elektronische rekenautomaten

WIJ EN DE ELECTRONICA

9

Elektronische rekenmachines

Met de naam „rekenmachine" kan men het oog hebben op twee verschillende groepen van instrumenten. Dit zijn enerzijds de digitale of cijferende machines, anderzijds de analogonmachines. In dit boekje zal de eerste soort uitvoerig worden besproken.

Analoog en digitaal

Het onderscheid tussen de begrippen *analoog* en *digitaal* kan toegelicht worden aan minder gecompliceerde apparaten, b.v. aan meetinstrumenten. Tijd wordt gemeten met een horloge, de snelheid van een auto met een snelheidsmeter. In beide gevallen moet de stand van een wijzer op een schaalverdeling worden afgelezen. Bij de snelheidsmeter wordt elke mogelijke snelheid, b.v. 60 km/h, 60,5 km/h of 60,53 km/h op de schaal voorgesteld door een bepaalde plaats. Grotere snelheden staan aan de ene kant daarvan, kleinere aan de andere zijde. De naald wijst, aangenomen dat de meter goed g ijkt is, de werkelijke snelheid aan.
Men noemt een horloge, snelheidsmeter, doch ook een rekenlineaal een *analogoninstrument* (fig. 1). Het typische ervan is, dat alle mogelijke waarden aangewezen kunnen worden, maar dat aan de aflezing steeds een zekere onnauwkeurigheid is verbonden.
Tegenwoordig heeft men ook klokken die de juiste tijd, b.v. in minuten nauwkeurig, aangeven niet met een wijzer, doch met getallen. Elke minuut verspringt er één of meer cijfers. Ook de kilometerteller in de auto geeft de afgelegde afstand in km met cijfers aan. Onderdelen van een km, b.v. decameters en meters, laat staan centimeters, worden niet aangewezen. Het aflezen is een echt „lezen".
Van onduidelijkheid is geen sprake — behalve op het moment dat een cijfer verspringt. Dergelijke met cijfers (digits) werkende toestellen noemt men *digitale instrumenten* (fig. 2). Ook een telraam hoort in deze groep thuis.

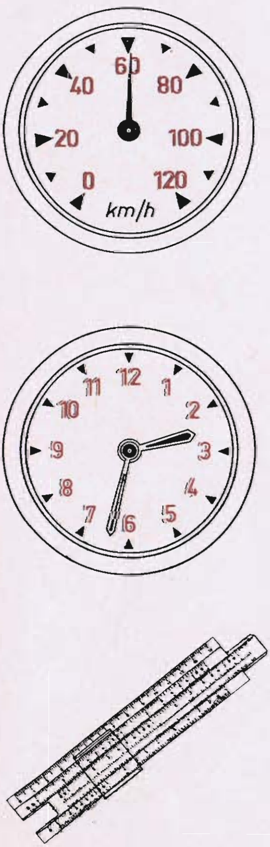


Fig. 1. Analogon-instrumenten

De analogon-rekenmachine

Deze machines zijn omstreeks 1930 voortgekomen uit de meetkundige instrumenten voor het bepalen van oppervlakken van vlakke figuren (planimeters) en werkten aanvankelijk geheel mechanisch. Moest b.v. een vraagstuk over warmtegeleiding door een wand heen met zo'n machine worden

opgelost, dan kwam de waarde van de temperatuur overeen met de hoekverdraaiing van bepaalde assen. Wijzers op deze assen maakten aflezing van de stand mogelijk. De assen in de machine waren door tandwielen en zgn. integratoren zodanig aan elkaar gekoppeld, dat hun beweging beheerst werd door dezelfde wetten als golden voor het temperatuurverloop in het warmtegeleidingsvraagstuk. De koppeling van de assen moest daartoe voor ieder nieuw probleem anders worden uitgevoerd. Op deze wijze vormde de machine een soort mechanisch *analogon* van het werkelijke vraagstuk.
De nauwkeurigheid van de antwoorden — meestal in de vorm van grafieken rechtstreeks door de bewegende assen opgetekend — werd hierbij geheel bepaald door de nauwkeurigheid waarmee men de hoekverdraaiing van de assen, o.a. in verband met speling in de tandraderen en dode gang, kon aflezen. Deze precisie kon nauwelijks beter dan 1% worden gemaakt.
Kort voor de tweede wereldoorlog is men elektrische machines van deze soort gaan bouwen. In plaats van de verdraaiing van assen gebruikte men nu de grootte van elektrische spanningen. Dank zij de ervaringen met de elektronentechniek is het mogelijk om met hoge nauwkeurigheid elektrische spanningen op te tellen, zodat de nauwkeurigheid van deze machines zeker met meer dan een factor tien is verbeterd. Desondanks is deze in vele gevallen nog veel te klein. In dit opzicht zijn de digitale machines hun verre de baas. Anderzijds bestaan er soorten vraagstukken, waarbij de analogonmachine sneller het antwoord gereed heeft. Dieper willen wij op dit soort machine echter niet ingaan.

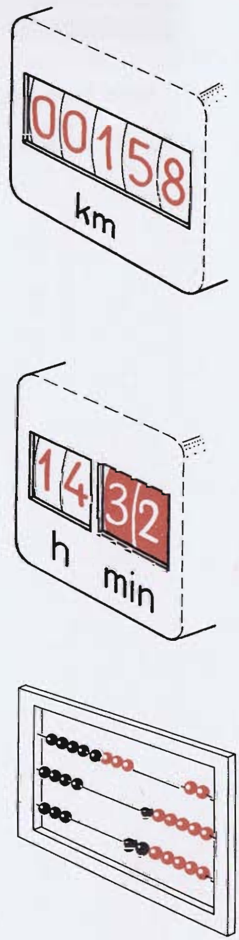
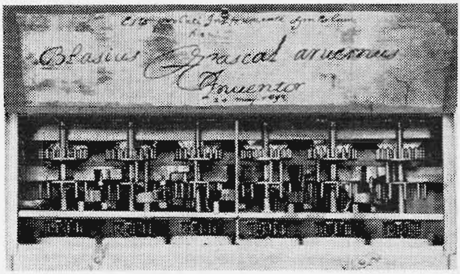


Fig. 2. Digitale instrumenten

De digitale rekenmachine

De digitale rekenmachines werken cijferend, d.w.z. zij bevatten *registers* welke getallen kunnen bevatten; optellingen, vermenigvuldigingen, enz. worden hier in principe cijfer voor cijfer uitgevoerd. Hieruit volgt direct dat de nauwkeurigheid van de uitkomsten zo groot is als met de registerlengte (d.w.z. het mogelijke aantal cijfers van de getallen) overeenkomt en dus gemakkelijk beter dan b.v. 0,1% gemaakt kan worden. Tot deze machines behoren b.v. de tafelrekenmachines en de kasregisters in winkels, machines, waarvan de eerste eenvoudige exemplaren reeds door *Schickard* (1623) en *Pascal* (1642) zijn uitgevonden (fig. 3). Bij deze, zowel als bij de



British Crown Copyright, Science Museum, London

Fig. 3. De rekenmachine van Blaise Pascal

huidige machines worden getallen met de hand in de machine gebracht, b.v. door het aanslaan van toetsen. Door het indrukken van een commandotoets kan men een zojuist ingebracht getal b.v. bij een reeds aanwezig getal optellen of het ermee vermenigvuldigen.

