

MASTER

De Digital Differential Analyser : een digitale rekenmachine voor het oplossen van differentiaal vergelijkingen

Bos, N.J.

*Award date:*  
1964

[Link to publication](#)

**Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Afdeling der Elektrotechniek

Groep Telecommunicatie-B

---

DE DIGITAL DIFFERENTIAL ANALYSER,  
EEN DIGITALE REKENMACHINE VOOR HET OPLOSSEN  
VAN DIFFERENTIAAL VERGELIJKINGEN

---

Afstudeeronderwerp

N. J. Bos

---

Technische Hogeschool Eindhoven

De Digital Differential Analyser,

een digitale rekenmachine voor het  
oplossen van differentiaalvergelijkingen.

Januari 1964

Afstudeerverslag van : N.J.Boo  
Hoogleraar : Prof.ir.A.Hoetman  
Wetenschappelijk Ambtenaar: Ir.C.P.J.Schnabel  
Typewerk : Mej.J.L.C.de Jong

HOOFDSTUK I

Algemene Inleiding en Principe van  
de Digital Differential Analyser.

|  |    |
|--|----|
| A. Inleiding.  | 4  |
| B. Digitale integratie.  | 4  |
| C. Interne organisatie van een D.D.A.  | 9  |
| D. Precisie en bandbreedte   | 11 |
| E. Bespreking van de afrondfout en realisering<br>van het uitgangssignaal.   | 11 |
| F. Het rekenen met positieve en negatieve getallen<br>in het binaire talstelsel zoals dat toegepast<br>wordt in de integrator en detectie van de over-<br>loop van het Y-register. | 15 |

HOOFDSTUK II

Het Geheugen voor de D.D.A.

|                           |    |
|---------------------------|----|
| A. Algemene beschouwing.  | 19 |
| B. Het gebouwde geheugen. | 21 |
| C. De drijverschakeling.  | 26 |
| D. De leesdetector.       | 34 |

HOOFDSTUK III

Logische Schakelingen voor de Integrator.

|  |    |
|--|----|
| A. Organisatie van de rekenschakeling. | 40 |
| B. De opteller.                        | 31 |
| C. De beslissingseenheden.             | 33 |
| E. De overloop-detectie.               | 34 |
| F. De inversie-schakeling.             | 35 |
| G. Het lezen van de tekens.            | 35 |

## HOOFDSTUK IV

### Besturing en selectie van integrator en verzorger.

---

|                           |    |
|---------------------------|----|
| A. Besturing.             | 50 |
| B. De integratorselectie. | 55 |

## HOOFDSTUK V

### Output-apparatuur.

|                               |    |
|-------------------------------|----|
| A. Digitaal-analoog-omzetter. | 58 |
|-------------------------------|----|

## HOOFDSTUK VI

### Eenvoudige Toepassingen van de D.D.A.

|   |    |
|---|----|
| A. Het oplossen van de vergelijking $y' - y = 0$ .  | 62 |
| B. Het oplossen van de vergelijking $y'' - y = 0$ . | 64 |
| C. Het oplossen van de vergelijking $y'' + y = 0$ . | 66 |

|         |    |
|---------|----|
| Besluit | 68 |
|---------|----|

|                                   |    |
|-----------------------------------|----|
| Verklaring van gebruikte symbolen | 69 |
|-----------------------------------|----|

---

## HOOFDSTUK I

### Algemene Inleiding en Principe van de Digital Differential Analyser.

#### A. Inleiding.

Het principe van de "differential analyser" (een rekenmachine die geschikt is voor het oplossen van differentiaalvergelijkingen) is reeds lang bekend.

In eerste vorm werd hij reeds in 1876 voorgesteld door Kelvin. De eerste voor wetenschappelijke doeleinden bruikbare machine werd ontwikkeld door Bush en Caldwell en kwam in 1930 gereed. Het was een elektromechanische rekenmachine die bestond uit achttien "kogel en schijf"-integratoren en bijbehorende verbindende apparatuur.

De ontwikkeling van de versterkertechniek gedurende de tweede wereldoorlog heeft het mogelijk gemaakt nauwkeurige elektronische reken-eenheden te maken. Dit heeft geleid tot de analoge rekenmachine, die een uitgestrekt toepassingsgebied heeft.

Eveneens gedurende de tweede wereldoorlog zijn de digitale technieken enorm ontwikkeld. De wens naar grotere nauwkeurigheid en flexibiliteit was de aanleiding tot de bouw van een digital differential analyser, die in 1950 gereed kwam bij de Northrup Aircraft Inc. en aangeduid werd met de benaming MADDIDA (MAGnetic Drum DIGital Differential Analyser).

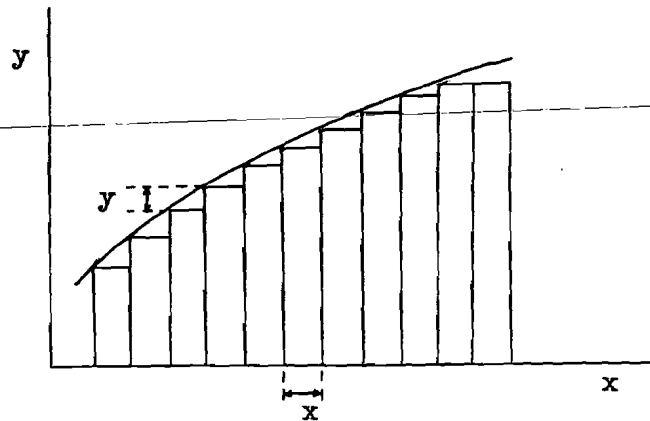
Verschilpunten van deze machine en een analoge rekenmachine waren: grotere nauwkeurigheid bereikbaar, moeilijker te programmeren en veel lagere rekensnelheid. De machine van dit type is verder ontwikkeld tot een snelle digitale rekenmachine die geschikt is voor real time simulations en eenvoudig te programmeren is.

#### B. Digitale integratie.

De basis-eenheid van de digital differential analyser (in het vervolg aangeduid met D.D.A.) is de integrator.

Dit is een rekeneenheid die werkt volgens een vast programma, dat in het volgende toegelicht zal worden.

Om in te zien hoe digitale integratie tot stand kan komen beschouwen we fig.I-1.



Figuur I-1.  
Digitale representatie van  
een continue functie.

Als  $y = f(x)$  dan geldt voor de rechthoekjes  $\Delta z = y\Delta x$ , of sommerend  $\Delta z = y\Delta x$ . Nemen we stapjes  $\Delta x$  oneindig klein dan schrijven we deze uitdrukking als

$$z = \int_{x_1}^{x_2} y dx ,$$

hetgeen de integraalvoorstelling representeert. Om een rekenkundige eenheid te kunnen maken die integratie van willekeurige functies kan realiseren zullen we de bewerking echter meer in details moeten analyseren.

Uit de formule  $\Delta z = y\Delta x$  zien we dat de waarden van de twee variabelen  $y$  en  $\Delta x$  in de integrator aanwezig moeten zijn en met elkaar vermenigvuldigd moeten worden om een waarde voor  $\Delta z$  te verkrijgen. Dit betekent dat een integrator twee ingangssignalen zal moeten ontvangen, n.l. een primair ingangssignaal  $\Delta x$  dat een van te voren bepaalde variatie ter grootte  $h_1$  van de integratie-variabele  $x$  representeert en een secundair ingangssignaal  $\Delta y$  dat een verandering ter grootte  $h_2$  van de functie  $y$  representeert.

We gaan nu  $\Delta x$  bewust beperken tot  $+1$ ,  $0$  en  $-1$  hetgeen betekent dat de absolute waarde van de variatie van de integratie-variabele gelijk is aan  $0$  of  $h_1$ . Dit brengt een vereenvoudiging met zich mee in die zin dat dat de zoëven genoemde vermenigvuldiging van  $y$  en  $\Delta x$  achterwege kan blijven en dat dus  $y_i$  bij de integralewaarde moet worden opgeteld,

afgetrokken of genegeerd moet worden.  $y$  daarentegen kan echter meer dan één differentiaaleenheid veranderen gedurende een cyclus. De waarden van  $y$  zijn beperkt tot  $-1 < y < 1$ .

Iedere integrator bevat twee registers, een Y-register en een R-register. Het Y-register accumuleert de waarde van  $y$ , d.w.z.

$$y_1 = y_0 + h_2 \sum_{s=1}^i (\Delta y)_s$$

waarin  $y_0$  de beginwaarde van  $y$  is en  $\sum (\Delta y)_s$  de algebraïsche som van de secundaire ingangssignalen.

De integrale waarde wordt geaccumuleerd in het R-register, d.w.z. gedurende de  $i^{\circ}$  cyclus wordt  $y_i$  bij de inhoud van het R-register opgeteld, afgetrokken of genegeerd, afhankelijk van de waarde van  $\Delta x$ .

Als de capaciteiten van het R-register voldoende groot is, dan is de inhoud van het R-register na de  $i^{\circ}$  cyclus

$$R_i = R_0 + \sum_{s=1}^i (\Delta x)_s y_s = R_0 + \int_{x_1}^{x_i} y dx$$

waarin  $R_0$  de beginwaarde die in het R-register staat geschreven aanduidt.

Als we een differentiaalvergelijking willen oplossen, dan zullen we integratoren onderling moeten koppelen. Dit heeft slechts zin als de integrator in staat zal zijn een uitgangssignaal te leveren en wel een signaal dat representatief is voor de integrale waarde.

In digitale technieken wordt de informatie gekarakteriseerd door het al of niet aanwezig zijn van een signaal. Dit betekent dat het aantal uitgangssignalen (die een van tevoren bepaalde waarde  $h_3$  hebben) binnen een bepaalde tijd representatief kan zijn voor de waarde van de integraal. Een uitgangssignaal kunnen we eenvoudig verkrijgen door de capaciteit van het R-register te beperken. Een eventuele carry die buiten de registercapaciteit valt (een "overflow" genaamd) is dan een maat voor de waarde van de integraal. Noemen we het uitgangssignaal  $\Delta z$  dan

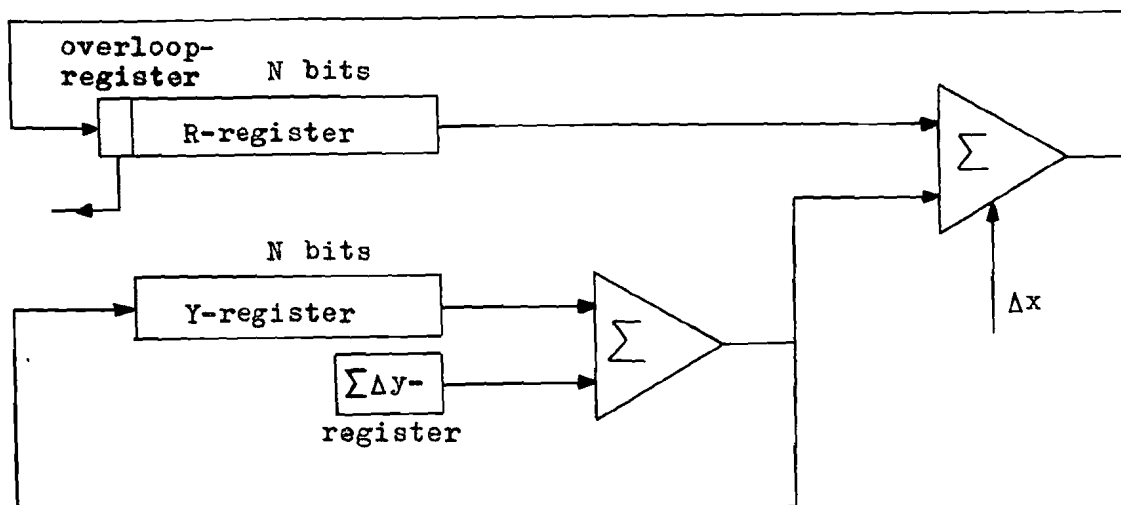
$$R_i = R_0 + \sum_{k=1}^i \{y_k (\Delta x)_k - (\Delta z)_k\} \quad \text{dus}$$

$$R_i = R_{i-1} + y_i (\Delta x)_i - (\Delta z)_i$$

$$(\Delta z)_i = R_{i-1} + y_i (\Delta x)_i - R_i$$



Deze  $\Delta z$ -output kan dienen als secundaire input voor een andere integrator (dan dient  $R$  als de  $y$ -waarde van die integrator) of als primaire input.



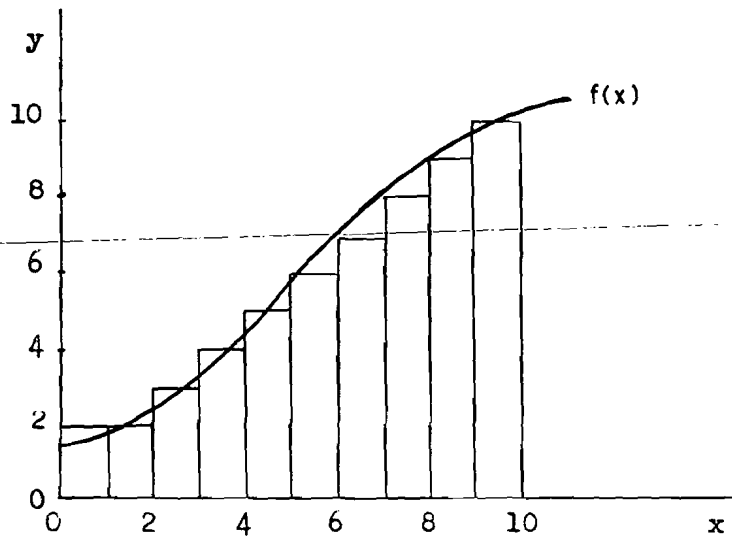
Figuur I-2.  
Blok-schema van een digitale integrator.

