MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

REKENAFDELING

CR 3

Principes der electronische rekenmachines

Cursus februari 1948

A. van WIJNGAARDEN.

1948

PRINCIPES DER ELECTRONISCHE REKENMACHINES.

Inleiding:

Ontwikkeld zijn in hoofdzaak twee types van rekenmachines voor wiskundige doeleinden, de zg. analogonmachines en de digitale machines. Onder de eerste categorie vallen bijv. de differentiaalanalysator, machines voor het oplossen van simultane algebraische vergelijkingen, integraalvergelijkingen etc. Allerlei physische principes kunnen worden toegepast, ook electronische. Over dit type electronische machines willen wij het niet hebben. De digitale categorie wordt bijv. vertegenwoordigd door de bekende tafelrekenmachines en allerlei types automatische rekenapparatuur als ponskaartmachines, electromechanische machines, relaismachines en ook weer electronische machines. De genoemde volgorde is voornamelijk die van snelheid. Een tafelrekenmachine vermenigvuldigt twee getallen van tien cijfers in ca. 10 seconden, ponskaartmachines en andere electromechanische machines in ongeveer de helft van deze tijd, relaismachines in ca. 1 à 0.1 seconde en een electronische machine in 3 à 0.2 milliseconde, althans met de hedendaagse stand der techniek. Over problemen welke samenhangen met rekenen met zulke snelheden zullen wij het hebben.
Het veld is nieuw. De Eniac is de enige machine welke op dit moment werkt, terwijl een dezer dagen de Selective Sequence Electronic Calculator van de I.B.M. gedemonstreerd zal worden. Een groot aantal projecten is echter een groot eind onderweg en de eerste twee jaren zullen vele interessante onthullingen brengen.

Organen van de machine: Iedere automatische rekenmachine moet noodzakelijk de volgende vijf organen bezitten:

1) Geheugenorgaan: Dit onthoudt alle in een berekening voorkomende getallen en de instructies voor de besturing.

2) Rekenkundig orgaan: Dit verricht de rekenkundige bewerkingen met getallen ter beschikking gesteld door het geheugen op commando van de besturing.

3) Besturingsorgaan: Dit leidt het rekenproces met behulp van de instructies in het geheugen.

4) Invoerorgaan: Dit brengt de begingevens en instructies in de machine.

5) Uitvoerorgaan: Dit haalt de resultaten van de berekening uit de machine.

Alvorens deze organen ieder onder de loupe te nemen, moeten wij enkele algemene kwesties bespreken over de aard van de getallen waarmede de machine omgaat.

Talstelsel.

Welk talstelsel te gebruiken? Practisch valt de keus tussen tientallig en tweetallig. Een electronische schakeling laat twee toestanden toe: "af" of "aan", een buis kan stroom doorlaten of niet, een potentiaal kan "Hoog" of "laag" zijn. De nauwkeurigheid laat niet met betrouwbaarheid onderscheid toe tussen meer dan twee toestanden.
Dan binair stelsel? Niet noodzakelijk, maar in elk geval wel binaire representatie van cijfers noodzakelijk. Wat zijn voordelen van het binaire stelsel. Eenvoud, hetgeen wij in het volgende aan zullen tonen.

Binaire arithmetiek.

Binaire opstelling is al heel eenvoudig. Er zijn slechts vier optellingen te onthouden.

Optellal:	0	0	1	1
Opteller:	0	1	0	1
Som :	0	1	1	0
Overdracht:	0	0	0	0

Voorbeeld

57 = 00111001
45 = 00101101

$\overline{102}$ = $\overline{01100110}$

Binaire aftrekking is direct te herleiden tot optelling van het complement van de aftrekker ten opzichte van een getal, dat juist buiten de capaciteit van de machine valt. Wanneer de machine 8 digita kan hanteren kunnen wij definieren als complement van 45 een getal $45' = 2^3 - 45$. Dit getal vormen wij door iedere 0 in 45 te veranderen in een 1 en iedere 1 in een 0 en vervolgens nog een 1 op te tellen. Dus:

45' = 11010011
57 = 00111001

57 - 45= $\overline{12}$ = 00001100

Hierbij is de overdracht geheel links weggesmokkeld. Deze valt toch buiten de capaciteit van de machine. In werkelijkheid is het vormen van een zuiver complement als hier beschreven een operatie die een optelling vereist (van de laatste 1), hetgeen de besturing enige moeite kost, terwijl de rest van de complementering een eenvoudige, niet arithmetische bewerking is, welke door de besturing gemakkelijk kan geschieden. Derhalve vormt men in de praktijk vaak een pseudocomplement gelijk aan het complement - 1. Hiervan behoeven we dus slechts de cijfers te inverteren. Bijv. 45" = 11010010. De overdracht links, welke in de optelling optreedt gebruiken wij nu om rechts weer in te voeren, zodat wij toch het goede antwoord verkrijgen. Is het resultaat van de aftrekking negatief, dan vindt links geen overdracht plaats en dan verkrijgen wij ook weer het pseudocomplement van het (negatieve) verschil. Een nadeel treedt aan het licht als we een getal van een gelijk getal aftrekken.

Vermenigvuldigen betekent eenvoudig al dan niet optellen van het vermenigvuldigtal bij het reeds gevormd partiele product naarmate het in aanmerking komend cijfer van de vermenigvuldiger een 1 dan wel een 0 is. Let op dat rekenkundig orgaan een verschuivingsmogelijkheid moet hebben.

Voorbeeld:

zie blad 3

```
57 =                    111001
45 =                    101101
                        ------
                        111001
                     111001
                     ------
                     100011101
                     111001
                     ------
                     1611.00101
                    111001
                    ------
57 x 45 =
2565 =            101000000101
```

of afgekort:

```
57 = 00111001
45 = 00101101
     --------
     00111001
     00001110
     --------
     01000111
     00000111
     --------
     01001110
     00000001
     --------
     01001111
```

Delen is een lastig proces in elk talstelsel, behalve in het binaire. Het is n.l. onwenselijk het gedeelte van het arithmetisch orgaan, dat het quotient ontvangt, te voorzien van overdrachtseigenschappen met het oog op de grote complicatie.
Het recept is nu: Trek de deler één keer af van het deeltal en noteer niets in het quotient. Verschuif de deler één plaats naar rechts. Afhankelijk van het feit of de rest positief dan wel negatief is trekken wij nu voortaan af dan tellen wij de deler op bij de rest, noteren een 1 dan wel een 0 in het quotient en verschuiven een plaats. Aan het eind houden wij dan een plaats in het quotient over, waar wij een 1 noteren. Voor het bewijs beschouwen wij twee hulpquotienten d en e, zoé danig, dat d een 1 krijgt voor iedere aftrekking en een 0 voor iedere optelling, terwijl e in die gevallen een 0 resp. een 1 krijgt. Dan is het ware quotient $Q = d - e = d - e' = 2d + 1$.

Gecodeerd decimale arithmetiek.

Wensen wij in het tientallig stelsel te werken, dan is het practisch, iedere decimale digit te coderen als een binair getal van minstens vier cijfers.
In de eerste plaats kunnen wij zonder meer voor elk tientallig cijfer zijn binaire vorm gebruiken. Dat kost een tetrade van binaire cijfers. Binnen deze tetrade is de arithmetiek zuiver binair, zolang geen tientallenoverdracht plaats vindt. Indien dit wel het geval is, moeten wij corrigeren door 6 bij te tellen. De besturing moet een tamelijk ingewikkelde opgave oplossen bij tientallenoverdracht. Complementeren van een cijfer t.o.v. 9 is moeilijk. Verder is niet eenvoudig te zien of een getal al dan niet kleiner is dan 5 (afronding!).
Een verbetering is de extra-drie-codering. Hierin geven wij niet de binaire voorstelling van het cijfer zelf, maar van het cijfer plus drie. Bij optelling blijft de binaire arithmetiek binnen de tetrade gehandhaafd afgezien van de correctie -3, zolang geen tientallenoverdracht plaats heeft, dan wel een correctie +3, indien die overdracht wel plaats vindt. Het is overigens weer niet eenvoudig te zien of overdracht plaats moet vinden.
Complementeren van een cijfer t.o.v. 9 is heel eenvoudig, want we behoeven slechts te inverteren.

Tenslotte is het heel gemakkelijk te zien of een cijfer al dan niet kleiner is dan 5. Als merkwaardigheid wijzen wij er verder op dan het cijfer 000 niet voorkomt! Dit is een belangrijk feit, omdat hier een zekere controle op storingen ligt. Een ander stelsel is de zgn. 2' - 4 - 2 - 1 code. Dit is binaire codering behalve dat de linkse kolom welke achttal zou voorstellen, weer een tweetal voorstelt. Hiermee kan men ook juist de cijfers van 0 tot 9 voorstellen. Er dreigt in de eerste plaats een zekere dubbelzinnigheid van representatie. Deze kan vermeden worden door te eisen, dat de 2' -kolom niet gebruikt mag worden voor de cijfers onder de 5 en gebruikt moet worden voor de cijfers 5 en hoger. Dan is direct ook te zien of een getal al dan niet kleiner is dan 5. De arithmetiek is binair binnen de tetrade maar in een aantal gevallen moet gecorrigeerd worden met +6, in een ander aantal gevallen met -6, terwijl tenslotte in weer andere gevallen niet moet worden gecorrigeerd. Complementering van een cijfer t.o.v. 9 betekent alleen weer invertering.

Wanneer wij meer binaire cijfers gebruiken is het mogelijk codeeringen te gebruiken, waarin aan zeker eisen voldaan is, bijv. dat altijd twee en slechts twee maal het cijfer 1 gebruikt wordt in de representatie van een cijfer, hetgeen een waardevolle controle op de juiste werking kan zijn. Uit arithmetisch oogpunt zijn dergelijke coderingen echter inferieur en gekunsteld.

<u>Conversie en deconversie</u>.

Een belangrijk vraagstuk is dat van conversie van een (gecodeerd) tientallig getal tot een binair getal en de deconversie in omgekeerde zin.

Vindt de conversie plaats in een speciale conversiemachine (bijv. een relaismachine) welke in het tientallig stelsel met 2 kan vermenigvuldigen, dan is de conversie uiterst eenvoudig. Denken we het getal positief en kleiner dan 1, dan behoeven wij slechts gedurig met 2 te vermenigvuldigen en een 1 dan wel een 0 te noteren al naar mate als resultaat van die vermenigvuldiging een 1 voor de komma verschijnt of niet. Wat voor de komma verschijnt wordt bij de volgende vermenigvuldiging dan weggelaten. Maar met een machine die zelf in het binaire stelsel werkt, gaat dit niet en wordt de opgave interessanter en moeilijker. Wij zullen aannemen, dat de machine een vaste komma heeft voor alle cijfers, zodat elk getal in de machine onder 1 ligt. Stel het getal a geschreven in de zuiver binair gecodeerd decimale vorm:

$$a' = \overline{0,\ a'_1\ ,\ a'_2\ ,\ a'_3\ \ldots a'_n\ ,}$$

waarbij de illeerstreep er op duidt, dat wij deze cijfercombinatie gewoon als een getal moeten lezen en niet als een product van factoren. De cijfers a'_1 zijn dus tetraden van 4 binaire cijfers, welke tetrade in het binaire stelsel het decimale cijfer a'_1 precies weergeeft. Blijkbaar is

$$a' = 10^{-n}\left\{\left[\ (a'_1 .10 + a'_2).\ 10 + a'_3\right] .10 + \ldots \right\} .10 + a'_n .$$

Wij voeren nu in de rechterzijde van het register van de machine a'_1 in, dus vier binaire cijfers en vermenigvuldigen met tien, hetgeen niet een als zodanig begrijpelijke opdracht voor de machine is; immers deze kent geen getallen boven de 1. Maar vermenigvuldigen met tien, dwz. in binaire vorm met 1010 kan worden geinterpreteerd als eenmaal naar links verschuiven, tijdelijk in het geheugen opbergen van dit deelresultaat, weer twemaal naar links verschuiven en optellen bij het achtvoud en het tweevoud, wat een proces is, dat voor de besturing begrijpbaar is. Bij de verkregen $a'_1 .10$ tellen wij a'_2 op etc. Tenslotte moet de straal nog in orde gebracht worden. Als in de machine p binaire plaatsen achter de komma zijn (p binalen), dan hebben we de cijfers a'_1 vermenigvuldigd met 2^{-p} door ze rechts

in het register in te voeren. We moeten dus nog vermenigvuldigen met $2^p \times 10^{-n}$. Maar dit is groter dan 1, want voor ieder ingevoerd decimaal cijfer hebben we vier binaire cijfers gekregen. Dus p moet op zijn gelijk zijn aan 4n. Aangezien $2^4 > 10$ is $2^p \times 10^{-n} > 1$ ⌐ Merk op dat dit het enige getal is, dat het geheugen moet onthouden. Het is overigens een voor alle conversies gelijke constante.

Voor de deconversie moeten we gedurig met tien vermenigvuldigen en tetraden noteren, welke voor de kromme verschijnen. De moeilijkheid is natuurlijk weer, hoe deze vermenigvuldiging uit te voeren zonder buiten het bereik van de machine te geraken. Dit kan, als we bedenken, dat tien = $2^4(2^{-1} + 2^{-3})$. Eerst verrichten we dus de binnen het bereik liggende operatie $(2^{-1} + 2^{-3})x$. Vervolgens schuiven we vier plaatsen naar links. Daarbij stroomt het register over! Maar de overstromende digite behoeven geen rekenkundige bewerking meer te ondergaan en kunnen dus opgevangen worden in een register, dat slechts schuiven toelaat. In dit register verschijnt dus geaandeweg de zuiver binair gecodeerd decimale vorm van het getal.