Meestal worden bij deze machines de cijfers weergegeven door de stand van tandwielen met tien tanden en wordt de optelling uitgevoerd door een cijferwiel van het éne getal zoveel tanden te verdraaien als de stand van het overeenkomstige cijferwiel van het andere getal aangeeft. Een moeilijkheid bij de constructie is het doorgeven van de *overdrachten* : als de som van twee cijfers boven de 10 komt, moet op de volgende, hogere cijferplaats het wiel één tand verder draaien (denk aan : $5 + 8 = 3$ en één onthouden !).

Omstreeks 1940 is men begonnen digitale rekenmachines elektrisch i.p.v. mechanisch te laten werken. Eerst deed men dit met *relais,* d.w.z. elektro-magnetische schakelaars, waarvan het gebruik uit de automatische telefooncentrales reeds goed bekend was en ook met elektronenbuizen. Om de mechanische telwielen met hun tien standen na te bootsen maakte men zgn. ringtellers met tien standen, die door het toevoeren van een elektrische puls (stroomstoot) aan de ingangsklemmen, van de ene stand in de volgende werden gebracht en waarbij dan op de tiende stand weer de eerste moest volgen.

Daar in relais altijd bewegende delen optreden, kan de schakelsnelheid ervan niet erg hoog zijn. Ongeveer 500 pulsen per seconde is wel een uiterste. Elektronenbuizen kunnen echter gemakkelijk 1000 maal sneller werken. Maar de eerste zo gebouwde machine, de ENIAC in de U.S.A., had voor elke ringteller, d.w.z. voor elke cijferplaats in één getalregister, liefst 20 elektronenbuizen nodig! Van de 18.000 buizen in dit apparaat dienden er alleen reeds 4.000 voor de 20 registers elk met 10 cijfers. De tijd nodig om de som van twee getallen te vormen bedroeg 200 μsec ! (1 μsec = 1 microseconde = 0,000.001 sec). Kort voor óns, maar voor de praktijk nog erg lang ! Andere elektronische schakelingen die tien verschillende standen kunnen aannemen, zijn later

nog wel overwogen, maar ook deze hadden het bezwaar van te kleine snelheid of te grote ingewikkeldheid.
De oorzaak van deze gecompliceerdheid is daarin gelegen, dat men over „elementen" moet beschikken, die zich elk successievelijk in tien verschillende toestanden moeten kunnen bevinden, om zodoende de cijfers 0 t/m 9 te kunnen aanduiden. Zit men nu aan die tien, dus aan het *tientallig stelsel* vast voor het met getallen aanduiden van aantallen ? Is dit stelsel het enig mogelijke ?
De wiskunde heeft daarop reeds lang geleden antwoord gegeven. Op elk getal is een talstelsel op te bouwen : op 12 (terug te vinden in het dozijn en het gros), op het getal 60 (denk aan de onderverdeling van een booggraad of van een uur in minuten en seconden), doch ook op het getal twee. Men heeft dan slechts twee symbolen nodig, b.v. 0 en 1, terwijl de „elementen" in de op dit *tweetallig stelsel* gebaseerde rekenmachines slechts twee standen behoeven te kennen. De overweging dat een elektronenbuis en ook zijn modernere plaatsvervangers, diode en transistor, eigenlijk bij uitstek twee toestanden kennen, nl. *stroomgeleidend* en *niet-geleidend,* heeft veroorzaakt dat veel machines voor het rekenen nu dit tweetallig stelsel gebruiken. Wij zullen later zien hoe dit in zijn werk gaat en hoe dit talstelsel is ingericht.

Rekenprogramma's

Eerst willen wij echter aandacht schenken aan een gedachte die van groot belang is geweest voor de ontwikkeling van de moderne rekenmachine. Dit voert ons terug naar de Engelsman *Charles Babbage* (1792-1871). In het voorgaande is gesproken over een tafelrekenmachine waarbij men getallen met de hand aanslaat, een bewerkingstoets indrukt en het ontstane resultaat afleest en overschrijft op een stuk papier. Wil men met zulk een machine lange berekeningen maken, dan moet na elke bewerking de rekenaar de nieuwe getallen aanslaan, de nieuwe bewerkingstoets indrukken en het resultaat weer op papier overnemen. Zo wordt het gehele *rekenprogramma* afgewerkt. Babbage doorzag nu dat het gehele rekenwerk automatisch door een machine gedaan zou kunnen worden, ook bij de ingewikkeldste problemen. De enige voorwaarde daarvoor was dat de oplosmethode, en daarmede dus het programma van de berekeningen, van te voren bekend moest zijn.
Laten wij als — eigenlijk véél te eenvoudig — voorbeeld eens nemen de vraag naar de wortels x_1 en x_2 van de vier-

kantsvergelijking $ax^2 + bx + c = 0$. Hierin stellen de coëfficiënten a, b en c bepaalde, overigens willekeurige getallen voor. Het is bekend dat de wortels gevonden kunnen worden uit :

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{en} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

De berekening loopt via een aantal tussenuitkomsten stapsgewijs als volgt :

1. Bereken $U_1 = b \times b$	6. Bereken $U_6 = -b + U_5$
2. Bereken $U_2 = a \times c$	7. Bereken $U_7 = -b - U_5$
3. Bereken $U_3 = 4 \times U_2$	8. Bereken $U_8 = 2a$
4. Bereken $U_4 = U_1 - U_3$	9. Bereken $x_1 = U_6 : U_8$
5. Bereken $U_5 = \sqrt{U_4}$	10. Bereken $x_2 = U_7 : U_8$

Laat ons nu nagaan waaraan een machine moet voldoen die dit lijstje van tien rekenstappen, het rekenprogramma, automatisch achter elkaar kan afwerken.