Een blok-schema van de integrator in zijn meest eenvoudige vorm kan dan worden gegeven als in fig.I-2.

In het beschreven integrerende systeem wordt de waarde van de integraal benaderd door een som van rechthoekige blokjes.

Er bestaan andere benaderingsmethode die weliswaar gecompliceerder zijn en ingewikkelder apparatuur vereisen, doch grotere nauwkeurigheid pretenderen. We noemen hiervan de benaderingsmethode volgens de trapeziumregel, die in vele moderne D.D.A.'s is toegepast. Dan wordt de integraalwaarde benaderd met een som van trapezia. In verband met de eenvoud wordt door de machine waarvan hier het ontwerp besproken wordt rechthoekige integratie toegepast. De functie  $y = f(x)$  (fig.I-3) wordt benaderd zoals is aangegeven, terwijl het verloop van de berekening genoteerd staat tabel 1.

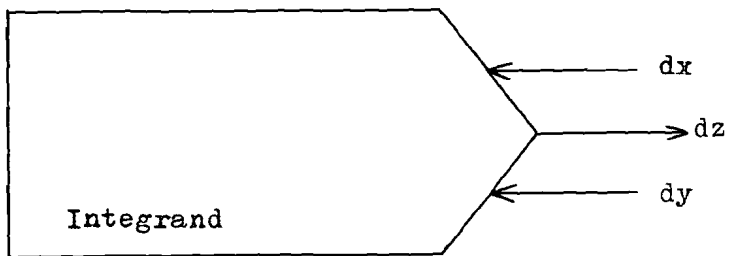
Het in de literatuur gebruikelijk symbool voor een digitale integrator wordt weergegeven in fig.I-4.



Figuur I-3.  
Rechthoekige integratie.

| cyclus | y | $y_i(\Delta x)_i$ | $(\Delta z)_i$ | $R_i$ | $\sum \Delta z_i$ |
|--------|---|-------------------|----------------|-------|-------------------|
| 0      | 0 | 0                 | 0              | 0     | 0                 |
| 1      | 0 | 2                 | 0              | 2     | 0                 |
| 2      | 0 | 2                 | 0              | 4     | 0                 |
| 3      | 1 | 3                 | 0              | 7     | 0                 |
| 4      | 1 | 4                 | 1              | 1     | 1                 |
| 5      | 1 | 5                 | 0              | 6     | 1                 |
| 6      | 1 | 6                 | 1              | 2     | 2                 |
| 7      | 1 | 7                 | 0              | 9     | 2                 |
| 8      | 1 | 8                 | 1              | 7     | 3                 |
| 9      | 1 | 9                 | 1              | 6     | 4                 |

Tabel 1  
Voorbeeld van de rekenwijze van een  
integrator in het decimale talstelsel.



Figuur I-4.  
Symbool van een digitale integrator.

### C. Interne Organisatie van een D.D.A.

Zoals reeds eerder is opgemerkt bestaat een D.D.A. uit een aantal integratoren (waarmee men zoals we later zullen zien ook andere operaties ~~dan integreren kan verrichten~~) die onderling gekoppeld zijn op een wijze die afhangt van het te behandelen probleem.

Er zijn vele manieren om een D.D.A. te organiseren en met betrekking tot de aard van de organisatie verdeelt men de machines in twee hoofdgroepen.

#### Simultane type

Iedere integrator heeft een eigen rekeneenheid die gelijktijdig opereren der integratoren mogelijk maakt.

1 iteratieperiode = 1 integratiestap = constant, onafhankelijk van het aantal integratoren dat voor het oplossen van een bepaald probleem gebruikt wordt.

#### Sequentie type

De integratoren hebben een gemeenschappelijke rekeneenheid en een centraal geheugen

De y en R getallen paren worden in succesie van het geheugen naar de centrale rekeneenheid gebracht om de rekenoperatie te ondergaan en teruggevoerd.

1 iteratieperiode = totale tijd, nodig om de rekenkundige bewerkingen der integratoren achtereenvolgens te volbrengen =  $N\tau$

$N$  = aantal ingeschakelde integratoren (hangt van het probleem af).

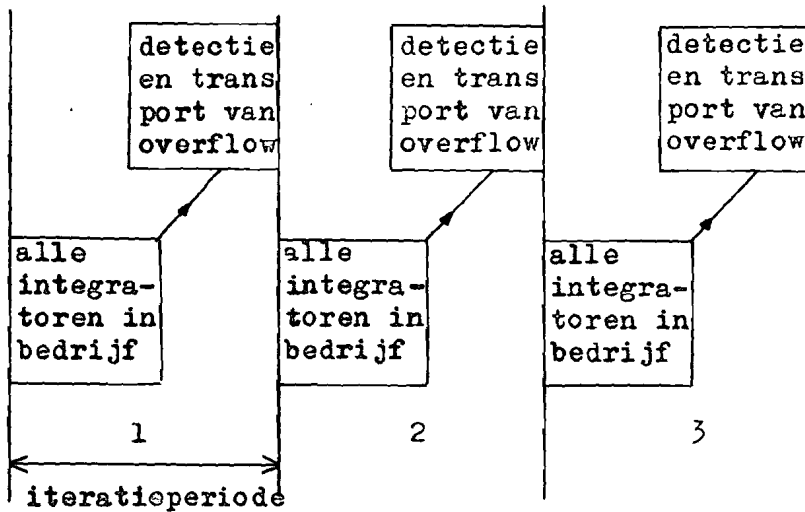
$\tau$  = integratieperiode van één integrator.

Elk van deze machinetypen heeft zijn voor- en nadelen. Zo heeft de simultane D.D.A. altijd dezelfde iteratieperiodeduur. Bij de sequentie machine neemt de iteratieperiodeduur lineair toe met het aantal integratoren dat gebruikt wordt, dus de machine is aanzienlijk langzamer voor meer complexe probleem (heeft dus een geringere bandbreedte: zie later). Daar staan echter weer de geringere kosten van minder uitgebreide apparatuur tegenover.

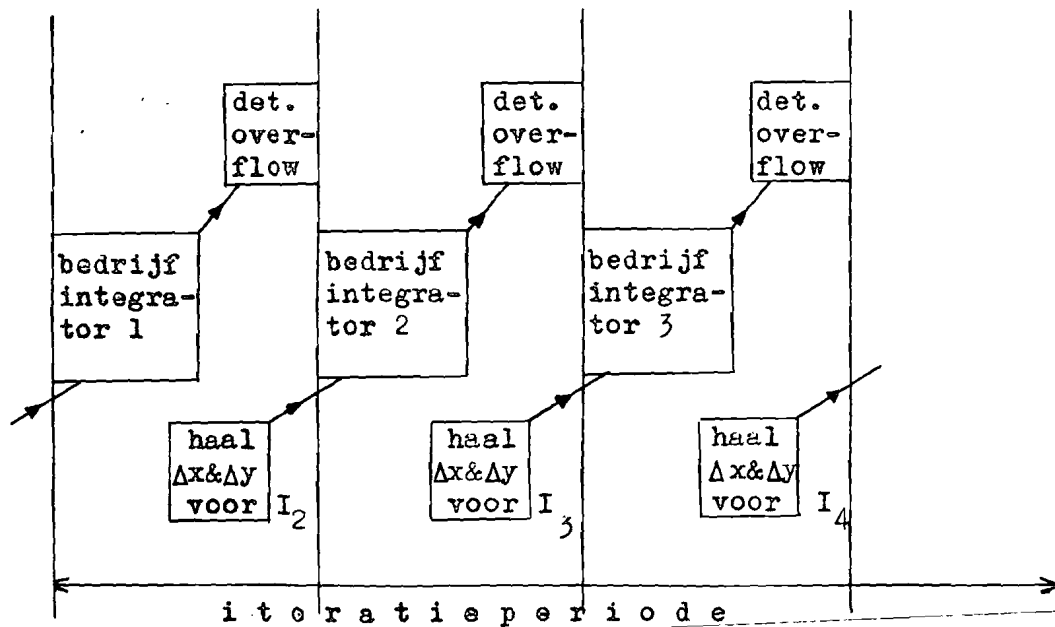
Met betrekking tot de wijze waarop de rekenoperatie verricht wordt, kan men D.D.A.'s verder onderverdelen in serie- en parallelmachines. Het zal duidelijk zijn dat het laatste machinetype sneller is. Voor de bouw van zo'n machine is echter aanzienlijk meer materiaal vereist terwijl men een zekere vrijheid in de keuze van de woordlengte verliest.

De integratoren worden direct met elkaar gekoppeld bij de simultane machine. ~~Men hoeft namelijk alleen te zorgen voor transport van de~~

overflow, die bij alle integratoren op hetzelfde tijdstip aan de uitgang verschijnt. Deze werkwijze is schematisch aangegeven in fig.I-5. Het programmeren van een sequentie D.D.A. is gecompliceerder. Daar de rekenkundige bewerkingen der integratoren achtereenvolgens gedaan worden, verschijnen de overflows niet gelijktijdig. Zij zullen dus naar een geheugen gebracht moeten worden om op het juiste tijdstip weer gelezen te kunnen worden (zie fig.I-6).



Figuur I-5.  
Informatieverwerking bij een  
simultane D.D.A.



Figuur I-6.  
Informatieverwerking bij  
een sequentie D.D.A.

D. Precisie en Bandbreedte.

Dit zijn karakteriserende grootheden voor de machine.

De precisie is een parameter die groter wordt als de nauwkeurigheid waarmee een grootheid gerepresenteerd wordt toeneemt en kan dus gedefinieerd worden met behulp van het aantal bits dat gebruikt wordt. Van een grootheid die door N bits gerepresenteerd wordt is de precisie gelijk aan  $2^N$ . Het is verstandig om de precisie, zo gedefinieerd, te onderscheiden van de nauwkeurigheid, daar de laatste ook van andere factoren afhangt.

De bandbreedte van de machine wordt bepaald door de harmonische functie met de grootst mogelijke precisie en de maximale frequentie die men met twee integratoren van de machine kan genereren.

De relatieve grootte van de differentiaal  $\Delta y$  en de iteratietijd T bepalen de maximale snelheid van de verandering  $\frac{\Delta y}{T}$  van een afhankelijke variabele.

Voor een sinus met maximale precisie is de maximale frequentie  $\omega_m$  die gegenereerd kan worden

$$\omega_m = \frac{\Delta y}{2\pi T} = \frac{1}{2\pi T \cdot 2^N} ,$$

waarbij aangenomen is dat het register over N significante plaatsen beschikt en we het tweetallige talstelsel toepassen. Het produkt van precisie en bandbreedte is dus een machineconstante:  $1/2\pi T$ . Bij sequentie machines is de iteratietijd  $T = N\tau$  (zie blad 9). Als nu de rekenkundige bewerkingen in de integrator van een simultane D.D.A. op dezelfde wijze worden uitgevoerd, dan zal de bandbreedte van de laatste bij benadering een factor N beter zijn dan die van de sequentie machine.

Een andere factor die de bandbreedte beïnvloedt, is de tijd die nodig is om twee bits bij elkaar op te tellen. Bij alle machinetypen is de integratietijd omgekeerd evenredig met de klokfrequentie.

E. Bespreking van de afrondfout en realisering van het uitgangssignaal.

Zoals uit het voorgaande is gebleken, bestaat de taak van de integrator uit het op het juiste tijdstip produceren van een overflow. We zullen daarom nader beschouwen hoe de overflow tot stand komt en mede aan de hand van een foutendiscussie bepalen welke betekenis er aan de overflow gehecht moet worden.

Opgemerkt is reeds dat de waarde van  $y$  beperkt is tot  $-1 < y < +1$  en dat we de capaciteit van het R-register moeten beperken om een output-sigitaal te kunnen verkrijgen. We krijgen een maximale output-snelheid als we de capaciteit van het R-register gelijk maken aan de capaciteit van het Y-register. Nu is de inhoud van het R-register in wezen een restje dat het gedeelte van de waarde van de integraal representeert dat niet als  $\Delta z$ -output-sigitaal getransporteerd is, en volgens de zojuist gedane aanname en het beschreven rekenschema waarden zal hebben die liggen tussen  $-1$  en  $+1$ . Deze rest is de afrondfout van  $z$ . Een  $\Delta z$  output-sigitaal kan slechts dan gegeven worden als de absolute waarde van de rest gelijk wordt aan  $1$  of deze waarde overschrijdt.

De grootte van de afrondfout kan echter begrensd worden door een output-sigitaal te genereren indien de absolute waarde van het getal in het R-register groter wordt dan of gelijk wordt aan  $0,5$ . Bij handhaving van de conceptie zoals die tot dusver beschreven is, kan de overflow, die vroeger automatisch door de carry ontstond, nu slechts verkregen worden door expliciet aftrekken toe te passen.

Dit doen we dus als  $y_t + R_t \geq 0,5$  of als  $y_t + R_t < -0,5$ .

We illustreren dit voor uitsluitend positieve getallen aan het volgende voorbeeld.

|                                    | niet afgerond |                                     | afgerond    |
|------------------------------------|---------------|-------------------------------------|-------------|
|                                    | $R_0$ 0,00    |                                     | 0,00        |
|                                    | $y_1$ 0,76    |                                     | 0,76        |
|                                    | <u>0,76</u>   |                                     | <u>0,76</u> |
| $\sum \Delta z = 0$<br>fout = 0,76 |               | $\sum \Delta z = 1$<br>fout = -0,24 |             |
|                                    | $R_1$ 0,76    | $0,76 - 1 = -0,24$                  |             |
|                                    | $y_2$ 0,81    |                                     | 0,81        |
|                                    | <u>0,57</u>   |                                     | <u>0,57</u> |
| $\sum \Delta z = 1$<br>fout = 0,57 |               | $\sum \Delta z = 2$<br>fout = -0,43 |             |

In dit geval kan het expliciet aftrekken eenvoudig omzeild worden door als beginwaarde in het R-register 0,5 te schrijven.