⌐ = Dit bezwaar omzeilen we door te delen door $10^4 \times 2^{-p}$, wat kleiner is dan 1.

De keuze van het talstelsel.

Te overwegen valt:

a) Het binaire stelsel vereenvoudigt aanzienlijk de benodigde apparatuur van het rekenkundig orgaan en van de besturing.

b) Het binaire stelsel is zuiniger in het gebruik van capaciteit. Immers het aantal binaire cijfers van een getal staat tot het aantal decimale cijfers als $\log 10 : \log 2$. Maar voor elke decimale digit te coderen hebben we 4 binaire cijfers nodig, dus tenslotte vereist een getal geschreven in binair gecodeerde decimale cijfers 20 % meer plaats dan wanneer het in zijn binaire vorm wordt geschreven. Bij gebruik van het octale stelsel is de lengte gelijk en behoeven geen nieuwe cijfertekens te worden ingevoerd, terwijl een getal in octale vorm geschreven nauwelijks meer ruimte vereist dan zijn decimale vorm.

Om alle getallen met een precisie van 10^{10} voor te kunnen stellen hebben wij binair 33 cijfers nodig en gecodeerd decimal 40. Dit heeft een merkwaardige consequentie. Stel, dat de machine 40 binaire plaatsen beschikbaar heeft voor elk getal. Dit betekent, dat een decimaal getal van de buitenwereld af ingevoerd kan worden met een teken en een precisie van 10^9. (samen tien tetrades). Na conversie behoeven we echter voor het teken nog maar één plaats en de overblijvende 39 plaatsen geven in het binaire stelsel een precisie van $10^{11.7}$. Wij kunnen dus de grotere precisie van het binaire stelsel toch niet uitbuiten. Maar de extra cijfers in de binaire vorm kunnen gebruikt worden als waakbinalen om het oplopen van afrondingspunten te ontgaan.

c) Het binaire stelsel vereist conversies en deconversies. In een grote wetenschappelijke berekening is dit onbelangrijk, omdat het allergrootste gedeelte van de getallen, welke in het rekenproces voorkomen nooit het daglicht behoeft te zien. In een machine welke veel gegevens aan eenvoudige bewerkingen moet onderwerpen is het een inconvenient. De duur van een conversie en een deconversie varieert al naar het type machine van 100 tot 5 msec. Het inbrengen van een getal in de machine kost minimaal een aantal milliseconden, zodat het invoerproces door de conversie niet belangrijk wordt vertraagd. In een goed georganiseerde machine vereisen de instructies tot conversie en deconversie ongeveer een procent van de geheugencapaciteit, dus dat is niet buitensporig. Bij gebruik van het octale stelsel in plaats van het decimale, wordt alles nog veel gunstiger.

d) Het binaire stelsel heeft het nadeel, dat de operateur, welke een fout moet opsporen, en dit doet door de machine stap voor stap te laten werken en de bewerkingen te controleren, op de hoogte moet zijn van binaire arithmetiek. Sommigen achten dit een bezwaar met het argument, dat deze rekenwijze moeilijk aan te leren zou zijn, voor minder geschoolde krachten. Een goede machine moet echter zo weinig fouten maken, dat eventueel een meer geschoolde kracht in moeilijke situaties te hulp kan worden geroepen.

Als samenvatting concluderen wij dat het gebruik van het tweetallig stelsel binnen de machine aanbevelenswaardig is voor machines voor wetenschappelijk werk. Voor commerciele machines en machines voor statistiekwerk is waarschijnlijk het tientallig stelsel te prefereren. Bij gebruik van het tweetallig stelsel in de machine is wellicht het achttallig stelsel buiten de machine te propaganderen. In het volgende denken wij altijd aan de machine, welke werkt in het tweetallig stelsel, voorzover niet anders vermeld.

Vaste of drijvende komma.

Een ander principieel punt is de ligging van de komma. In het stelsel van de vaste komma definieren wij de komma te liggen direct na het eerste cijfer, welke als tekenindicatie fungeert, dus elk getal x in de machine voldoet aan $-1 < x < 1$ als het eerste cijfer eenvoudig het teken voorstelt en aan $-1 \leqq x \leqq 1$ als wij bovendien complementen gebruiken voor negatieve getallen. In het laatste geval dat sterk aan te bevelen is, rekenen wij eigenlijk modulo 2. Bij de methode van de drijvende komma is ook weer de komma na het eerste cijfer te denken, maar dit getal moet dan nog vermenigvuldigd gedacht met twee tot een zekere macht, welke exponent afzonderlijk achter het getal gegeven wordt.

Het gebruik van de vaste komma vereist zorgvuldige analyse van het vraagstuk voor het in de machine komt. Immers van elk getal moet gegarandeerd worden, dat de modulus kleiner is dan 1. Dit lijkt een ernstige complicatie, maar een zodanige analyse moet toch, zij het minder rigoreus plaats vinden, om de garantie te geven, dat de gebruikte rekenmethode voldoende stabiliteit bezit. (denk bijv. aan het oplossen van een groot aantal Algebraische vergelijkingen met vele onbekenden!). Overigens kan men instructies geven aan de machine, om in geval van "overstroming" zelf correcties van de schaal aan te brengen.
Bij gebruik van de drijvende komma is het bestaan van de wiskundige zorgelozer, daarentegen dat van de machine veel zwaarder.
Bij iedere optelling moeten de exponenten vergeleken en de nodige verschuivingen toegepast worden. Voorts kost het een aanzienlijke extra capaciteit van het geheugen om de exponenten te onthouden. Dit is misschien niet zo erg als het lijkt, omdat wij wellicht met minder cijfers behoeven te werken. In het volgende wordt, zo niet anders vermeld, gedacht aan machines met vaste komma.

Algemene strategie.

Van groot belang is de kwestie hoeveel bewerkingen parallel in de machine kunnen geschieden. Hierbij dient onderscheid gemaakt te worden tussen parallelie in het groot en in het klein.

Onder parallelie in het groot verstaan wij de mogelijkheid om meerdere rekenkundige bewerkingen op verschillende getallen tegelijk uit te voeren. Daarvoor zijn natuurlijk meerdere rekenkundige organen nodig, bijv. 4 optellers of 2 vermenigvuldigers. Het heeft al direct het voordeel van versnelling van de berekening. Daarenboven heeft een zodanig veelvoudig rekenorgaan een niet verwaarloosbare geheugencapaciteit. Bij de Eniac bestaat practisch de gehele snelgeheugencapaciteit uit die van zijn optelmechanismen. Het verdelen van het rekenprogramma over deze organen is echter geen sinecure indien wij werkelijk het geheel efficient willen laten werken. Het meest doeltreffende middel om de Eniac tot een meer hanteerbare machine te maken is dan ook dat van von Neumann, waarin 19 van de 20 optellers eenvoudig gedegradeerd worden tot geheugen. Grote ingenieusiteit is in sommige machines ten toon gespreid (bijv. in de Stibits-relaismachine) om de machine naar willekeur te splitsen in enkele onafhankelijke machines of wel als een geheel te laten werken.

Ook de differentiaalanalysator berust op dit principe. Binnen zekere grenzen kan men met een dergelijke parallelie in het groot een wat langzamer geheugen combineren dan zonder dat het geval zou zijn, zonder de snelheid van de machine als een geheel te schaden. Het is echter in het algemeen slechts in beperkte mate mogelijk onderdelen van een berekening te parallelliseren. In het volgende wordt, zo niet anders vermeld, uitsluitend gedacht aan machines zonder parallelie in het groot. De klasse machines, welke dan overblijft willen wij aanduiden als enkelwerkend. Zo'n enkelwerkende machine behoeft dus slechts een enkel arithmetisch orgaan te bezitten. Wij krijgen nu een onderscheid tussen parallelie en seriewerking in het klein. Wij kunnen het rekenkundig orgaan nl. zo inrichten, dat het de twee getallen waarmee een bewerking moet worden uitgevoerd in zijn geheel in zich opneemt en aan de bewerking onderwerpt dan wel cijfer voor cijfer onder handen neemt. In het laaste geval is de apparatuur benodigd voor het eigenlijke rekenkundige orgaan minimaal. Het behoeft dan slechts met die cijfers rekening te houden, nl. met een cijfer uit elk getal en nog een eventuele overdracht van een vorige bewerking. Wij lichten het toe aan de optelling van twee getallen bijv. 011011 en 011010. Wij starten de optelling door het laatste cijfer van elk getal in het rekenkundig orgaan in te voeren. Dit telt de 1 bij de 0 op en levert de som nl. 1 af aan het resultaatregister. Vervolgens worden de 2 cijfers van rechts af ingevoerd, nl. 1 en 1 en het rekenkundig orgaan levert de som nl. 0 af en houdt de overdracht 1 in petto voor de optelling van de volgende twee cijfers. Deze zijn 0 en 0. Tezamen met de overdracht 1 geeft dit cijfer een 1 welke afgeleverd wordt. Dan komt weer een 1 en een 1, waardoor een 0 afgeleverd wordt en een overdracht 1 in petto gehouden wordt. Vervolgens komen weer een 1 en een 1, welke tezamen met de overdrachts 1 een 1 afleveren en weer een overdracht 1 in petto veroorzaken. Deze laatste veroorzaakt tenslotte met de 0 en 0 een 1 en de afgeleverde som is 110101. De tijd nodig voor zulk een optelling is blijkbaar gelijk aan het aantal cijfers maal de tijdspanne waarin wij de nieuwe cijfers aan kunnen voeren (van de orde van 1 mikroseconde).

Bij een parallelwerkende apparatuur neemt het rekenkundig orgaan beide getallen in zijn geheel in zich op.

```
                                        011011
                                        011010

en formeert een somkolom voor kolom     000001
tevens de overdrachten noterend          1 1 1

Deze telt het weer verschoven op        110101
```

De tijd benodigd voor een deelbewerking (over de gehele linie natuurlijk) is van de orde van 100 nanoseconden. Het schijnt dus, dat deze methode onvergelijkelijk veel vlugger is dan de eerste. Maar de tweede optelling kan weer overdrachten verwekken, enz., en in het meest ongunstige geval (nl. wanneer we een enkele 1 optellen bij een hele lijn van 1's) is het aantal deelprocessen juist weer gelijk aan het aantal cijfers. Wij komen hierop terug bij de behandeling van het rekenkundig orgaan, maar onthouden voorlopig, dat ten koste van een uitgebreider rekenkundig orgaan een grote snelheidwinst geboekt kan worden door parallelwerking ten opzichte van seriewerking. Er blijkt nu echter, dat de grotere complexiteit van het rekenkundig orgaan grotendeels gecompenseerd wordt door een aanzienlijke vereenvoudiging van de besturing. Het heeft er alle schijn van, dat parallel in het klein alleszins te prefereren is boven serie in het klein. Er is nog een combinatie mogelijk, omdat wij in het tijdsinterval van de serieinbrenging een heel pakket cijfers kunnen afwerken in parallel. Zo is het wellicht mogelijk om een getal van 40 cijfers te behandelen als vier in serie behandelde pakketten van 10 in parallel behandelde cijfers. Wij komen hier nog op terug.
Wij hebben nu genoeg voorbereiding om tot de behandeling van de organen over te gaan.

HET GEHEUGEN (G).

Het geheugen van de machine is een buitengewoon belangrijk orgaan. Niet in die zin, dat het meer onmisbaar zou zijn dan één van de andere vier organen, want ze zijn alle onmisbaar, maar door de specifieke physische zijde van het geheugen, zoals wij dit zullen leren kennen, is de vorm ervan in hoge mate maatgevend voor de vorm van de andere organen en de algemene strategie. Zo zijn er vormen welke bij uitstek geschikt zijn voor serie-, andere voor parallel-werking.
Het geheugen komt in vier gedeelten in de machine voor:

1e Het supersnelle geheugen (G1). Dit is een versnipperd orgaan van kleine capaciteit, hetwelk verspreid voorkomt in de registers van het rekenkundig orgaan en in de onderdelen van de besturing. De gegevens welke het bevat zijn vele malen tijdens een enkele rekenkundige bewerking nodig en moeten derhalve met het technisch mogelijk minimum van vertraging ter beschikking staan en gewijzigd kunnen worden, bijv. in 100 nanoseconden. Dit is geen bezwaar omdat de capaciteit gering is.

2e Het snelle geheugen (G2). Dit is het grote centrale geheugenorgaan met aanzienlijke capaciteit. De gegevens moeten slechts één maal per rekenkundige bewerking gebruikt worden, dus zonder aan de algemene snelheid van de machine afte doen kan het aanzienlijk trager zijn, bijv. 100 mikroseconden. Het is technisch echter buitengewoon moeilijk redelijk capaciteit met deze snelheid te verbinden.

3e Het matig snelle geheugen (G3). Dit is een orgaan met immense capaciteit. De gegevens ervan zijn niet normaal tijdens het rekenproces nodig, maar nu en dan worden grote blokken getallen (bijv. een paar duizend) van of naar het snelle geheugen getransporteerd. Per bewerking is dan de verlibstijd niet belangrijk en de snelheid kan dus weer een orde minder zijn, bijv. 10 milliseconden. Het is technisch zeer wel uitvoerbaar.

4e Het trage geheugen (G4). Dit tenslotte is het orgaan, waar de operateur zijn gegevens op vermeldt en zijn resultaten op ontvangt. De snelheid is beperkt door menselijke en mechanische snelheid, bijv. 1 seconde. Het is technische standaard-apparatuur.

Het supersnelle geheugen (G1).

