cijfers als letters te typen. Ook de lettershift kan door de machine worden bediend, zodat ook hoofdletters e.d. gebruikt kunnen worden.
De hierbij gebruikte code vindt men in onderstaand schema.

(TP)	kl.	HL	(TP)	kl.	HL
0	0	½	24	d	D
1	1	"	25	e	E
2	2	½	26	f	F
3	3	£	27	g	G
4	4	$	28	h	H
5	5	%	29	i	I
6	6	*f*	30	j	J
7	7	&	31	k	K
8	8	(	32	l	L
9	9	)	33	m	M
10	Tab		34	n	N
11	TWNR		35	o	O
12	-	_	36	p	P
13	+	=	37	q	Q
14	.	;	38	r	R
15	,	?	39	s	S
16	/	:	40	t	T
17	'	"	41	u	U
18	HL		42	v	V
19	kl		43	w	W
20			44	x	X
21	a	A	45	y	Y
22	b	B	46	z	Z
23	c	C	56 t/m 63	Spatie	

De opmerking omtrent de timing welke men aldaar vindt onder N.B. geldt evenzeer voor de typeopdrachten.

48-63/64/16 Stopopdracht

De stopopdracht behoeft geen commentaar. We wijzen er slechts op dat door gebruikmaking van de conditie-gehoorzamende versies hieronder dus ook de conditionele stopopdrachten vallen.

De ponskaarten-invoer en -uitvoerorganen

Voor de ponskaarten-invoer en -uitvoer wordt gebruik gemaakt van een ponskaartenmachine (reproductrice) welke over 2 banen beschikt. Een der banen kan uitsluitend voor invoer gebruikt worden, terwijl de andere zowel voor in- als voor uitvoer geschikt is.

De communicatie tussen de X-1 en de ponskaartenmachine is niet direct, doch vindt plaats via de tussenkomst van z.g. buffers. Dit houdt in dat bij het lezen de informatie van een kaart eerst voorlopig in een apart tussengeheugen wordt opgeborgen waaruit de machine dan later kan putten, terwijl het te ponsen cijfermateriaal van de X-1 uit eerst ook tijdelijk in een buffer geheugen bewaard wordt. Dit enigszins gecompliceerde systeem biedt enorme voordelen zowel uit een oogpunt van gemak voor de programmeur als uit snelheidsoverwegingen. We bespreken nu eerst de gang van zaken in de leesbaan, welke van beide de eenvoudigste structuur heeft.

In deze baan bevinden zich twee stellen van 80 afleesborstels welke precies een kaartafstand van elkaar verwijderd zijn. Het eerste stel noemen we de leesborstels, het tweede de controleborstels. Wanneer de ponskaartmachine een cyclus doorloopt (een slag maakt) wordt door de leesborstels een kaart afgetast welke zojuist uit het aanvoermagazijn is gekomen, terwijl de controleborstels de kaart aftasten welke gedurende de vorige slag onder de leesborstels is doorgegaan. De tijdscyclus van een slag van de ponskaartmachine (ca. $\frac{1}{2}$ seconde) denkt men zich verdeeld in 18 "punten", waarvan er 12 overeenkomen met de tijd nodig voor het lezen der 12 rijen ponsgaten in de kaart. De overige 6 vormen de zogenaamde intercyclus gedurende welke geen informatie uit de kaart wordt gelezen. De punten welke met de rijen van de kaart overeenkomen dragen de overeenkomstige nummers (X en Y is resp. punt 11 en 12), zodat de intercyclus de punten 13 t/m