Indien we met positieve getallen te maken hebben, wordt het  $\Delta z$ -uitgangssignaal verzorgd door de carry, die nu automatisch optreedt indien de inhoud van het R-register, afkomstig van de berekening, de waarde 0,5 bereikt of overschrijdt.

We willen een negatieve  $\Delta z$ -output als  $R \ll -0,5$ . We moeten dan echter  $-1$  aftrekken. Dit is echter niet nodig als we het 10-complement van  $y$  gebruiken als het produkt  $y \times$  negatief is. De som van de Y en R registers is dan  $(1 + y_t) + (R_t + 0,5) = y_t + R_t + 1,5$ . Een output-sig-naal willen we dan als deze som kleiner is dan 1, d.w.z.  $R_t + y_t \ll -0,5$ .

Daar nu geen automatische carry optreedt, fungeert de toevoeging van  $+1$ , ontstaan door het gebruik van de complementaire vorm van  $y$ , als correctie van het te genereren output-sig-naal. Deze rekenwijzen wordt als volgt ge-illustreerd:

|              |             |
|--------------|-------------|
| $R_0 = 0,50$ | 0,50        |
| $y_1 = 0,76$ | <u>0,76</u> |
| 0,26         | 0,26        |

$\sum \Delta z = 1$   
fout = -0,24

$\sum \Delta z = +1$

|              |             |
|--------------|-------------|
| $R_1 = 0,26$ | 0,26        |
| $y_2 = 0,75$ | <u>0,75</u> |
| 0,01         | 0,01        |

$\sum \Delta z = 2$   
fout = -0,49

$\sum \Delta z = +2$

|               |              |
|---------------|--------------|
| $R_2 = 0,01$  | 0,01         |
| $y_3 = -0,74$ | <u>-0,26</u> |
| -0,73         | 0,27         |

$\sum \Delta z = 1$   
fout = -0,23

$\sum \Delta z = +1$

|                          |              |
|--------------------------|--------------|
| $-0,73 + 1 = R_3 = 0,27$ | 0,27         |
| $y_4 = -0,75$            | <u>-0,25</u> |
| -0,48                    | 0,52         |

$\sum \Delta z = 0$   
fout = 0,02

$\sum \Delta z = 0$

$-0,48 + 1 = R_4 = 0,52$

De regels voor de waardebeëpalig van  $\Delta z$  samenvattend:

1.  $y \Delta x$  is positief. Een carry op de meest significante plaats betekent dat  $R_t + y_t \gg 0,5$ , dus  $\Delta z = +1$ .

Geen carry:  $R_t + y_t \leq 0,5$ , dus  $\Delta z = 0$ .

2.  $y \Delta x$  is negatief. Geen carry op de meest significante plaats betekent dat  $R_t + y_t \ll -0,5$ , dus  $\Delta z = -1$ . Wel een carry zegt dat

$R_t + y_t \gg -0,5$ , dus  $\Delta z = 0$ .

We merken hierbij op dat het teken van de restwaarde geen rol speelt. De inhoud van het R-register is dan ook altijd positief. Verder is het van belang dat dezelfde regels en bewerkingen gelden indien de rekenkundige bewerkingen worden uitgevoerd in het 2-tallige stelsel. Aan de hand van de zojuist genoemde regels is het mogelijk met behulp van de truth-table de logische schakeling te bepalen die het  $\Delta z$ -uitgangssignaal genereert. Daartoe gaan we uit van de volgende definities: Een positief getal wordt aangegeven door een 0 op de meest significante plaats, een negatief getal staat in de twee-complementaire vorm en wordt herkend aan de 1 op de meest significante plaats.

We krijgen dan voor het uitgangssignaal

$$|\Delta z| = \begin{cases} +1 = 01 \\ 0 = 00 \\ -1 = 11 \end{cases}$$

|       |       | C  |    |
|-------|-------|----|----|
| $T_y$ | $T_x$ | 0  | 1  |
| 0     | 0     | 00 | 01 |
| 0     | 1     | 11 | 00 |
| 1     | 1     | 00 | 01 |
| 1     | 0     | 11 | 00 |

a.  $|\Delta z|$

|       |       | C |   |
|-------|-------|---|---|
| $T_y$ | $T_x$ | 0 | 1 |
| 0     | 0     | 0 | 0 |
| 0     | 1     | 1 | 0 |
| 1     | 1     | 0 | 0 |
| 1     | 0     | 1 | 0 |

b.  $T_{\Delta z}$

|       |       | C |   |
|-------|-------|---|---|
| $T_y$ | $T_x$ | 0 | 1 |
| 0     | 0     | 0 | 1 |
| 0     | 1     | 1 | 0 |
| 1     | 1     | 0 | 1 |
| 1     | 0     | 1 | 0 |

c.  $|\Delta z|$

Tabel 2.  
Functietabellen.

C = overloop van het R-register.



De functietabel voor  $z$  wordt dan als tabel 2a. Voor het karakteriseren van de drie waarden die  $z$  kan aannemen zijn minimaal twee binaire geheugenelementen zoals b.v. flipflops noodzakelijk. We splitsen daarom ~~de functietabel in een tabel voor de teken flipflop (tabel 2b) en een tabel voor de floplop die  $z$  aangeeft (tabel 2c).~~

Uit tabel 2b volgt:

$$T_{\Delta z} = \bar{T}_y T_{\Delta x} \bar{C} + T_y \bar{T}_{\Delta x} C = (T_y \otimes T_{\Delta x}) \bar{C}$$

Uit tabel 2c volgt:

$$\begin{aligned} \Delta z &= T_y T_{\Delta x} C + \bar{T}_y T_{\Delta x} \bar{C} + T_y \bar{T}_{\Delta x} C + T_y \bar{T}_{\Delta x} \bar{C} = \\ &= (\bar{T}_y \bar{T}_{\Delta x} + T_y T_{\Delta x}) C + (T_y \bar{T}_{\Delta x} + \bar{T}_y T_{\Delta x}) \bar{C} = (T_{\Delta x} \otimes T_y) \bar{C} + (\overline{T_{\Delta x} \otimes T_y}) C \end{aligned}$$

Als we  $T_x \otimes T_y = P$  noemen, dan is

$$|\Delta z| = P\bar{C} + \bar{P}C = P \otimes C,$$

$$\text{dus } |\Delta z| = (T_{\Delta x} \otimes T_y) \otimes C = T_x \otimes T_y \otimes C.$$

Verder dient nog opgemerkt te worden dat het uitgangssignaal altijd gelijk moet zijn aan nul indien  $|\Delta x| = 0$ .

F. Het rekenen met positieve en negatieve getallen in het binaire talstelsel zoals dat toegepast wordt in de integrator en detectie van de overloop van het Y-register.

De voorstelling van positieve getallen in het tweetallige stelsel wordt volledig bekend verondersteld. Indien men ook met negatieve getallen wilt gaan werken, dient het teken apart te worden aangegeven. Voor  $y$  en  $\Delta x$  zijn reeds definities gekozen, die verder gehandhaafd zullen worden. Het optellen en aftrekken van positieve en negatieve getallen kan algebraïsch altijd gereduceerd worden tot het optellen van een positief of een negatief getal. Ditzelfde kan men rekenkundig doen door als men een negatief getal moet optellen, het complement van dat getal te nemen, waarbij men, indien men binair rekent, ~~nog de keuze heeft tussen het een en het twee-complement.~~ In ons geval, waar een serie-opteller zal worden toegepast, geeft het twee-complement de minste complicaties.

Als we genoemde definities toepassen en voor de tekenbit volgens de conventies de meest significante plaats reserveren dan betekent een 0 voor binaire getal dat het getal positief is en een waarde heeft die het getal volgens het zuiver binaire coderingssysteem aangeeft. Een 1 op de meest significante plaats van het getal betekent dat het getal negatief is en in twee-complementaire vorm staat. Met de tekenbit kan gerekend worden als ware het een normale significante bit.

We willen nu onze aandacht wijden aan het optellen van de getallen  $y$  en  $y$ , waarbij we de volgende gevallen moeten onderscheiden:

- a.  $|y| \quad |\Delta y|$
- b.  $|y| = |\Delta y|$
- c.  $|y| \quad |\Delta y|$

De tekens  $T_y = \alpha$  en  $T_{\Delta y} = \beta$  moeten ook in rekening gebracht worden, zodat we in totaal 12 gevallen onderscheiden.

Het is van belang dat de som van  $y$  en  $\Delta y$  de capaciteit van het Y-register waarin het nieuwe resultaat komt, niet wordt overschreden.

Indien dit wel het geval is, zullen rekenfouten ontstaan, die gesignaleerd moeten worden. Uit het volgende blijkt dat men de overlooptdetectie kan uitvoeren aan de hand van een tekenbit-controle.

| $ y  \begin{matrix} > \\ = \\ < \end{matrix}  \Delta y $ | $\alpha$ | $\beta$ | qualificatie antwoord   |
|--|----------|---------|---|
| 1. $>$   | +        | +       | echte overflow als $\delta = 1$ alarm = 001 = $\alpha\beta\delta$   |
| 2. $=$   | +        | +       |   |
| 3. $<$   | +        | +       |   |
| 4. $>$   | +        | -       | 1 die buiten registercapaciteit valt weglaten, dan antwoord juist.  |
| 5. $=$   | +        | -       |   |
| 6. $<$   | +        | -       |   |
| 7. $>$   | -        | +       | antwoord direct goed.   |
| 8. $=$   | -        | +       | 1 die buiten registercapaciteit weglaten, dan is het antwoord juist.  |
| 9. $<$   | -        | +       |   |
| 10. $>$  | -        | -       | echte overflow indien $\delta = 0$ . Er treedt altijd een 1 op die buiten de registercapaciteit valt en weggelaten moet worden. Alarm = 110 = $\alpha\beta\delta$ . |
| 11. $=$  | -        | -       |   |
| 12. $<$  | -        | -       |   |

Tabel 3.  
Voorwaarden voor het optreden van een overflow.

De verschijnselen die kunnen optreden gaan we na voor alle gevallen zoals die opgesomd zijn in tabel 3. De gevallen 1, 2, en 3 zijn rekenkundig hetzelfde. Als het aantal significante plaatsen en de plaats voor de tekenbit binnen de registercapaciteit vallen, dan verloopt de berekening zonder complicaties. We laten dit zien aan de hand van voorbeelden en gebruiken hiervoor getallen met vier significante plaatsen, waarvoor een tekenbit staat.

| <u>Vb.</u> | decimaal    | binair            |
|------------|-------------|-------------------|
|            | +3          | 0   0011          |
|            | <u>+5</u> + | <u>0   0101</u> + |
|            | +8          | 0   1000          |

We zien dat getalwaarde en tekenbit correct zijn.

Hebben we echter meer dan vier significante plaatsen nodig om het antwoord te schrijven, dan treedt een fout op ter plaatse van de tekenbit: de uitkomst zal door de machine als een negatief getal geïnterpreteerd worden.

|            |              |                   |
|------------|--------------|-------------------|
| <u>Vb.</u> | + 8          | 0   1000          |
|            | <u>+11</u> + | <u>0   1011</u> + |
|            | +19          | 1   0011          |

Fout is dus de tekencombinatie  $\alpha\beta\delta = 001$ .

Geval 4 levert altijd een positieve uitkomst. Als we de 1 die buiten de registercapaciteit valt altijd zonder meer weglaten, is het antwoord juist.

|            |              |                   |
|------------|--------------|-------------------|
| <u>Vb.</u> | +13          | 0   1101          |
|            | <u>- 7</u> + | <u>1   1001</u> + |
|            | + 6          | 1 0   0110        |

In geval 5 krijgen we altijd de uitkomst 0, als we de 1 die buiten de registercapaciteit valt weer weglaten.

|            |              |                   |
|------------|--------------|-------------------|
| <u>Vb.</u> | + 9          | 0   1001          |
|            | <u>- 9</u> + | <u>1   0111</u> + |
|            | 0            | 1 0   0000        |

In geval 7 krijgen we hetzelfde resultaat als in geval 6, 8 is hetzelfde geval als 5 en 9 hetzelfde als 4.

Ook de gevallen 10, 11 en 12 zijn rekenkundig hetzelfde. Hier treedt ~~altijd een 1 op die buiten de registercapaciteit valt~~ en weggelaten moet worden. Als we om de absolute waarde van het antwoord te kunnen schrijven meer dan vier bits nodig hebben, treedt een fout op ter plaatse van de tekenbit.

$$\begin{array}{r}
 \text{Vb.} \quad -8 \qquad \qquad 1 \text{ ; } 1000 \\
 \quad \underline{-3} \text{ +} \qquad \qquad \underline{1 \text{ ; } 1101} \text{ +} \\
 \quad -11 \qquad \qquad 1 \text{ 1 ; } 0101
 \end{array}$$

Dit resultaat wordt als we de 1 op de meest significante plaats weglaten inderdaad als -11 geïnterpreteerd.

$$\begin{array}{r}
 \text{Vb.} \quad -8 \qquad \qquad 1 \text{ ; } 1000 \\
 \quad \underline{-9} \text{ +} \qquad \qquad \underline{1 \text{ ; } 0111} \text{ +} \\
 \quad -17 \qquad \qquad 1 \text{ 0 ; } 1111
 \end{array}$$

Dit antwoord wordt door de machine als een positief getal opgevat en is dus fout. Fout is daarom de tekencombinatie  $\alpha\beta\delta = 110$ . De resultaten zijn samengevat in tabel 3. Samenvattend concluderen we dat een alarmsignaal gegeven moet worden indien we voor  $\alpha\beta\delta$  de combinatie 001 of 110 krijgen.

