

MATHEMATISCH CENTRUM

2e BOERHAAVESTRAAT 49

AMSTERDAM

REKENAFDELING

Leiding: Prof. Dr Ir A. van Wijngaarden

FUNCTIONELE BESCHRIJVING VAN DE ARRA.

Door

E.W. Dijkstra.

MR 12

1953.

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM



VOORWOORD

Het volgende heeft betrekking op de ARRA, zoals hij zal zijn na de ingrijpende veranderingen, waartoe in de laatste maanden van 1952 werd besloten.

Na een functionele beschrijving wordt in eerste begin uiteengezet, op welke wijze wij de machine denken te gebruiken. De meest fundamentele afspraken, die in het invoerprogramma, de ponsconventies en de bouw der subroutines, tot uiting komen, zijn in dit rapport vastgelegd.

- - -

INHOUD:

	Blz.
1. INLEIDING	1
2. DE BESCHRIJVING DER OPDRACHTEN	3
Vermenigvuldigen	5
Delingen	6
+ en - 0	7
Quasi-opdrachten	8
Schuifopdrachten	8
3. DE OPDRACHTEN IN HET GEHEUGEN	10
4. DE COMMUNICATIE OPDRACHT	12
5. DE TIJDSDUUR DER OPDRACHTEN	14
6. DE BEDIENING	17
7. DE PONS BAND	20
De decimaal-binale conversie, gevolgd door de correctie	20
Opdrachten	21
Getallen	22
Controle-combinatie	22
"Blank tape"	23
8. DE BESCHRIJVING VAN HET INVOERPROGRAMMA	24
Lezing uit het handregister	24
Lezing van de band	26
Blok-schema (Lezing handregister)	29
Blok-schema (Lezing band)	30
Het invoerprogramma	31
9. DE TAAK VAN DE PROGRAMMEUR	33
10. HET GEBRUIK VAN SUBROUTINES	36
11. VOORBEELDEN	40
12. DE PERMANENTE GEHEUGENINHOUD	47
13. TABEL VAN DE MACHTEN VAN 2	49

- - - -

1. Inleiding.

De ARRA is een ADM (automatische digitale machine).

In het volgende wordt deze machine beschreven, voorzover dit van belang is voor degene, die de machine gebruikt: er zal beschreven worden, wat de machine doet, niet hoe de machine werkt.

Het automatisme bestaat daaruit, dat de machine na de nodige voorbereiding aan de gang gezet kan worden en dan lange reeksen elementaire operaties, als optellen, aftrekken, vermenigvuldigen en delen, achter elkaar op de juiste getallen uitvoert, zonder dat tijdens dit rekenproces nog door mensenhand hoeft worden ingegrepen; de gewenste antwoorden worden automatisch met behulp van een elektrische schrijfmachine getypt.

Waar deze nodige voorbereiding uit bestaat, en hoe deze moet worden ingericht, is het onderwerp van dit rapport.

Het is duidelijk, dat, voordat de machine aan het rekenen kan gaan, zowel de reeks operaties, specifiek voor het betroffen probleem, als de zuiver numerieke gegevens, op een zodanige wijze reeks voorhanden moeten zijn, dat ze voor het rekenend gedeelte van de machine toegankelijk zijn. Daartoe worden deze gegevens - noodzakelijke informatie dus! - van te voren, in een speciaal daartoe ingericht gedeelte van de machine, het geheugen, "geschreven". Tevens zal het geheugen gebruikt worden om tussentijdse resultaten, die verder niet interessant zijn, op te bergen, te "onthouden", totdat ze later in de berekening aan bod zijn, om weer verder verwerkt te worden.

Omdat een geheugen, waarin getallen etc. opgeborgen worden als knikkers in een zak, voor een efficiënte gang van zaken nogal onpractisch zou zijn, is het geheugen systematisch ingericht: het geheugen is analoog met een bord met genummerde vakjes, waarvoor men staat met het krijt in de ene, met de spons in de andere hand. Het geheugen bevat 102^4 dergelijke vakjes, de zg. "geheugenplaatsen" of "adressen", genummerd van 0 t/m 1023. (Het woord "adres" wordt in twee betekenissen gebruikt, zowel voor de geheugenplaats zelf, als voor zijn nummer). Deze adressen zijn in staat zg. "woorden" te bergen: een woord bestaat uit een reeks, voor het geheugen bevatte, voor de rest van de machine begrijpelijke symbolen. Een woord kan onbepaald vaak gelezen worden en blijft in dit adres staan, totdat het door een ander woord overschreven wordt. (De spons wordt zo laat mogelijk gehanteerd: men maakt een vakje pas schoon, als men er iets anders in wil schrijven. Dit geschiedt automatisch: "vergeten" is geen aparte operatie).

Zo wordt een getal, dat als tussenresultaat gevormd wordt, op een bepaald adres neergeschreven. Iedere keer, dat later de berekening dit getal gebruikt, wordt verwezen naar "het getal op dat en dat adres". In overeenstemming hiermede pleegt de specificatie van de operaties, naast de aard van de bewerking (aftrekken of vermenigvuldigen etc.) naar adressen te refereren.

Woorden doen ook dienst ter specificatie van de reeks van operaties, die uitgevoerd moeten worden, d.w.z. woorden kunnen door de machine op verschillende wijze geïnterpreteerd worden.

De inhoud van adres n , dus het woord op adres n , geven we met het symbool (n) aan. Willen we ook de interpretatie van dit woord tot uitdrukking brengen, dan bezigen we de volgende notaties:

met $[n]$ geven we aan het woord (n) , geïnterpreteerd als geheel getal. Dit getal, dat zowel positief als negatief kan zijn, moet in absolute waarde kleiner zijn dan $2^{29} = 536879012$;

met $\{n\}$ geven we aan het woord (n) , geïnterpreteerd als breuk. Deze breuk, die zowel positief als negatief kan zijn, moet in absolute waarde kleiner zijn dan 1;

met $\langle n \rangle$ geven wij aan het woord (n) , beschouwd als aggregaat van twee opdrachten. Een "opdracht" is een hoeveelheid informatie, die een elementaire operatie specificeert. Er gaan twee opdrachten in een woord.

Een tweede orgaan van de machine is de besturing. Dit orgaan (een goed verstaander!) haalt een half woord uit het geheugen, interpreteert dit als opdracht en voert deze opdracht uit: meestal zal bij deze uitvoering voor een tweede maal contact met het geheugen tot stand moeten worden gebracht: of om een getal te lezen (dat bv. opgeteld moet worden, of waardoor gedeeld moet worden), of om een resultaat ergens in het geheugen neer te schrijven, opdat het te zijner tijd beschikbaar zij. De uitvoering van deze opdracht eindigt met het signaal, dat nu de volgende opdracht kan worden gelezen en uitgevoerd.

Het rekenkundig orgaan is het derde onderdeel van de machine. Dit bestaat uit twee registers, A en S genaamd, welke inhoud, in analogie met de adressen, met (A) en (S) wordt aangegeven. In functie vertonen zij enige overeenkomst met het resultaatregister en het omwentelingenregister van een tafelrekenmachine. Zo verschijnt bij deling een quotiënt in S, terwijl de rest in A achterblijft. Ze hebben echter dezelfde lengte (capaciteit van een woord, net als de adressen in het geheugen) en in beide kan worden opgeteld en afgetrokken. Het derde noodzakelijke register van een rekenmachine, het instelregister of te wel het toetsenbord, bestaat wel, nl. het register, genaamd M, maar de programmeur heeft hiermede niets te maken, omdat de besturing de getallen, die uit het geheugen komen, er automatisch op "inslaat".

2. De beschrijving der opdrachten.

De opdrachten, tot welker uitvoering de ARRA in staat is, bestaan uit twee gedeelten:

- 1) functiegedeelte, dat de aard van de bewerking specificieert,
- 2) adres, d.w.z. een getal, dat op enkele uitzonderingen na, de geheugenplaats aangeeft, waar het getal staat, of komt te staan, waarop de bewerking betrekking heeft.

De opdrachten zijn genummerd van 0 t/m 24 en verrichten de volgende operaties:

0/n	vervang (A) door (A)+(n)
1/n	vervang (A) door (A)-(n)
2/n	vervang (A) door (n)
3/n	vervang (A) door -(n)
4/n	vervang (n) door (A)
5/n	vervang (n) door -(A)
6/n	conditionele besturingsverplaatsing naar n a
7/n	besturingsverplaatsing naar n a
8/n	vervang (S) door (S)+(n)
9/n	vervang (S) door (S)-(n)
10/n	vervang (S) door (n)
11/n	vervang (S) door -(n)
12/n	vervang (n) door (S)
13/n	vervang (n) door -(S)
14/n	conditionele besturingsverplaatsing naar n b
15/n	besturingsverplaatsing naar n b
16/n	vervang [AS] door [n].[S]+[A]
17/n	vervang [AS] door -[n].[S]+[A]
18/n	vervang [AS] door [n].[S]
19/n	vervang [AS] door -[n].[S]
20/n	deel [AS] door [n], plaats quotient in S, rest in A
21/n	deel [AS] door -[n], plaats quotient in S, rest in A
22/n	"schuif A→S", d.w.z. vervang [AS] door [A]. ²⁹⁻ⁿ
23/n	"schuif S→A", d.w.z. vervang [SA] door [S]. ³⁰⁻ⁿ
24/n	communicatie opdracht

De zuiver additieve (0,1,8,9) en administratieve (2,3,4,5,10,11,12,13) opdrachten zijn door de bovengegeven omschrijving duidelijk; voor het begrip van de andere opdrachten is echter een meer gedetailleerde kennis nodig over de wijze waarop de getallen in de registers worden voorgesteld.

De ARRA werkt in het tweetalig stelsel, geheugenplaatsen (adressen) en de registers A en S zijn alle in staat een getal te bergen

van 30 binaire cijfers, die elk slechts de waarde 0 of 1 aan kunnen nemen. Deze cijfers, die we nummeren $d_0, d_1, d_2, \dots, d_{29}$, te beginnen met de plaats van de eenheden, geven resp. aan het aantal eenheden, tweevouden, viervouden etc. Een bepaald cijfer uit een bepaald woord geven we aan door een index, rechts onderaan het symbool van het woord: dus $(x)_1$ is het cijfer op de eenhedenplaats van x ; evenzo $(A)_k$ en $(S)_1$. Het cijfer d_{29} heeft een speciale functie en bepaalt het teken van het getal: als $d_{29} = 0$ is, interpreteert de machine het getal positief, als $d_{29} = 1$ is, interpreteert de machine het getal negatief, met die absolute waarde, die men krijgt, door op alle cijferplaatsen 0 door 1, 1 door 0 te vervangen.

$$\text{In formule } x = \sum_{k=0}^{29} (d_k - d_{29}) 2^k = \sum_{k=0}^{28} (d_k - d_{29}) 2^k.$$

De eerste consequentie is, dat er twee representaties voor nul zijn, te weten alle $d_k = 0 (+0)$ en alle $d_k = 1(-0)$; de tweede consequentie is, dat de maximum absolute waarde, die de machine aan een getal toe kan kennen, gelijk is aan $2^{29} - 1 = 536870911$.

Omdat d_0 betrekking heeft op de laagste macht van 2, wordt gezegd, dat d_0 de laagste cijferplaats is; d_{29} , het hoogste cijfer wordt het tekencijfer genoemd. Rekenkundig gedraagt d_{29} zich, wat betreft optelling en aftrekking, als andere cijfers. Noemen wij de waarde, die de machine interpreteert (met inbegrip van teken) x , de (non-negatieve) waarde, die het getal aanneemt, als men d_{29} zonder meer op de macht 2^{29} betrekking laat hebben, y , dan hangen deze samen volgens de betrekking

$$x = (y + 2^{29} - 1) \pmod{2^{30} - 1} - (2^{29} - 1). \quad (\text{Zie blz. 9 onderaan})$$

Het symbool (AS) is alleen van toepassing, als de tekencijfers van A en S aan elkaar gelijk zijn. Met het symbool (AS) wordt genoteerd het getal, dat men krijgt door A aan de hoge kant van S te zetten, "ze achter elkaar staand te beschouwen", met weglating van het tekencijfer van S, dat slechts een herhaling is van het tekencijfer van A. Zo is $(AS) = (A) \cdot 2^{29} + (S)$. Effectief beslaat dus (AS) 58 binalen en een tekencijfer. Met (SA) wordt genoteerd het getal, dat men krijgt door S aan de hoge kant van A te zetten, waarbij het tekencijfer van A de rol speelt van een gewoon binair cijfer; het teken wordt bepaald door het tekencijfer van S. Effectief beslaat (SA) dus 59 binalen plus het tekencijfer.

Echter kunnen wij de machine ook met breuken laten manipuleren, door boven beschreven bewerkingen op gehele getallen systematisch anders te interpreteren. Bij de additieve bewerkingen is de plaats waar we de komma denken, irrelevant, mits deze in de termen der som maar op dezelfde plaats staat; dit is ook de plaats, waar we de komma in het antwoord moeten denken. Bij de vermenigvuldiging is dit echter anders. De twee meest voorkomende andere interpretaties van

de vermenigvuldiging bestaan daar uit, dat men in één van de of in beide factoren de komma beschouwt, onmiddellijk achter het tekencijfer; men beschouwt dan dus breuken, in absolute waarde kleiner dan 1. Deze twee interpretaties zijn dienovereenkomstig verbonden door de relatie $[x]2^{-29} = \{x\}$.

Vermenigvuldiging van twee breuken.

De vermenigvuldigopdracht $18/n$ is beschreven, wanneer we (AS), (n) en (S) interpreteren als gehele getallen $[AS]$ (abs. kleiner dan 2^{58}), $[n]$ en $[S]$ (beide abs. kleiner dan 2^{29}).