18 omvat. (punt 10 ontbreekt) Wanneer de ponskaartenmachine na het lezen van een kaart stopt bevindt hij zich in het begin van punt 14, d.w.z. ongeveer 1 punt na het lezen van de laatste rij van de kaart.
Bij de communicatie tussen ponskaartmachine en X-1 doen zich 2 principiële moeilijkheden voor. De eerste is dat beide machines onderling niet gesynchroniseerd zijn en de tweede dat de kaart in eerste instantie rijgewijs gelezen wordt, terwijl de programmeur gewoonlijk slechts in de afzonderlijke kolommen geinteresseerd is.
Om aan deze moeilijkheden tegemoet te komen is nu voor de leesbaan een buffergeheugen beschikbaar waarin precies één kaartinhoud (960 bits) bewaard kan worden. Gedurende de punten 9....0, 11 en 12 wordt hierin de inhoud van de nieuw gelezen kaart opgeborgen. Deze bewerking geschiedt volkomen automatisch onder controle van signalen die uit de ponskaartmachine zelf worden afgeleid. De X-1 kan dus gedurende de tijd waarin dit gebeurt gewoon verder rekenen. Bovendien wordt eveneens automatisch de vroegere inhoud van de buffer, d.w.z. dus de informatie van de vorige gelezen kaart vergeleken met de gegevens welke nu aan de controleborstels verschijnen. Indien een afwijking geconstateerd mócht worden, wordt dit genoteerd in een apart geheugenelement dat later door middel van een speciale instructie door het programma kan worden uitgelezen. De programmeur kan dus zelf bepalen welke maatregelen in dat geval getroffen dienen te worden. Gedurende de tijd nodig voor lezen en vergelijken is het buffergeheugen van de X-1 uit niet te bereiken. Probeert men dit toch dan stopt de X-1. De X-1 kan de informatie in de buffer uitsluitend gebruiken in de intercyclus volgend op het lezen van de betreffende kaart. In dat interval, dat door het uitstellen van een nieuwe startopdracht uiteraard een onbepaalde lengte kan krijgen, is de buffer kolomsgewijze uitleesbaar. Elke kolom heeft een adres (uit de groep adressen welke voor het levend geheugen gereserveerd zijn) en het lezen van een kolom verloopt volkomen analoog aan het lezen van een snelle geheugenplaats. Alle instructies welke gewoonlijk op een geheugenplaats betrekking kunnen hebben kunnen ook nu worden gebruikt, behalve, in

het geval van de leesbaan, de schrijfinstructies.
Tenslotte doet zich nog de moeilijkheid voor, dat de code waarin de informatie op de kaart staat verschilt van de binaire code, welke in de machine wordt gebruikt, zodat in vele gevallen conversie zal dienen plaats te vinden. Om ook hieraan zoveel mogelijk tegemoet te komen, heeft elke kolom van de leesbaanbuffer niet één, maar twee adressen. Wanneer deze kolom met behulp van het eerste van deze wordt aangevraagd verschijnt in de machine eenvoudig een replica van het kolombeeld in de kaart d.w.z. een woord van 27 bits waarvan de 12 minst significante cijfers overeenstemmen met het gaatjespatroon van de kaartkolom em de overige 15 alle 0 zijn. Hierbij komt de Y-rij aan de minst significante, de 9-rij aan de meest significante zijde.
Wordt daarentegen het tweede adres gebruikt dan verschijnt in de machine een binair cijfer (0....9) in de normale machinecode. Dit binaire cijfer wordt gevormd uit de 10 cijferplaatsen van de betreffende kolom; de X en Y posities worden hierbij niet in aanmerking genomen. Het zal duidelijk zijn dat in dit geval alleen dan een correct binair cijfer ≤ 9 gevonden wordt wanneer in de cijferposities van de kolom hoogstens 1 gaatje geponst is. (een geponste nul en een kolom zonder ponsing geven beide een nul als resultaat)
Bij het hier gebruikte systeem wordt dus verondersteld dat de programmeur in staat is op een of andere manier uit te maken of de ponskaartmachine zich al dan niet in de intercyclus bevindt. Hiertoe dient een speciale instructie welke in de vorm van een conditiezetting antwoord verschaft op de vraag of de intercyclus aan de gang is. Het zal duidelijk zijn dat deze instructie in combinatie met een conditionele sprongopdracht de mogelijkheid schept op het begin van de volgende intercyclus te wachten wanneer in eerste instantie blijkt dat deze nog niet bereikt is.
In werkelijkheid is er ook nog een instructie welke in de vorm van een conditiezetting de vraag beantwoord of de ponskaartmachine zich in het "vrije" stuk tussen punt 0 en punt 11 bevindt. Met vrij wordt hier bedoeld het stuk tussen deze punten waar de programmeur desgewenst het tot dusver gelezen stuk van

de kaart uit de buffer kan laten lezen. Op dit punt in de cyclus zijn de punten 9-0 inmiddels gepasseerd zodat de numerieke informatie volledig beschikbaar is hoewel de X en Y ponsingen nog ontbreken. In feite bevat de buffer op dit ogenblik nog de X en Y ponsingen van de vorige kaart. Voor de leesbaan is deze faciliteit relatief onbelangrijk doch in de ponsbaan kan er dikwijls met voordeel gebruik van worden gemaakt.