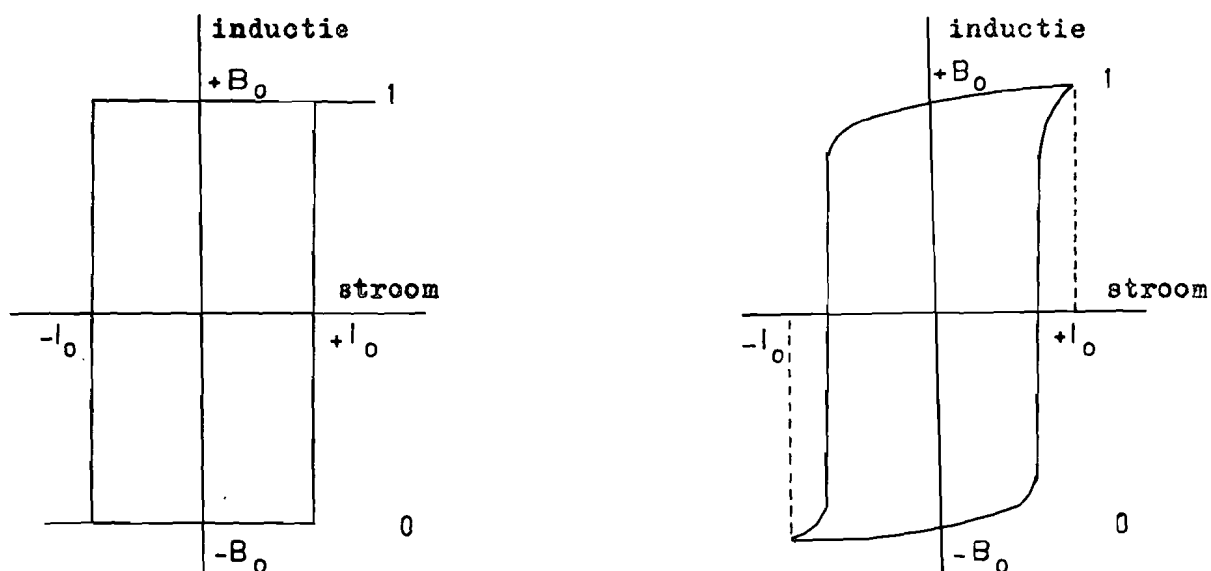
## HOOFDSTUK II

### Het geheugen voor de D.D.A.

#### A. Algemene beschouwing.

Het ligt in de bedoeling om integratoren te bouwen met een registercapaciteit van zestien bits, waarbij de tekenbit is inbegrepen.

Men kan nu voor het Y en het R-register van de integrator twee afzonderlijke flipflop-schuifregisters gebruiken. Dit is een eenvoudige oplossing die echter tot aanzienlijke materiaal- en ruimtekosten aanleiding geeft indien men een groot aantal integratoren gaat bouwen. Een goedkopere oplossing bestaat dan in de vorm van het kerngeheugen. In kerngeheugen wordt de informatie bewaard in ringetjes (kernen) die gemaakt zijn van een keramisch ferromagnetisch materiaal.



a. Geïdealiseerde rechthoekige hysteresislus van een ferroxcube geheugen kern.

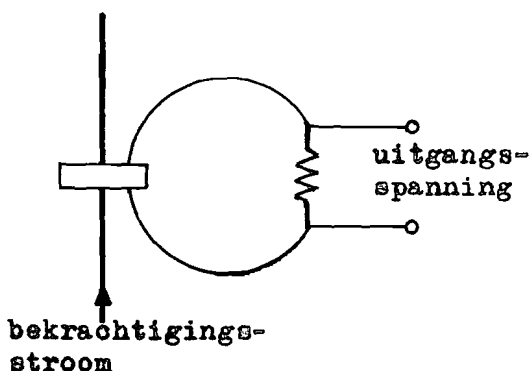
b. De werkelijke vorm van de hysteresislus.

Figuur II-1.

Als zo'n ring eerst in de verzadiging wordt gestuurd, terwijl daarna de bekrachtiging wordt weggenomen, kan de inductie slechts de waarde +B<sub>0</sub> of -B<sub>0</sub> aannemen, afhankelijk van de richting van de uitwendig aangelegde bekrachtiging. (figuur I-1a geeft de geïdealiseerde rechthoekige hysteresislus. Figuur I-1b geeft een meer reële vorm van de lus. In beide figuren zijn de twee mogelijke toestanden aangegeven).

Als men nu b.v. de toestand +B<sub>0</sub> "1" noemt en de toestand -B<sub>0</sub> "0", dan is de kern geschikt geworden om informatie te bevatten in de vorm van

een bit. De informatie wordt uit de kern gelezen door een bekrachtiging aan te brengen dusdanig dat een kern die zich in de "1" stand bevindt, omklapt naar de "0" stand, zodat een omklappende kern een spanning induceert in een draad die door de opening van de kern gestoken is. Een kern die reeds in de "0" stand was, zal niet omklappen en daarom ook geen spanning in een draad kunnen induceren (zie fig.II-2).



Figuur II-2  
Het lezen van de kern.

Het lezen van kernen brengt met zich mee dat de informatie verloren gaat. Immers na het lezen bevindt de kern zich altijd in de "0" stand. Men zal de informatie al dan niet gewijzigd terug moeten schrijven om op een ander tijdstip weer over de informatie te kunnen beschikken. Een "1" kan men schrijven door een bekrachtigingsstroom in tegengestelde richting toe te voeren, een "0" schrijft men door de bestaande toestand te handhaven, dus door geen bekrachtigingsstroom toe te voeren.

Uit fig.II-1 ziet men dat de be-

krachtigingsstroom de waarde  $I_0$  moet overschrijden om de magnetisatie van de kern te doen omklappen. Deze eigenschap maakt het ons mogelijk om uit een groot aantal kernen één kern te selecteren. Men rangschikt de kernen op de wijze van elementen in een matrix (zie fig.II-3). Men steekt nu door elke rij kernen een draad, evenals door elke kolom, zodat door iedere kern twee draden gaan. Selecteert men nu een rij en een kolom en voert men door de betreffende draden een stroom ter grootte  $\frac{1}{2}I_0$ , dan zal door slechts één kern een stroom  $I_0$  gaan, zodat alleen de geselecteerde kern kan omklappen, afhankelijk van de stroomrichtingen en de richting van de reeds aanwezige kernmagnetisatie. De selectie vindt dus plaats met behulp van de coïncidentie van twee stromen. Geheugens waarin de kernen op deze wijze geselecteerd worden, noemt men bitgeorganiseerde geheugens. Men kan ook een hele rij kernen gelijktijdig uitlezen. Dan is coïncidentie van stromen niet noodzakelijk, als de bekrachtigingsstroom door de draad maar groot genoeg is. Deze selectiemethode wordt toegepast bij woordgeorganiseerde geheugens.

Welk van de beide systemen de voorkeur verdient, hangt o.a. af van de capaciteit van het geheugen en de toepassingen waarvoor het bestemd is.

---

B. Het gebouwde geheugen.

Daar de te bouwen rekenmachine van het simultane type zal moeten zijn en de rekenoperatie in serie uitgevoerd wordt, ligt het voor de hand om een woordgeorganiseerd geheugen te ontwerpen. Gekozen zijn geheugenmatjes die  $32 \times 32$  kernen bevatten (zie fig.II-3).

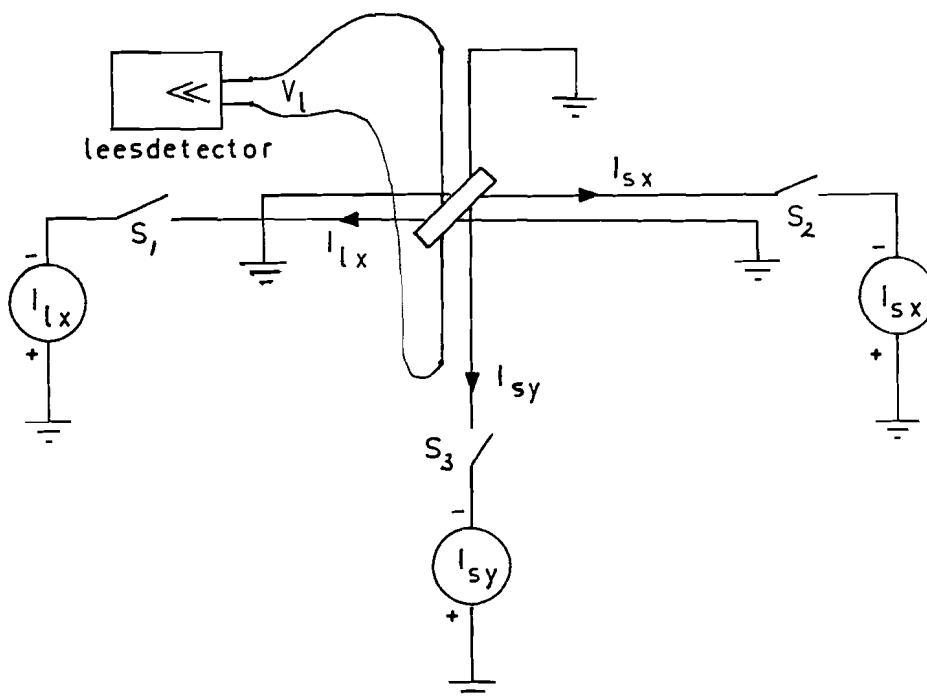
Figuur II-3  
Geheugenmatrix.

Een rij kernen in de x-richting vormt dan een woord van twee en dertig bits dat gelezen kan worden door een voldoende grote stroom door een gemeenschappelijke draad door de kernen. Daar de integrator twee registers bevat, is de capaciteit van een matje toereikend voor zestien integratoren.

De kolommen in de y-richting bestaan eveneens uit twee en dertig kernen, waarvan er slechts zestien benut worden. De overige kernen worden gereserveerd voor een eventuele uitbreiding van de woordlengte.

De geheugenmatrix bevat in de x-richting een draad voor de leesstroom  $I_{lx}$  en een voor de schrijfstroom  $I_{sx}$ .

De schrijfstroom is in richting tegengesteld aan de leesstroom. In de y-richting is een draad nodig voor de digitschrijfstroom  $I_{sy}$ , die het mogelijk maakt een "0" of een "1" te schrijven en een draad waarin een omklappende kern een stroom kan opwekken: de leesdraad. Door een kern gaan dus twee x en twee y-draden (zie fig.II-4).



Figuur II-4  
De ligging van de draden door een kern en de richtingen van de lees- en schrijfstromen.

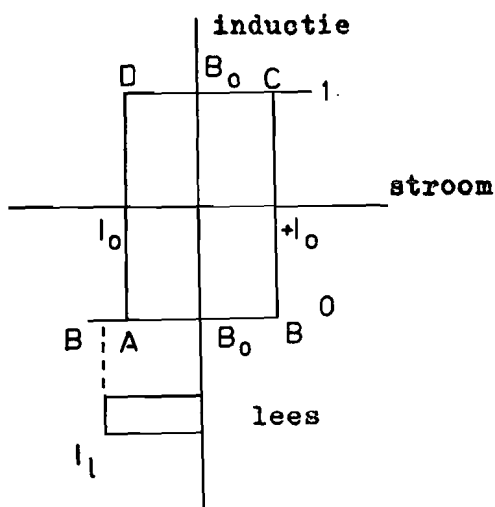
Uit het voorgaande blijkt dat we twee operaties moeten verrichten, n.l. lezen en nadat de informatie een bewerking heeft ondergaan, schrijven. Voor een bepaalde kern gebeuren deze operaties direct na elkaar, d.w.z. als een woord wordt gelezen, dan wordt het al dan niet gewijzigd teruggeschreven voordat er een nieuw woord gelezen wordt. We kunnen dus spreken van een lees-schrijfcyclus.

Hoe deze cyclus uitgevoerd wordt, lichten we in het volgende toe.

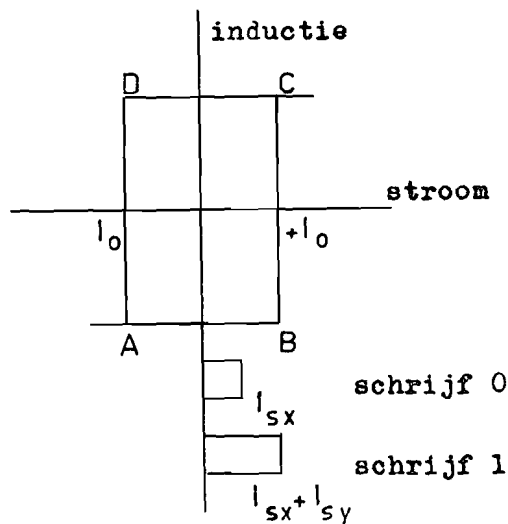
Figuur II-5 geeft de geschematiseerde hysteresiskromme weer, als in figuur II-1. Lezen geschiedt door de winding te bekrachtigen met een



stroom  $I_{lx}$ . Als we er van uitgaan dat de kern zich in de "1" stand bevindt, dan zal de magnetisatie omklappen van de waarde  $B_0$  naar  $-B_m$  via het gedeelte  $B_0$  A- $B_m$  van de lus. Voorwaarde hiervoor is dat de leesstroom in grootte op zijn minst gelijk is aan  $I_0$  of deze waarde overschrijdt.