Hoe we het antwoord moeten interpreteren, als we $\{n\}$ en $\{S\}$ beschouwen, vinden we, als we $[n]2^{29} = \{n\}$ en $[S]2^{-29} = \{S\}$ substitueren in

$$\begin{aligned} [n] \cdot [S] &= [AS] \quad , \text{ wat geeft:} \\ \{n\} \cdot \{S\} &= [AS] \cdot 2^{-58} = \{AS\} . \end{aligned}$$

Dit betekent, dat we bij de vermenigvuldiging van twee breuken het antwoord in (AS) vinden, wanneer we de komma "58 plaatsen naar links" interpreteren, d.w.z. onmiddellijk achter het tekencijfer van A. Deze interpretatie, die leidt tot een breuk, die eveneens in absolute waarde kleiner is dan 1, kunnen we doeltreffend als $\{AS\} = [AS] \cdot 2^{-58}$ noteren. Niet afgerond vindt men de vermenigvuldiging van twee breuken door $18/n$ beschreven als

$$\{n\} \cdot \{S\} = \{A\} .$$

Wanneer we afgerond wensen te vermenigvuldigen, dan moet dit product (van te voren) vermeerderd worden met $\pm 0.5 \times 2^{-29} = \pm 2^{-30}$ (afh. van het teken van het product). Indien we het product even positief veronderstellen, vinden we als volgt de procedure voor de afgeronde vermenigvuldiging. We wensen uit te rekenen:

$$\begin{aligned} \{AS\} &= \{n\} \cdot \{S\} + 2^{-30} \quad ; \text{ vermenigvuldiging met } 2^{58} \text{ geeft} \\ [AS] &= [n] \cdot [S] + 2^{28} . \end{aligned}$$

Dit resultaat kan met behulp van de additieve vermenigvuldigopdracht $16/n$ gevormd worden, als men met een 2-opdracht aanvankelijk $[A] = 2^{28}$ (dus $\{A\} = 0.5$ maakt. Blijkt het product negatief te zijn, dan had de afronding in de andere richting plaats moeten hebben. $\{A\}$ worde alsnog met 2^{-29} (dus $[A]$ met 1) verminderd.

Vermenigvuldiging van breuk en geheel getal.

Dit geval doet zich voor, als we $[n] \cdot \{S\}$ of $\{n\} \cdot [S]$ willen uitrekenen. Substitutie van $[S] \cdot 2^{-29} = \{S\}$ of $[n] \cdot 2^{-29} = \{n\}$ geeft:

$$[n] \cdot \{S\} = \{n\} \cdot [S] = [AS] \cdot 2^{-29} .$$

In (AS) moet de komma dus "29 plaatsen naar links" geïnterpreteerd worden, d.w.z. juist tussen A en S: in A staat het gehele gedeelte, in S het breukgedeelte, en de beschrijving van 18/n luidt in deze interpretatie:

$$[n] \cdot \{S\} = \{n\} \cdot [S] = [A] + \{S\} .$$

Delingen.

Bij de delingen is het noodzakelijk, dat (A) en (S) van te voren hetzelfde teken hebben. De rest is bepaald als de in absolute waarde kleinste rest met dit teken (dus het teken van het deeltal).

De deling 20/n is beschreven voor het geval dat (AS), (n), (S) en (A) geïnterpreteerd worden als [AS] (abs. kleiner dan 2^{58}), [n], [S] en [A] (abs. kleiner dan 2^{29}).

Indien het deeltal alleen S beslaat, moet de deling voorafgegaan worden door 23/0: A wordt dan schoongemaakt onder controle van het teken van (S). Indien het deeltal alleen A beslaat, moet de deling voorafgegaan worden door 22/0: S wordt schoon gemaakt onder controle van het teken van (A).

Delingen door breuk, met als quotiënt een geheel getal.

De deling, die geschreven kan worden als $[AS] \rightarrow [n] \cdot [S] + [A]$, kan geïnterpreteerd worden als deling, die een geheel quotiënt aflevert, terwijl we door een breuk delen (bv. 1.75, gedeeld door 0.5 = 3, rest 0.25), als we de betrekking met 2^{-29} vermenigvuldigen:

$$2^{-29} [AS] = [A] + \{S\} \rightarrow \{n\} \cdot [S] + \{A\} .$$

Delingen met als quotiënt een breuk.

Vermenigvuldigen we de betrekking $[AS] \rightarrow [n] \cdot [S] + [A]$ met 2^{-58} , dan vinden we de interpretatie, waarbij twee breuken op elkaar gedeeld worden, en ook het antwoord een breuk is, nl.

$$\{AS\} \rightarrow \{n\} \cdot \{S\} + \{A\} \cdot 2^{-29}$$

De rest $\{A\} \cdot 2^{-29}$ zal meestal dan niet meer van belang zijn.

Als het deeltal als breuk van één woordlengte gegeven is, delen we zodoende $\{A\}$ door $\{n\}$. Willen we deze deling afgerond laten geschieden, dan moet het quotiënt met $\pm 2^{-30}$ worden vermeerderd, al naar gelang van het teken van het quotiënt. Dit betekent, dat we in S de halve deler, met het teken van het deeltal moeten plaatsen, dus

$$\{S\} = \pm \frac{1}{2} \{n\}, \quad \text{met het teken van } A;$$

dan deelt $20/n \{A\}$ afgerond door $\{n\}$.

Deelt men een geheel getal op een ander geheel getal, zodat het antwoord een breuk is, dan deelt men $[A]$ door $[n]$; zij deze deling afgerond, dan wordt $[S] = \pm \frac{1}{2} [n]$ gemaakt.

Voor alle delingen moet (A) absoluut kleiner zijn dan (n), waarbij (A) en (n) gelijkkelijk geïnterpreteerd dienen te worden.

+ en - nul.

Additieve bewerkingen in A en S, die als uitkomst nul opleveren, geven als regel -0.

+0 verschijnt in A

- 1^e bij een opgaande deling met positief deeltal
- 2^e bij een vermenigvuldiging, die een positief resultaat aflevert, dat alleen S beslaat
- 3^e door +0 uit het geheugen met een 2-opdracht in A te plaatsen
- 4^e door -0 uit het geheugen met een 3-opdracht in A te plaatsen
- 5^e door 23/0, als (S) positief is
- 6^e als (A) = +0, wordt dit niet gewijzigd door optellen van +0 of aftrekken van -0.

+0 verschijnt in S

- 1^e bij een deling, als het quotiënt nul is, en deler en deeltal hetzelfde teken hebben
- 2^e bij een vermenigvuldiging, die een positief resultaat aflevert, dat alleen A beslaat
- 3^e door +0 uit het geheugen met een 10-opdracht in S te plaatsen
- 4^e door -0 uit het geheugen met een 11-opdracht in S te plaatsen
- 5^e door de opdracht 22/0, als (A) positief is
- 6^e als (S) = +0, wordt dit niet gewijzigd door optellen van +0 of aftrekken van -0.

De "schone vermenigvuldigingen" $18/n$ en $19/n$ geven, als het antwoord nul is, nul in A en S, met het teken van het product (dus $18/n$: teken (n) x teken (S), en $19/n$: - teken (n) x teken (S)).

De additieve vermenigvuldiging, $16/n$ en $17/n$ geven, indien het antwoord toevallig nul is -0 , behalve als aanvankelijk $(A) = +0$, $(S) = +0$, en $n = +0$ bij $16/n$ of $(n) = -0$ bij $17/n$.

Quasi-opdrachten.

Komt de besturing abusievelijk een van de quasi-opdrachten $25/..$ t/m $31/..$ tegen, dan stopt de machine.

De schuifopdrachten.

Nauwkeurige analyse der schuifopdrachten $22/n$ en $23/n$ leert ons, dat we deze opdrachten ook als volgt kunnen beschrijven

$$22/n, A \rightarrow S, n \leq 31.$$

Eerste slag: S wordt schoongemaakt onder controle van A, d.w.z. alle cijfers in S worden gelijkgesteld aan het tekencijfer van A $= (A)_{29}$.

Tweede slag: De inhoud van A wordt n plaatsen naar de lage kant opgeschoven, de aan de hoge kant vrijkomende cijfers in A worden aangevuld met het tekencijfer van A, de n laagste cijfers van A verschijnen in dezelfde volgorde op de n hoogste plaatsen in S, het tekencijfer van S niet meegerekend. ("Schuiven onder het tekencijfer van S door").

$$23/n, S \rightarrow A, n \leq 31.$$

Eerste slag: A wordt schoongemaakt onder controle van S, d.w.z. alle cijfers in A worden gelijkgesteld aan het tekencijfer van S $= (S)_{29}$.

Tweede slag: De inhoud van S wordt n plaatsen de lage kant opgeschoven, de aan de hoge kant in S vrijkomende cijferplaatsen worden met het tekencijfer van S aangevuld, de n laagste cijfers van S verschijnen op de n hoogste plaatsen in A, het tekencijfer van A wel meegerekend. ("Schuiven door het tekencijfer van A heen").

Om ± 0 te verkrijgen in A of S geeft men resp. de opdrachten $23/0$, $22/0$. $22/n$ is te beschouwen als deling door 2^n van (A), die het niet afgeronde quotiënt in A achterlaat. Evenzo geldt dit met betrekking tot $23/n$ en S.

$22/29$ schuift juist A naar S.

$23/30$ schuift juist S naar A.

$22/30$ en $22/31$ schuiven 1, resp. 2 plaatsen verder; de 1, resp. 2 binalen, die aan de lage kant van S verdwijnen, komen niet aan de hoge kant van A terug! Mutatis mutandis geldt hetzelfde met betrekking tot $23/31$.

Deze gevallen uitsluitend, zijn deze opdrachten als volgt te omschrijven, als A' en S' betrekking heeft op A en S voor, A'' en S'' op A en S na de bewerking.

22/n	$n \leq 29$	23/n	$n \leq 30$
	$(A'')_k = (A')_{29}$ $29-n \leq k \leq 29$		$(S'')_k = (S')_{29}$ $29-n \leq k \leq 29$
	$(A'')_k = (A')_{k+n}$ $0 \leq k \leq 29-n$		$(S'')_k = (S')_{k+n}$ $0 \leq k \leq 29-n$
	$(S'')_{29} = (A')_{29}$		$(A'')_k = (S')_{k+n-30}$ $30-n \leq k \leq 29$
	$(S'')_k = (A')_{k+n-29}$ $29-n \leq k \leq 28$		$(A'')_k = (S')_{29}$ $0 \leq k < 30-n$
	$(S'')_k = (A')_{29}$ $0 \leq k < 29-n$		

Opm. Bij de aanvang van vermenigvuldigingen, delingen en schuifoperaties is $-(1-p)$ (alias $-(2^{29}-1)$) als registerinhoud verboden, tenzij het direct uit het geheugen in het register is geplaatst (door 2/n, 3/n, 10/n of 11/n).

Als een vermenigvuldiging onmiddellijk door een deling gevolgd wordt, geldt voor deze deling de restrictie niet.

Explicatie van de notatie op blz.4:

$$x = a \pmod{b}, \text{ als } x \equiv a \pmod{b} \text{ en } 0 \leq x < b .$$

3. De opdrachten in het geheugen.

Het geheugen bestaat uit 1024 adressen, genummerd van 0 t/m 1023, elk in staat een woord van 30 binaire cijfers te bergen. Het adres in een opdracht, maximaal $1023 = 2^{10} - 1$, kan dus met behulp van 10 binaire cijfers worden gespecificeerd. Het opdrachtnummer, dat altijd kleiner is dan $32 = 2^5$ kan met behulp van 5 binaire cijfers worden vastgelegd: voor de specificatie van een willekeurige opdracht zijn dus 15 binairen nodig en voldoende; in overeenstemming hiermede wordt de opdracht f/n dan ook in principe geborgen als het getal $1024 \cdot f + n$. Omdat dit getal slechts de halve woordlengte beslaat, worden twee opdrachten per woord geborgen, die ter onderscheiding van elkaar de a-opdracht (lage kant) en de b-opdracht (hoge kant) genoemd worden:

de a-opdracht: $d_{14}, d_{13}, d_{12}, d_{11}, d_{10}$ geven als binair getal <32 , het opdrachtnummer aan.

$d_9, d_8, d_7, d_6, d_5, d_4, d_3, d_2, d_1, d_0$ geven als binair getal <1024 het adres aan.

de b-opdracht: $d_{29}, d_{28}, d_{27}, d_{26}, d_{25}$ geven als binair getal <32 het opdrachtnummer aan.

$d_{24}, d_{23}, d_{22}, d_{21}, d_{20}, d_{19}, d_{18}, d_{17}, d_{16}, d_{15}$ geven als binair getal <1024 het adres aan.

Staat op na de opdracht u/x, op na de opdracht u/y, dan wordt dit genoteerd $\langle n \rangle = v/y; u/x$, dus de b-opdracht voorop.

Doordat de besturing de opdrachten in een bepaalde volgorde uit het geheugen selecteert, is de machine in staat een lange reeks bewerkingen achtereenvolgens te verrichten, n.l.: behoudens besturingsverplaatsingen geldt:

na een a-opdracht wordt de b-opdracht uit dezelfde geheugenplaats gelezen, na een b-opdracht is de a-opdracht uit de volgende geheugenplaats aan de beurt; hiermede is "de volgende opdracht" gedefinieerd.

Ontmoet echter de besturing in een dergelijke reeks de opdracht 7/n, dan "verspringt de besturing" en gaat na (en volgende, etc.) gehoorzamen; de opdracht 15/n verwijst de besturing analoog naar na.b.

Opdrachten 6/n en 14/n hebben hetzelfde resultaat als 7/n, resp. 15/n, indien bij het laatste schrijven uit A of S het betrokken getal het arithmetisch orgaan als positief getal (incl. +0) verlaten heeft; deze voorwaarde kan op vijf wijzen vervuld zijn:

door de opdracht	4/n	als (A)	op dit moment	positief	is (incl.+0)
" "	"	5/n	" " " "	negatief	" (" -0)
" "	"	12/n	" (S)	positief	" (" +0)
" "	"	13/n	" " " "	negatief	" (" -0)
" "	"	24/8	" (A)	positief	" (" +0)

(Typ opdracht)

Is de voorwaarde niet vervuld, dan wordt de besturingsverplaatsing "niet gehoorzaamd", d.w.z. de (als boven gedefiniëerde) volgende opdracht wordt gelezen, etc. (m.a.w. de machine gaat dan gewoon door).

Het vervuld zijn van boven genoemde voorwaarde wordt elliptisch aangeduid met "laatst geborgen getal positief", het niet vervuld zijn met "laatst geborgen getal negatief".