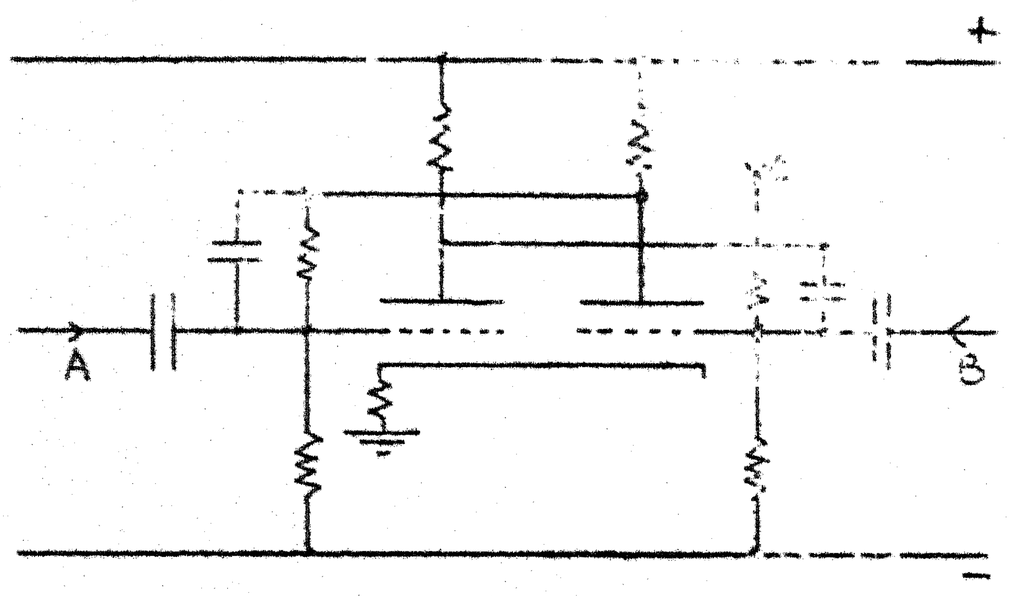
Hiervoor is een zeer bruikbare apparatuur bekend, nl. het zg. trigger-circuit, speciaal bekend als flip-flop.

Er zijn twee uitvoeringen voor ons van belang. In de eerste plaats de "toggle". Dit is een element, hetwelk een "1" of een "0" kan onthouden (en daarvan te allen tijde mededeling van kan doen) en dat door een geschikt signaal naar willekeur één van beide cijfers opgedrongen kan krijgen. In de tweede plaats de "binaire teller" of kortweg "teller", wat slechts op één signaal reageert en wel door van gedachten te veranderen en in plaats van het cijfer, dat het in gedachten had, het andere cijfer te gaan onthouden. (Dit is binair optellen van een 1). Wij gaan nu de physische realisatie hiervan na.

De toggle

De klassieke toggle, massaal toegepast in de Eniac is weergegeven in fig. 1. Deze is van het zg. symmetrische type. Zij bestaat uit een dubbeltriode, waarvan de kathode op vaste potentiaal wordt gehouden. Een dergelijk circuit kan in twee stabiele toestanden verkeren, representerende een "0" of "1".

fig. 1

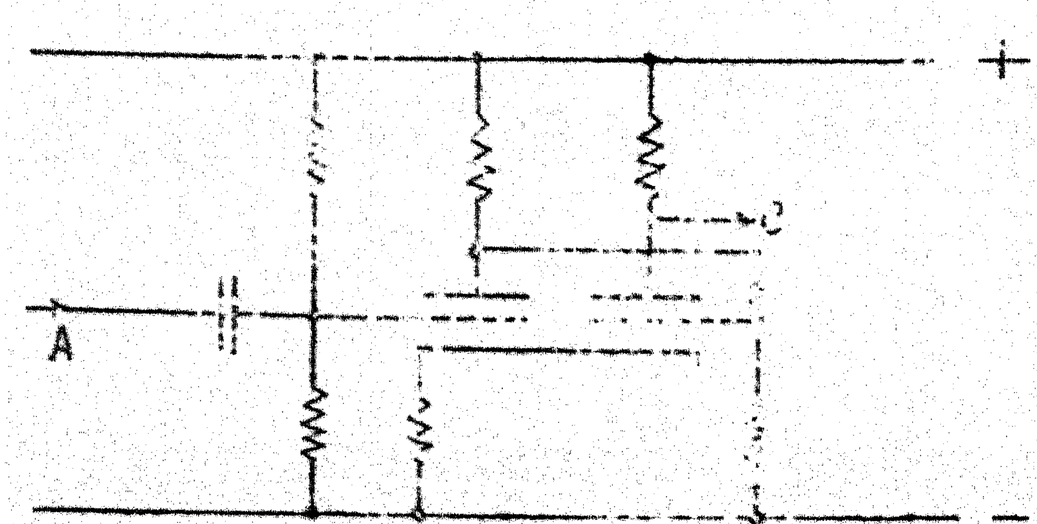


Als de linkerhelft van de buis stroom draagt is de potentiaal van de linkerplaat laag en door het gelijkstroomdwarsnetwerk ook het rechterrooster. De rechterhelft draagt dus geen stroom en de potentiaal van de rechterplaat is hoog en daarmee ook het linkerrooster in overeenstemming met wat nodig is voor de stabiliteit van de toestand. Dit zij bijv. de toestand welke e n "0" representeerd. Door overal links en rechts in deze beschouwing te verwisselen vinden wij de andere stabiele toestand, welke een "1" veronderstelt. Door bijv. in 0 de potentiaal te meten lezen wij de toestand van de toggle af: lage potentiaal voor een "0", hoge potentiaal voor een "1".

Door in A een positieve puls toe te voeren van voldoende amplitude en tijdsduur kunnen wij de toggle in toestand "0" brengen, onverschillig wat de oorspronkelijke toestand was. Een positieve puls in B bewerkt een instellen van de "1" toestand. Natuurlijk zijn er allerlei variaties. Men kan op de platen pulsen in plaats van op de roosters en ook kan men bijv. in A positief of negatief pulsen i.p.v. in A of B. De condensators in het dwarsnetwerk zijn niet essentieel voor de werking. Zij zijn klein en slechts ter versnelling bedoeld. Het systeem heeft een ernstig nadeel. Als wij nl. in A positief pulsen terwijl de rechterbuis "aan" was, flipt de linker buis aan, het rechterrooster valt, de rechterbuis flopt uit, de rechter plaat schiet omhoog en dus komt er na enige vertraging in A een tweede positieve puls aan. Het systeem werkt positief terug, wat heel onprettig is voor het element, dat de puls in A verwekt heeft. Buitendien zijn de tijdsconstanten van het systeem niet te best.

Hieraan is tegemoetgekomen in de asymmetrische toggle (zie fig.2).

fig. 2



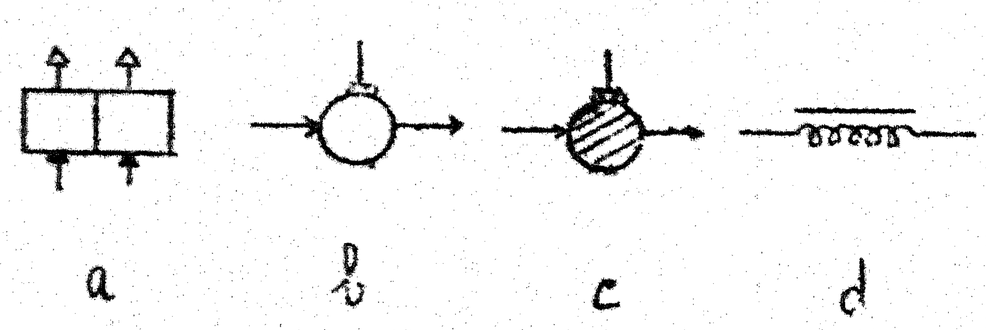
De kathode heeft nu geen vaste potentiaal maar hangt aan het hoogste rooster dankzij een hoge kathodeweerstand. Het linker rooster heeft een vaste potentiaal. In de "0"-toestand draagt de linkerplaat weer stroom, is dus laag en zo ook het rechterrooster. De kathode heeft nu ongeveer de potentiaal van het linkerrooster. In de "1"-toestand draagt de rechterplaat stroom. De linkerplaat is hoog en het rechterrooster is hoger dan het linker. Zo ook de kathode en de linkerhelft is dus solide afgesloten. Instellen van de toestand geschiedt door positief of negatief in A te pulsen.

Merk op dat voor de werking helemaal niet nodig is, dat de rechterplaat een weerstand heeft. Slechts om in C af te kunnen lezen brengt men een kleine weerstand aan. Terugwerking op de signaalleiding is nu uitgesloten omdat er geen verbinding van rechts naar links is. Bovendien zijn de tijdsconstanten beter dan bij de symmetrische toggle.

De teller

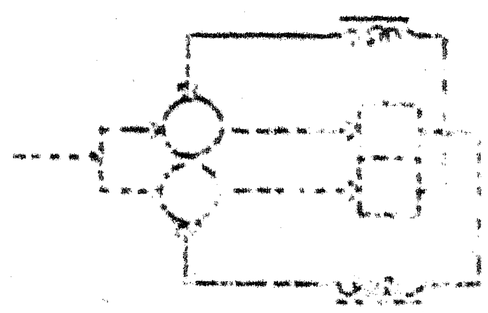
Als teller kunnen wij het schema van fig. 1 aanhouden. Slechts moeten wij het signaal niet asymmetrisch aanbrengen maar evenzeer symmetrisch aanbrengen b.v. aan beide roosters negatief of aan beide platen negatief of aan de kathode positief, en wel zo sterk, dat beide helften afgesloten worden. Nu zijn de condensators in de dwarsnetwerken essentieel en groot. Stel bijv.dat de linker helft van de buis geleidde. Als wij dan bijv. doorde kathode plotseling omhoog te brangen beide helften afsluiten vliegt de linkerplaat op en laadt de rechtercondensator. Als de puls afgelopenis ontlaadt deze condensator zich, waardoor het rechterrooster positief wordt en de rechter buis aanslaat. De condensators vormen nu dus een klein hulpgeheugen, dat alleen dient voor de overgangstijd te oberbruggen. Hier rijst nu een typisch vraagstuk. In het voorgaande immers werd van de pulsamplitude en duur slechts geeist, dat ze groot genoeg waren om de omslag te bewerken. Dit is een binair criterium: Boven of onder de maat. Hier echter is de pulsduur nog aan een tweede grens gebonden. Hij mag nl. niet zo lang zijn, dat de condensator zijn informatie vergeten is. Dit is niet binair, maar scalair en is bedenkelijk. Wel kunnen wij proberen de herinneringstijd van de condensator wat op te voeren door een grote te nemen, maar erg veilig is de zaak niet. Hier is een goed voorbeeld om te laten zien hoe we door een logische constructie verbetering in een dergelijke situatie kunnen brengen. Wij voeren daartoe enige logische symbolen in. In fig 3a staat het symbool voor de toggle.

fig. 3



Een puls wordt voorgesteld door een weerhaakpijl. De bedoeling van het symbool is, dat het een element is met in principe twee (puls-)ingangen en twee statische uitgangen. Een puls op een helft heeft de statische spanning uit dezelfde helft tengevolge. Fig. 3b stelt een poort vóór welks physische realisatie wij later zullen leren kennen. Het is een element, dat een puls doorlaat maar alleen indien eenzekere statische spanning aangelegd is. Fig. 3c is ook een poort, maar deze is normaal open en juist gesloten als de statische spanning is aangelegd. Fig.3d stelt een element voor dat een puls doorlaat, maar met zekere vertraging. Beschouw nu de schakeling in fig. 4. Hier is dus geen teller aanwezig maar een toggle.

fig.4



Stel de bovenste helft is "aan". Dan is alleen de onderste poort open en een inkomende puls begint dus met de toggle om te zetten. Nu mag de pulsduur natuurlijk niet zo lang zijn als de vertragingstijd van de ingebouwde vertragingslijnen, want anders zou zijn staart door de andere poort gaan en de toggle terugzetten. Maar deze vertragingstijd is gemakkelijk groot te maken en de bovengrens is dus aanzienlijk opgeschoven.
Overigens ook de tijd waarbinnen de teller opnieuw aan een puls blootgesteld mag worden! De poorten kunnen bijv. als dubbeltriodes uitgevoerd worden en in dat geval gebruikt deze schakeling drie dubbeltriodes tegenover 1 voor de eenvoudige teller!

Door nog meer buizen te gebruiken is een teller te maken, waarbij de bovengrens voor de pulsduur in het geheel niet meer aanwezig is.

Het snelle geheugen G2

Hoe groot moet het snelle geheugen zijn en aan welke andere eisen moet het voldoen,
Capaciteit: Liefst zo groot mogelijk. Maar dit is slecht combineerbaar met de andere eisen. Daarom moet een noodzakelijk minimum geschat worden. Het zal echter van het type vraagstuk afhangen, wat nodig is. Voor de machines, welke zich niet zuiver met statistische of logische problemen bezig houden zal het oplossen van partiele differentiaalvergelijkingen van groot belang zijn. Hier treden in hoofdzaak twee groepen van vraagstukken op, al naar gelang de differentiaalvergelijking van hyperbolisch of parabolisch dan wel van elliptisch karakter (niet noodzakelijk type) is.
De eerste groep is bijv. gekarakteriseerd door:

$$u_{xx} = u_{tt}$$

Is de oplossing bekend tot aan $t = t_o$, dan kunnen wij de oplossing berekenen voor $t = t_o + A\ t_1$ enz. Het geheugen moet dus de gegevens langs twee lijnen minstens kunnen onthouden en bovens een oude lijn kunnen vervangen door een nieuwelijn. Verder zullen in het algemeen een of meer functies van de coordinaten in de vergelijking voorkomen, welkeonthouden moeten worden.
De tweede groep is bijv. gekarakteriseerd door:

$$u_{xx} + u_{yy} = 0$$

Het geheugen moet in elk roosterpunt minstens een getal kunnen onthouden dat een benadering van u representeert, Verder zullen er natuurlijk weer functies van de coordinaten onthouden moeten worden. Blijkbaar zijn er meer data hier te onthouden, maar vaak zal de roosterwijdte groter kunnen zijn.