De machine moet een orgaan bevatten dat de bewerkingen : optellen, aftrekken, vermenigvuldigen, delen en worteltrekken (vaak een heel programma van eenvoudige berekeningen) kan uitvoeren, het zgn. *rekenorgaan.*

Dit is niet verwonderlijk; belangrijker is dat de tussenresultaten U_1 t/m U_8 tijdelijk bewaard moeten kunnen blijven. Er moeten dus een aantal *geheugenregisters* zijn waarin de, als tussenresultaten uit het rekenorgaan verkregen getallen kunnen worden vastgelegd. Bovendien moet de inhoud van deze registers bij volgende bewerkingen weer naar het rekenorgaan kunnen worden overgebracht. Het geheugen moet dus *beschrijfbaar* en *leesbaar* zijn; er moeten getallen kunnen worden ingeschreven en worden uitgelezen. Ook de coëfficiënten a, b en c van de gegeven vergelijking moeten tevoren in dat geheugen aanwezig zijn. Elk der geheugenregisters waaruit het geheugen is opgebouwd zal een nummer moeten hebben (het zgn. *adres* van dat register) om het van andere te kunnen onderscheiden.

Een heel belangrijk punt is echter dat ook de tien *rekenopdrachten* tevoren aan de machine bekend moeten zijn,

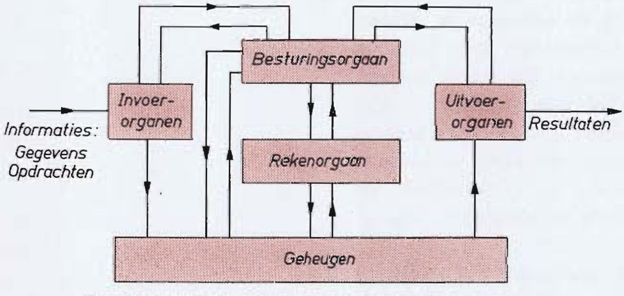


Fig. 4. Schema van een rekenautomaat

wil deze automatisch de ene opdracht na de andere kunnen afwerken. Dit programma moet dus eveneens in een (liefst in hetzelfde) geheugen zijn vastgelegd. Omdat de geheugenregisters slechts getallen kunnen bevatten, moeten de opdrachten in getalvorm zijn gecodeerd. Dit kan op allerlei wijzen geschieden, b.v. volgens de *drie-adrescode*. Hierbij bestaat de code voor een opdracht uit vier stukken, die resp. aangeven :

soort bewerking	1e adres	2e adres	3e adres

De soort bewerking kan in getalvorm worden aangegeven volgens een afgesproken code, b.v. optellen = 1, aftrekken = 2, vermenigvuldigen = 3, enz. Aannemende dat adres no. 1 het getal a bevat en adres no. 3 het getal c, zal de opdracht voor de tweede stap in het als voorbeeld gegeven programma nu in code kunnen luiden : | 3 | 1 | 3 | 7 | hetgeen betekent : „Vermenigvuldig het getal a, uit te lezen op adres no. 1 van het registergeheugen met het getal c, uit te lezen op adres no. 3 en schrijf de uitkomst in op adres no. 7 (en ga voort met de volgende opdracht)".

Het opslaan van het programma in het geheugen is echter nog niet voldoende. Er is nog nodig een *besturingsorgaan* dat de opdrachten in de juiste volgorde uit het geheugen haalt, hun betekenis ontrafelt, de betrokken getallen uit het geheugen opzoekt en overbrengt naar het rekenorgaan, de goede bewerking erop doet plaatshebben, het resultaat op de door het derde adres aangegeven plaats opbergt en nadat dit alles klaar is, de volgende opdracht uit het geheugen haalt en daarmee op geheel gelijke wijze handelt.

Het is merkwaardig dat deze hele gedachtengang in principe reeds omstreeks 1830 door Babbage was gevonden en dat deze daarvoor een mechanische oplossing wist aan te geven. Een gedeelte van zijn machine is ook werkelijk geconstrueerd. Door de, voor die tijd te hoge eisen die aan de afwerking moesten worden gesteld en door gebrek aan middelen is de gehele „Analytical Engine" nooit klaar gekomen.

Conditionele sprongopdrachten

In ons voorbeeld zullen wij nog één verandering aanbrengen, aan de noodzaak waarvan ook Babbage reeds gedacht had. De berekening heeft in de gegeven vorm alleen zin als de uitdrukking $b^2 - 4ac$ positief is. De wortel uit een negatief getal is nl. niet bestaanbaar. De berekening is dan niet

mogelijk en de rekenmachine zal, hetzij direct moeten stoppen, hetzij moeten overgaan op een nieuw programma dat nog in het geheugen wacht.
Men moet de machine dus een opdracht geven tot het afwijken van de normale volgorde van het rekenprogramma na de eerste vier stappen. Deze afwijk- of sprongopdracht geldt echter alleen onder de voorwaarde (conditie) dat $U_4 = b^2 - 4ac$ negatief is, of dat $b^2 < 4ac$, dus dat $U_1 < U_3$.
Men noemt dit een *conditionele sprongopdracht*. In de „drie-adrescode" luidt deze als volgt :

conditionele sprong	1e adres	2e adres	3e adres

hetgeen betekent :
„Als het getal U_1, aanwezig op het 1e adres, kleiner is dan het getal U_3 aanwezig op het 2e adres, voer dan de opdracht uit, aanwezig op het 3e adres. (Is dit niet het geval, ga dan, zoals vanzelf spreekt, verder met het normale programma)".
Met dergelijke conditionele sprongopdrachten kan men zorgen dat de machine tijdens het werken, op grond van verkregen resultaten beslissingen neemt die vooraf, bij het maken van het programma, nog niet waren te overzien.
Het blijkt juist deze mogelijkheid te zijn die de elektronische rekenmachines hun grote veelzijdigheid geeft en ze werkelijk tot rekenautomaten maakt die lange en ingewikkelde programma's zonder tussenkomst van een operateur uitvoeren. Wij zullen daarom voor dit soort machines voortaan de naam *rekenautomaten* (Engels : computers) gebruiken, in onderscheid met de naam *rekenmachines* voor de niet-automatische apparaten.
In het hierboven gegeven voorbeeld moet het geheugen bevatten :

1. de getallen a, b en c; hiervoor zijn drie adressen nodig.
2. de constanten 4 en 2, voorkomende in de opdrachtstappen no. 3 en 8.
3. ruimte voor de tussenuitkomsten U_1 t/m U_8. Daar deze niet alle tot het einde bewaard behoeven te worden, kan men voor verschillende hiervan na elkaar éénzelfde adres gebruiken. Drie zijn er in het gegeven programma nodig.
4. de elf opdrachten, inclusief de sprongopdracht, van het programma.