4. De communicatie-opdracht.

De opdrachten 24/n spelen een speciale rol, omdat het hier gaat om operaties, tijdens welke uitvoering niet ten tweeden male verbinding met het geheugen tot stand gebracht wordt. De naam "communicatie-opdracht" is ontleend aan het feit, dat onder deze groep vallen, die opdrachten, die de communicatie met de buitenwereld (bandlezen en typen) verzorgen. De 24-opdrachten omvatten verder een groep min of meer onafhankelijke operaties; het numerieke gedeelte, dat in dit geval uit de aard der zaak niet meer betrekking heeft op een plaats in het geheugen, blijft men het adres noemen; het adres specificiert nu de aard der bewerking volgens onderstaande tabel:

24/0 en 24/1	"skip-opdracht", d.w.z. zonder noemenswaardig tijdsverlies gaat de besturing op de volgende opdracht over.
24/2	"+ conditionele stop", d.w.z. als "laatst geborgen getal positief" stopt de machine, als "laatst geborgen getal negatief" gaat de besturing op de volgende opdracht over.
24/3	"- conditionele stop, d.w.z. als "laatst geborgen getal negatief" stopt de machine, als "laatst geborgen getal positief" gaat de besturing op de volgende opdracht over.
24/4	"lees band", d.w.z. vervang $[A]$ door $-[B]$, als B het register is, waarin de gelezen pentade primair wordt opgeborgen. $[B]$ is dus niet groter dan 31; na afloop wordt de band een plaats opgeschoven. (Zie de Ponsband).
24/5	"Vervang $[A]$ door $-[H]$ ", d.w.z. lezing uit het handregister, voor het invoeren van decimale getallen. $[H]$ is niet groter dan 15. (Zie de Bediening).
24/6	"Vervang (A) door (OT)" (OT = opdrachtenteller). Dit is de enige opdracht, waarbij de bewerking beïnvloed wordt door de plaats in het geheugen, waar de opdracht staat: de 15 hoogste cijfers van A worden gelijkgesteld aan nul; de 15 laagste cijferplaatsen (de a-positie in A) wordt geplaatst: $7/x+1$ (d.w.z. $7 \times 10^{24} + x+1$) als 24/6 staat op xa

15/x+1 (d.w.z. $15 \times 1024 + x+1$) als 24/6 staat op xb.

In de a-positie van de accumulator komt dus de opdracht, die, indien gehoorzaamd, de besturing verwijst naar de opdracht, die 2 (halve) plaatsen verder staat dan 24/6.

- 24/7 "Vervang (A) door (GS)" (GS = getalschakelaar, d.w.z. 30 schakelaars elk met twee standen, die een willekeurig woord van 30 binanen kunnen aangeven (zie: De bediening).
- 24/8 "Type-opdracht, vervang [U] door het gehele getal, gevormd door de vier laagste cijfers van A, en type een hierdoor bepaald symbool. (Zie onder).
- 24/12 "Vervang [A] door [U] (voor controle op uit-typing en deconversie).
- 24/16 "Vervang [AS] door $10 \times [S] + [A]$.
- 24/17 "Vervang [AS] door $10 \times [S]$.
- 24/n "Als $512 \leq n \leq 767$, vervang [A] door $[A] + n - 512$.
Als $768 \leq n \leq 1023$, vervang [A] door $[A] + n - 1023$.

ad 24/8: onderstaande tabel geeft de relatie tussen [U] en de operatie die door de schrijfmachine wordt verricht:

[U]	operatie	[U]	operatie:
0	typt 0	8	typt 8
1	typt 1	9	typt 9
2	typt 2	10	Tabuleert
3	typt 3	11	TWNR *)
4	typt 4	12	typt -
5	typt 5	13	typt +
6	typt 6	14	typt .
7	typt 7	15	typt spatie

*) TWNR = Terug Wagen, Nieuwe Regel.

5. De tijdsduur der opdrachten.

De 1024 geheugenplaatsen zijn verdeeld in 64 "kanalen", elk bestaand uit 16 opeenvolgende plaatsen: de 6 hoogste cijfers van het adres specificeren dan het kanaal, de 4 laagste de plaats in het kanaal; de selectie van het kanaal geschiedt met behulp van relais, de selectie van de juiste plaats in het kanaal geschiedt elektronisch.

De selectie-apparatuur is in duplo uitgevoerd:

de opdracht-selectie, die, onder controle van de inhoud van de opdrachtenteller (en a-b indicatie) er voor zorgt, de juiste (a of b) opdracht gelezen wordt, en

de getal-selectie, die tijdens de uitvoering van een opdracht, onder controle van het adres van de zojuist gelezen opdracht, met de bedoelde geheugenplaats verbinding tot stand brengt, hetzij om te schrijven, hetzij om te lezen. Bij de opdrachten 6/n, 7/n, 14/n, 15/n, 22/n, 23/n en 24/n treedt de getalselectie niet in werking.

Indien twee onmiddellijk na elkaar gelezen opdrachten uit hetzelfde kanaal komen, blijven de relais van de opdrachtselectie onveranderd staan; hetzelfde geldt voor de relais van de getalselectie, indien twee opeenvolgende getalselecties betrekking hebben op hetzelfde kanaal. Door het programma zo in te richten, dat zowel in de opdrachtselectie, als in de getalselectie kanaalwisselingen zoveel mogelijk vermeden worden, is aanzienlijke tijdswinst te boeken; elke kanaalwisseling betekent nl. een tijdsverlies van 20 à 40 msec.

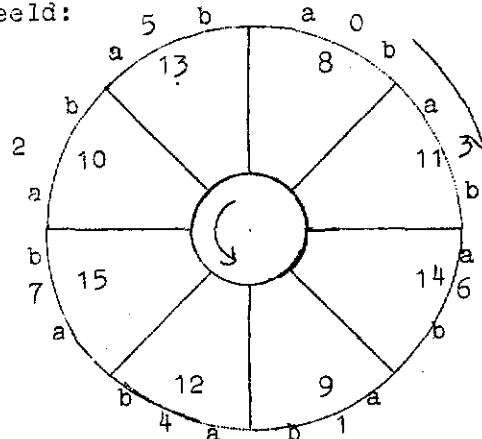
Een tweede manier om de snelheid van een programma te verhogen is het bekorten van de zg. wachttijden. De constructie van het geheugen (magnetische trommel met omwentelingstijd = 20 msec) brengt met zich mee, dat verbinding met een bepaalde geheugenplaats slechts periodiek mogelijk is. Over het algemeen zal dan gewacht moeten worden zowel voor de uitvoering van de opdrachtselectie, als voor die van de getalselectie. Deze wachttijden kunnen meestal bekort worden, mits we weten, wanneer (d.w.z. in welke volgorde) de geheugenplaatsen bereikbaar zijn.

De 64 kanalen corresponderen met de 64 langs een beschrijvende van de cilindrische trommel opgestelde koppen, die in het eigen kanaal kunnen lezen en schrijven. De 16 woorden van een kanaal zijn langs een omtrek geschreven, maar deze omtrek is "dubbel beschreven", d.w.z. elk woord beslaat 1/8 deel van de omtrek, welk stuk het met een ander woord moet delen (van de 60 binalen, die op 1/8 omtrek staan, specificeren de evene het ene, de onevene het andere woord). In 1 getaltijd = 1/8 omwentelingstijd kan of de ene, of de andere van twee simultaan beschikbare geheugenplaatsen bereikt worden.

In het n^{de} kanaal komen in één omwenteling in onderstaande volgorde de tweetallen onder de lees-koppen:

getaltijden: 0^{de} 1^{ste} 2^{de} 3^{de} 4^{de} 5^{de} 6^{de} 7^{de}
 $16n+0, 16n+3, 16n+6, 16n+1, 16n+4, 16n+7, 16n+2, 16n+5, \dots \text{etc.}$
 $16n+8, 16n+11, 16n+14, 16n+9, 16n+12, 16n+15, 16n+10, 16n+13, \dots \text{etc.}$

Overeenkomstige geheugenplaatsen uit verschillende kanalen komen tegelijkertijd langs de koppen. De laagste cijferplaats komt voorop; de a-opdracht komt voor de b-opdracht. Dwarsdoorsnede van de trommel geeft dit beeld:



De nu volgende tijden gelden, wanneer geen kanaalwisseling optreedt.

Wanneer we de tijdsduur van een opdracht definiëren als de tijd, die verstrijkt vanaf het begin van deszelfs lezing uit het geheugen tot aan het moment dat, de volgende opdracht gelezen kan worden, zien wij, dat een a-opdracht minstens $1/16$, een b-opdracht minstens $5/16$ omwenteling kost (afgezien van besturingsverplaatsingen); dit is gemiddeld $3/16$ omwenteling = 4 msec per opdracht.

Voor vermenigvuldigingen en delingen (16 t/m 21) komen hier voor het ten uitvoer brengen van de bewerkingen 5 omwentelingen bij, dus dus totaal 104 msec.

Voor additieve en administratieve opdrachten (0 t/m 5, 8 t/m 13) komt hier 1 omwenteling bij, dus 24 msec.

In het geval, dat het een b-opdracht betreft, kan op deze operaties 20 msec bespaard worden, indien het getal, dat door de getalselectie wordt uitgezocht, een getaltijd eerder aan bod is, dan de volgende opdracht (bv. $3/64$, staande op 2b).

Schuifoperaties: kosten altijd 1 omwenteling extra, dus 24 msec.

Besturingsverplaatsing. Een niet gehoorzaamde (dus conditionele) besturingsverplaatsing kost op een a-plaats 24 msec, op een b-plaats 4 msec.

Een (gehoorzaamde conditionele) besturingsverplaatsing kost dat deel van een omwenteling, dat men noodzakelijk wachten moet, totdat

bedoelde plaats onder de koppen komt, met dien verstande, dat de snelste sprong van een a-plaats is naar de a-opdracht, die twee getaltijden later aan bod komt, van een b-plaats naar de a-opdracht die 2.5 getaltijd later aan bod is, zodat in het ongunstigste geval het iets meer dan een omwenteling kost. In geval van kanaalwisseling komen hier 1 of 2 omwentelingen bij.

Communicatieopdrachten. De skipopdracht 24/0 en 24/1 kosten geen omwenteling, dus 4 msec, ook op de a-plaats! 24/16 en 24/7 kosten altijd 24 msec. De overige 24-opdrachten kosten 24 msec. op een a-plaats, 4 msec. op een b-plaats.

Schrijvende en lezende opdrachten, in de a-positie, die betrekking hebben op een geheugenplaats, die in dezelfde getaltijd beschikbaar is als de betroffen a-opdracht (en zijn volgende b-opdracht) kosten een omwenteling extra! Immers de a-opdracht is gelezen, pas een omwenteling later kan gelezen of geschreven worden, maar dan is de b-opdracht juist voorbij, en moet weer een omwenteling gewacht worden!

Blokkade bandlees-opdracht.

Als de opdracht 24/4 gehoorzaamd is, wordt deze opdracht (omdat de band opgeschoven moet worden) 60 msec. lang geblokkeerd. Ontmoet de besturing binnen deze periode weer de opdracht 24/4, dan wacht de machine, totdat de blokkade opgeheven is. Om de drie omwentelingen kan juist een pentade gelezen worden.

Blokkade type-opdracht.

Als de opdracht 24/8 gehoorzaamd is, wordt deze opdracht (omdat het hamertje van de schrijfmachine terug moet komen) 100 msec. lang geblokkeerd. Ontmoet de besturing binnen deze periode weer de opdracht 24/8, dan wacht de machine, totdat de blokkade opgeheven is. Om de vijf omwentelingen kan juist een symbool getypt worden. Het is aan de programmeur, om er zorg voor te dragen, dat na "TWNR" niet te snel een type-sigitaal komt.

6. De bediening.

De machine wordt uitsluitend bediend van de bedieningstafel af, waarop zich bevinden de bandlezer, de schrijfmachine, de 2 oscillografen, het schakelaarspaneel en het toetsenpaneel. Als de machine werkt zijn alle toetsen van het toetsenpaneel geblokkeerd, met uitzondering van de stopknop; deze blokkade is electrisch: men kan de toets wel indrukken, het heeft echter geen enkel effect.

De rechterhelft van het toetsenpaneel (het handregister).

Het handregister bevindt zich op de rechterhelft van het toetsenpaneel en bestaat uit 14 toetsen (drukknopjes), de 4 tekentoetsen, voorzien van de etiketten +, -, +. en -. en de 10 cijfertoetsen de opschriften 0 t/m 9 dragend.

Met het handregister kan men - dank zij een onderdeel van het invoerprogramma (zie aldaar) - positieve of negatieve integers (gehele getallen) of breuken in S brengen. Voor integers geldt de restrictie, dat ze in absolute waarde kleiner moeten zijn, dan 2^{29} ; breuken mogen maximaal (maar hoeven niet) 8 cijfers achter de komma hebben.

Men brengt de getallen in, door eerst de tekentoets in te drukken, vervolgens de cijfertoetsen, het betroffende getal "normaal van links naar rechts" lezend. In het geval van breuken moet men nullen onmiddellijk volgend op de komma, wel aanslaan, dit in tegenstelling tot de ponsconventie voor breuken op de band, waar men, zoals zal blijken, na de pentade +. of -. het 10^8 -voud van de breuk "facultatief" ponsst.

Als men een toets van het handregister indrukt, begint de machine op plaats 0a, en stopt enige tijd later. Pas dan mag men de volgende toets indrukken. Heeft men te vroeg, terwijl de machine nog werkt, de volgende toets ingedrukt, dan heeft dit geen effect; men moet de toets eerst losgelaten hebben, en kan dan de toets alsnog indrukken.

De cijfertoetsen van het handregister zijn bovendien onderworpen aan een speciale blokkade, die alleen opgeheven kan worden door het indrukken van een tekentoets, en die weer van kracht wordt, zodra een van de toetsen van de linkerhelft van het toetsenpaneel wordt ingedrukt: m.a.w. het indrukken van cijfertoets moet onmiddellijk voorafgegaan worden door het indrukken van één van de toetsen van het handregister.

Een fout kan hersteld worden door het getal opnieuw te beginnen.

Het schakelaarspaneel.

Het schakelaarspaneel bevat 4 rijen schakelaars, die alle twee standen hebben: schakelaar naar beneden als 0, naar boven als 1, interpreterend kan men hier een binaal getal "inzetten".

Getalschakelaars.

De onderste rij, met onderschrift "GETAL" bestaat uit 30 schakelaars: het getal, dat door deze schakelaars gevormd wordt, wordt in A ingelezen, als de machine communicatie-opdracht 24/7 gehoorzaamt.

De drie andere rijen (met namen: "BEGIN ADRES", "STOP ADRES" en "OPDRACHT") worden slechts gebruikt ter specificatie van operaties, die worden uitgevoerd onder controle van de toetsen van de linkerhelft van het toetsenpaneel, waarvan de beschrijving nu volgt.

De linkerhelft van het toetsenpaneel.

Deze bestaat uit drie rijen.

De onderste rij onder het etiket "GEKOZEN OPDRACHT", bestaat uit drie toetsen, van links naar rechts de namen "HERHAAL", "DOE" en "SCHRIJF" dragend.

Deze knoppen, die slechts gebruikt kunnen worden als de machine niet werkt, hebben betrekking op de opdracht, die in het opdrachtregister staat.