In beide gevallen moeten natuurlijk ook de instructies onthouden worden. Het is natuurlijk niet mogelijk precies een capaciteit in het algemeen vast te stellen. Wel is het duidelijk (denk aan 3- en 4 dimensionale vraagstukken), dat ze aanzienlijk moet zijn.
Een ander probleem van formaat is bijv. matrixvermenigvuldiging. Bijv. een vermenigvuldiging van twee matrices van 30 bij 30 eist alleen voor het bergen van de elementen al een capaciteit van 2700 getallen.
Statistische problemen vereisen wellicht weinig instructies, maar het aantal gegevens zal in het algemeen buitengewoon groot zijn.
Beschouwingen door verschillende mensen hebben geleid tot de mening dat een capaciteit van tussen 10^3 en 10^4 getallen voor het snelle geheugen wenselijk en voldoende is. Hier moeten dan ook de instructies voor de besturing geborgen worden, die op later te bespreken wijze als getallen gecode-erd kunnen worden. Wij zullen in het vervolg daarom niet meer spreken van getallen of instructies, maar kortweg over woorden, dat dus ketens van binaire cijfers zijn met niet gespecifieerde betekenis.

Van belang voor de benodigde capaciteit is natuurlijk niet alleen het aantal te bergen woorden, maar evenzeer de lengte van deze woorden, d.w.z. hunner cijfers. Er moet dus een beslissing genomen worden op dit punt. Men bedenke dat een electronische rekenmachine voor omvangrijke berekeningen gebruikt zal worden. Daarvoor is hoge precisie i.h.a. noodzakelijk en een capaciteit van 32 à 40 cijfers per getal redelijk. Wanneer geen genoegen genomen kan worden met deze precisie, dan kan men door geschikte programmmering woorden "gangen" en de precisie daarmee willekeurig hoog opvoeren. Het is echter niet wenselijk onnodig te gangen met het oog op de extra complicaties, dus de nauwkeurigheid van 20 cijfers per getal zou zeker aan de te lage kant zijn. Het preciese aantal cijfers, dat we in een woord wensen te bergen hangt nu ten nauwste samen met de aard van de instructies, welke wij wensen te gebruiken en met het aantal woorden in het geheugen! Immers een minimum instructie moet inhouden het plaatsnummer van een getal in het geheugen en de daarmee te verrichten handeling, bijv. tel getal op plaats nummer 3684 van het geheugen op bij het zich reeds in het rekenkundig orgaan bevindende getal. De besturing kan daarna automatisch de volgende instructie lezen, zodat daarvoor geen plaatsnummer gegeven behoeft te worden. (Men kan andere systemen prefereren, waarover later). Is nu de capaciteit van het geheugen bijv. 2^{12}, dan moeten 12 plaatsen van de instructie alvast gereserveerd worden voor de aanduiding van het plaatsnummer. Verder zal de bewerking zelf aangegeven dienen te worden. Men kan waarschijnlijk niet, zonder de soepelheid van de codering van een vraagstuk te verliezen, met 32 bewerkingsopdrachten uitkomen, maar 64 is zeker ruimschoots voldoende. Dit betekent dus 6 plaatsen voor de bewerking, dus samen 18 plaatsen per order. Maar als we een woord dan minstens 36 "letters" lang maken kunnen wij er twee instructies in onderbrengen, wat economisch is. Wij zullen ons dus in het vervolg de lengte van een woord voorstellen als ca. 40 letters. Maar het bergen van zeg 4000 x 40 = 160 000 binaire cijfers is geen kleine opgave.

Snelheid

De tijd waarbinnen wij een woord na aanvrage ter beschikking willen hebben moet zo klein zijn, dat de snelheid van de rekenkundige bewerkingen niet tevergeefs zijn. Het zal dus afhangen van deze snelheid hoe groot de snelheid van het geheugen dient te zijn. Bij een snel parallel werkend rekenkundig orgaan zullen wij een sneller geheugen wensen dan bij een serieorgaan, dat het kalmer aandoet.

In het parallel geval zal een optelling bijv. kunnen geschieden binnen 10 mikroseconden en een vermenigvuldiging binnen 200 mikroseconden. De wachttijd zal dus bij voorkeur niet langer dienen te zijn dan enkele tientallen mikroseconden. In het seriegeval zal een optelling bijv. pulstijden duren, dus bijv. 20 mikrosecunden en een vermenigvuldiging 800 mikrosecunden. De wachttijd zal dus bij voorkeur niet langer dienen te zijn dan enige honderden mikrosecunden. Deze beschouwingen gelden niet alleen voor "lezen", maar óok voor "schrijven".

Uitwisbaarheid: Van groot belang is de eis van uitwisbaarheid en overschrijfbaarheid van het geheugen. Bij voorkeur dient het inschrijven van een nieuw gegeven het oude gegeven op die plaats automatisch te vernietigen. Het is wel denkbaar een niet uitwisbaar geheugen te gebruiken, maar de soepelheid zal aanzienlijk er door lijden.

Permanentie: Eenzeer nuttige eigenschap is het blijven bestaan van de informatie, ook tijdens afwezigheid van electrische bekrachtiging. Absoluut nodig is het weer niet, want men kan af en toe tijdens een berekening even onderbreken en de inhoud van G2 op G3 vastleggen (dat in ieder geval permanent dient te zijn) en in geval van storing kan men dan teruggrijpen op het tijdstip van deze laatste onderbreking.

Algemeen technische eisen: Betrouwbaarheid kan natuurlijk niet absoluut zijn, maar 1 fout op 10^2 operaties is bijv. vele ordes te onnauwkeurig. Het signaalniveau moet dus hoog liggen ten opzichte van het ruisniveau. Kosten per cijfer en ook het volume per cijfer moet redelijk zijn.

Macroscopisch versus microscopisch

Het is duidelijk dat een macroscopisch G2 uitgesloten is met het oog op de capaciteitseis. Een flip-flop, hoe ideaal ook, kan moeilijk in 100 000-voud worden toegepast! Trouwens het is niet alleen het geheugenelement zelf, dat een rol speelt. Ook het schakelprobleem, dat is het zoeken van een getal op een bepaalde plaats is een belangrijk vraagstuk. Dit schakelprobleem is eigenlijk het centrale probleem, dat de snelheid grotendeels bepaalt. Het schakelvraagstuk kan van ruimtelijk, tijdelijk of gemengd karakter zijn.
Is het geheugen zuiver statisch, dan is het vinden van een bepaalde plaats een ruimtelijk schakelprobleem, dus snel, maar omvangrijk, tenzij door speciale omstandigheden in de constructie van het geheugencomplex een schakelingselement aanwezig is. (Indien dit niet het geval is, is trouwens de schakelapparatuur op zichzelf een complex vergelijkbaar met een macroscopisch geheugen, dus het gestelde doel van een microscopisch geheugen is niet bereikt). Is het geheugen cyclisch, d.w.z. doorloopt de informatie van een bepaald gedeelte van het geheugen cyclisch een aflesstation, dan is het zoeken van een bepaalde plaats een kwestie van tellen, dus traag en eenvoudig. Is het geheugen scanderend, d.w.z. doorloopt het leesorgaan al of niet cyclisch een gedeelte van de statisch geborgen informatie, dan hebben we in vele opzichten te doen met een middenweg, zowel wat snelheid als ingewikkeldheid betreft.
Wil het cyclische type niet aan zijn doel, een microscopisch geheugen te zijn, te kort schieten, dan moet elke groep minstens honderden cijfers bergen. Passeert deze informatie met een snelheid van s megahers het leesstation en zijn er n cijfers in de groep, dan is de wachttijd gemiddeld n/2s mikroseconden.

Wij kunnen s voorlopig zekerniet gunstiger dan 5 verwachten. Stel, dat we 100 parallelle organen toelaten met 10 cijfers willen bereiken. Dan is n = 1000 en de wachttijd bedraagt 100 mikroseconden op zijn minst.
Bij het veranderde geheugen kan de mogelijkheid geschapen worden de normale cyclische scandering te onderbreken en de aangevraagde plaats op te zoeken. Op deze wijze kunnen wellicht hoge snelheden verwezenlijkt worden.

Physische realisatiemogelijkheden van G2.

Op vele wijzen is het mogelijk physische effecten te benutten voor de bouw van een G2, maar het is buitengewoon moeilijk aan de gehele verlanglijst te voldoen. Maar vele voorgestelde methoden berusten op pas kort bekende effecten en het is dus geenszins uitgesloten, dat nog betere methoden toegepast kunnen worden. Enkele redelijke mogelijkheden zijn toepassing van optische zwarting, fluorescentie, acoustische golven, remanent magnetisme, electrische weerstand, electrische ladingen op een isolator. Hiervan zullen wij er enige nu bespreken.

Optische zwarting

Een zeer voor de hand liggende methode is het toepassen van film waarop langs optische weg merktekens aangebracht worden, welke weer langs optische weg en een fotocel afgelezen kunnen worden. De capaciteit zou zeer groot gemaakt kunnen worden bij gebruik van fijnkorrelige of korrelloze film, wat evenwel de snelheid ongunstig beinvloedt. Ontwikkeling van het beeld is nodig, wat de effectieve snelheid zeer verlaagt en bovendien de aanwezigheid van chemicalien veronderstelt. Verder geeft de film meestal een te lineair beeld, terwijl een binair beeld (zwart of wit zonder tussentonen) vereist wordt. Het beeld is verder niet uitwisbaar. Het is echter best mogelijk, dat filmmedia gevonden worden welke voor ons doel meer geschikt zijn.

Fluorescentie.

Sommige stoffen vertonen de eigenschap na voorafgegane belichting gedurende enige tijd licht uit te stralen. Na enige tijd is deze informatie verdwenen en kan het medium opnieuw beschreven worden. Enkele speciale stoffen hebben de eigenschap door licht van een bepaalde intensiteit niet tot fluorescentie gebracht te worden maar wel tot een metastabiele toestand, welke toestand vrij lang (van de orde van vele milliseconden) bewaard blijft. Als wij nog eens op dezelfde plaats licht te laten vallen verraadt de plek of ze al eerder belicht is geweest. Is dat nl. het geval, dan geraakt ze van de metastabiele in de fluorescerende toestand, welke snel ontaardt in de gewone niet geexciteerde toestand. Was de plek niet belicht geweest, dan gebeurt er niets.

Om dit toe te passen kan men een snel roterende schijf nemen, welks omtrek met een zodanige stof is bedekt. Op een bepaalde plek is een schrijfstation, dat, wanneer een cijfer 1 moet worden geschreven een lichtstraal zendt naar de schijf, daarmede het getroffen plekje in de metastabiele toestand brengend. Op zekere afstand, bijv. 60° is een leesstation gevestigd. Dit emitteert iedere keer, dat een plekje voorbijkomt een lichtsignaal. Bevat het plekje een 0, dan gebeurt er niets. Bevat het een 1, dan slaat het over in de fluorescerende toestand en een fotocel leest dit af en zendt het behalve eventueel aan de aanvrager ook door aan het schrijfstation, dat het gelijk weer inschrijft op de schijf. De resterende 300° van de schijf dienen dan uitsluitend voor het afsterven van het signaal, zodat de schijf weer onbeschreven onder het schrijfstation doorloopt. Het systeem is waarschijnlijk inferieur aan het later te behandelen magnetische systeem.

Electrische weerstand.

Verschillende stoffen vertonen in hun electrische weerstand een indicatie van vooraf plaats gehad hebbende of plaats vindende stroom-

doorgang. Zo is papier een isolator maar wanneer het doorgeslagen is door een vonk is het vanwege de bij de verbranding vrijgekomen koolstof een geleider. Neon heeft een hoge weerstand zolang geen stroomgeleiding plaats heeft en een veel lagere indien dit wel het geval is. Op die grondslag is een geheugen realiseerbaar, bij voorkeur uitgevoerd volgens het zg. cross-bar principe. Een medium als papier is niet uitwisbaar. Bij het gebruik van neon bestaat het gevaar, dat nabije kruispunten ontstoken worden door weglekkende ionen.

Acoustische golven.