In totaal moeten dus 19 adressen ter beschikking staan.
Wij bespraken hier maar een zeer bescheiden programma; voor ingewikkelde berekeningen zijn programma's die b.v. een paar duizend geheugenadressen in beslag nemen, geen zeldzaamheid.

Invoer- en uitvoerorganen

In het voorgaande maakten wij reeds kennis met drie hoofdonderdelen van iedere rekenautomaat te weten : het rekenorgaan, het geheugen en het besturingsorgaan.
Hieraan moeten nu nog worden toegevoegd de *organen voor in- en uitvoer.* Om met het laatste te beginnen, natuurlijk zal het nodig zijn om de antwoorden in leesbaar schrift afgedrukt te krijgen. De meeste installaties zijn daartoe uitgerust met een door de automaat bediende schrijfmachine, maar vaak ook met een zgn. *regeldrukker.* Hiermede kunnen hele regels van vaak ca. 120 tekens met een snelheid tot 15 regels per seconde worden afgedrukt. Zelfs deze enorme snelheid is nauwelijks voldoende om het rekenapparaat bij te houden.
In sommige gevallen is het gewenst de antwoorden vast te leggen in *ponskaarten* of *ponsband.* Ponskaarten worden reeds lang in de gemechaniseerde administratie van grote bedrijven en instellingen gebruikt. Het zijn kaarten waarin een groot aantal rechthoekige gaatjes op schijnbaar willekeurige plaatsen geponst is. In een kaart kunnen maximaal 80 tekens voorkomen. Ook de lidmaatschapkaarten van de A.N.W.B. zijn zo uitgevoerd. Ponsband is een lange reep papier waarop naast elkaar vijf of meer rijen gaatjes voorkomen. Met het aanwezig zijn of ontbreken van een gat op een bepaalde plaats wordt een gegeven in een tweetallige codevorm vastgelegd. Ook telex-apparaten maken hiervan gebruik.

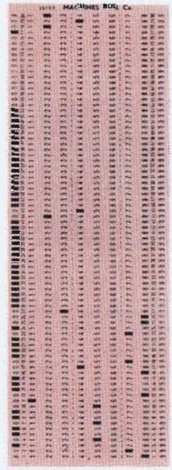
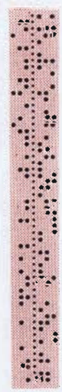


Fig. 5. Een ponskaart en ponsband



Ponskaarten en ponsbanden kunnen in heel snel tempo door het *uitvoer-orgaan* worden voorzien van de informatiedragende gaatjes; ponsband b.v. met 300 tekens per sec. Doch zij zijn machinaal ook heel snel te lezen, b.v. ca. 1000 tekens per sec uit ponsband of 10 kaarten per sec bij ponskaartlezers. Daarom worden ze ook gebruikt als invoermiddel voor de automaat. Via het *invoerorgaan* kan het rekenprogramma alsmede de bijkomende getallen vanaf de kaarten en banden in het geheugen worden ingebracht. De mens is daar veel te traag voor; de machine zou veel te lang werkeloos staan.
Om de tekens door het invoerorgaan te laten lezen en op de goede manier met elkaar gecombineerd in het geheugen te zetten, is nog een *invoerprogramma* nodig, dat steeds in de machine aanwezig dient te zijn.
Door middel van knoppen op de bedieningslessenaar wordt eerst het invoerprogramma in werking gezet waarna met het ingevoerde programma zelf begonnen wordt.

Het tweetallig of binair stelsel

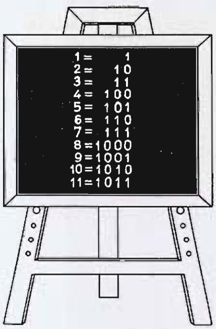


Fig. 6. Decimale en binaire getallen

$$\begin{array}{r} 6 \\ 3\,+ \\ \hline 9 \end{array} \qquad \begin{array}{r} 110 \\ 11\,+ \\ \hline 1001 \end{array}$$

$$\begin{array}{r} 6 \\ 3\,\times \\ \hline 18 \end{array} \qquad \begin{array}{r} 110 \\ 11\,\times \\ \hline 110 \\ 110 \\ \hline 10010 \end{array}$$

Fig. 7. Voorbeeld van enkele berekeningen

Wij komen nu terug op de opmerking dat het tweetallig stelsel voor een elektronische rekenautomaat zoveel voordelen biedt. Daarom eerst iets over dit talstelsel zelf.

Gewoonlijk zijn wij gewend te rekenen en te tellen in het tientallig stelsel. Dit betekent dat wij aantallen splitsen in *eenheden, tientallen, honderdtallen* enz. en voorzover het over breuken gaat in *tienden, honderdsten, duizendsten,* enz. Alle zijn dit machten van 10. Als voorbeeld van een aantal, dat in getalvorm moet worden geschreven, nemen we het aantal dagen in een (normaal) jaar. Hiervan splitsen we eerst zoveel mogelijk honderdtallen H af, dan zoveel mogelijk tientallen T, zodat dan een aantal eenheden E overblijft. We krijgen dan $3 \times H + 6 \times T + 5 \times E$, waarin dus $H = 100 = 10^2$, $T = 10 = 10^1$ en $E = 1 = 10^0$.

Deze schrijfwijze is onpraktisch. Men is gewend de getallen *3, 6* en *5* naast elkaar te schrijven : *365*. Daarbij is afgesproken, dat de plaats van het cijfer in het getal aangeeft of men te maken heeft met eenheden, tientallen, honderdtallen, enz., daarbij rechts beginnende. Het is zodoende duidelijk dat men met 365 niet bedoelt $3 \times 6 \times 5$. Men noemt dit de *positionele schrijfwijze* van getallen.

In het tientallig stelsel heeft men daarbij genoeg aan de tien symbolen 0, 1, 2, 3, 4, 5, 6, 7, 8 en 9. Voor „twaalf" b.v. heeft men geen symbool nodig, daar hiervan een tiental af te splitsen is, waarna er twee eenheden overblijven. Twaalf wordt dus positioneel in het tientallig stelsel geschreven als $(1 \times 10 + 2 =)$ 12.