(Als de machine stopt, hetzij door 24/2, door 24/3, door de stopschakelaar of de stopknop (zie onder) dan staat de laatst uitgevoerde opdracht in het opdrachtregister; in het eerste geval door 24/2 of 24/3 zodat men kan zien, waardoor de machine gestopt is).

Drukt men op de toets "HERHAAL", dan gaat de machine de opdracht, die op dat moment in het opdrachtregister staat, bij herhaling uitvoeren, totdat de stopknop uit de bovenste rij wordt ingedrukt.

Drukt men op de toets "DOE", dan voert de machine de opdracht, die in het opdrachtregister staat, één maal uit, en stopt.

Drukt men op de toets "SCHRIJF", dan wordt de opdracht, die door de 15 schakelaars, genaamd "OPDRACHT" op het schakelaarspaneel is gespecificeerd, in het opdrachtregister overgenomen.

De middelste rij bestaat uit drie toetsen links, betrekking hebbende op de rij "BEGIN ADRES" op het schakelaarspaneel, en rechts de stopschakelaar (de enige schakelaar op het toetsenpaneel) betrekking hebbende op de rij "STOP ADRES" op het schakelaarspaneel. Deze twee rijen bestaan elk uit 11 toetsen, waarvan de linkse 10 als binaal getal een adres specificeren, terwijl de laatste schakelaar de a- of b-positie aangeeft: naar beneden, verwijst naar de a-, naar boven, naar de b-opdracht.

Schakelaar "STOP": Als de stopschakelaar "naar de operateur toe" staat, gebeurt er niets; als de stopschakelaar "van de operateur af" staat, stopt de machine, als de opdracht, op het stopadres gespecificeerd, is aangehaald en uitgevoerd.

Toets "BEGIN": De machine gaat werken, te beginnen op het adres, dat op het schakelaarspaneel in de rij "BEGIN ADRES" is aangegeven.

Toets "DOE": De machine handelt als bij "BEGIN", maar stopt na uitvoering van de eerste opdracht.

(De functie van "DOE" is op te bouwen uit "STOP" en "BEGIN", als men de "STOP"-schakelaar in de aan-stand zet en het adres in "BEGIN ADRES" in "STOP ADRES" overneemt).

Toets "LEES": De machine leest de opdracht, op de plaats, door "BEGIN ADRES" aangegeven in het opdrachtregister, in. Dit verschijnt in lampjes op de machine; zo kan gekeken worden, of een opdracht goed in het geheugen staat.

De bovenste rij onder het etiket: "VOLGENDE OPDRACHT" bestaat uit vier toetsen, die eveneens de namen "BEGIN", "DOE", "LEES" en "STOP" dragen.

Drukt men op de "STOP" toets, dan voltooit de machine de opdracht, aan welker uitvoering gewerkt wordt, en stopt vervolgens.

De functie van de toetsen "BEGIN", "DOE" en "LEES" is gelijk aan die van de rij er onder (zie boven) met dien verstande, dat de functie van de rij schakelaars, "BEGIN ADRES" overgenomen is door de opdrachtenteller, die na voltooiing van een opdracht, plaats en positie van de volgende te gehoorzamen opdracht aangeeft.

Als bv. 24/2 in 13 a gehoorzaamd is, stopt de machine; de inhoud van het opdrachtregister is dan 24/2, de laatst gehoorzaamde opdracht; de inhoud van de opdrachtenteller is 13b.

Na lezing wordt de opdrachtenteller ook een (halve) plaats opgehoogd, zodat men door herhaald indrukken van de toets "LEES VOLGENDE OPDRACHT" de opdrachten ter controle achtereenvolgens in de lampjes van het opdrachtregister kan laten verschijnen. De machine doet dan niets, en de "volgende opdracht" betekent: "de opdracht op de volgende plaats in het geheugen" met veronachtzaming van alle bestuursverplaatsingen.

Het is zinloos een toets uit de bovenste rij in te drukken, als niet eerst een toets uit de middelste rij is ingedrukt. (Eventueel uit het handregister, maar dat zal zelden voorkomen).

7. De ponsband.

Door het aanslaan van een toets van de ponsmachine kunnen wij in de band al of niet gaatjes ponsen op vijf plaatsen, die op een rijtje liggen, loodrecht op de richting van de band; welke configuratie van gaatjes en niet-gaatjes geponst wordt, hangt af van de toets, die men aanslaat; een dergelijk rijtje wordt een pentade genoemd. Deze vijf plaatsen corresponderen met de vijf laagste cijferplaatsen in A; bij de bandlezing verschijnt in A een 0 op die cijferplaatsen, die overeenkomen met de plaatsen op de band, waar een gaatje was geponst, terwijl op alle andere plaatsen in A een 1 verschijnt. De absolute waarden van het zo ingelezen getal (die maximaal 31 p is) wordt genoemd de bandwaarde van de betrokken pentade of van de betrokken toets; de bandwaarden corresponderen als volgt met de symbolen op de etiketten der toetsen, de toetssymbolen:
toetssymbolen: X,A,B(-.),C(+.),D(-),E(+),F,0,1,2,.....,23,24
bandwaarden: 0,1,2, , 3 , 4 , 5 ,6,7,8,9,.....,30,31
sluitlettertoetsen cijfertoetsen

Op een adresplaats kan door het invoerprogramma (zie aldaar) een molecuul informatie geborgen worden; een molecuul informatie bestaat uit

- 1) een getal (hetzij als breuk, hetzij als geheel getal opgevat, en met willekeurig teken)
- 2) twee opdrachten, die elk de helft van de geheugenplaats beslaan. Ter onderscheiding noemen we deze de a-opdracht en de b-opdracht: de **a-opdracht** beslaat de 15 laagste cijferplaatsen, de **b-opdracht** beslaat de 14 hoogste cijferplaatsen, plus het tekencijfer. De machine gehoorzaamt normaliter na een a-opdracht de b-opdracht uit dezelfde geheugenplaats, gaat daarna de a-opdracht uit de volgende geheugenplaats gehoorzamen etc.. Dit is ook de volgorde, waarin de opdrachten op de band geponst worden.

De decimaal-binale conversie, gevolgd door de correctie.

Het invoerprogramma verzorgt tijdens het inlezen waar nodig de omzetting van het 10-tallig stelsel in het 2-tallig stelsel op de volgende manier (in feite met gewisseld teken): met behulp van de cijfertoetsen (zie boven) worden de decimale cijfers achtereenvolgens op de band geponst in hun binale representatie (vermeerderd met 7, een vermeerdering, die het invoerprogramma er onmiddellijk weer van aftrekt). Voordat het eerste cijfer van de band gelezen wordt, wordt de inhoud van S gelijk gemaakt aan nul; als een cijfer gelezen is, wordt het, vermeerderd met het 10-voud van [S], in S

geplaatst en de volgende pentade wordt gelezen. Een grootheid op de band (sluitletter), die kleiner is dan 7, dient als indicatie, dat het proces van de decimaal-binale conversie is afgelopen. Het getal, gevormd door de pentades aan de sluitletter voorafgaand, staat dan (in eenheden p) in S. De sluitletter heeft een tweede functie: was de sluitletter X (bandwaarde 0, dus "blank tape"), dan wordt (S) onveranderd gelaten; in het geval van de sluitletters A,B,...,F wordt (S) vermeerderd met resp. (73), (74), ..., (78). Het proces van de decimaal-binale conversie wordt dus steeds beëindigd door eventuele correctie, m.a.w. elke rij decimale cijfers wordt op de band door een sluitletter gevolgd. Boven beschreven proces wordt toegepast bij het invoeren van gehele getallen, breuken, de adressen der opdrachten en het adres van de controle-combinatie (zie onder).

Opm.1. Waar wij spreken over "gehele getallen", beschouwen wij deze, staande in eenheden $p(2^{-29})$; breuken, zoals ze worden ingevoerd, zijn te beschouwen als gehele getallen, geborgen in eenheden 10^{-8} (zie onder, zie invoerprogramma).

Opm.2. Uit het conversieproces volgt, dat het - afgezien van tijdverlies - onschadelijk is, het decimaal getal vooraf te laten gaan door enige pentades, die corresponderen met de 0-toets. Voor het invoeren van het getal nul mag direct de sluitletter X geponst worden; opdrachten, waarin "het adres ontbreekt", d.w.z. waarvan het adres gelijk is aan nul of de inhoud van een van de adressen 73 t/m 78 kunnen zo met twee pentades geponst worden: functieletter, onmiddellijk gevolgd door sluitletter.

Opm.3. Wanneer het getal, gevormd door de hoogste twee decimale cijfers 24 niet overschrijdt, mag dit met een aanslag op de corresponderende cijfertoets worden geponst. Dit bekort de inleestijd met 60 msec. Dit mag alleen aan het begin van het getal!

Opdrachten.

Opdrachten ("halve moleculen") kunnen slechts getweeën worden ingelezen, nl. de a-opdracht, onmiddellijk gevolgd door de bijbehorende b-opdracht. Een a-opdracht wordt op dezelfde manier geponst als een b-opdracht: of een half molecuul als a-opdracht of als b-opdracht wordt geborgen, wordt bepaald door de plaats op de band. Beide halve moleculen bestaan uit twee gedeelten:

1) de functieletter

dit is altijd één pentade, die geponst wordt door die cijfertoets aan te slaan, waarvan het toetssymbool overeenkomt met het nummer van de opdracht. Onmiddellijk hierna wordt geponst

2) het adres

dit wordt geponst in volkomen overeenstemming met wat gezegd is over de decimaal-binale conversie en bestaat dus uit een aantal (mag nul zijn!) cijferpentades, gevolgd door een sluitletterpentade.

Getallen.

Zoals bij opdrachten het halve molecuul, wordt bij getallen het hele molecuul in tweeën geponst; deze delen zijn:

1) de tekenletter

dit is altijd een pentade, die geponst wordt door een van de (dubbelbenoemde) toetsen +, +., -, -. aan te slaan; onmiddellijk hierna wordt geponst

2) numeriek gedeelte

dit wordt eveneens geponst in volkomen overeenstemming, met wat gezegd is over de decimaal-binale conversie en bestaat dus uit cijferpentades, gevolgd door een sluitletter.

Hierbij moet worden opgemerkt:

na + of - wordt het volgende als geheel getal ingelezen en na eventueel correctie pos. resp. neg. geborgen; de absolute waarde mag niet groter zijn dan 536870911;

na +. of -. wordt na afloop van de decimaal-binale conversie en na eventuele correctie het getal gedeeld door 10^8 ; het quotiënt wordt dan pos. resp. neg. geborgen. Dit impliceert, dat we altijd breuken inlezen met 8 cijfers achter de komma, zodat we zondig de breuk aan de lage kant met nullen aanvullen; wordt de komma eerst gevolgd door enige nullen, dan mogen deze nullen tot aan het eerste (hoogste) cijfer, dat niet nul is, bij de ponsing worden overgeslagen.

Controle-combinatie.

Elke band begint met een indicatie, waar wat volgt in het geheugen moet worden geborgen; deze noemt men een controle-combinatie. De controle-combinatie is als molecuul informatie te beschouwen, maar wijkt daarin van een koppel opdrachten en een getal af, dat het niet als zodanig in het geheugen geborgen wordt, maar slechts bepaalt, waar de volgende (echte) moleculen worden opgeborgen. De op de band geponste moleculen worden op achtereenvolgende geheugenplaatsen geborgen, d.w.z. zodra een molecuul in het geheugen is weggeborgen, wordt eerst het adres van de wegbergopdracht in het invoerprogramma, die dit verzorgde, met 1 opgehoogd, voordat aan de lezing van een

nieuw molecuul begonnen wordt. Het adres, waar het eerste molecuul van een rij geborgen moet worden, wordt gespecificeerd door een aan de rij voorafgaande controle-combinatie, die bestaat uit

1) de controle-pentade

deze wordt geponst door toets A aan te slaan en wordt onmiddellijk gevolgd door

2) het adres

van de geheugenplaats waar het eerste molecuul van de nu volgende rij geborgen moet worden. Ten aanzien van de ponsconventies onderscheidt dit adres zich niet van de adressen, zoals ze in de opdrachten voorkomen.

"Blank tape".

De sluitletter X heeft de bandwaarde 0, d.w.z. er wordt niets geponst. Deze pentade komt echter nooit voor als eerste pentade van een molecuulinformatie en daarom kon het invoerprogramma zo ingericht worden, dat als de bandwaarde 0 gevonden wordt op het moment, dat een nieuw molecuul aan bod is, de band een plaats opschuift en het programma opnieuw probeert of nu de eerste pentade van het nieuwe molecuul gekomen is. Dit betekent, dat tussen de moleculen blank-tape genegeerd wordt, anders geformuleerd, het programma negeert blank tape aan het begin van de band en na elke sluitletter, maar niet na de sluitletter van een a-opdracht. Dit is de explicatie, waarom in het bovenstaande af en toe de woorden "onmiddellijk gevolgd door" voorkomen: dit zijn plaatsen, waar een extra blank tape niet is toegestaan. Deze faciliteit is ingelast ten behoeve van de overzichtelijkheid van langere banden, speciaal met het oog op het gebruik van de reproductor.

Een blanco pentade tussen de moleculen wordt geskipt in 60 msec.

8. De beschrijving van het invoerprogramma.

Het invoerprogramma, dat perpetueel in de eerste vijf kanalen van het geheugen aanwezig is, verzorgt het inbrengen van informatie (zowel getallen als opdrachten) hetzij van het toetsenpaneel, hetzij van de ponsband. Hieronder volgt een gedetailleerde beschrijving van het invoerprogramma.

Lezing uit het handregister.

Het handregister, dat gebruikt wordt voor het (incidenteel) invoeren van getallen, bestaat uit 14 toetsen: 10 cijfertoetsen, corresponderend met de cijfers 0 t/m 9, en 4 tekentoetsen, die de opschriften +, -, +. en -. dragen.

De opdracht 24/5 leest een getal (op de 4 laagste plaatsen) in A, afhankelijk van welke toets op dat moment is ingedrukt; komt in de loop van een programma de besturing de opdracht 24/5 tegen, terwijl de machine de laatste maal niet is gestart door het indrukken van een van de toetsen van het handregister, dan wordt -15 gelezen. Bij de toetsen 0 t/m 9 wordt door de opdracht 24/5 [A] door resp. -0 t/m -9 vervangen, bij de toetsen +, -, +. en -. door resp. -10, -11, -12 en -13.