Om dit laatste te verduidelijken dient nu een meer gedetailleerde beschrijving van de situatie in de ponsbaan gegeven te worden.

In het begin van deze baan, d.w.z. direct na het aanvoermagazijn bevindt zich een volledig stel (80) borstels waarmee de aangevoerde kaart gelezen kan worden.
Deze borstels hebben in twee gevallen hun nut. In de eerste plaats is het mogelijk de ponsbaan als leesbaan te gebruiken. In dit geval zal dus nimmer geponst worden. Het eerste stel borstels wordt in dit geval gebruikt op volkomen analoge wijze aan het eerste stel borstels in de leesbaan. Voorts is dit stel ook nodig wanneer de ponsbaan gebruikt wordt om additionele gegevens te ponsen in kaarten welke reeds bepaalde ponsingen bevatten. Om een volledige controle te bewerkstelligen dient dan n.l. na het ponsen niet alleen geconstateerd te worden dat de gewenste kolommen op de juiste wijze geponst zijn doch tevens dat geen extra ponsingen zijn verschenen in de oude kolommen op de kaart. Hiertoe is het dus nodig het kaartbeeld zoals dat er voor de ponsoperatie uitzag te kennen.

In de cyclus na die waarin de kaart gelezen is wordt het ponsen van de kaart voorbereid, in die zin dat gedurende de punten 9-12 van deze cyclus het ponsblok wordt ingesteld op het te ponsen patroon (Met de werkelijke ponsoperatie welke daarna plaats vindt hebben we hier weinig te maken).

In de 3e cyclus gebeurt er met de kaart welke we nu beschouwen niets. In de 4e cyclus passeert hij wederom een volledig stel leesborstels waarmee dus de uiteindelijke inhoud van de kaart na (bij)-ponsing kan worden vastgesteld.

Aangezien dus de controle pas drie cycli na het aanvankelijk lezen plaats kan vinden, is het arrangement van de benodigde buffers in het geval van de ponsbaan aanzienlijk gecompliceerder dan in het geval van de leesbaan. De inhoud van de kaart wordt in de eerste cyclus opgenomen in een buffer en is gedurende de eerstvolgende intercyclus ter beschikking van de programmeur. In de intercyclus wordt tevens een tweede buffer, de z.g. ponsbuffer van de X-1 uit gevuld met de te ponsen informatie welke tevens in de eerste buffer aan het daar reeds aanwezige patroon wordt toegevoegd. In de tweede cyclus wordt de inhoud van de ponsbuffer nu automatisch verwerkt en gebruikt tot instelling van het (mechanische) ponsmagazijn van de ponskaartmachine. De inhoud van de buffer met het nu volledige kaartbeeld blijft gedurende de ponscyclus en de daaropvolgende loze cyclus verder onaangetast en is ook voor de programmeur niet meer bereikbaar. In de 4e cyclus wordt op analoge wijze als in de leesbaan automatisch de inhoud van deze buffer vergeleken met de inhoud van de nu opnieuw gelezen kaart en de buffer opnieuw gevuld met de inhoud van de kaart welke nu uit het aanvoermagazijn komt en het eerste stel leesborstels passeert. Aangezien echter in de tussentijd nog twee andere kaarten uit het aanvoermagazijn kunnen zijn aangevoerd zullen dus drie buffers nodig zijn om een continue kaarttoevoer volgens het bovengeschetste systeem te verwezenlijken. Deze zijn dan ook inderdaad aanwezig, hoewel de programmeur hiervan niets merkt. Gedurende elke intercyclus heeft hij acces tot de zojuist gelezen kaart en hoewel deze dus telkens in een andere buffer wordt opgeborgen, kunnen voor het uitlezen hiervan toch steeds dezelfde adressen gebruikt worden. De nodige correcties hierop komen namelijk automatisch tot stand. In geval bij de controle een fout geconstateerd wordt, wordt dit ook bij de ponsbaan in een apart geheugenelement onthouden dat weer door middel van een speciale instructie die conditiezetting veroorzaakt ter beschikking van de programmeur staat.
Voor het lezen van informatie in de ponsbaan geldt hetzelfde als voor de leesbaan: elke kolom in de buffer heeft twee adressen welke respectievelijk een replica van de kolom of een binair

gecodeerd getal opleveren.