In het tweetallig stelsel verdeelt men een aantal in eenheden, tweetallen, viertallen, achttallen enz. en, voor zover het breuken betreft in halven, vierden, achtsten, enz. Alle zijn dit machten van het grondtal 2. Om het aantal dagen in een jaar in het tweetallig stelsel te schrijven splitst men eerst zoveel mogelijk $2^8 = 256$-tallen af, dan de $2^7 = 128$-tallen, vervolgens de $2^6 = 64$-tallen enz., tot de $2^2 = 4$-tallen, $2^1 = 2$-tallen en eenheden (2^0) toe. Men krijgt dan :

$$\begin{array}{c} 1 \times 256 + 0 \times 128 + 1 \times 64 + 1 \times 32 + 0 \times 16 \\ + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = \\ 1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 \\ + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \end{array}$$

Dit kan weer vereenvoudigd worden met de positionele schrijfwijze tot *101101101*.

Dit getal stelt dus in het tweetallig stelsel het aantal dagen

voor in een jaar; voor een schrikkeljaar wordt dat *101101110*.
In dit binair stelsel heeft men genoeg aan twee symbolen *0* en *1*. Voor het aantal „drie" is geen symbool nodig, omdat hiervan een tweetal afgesplitst kan worden, waarna een eenheid overblijft. Men krijgt dus : drie = *11*.
Berekeningen in het tweetallig stelsel gaan daarom zo gemakkelijk, omdat de tafels van optelling en vermenigvuldiging zich beperken tot : $0 + 1 = 1$, $1 + 1 = 10$, $0 \times 1 = 0$ en $1 \times 1 = 1$. (Zie fig. 7).
Een nadeel van dit stelsel is dat de getallen langer zijn dan in het tientallig stelsel.
Dit weegt wat de rekenmachines betreft echter ruimschoots op tegen het grote voordeel dat slechts twee symbolen nodig zijn.
De elektronische schakelingen behoeven slechts twee toestanden te kennen en hiervoor komen bij uitstek in aanmerking : „geleidend zijn" en „niet-geleidend zijn" !

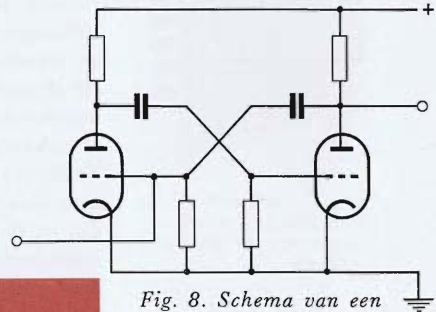


Fig. 8. Schema van een flip-flop schakeling

Elektronische binaire elementen

Op elke cijferplaats van een tweetallig (binair) geschreven getal kan alleen òf een 0, of een 1 voorkomen. Om nu elektronische registers te maken die tweetallige getallen kunnen bewaren, heeft men voor elke cijferplaats een schakelelement nodig dat zich slechts in twee toestanden kan bevinden, welke men dan resp. de 0-toestand en de 1-toestand kan noemen.
Een belangrijke schakeling van deze soort is de zgn. *flip-flop-schakeling* die in principe uit twee buizen of transistors bestaat. Deze zijn zo geschakeld dat steeds één van beide in de geleidende toestand kan verkeren, óf de ene, óf de andere. Door een elektrische puls gaat de ene toestand in de andere over. Dit gedrag lijkt veel op een wip uit de kinderspeeltuin; hij staat óf in de ene, óf in de andere stand en kan door een stoot worden omgezet.
De registers in het *rekenorgaan*, betrekkelijk klein in aantal, worden vaak met flip-flops ingericht. Zou men de registers van het *geheugen* ook zo uitvoeren, dan krijgt men echter ondoenlijk veel buizen of halfgeleiders. Hiervoor gebruikt men daarom andere oplossingen, meestal berustend op magnetisatie van hard-magnetische stoffen.
Een ander schakelelement dat zich in twee toestanden kan bevinden is de *magneetring*, vaak ook *ringkerntje* genoemd. Deze zijn gemaakt van *ferriet*, een kunststof met hardmag-

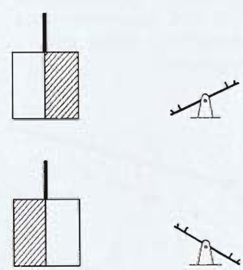


Fig. 9. Symbool voor een flip-flop schakeling

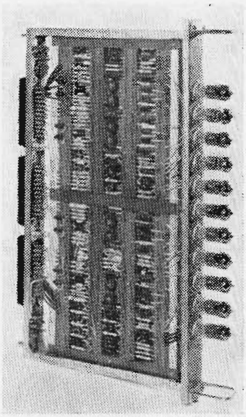


Fig. 10. Een der 42 delen van het reken-orgaan van de Philips PASCAL

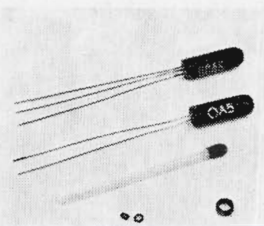


Fig. 11. Enkele elektronische componenten van een rekenautomaat: transistor, diode en ringkerntje

netische eigenschappen (ook Ferroxdure en Ferroxcube zijn ferrieten; zie pag. 11 van het deeltje 7 van deze serie). De diameter van deze ringen is slechts 2 mm of nog kleiner. Door deze ringetjes lopen koperdraden. Elektrische stromen hierdoor kunnen met hun magnetisch veld de ringetjes in een richting langs hun cirkelomtrek magnetiseren.

Een enkele stroomdraad heeft om zich heen een magnetisch veld met gesloten, cirkelvormige veldlijnen, waarvan de richting bepaald wordt door de stroomrichting. Deze twee richtingen blijken bij elkaar te horen net als de roterende en voortgaande richting bij een kurketrekker (zie fig. 12).

De elementaire magneetjes in de ringkerntjes zullen met hun noordpooltje in de richting der veldlijnen gaan staan, dus óf linksom, óf rechtsom, afhankelijk van de stroomrichting.

Verdwijnt de stroom, dan blijft deze toestand behouden; de kerntjes zijn immers van hardmagnetische stof gemaakt. Men kan nu de elementaire magneetjes in de andere richting doen omklappen, door een stroom in de andere richting... mits deze echter een bepaalde sterkte heeft, die wij hier gemakshalve op 1 ampère stellen. Zwakkere stromen hebben bij dit soort ferriet geen enkele invloed.