Op meer succes kan gehoopt worden bij de toepassing van acoustische golven, bijv. in een met kwik gevulde buis. Een roestvrije stalen buis met kwik gevuld, is aan weerszijden afgesloten met een stop, waarop een dun kwartskristalplaatje bevestigd is. De ene zijde maakt dus contact met het kwik, terwijl de andere zijde, niet aan het kwik blootgesteld, verzilverd is. Brengen we op dit laagje een electrische puls aan, dan verwekken wij door het verschijnsel van de piezoelectriciteit een longitudinale trilling in het kwartsplaatje, welke trilling op het kwik wordt overgebracht en als acoustische golf wordt voortgeplant. Deze golf treft het plaatje aan de andere kant en verwekt daar omgekeerd een electrisch pulsje, ca. 1000 keer verzwakt overigens door de grote koppelingsverliezen. Dit uitgangssignaal wordt versterkt en gelijkgericht en door een poort geleid, waarvan het wachtsignaal bediend wordt door de klok, dat is een vibrator, welke met constante frequentie een standaardpuls uitzendt. Is de acoustische vertragingstijd een geheel aantal maal de trillingstijd, dan wordt het gelijkgerichte en sterk verbrede signaal tot een nette standaardpuls bijgeknipt, welke weer op het voorste kwartsplaatje wordt toegevoerd. Zodoende blijft deze puls circuleren. Is de frequentie bijv. 1 megaherz en de lengte van de buis ca. 1.5 meter, dan is de acoustische vertraging ca. 1 milliseconde. Immers de voortplantingssnelheid van het geluid in kwik bedraagt ca. 1.5 km/sec. Het is duidelijk, dat er dan dus eventueel een trein van 1000 pulsen kan circuleren. Wanneer wij afspreken, dat een puls het cijfer 1 voorstelt en het ontbreken van een puls op een plaats, waar er één had kunnen zijn, een nul, dan hebben we een geheugen. Om te lezen in dit geheugen maken wij een aftakking in de circulatieleiding welke wij openen, zodra een teller ons vertelt dat het gezochte cijfer aan het eind gearriveerd is. Schrijven kunnen wij op analoge wijze doen. De gemiddelde wachttijd voor lezen en schrijven bedraagt in het geschetste geval 500 mikrosecconden. Dit kan verbeterd worden door de frequentie op te voeren (bij gelijke capaciteit van een buis neemt dan meteen zijn lengte af) en ook door bij het programmeren er voor te zorgen, dat getallen op een plaats staan, die bij aanvrage al bijna aan het eind

van de buis zijn. Dit laatste dan alleen bij een machine met strenge tijdregelmaat in het rekenproces, dus speciaal bij een seriemachine. Bij een seriemachine zal men een woord in één buis laten circuleren. Bij gebruik op een parallele machine is het voordelig dit woord ook parallel in evenzovele buizen te laten circuleren als er cijfers in het woord zijn.

Technische moeilijkheden zijn er vele. De temperatuurcontrole is van groot belang. Immers een verandering van de temperatuur met 3 celsiusgraden veroorzaakt een variatie van 1°/oo in de geluidsnelheid. De temperatuur moet dus binnen een fractie van een graad in alle buizen gelijk worden gehouden. Hieraan kan worden tegemoetgekomen door alle acoustische stralen in de grote cylinder onder te brengen. Het "over en weer praten" van stralen schijnt niet op te treden.

Remanent magnetisme.

Wij nemen een torus (van gelamelleerd weekijzer bijv.) met een spoeltje om een gedeelte van de omtrek. Bij bekrachtiging van het spoeltje verwekken wij dus een flux in de tonus. Door de tonus door een dunne spleet te onderbreken verhogen wij de magnetische weerstand van het circuit enorm en de krachtlijnen zoeken een andere mogelijkheid om rond te lopen. Die kan men bieden door langs of door de spleet een dunne stalen draad of band te laten lopen, van een magneetstaal met hoge remanentie. Bij bekrachtiging van de spoel lopen de krachtlijnen door dit staal en laten na afloop door de remanentie een magnetisch gebiedje na. Wij kunnen plus en min pulsen op de spoel en zodoende een 1 en een 0 onderscheiden door de polariteit van het magnetische veld. Lezen gaat door de band met een zekere snelheid voorbij hetzelfde apparaat te bewegen. Het magneetveld op de band induceert dan via de tonus een e.m.k. in de spoel, welke wij kunnen aflezen.

Voor een snel geheugen kan men een roterende trommel gebruiken, bedekt met een dun laagje magnetisch materiaal. Langs een beschrijvende monteert men een rij schrijf- en afleeskoppen als boven beschreven en men kan dus de hele trommel volschrijven met magneetveldjes. Technisch mogelijk is een pakkingsdichtheid van 4 pulsen per mm, dus op een trommel van ca. 8 cm middellijn kan men 1000 pulsen in omtreksrichting schrijven analoog aan de capaciteit van een kwikbuis. Met een beetje goede wil kan men in de richting van de beschrijvende een paar mm per cijfer realiseren, zodat op een lengte van enige decimeters en een diameter van 8 cm ongeveer 100 000 cijfers genoteerd kunnen worden. Het aantal omwentelingen zou mogelijk 1000 per seconde kunnen bedragen, wat een gemiddelde wachttijd van 500 mikroseconden betekent. Door b.v. langs 10 beschrijvenden dergelijke rijen schrijfkoppen aan te brengen zou men deze tijd tot 50 mikroseconden terug kunnen brengen, wat zelfs

voor een parallele machine een heel mooie tijd zou zijn.

Men kan een getal in omtreksrichting opschrijven (voor een seriemachine) of in de richting van een beschrijvende (voor een parallele machine). Aan het begin van elke "regel" kan men een constante minpuls aanbrengen, door de tellers de stand van de trommel te laten bepalen en bovendien kan men één enkele regel van een extra puls voorzien om een nulpunt voor de telling vast te leggen.

Electrische ladingen op een isolator.

De meest pretentieuze vorm van geheugens zijn die welke ontwikkeld zijn uit de televisietechniek en verwante gebieden. Aangezien men dan met een al of niet speciale kathodestraalbuis werkt is te verwachten, dat dit de manier zal zijn om uiterste snelheid te behalen.

Wanneer men op een isolator als glas of mica plaatselijk een bepaalde lading aanbrengt, zal deze in de loop van de tijd weglekken en wij moeten dus een methode hebben om deze lading te fixeren. Dit kan geschieden door een electronenregen op het oppervlak te zenden. Deze electronenregen verwekt n.l. secundaire emissie en nu is het zo gesteld, dat de verhouding tussen de uitgezonden secundaire electronen tot de inkomende primaire afhangt van de energie van de inkomende electronen. Deze verhouding is n.l. kleiner dan 1 bij lage energie en groter dan 1 bij hoge energie. Nu hangt de energie af van het verschil van de potentiaal op het getroffen oppervlak en die van de kathode, welke de electronen uitzendt. Op een plaats op het oppervlak met een potentiaal betrekkelijk dicht boven de kathode, dus laag, komen de electronen in met lage energie. Zij maken daar dan minder secundaire electronen vrij dan hun eigen aantal bedraagt en er is een surplus van inkomende electronen. Dit surplus verlaagt de spanning nog meer tot een stabiele toestand bereikt wordt, waarbij de potentiaal gelijk is aan de kathodepotentiaal (in werkelijkheid een weinig hoger, omdat lek optreedt). Is de oorspronkelijke potentiaal tamelijk hoog boven die van de kathode, dan komen de electronen met veel energie in en maken meer electronen vrij dan hun eigen aantal bedraagt, zodat de potentiaal nog verder opgevoerd wordt. Ook hier is een bovengrens, nl. de potentiaal van het rooster (de collector), dat de secundaire electronen op moet vangen. De potentiaal van het oppervlak kan hier niet boven komen, omdat de secundaire electronen dan weer terugregenen (met lage energie!) op het oppervlak. Wij zien, dat hier een typisch binaire situatie ontstaat. Er zijn nl. maar twee stabiele evenwichtpotentialen, nl. de kathodepotentiaal en de collectorpotentiaal.

Nu is het de kunst om deze ladingen in verschillende punten aan te brengen en af te lezen. Daar zijn verschillende methodes voor in ont-

wikkeling.

De meest gespecialiseerde constructie is de zg. selectron. Dit is een grote vacuumbuis met een groot aantal electroden. Allereerst zijn er één of meer kathodes, die de electronenregen verwekken. Deze electronen passeren op hun weg naar de isolator allereerst een gewoon versnellingsrooster, vervolgens twee selectieroosters, dan het reeds genoemde collectorrooster en treffen dan de micaisolator. De drie laatst genoemde roosters bevinden zich dicht bij de isolator. De selectieroosters bestaan beide uit een aantal evenwijdige staven, en beide stelsels zijn onderling loodrecht. Vanaf de kathode kunnen de electronen dus de isolator zien door een stelsel vensters, waarvan de zijden gevormd worden door de beide stelsels selecteerstaven. De plaatsen op de isolator, waar ladingen aangebracht zullen worden, zijn op deze wijze bepaald. Wanneer alle vier staven, welke een bepaald venster omringen en positieve potentiaal hebben schieten de electronen van de kathode recht door het venster heen. Zijn alle vier negatief dan is het venster geblokkeerd en ook wanneer één, twee of drie van de staven positief zijn kunnen de electronen er niet door heen, omdat ze dan van de negatieve staven afbuigen, naar de positieve toebuigen en daar opgevangen worden. Een venster is dus alleen open, wanneer alle vier omringende staven positief zijn. In de toestand van "onthouden" zijn alle staven positief, alle vensters dus open en alle plaatsen op de isolator worden gefixeerd. Willen we een cijfer lezen, dan blokkeren we even alle vensters, behalve het met het cijfer corresponderende en alleen de betreffende plaats op de isolator wordt bestraald. De eenvoudigste manier om nu de toestand af te lezen is deze, dat de isolator bedekt is met een laagje van een fluorescerend materiaal dat oplicht wanneer electronen met hoge energie inkomen en niet oplicht wanneer dit niet het geval is. Wanneer we een stof gebruiken als ZnO, dat snel reageert, is het lichten van alle andere vensters (ca. 50%!) na enige mikroseconden voldoende sterk afgenomen om met een fotocel, die de hele buis kan zien, te bepalen of één lichtend venster overgebleven is of niet, dus of de gezochte plaats een 1 dan wel een 0 bevatte. Zodra het lezen afgelopen is, moeten natuurlijk alle vensters weer open, om de langzaam weglekkende potentialen weer op peil te brengen.

Om te kunnen schrijven is nog een met de werkende kant van de isolator capacitief gekoppelde metalen plaat aangebracht (dus genoeg om het fluorescerende licht door te laten). Willen we nu op een bepaalde plaats schrijven, dan blokkeren we weer alle vensters behalve het corresponderende. Vervolgens brengen we op de plaat een puls aan met een zeer steil front, maar zeer geleidelijk afnemend. Willen wij een 1 schrijven, dan gebruiken wij een positieve puls en voor een 0 een negatieve puls. Stel wij willen een 1 schrijven en het venster bevatte een

0. Door het steile pulsfront overweldigen wij de stabilisatie van de electronenregen en de potentiaal wordt hoog. De puls neemt echter geleidelijk af en de stabilisatie verhindert nu het weer teruglopen naar de oude toestand. Alle andere plaatsen van de isolator zijn natuurlijk ook opgelopen maar deze worden nit gestabiliseerd en lopen dus weer terug. Dan openen wij weer alle vensters en de nieuwe toestand wordt weer gestabileseerd. Het is duidelijk wat er plaats vindt in andere gevallen dan het hier beschrevene.

Zowel het lees- als schrijfproces duurt een aantal mikroseconden, dus is buitengewoon snel. Nu moeten wij nog een eenvoudige methode hebben om de vensters te bedienen. Gelukkig is een venster geblokkeerd zodra maar minstens één van de vier omringende staven negatief is. Wij beschouwen nu één selectorrooster. De staven delen wij in twee families in. Stel er zijn 64 staven. Nu maken wij 16 aansluitingen en verbinden die ieder met vier staven. Laten wij nu de staven aansluiten volgens het schema: 1,2,3,4,5,6,7,8 en 1,2,3,4,5,6,7,8. Wanneer wij nu de staven aansluiten volgens het schema:

1122314251627182132433445364738415263546556678617283748576877 88

dan realiseert men, dat door éé staaf van de ene familie en één van de andere positief te maken, wij juist één en maar één plaats definieren, waar twee positieve staven zich naast alkaar bevinden. Hetzelfde geldt voor het andere selecteerrooster en zodoende kunnen wij met 32 aansluitingen 4096 vensters controleren, i.h.a. met N aansluitingen $(N/4)^4$ vensters, dus bijv. met 128 aansluitingen kan men 1049576 vensters controleren. Men kan andere stelsels bedenken, die overigens ook hun bezwaren hebben, welke nog machtiger zijn. Bij gebruik van meerdere selectieroosters kan men zelfs met N aansluitingen $2^{N/2}$ vensters controleren. Dit zou met 32 aansluitingen al 65536 vensters betekenen!

De hier geschetste buis is een typisch voor een parallele machine geschikt geheugenelement. Bij het genoemde aantal van 4096 vensters per buis zouden dan ca. 40 buizen voldoende zijn om het gehele snelle geheugen te vormen. Op corresponderende plaatsen van de verschillende buizen staan dan de cijfers van één getal. De hele selectieschakeling is dan dezelfde voor alle buizen! De technische mocilijkheden zijn vele.

Het nadeel van een speciale buis als de selector zijn de hoge kosten. Daarom zijn allerlei pogingen gedaan om met meer normale apparatuur een soortgelijk resultaat te verkrijgen.

Zo is een soort kathodestraalbuis ontwikkeld, waarbij de informatie in een matrix op het scherm geborgen kan worden en door een electronen regen vanuit een centrale kathode gestabiliseerd wordt op de beschre wijze. In plaats van de constructief ingewikkelde selecteerroosters

wordt lezen en schrijven bewerkstelligd door twee extra kathodes met afbuigapparatuur, welke een geconcentreerde straal kunnen werpen op een bepaald punt van het scherm, wat door doelmatige bekrachtiging van de afbuigapparatuur kan worden bereikt, maar hoge precisie vereist.