Bij de ponsbuffer is de situatie iets anders. Hier zijn n.l. voor elke kolom 3 adressen beschikbaar. Gebruikt men het eerste dan worden de 12 minst significante cijfers van een der interne registers als kolom naar de ponsbuffer gevoerd.

Met behulp van het tweede adres wordt het binaire getal $\leqslant 9$ dat zich in de 4 minst significante posities van een register bevindt vertaald in kaartcode en getransporteerd naar de tien cijferrijen van de gekozen kolom. Dit is dus precies de inverse operatie van die welke uit gebruik van het tweede adres in een bufferleesoperatie resulteert. Het derde adres kan worden gebruikt om de 2 minst significante bits uit een register te plaatsen in de X en Y posities van een bepaalde kolom. Dit adres werd toegevoegd om het op eenvoudige wijze mogelijk te maken X en Y ponsingen toe te voegen aan de te ponsen informatie nadat de numerieke gegevens afzonderlijk behandeld zijn. Wanneer naar een kolom met behulp van zijn verschillende adressen (of desnoods met herhaalde gebruikmaking van hetzelfde adres) verschillende gegevens worden gestuurd vindt in de ponsbuffer eenvoudig een superpositie plaats.

We dienen thans nog het geval te bezien dat men extra ponsingen wenst aan te brengen in een reeds gedeeltelijk geponste kaart en het bovendien zo is dat de aard van deze extra ponsingen afhangt van de oorspronkelijke gegevens op de kaart. Wil men in dit geval de ponskaartmachine op volle snelheid laten lopen dan heeft men er, gezien het bovenomschreven arrangement in de ponsbaan, dus voor te zorgen dat alle desbetreffende berekeningen in de intercyclus volgend op het lezen van deze kaart door het eerste stel borstels voltooid zijn.
Bovendien moeten in dit tijdsbestek de gevonden resultaten naar het ponsmagazijn worden gestuurd. Het valt te verwachten dat de hier beschikbare tijd in bijna alle gevallen voldoende zal zijn, doch er is niettemin gezocht naar middelen nog iets meer tijd beschikbaar te stellen.
Daartoe is de reeds in het voorgaande vermelde mogelijkheid geopend over de numerieke rijen van de gelezen kaart reeds te

beschikken in de vrije periode tussen de punten 0 en 11. Daardoor wordt althans wat de numerieke gegevens betreft de intercyclus als het ware 2 punten langer.
Opgemerkt moet nog worden dat het niet geoorloofd is informatie naar de ponsbuffer te sturen in de tijd tussen dit vrije interval en de intercyclus.

We geven nu een overzicht van de instructies welke met de operaties van de ponskaartmachine verband houden.
Daarbij noemen we allereerst de startopdrachten. Deze dragen de coderingen:

48/64/4 Start reproducer zonder kaarttoevoer

48/64/5 Start reproducer met kaarttoevoer in de leesgang

48/64/6 Start reproducer met kaarttoevoer in de ponsgang

48/64/7 Start reproducer met kaarttoevoer in beide gangen.

De startoperatie kan slechts plaatsvinden wanneer de machine zich bevindt in het stuk tussen het begin van punt 13 en punt 14 (Dit sluit de stoppositie in)
Wordt dus een startopdracht gegeven in dit interval, dan wordt deze onmiddellijk uitgevoerd. Geeft men echter een startopdracht op een zodanig ogenblik dat de machine niet in dit interval is, dan wordt met de uitvoering van deze opdracht gewacht tot dit interval bereikt wordt.
De normale gang van zaken waarbij de ponskaartmachine continu moet blijven lopen zal dus zijn dat men direct na het begin van de intercyclus een nieuwe startopdracht geeft. Indien de berekening welke men uit wil voeren zo lang duurt dat deze, wanneer de ponskaartmachine doorloopt niet binnen de gegeven periode kan worden afgewerkt, mag men echter de startopdracht gerust later geven. De ponskaartmachine stopt dan op punt 14 en gaat pas weer verder nadat de startopdracht is ontvangen.
De verschillende uitvoeringsvormen van de startopdracht zijn slechts onderscheiden in de wijze van kaarttoevoer. Dit behoeft geen nadere toelichting. Slechts zij er nog op gewezen dat het dus wel mogelijk is de kaarttoevoer in beide banen afzon-

derlijk te regelen doch niet het kaarttransport in de machine: men kan de kaartbeweging in beide banen tegelijkertijd doen plaatsvinden of stoppen, doch niet b.v. de kaarten in de pons-baan laten uitlopen en die in de leesbaan laten liggen.