De magneetring kan dus in twee stabiele toestanden voorkomen, óf linksom gemagnetiseerd, óf rechtsom. Door hun kleine afmetingen en lage prijs zijn zij zeer geschikt om er het grote aantal geheugenregisters mee te maken dat in een rekenmachine wenselijk is.

Wij willen er hier reeds op wijzen, dat de stroom van 1 A, benodigd om de ringkerntjes te doen omklappen, ook verdeeld kan worden in stromen van elk ½ A door twee verschillende draden, b.v. een horizontale en een vertikale (zie fig. 19). Men moet er daarbij voor zorgen, dat beide stromen het ringetje aan dezelfde kant binnengaan, zodat hun wer-

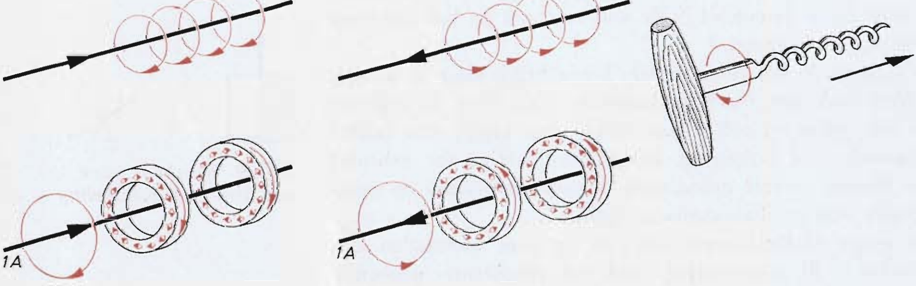


Fig. 12. Het verband tussen stroomrichting en veldrichting

kingen bij elkaar opgeteld kunnen worden. Wij zullen spoedig zien, hoe men van deze kunstgreep gebruik maakt.

Geheugens

Figuur 14 toont ons een der vele identieke delen van een magneetringgeheugen; figuur 15 is een vergrote afbeelding van een klein deel daarvan. De ringkerntjes zijn geregen in een matje van koperdraden. Per matje komen b.v. 32 × 32 ringetjes voor, terwijl een geheel geheugen uit vele matjes bestaat en zelfs wel een miljoen ringetjes kan bevatten. Er is daarin dus plaats voor het vastleggen van miljoen binaire cijfers, ook wel *bits* (*bi*naire digi*ts*) genoemd.
In grote installaties is vaak een ringengeheugen van miljoen bits nog niet groot genoeg. Men neemt dan zijn toevlucht tot magnetische trommels of magnetische bandgeheugens, waarvan de werking gelijkt op de magnetische geluidsregistratie.
In deeltje 7 van deze serie hebben we reeds gezien, hoe men geluid magnetisch kan vastleggen. Het is niet verwonderlijk, dat *magnetische band* ook gebruikt wordt als geheugen voor veel minder gecompliceerde signalen. Ook hier laat men de band lopen langs een magneetkop. Afhankelijk van de stroomrichting in de spoel van de kop kan men op de band *na* en dus *naast* elkaar kleine gebiedjes krijgen met wisselende magnetisatierichting. Een gebiedje waarin de noordpooltjes van de ijzerdeeltjes naar boven gericht zijn stelt b.v. het symbool 0 voor, terwijl een 1 voorgesteld wordt door een stukje met naar beneden gerichte noordpooltjes. Per symbool blijkt slechts een bandlengte van 0,1 mm nodig te zijn!

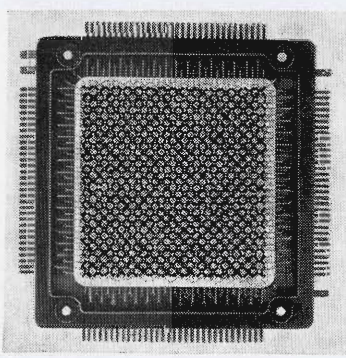


Fig. 13. Deel van een magneetringgeheugen

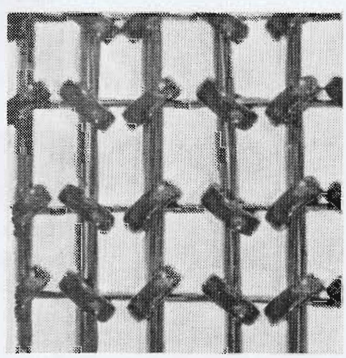


Fig. 14. Detail van het matje

Past men brede band toe (b.v. 25 mm) dan is er voldoende ruimte om 16 magneetkoppen naast elkaar elk hun eigen spoor te laten schrijven. Eén meter band bevat zodoende liefst 160.000 binaire symbolen, overeenkomende met ca. 50.000 decimale cijfers !
De magnetische band vormt dus een geheugen met zeer grote capaciteit; men gebruikt zelfs banden tot 1000 m lengte. Een groot bezwaar is echter, dat voor het opzoeken van een bepaald gegeven de gehele band afgespeeld moet worden en dat kost veel tijd.
Sneller bereikbaar zijn de gegevens als ze geregistreerd worden op *magnetische trommels* (fig. 15) of *schijven*. Deze draaien b.v. 100 keer per sec rond, zodat alle gegevens, vastgelegd in een groot aantal naast elkaar gelegen sporen, door even zo vele magneetkoppen binnen 0,01 sec kunnen worden uitgelezen. Geheugens uitgerust met trommels of schijven zijn dus sneller uitleesbaar dan die voorzien van band; hun capaciteit is echter veel kleiner.

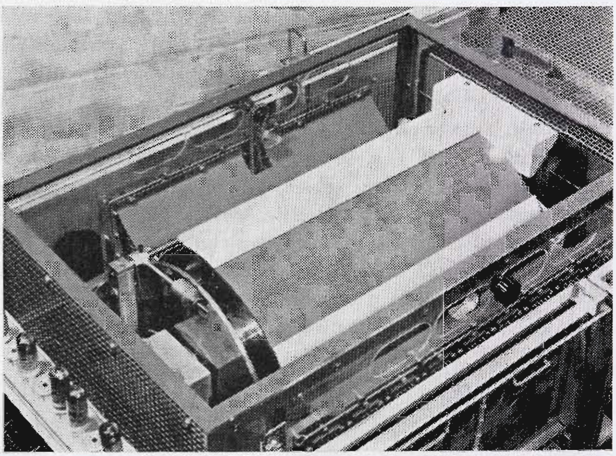


Fig. 15. Trommelgeheugen van de PASCAL

Het inschrijven en uitlezen van gegevens

Hoe kan men nu in een rij (van tenminste vijf) ringen b.v. het getal 22 in binaire vorm vastleggen ? Eerst stuurt men dan door de gemeenschappelijke horizontale draad (zie fig. 16) een stroom van tenminste 1 A naar rechts. Hiermede wordt als het ware het bord schoongeveegd : alle ringetjes komen in een stand die we 0 noemen.