Een ander idee vormt het gebruik van een gewone kathodestraalbuis, waarbij een straal het scherm scandeert op de wijze analoog als bij televisie gebruikelijk is. Wordt de buitenzijde van het scherm van een dunne metaallaag voorzien, dan kan men daarop aanduiding verkrijgen van de toestand, waarin de straal het scherm vindt. Bijv. kan men een woord schrijven door variatie van de straalintensiteit en wanneer de nu ongevarieerde straal later weer dit woord passeert kan men op de metaallaag een potentiaal verkrijgen, welke met de oorspronkelijke variatie overeenkomt. Dit passeren van de straal verwoest het woord maar het signaal van de metaalplaat kan worden gebruikt om een andere buis het woord weer op te laten schrijven opwel door tussenschakeling van een vertraging kan men dezelfde straal het woord weer neer laten schrijven. Een ander systeem berust hierop, dat men een nul weergeeft door een stipje en een 1 door een streepje op het scherm. Vindt de straal een streepje, dan geeft de metaalplaat een negatieve puls af waarmee wij de straal direct afkappen, zodoende het stipje weer restaurerende. Vindt de straal een streepje, dan is een positieve puls het gevolg, welk signaal wij kunnen gebruiken, om de straal weer een streepje te laten schrijven. Bij al deze systemen is een belangrijke electronische apparatuur noodzakelijk.

HET REKENKUNDIG ORGAAN

Evenals het geheugen is het rekenkundig orgaan niet alleen een zelfstandige eenheid maar maakt ook op bescheiden schaal deel uit van andere organen. Zo was het bijv. al gebleken bij verschillende types van geheugen zgn. tellers te gebruiken, welke eenvoudige optelmechanismes zijn, alleen in staat om bij het getal, dat ze reeds bevatten een 1 op te tellen. Zij worden ook toegepast in allerlei andere physische instrumenten. Wij hebben reeds kennis gemaakt met de binaire teller. Door een aantal van deze binaire tellers aanéén te schakelen kan men een complete teller maken. Een schema is gegeven in figuur 4. Hierin is slechts één kolom van de teller weergegeven. Links bevindt zich de reeds bekende binaire teller, dus een symmetrische flip-flop met flinke condensators C_1, C_2. Deze wordt in de andere toestand overgebracht door een negatieve puls via de condensator C_3, welke een tweetallenoverdracht van de vorige kolom voorstelt. Wanneer de linkerhelft van de flipflop "aan" is, stelt de toestand een 0 voor. Laten wij aannemen, dat dit het geval is. Een puls door C_3 slaat de rechterhelft van de buis aan en de flipflop stelt nu een 1 voor, zoals gewenst. Daarbij valt het rooster van de triode rechts van de flipflop beneden de door het verdeelnetwerk

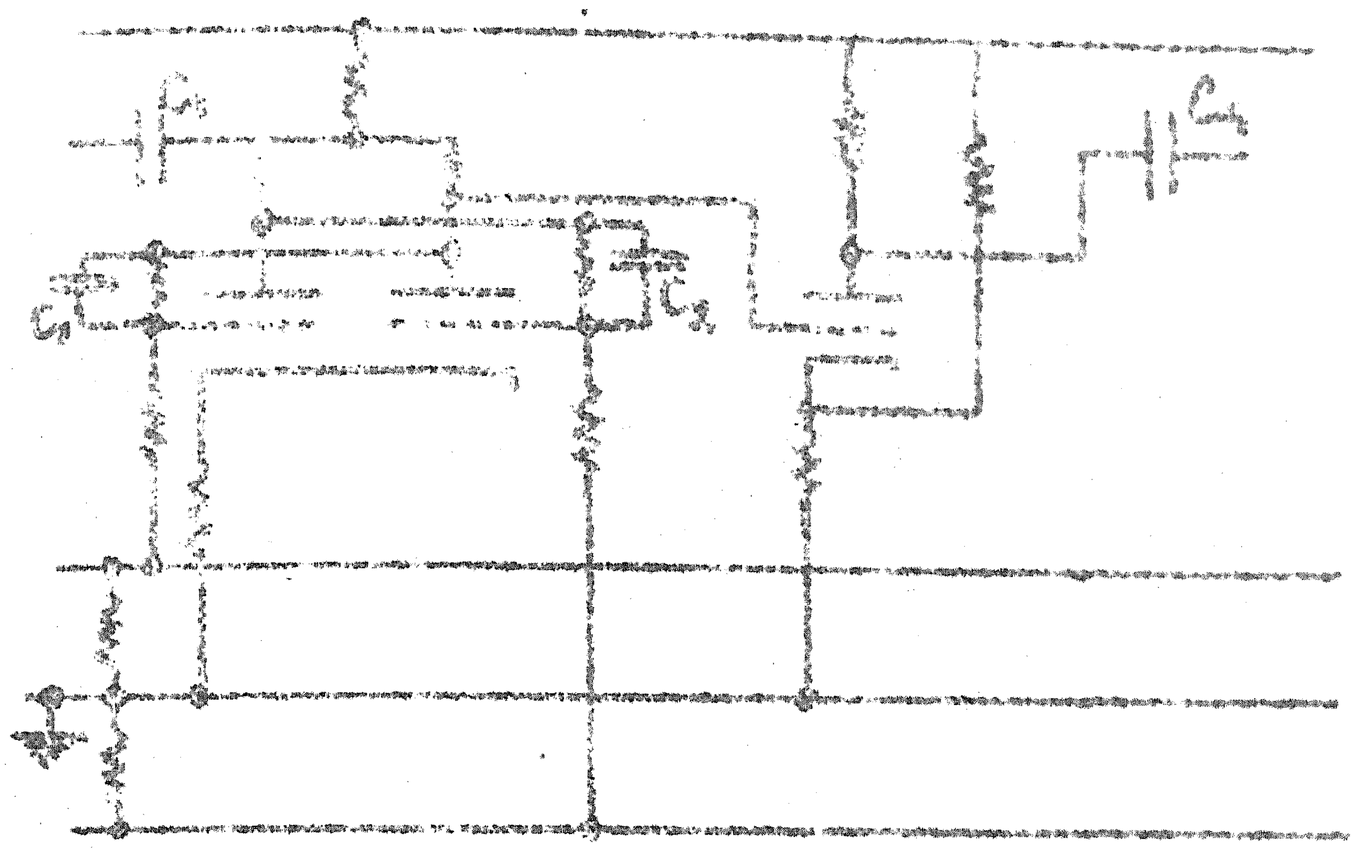


fig. 4

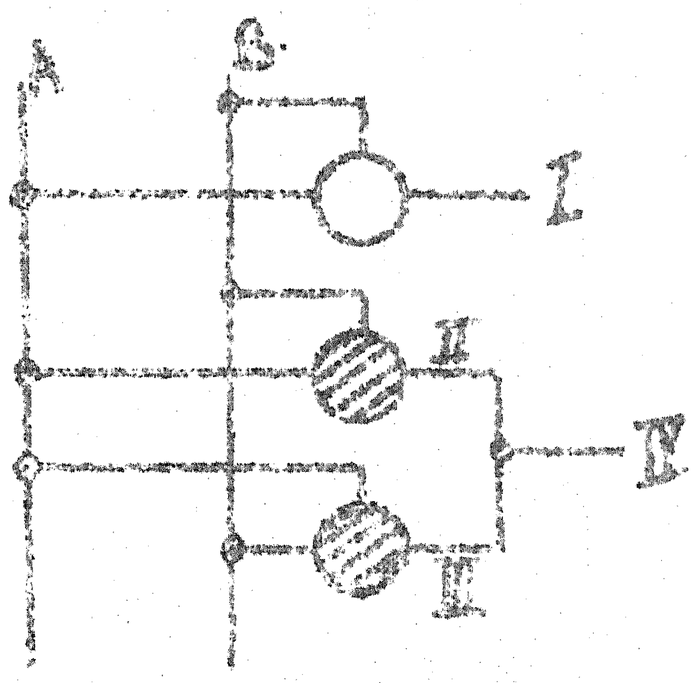


fig. 5

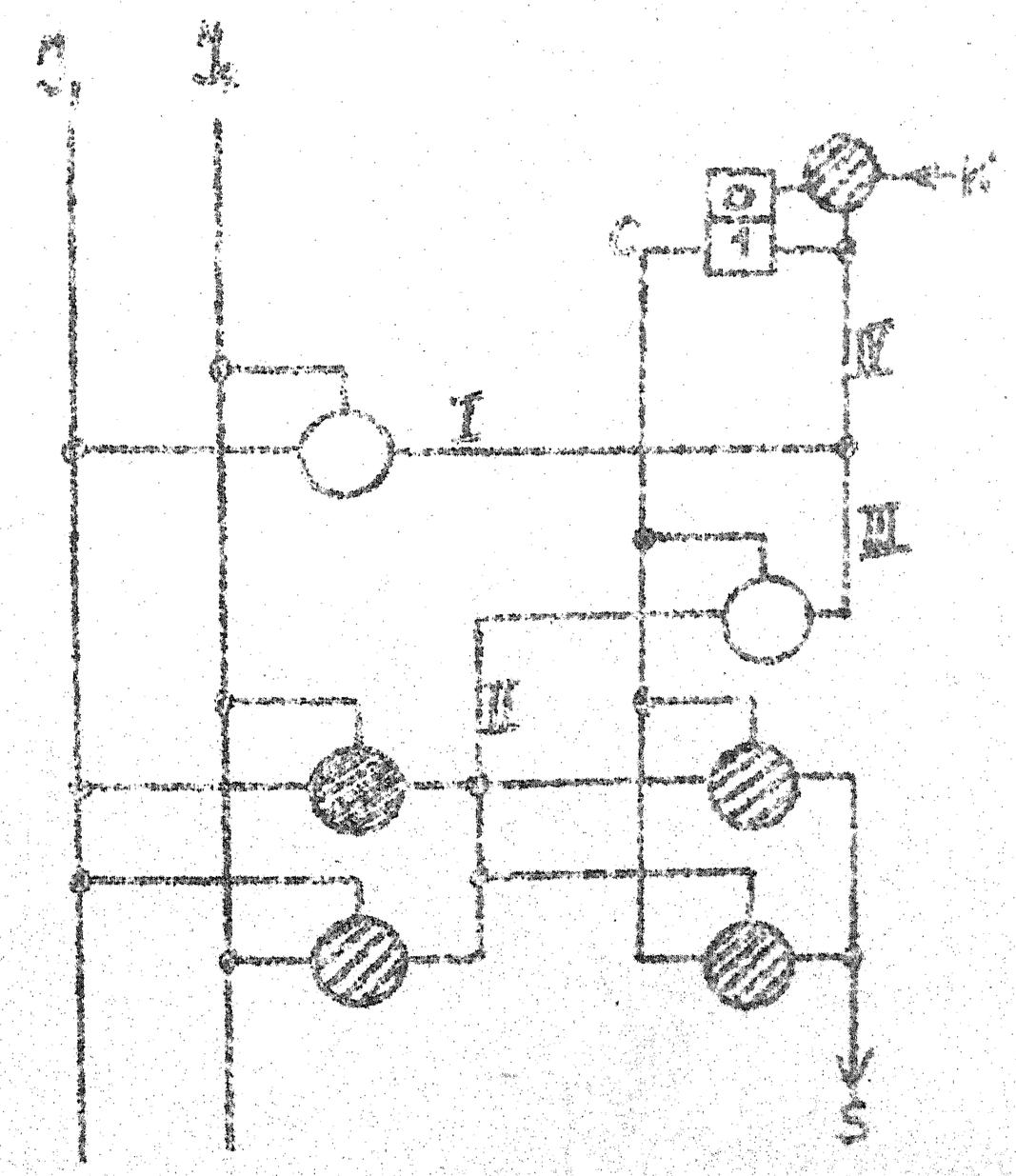


fig. 6

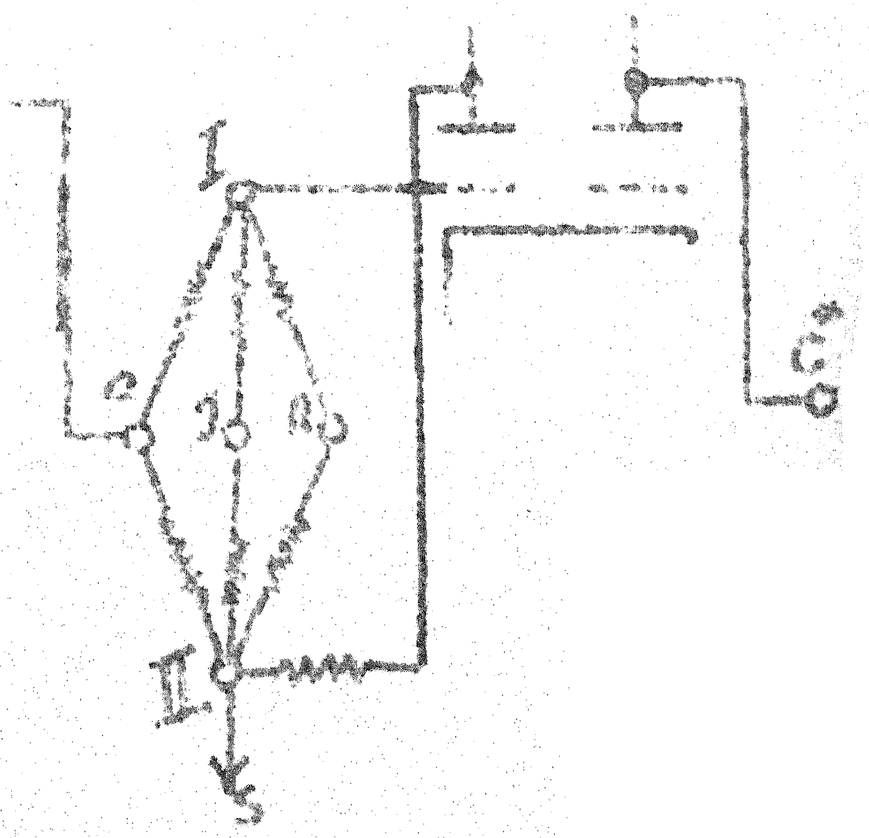


fig. 7

bepaald potentiaal en de triode slaat af. Dit veroorzaakt een positieve puls door C_4, welke eenvoudig voor de volgende (d.w.z. links ervan staande!) kolom hetzelfde is als voor de beschouwde kolom de C_3. Deze puls heeft echter geen effect, omdat de flipflop niet reageert op een positieve puls op zijn anode. Komt nu weer een negatieve puls door C_3 binnen, dan springt de flipflop weer in zijn 0-toestand, maar moet nu een tweetallenoverdracht bewerken in de volgende kolom. Dit geschiedt, omdat de rechteranode nu stijgt en dus ook het rooster van de tussengeschakelde triode. Deze slaat aan en een negatieve puls schiet door C_4, wat in de volgende flipflop een omzetting bewerkt. De reden, dat de linkerroosters van de flipflop via de weerstanden verbonden zijn met een aparte lijn I, welke weer via een weerstand verbonden is met de aardleiding, is, dat men door deze lijn I op potentiaal te brengen, de hele teller kan schoonmaken, d.w.z. alle flipflops op nul stellen.