Vervolgens de 2 opdrachten welke de programmeur vertellen of de ponskaartmachine zich in een vrije periode bevindt of niet.

48/64/8 Zet de conditie op 1 wanneer de ponskaartmachine in de intercyclus is. Zet anders de conditie op 0.

48/64/9 Zet de conditie op 1 wanneer de ponskaartmachine in de vrije periode tussen punt 0 en punt 11. Zet anders de conditie op 0.

Opmerking: Wanneer door gebruikmaking van een dezer opdrachten is vastgesteld dat de ponskaartmachine zich op het ogenblik waarop de test werd uitgevoerd in een van beide periodes bevond, houdt dit uiteraard geen enkele garantie in dat dit ook na het beeindigen van deze instructie nog het geval zal zijn.

Hierna noemen we de 2 opdrachten welke met het constateren van fouten bij het ponskaartlezen verband houden.

48/64/10 Zet de conditie op 1 wanneer de fout-indicatie van de leesgang een 1 aanwijst en herzet de fout-indicatie op 0.
Zet anders de conditie op 0.

48/64/11 Idem voor de ponsgang.

Opmerking: Het zal dus alvorens men begint met kaartlezen gewenst zijn de beide foutindicaties, door ze elk een maal uit te lezen, op nul te zetten daar men anders bij het constateren van een eventuele fout niet weet of deze in de nieuw gelezen kaart(en) is opgetreden of nog als een overblijfsel van vroeger moet worden opgevat.

Het zal duidelijk zijn dat voorzover de hiergenoemde opdrachten automatisch invloed op de conditie uitoefenen de normale conditiezettende faciliteiten niet gebruikt moeten worden.

We geven nu een beschrijving van de wijze waarop de adressen van de diverse bufferkolommen kunnen worden afgeleid. Dit ge-

beurt in hoofdzaak om der wille van de volledigheid aangezien de programmeur met de numerieke waarde van deze adressen zelden of nooit te maken zal hebben. Een korte aanduiding van de normale notatie van deze adressen zullen we laten volgen.
Ter wille van de bufferadressering worden 1024 adresplaatsen gereserveerd welke we ons voorlopig denken als gelegen aan het einde van de groep die voor het levend geheugen beschikbaar is.
Voor al deze adressen geldt dan: OR14 = 0, OR13 = OR12 = OR11 = OR10 = 1.
De overblijvende adresdigits hebben de volgende betekenis:

OR9, OR8 =	0 0	voor een decimaal adres
	0 1	voor een replica-adres
	1 1	voor een adres waarin een X Y ponsing wordt geplaatst.
OR7 = 0		specificeert de leesgang
OR7 = 1		specificeert de ponsgang
OR6...ORo		worden gebruikt om het kolom nummer (1-80) aan te geven.

Op papier ziet de notatie er echter als volgt uit.
Men schrijft na de gewenste opdrachtcode eerst het nummer van de gewenste kolom, daarna PR wanneer men de leesgang en PP wanneer men de ponsgang bedoelt en tenslotte een der getallen 1, 2 of 3 al naar gelang men een decimale, een replica of een X Y versie wenst. De combinaties PP en PR dragen hierbij het karakter ven lettercombinaties waarvoor de normale voorschriften gelden (zie hiervoor de invoerconventies). De daarna komende cijfers 1, 2 of 3 hebben echter niet de gewone paginaincrement-betekenis doch moeten uitsluitend als nummers worden gezien.
Opmerking: Zoals men ziet wordt het onderscheid tussen ponsen en lezen in de ponsbaan niet in het adres tot uitdrukking gebracht. Dit gebeurt automatisch doordat men voor lezen van in-opdrachten en voor ponsen van uitopdrachten gebruik maakt. Hierbij dient men er nog wel op te letten dat bij de bufferadressen geen gebruik gemaakt mag worden van de additieve uitopdrachten doch uitsluitend van de schoon uitopdrachten.

Tenslotte merken we nog op dat de machine zal stoppen wanneer een van de normale beveiligingen van de ponskaartmachine in actie komt (lege aanvoermagazijnen, volle aflegmagazijnen, verkreukelde kaarten e.d.) De stopoorzaak wordt in dat geval op lampjes op de console geindiceerd.
Een extra beveiliging is aanwezig welke er voor zorgt dat niet tengevolge van onoordeelkundige programmering of storingen in de electronische apparatuur teveel gaatjes in een kaart worden geponst waardoor het ponsblok beschadigd zou kunnen worden.