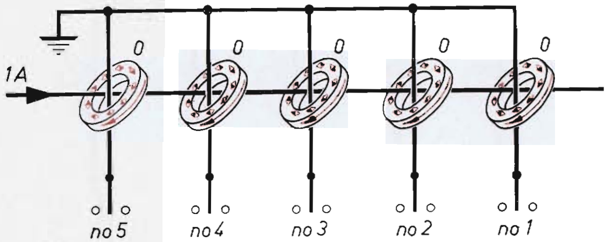


Fig. 16. Het schoonvegen van een ringkerngeheugen

Vervolgens zendt men door deze draad een stroom van $\frac{1}{2}$ A, nu naar links. Dit heeft, zoals we reeds bespraken, geen gevolgen. Bovendien laat men nu echter door de vertikale draden van het 2e, 3e en 5e ringetje (van rechts af gerekend) ook een stroom gaan van $\frac{1}{2}$ A en wel omhooggaande (fig. 17). Door deze ringetjes gaat dus in totaal 1 A; deze — en alléén deze — zullen dus omklappen in de stand 1. Het doel is bereikt, er staat nu 10110 !

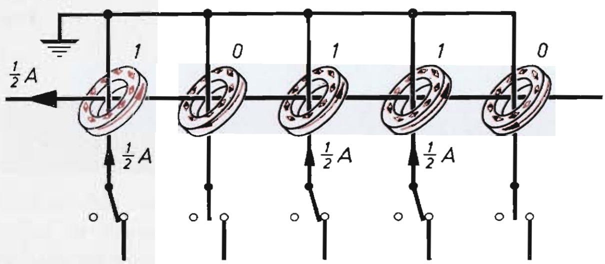


Fig. 17. Het inschrijven van een getal in een ringkerngeheugen

Aan de ringetjes zelf is natuurlijk niet te zien in welke magnetische stand zij verkeren. Hoe kan men nu het geregistreerde getal uitlezen en bv. overbrengen naar de flip-flop-schakelingen van het rekenorgaan ?

De vertikale draden worden daartoe elk met zo'n flip-flop verbonden, die eerste alle in de nulstand gebracht zijn (fig. 18). Men zendt nu weer een stroom van tenminste 1 A naar rechts door de gemeenschappelijke horizontale draad, zodat alle ringetjes weer in de 0-stand komen. Drie ervan zijn dus omgeklapt. Bij dit omklappen zullen in de betreffende vertikale draden door inductie (zie pag. 4 in deeltje 7) korte stroomstootjes optreden die de flip-flops no. 2, 3 en 5 — en alléén deze — omzetten in de 1-stand. Daar staat dus nu 10110.

Opgemerkt moet worden, dat het uitlezen van een ringgeheugen de ingeschreven gegevens vernietigt. Op de voor-

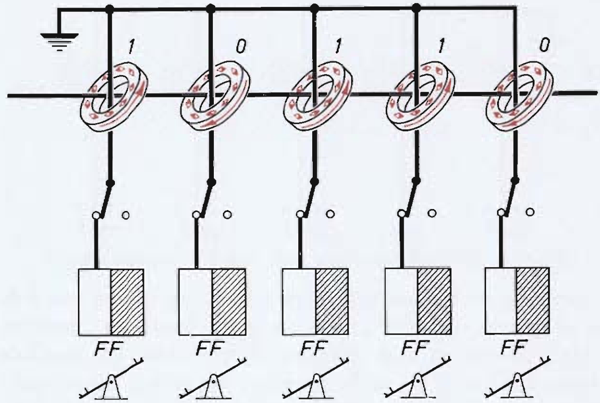


Fig. 18. Schakeling voor het uitlezen van het geheugen

zieningen die daarom nodig zijn om de uitgelezen gegevens weer automatisch in het geheugen terug te schrijven, gaan wij echter niet in.

Elektronisch cijferen

Wij willen nu in het kort aanduiden hoe met elektronische middelen het eigenlijke *cijferwerk* gedaan wordt en zullen ons daarbij beperken tot het *optellen* van twee getallen in

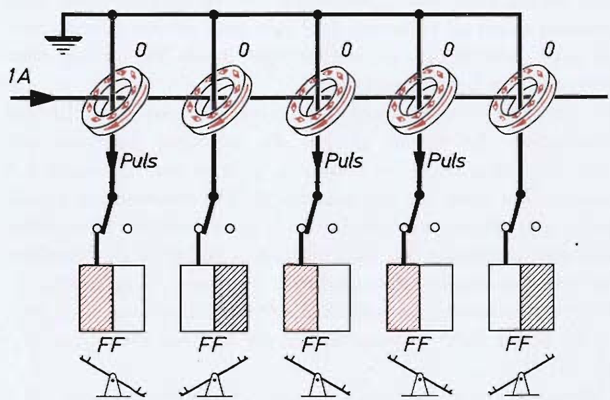


Fig. 19. Het uitlezen van het geheugen

het tweetallig stelsel. Het zal voldoende zijn als wij, voor een willekeurige cijferplaats van de beide getallen, weten aan te geven hoe het cijfer *s* van de som tot stand komt uit de cijfers *a* en *b* van de getallen zelf en de eventuele overdracht *c* die uit de voorafgaande cijferplaats binnenkomt, en bovendien hoe een nieuwe overdracht *d* naar de volgende cijferplaats ontstaat.