Een getal wordt ingevoerd door na een tekentoets de decimale cijfers achtereenvolgens aan te slaan, d.w.z. beginnend bij het hoogste cijfer. In het geval van breuken moeten eventuele nullen, die direct achter de komma volgen, wel worden aangeslagen. Het aanslaan van een cijfertoets, voordat een tekentoets is aangeslagen, wordt door een schakeling in het toetsenbord genegeerd en heeft geen effect. Als een tekentoets wordt aangeslagen, start de machine, te beginnen bij opdracht 0a; hetzelfde geldt daarna voor de 10 cijfertoetsen, zolang niet aan van de andere startknoppen is gebruikt.

De eerste opdracht 24/5 leest uit (de aangeslagen toets van) het handregister; onmiddellijk wordt er 10 bij opgeteld en het resultaat wordt negatief weggeschreven op 65, opdat het teken getest kan worden. De scheiding in tekentoetsen en cijfertoetsen wordt hier gemaakt: de conditionele besturingsverplaatsing wordt gehoorzaamd, als (65) positief, dus [A] negatief is, (in casu -0, -1, -2 of -3), d.w.z. als een tekentoets is aangeslagen. In dit geval gaat de besturing naar 45a waar [A] met twee wordt vermeerderd door de opdracht 24/514; dit resultaat wordt door 4/66 zonder tekenwisseling weggeschreven op 66, waar dan, als niet gedeeld moet worden (+ en -) een positief getal (+2, resp. +1), als wel gedeeld moet worden (+. en -.)

een negatief getal (-0, resp. -1) staat; in het laatste geval wordt de besturingsverplaatsing 6/47 (opdracht 46a) genegeerd, en de accumulator wordt ten tweeden male met twee vermeerderd; inmiddels is $A = 2$ (+ en +.) of 1 (- en -.). Na aftrekking van 1 door 24/1022 wordt (A) met tekenwisseling in 64 neergeschreven, waar dus in het geval van + en +. een negatief getal (-1), in het geval van - en - . een positief getal (+0) geborgen staat.

Uit de ingelezen "tekenwaarde" is de noodzakelijke informatie: in het vervolg al of niet delen, al of niet teken wisselen, geëxtraheerd en als getallen van diverse tekens geplaatst in 66 en 64. De nu nog volgende bewerkingen zijn voorbereidselen voor de decimaalbinale conversie: 23/0 vervangt (A) door +0, wat in 67 wordt neergeschreven (d.w.z. er is nog geen cijfertoets aangeslagen); de volgende twee opdrachten plaatsen 10 in 70 (omdat in het geval van bruiken het eerstvolgende cijfer door 10 gedeeld moet worden); 24/2 stopt de machine en de eerste cijfertoets kan worden aangeslagen.

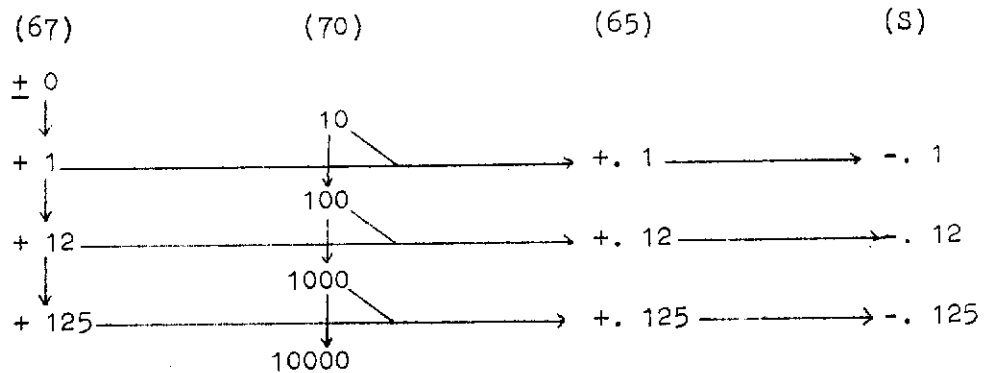
De inhoud van 66 en 64 wordt nu voorlopig niet meer gewijzigd; omdat niet van te voren bekend is, hoeveel cijfers (maximaal 8 in 't geval van een breuk) er worden aangeslagen, moet het programma nu verder zo zijn, dat, als de machine stopt na de verwerking van een cijfer, ergens (nl. in S) het getal staat, voorzover het is aangeslagen.

Drukken we nu een cijfertoets in, dan begint de machine weer op 0a, de besturingsverplaatsing in 1b wordt nu echter genegeerd. -(67) wordt in S gezet, de vermeerdering met 10 wordt weer te niet gedaan, $-10.[67] + [A] = - (10.[67] + (-[A]))$ wordt uitgerekend en met tekenwisseling in 67 en 65 weggeschreven; 65 wordt nu verder onder controle van de tekens van 64 en 66 gemodificeerd, omdat het betroffen cijfer het laatste geweest kan blijken te zijn; is dit niet het geval, dan wordt bij de verwerking van het nieuwe cijfer niet meer (65) maar [67] gebruikt. Dit is, om ophoping van afrondingsfouten te voorkomen, noodzakelijk.

Als een geheel getal gevormd moet worden, is (66) positief, de opdracht 6/10 (in 5b) wordt dan gehoorzaamd; afhankelijk van het teken van (64) wordt in S dan +[65] of -[65] geplaatst, en de machine stopt.

Was echter (66) negatief, dan zou de besturingsverplaatsing in 5b genegeerd zijn: (65) wordt vervangen door het quotiënt, dat ontstaat na afgeronde deling van [65] door [70] en [70] wordt, omdat nog een cijfer komen kan, door zijn 10-voud vervangen. Dan wordt wederom de inhoud van 65 met een van beide tekens in S geplaatst.

In onderstaand schema is ter verduidelijking het invoeren van -. 125 met behulp van het handregister weergegeven:



Lezing van de band.

Het grootste deel van het invoerprogramma verzorgt het lezen van de band, d.w.z. het opbouwen van een molecuul informatie uit een aantal pentades:

Hier zij nogmaals de nadruk gevestigd op het feit, dat een molecuul kan bestaan uit één getal of twee opdrachten (zie de Ponsband).

Om het bandlezen in gang te zetten moet men de machine laten beginnen op 16a: er wordt een pentade gelezen, het zojuist gelezene wordt met 1 vermeerderd en op (70) weggeborgen, wat impliceert, dat de conditionele besturingsverplaatsing 6/16 in 17b slechts dan wordt gehoorzaamd, als de pentade 0 gelezen is. In alle andere gevallen blijft (A) en daarmee (70) negatief. Dit viertal opdrachten negeert blank tape, de besturing verlaat dit cyclusje, als de eerste van nul afwijkende pentade gelezen is, met achterlating van diens bandwaarde (negatief, en dan vermeerderd met 1) in 10. Dan worden S en 72 schoongemaakt (S ten behoeve van de binale opbouw van het decimaal geponste getal, 72 ten behoeve van de sluitletter X, die het adres onveranderd late), waarna de besturing verwezen wordt naar 20b, waar zich de bandleesopdracht van de conversie cyclus bevindt. Elke gelezen pentade wordt met 7 vermeerderd; is het antwoord negatief (A bevat dan juist minus de waarde, door het toetsymbool aangegeven!) dan wordt [A] door de opdracht 24/16 vermeerderd met het 10-voud van [S] in S geplaatst. Is het antwoord echter positief (+7, +6, t/m +1) dan is een van de sluitletters X, A t/m F gelezen en de zg. correctie-opdracht wordt (negatief) in A gevormd door aftrekking van (68) en met tekenwisseling in 24 neergezet. De variabele opdracht 9/72, 9/73...9/78 in 24a (voor sluitletters resp. X, A t/m F) voert in S, waar het decimaal getal negatief is opgebouwd, de correctie uit, dan wordt de negatieve met 1 vermeerderde opening-

letter uit 70 in A geplaatst en nog met 6 vermeerderd. De uitkomst (negatieve openingsletter +7) wordt aan een tekentest onderworpen die slechts gehoorzaamd wordt als de openingsletter de functieletter van een opdracht is.

Laat ons echter eerst nagaan wat geschiedt, als de controle combinatie gelezen is. De bandwaarde van de controlecombinatie is 1, -1 is op 16a ingelezen, -0 is op 70 opgeborgen, met +6 in A wordt de conditionele besturingsverplaatsing 6/32 (in 26a) genegeerd; 24/1018 (in 26b) vermindert [A] met 5, de besturing verspringt ongeconditioneerd naar 29b, waar [A] (nu +1) negatief wordt weggeschreven, zodat de conditionele besturingsverplaatsing 14/50 (in 30a) genegeerd wordt. De inhoud van S (het negatieve adres n van de controle combinatie) wordt in A geschoven, <69> wordt er van afgetrokken, de opdracht 5/66 (in 28b) schrijft <A> met tekenwisseling naar op 66, hier 13/n als a-, 15/27 als b-opdracht plaatsend. De besturing wordt naar 16a verplaatst en een nieuw molecuul kan gelezen worden.

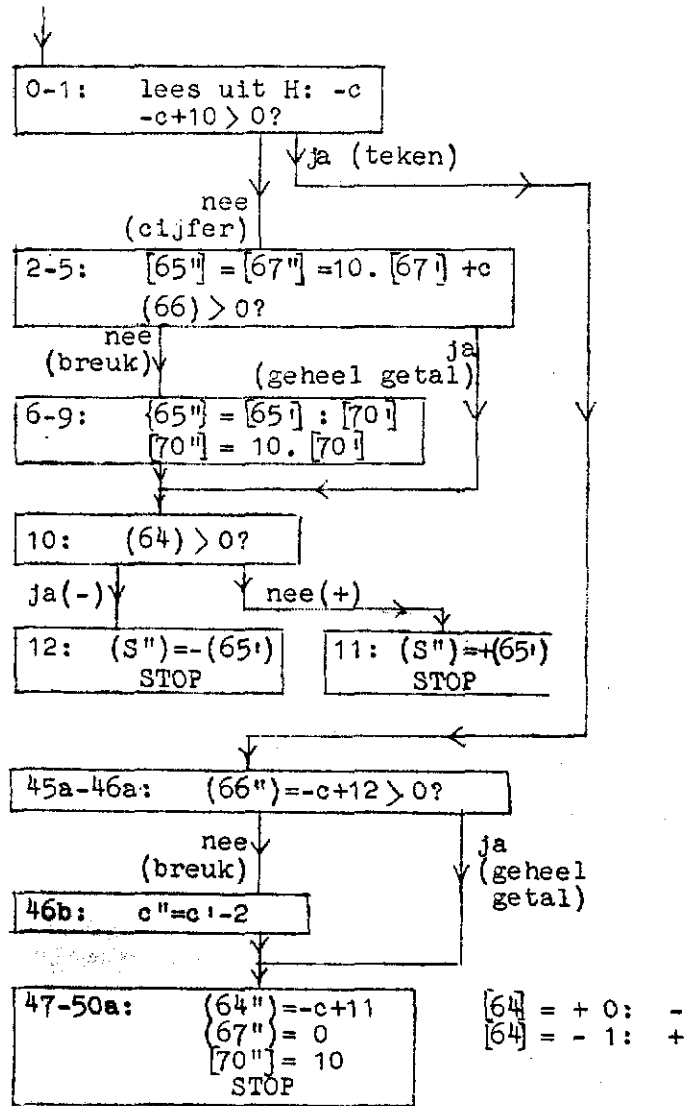
Had echter het molecuul informatie uit een koppel opdrachten bestaan, dan zou de opdracht 6/32 (in 26a) wel gehoorzaamd zijn; op dat moment bevindt zich het gecorrigeerde adres negatief in S, de functieletter negatief in A. De eerste wordt door 12/71 even opgeborgen, omdat anders het adres verloren zou gaan door de schuifopdracht, die de functieletter (steeds negatief) uit A in de a-positie in S schuift; het negatieve adres uit 71 wordt er bij opgeteld in S wordt in 71 neergeschreven; 71 bevat op dat moment de negatieve a-opdracht. Intussen is in A een nieuwe pentade gelezen, waarvan we nu weten, dat het de functieletter van de bijbehorende b-opdracht moet zijn (zie de Ponsband). Met weglating van enkele nu overbodige tekentesten wordt de b-opdracht op dezelfde wijze opgebouwd als in het vorige kanaal de a-opdracht. Na uitvoering van de opdracht 8/72 in 42b staat de b-opdracht negatief, voltooid in de a-positie in S. 23/15 (in 43a) schuift deze ("door het tekencijfer van A heen") in de b-positie van A, terwijl, omdat voor de schuiving S negatief was, de 15 laagste cijfers van A gelijk zijn aan 1. 0/71 telt er de negatieve a-opdracht in de goede positie bij op (in 71 waren de 15 hoogste cijferplaatsen gelijk aan 1), zodat nu in A het uiteindelijk molecuul, negatief (en vermeerderd met -0) staat; 22/29 schuift het voltooide molecuul naar S, in A + 0 achterlatend. De besturing wordt naar 66a verplaatst waar <S> met tekenwisseling ter plaatse in het geheugen wordt geschreven. De besturing wordt door 15/27 (in 66b) naar 27b verplaatst, waar het adres van de 13-opdracht in 66a met één wordt verhoogd, alvorens de besturing naar 16a wordt verplaatst om te beginnen met de lezing van een nieuw molecuul.

Thans rest ons nog slechts, na te gaan wat gebeurt, als de bandwaarde van de openingspentade 2, 3, 5 of 5 (-., +., - of +) is. Evenals bij de controle combinatie wordt de conditionele besturingsverplaatsing 6/32 (in 26a) genegeerd, echter niet 14/50 (in 30a), die de besturing naar 50b verwijst, terwijl op dat moment [A] gelijk is aan -0, -1, -2 of -3, al naar gelang de resp. -., +., - of + als tekenletter geponst was. 24/514 (in 51b) vermeerderd [A] met 2, welke uitkomst negatief geborgen wordt in 61; was de uitkomst negatief (-0, -1 voor -, +), dan springt de besturing meteen naar 55a; in het andere geval ([A] = 2, 1 voor -., +.) wordt (S) vervangen door het quotiënt, dat verkregen wordt door [S] afgerond door 10^8 te delen. Opdrachten 3/61, 24/1021 plaatsen nu in A -0, als -, en -1 als +. de tekenletter was. Was echter de besturing van 51b in 55a aangekomen, dan was -0, als -, -1 als + de tekenletter was geweest. Door optelling van 1 wordt onderscheid tussen deze gevallen gemaakt. Nadat door 23/0 A schoongemaakt is (ten behoeve van de additieve opdracht 24/513 in 27b) wordt al of niet na tekenwisseling de besturing naar 66a verplaatst, waarna de voltooiing van het inlezen verder volledig geschiedt zoals bij het koppel opdracht is beschreven.