Figuur II-5a  
Lezen



Figuur II-5b  
Schrijven

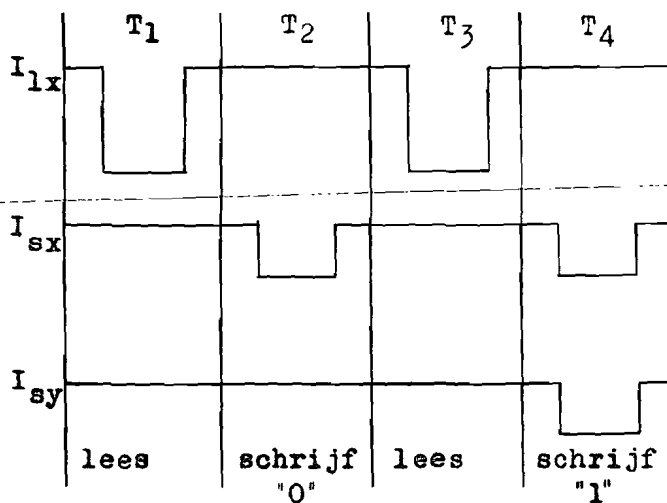
Is de stroom kleiner dan  $I_0$ , dan zal de magnetisatie van de kern niet omklappen. Hiervan maken we gebruik bij het schrijven (fig.II-5b). We sturen dan altijd een schrijfstroom  $I_{sx}$ , die we niet groter nemen dan  $\frac{1}{2}I_0$  om gedeeltelijk omklappen van de magnetisatie te voorkomen. Als we een "0" willen schrijven, sturen we geen schrijfstroom in de y-richting. Willen we een "1" schrijven, dan sturen we in de y-richting gelijktijdig met  $I_{sx}$  een stroom  $I_{sy}$ , die we eveneens ter grootte  $\frac{1}{2}I_0$  nemen. Een grotere waarde is niet toegestaan omdat anders de magnetisatie van andere kernen in de betreffende kolom gedeeltelijk kan omklappen. De stroomrichtingen zijn aangegeven in figuur II-4.

Het puls-tijd diagram voor deze bewerkingen vinden we in figuur II-6. Hier lezen we eerst een kern en schrijven daarna een "0" terug, vervolgens lezen we een volgende kern en schrijven dan een "1" terug door coïncidentie van de stromen  $I_{sx}$  en  $I_{sy}$ . De tijdsintervallen  $T_1 + T_2$  en  $T_3 + T_4$  zijn nu beide een lees-schrijf cyclustijd.

In het geheugen zijn toegepast de 6C1 kernen.

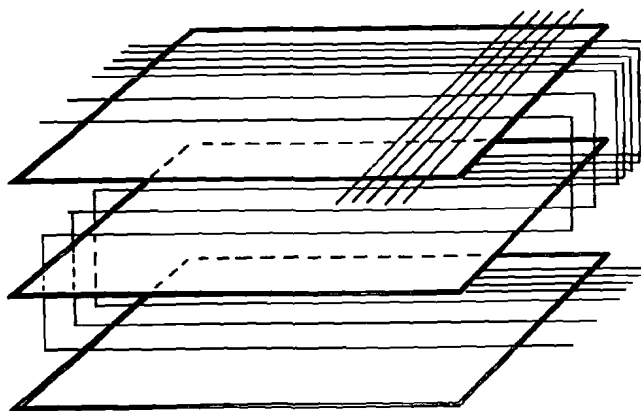
Deze kernen moeten we lezen met een stroom van minstens 500 mA. De schrijfstroom bedraagt 275 mA. Op de leesdraad kunnen we dan pulsen verwachten die groter zijn dan 36 mV. (1)

De lees- en schrijfstromen worden geschakeld met behulp van getransistoriseerde drijverschakelingen, die geselecteerd worden vanuit de centrale

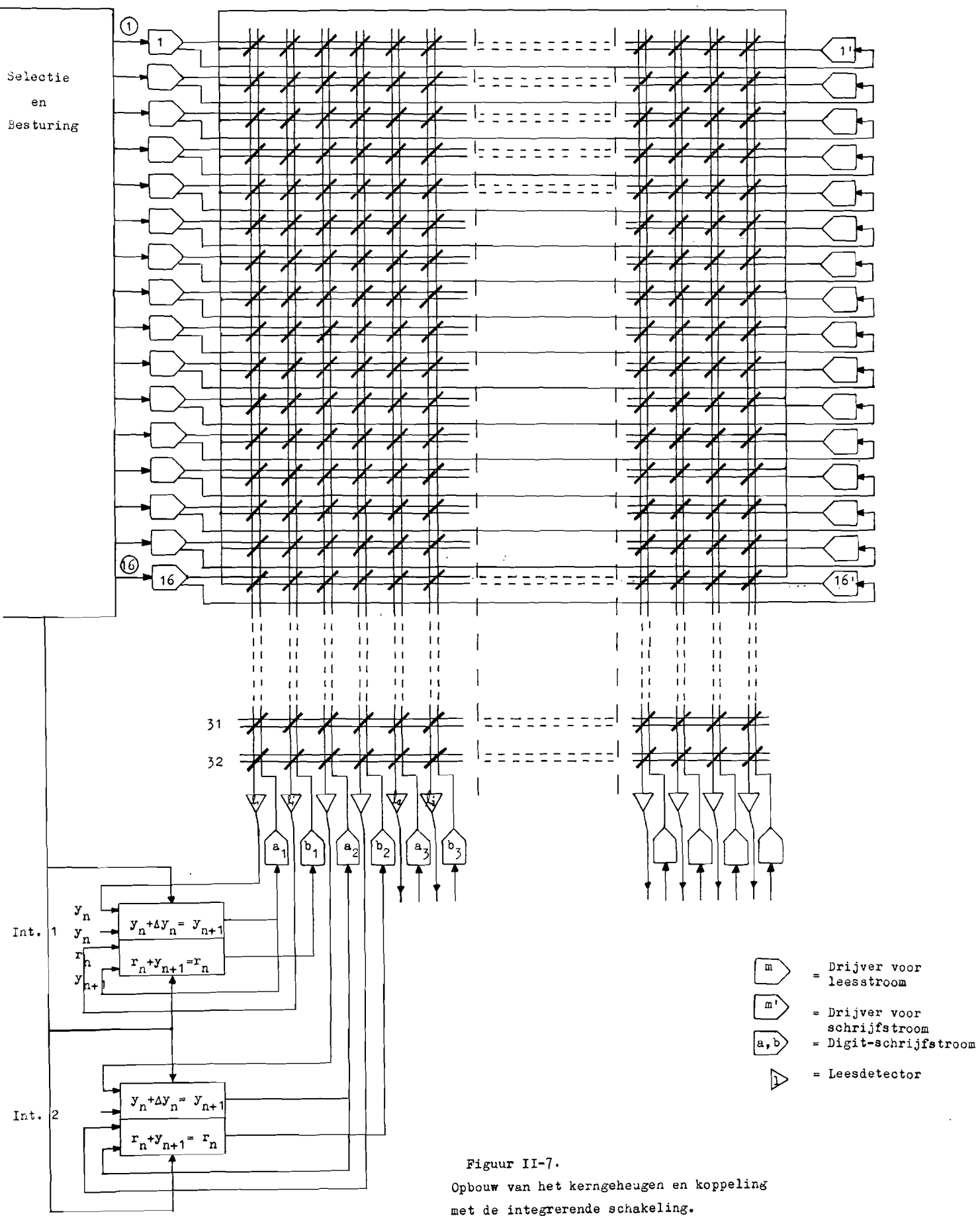


Figuur II-6  
Het lezen en schrijven  
van de ferriet-kern.

besturing. De signalen die op de leesdraad geïnduceerd worden, worden gedetecteerd door de leesdetector, vanwaar de informatie naar de optellers gevoerd wordt. Hoe de geheugendraden met de diverse schakelingen verbonden worden, zien we in fig.II-7, waar een geheugenmatje met bijbehorende apparatuur getekend is. Het is mogelijk om de geheugen capaciteit uit te breiden in horizontale richting door meer geheugenmatjes te gebruiken. Men moet dan de x lees- en schrijfdraden doorverbinden. Dit is getekend in fig.II-8.



Figuur II-8  
Het doorverbinden van de  
geheugenmatjes.



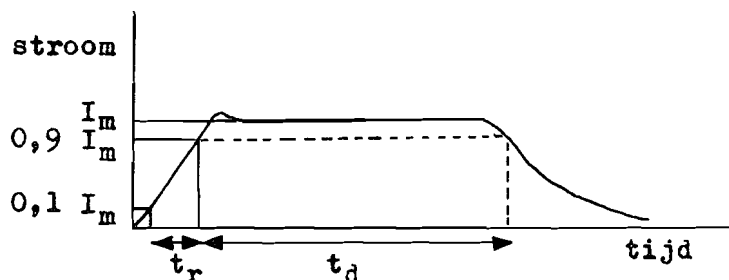
Figuur II-7.  
Opbouw van het kerngeheugen en koppeling met de integrerende schakeling.

### C. De drijverschakeling.

De stroomimpulsen die nodig zijn om het ommagnetiseren van de kernen in het geheugen te bewerkstelligen worden in moderne geheugens geleverd door transisterschakelingen. Daarbij zijn de stijgtijd en de pulsduur van belang.

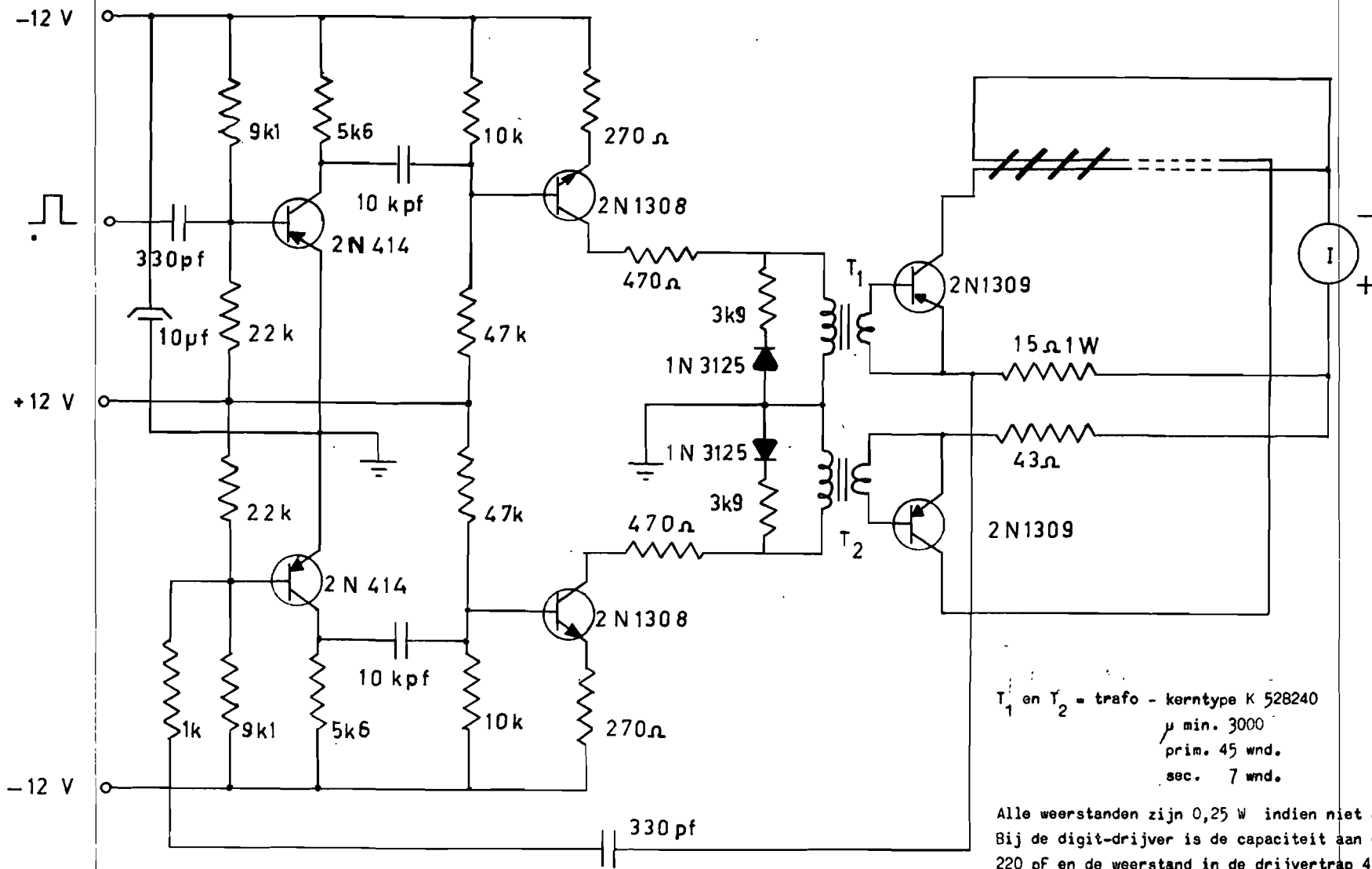
Voor grotere geheugens is het nodig dat deze schakelingen zich als een stroombron gedragen. Immers, de draden in het geheugen vormen een impedantie met een reactieve component die een ongunstige invloed heeft op die stijgtijd van de puls. Daarom neemt men in de keten meestal een serieweerstand op. Om nu toch tot de vereiste collectorstroom te kunnen komen zijn betrekkelijk hoge spanningen nodig. Voor de D.D.A. gebruiken we echter een klein geheugen, zodat de draden kort gehouden kunnen worden. Een spanning van 12 Volt bleek voldoende.