Veel moeilijker is het optellen van twee complete getallen met al hun tweetallenoverdrachten. Wij zullen enkele schema's aangeven. Het eerste is voor een seriemachine. Hierbij hebben wij dus twee getallen I_1 (incident 1) en I_2 (incident 2), voorgesteld als pulstreinen op twee leidingen. Ieder moment, dat een cijfer op beide lijnen in de opteller arriveert (de staarten komen eerst!) moet de opteller het somcijfer vormen en uitzenden op de uitgangslijn, daarbij rekening houdende met een eventueel van de vorige optelling wachtende tweetallenoverdracht C ("carry"). Hier blijkt het essentieel andere: er moeten drie cijfers opgeteld worden, nl. I_1, I_2 en C. Een eventueel bij de optelling optredende carry moet gevormd worden en opgeborgen in een flipflop om paraat te zijn als de cijfers van de volgende kolom arriveren. Om het eigenlijke schema te verduidelijken beschouwen wij eerst een paar onderdelen (Zie fig. 5). Laten A en B twee lijnen zijn, waar op een gegeven moment al of niet een puls kan arriveren (d.w.z. een 1 dan wel een 0). Op de lijn I komt dan alleen een puls, als beide lijnen een puls dragen. Op lijn II komt alleen een puls als lijn A een puls draagt en lijn B niet. Op lijn III komt alleen een puls als lijn A geen puls draagt en lijn B wel. Op lijn IV komt dus dan en alleen dan een puls, wanneer één en slechts één van de lijnen A en B een puls draagt. Nu kunnen wij een opteller samenstellen (Zie fig. 6). Zie eerst onderaan. Op lijn II komt alleen een puls als een 0-1 combinatie op de I_1 en I_2 lijnen aanwezig is. Op de S-lijn is het trucje nog eens toegepast, nu met de II-lijn en de carry-lijn. Op de S-lijn komt dus alleen een puls als op de I_1, I_2 en C lijnen één van de vier combinaties 001, 010,

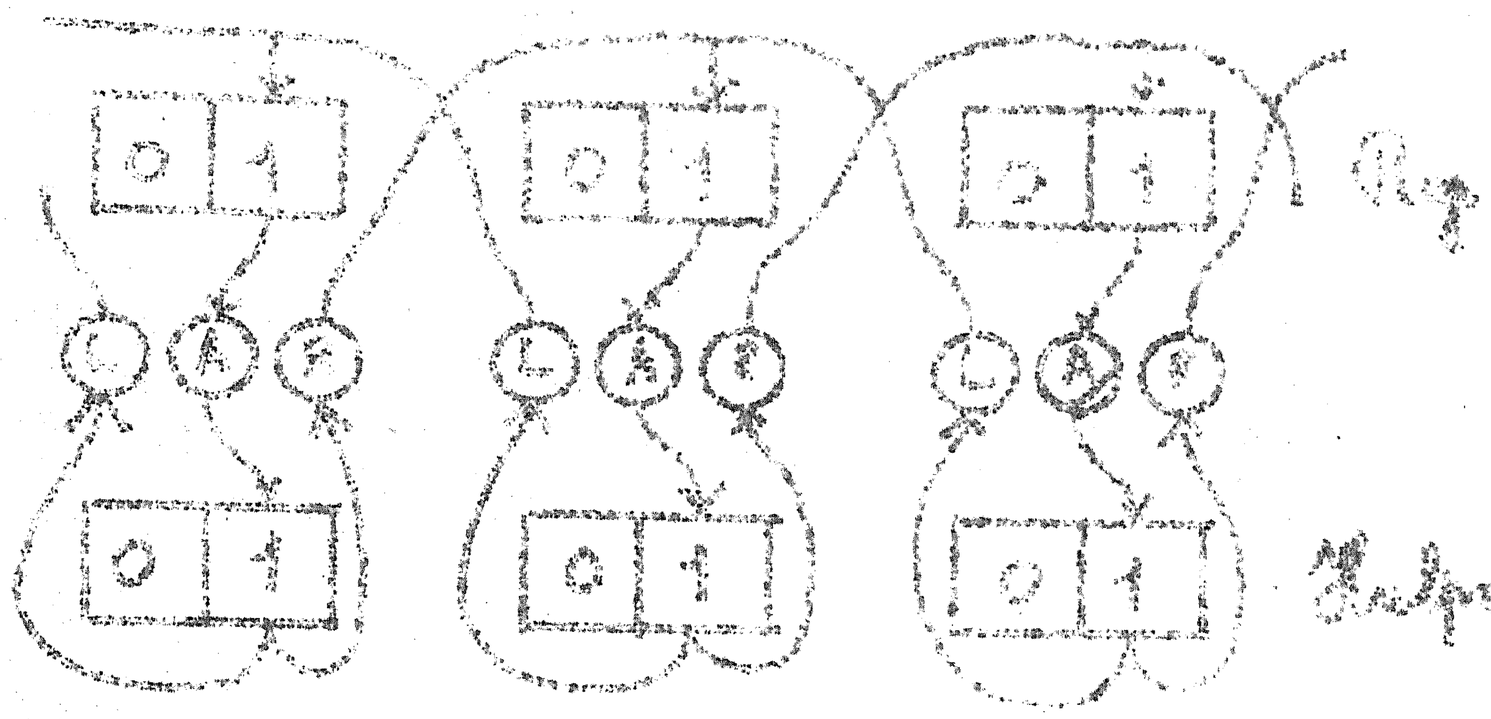


fig. 4

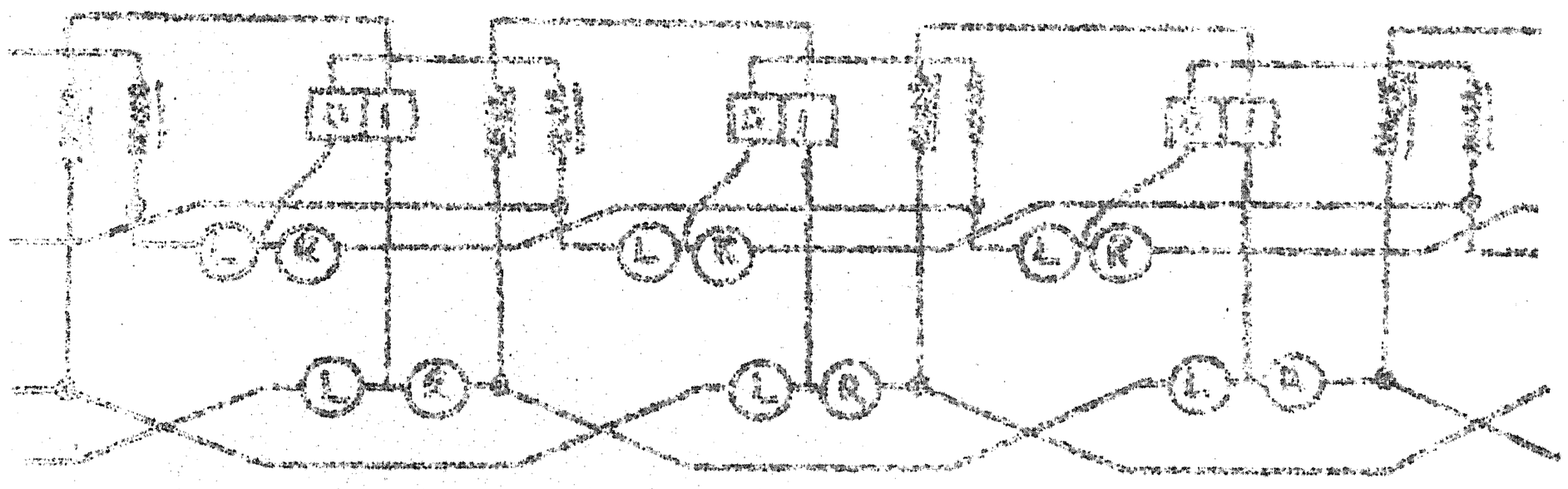


fig. 5

100 of 111 aanwezig is, dat zijn juist de combinaties waarin het somcijfer een 1 moet zijn. Op de S-lijn verschijnt dus precies het somcijfer. Nu moet nog de carry klaargezet worden voor de volgende kolom. Op de lijn I verschijnt alleen een puls als een 1 zowel aanwezig is op de I_1 en de I_2 lijn, onverschillig of op de C-lijn een 0 dan wel een 1 aanwezig is. Op lijn III komt alleen een puls als zowel een 1 aanwezig is op de II-lijn als op de C-lijn. Op lijn IV komt dus een puls als een van de combinaties 110, 111, 011 of 101 aanwezig is, d.w.z. juist in die vier gevallen wanneer een carry verwekt moet worden. Lijn IV is dus de carrylijn. Deze puls, indien aanwezig, stelt de carry flipflop op 1. Is de puls niet aanwezig, dan zet de puls van de klok K de carry flipflop op 0. Natuurlijk moeten de vertragingen zodanig zijn, dat de flipflop eerst omslaat als de staarten van de pulsen door de poorten zijn.

Als voorbeeld van opteller voor een parallele machine geven wij het schema van een zgn. Kirchhoff-opteller. Men stelle zich voor, dat twee registers, bestaande uit een rij flipflops, aanwezig zijn, een voor het incidente (I) en één voor het residente cijfer (R). Bij de optelling per kolom moet natuurlijk ook de carry (C) van de rechtsgelegen kolom in rekening gebracht worden. Tevens moet het somcijfer S en de carry C^* voor de linksgelegen kolom verwekt worden.

Laat op iedere lijn een 1 voorgesteld worden door een hoge en een 0 door een lage potentiaal. De potentiaal van het sterpunt I kan vier waarden hebben, al naar gelang er op de C, I, R aansluitingen geen, één, twee of drie cijfers 1 aanwezig zijn. De carry C^* moet verwekt worden bij twee of drie cijfers 1, d.w.z. als de potentiaal van I heel hoog is. Wordt de potentiaal van het rechterrooster gehouden op een waarde beneden de waarde behorende bij twee of drie cijfers 1, maar boven de waarde behorende bij geen of een cijfer 1, dan is de rechterbuis aan, wanneer geen C^* moet worden verwekt en af, wanneer wel een C^* moet worden verwekt. Dit klopt met de potentiaal van de rechteranode, welke verbonden is met de carryaansluiting van de links gelegen kolom. Vervolgens moet het somcijfer worden verwekt. Dit is gelijk aan de som van de cijfers C + I + R vermiderd met tweemaal C^*. De negatieve C^* wordt geleverd door de linkse anode en de optelling wordt weer in een netwerk geleverd. Let op, dat eerst de carry wordt gevormd en dat pas daarna het somcijfer gevormd wordt. Dit is logisch, want de carry moet verder op zijn reis en heeft dus prioriteit.

Let het oog op het vermenigvuldigen met of delen door 2, het zg. schuiven, wat een veelvuldig voorkomend proces is, bijv. in vermenigvuldigen reeds voorkomt, moet een flipflopregister eventueel voorzien worden van schuivingsmogelijkheid. Een foolproof schema om een parallel register te laten schuiven is gegeven in fig 9 . Parallel aan het eigenlijke register is een hulpregister gebouwd. Schuiven vereist vier handelingen.

De eerste is schoonmaken van het hulpregister.Tweede handeling is:open alle A poorten;dan wordt de inhoud van het hulpregister gelijk aan die van het eigenlijke register.Derde handeling:maak register schoon.Vierde handeling:open de L-poorten of de R-poorten,dan komt de inhoud van het hulpregister één plaats naar links of rechts verschoven in het register terug.Bij deze uitvoering is veel materiaal benodigd en het schuiven duurt lang.Als we niet alleen 1 met 1 maar ook 0 met 0 verbinden door poorten kan het schoonmaken van de registers achterwege blijven.Dat scheelt alvast een factor twee in tijd.Wat minder zekermaar veel eenvoudiger is het schema van fig.9.4. Hier zijn vertragingslijntjes ingebouwd,welke er voor moeten zorgen,dat de flipflops hun nieuwe stand pas innemen,nadat de poorten weer dicht zijn.Hier bestaat het hele schuifproces uit het even openen van de L-of de R-poorten.