...... *a* ......
d ← ← *c*
...... *b* ......
———————— +
s

Wij moeten dus *s* en *d* bepalen voor alle mogelijke combinaties van *a, b* en *c*. Bekijken wij eerst de waarde van de nieuwe overdracht *d* dan geldt:

$d=1$ als *a, b* èn *c* alle drie = 1 zijn
óf als twee ervan = 1 zijn en de derde nièt = 1

Voor s geldt overeenkomstig:

$s=1$ als *a, b* èn *c* alle drie = 1 zijn
óf als één ervan = 1 is en de twee anderen nièt = 1

Dit zijn uitspraken waarbij gebruik gemaakt is van de begrippen „èn", „òf" en „nièt", begrippen die ook in de formele logica voorkomen. Deze kunnen worden vertaald in elektronische schakelingen, de zgn. logica-schakelingen, ook wel *poorten* genoemd. In fig. 20 zijn de „èn-poort", de „òf-poort" en de „nièt-poort" afgebeeld. Daarin heeft *P* alleen dan een spanning van 1 V, als *a,* èn *b,* èn *c* een spanning van 1 V hebben; anders is de spanning 0 V. Zo heeft *Q* alleen dan een spanning van 1 V, als òf *a,* òf *b,* òf *c,* òf 2 ervan, òf alle drie een spanning van 1 V hebben. *R* tenslotte heeft alleen een spanning van 1 V, als die van *a* nièt 1 V is.

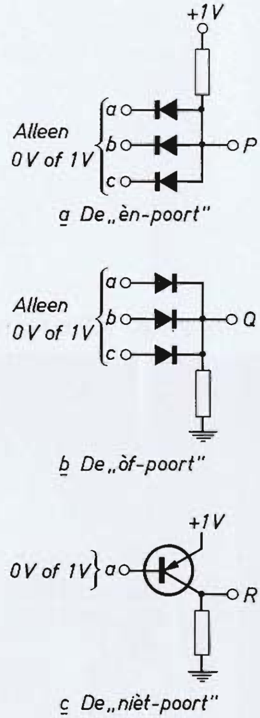


Fig. 20. Schema van enkele poortschakelingen

Toepassingen

Tenslotte nog iets over de toepassingen van elektronische rekenautomaten. Zij kunnen zeer snel werken. Machines die gemiddeld 100.000 programmapunten per sec afwerken komen op het ogenblik reeds voor. Anderzijds moet het rekenprogramma eerst worden samengesteld en in kaarten of papierband geponst worden vóór de machine eraan kan beginnen. Dit is een werk, dat voor moeilijke vraagstukken gemakkelijk een aantal dagen kan duren, nog afgezien van het wiskundige werk dat hieraan vooraf moet gaat om vast te stellen welke rekenmethode gevolgd zal worden. Dit is de taak van de zgn. *programmeurs.* Hieruit volgt dat de rekenmachines alleen toepassing zullen vinden

a. voor zo ingewikkelde of moeilijke vraagstukken dat de berekening op ouderwetse manier onmogelijk is of veel te lang zou duren.

b. voor eventueel gemakkelijke berekeningen die echter zo vaak voor verschillende gevallen moeten worden herhaald, dat bewerken met de hand veel te lang zou duren.

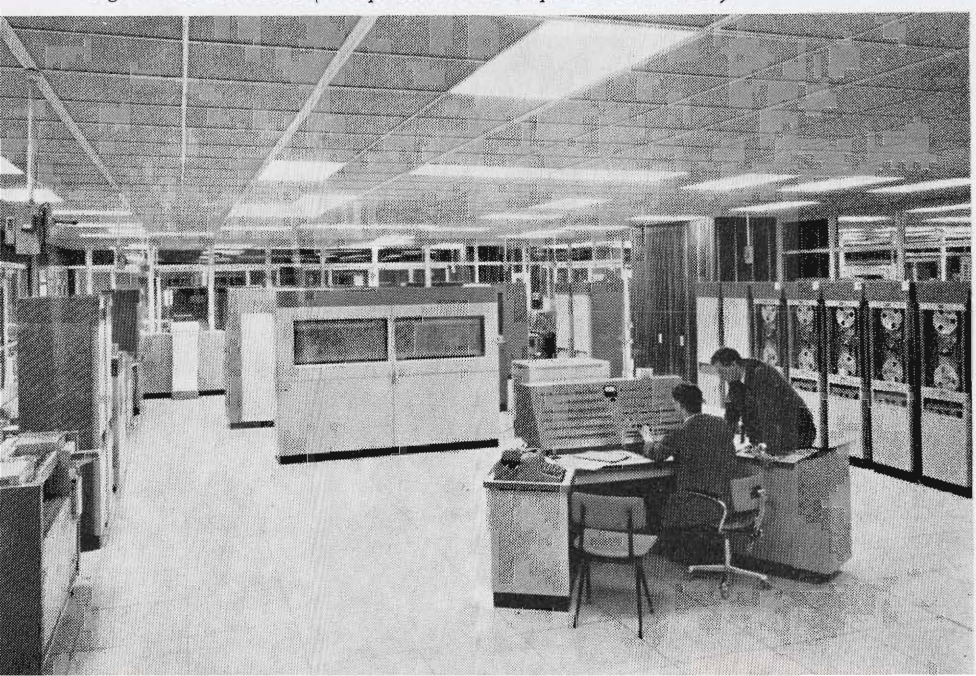
In beide gevallen loont het om het betrekkelijk tijdrovende programmeerwerk voor een elektronische rekenmachine te doen. Tot de groep a. behoren de meeste wetenschappelijke vraagstukken, tot groep b. vele administratieve problemen.

Als een voorbeeld van groep a. moge dienen b.v. de baanberekening van een raket, de correctie ervan uit binnengekomen waarnemingen en de berekeningen van het juiste moment en het juiste bedrag van b.v. hulpaandrijving om aan de baan een gewenste vormverandering te geven.

Een voorbeeld van groep b. is de loonberekening, waarbij voor elke werknemer in principe hetzelfde rekenschema geldt, hoewel de loonschaal, kinderaftrek, loonbelasting en diverse premies per persoon telkens weer anders liggen. De hierbij betrokken berekeningen zijn in de regel slechts eenvoudige optellingen, vermenigvuldigingen en delingen.

De rekenautomaat is een zeer waardevol instrument, een rekentuig dat met ongelooflijke snelheid de berekeningen uitvoert die de mens hem opdraagt. Zonder de mens is de automaat hulpeloos... Woorden als „rekenrobot" en „elektronisch brein" geven een onjuiste indruk van waartoe een rekenautomaat of „computer" in staat is.

Fig. 21. De PASCAL (Philips Automatic Sequence Calculator)



WOORDEN EN BEGRIPPEN

Gedrukt in Nederland

Nr. 9 maart 1963

PHILIPS NEDERLAND N.V. EINDHOVEN
AFD. ONDERWIJSVOORLICHTING

PHILIPS