Figuur II-9 geeft het principeschema van de drijver. Deze schakeling schakelt zowel de leesstroom  $I_{lx}$  als de schrijfstroom  $I_{sx}$  en is eigenlijk opgebouwd uit twee overeenkomstige schakelingen, die symmetrisch zijn getekend t.o.v. de +12 Volt-lijn. Voor de 6Cl-kernen moet de  $I_{lx}$ -drijver in staat zijn pulsen te leveren die voldoen aan de volgende eisen: stijgtijd  $t_r = 0,2 \mu s$ ; pulsduur  $t_d = 2 \mu s$ ;  $I_m = 500 \text{ mA}$ . (zie fig.II-10) (In deze gegevens is een tolerantie van 10% toegestaan). Een lees-schrijfcyclus zal dan minstens  $5 \mu s$  moeten duren. De kern levert dan pulsen op de leesdraad die voldoen aan de door de fabriek gespecificeerde waarden.



Figuur II-10  
Stroompuls voor het ommagnetiseren  
van een ferrietgeheugenkern.

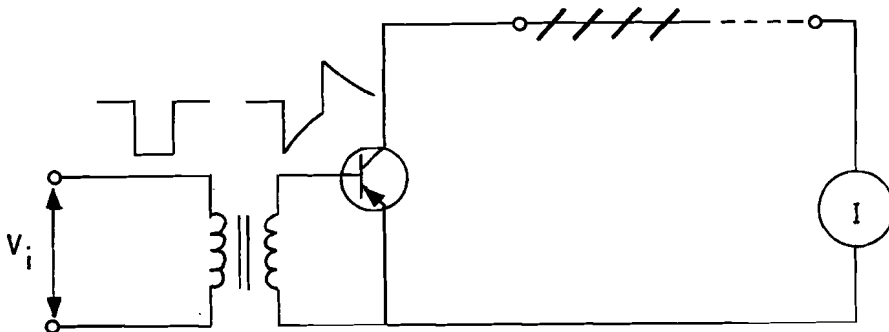
Om een groter "1" signaal op de leesdraad te krijgen is voor  $I_{lx}$  een stroom van ca. 650 mA. gekozen. Deze stroom kan geschakeld worden door een 2 N 1309 transistor die een piekstroom van 1 A mag hebben en daarbij een  $\beta = 20$  heeft. Dit houdt in dat we gedurende bijna  $2,5 \mu s$  een basisstroom van meer dan 30 mA aan de basis moeten kunnen leveren. Om echter de nodige flanksteilheid te verkrijgen, moeten de transistoren



Figuur II-9

Het prinsipschema van de drijver voor lezen en schrijven.

overstuurd worden. Dit betekent dat het z.g. "hole-storage"-effect, dat toeneemt met de basisstroom, storend kan gaan optreden. Met name de afvaltijd kan aanzienlijk toenemen, hetgeen hinderlijk is, daar de lees-schrijfcyclustijd hierdoor groter wordt, terwijl een langere pulsduur voor het omklappen van de kernmagnetisatie niet meer van belang is. Men zal dus maatregelen moeten nemen om het hole-storage verschijnsel, dat van transistor tot transistor variëert en een aanzienlijke spreiding vertoont, binnen de perken te houden. Men kan dit bereiken door zodra de stroom naar nul moet terugkeren de resterende lading uit de basis te zuigen. Daarom wordt de basis van de schakeltransistor vaak met de voorgaande trap gekoppeld via een transformator. Deze heeft de neiging om de ingangspuls te differentiëren (zie fig.II-11). Tijdens het af-

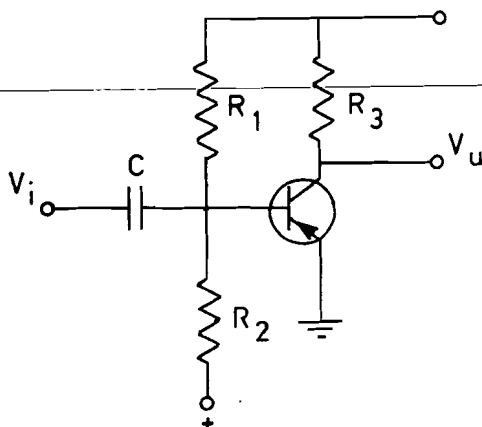


Figuur II-11.  
Reductie van het hole-storage  
verschijnsel bij de drijver-  
transistor.

vallen van de ingangspuls komt dan over de basis-emitterovergang van de transistor een positieve spanning te staan die de ongewenste elektronen in de basis wegzuigt, waardoor de transistor snel dicht gezet wordt. Het toepassen van een pulstransformatortje levert ons bovendien, nog het voordeel dat we eenvoudig stroomversterking kunnen verkrijgen door het aantal secundaire wikkelingen kleiner te kiezen dan het aantal primaire. De stroom die door de voorgaande trap geschakeld moet worden, hoeft dan niet groter te zijn dan 10 mA.

De drijvers worden geselecteerd vanuit de centrale besturing, d.w.z. door een signaal dat afkomstig is van een flipflop. Deze signalen worden ook gebruikt om andere eenheden te sturen en mogen daarom niet zwaar belast worden. Ook hebben we een stuursignaal met een korte stijgtijd nodig, zodat we eerst een vermogensversterking toepassen en gelijktijdig een stijgtijdverbetering bewerkstelligen. Hiertoe schakelen we een transistor

als in fig.II-12.

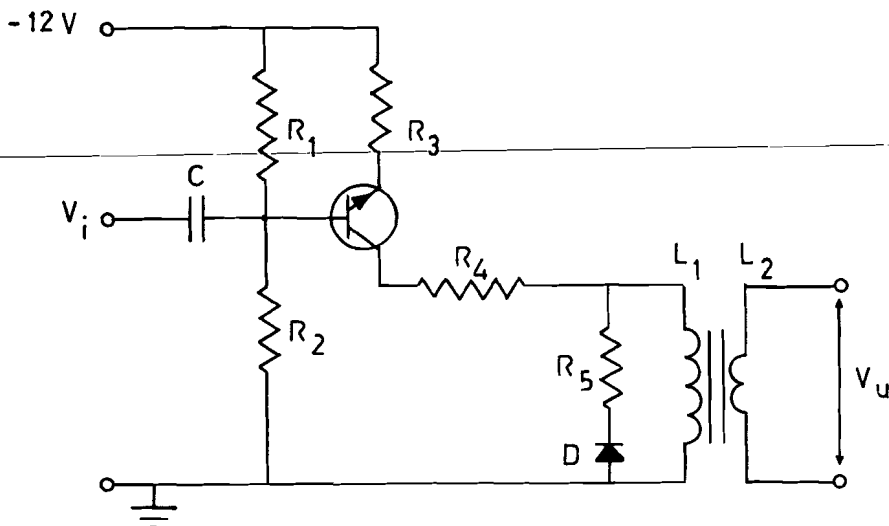


Figuur II-12  
Ingangstrap van de drijver.

Deze transistor is in rusttoestand geleidend. De emitter is met aarde verbonden, terwijl de basis een voorspanning heeft die vastgelegd is met behulp van de spanningsdeler  $R_1R_2$ . Deze spanning is slechts weinig negatief, want ook de basisstroom loopt door de weerstand  $R_1$ . Dan is een kleine positieve spanningsprong op de ingang al voldoende om de transistor dicht te zetten, waardoor de stroom door  $R_3$  ophoudt, zodat de collector aan de negatieve pool van de batterij komt te liggen: we krijgen aan de uitgangsklem een negatief gaande spanningspuls, waarvan de breedte behalve van het ingangssignaal afhangt van de dimensionering van het RC-ingangnetwerk. De pulsbreedte is binnen zekere grenzen instelbaar met behulp van  $R_2$ .

Ingangspulsen van 8 Volt en een stijgtijd van ca.  $1/\mu\text{s}$  veroorzaken aan de uitgang spanningspulsen van 12 Volt met een stijgtijd van  $0,2/\mu\text{s}$ . Deze stijgtijdverkorting kunnen we verklaren door te bedenken dat de ingangspuls niet zijn eindwaarde hoeft te bereiken om de transistor dicht te zetten en er zodoende slechts een gedeelte van de voorflank gebruikt wordt. Met het uitgangssignaal schakelen we de volgende trap. Deze bestaat uit een NPN-transistor die de pulstransformator in de collectorstroom heeft en in rusttoestand eveneens geleidend is (zie fig.II 13). De collectorstroom (en dus ook de stroom door de primaire wikkeling van de transformator) bedraagt dan ca. 10 mA en kan ingesteld worden met behulp van de spanningsdeler  $R_1R_2$ .

De weerstand  $R_4$  is opgenomen om de spanning over de transistor en daarmee de in de transistor gedissipeerde warmte te reduceren. Een negatief gaande spanningspuls aan de ingangsklem zet de NPN-transistor plotseling dicht.



Figuur II-13  
Circuit dat de basisstroom  
levert voor de drijvertransistor.

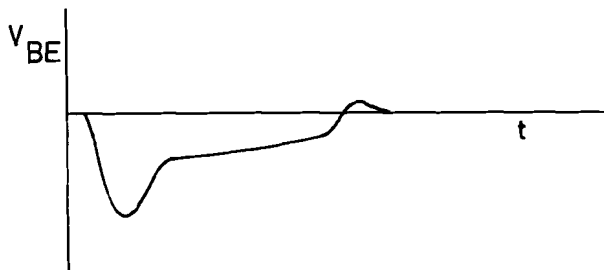
De stroom door de zelfinductie  $L_1$  wijzigt dan in een zeer korte tijd van 10 naar 0 mA. De spanning over  $L_1$  is in formule gelijk aan  $L_1 \frac{di}{dt}$ . Nemen we aan dat de schakeltijd  $0,2 \mu s$  is een  $L_1 = 1 \text{ mH}$ , dan kunnen we aan de ingangsklemmen voor de pulstransformator een spanning van ongeveer 50 Volt verwachten, indien de secundaire wikkeling onbelast is:

Als de drijvertransistor is aangesloten, bedraagt de reële primaire spanning ongeveer 20 Volt.

De basisspanning en -stroom zijn dan als weergegeven is in fig.II-14.

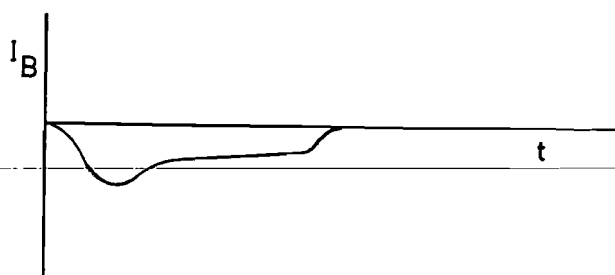
De diode en de weerstand over  $L_1$  dienen slechts om ongewenste trillingen te dempen.

Daar gedurende de gehele stroompulsduur voldoende basisstroom geleverd moet kunnen worden, dient de tijdconstante  $\frac{R}{L}$  enkele malen groter te zijn dan de gewenste maximale pulsbreedte.



Figuur II-14a  
Basisspanning.





Figuur II-14b  
Basisstroom.

Basisspanning en stroom van  
de drijvertransistor.

Figuur II-15  
Leesstroompuls  $I_{1x}$ .  $t = 2 \mu\text{s}/\text{cm}$

Figuur II-16  
Schrijfstroompuls  $I_{sx}$ .  $t=2\mu s/cm$

De schakeling kan stroompulsen leveren tot 1 A met een stijgtijd van ca.  $0,3\mu s$ . Kleinere stromen kunnen sneller geschakeld worden. Zo heeft de leesstroompuls van 650 mA een stijgtijd van  $0,20$  à  $0,25\mu s$ , (fig.II-15) terwijl de stijgtijd van  $I_{sx}$ , die slechts 275 mA is, ongeveer  $0,15\mu s$  bedraagt (fig.II-16).

De lees-drijver mag één maal gedurende  $50\mu s$  een puls afleveren van  $2\mu s$  pulsbreedte. Dit houdt verband met de toegestane dissipatie in de 2 N 1309 en de emitterweerstand waarvan de waarde de amplitude van de stroompuls bepaald. Stromen tot 400 mA kunnen geschakeld worden met een herhalingsfrequentie van 250 kHz, indien in de emitter een weerstand van 5 Watt gemonteerd wordt. Dit is van belang van de digit-drijver die de stroom  $I_{sy}$  schakelt en daarmee een "0" of een "1" schrijft. Deze moet elke cyclustijd een puls kunnen ontvangen. De schakeling van de digit-drijver is in wezen dezelfde als die voor de andere drijvers. Slechts enkele componenten hebben andere waarden.

Het lees- schrijf-systeem, is dusdanig dat de  $I_{sx}$ -drijver altijd gestart moet worden direct nadat de  $I_{lx}$ -puls afgelopen is. (zie fig.II-6) Daarom kunnen we de eerste starten op de achterflank van de  $I_{lx}$ -puls. Daartoe is de ingang van de  $I_{sx}$ -drijver, die gevormd wordt door een serie-schakeling van een condensator ( $330\text{ pF}$ ) en een weerstand ( $1\text{ K}\Omega$ ). Zie fig.II-9), verbonden met de emitterweerstand van de  $I_{lx}$ -drijvertransistor. Met het oog

Figuur II-17.