Ten behoeve van een snelle conversie van 10-tallig- naar 2-tallig stelsel en omgekeerd wanneer van de ponskaartinvoer en -uitvoer gebruik wordt gemaakt zijn nog 2 extra opdrachten toegevoegd, welke bekend staan als de "snelle vermenigvuldigingen met tien".
Hun omschrijvingen luiden als volgt:

62/32/0 Het 10-voud van de inhoud van het S-register wordt in AS geplaatst. Wanneer we het getal in S dus als een breuk opvatten (komma geheel links) staat na afloop van de operatie het gehele gedeelte van het 10-voud in A, het breukgedeelte in S.
De oorspronkelijke inhoud van A gaat verloren.
Deze instructie is speciaal van belang voor de overgang van 2- naar 10-tallig stelsel.

62/32/1 Het 10-voud van de inhoud van het S-register wordt in S geplaatst. Wat eventueel aan de meest significante kant uit S loopt gaat verloren. A blijft onaangetast.
Deze instructie is van belang voor de overgang van 10- naar 2-tallig stelsel.

De tijdsduur van beide opdrachten is 64 μ sec.

V. Interne controle faciliteiten

De beide vormen van geheugen van de X-1, dood en levend geheugen zijn onderworpen aan een ingebouwde z.g. pariteits-controle, d.w.z. het aantal énen per woord bij het wegschrijven van een getal in het levend geheugen of bij het bedraden in het dood geheugen door toevoeging van een 28e cijfer steeds oneven gemaakt.

Bij het aanhalen van opdrachten en getallen wordt steeds gecontroleerd of dit klopt. Wordt een afwijking geconstateerd dan stopt de machine direct, zodat men kan zien in welk woord de fout is opgetreden. In vele gevallen zal men ook het cijfer waarin de fout schuilt kunnen achterhalen zodat snel kan worden vastgesteld waar de storing precies is opgetreden. Met deze controle wordt dus iedere enkelvoudige (algemeen oneven-voudige) fout direct gevonden.

VI. Snelheid

De begrippen "additief" en "schoon" slaan op het onderscheid tussen opdrachten waarbij een echte optelling te pas komt en die waarbij slechts van een transport sprake is of ev. van een logische optelling of vermenigvuldiging.

Tijdsduur der opdrachten in X-1

1.	Normaal en B-ongemodificeerd	in	64 µsec
2.	B-gemodificeerd	in	72 µsec
3.	Normaal en B-ongemodificeerd	additief uit	76 µsec
4.	B-gemodificeerd	additief uit	84 µsec
5.	Normaal en B-ongemodificeerd	schoon uit	64 µsec
6.	B-gemodificeerd	schoon uit	72 µsec
7.	Absoluut en Communicatie additief		44 µsec
8.	Absoluut en Communicatie schoon		36 µsec
9.	Skip en niet-arithmetische Communicatie		32 µsec
10.	Vermenigvuldiging en Deling (alle gevallen)		500µsec

Al deze tijden gelden inclusief de accestijden voor de extractie van de betreffende opdracht en de te verwerken getallen.

VII. Handbediening en indicatieapparatuur

Ten behoeve van de operateur, speciaal bij het testen van nieuwe programma's, is op het controlepaneel van de machine een vrij uitgebreid aantal schakelaars, drukknoppen en indicatielampjes aanwezig. Met bedieningsknoppen is het o.a. mogelijk het programma stap voor stap te volgen, daarbij op lichtjes steeds de binaire inhoud der interne registers zichtbaar makend. Ook is het mogelijk de machine hetzij direct hetzij op een van te voren in schakelaars gespecificeerde punt van het programma te stoppen.
Opdrachten en getallen kunnen direct van uit schakelaars in binaire vorm worden ingebracht en eventueel uitgevoerd. Een decimaal toetsenbord maakt het tevens mogelijk decimaal gegeven getallen direct met de hand in een der interne registers te voeren. Tevens kan men met behulp van de decimale toetsen verschillende andere nuttige handelingen verrichten.

Voor nadere bijzonderheden zij verwezen naar het aparte geschrift dat de invoerconventies behandelt.