Als de controle combinatie n adrespentades heeft, een koppel opdrachten n, resp. m, adrespentades heeft, als een getal n cijferpentades heeft, dan is de duur voor het invoeren:

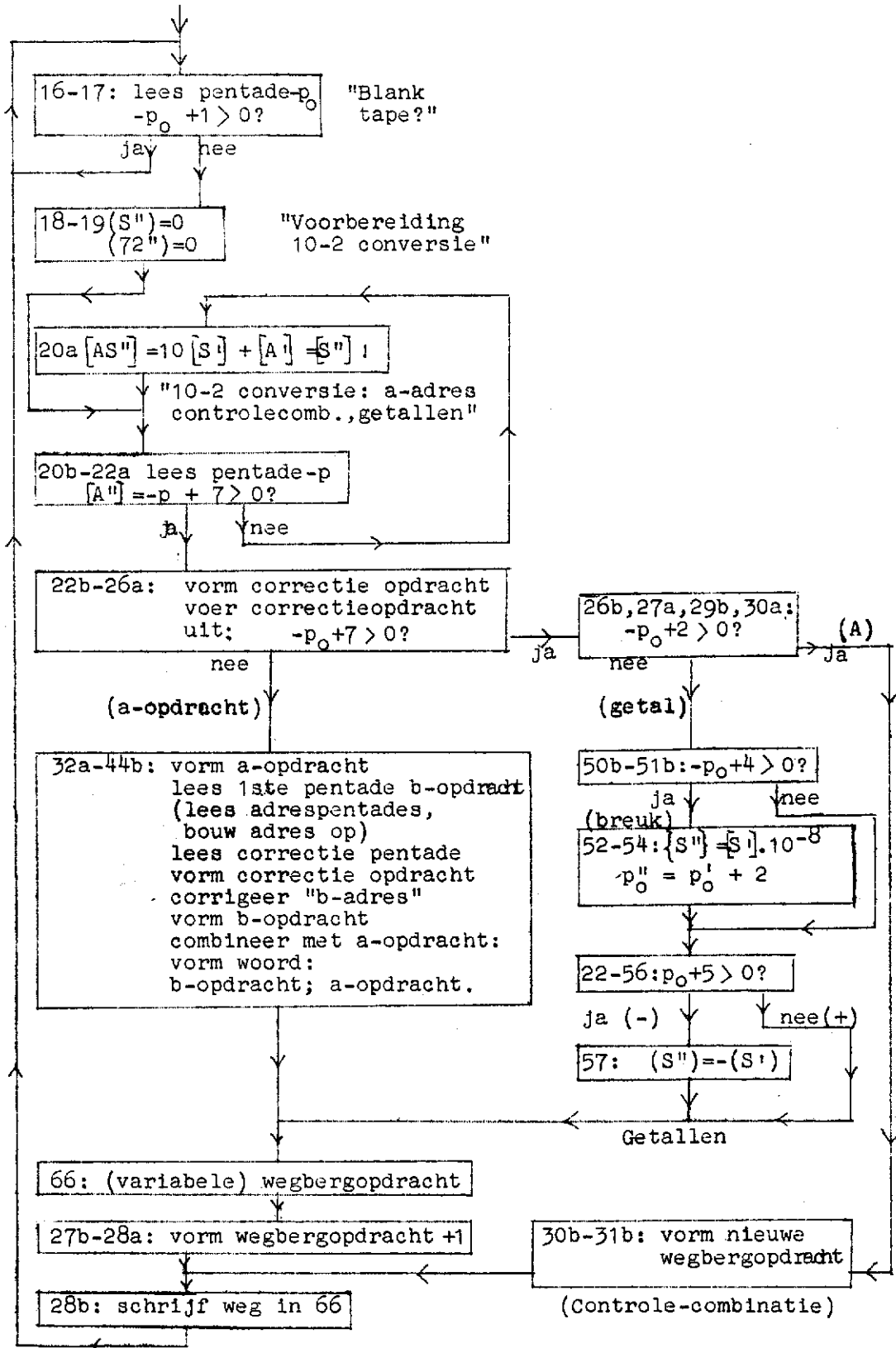
		<u>gemiddeld:</u>
Controle combinatie:	420+60n msec.	$\frac{1}{2}$ sec.
Twee opdrachten:	840 + 60(n+m)msec.	$\frac{1}{2}$ sec.per opdracht
Getal: geheel getal, pos.	460+60n msec.]	kleine sec.per geh.getal
" " ,neg.	480+60n msec.]	
breuk, pos.	600+60n msec.]	ruime sec. per breuk.
" neg.	620+60n msec.]	

Blokschema. (Lezing handregister).



(Enkel accent: woord voor de operatie,
Dubbel accent: woord na de operatie).

Blokschema. (Lezing band).



Het invoerprogramma.

Voor lezing uit het handregister:

Gelezen wordt: -0,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10,-11,-12,-13
 voor de symbolen: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, +., -.

Voor lezing van de band:

Gelezen wordt: -0,-1, -2 , -3 , -4 , -5 , -6,-7,-8,-9,...,-31
 voor de symbolen: X, A,B(-.),C(+.),D(-),E(+), F, 0, 1, 2,...,24.

0	a	24/5	29a,17b	16	a	24/4
	b	24/522			b	24/513
1	a	5/65		17	a	4/70
	b	6/45 →			b	6/16 →
2	a	11/67		18	a	22/0
	b	24/1013			b	13/72
3	a	24/16		19	a	15/20 ⇒
	b	13/65			b	
4	a	13/67	22a ⇒	20	a	24/16
	b	2/66	19a →		b	24/4
5	a	4/66		21	a	24/519
	b	6/10 →			b	5/67
6	a	11/70		22	a	6/20 →
	b	23/1			b	1/68
7	a	3/65		23	a	24/0
	b	20/70			b	5/24
8	a	13/65		24	a	(9/...)
	b	10/70			b	(2/70)
9	a	24/17		25	a	24/518
	b	12/70			b	5/71
5b →	10	a	2/64	26	a	6/32 →
		b	4/64		b	24/1018
	11	a	14/12 →	27	a	15/29 ⇒
		b	10/65	66b ⇒	b	24/1022
	12	a	24/3 - STOP	28	a	1/66
11a ⇒		b	11/65	31b →	b	5/66
62b →	13	a	24/2 + STOP	29	a	7/16 ⇒
		b	24/3 - STOP	27a ⇒	b	5/67
	14	a	8/60	30	a	14/50 →
		b	24/7		b	23/30
	15	a	12/62	31	a	1/69
		b	7/62 ⇒		b	15/28 ⇒

26a \Rightarrow	32	a	12/71		48	a	23/0
		b	22/19			b	4/67
	33	a	8/71		49	a	24/522
		b	24/4			b	4/70
	34	a	12/71		50	a	24/2 + STOP
		b	4/70	30a \Rightarrow		b	24/514
	35	a	22/0		51	a	5/61
		b	15/36 \Rightarrow			b	6/55 \rightarrow
38a \Rightarrow	36	a	24/16		52	a	23/30
35b \rightarrow		b	24/4			b	10/58
	37	a	24/519		53	a	24/0
		b	5/67			b	20/59
	38	a	6/36 \rightarrow		54	a	3/61
		b	1/68			b	24/1021
	39	a	24/0	51b \rightarrow	55	a	24/513
		b	5/40			b	5/61
	40	a	(9/...)		56	a	23/0
		b	(2/70)			b	6/66 \rightarrow
	41	a	12/72		57	a	13/63
		b	24/519			b	10/63
	42	a	22/19		58	a	-49997757 =
		b	8/72			b	6/66 \Rightarrow
	43	a	23/15		59	a	10 ⁸
		b	0/71			b	
	44	a	22/29		60	a	4/0
		b	7/66 \Rightarrow			b	7/13
1b \Rightarrow	45	a	24/514		61	a	werkruimte
		b	4/66			b	
	46	a	6/47 \rightarrow		62	a	werkruimte
		b	24/514			b	
46a \rightarrow	47	a	24/1022		63	a	werkruimte
		b	5/64			b	
	64		werkruimte		67		werkruimte
	65		werkruimte		68	a	9/79
						b	2/70
\Rightarrow	66	a	(13/...)		69	a	13/0
		b	(15/27) \Rightarrow			b	15/27
					70..78		werkruimte

9. De taak van de programmeur.

Thans is beschreven, tot welke verrichtingen de machine in staat is. De opgave, waarvoor de programmeur zich nu geplaatst ziet, luidt: "Ge weet nu, wat de machine kan. Wat kunt gij nu met de machine?"

De specifieke taak van de programmeur is een onderdeel van de voorbereiding, die nodig is, voordat de machine kan gaan rekenen. Volledigheidshalve volgen hier de belangrijkste stadia:

- 1^e. mathematische formulering van het probleem,
- 2^e. mathematische oplossing van het probleem,
- 3^e. keuze of constructie van numerieke processen, die (in het licht van 2^e) tot het gewenste antwoord leiden,
- 4^e. programmering: gedetailleerde opbouw van de onder 3^e genoemde processen uit de elementaire bewerkingen, waartoe de machine direct in staat is,
- 5^e. codering: uitschrijven van het programma in de code der machine, zodat hierna de hand onmiddellijk geponst kan worden.

Bij de numerieke processen, die onder 3^e genoemd worden, is gedacht aan globale instructies als "...en integreer deze functie over dat interval..." of "...en neem hier de wortel uit...". Bij de beslissingen, die onder 3^e genomen worden, speelt het feit, dat uiteindelijk de berekening door een automatische rekenmachine zal worden uitgevoerd, een betrekkelijk ondergeschikte rol. Vanaf 3^e worden de speciale eigenschappen en conventies geldend bij het gebruik van de individuele machine toenemend belangrijk. De programmering is echter **nog wel** zo'n algemeen probleem, dat hierover dingen gezegd kunnen worden, die niet alleen voor de ARRA van toepassing zijn.

De idealen, die men bij het maken van een programma nastreeft, zijn de volgende:

- 1^e. maximale snelheid, vereist om het programma uit te voeren,
- 2^e. minimale geheugenruimte, vereist om het programma op te bergen,
- 3^e. maximale veiligheid,
- 4^e. maximale accuratesse,
- 5^e. maximale souplesse,
- 6^e. maximale overzichtelijkheid.

1^e en 2^e. De eerste twee idealen zijn tot op zekere hoogte al strijdig.

Een essentieel bestanddeel van elk programma zal blijken te zijn een cyclus opdrachten, die een aantal malen doorlopen wordt. (Het invoeren van een opdracht duurt circa 25 maal zo lang als een keer uitvoeren.) De ARRA zou slechts één minuut kunnen werken, als het hele

geheugen volgeschreven zou worden met opdrachten, die slechts een maal gehoorzaamd mogen worden).

Moet een bepaalde operatie, die in een zeker aantal elementaire bewerkingen gesplitst kan worden, een van te voren bepaald aantal malen herhaald worden (dit is een veel voorkomend geval), dan heeft men de keuze uit twee mogelijkheden:

- a) men schrijft alle opdrachten achter elkaar uit,
- b) men schrijft de opdrachten één maal uit, en last een subroutine in, die er voor zorgt, dat de cyclus het juiste aantal malen doorlopen wordt.

Nu kost a) veel geheugenruimte, b) spaart geheugenruimte, het programma echter duurt langer, omdat bovendien naast de eigenlijke processen de telling moet worden uitgevoerd. Dit is de explicatie van de paradox "Het langere programma is sneller".

Het is duidelijk, dat de beslissing in een snelle machine met een klein geheugen anders uit zal vallen dan in een langzame machine met een groot geheugen. Methode b) heeft ten alle tijde dit voordeel, dat het de souplesse ten goede komt: hetzelfde programma kan, door de telroutine met een variabele telconstante binnen te lopen, voor een variabel aantal herhalingen worden gebruikt.

3^e. Het streven naar Veiligheid uit zich in de voorzorgen, dat de machine geen fout antwoord aflevert.

Een van de manieren is, dat men de berekening in stukken verdeelt, en elk onderdeel zovaak uit laat rekenen, totdat twee maal achter elkaar hetzelfde antwoord is gevonden. Tegenover het voordeel, dat deze vorm van controle makkelijk in elk programma is in te lassen, staan de volgende, niet te onderschatten nadelen:

- 1^e. omdat de machine de berekening minstens in duplo uitvoert, ontstaat een groot tijdverlies,
- 2^e. het is niet zo'n scherpe controle op incidentele fouten in de werking van de machine: als de machine een fout maakt, is het helemaal niet zo onwaarschijnlijk, dat bij herhaling precies dezelfde fout weer gemaakt wordt,
- 3^e. het is geen controle, dat er geen fouten in het programma zitten (d.w.z. de machine kan twee maal achtereenvolgens de zelfde onzin produceren).

Een fraaiere methode ligt in de zg. mathematische check, d.w.z. men laat de machine naast de eigenlijke berekening een accessoir resultaat afleveren, waarvan het resultaat bekend is.

Als voorbeelden noem ik:

Substitutie van gevonden wortels in een vergelijking.

Integratie uitvoeren met twee verschillende intervallen, bv. enkel en dubbel.

Bij het inlezen van een rij getallen (tabel) leest men een getal te veel in, dat zo gekozen is, dat de som van alle getallen nul is. Dit laat men de machine controleren.

Bij het uittypen van getallen (overgang van het 2- naar het 10-talig stelsel) wordt elk getypt cijfer weer uit de schrijfmachine terug gelezen (24/12); met behulp hiervan wordt het getal weer binair opgebouwd en met het oorspronkelijke getal vergeleken.

Iets dergelijks is echter niet in alle gevallen elegant mogelijk.

4^e. Het ideaal der accuratesse brengt met zich mee, dat men ervoor waakt, dat niet door onhandig gekozen methodes, nodeloos precisie verloren gaat.

Men moet zich bv. er terdege van bewust zijn of het zin heeft, om af te ronden: zo ja, dan mag men dit niet nalaten, zo nee, dan hoeft men het alleen te doen, als het kan zonder gevoelig tijdverlies.

5^e. Het streven naar souplesse komt zeer duidelijk tot uiting in de bouw der subroutines (zie aldaar) waar een reeks opdrachten eenmaal in het geheugen staat, die een ingewikkelde operatie uitvoert, die in de loop van een programma enige malen, bv. in het begin, in 't midden, en aan 't einde moet geschieden; hetzelfde streven naar soepelheid ziet men in de band van een subroutine: deze band is nl. altijd gelijk, waar ook in het geheugen we de subroutine willen plaatsen. De plaats van de subroutine wordt nl. bepaald door de inhoud van adres 73, het eerste correctie-adres, dat van te voren, door het inlezen van de zg. voorponsing wordt ingevuld. Band en invoerprogramma (zie aldaar) zijn zo ingericht, dat tijdens het inlezen de informatie, die staat in 73, waar nodig op de juiste manier wordt verwerkt.