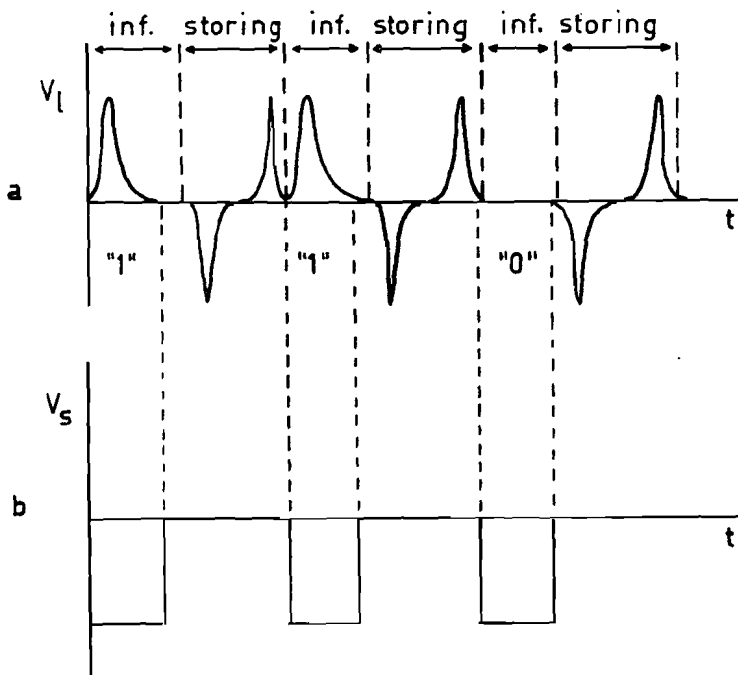
Een drijver voor lezen en schrijven gemonteerd op een insteekplaat.

op flexibiliteit is het verstandig om deze verbinding uitwendig aan te brengen. Het is mogelijk om de  $I_{1x}$  drijver en de  $I_{1y}$  drijver samen op één plaat te monteren die, evenals de standaardschakelingen, in combinatie met een Varipakrek gebruikt kan worden (zie fig.II-17). Evenzo kan men twee  $I_{sy}$ -drijvers op één plaat monteren.

D. De leesdetector.

De signalen die we op de leesdraad vinden hebben een amplitude die in de buurt van 50 mV ligt. Deze signalen kunnen dus niet zonder meer gebruikt worden om andere componenten te sturen, maar moeten eerste versterkt worden.

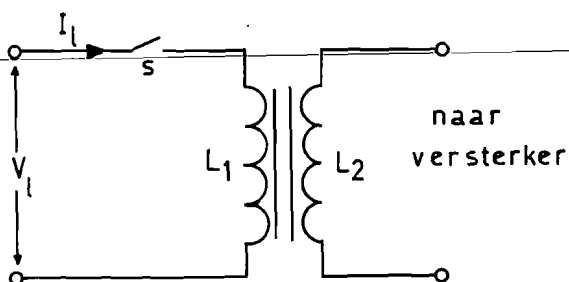
Een vervelende bijkomstigheid is echter dat we op de leesdraad stoor-signalen vinden die van dezelfde grootte-orde zijn als de te detecteren puls. Deze ontstaan doordat de  $I_{sy}$ -schrijfdraad en de leesdraad naast elkaar in het geheugen liggen en zodoende sterk gekoppeld zijn (zie fig. II-3 en II-4). Het signaal dat we op de leesdraad vinden is dan b.v. zoals



Figuur II-18  
a. signaal op de leesdraad.  
b. poortstuursignaal.

weergegeven is in fig.II-18a. Een amplitudedetectie heeft geen zin en we zullen daarom een sample-circuit in de leesversterker moeten aanbrengen,

die alleen een signaal doorlaat op het moment dat men werkelijke informatie kan verwachten.



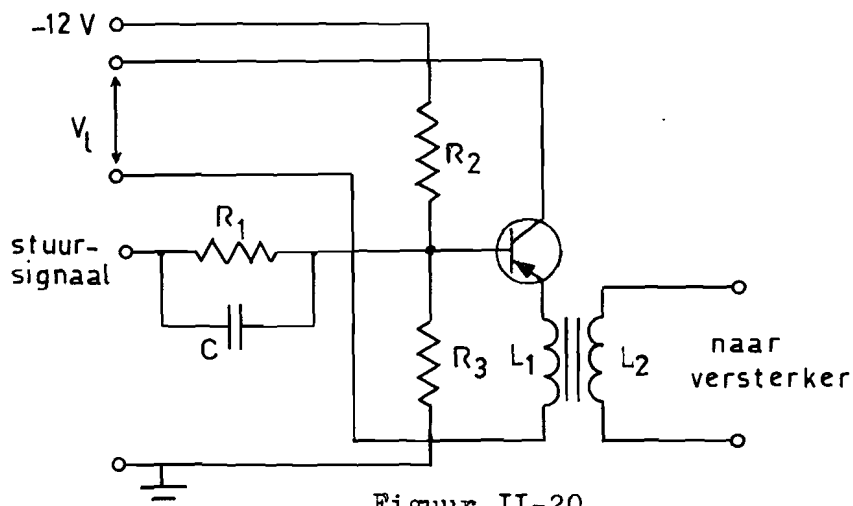
Figuur II-19  
Ingang van de lees-  
detector.

In zijn principiële vorm kan dit circuit voorgesteld worden door een schakelaar die alleen gesloten is als nuttige informatie aan de ingangsklemmen kan verschijnen (fig.II-19).

Via deze schakelaar voeren we het in de leesdraad geïnduceerde signaal naar de primaire ingangsklemmen van een transformatortje dat de spanning optransformeerd tot een waarde die voldoende is om een transistor open te zetten.

De schakelfunctie wordt gerealiseerd met behulp van een transistor, die een dusdanige gelijkstroominstelling heeft, dat hij in rusttoestand gesloten is (zie fig.II-20). In de emitterketen is dan de primaire transformatorwikkeling opgenomen.

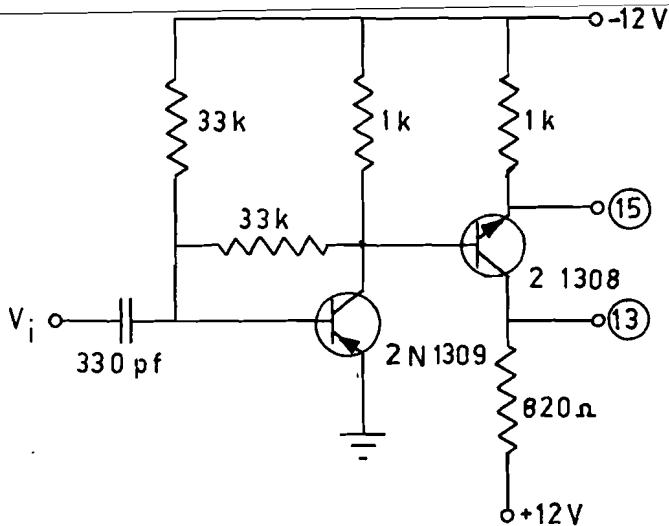
Door negatief gaande signalen wordt de transistor geopend.



Figuur II-20  
De poortschakeling.

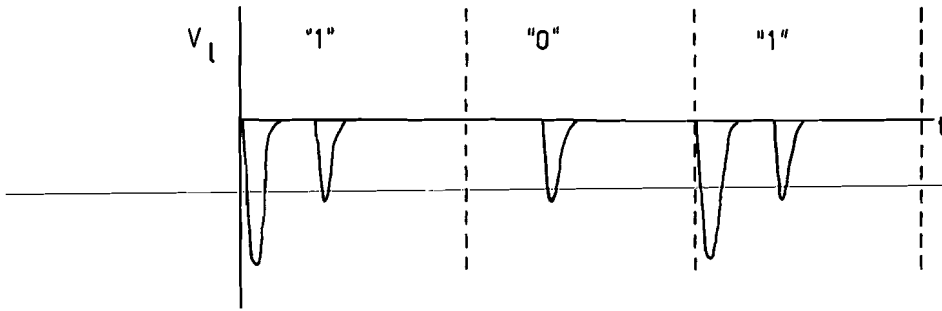
De stuursignalen zijn afkomstig van de schakeling SS (zie fig.V-3 : besturing). Het principieschema van deze schakeling vinden we in fig.II-21. De ingangstrap wordt gestuurd door een pulstrein bestaande uit zestien

pulsen, die afkomstig zijn van de scanner. Deze pulsen bepalen ook het moment waarop de leesstroom door een geselecteerde draad van het kern-



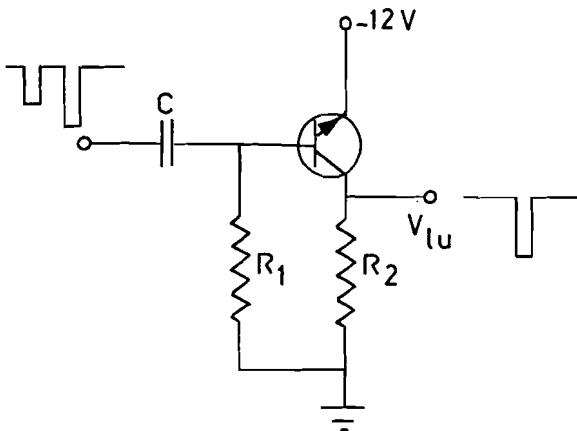
Figuur II-21  
Eenheid die het sample-circuit  
van de leesdetector stuurt.

geheugen vloeit. De 2 N 1309 transistor wordt door de positief gaande spanningpulsen dichtgezet, zodat we over de collectorweerstand negatief gaande signalen krijgen. Een gelijkstroomterugkoppeling naar de ingang maakt het ons mogelijk om de pulsbreedte te regelen en op een geschikte waarde in te stellen. De poort moet namelijk op tijd weer dicht zijn. De volgende trap levert ons gelijktijdig een vermogensversterking en een inversie, zodat we de schakeling tevens kunnen toepassen als pulsvormer. Van de uitgang 15 worden de kloksignalen voor de optellers in de integrator afgeleid. Uitgang 13 mag worden belast met vier leesdetectors. In fig.II-22 zien we het volledige principeschema van de leesdetector. De signalen op de transformatoruitgangsklemmen zetten, indien een "1" gedetecteerd wordt, de 2 N 1308 transistor open en geven dan negatief gaande spanningpuls over de collectorweerstand. Over deze weerstand vinden we echter ook een stoorsignaal dat afkomstig is van de samplex-sturing, maar in amplitude kleiner is dan de informatiepulsen (zie fig.II-23).



Figuur II-23  
Informatiepulsen en stoor-  
signalen na sampling en ver-  
sterking.

Niettemin kunnen logische schakelingen op deze stoorsignalen reageren, zodat we niveau-detectie moeten toepassen. Daartoe gebruiken we een transistor waarvan de gelijkstroominstelling dusdanig is dat hij in rusttoestand geleidend is (zie fig.II-24). De basisstroom door  $R_1$  is niet voldoende om de transistor bijna dicht te zetten.

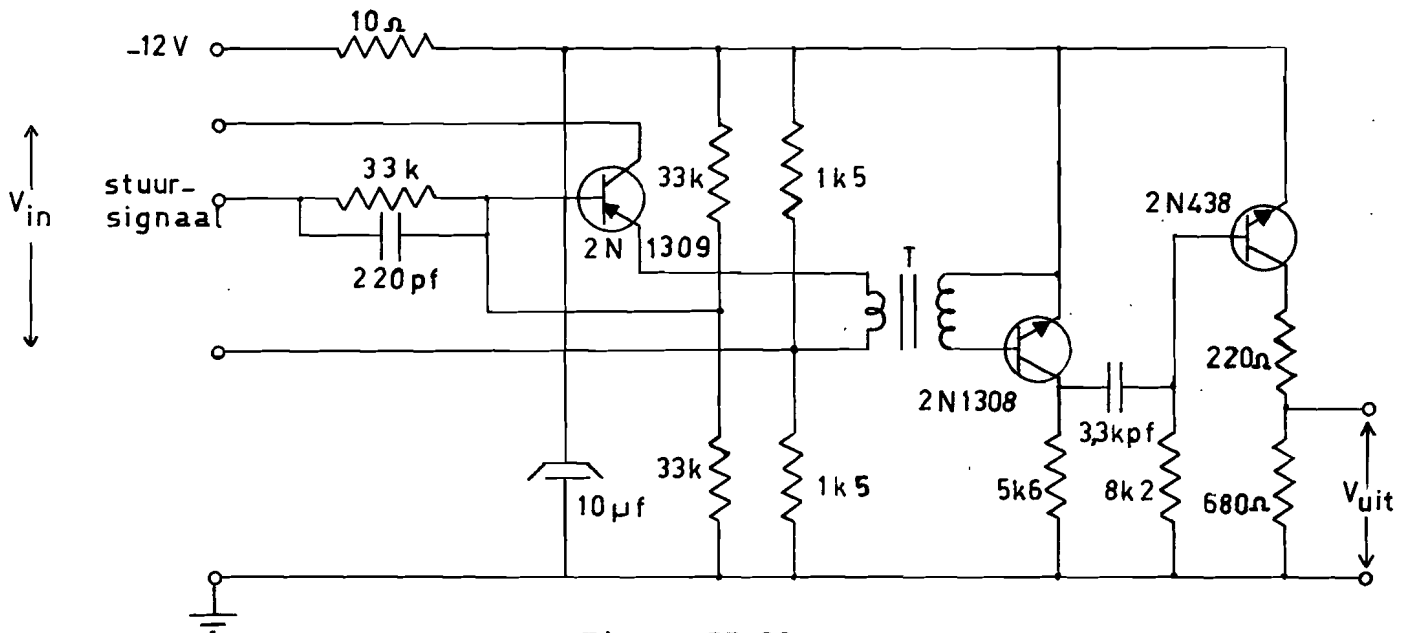


Figuur II-24  
Niveau detector.

We dimensioneren de weerstanden  $R_1$  en  $R_2$  zó dat stroompulsen met een amplitude groter dan die van het stoorsignaal nog niet voldoende stroom kunnen leveren om de transistor dicht te zetten. De informatiepulsen zijn nog groter en kunnen dat wel. We krijgen dan over de collectorweerstand positief gaande spanningspulsen van 12 V. (zie fig.II-25). Met behulp van een spanningsdeler kunnen we dan pulsen van de gewenste amplitude verkrijgen, die een "one-

shot" sturen. Het is mogelijk om twee leesdetectors te monteren op een plaatje dat gebruikt kan worden in combinatie met een Varipak-rek. (zie fig.II-26).