Waaruit moet nu het Rekenkundig orgaan tenslotte bestaan?Voldoende is voor optellen,aftrekken,vermenigvuldigen en delen van een parallelemachine een drietal componenten,t.w.een nietschuivend gehnugen register (GR),een schuivend rekenkundig register (RR) en een schuivend optelregister (OR).Het (GR) is eenvoudig de uitgang van het geheugen.Het OR is het enig eigenlijk rekenende register,dat een getal uit GR op kan nemen en optellen bij wat het reeds bevat.Willen we vermenigvuldigen,dan plaatsen wij de vermenigvuldiger in RR en het vermenigvuldigtal in GR.De besturing onderzoekt nu of het laatste cijfer in RR een 1 if of een 0.Is het een 1,dan telt hij GR op in OR en schuift vervolgens zowel RR als OR een plaats naar rechts.Is het een 0,dan schuift hij alleen zowel RR als OR naar rechts.Bij het schuiven van RR verliezen wij het dan laatste cijfer, maar dat interesseert ons niet.Bij het schuiven van OR zouden wij echter ook de laatste cijfers van het product verliezen.Dit is nu ongewenst, omdat het voor kan komen,dat wij van een product van twee getallen van 40 binalen alle 80 binalen wensen te kennen voor precisieberekeningen (zg."gangen").Wij kunnen nu gebruik maken van het feit,dat bij het schuiven van het RR links telkens een plaats vrij komt.Nu verbinden wij het OR zo met het RR,dat bij een schuiving naar rechts van OR en RR gelijktijdig het laatste cijfer van OR voorin RR gestopt wordt.Op die wijze houden wij na afloop van de vermenigvuldiging de 40 hoofdbinalen van het product in OR en de volgende 40 binalen in RR.

Voor delen plaatsen wij het deeltal in OR en de deler in GR.In RR verschijnt dan het quotiënt.

Bij de besturing van de optelling kunnen wij twee systemen volgen. In het eerste systeem geven wij OR een bepaalde tijd om de optelling te verrichten en gaan dan over naar de volgende bewerking.In het twe de systeem laten wij OR zelf een teken geven,dat het klaar is,waarna verder gegaan kan worden.Zulk een teken kan bv.verkregen worden,door te luisteren aan de lijnen,welke de carries transporteren van kolom naar kolom. Als wij daar bv.een microseconde lang niets horen,passeren is het proces klaaren kan het signaal gegeven worden.In het eerste systeem moet de ter beschikking gestelde tijd klaarblijkelijk ruim genoeg bemeten zijn om ook het ongunstigste geval van 40 overdrachten toe te laten.Gemiddeld echter is het proces veel vlugger klaar en het tweede systeem is dus vlugger.Hoeveel vlugger?

Laat $p_n(v)$ de waarschijnlijkheid zijn,dat de grootste carryvoortplantingslengte in de optelling van twee getallen van n-cijfers, groter dan of gelijk aan v is.Dan is de waarschijnlijkheid,dat de grootste lengte precies v is,blijkbaar gelijk aan $p_n(v)-p_n(v+1)$ en de gemiddelde grootste lengte is

$$a_n = \sum_{v=0}^{n} v\{p_n(v)-p_n(v+1)$$

Nu is $p_n(v) = 0$ als $v > n$.en dus $a_n = \sum_{1}^{n} p_n(v)$.

Om nu iets te weten te komen over $p_n(v)$ redeneren we als volgt:Er zijn twee mogelijkheden.

A) De eerste $n-1$ cijfers van de twee getallen opzichzelf produceren al een carry $\geqq v$.Dit heeft een waarschijnlijkheid $p_{n-1}(v)$.

B) Dit is niet het geval.Dan moet het laatste cijfer van beide getallen de carry ontsteken en dit kan alleen als zij beiden 1 zijn (kans ¼). Verder moeten dan ($v-1$)cijfers ervoor steeds de combinatie 1,0 of 0,1 vormen,want noch 0,0,noch 1,1 plant een carry voort.De kans op de combinatie 0,1 of 1,0 is ½ en dus is de totale kans $\frac{1}{4}(\frac{1}{2})^{v+1} = (\frac{1}{2})^{v+1}$

De overblijvende n-v cijfers mogen geen carry $\geqslant$ v bevatten.Dit heeft een kans $1 - p_{n-v}(v)$. Door combinatie van A en B vinden wij: Dus de totale waarschijnlijkheid van geval B is: $[1 - p_{n-v}(v)]/2^{v+1}$

$$p_n(v) = p_{n-1}(v) + [1 - p_{n-v}(v)]/2^{v+1} \quad \text{en } p_n(v) = 0 \text{ voor } v > n$$

dus

$$p_n(v) - p_{n-1}(v) \geqslant 1/2^{v+1}$$

Nu is

$$p_n(v) = \sum_{i=v}^{n} [p_i(v) - p_{i-1}(v)] ,$$

want $p_{v-1}(v) = 0$ In deze som zijn n-v+1 termen en dus is $p_n(v) \leqslant (n-v+1)/2^{v+1}$ Verder is $p_n(v)$ als waarschijnlijkheid $\leqslant$ 1.Derhalve geldt:

$$p_n(v) \leqslant \mathrm{Min}\,[1, (n-v+1)/2^{v+1}$$

ies k zo,dat $2^k \leqslant n \leqslant 2^{k+1}$ Dan is voor v $\geqslant$ k: $(n-v+1)/2^{v+1} \leqslant \frac{n}{2^{v+1}} \leqslant 1$

$$a_n = \sum_{v=1}^{k-1} p_n(v) + \sum_{v=k}^{n} p_n(v) \leqslant \sum_{v=1}^{k-1} 1 + \sum_{v=k}^{n} n/2^{v+1} < k-1 + n/2^k$$

Deze laatste uitdrukking is lineair in n in het interval $2^k \leqslant n \leqslant 2^{k+1}$ en aan het uiteinde $n = 2^k$ is zij =k,terwijl zij aan het andere uiteinde $n=2^{k+1}$ is zij =k+1,d.w.z.zij is $=\log_2 n$ aan beide uiteinden.Omdat de logarithmische functie bol naar boven is,ligt zij altijd bovende koorde. Dus is $a_n < \log_2 n$.Dit geldt voor alle k,dus voor alle n.In het geval n=40,vinden wij op deze wijze $a_{40} < \log_2 40 = 5,3$.Een nauwkeuriger berekening levert $a_{40} = 4.62$.

Wij zien dus dat het tweede systeem van optellen circa vijfmaal zo snel is als het eerste.

Codering van vraagstukken.

In het vorige hebben wij getoond hoe verschillende essentiele functies van een electronische rekenmachine logisch en physisch gerealiseerd kunnen worden.Het gaat buiten het bestek van deze cursus de zoveel ingewikkelder logische schakelingen van de besturing van de machine te analyseren. Wij nemen daarom aan,dat het mogelijk is om de besturing op gewenste wijze te laten reageren op zekere standaardinstructies,gecodeerd als binaire getallen,welke hij vindt in het geheugen.

Hoe moeten dergelijke instructie s er uit zien?Voor zover met de instructie een getal betrokken is,dat moet worden opgeroepen uit het geheugen,dan wel er naar toe moet worden afgevoerd,moet de instructie het adres van dat getal bevatten,dus weer een getal,waarmee het rekenkundig orgaan van de machine ev.nog kan manipuleren of zelfs het geheel en al berekenen op grond van de uitkomst van de berekeningen.Verder moet de instructie nog een of andere bewerking met het getal vermelden,bv. "Neem het getal op adres 384 van het geheugen en plaats het in het RR". Dit zullen wij de 1-adres code noemen.Bij gebruik hiervan moet de besturing zo ingericht zijn,dat na afloop van de uitvoering van de instructie, de instructie welks adres direct volgt op dat van de pas uitgevoerde instructie,gelezen wordt.Afwijking van deze routine kan worden gebracht door een instructie,die luidt:"Ga over naar instructie,te vinden op adres 586".

Er zijn vele andere opvattingen,bv.twee-adrescodes:"Neem het getal op adres 124 en tel het op bij het getal op adres 791",of vier adrescodes:"Neem het getal op adres 935 en dat op adres 334,vermenigvuldig ze met elkaar,breng het product op adres 50 en lees de instructie op adres 881."Wij zullen ons beperken tot de een-adrescode.Ook nu is nog verschil van mening genoeg mogelijk!

Een geraamte van een stel instructies geven wij hieronder. Hierbij herinneren wij eraan, dat de instructies zo klein zijn, dat er twee in ee n woord geborgen kunnen worden. Vandaar: linkse en rechtse instructies.

Nummer.	Afk.	Inhoud.
1.	x	Tel getal op adres x op in het schoongemaakte OR
2.	x-	Trek getal op adres x af van het schoongemaakte OR
3.	xM	Tel modulus van getal op adres x op in het schoongemaakte OR
4.	x-M	Trek modulus van getal op adres x af van het schoongemaakte OR
5.	xh	Als 1 } met vervanging van "schoongemaakt"door "niet schoongemaakt"
6.	xh-	Als 2 }
7.	xhM	Als 3 }
8.	xh-M	Als 4 }
9.	xR	Tel getal op adres x op in het schoongemaakte RR
10.	RO	Schuif het getal uit RR in OR.
11.	OR	Schuif het getal uit OR in RR.
12.	xO	Breng het getal in OR naar adres x.
13.	xR	Breng het getal in RR naar adres x.
14.	xX	Maak OR schoon en vermenigvuldig getal op adres x met getal in RR, daarbij de leidende cijfers in OR en de staart in RR plaatsende.
15.	x÷	Maak RR schoon en deel het getal op adres x op getal in OR, daarbij de rest in OR latende en het quotient in RR brengende.
16.	xC	Verplaats besturing naar instructie op adres x links.
17.	xC'	Verplaats besturing naar instructie op adres x rechts.
18.	xCc	Als getal in OR ≥ 0, dan als 16.
19.	xC'c	Als getal in OR ≥ 0, dan als 17.
20.	xS	Vervang het adres in instructie op adres x links door de corresponderende cijfers van OR.
21.	xS'	Vervang het adres in instructie op adres x rechts door de corresponderende cijfers van OR.
22.	Rn	Schuif de inhoud van OR n plaatsen naar rechts, op de links openvallende plaatsen het begincijfer herhalend.
23.	Ln	Schuif de inhoud van OR na het eerste cijfer en van RR n plaatsen naar links, daarbij de vrijkomende cijfers van OR achter in RR invoerende.

Als voorbeeld van toepassing programmeren wij een worteltrekking, $z = \sqrt{u}$.

Wij passen iteratie toe:

$$z_{i+1} = \tfrac{1}{2}\left(z_i + \frac{u}{z_i}\right) \qquad \lim_{i \to \infty} z_i = \sqrt{u}$$

Ieder getal moet liggen tussen -1 en +1 en in ons geval is dus blijkbaar $0 \leqslant u < 1$ en we willen natuurlijk de positieve wortel,dus alvast $0 \leqslant z < 1$.Nu zien wij in de formule getreden u/z_i en dit is altijd < 1 als wij er maar voor zorgen,dat altijd $z_i > u$.De combinatie $z_i + u/z$ is nu niet altijd < 1,maar wij kunnen schrijven

$$z_{i+1} = \frac{1}{2}\left(\frac{u}{z_i} - z_i\right) + z_i$$

en dan is de moeilijkheid omzeild.Verder is

$$z_{i+1} - \sqrt{u} = \frac{1}{2z_i}\left(z_i - \sqrt{u}\right)^2$$

Dus is $z_{i+1} \geqslant \sqrt{u}$ ~~z~~,dus alle $z_i \geqslant \sqrt{u}$,mogelijk z_0 uitgezonderd. Deze uitzondering kunnen wij uitschakelen door te kiezen $z_0 = 1$,dat is zeker $> \sqrt{u}$ Nu is dus zeker $z_i \geqslant u$ en dus $u/z_i \leqslant 1$,zoals verlangd.Maar ook is $u/z_i \leqslant z_i$, us ~~z~~ $z_{i+1} \leqslant z_i$ Wij hebben dus de reeks:

$$1 = z_0 \geqslant z_1 \geqslant z_2 \geqslant \ldots\ldots \quad \text{en convergerend tot } \sqrt{u}.$$

Nu is het nog slechts de kwestie hoever we door moeten gaan met itereren. Daartoe berekenen wij aan het eind van iedere iteratie $z_{i+1} - z_i = \varepsilon_i$ Uit de iteratieformule volgt dan: $z_{i+1}^2 = u + \varepsilon_i^2$

Verkenwein 40 binalen,dan is het blijkbaar voldoende als $\varepsilon_i = z_{i+1} - z_i < 2^{-20}$ Tenslotte is er nog de moeilijkheid,dat $z_0 = 1$ niet geborgen kan worden. Maar wij kunnen ook even goed $z_1 = \frac{1}{2}(1+u)$ als beginwaarde gebruiken,dat spaart zelfs nog een iteratie!

Wij nemen nu aan,dat U zich bevindt op adres x.Nu nemen wij een veertiental ordersin ons geheugen,die wij maar zullen nummeren van 0 tot en met 13.Wij brengen x naar plaats 8 en dan zien de orders er als volgt uit:

0	8 ,	R	7	13	Cc ,	2	C
1	10 h ,	9S	8		u		
2	8 ,	9÷	9		-		
3	RO,	9h-	10		2^{-1}		
4	R ,	11S	11		-		
5	9 h ,	9S	12		2^{-20}		
6	11 ,	12h	13	e	C ,		

Dan wordt op plaats 9 vanzelf na enige tijd $\sqrt{u}$ afgeleverd.De afleiding van dit resultaat laten wij aan den lezer over.

Tenslotte wijzen wij er nog op,dat er één enkele instructie is,welke ongemeen veel algemener en practischer is dan de tot nu toe hier behandelde.Deze luidt in de drie-adrescode:"Neem Uw omvangrijke rekenproblemen van Uw eigen adres en stuur ze aan het adres van het M.C.(Wyttenbachstraat 5,Amsterdam),welke de antwoorden weer aan Uw adres retourneert!"