De souplesse eist verder, dat de programmeur het programma zo inricht, dat, als er een fout in ontdekt wordt (wat haast altijd het geval is), hij niet het hele programma hoeft over te maken.

6^e. De overzichtelijkheid is een vanzelfsprekend gebod.

Bij de codering vallen de laatste beslissingen, o.a. waar men de diverse flarden van het programma in het geheugen wil onderbrengen. Dan moet, in overeenstemming met de ponsconventies de band opge maakt worden.

10. Het gebruik van subroutines.

Onder een subroutine verstaat men een reeks opdrachten, die tezamen een standaard operatie uitvoeren.

Het is immers de taak van de programmeur om een bepaalde berekening op te bouwen uit de opdrachten, zoals die in het voorafgaande beschreven zijn. Het zou echter voor hem nodeloos veel werk zijn, als hij elk programma moest opbouwen uit deze minuscule bouwstenen, omdat elke berekening gesplitst kan worden in grotere onderdelen, waarvan de functie zo algemeen is, dat een dergelijke rij opdrachten ook vast wel in een ander programma voor zullen komen, bijvoorbeeld het programma dat de overgang van het tweetalig naar het tientalig stelsel bewerkstelligt en de cijfers van het tientalig getal typt.

Een dergelijk programma is vervat in alle type-subroutines, waarvan het gebruik beschreven zal worden ter illustratie van de wijze, waarop in het algemeen subroutines gebruikt worden. Stel dat we de inhoud van geheugenplaats x als geheel getal willen uittypen; in de subroutine T_2 bevindt zich een reeks opdrachten, die het getal, dat staat in S , als geheel getal facultatief uittypen. Nu zou men deze reeks opdrachten zonder meer in standaardvorm op een band kunnen ponsen, en iedere keer, dat er een getal getypt moet worden, het programma zover maken, dat het te typen getal in S staat, vervolgens de reeks opdrachten van het standaardprogramma op de band copieren, en dan weer doorgaan met het gewone programma (zg. open subroutine). Dit zou echter impliceren, dat als later in het programma weer een geheel getal getypt moet worden, met ten tweede male de standaardopdrachten inlast: een identieke reeks opdrachten moet dan tweemaal ingelezen worden en staat in duplo in het geheugen, wat wel erg verkwistend is met geheugenruimte.

Gaarne zou men beide keren de besturing dezelfde reeks opdrachten laten doorlopen. Dit echter betekent, dat na afloop van het typen de besturing een variabele besturingsverplaatsing moet ontmoeten, al naar gelang het eerste of het tweede getal getypt is. Deze variabele besturingsverplaatsing noemt men de koppelopdracht (= link).

De feitelijke uitvoering is nu als volgt:

Ergens in het geheugen staat de reeks opdrachten, die de gewenste operatie (in dit geval het typen van een als geheel getal geïnterpreteerde registerinhoud) volbrengen. Elke keer, dat een getal getypt moet worden, zendt men de besturing naar deze reeks opdrachten, met een dubbele informatie: in S het getal, dat getypt moet worden, in A (in de a -positie) de 7- of 15-opdracht, die de besturing na afloop naar de gewenste plaats in het programma terugstuurt. De reeks

wordt nu voorafgegaan door een 4-opdracht, die deze 7- of 15-opdracht op een blanco plaats achter aan de reeks arithmetische opdrachten plaatst. In het algemene geval is het de opdracht $24/6$ (zie communicatieopdrachten), die er zorg voor draagt, dat deze 7- of 15-opdracht in A geplaatst wordt. Als c bv. de eerste geheugenplaats van de subroutine T 2 is, en we willen het getal in x typen, dan kan de aanroep als volgt er uitzien.

We nemen aan, dat we bij een programma, op geheugenplaats d gearriveerd, willen typen.

	d+0	a	10/x	plaatst (x) in S
		b	$24/6$	plaatst $15/d+1$ in de a-positie van A
	d+1	a	$7/c \Rightarrow$	stuurt naar Tg2
Van T 2..	\Rightarrow	b	..	
terug				
of	d-1	a		
		b	$10/x$	plaatst (x) in S
	d+0	a	$24/6$	plaatst $7/d+1$ in de a-positie in A
		b	$7/c \Rightarrow$	
van T 2	\Rightarrow	d+1	a
terug				

In het eerste geval wordt (x) in S geplaatst, en de opdracht $24/6$, die nu in d+0, b staat, plaatst $15/d+1$ in A. In de subroutine wordt deze opdracht onmiddellijk "achteraan gezet", de conversie en het typen gaat beginnen, en als dit klaar is, ontmoet de besturing de bij inloop neergezette opdracht $15/d+1$, die de besturing terug doet komen in $15/d+1$, de opdracht volgend op de opdracht $7/c$, de laatste gehoorzaamde, voordat naar de subroutine verwezen werd.

In het tweede geval is het drietal "een halve geheugenplaats verschoven". De opdrachten zelf zijn, dank zij de faciliteit van de opdracht $24/6$, die, staande op de a-plaats nu $7/d+1$ in A plaatst, precies dezelfde. In tegenstelling tot de opdrachten in de subroutine, zegt men, dat de opdrachten in d en d+1 in het hoofdprogramma staan.

De opdrachten $24/6$, $7/...$ vormen het standaard onderdeel van de aanroep, (de subroutine wordt door deze opdrachten aangeroepen).

$10/x$ behoort bij de praeliminairen, is variabel (kan ook $10/y$ zijn, als men het getal (y) wil typen, en kan desnoods ontbreken, als het te typen getal onmiddellijk in in S gevormd wordt).

$24/6$ plaatst de koppelopdracht in A plaatsindicatie in het hoofdprogramma wordt "meegegeven".
 $7/c$ verwijst naar de subroutine

De subroutine plaatst de koppelopdracht ergens, waar na afloop de besturing hem tegenkomt, en de besturing komt terug in het hoofdprogramma. Zo verschijnen de subroutines functioneel als uitbreiding van de de programmeur ter beschikking staande opdrachten.

Voor een grote groep operaties loont het de moeite, om subroutines eens en voor al te maken. Het is duidelijk, dat men over het algemeen alleen die subroutines in zal lezen, die door een programma inderdaad gebruikt worden.

In het beschreven geval verwerkte de routine de inhoud van S, en gaf het resultaat van de bewerking aan de buitenwereld prijs. Vaker echter zal men het resultaat intern verder willen gebruiken. Dan is de subroutine zo ingericht, dat de besturing met het resultaat in S of A in het hoofdprogramma terugkomt. De elliptische beschrijving van de werking van bv. de tweedemachtswortel-subroutine luidt: "Plaatst $(S)^{\frac{1}{2}}$ in S".

De voordelen van het gebruik van subroutines zijn de volgende:

- 1^e. Het beperkt de kans op fouten in het programma
(Subroutines worden pas officieel in de bibliotheek der subroutines opgenomen, als met de machine geprobeerd is, of ze werkelijk werken).
- 2^e. Het reduceert de tijd, nodig om het programma op te stellen.
- 3^e. Het reduceert de ponstijd (omdat subroutines met behulp van een duplicator van de standaardband automatisch worden geponst).
- 4^e. Het geeft het programma meer overzichtelijkheid (zozeer, dat vaak stukken berekening, specifiek voor een bepaald programma, in de vorm van subroutines worden gemaakt, en vervolgens door een hoofdprogramma "aan elkaar worden gelijmd").
- 5^e. Het kan, en zal haast altijd, belangrijke tijdwinst geven. (Aan het opstellen van de standaardsubroutines kan aanmerkelijk meer zorg worden besteed dan aan programma ad hoc. Omdat in deze machine de b-opdrachten vaak een omwenteling bespaard kan worden, kan met veel passen en meten een belangrijke tijdwinst geboekt worden. Om aan een normaal programma, dat voor één probleem gemaakt wordt, deze zorg te besteden, is oneconomisch, sterker, onbegonnen werk. Om in subroutines schakeltijd te vermijden, zullen ze over het algemeen juist één, of juist anderhalf kanaal vullen).
- 6^e. Het kan, en zal haast altijd, belangrijke ruimtebesparing geven.

De nadelen van het gebruik van subroutines zijn de volgende:

- 1^e. Het verhoogt het aantal conventies: iedere subroutine heeft zijn nauw omschreven werking. Een duidelijke beschrijving in een catalogus van de bibliotheek van subroutines moet hieraan tegemoet komen.
- 2^e. Een foutenbron kan optreden, doordat een programmeur gebruik maakt van een subroutine, zonder precies helder voor ogen te hebben staan, wat er opdracht voor opdracht in deze subroutine gebeurt.
- 3^e. Het kan, maar zal niet vaak, tijdverlies geven.
(Waar heel korte subroutines gebruikt worden en schakeltijd van en naar het hoofdprogramma niet ingehaald kan worden. Toch kan men, om der wille van overzichtelijkheid van een dergelijke subroutine gebruik willen maken).
- 4^e. Het kan ruimteverlies betekenen (vooral wanneer een subroutine maar vanuit één plaats in het hoofdprogramma wordt aangeroepen. Het hele gedoe met de koppelopdracht kan dan vermeden worden).
- 5^e. De machine verricht onherroepelijk extra operaties bij de verzorging van de koppelopdracht, en de aanpassing aan de speciale conventies van de subroutine. (Een subroutine krijgt een getal altijd mee in S, afgezien van waar het getal ontstaan is, vanuit welk register de machine het verder verwerken zal. Een resultaat wordt altijd of in S, of in A afgeleverd, afgezien vanuit welk register het hoofdprogramma het verder verwerken zal).

11. Voorbeelden.

Ter illustratie van het voorafgaande nog enige voorbeelden.

1^{ste} Voorbeeld.

Gevraagd wordt om een programma te maken, dat de som van de getallen in 100, 101, 102, ... t/m 199 in S plaatst.

We kunnen nu 100 optelopdrachten achter elkaar plaatsen. Dit kost echter erg veel ruimte (meer dan drie kanalen); door een cyclusje te maken kunnen we met $\frac{3}{4}$ kanaal volstaan. In dit cyclusje komt ten eerste de optel-opdracht voor, ten tweede wordt er voor gezorgd, dat, voordat deze optel-opdracht weer gehoorzaamd wordt, het adres hiervan is opgehoogd, en ten derde, wordt onderzocht, of het laatste getal is opgeteld. Zo nee, dan wordt de besturing weer naar het begin van de cyclus gestuurd, zo ja, dan verlaat de besturing de cyclus. Door een cyclus te maken is bovendien het aantal getallen, en de plaats waar de reeks getallen staat, soepel te veranderen. We nemen aan, dat het programma alleen maar gebruikt zal worden, om een even aantal getallen op te tellen. Dit stelt ons in staat, om per doorgang van de cyclus twee getallen bij de partiele som te voegen. We nemen aan, dat het onderstaande programma begint op adres c; om schakeltijd tot een minimum te reduceren, is het gewenst, dat adressen c+2, t/m c+11 in hetzelfde kanaal staan.

	c+0	22/0	maakt S schoon
		2/c+3	plaatst eerste optelopdracht:
	1	4/c+5	8/101; 8/100 in c+5
		15/c+4 \Rightarrow	
	2	0/2	constante voor ophoging adres-
		0/2	sen.
	3	8/100	constante, indicierend het be-
		8/101	gin van de reeks getallen
9'a \Rightarrow	4	2/c+5	voorbereiding ophoging
1'b \rightarrow		0/c+2 (w)	adressen
	5	(8/100+21	variabele optelopdrachten
		8/101+21)	
	6	15/c+7 \Rightarrow	
	7		
6'a \Rightarrow		4/c+5 (w)	plaatst "nieuwe" optelopdrach-
			ten

8	1/c+10		
	5/c+11		test of einde bereikt is
9	6/c+4	→	zo nee
	7/c+12	⇒	zo ja
10	8/198		constante indicierend het einde
	8/199		van de reeks getallen
11	werkruimte		

9'b ⇒ 12

Het bovenstaande programma kost 9 seconden; hadden we het "uitgeschreven" (d.w.z. de 100 optelopdrachten achter elkaar geplaatst) dan was de som in 2,5 seconde gevormd. Hadden we het programma zo ingericht, dat per doorgang 1 getal was opgeteld, dan had het programma 14 seconde geduurd.

De eigenlijke telling geschiedt door de opdrachten in 8a, 8b en 9a. In 7b schrijft 4/c+5 de optelopdrachten weg, zoals deze de volgende keer gehoorzaamd gaan worden; hier wordt 8/199, 8/198 van afgetrokken, en dit wordt met tekenwisseling, weggeschreven. De besturingsverplaatsing 6/c+4 in 9a wordt dus gehoorzaamd, zolang het door 1/c+10 gevormde verschil negatief is. Dit is voor het laatst, als door 4/c+5 in 7b 8/99; 8/198 is weggeschreven, dan wordt nl. -0 gevormd. Deze opdrachten worden dus nog gehoorzaamd; daarna wordt 8/201; 8/200 gevormd en weggeschreven, maar na aftrekking blijft er een positief resultaat (0/2:0/2) over, en de besturing verlaat de cyclus, zodat dus (198) en (199) het laatste tweetal is, dat opgeteld wordt.

De met (w) gemerkt opdrachten zijn zodanig, dat bij gehoorzaming een omwenteling bespaard wordt. Bij de gehoorzaming van 8/101-21 in 5b wordt in een van de 8 keer een omwenteling bespaard. De opdracht 15/c+7 in 6a is slechts neergezet om met 4/c+5 een omwenteling te kunnen besparen. Dit suggereert een efficiënter programma als men weet, dat het aantal te sommeren getallen een viervoud is. Onderstaand programma telt de honderd getallen op in 6 seconden, met verlies van een geheugenplaats, en de restrictie, dat het aantal te sommeren getallen een viervoud is.

c+0	22/0		7	24/0 (w) Skip!
	2/c+3		11a →	4/c+5 (w)
1	15/c+7 ⇒		8	0/c+2
				4/c+6 (w)
2	0/2		9	1/c+11
	0/2			5/c+12
3	8/100		10	6/c+4 →
	8/101			7/c+13 ⇒
101a ⇒ 4	2/c+6		11	8/198
	0/c+2 (w)			8/199
5	(8/100+41)		12	werkruimte
	8/101+41)			
6	(8/102+41	101b ⇒	13
	8/103+41)			

Als voorbeeld van bandponning wordt nu de band gegeven, die dit programma invoert, waarbij c, de plaats van het eerste getal (100) en de plaats van het laatste getal (199) in de voorponning worden gespecificeerd.

	A	73	X
(73)(A)	+	C	X
(74)(B)	+	100	X
(75)(C)	+	199	X
	A		A
c+0	22		X
	2	3	A
c+1	15	7	A
	0		X
c+2	0	2	X
	0	2	X
c+3	8		B
	8	1	B
c+4	2	6	A
	0	2	A
	A	7	A
c+7	24		X
	4	5	A
c+8	0	2	A
	4	6	A
c+9	1	11	A
	5	12	A
c+10	6	4	A
	7	13	A
c+11	7	1023	C
	8		C

Voorponning

(c+5) en (c+6) worden door het programma ingevuld!

Zoals men ziet, komen onder de stippellijn c, 100 en 199 niet meer voor.

2^{de} Voorbeeld. Subroutine Sw 1.

Als voorbeeld van een subroutine volgt de subroutine Sw 1 (S = Speciale functies; w = wortels) die de wortel uit een breuk trekt. De berekening geschiedt door iteratie volgens Newton. Als $x = \sqrt{a}$ moet worden uitgerekend is de formule

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right) \quad x = \lim x_n.$$

Voor de ARRA moet deze formule wat anders worden opgeschreven, omdat $x_n + \frac{a}{x_n}$ groter dan 1 kan zijn; daarom schrijven we

$$x_{n+1} = x_n + \frac{1}{2} \left(\frac{a}{x_n} - x_n \right)$$

Wanneer men nu er voor zorgt, dat de beginschatting x_0 groter is dan \sqrt{a} (de eerste deling kan dan worden uitgewerd), dan geldt vanaf dat moment, dat $x_{n+1} < x_n$, dus dat de limiet van de bovenzijde wordt benaderd. We kiezen $x_0 = \frac{1}{2} a + \frac{1}{2}$.

Omdat nu zowel $\frac{a}{x_n}$ als x_n , en daarmee hun verschil, de capaciteit niet kan overschrijden, is deze wijze van berekening geschikt.

Hoever moet men echter doorgaan met itereren? Doorgaans is a niet exact gegeven, maar is slechts bekend een grootheid $a' = a + \delta$, met in het gunstigste geval $|\delta| < \frac{1}{2}$. Men kan dus stoppen, zodra $x_{n+1}^2 - a < \delta/2$. Nu is $2x_n x_{n+1} = x_n^2 + a$, dus

$$(x_n - x_{n+1})^2 = x_{n+1}^2 - a; \text{ men kan dus stoppen, zodra}$$

$$x_n - x_{n+1} < \sqrt{\delta/2} = 2^{-15} \text{ bij de ARRA.}$$

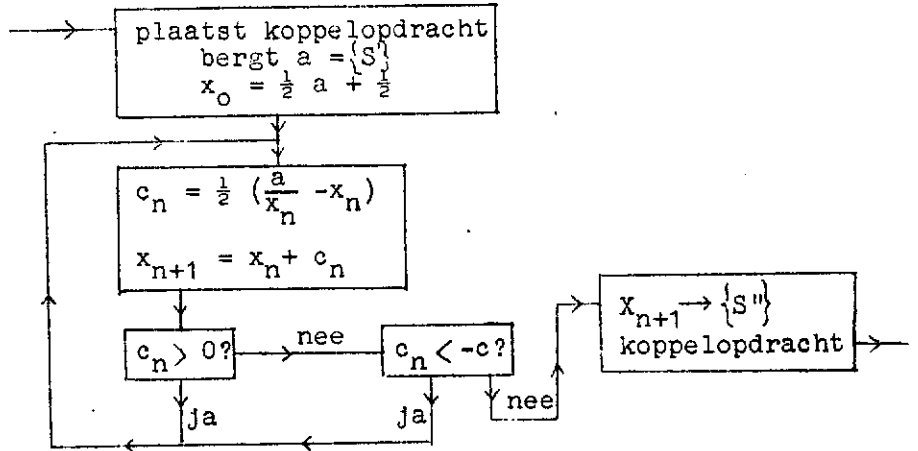
Het kan natuurlijk geen kwaad, een iets kleiner criterium te kiezen: we werken met 14.2^{-19} . De charme van een iteratieve methode is, dat, mocht er een incidenteel rekenfoutje gemaakt worden, er nog geen man overboord is: de berekening duurt slechts iets langer. Wel moet men er zorg voor dragen, dat de test, of het einde bereikt is, niet in de volgende zin slordig is.

We weten, dat $x_n - x_{n+1}$ positief is. We zijn dus klaar, als $x_n - x_{n+1} < 2^{-15}$. Stel, dat door een rekenfoutje een x_{n+1} gevonden wordt, die groter is, dan x_n ! Deze x_{n+1} is beslist fout! $x_n - x_{n+1}$ is echter niet alleen kleiner dan 2^{-15} , het is zelfs negatief! We moeten dus een "veiligheid" er in hebben, dat de laatste correctie niet het verkeerde teken hebben mag. Wel mogen we - sterker, moeten we - zo mogelijk er vrucht van plukken, dat, als alles goed gaat (d.w.z. bijna altijd) $x_n - x_{n+1}$ positief is; deze wetenschap kan tijd-

winst per iteratie geven.

Blokschema (tweedemachtswortel).

Het rekenprocedé voor de worteltrekking wordt door het onderstaande blokschema beschreven.



c_n is de n^{de} correctie, het criterium $c = 14,2^{-19}$.

Om tijdwinst te bereiken, is het bovenstaande blokschema in het programma niet op de voet gevolgd.

Het onderstaande programma rekent uit de wortel uit $\{S\}$ en plaatst het resultaat in S, in formule:

$$\{S''\} = \{S\}^{\frac{1}{2}}$$

(enkel accent slaat op het register bij aanroep, dubbel accent slaat op het register bij terugkomst in het hoofdprogramma).

Hieronder volgen programma, band en de beschrijving, zoals deze staat in de catalogus der subroutines.

Programma.

	☆	c+0	12/c+8	bergt grondtal a
	(w)		4/c+14	plaatst koppelopdracht
		1	23/1	{S} = 1/2 a
	(w)		8/c+15	{S} = 1/2 a + 1/2 = x_0
7'a, 13'b	→	2	12/c+9	bergt iteratie
	(w)		2/c+8	
		3	20/c+9	{S} = a/x_n
	(w)		9/c+9	{S} = a/x_n - x_n
		4	23/1	{S} = 1/2 (a/x_n - x_n)
	(w)		13/c+10	berg-correctie
	(w)	5	24/0	Skip
	(w)		3/c+11	-criterium in A

(w)	6	$6/c+12$	\rightarrow	correctie negatief
		$8/c+9$		rekenfoutje geweest
	7	$7/c+2$	\Rightarrow	doorrekenen
	8	werkruimte (grondtal)		
	9	werkruimte (opeenvolgende iteraties)		
	10	werkruimte (testeinde)		
	11	$14/0$		criterium
		$0/0$		
6'a \Rightarrow	12	$0/c+10$		
(w)		$4/c+10$		
	13	$8/c+9$		
		$6/c+2$	\rightarrow	correctie nog te groot
	14	(koppelopdracht) \Rightarrow		klaar
	15	$\frac{1}{2}$		

Bij de met (w) aangegeven opdrachten, waaronder een skip opdracht, wordt een omwenteling bespaard; $6/c+12$ heeft de (w) indicatie, omdat het een heel gunstige sprong is, die haast altijd wordt gehoorzaamd. Bij $c+0$, a wordt een omwenteling verloren!

De band wordt als volgt geponst:

	A	A	A
c+0	12	8	A
	4	14	A
c+1	23	1	X
	8	15	A
c+2	12	9	A
	2	8	A
c+3	20	9	A
	9	9	A
c+4	23	1	X
	13	10	A
c+5	24		X
	3	11	A
c+6	6	12	A
	8	9	A
c+7	7	2	A
	0		X
	A	11	A

c+8, c+9, c+10 vult het programma in

12. De permanente geheugeninhoud.

x	[x]	{x}	(b) <x>	(a)
0	- 2513 05978	- . 4680 93861	24/522 ;	24/5
1	+ 2028 06337	+ . 3777 56240	6/45 ;	5/65
2	- 2352 30140	- . 4381 50279	24/1013 ;	11/67
3	+ 4383 62128	+ . 8165 13091	13/65 ;	24/16
4	+ 692 84931	+ . 1290 53241	2/66 ;	13/67
5	+ 2016 58434	+ . 3756 18104	6/10 ;	4/66
6	- 3019 45785	- . 5624 17852	23/1 ;	11/70
7	- 4003 56286	- . 7457 21694	20/70 ;	3/65
8	+ 3378 51457	+ . 6292 97377	10/70 ;	13/65
9	+ 4049 71537	+ . 7543 18269	12/70 ;	24/17
10	+ 1363 16992	+ . 2539 10184	4/64 ;	2/64
11	+ 3376 88588	+ . 6289 94010	10/65 ;	14/12
12	+ 3712 53251	+ . 6915 13067	11/65 ;	24/3
13	- 2683 12573	- . 4997 71113	24/3 ;	24/2
14	- 2681 97827	- . 4995 57381	24/7 ;	8/60
15	+ 2369 24990	+ . 4413 07183	7/62 ;	12/62
16	- 2516 00821	- . 4686 43179	24/513 ;	24/4
17	+ 2018 55046	+ . 3759 04322	6/16 ;	4/70
18	+ 4385 89440	+ . 8169 36493	13/72 ;	22/0
19	+ 15382	+ . 2000 28647	0/0 ;	15/20
20	- 2682 79791	- . 4997 10051	24/4 ;	24/16
21	+ 1699 22711	+ . 3166 36099	5/67 ;	24/519
22	+ 357 88820	+ . 0666 61872	1/68 ;	6/20
23	+ 1685 83168	+ . 3140 10620	5/24 ;	24/0
24			(2/70 ;	9/...)
25	+ 1701 23782	+ . 3168 80237	5/71 ;	24/518
26	- 2350 71455	- . 4378 54705	24/1018 ;	6/32
27	- 2349 31170	- . 4375 93404	24/1022 ;	15/29
28	+ 1699 35938	+ . 3165 30351	5/66 ;	1/66
29	+ 1699 74800	+ . 3166 02737	5/67 ;	7/16
30	- 3009 92461	- . 5606 42147	23/30 ;	14/50
31	+ 5042 35077	+ . 9392 11020	15/28 ;	1/69
32	- 3349 09368	- . 6238 17310	22/19 ;	12/71
33	- 2682 96120	- . 4997 40466	24/4 ;	8/71
34	+ 1365 23847	+ . 2542 95481	4/70 ;	12/71
35	+ 5045 18656	+ . 9397 39227	15/36 ;	22/0
36	- 2682 79791	- . 4997 10051	24/4 ;	24/16
37	+ 1699 92711	+ . 3166 36099	5/67 ;	24/519
38	+ 357 88836	+ . 0666 61902	1/68 ;	6/36
39	+ 1691 07456	+ . 3149 87183	5/40 ;	24/0
40			(2/70 ;	9/...)
41	- 2514 16503	- . 4682 99730	24/519 ;	12/72
42	+ 2708 17299	+ . 5044 36528	8/72 ;	22/19
43	+ 23 50095	+ . 0043 77393	0/71 ;	23/15
44	+ 2370 66269	+ . 4415 70336	7/66 ;	22/29
45	+ 1364 05506	+ . 2540 75054	4/66 ;	24/514
46	- 2515 86512	- . 4686 16396	24/514 ;	6/47
47	+ 1698 34910	+ . 3164 53930	5/64 ;	24/1022

x	[x]	{x}	(b) < x >	(a)
48	+ 1364 36736	+. 2541 33224	4/67 ;	23/0
49	+ 1365 36586	+. 2543 19210	4/70 ;	24/522
50	- 2515 68125	-. 4685 82148	24/514 ;	24/2
51	+ 2031 34013	+. 3783 66584	6/55 ;	5/61
52	+ 3374 68446	+. 6285 83964	10/58 ;	23/30
53	- 4006 95295	-. 7463 53147	20/59 ;	24/0
54	- 2349 76194	-. 4367 77268	24/1021 ;	3/61
55	+ 1697 96097	+. 3162 69876	5/61 ;	24/513
56	+ 2035 12832	+. 3790 72189	6/66 ;	23/0
57	+ 3376 22079	+. 6288 70128	10/63 ;	13/63
58	- 499 97757	-. 0931 28080	30/522 ;	6/66
59	+ 1000 00000	+. 1862 64515	2/1003 ;	24/256
60	+ 2353 11104	+. 4383 01086	7/13 ;	4/0
61				
62				
63				
64				
65				
66				
67				
68	+ 694 11919	+. 1292 89774	2/70 ;	9/79
69	+ 5042 14528	+. 9391 72745	15/27 ;	13/0

13. Tabel van machten van 2.

n	2^n	2^{-n}
0	1	1
1	2	.5
2	4	.25
3	8	.125
4	16	.0625
5	32	.03125
6	64	.01562 5
7	128	.00781 25
8	256	.00390 625
9	512	.00195 3125
10	1024	.00097 65625
11	2048	.00048 82812 5
12	4096	.00024 41406 25
13	8192	.00012 20703 125
14	16384	.00006 10351 5625
15	32768	.00003 05157 78125
16	65536	.00001 52587 89062 5
17	1 31072	$10^{-5} \times .76293$ 94531 25
18	2 62144	$10^{-5} \times .38146$ 97265 625
19	5 24288	$10^{-5} \times .19073$ 48632 8125
20	10 48576	$10^{-5} \times .09536$ 74316 40625
21	20 97152	$10^{-5} \times .04768$ 37158 20312 5
22	41 94304	$10^{-5} \times .02384$ 18579 10156 25
23	83 88608	$10^{-5} \times .01192$ 09289 55078 125
24	167 77216	$10^{-5} \times .00596$ 04644 77539 0625
25	335 54432	$10^{-5} \times .00298$ 02322 38768 53125
26	671 08864	$10^{-5} \times .00149$ 01161 19384 76562 5
27	1342 17728	$10^{-5} \times .00074$ 50580 59692 38281 25
28	2684 35456	$10^{-5} \times .00037$ 25290 29846 19140 625
29	5368 70912	$10^{-5} \times .00018$ 62645 14923 09570 3125
30	10737 41824	$10^{-5} \times .00009$ 31322 57461 54785 15625