Figuur II-25  
Uitgangssignalen van de  
leesdetector.  $t = 5 \mu\text{s/cm}$



Figuur II-22  
Principeschema van de leesdetector  
 $T = \text{trafo.}$   
Kerntype K 528240  
Prim. 7 wdn.  
Sec. 30 wdn.  
Alle weerstanden zijn  $0,25\text{ W.}$



Figuur II-26.  
Twee leesversterkers gemonteerd op een insteekplaat.

HOOFDSTUK IIILogische Schakelingen voor de Integrator.A. Organisatie van de rekenschakeling.

In het voorgaande is reeds opgemerkt dat de berekeningen in serie zullen worden uitgevoerd. In hoofdstuk I-F hebben we gezien dat we voor zover het rekenkundige bewerkingen betreft kunnen volstaan met uitsluitend optellen. We tellen dan allereerst de eerste bit van  $y_{n-1}$  op bij die van  $\Delta y_n$ . Zodra dat resultaat bekend is, kunnen we dat optellen bij de eerste bit van  $r_{n-1}$ , enz. De beide optellingen liggen dus één elementaire cyclustijd verschoven en worden overigens gelijktijdig uitgevoerd. We bedenken hierbij dat  $r$  altijd positief is en er dus geen tekenbit voor  $r$  hoeft te staan. We hebben nu dus twee optellers nodig. De uit te voeren operaties zijn in een tijd-volgordetabel samengevat, waarin de indices de plaats van de bit aangeven, te beginnen bij de minst significante.

1. a. lees  $y_0$  en  $\Delta y_0$   
tel onmiddellijk op:  
 $y_0 + \Delta y_0 = y'_0$  en onthoudt eventuele carry  
b. schrijf terug  $y'_0$
2. a. lees  $y_1$  en  $\Delta y_1$   
lees  $r_0$   
tel onmiddellijk op:  
 $y_1 + \Delta y_1 + \text{carry} = y'_1$  en onthoudt de carry's  
 $r_0 + y'_0 = r'_0$   
b. schrijf terug  $y'_1$  en  $r'_0$
3. a. lees  $y_2$  en  $\Delta y_2$   
lees  $r_1$   
tel op:  $y_2 + \Delta y_2 + \text{carry} = y'_2$  en onthoudt de carry's  
 $r_1 + y'_1 + \text{carry} = r'_1$   
b. schrijf terug  $y'_2$  en  $r'_1$
- k+1. a. lees  $y_k$  en  $\Delta y_k$   
lees  $r_{k-1}$   
tel op:  $y_k + \Delta y_k + \text{carry} = y'_k$  en onthoudt de carry's  
 $r_k + y'_k + \text{carry} = r'_k$ 

---

b. schrijf terug  $y'_k$  en  $r'_k$

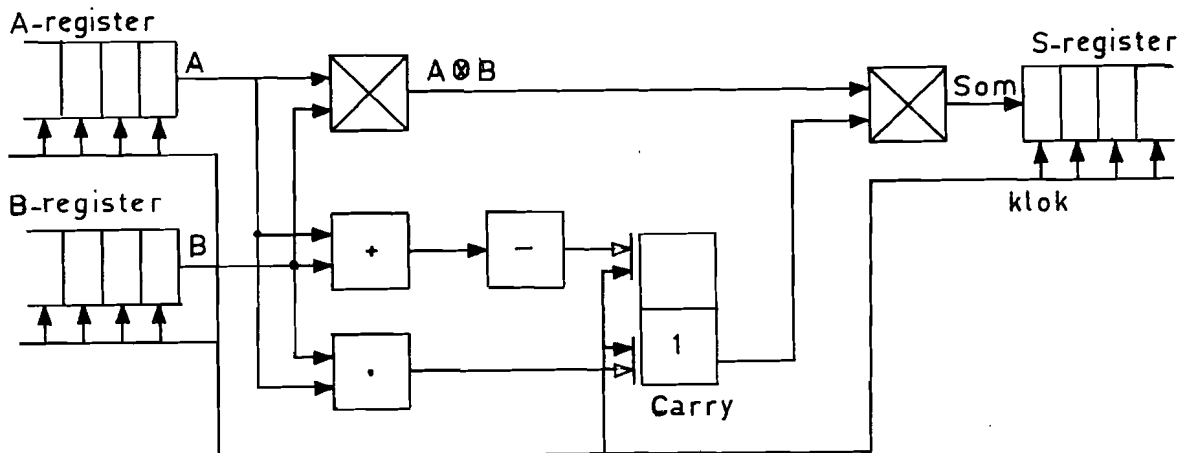
B. De opteller

Daar de informatie in serie verwerkt wordt, zullen we een serie-opteller nodig hebben. De Boolse formules voor een binaire optelling zijn:

$$\text{Som} = \bar{A}B + A\bar{B} = A \oplus B$$

$$\text{Carry} = AB$$

waarin A en B binaire cijfers zijn. Deze logische functies worden gerealiseerd door een tweevoudige "exclusive-or" en een tweevoudige en-poortschakeling. De som is echter pas volledig als de eventuele carry van de direct aangrenzende minder significante bitpositie ook bij het resultaat wordt opgeteld. Daarom zijn twee exclusive-orschakelingen nodig om de totale som van A, B en de carry C te genereren. De beide exclusive-orschakelingen worden dan achter elkaar geschakeld, terwijl de carry gedurende een cyclustijd in de flipflop bewaard wordt (zie fig. III-1).

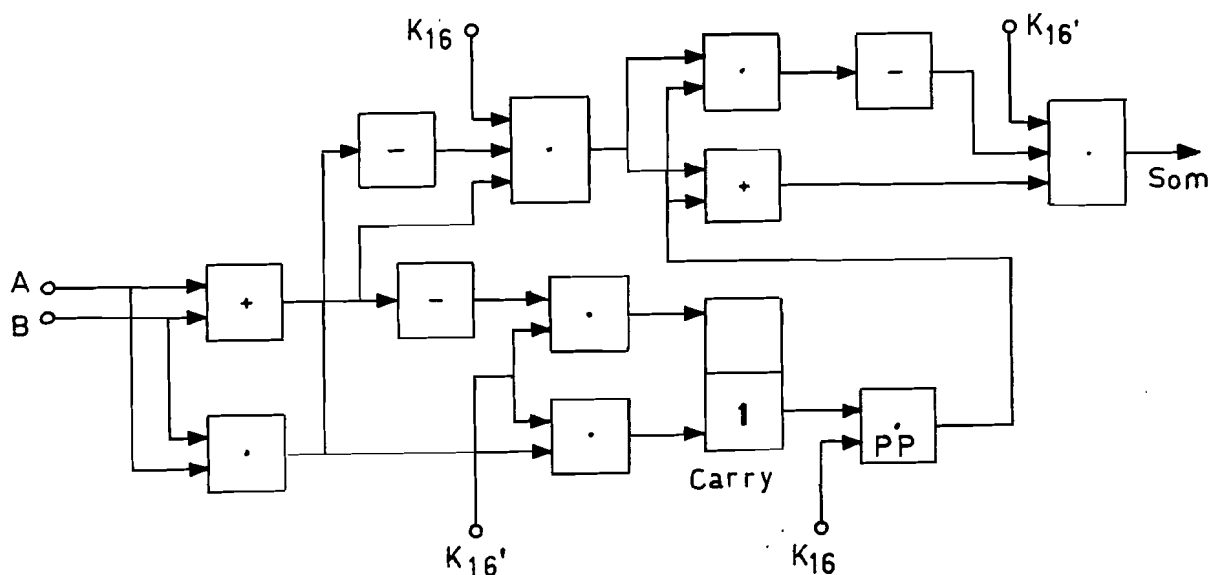


Figuur III-1.  
De serie-opteller.

De opteller functioneert dan als volgt. Stel dat A en B in de respectievelijke registers staan. Als we nu een klokpuls aan de schakeling toevoeren, dan wordt de som  $\bar{A}B + A\bar{B}$  op de meest linkse plaats in het somregister geschreven. De carry-flipflop wordt in de "0" of "1"-stand gezet, afhankelijk van de toestand der controle-inputs. Als  $AB = 1$ , dan wordt de flipflop in de "1"-stand gezet, zodra echter  $AB = 0$  valt hij bij de volgende klokpuls terug naar "0".

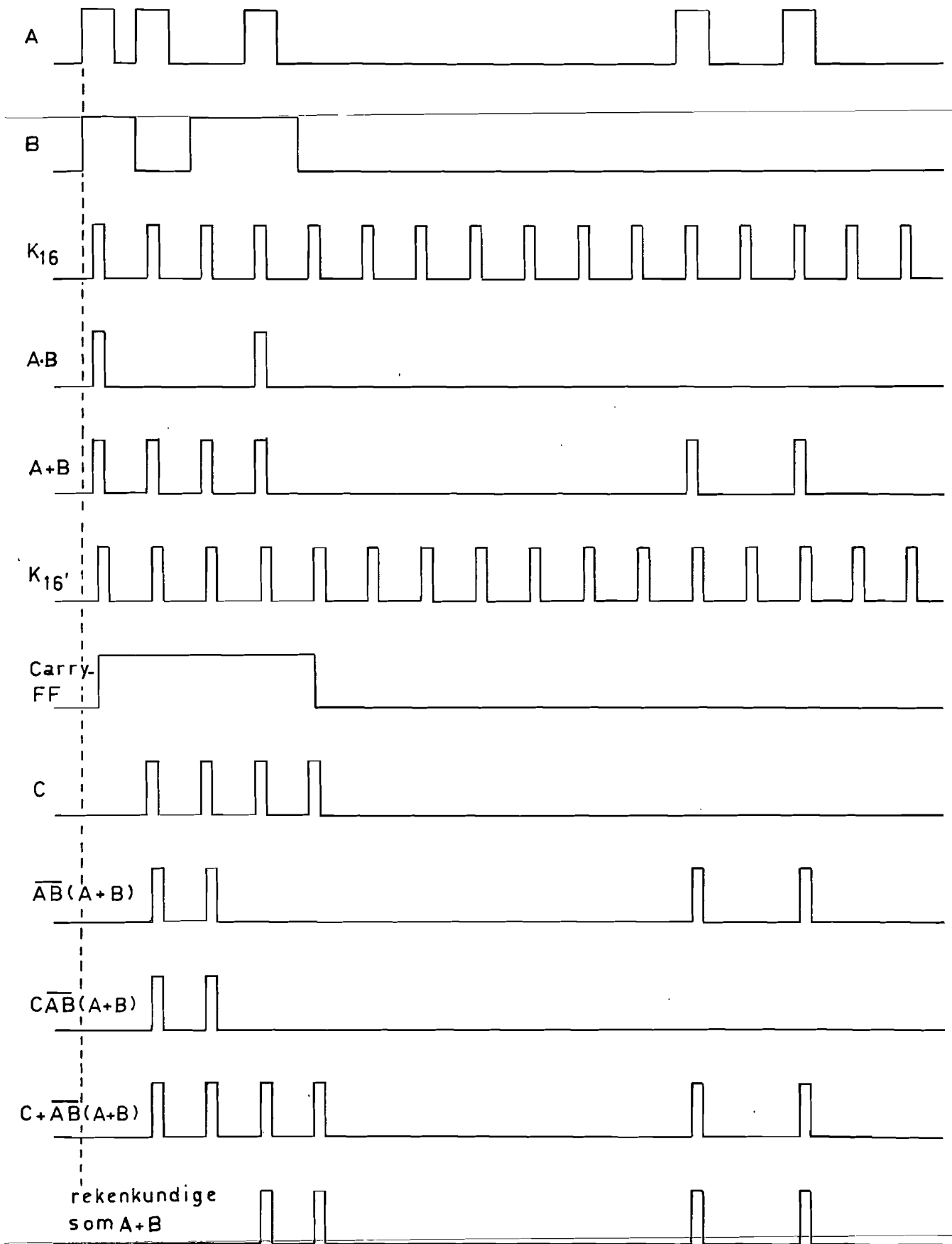
Voor de registers kan men bijvoorbeeld schuifregisters nemen. Wij gebruiken echter een kerngeheugen (zie hoofdstuk II) waaruit we de informatie krijgen in de vorm van pulsen, dit in tegenstelling tot de informatie uit een flipflop-schuifregister, die meer gekarakteris-

seerd is door het niveau. In het laatste geval beschikken we dan ook gedurende de gehele cyclustijd over de informatie, in het eerste slechts gedurende een gedeelte van de cyclustijd. Het heeft enig voordeel met betrekking tot de tijd om de techniek met pulsen toe te passen. We zullen dan over een signaal moeten beschikken dat de informatie bevattende tijdstippen nauwkeurig aangeeft. We geven de pulsen uit de leesdetector een duur van ca.  $3 \mu s$  met behulp van een one-shot. De  $\Delta y$ -signalen komen uit een  $\Delta y$ -opteller of een outputregister, d.w.z. een schuifregister. De pulsbreedte bedraagt dan  $5 \mu s$ . Aangezien de C 900 te traag is voor deze techniek, realiseren we de exclusive-or-functie met poorten:  $A \oplus B = \overline{AB}(A+B)$ . Met behulp van en-poorten en het kloksignaal lezen we de informatie op het juiste moment en schrijven deze direct terug in het kerngeheugen. De schakeling voor de  $y + \Delta y$ -opteller wordt dan als in fig. III-2.



Figuur III-2.  
Snelle serie-opteller.

In het pulsdigram van figuur III-3 is uitgewerkt hoe de serie-opteller twee binaire getallen A en B verwerkt. We zien daar o.a. in dat de carry door de pulspoort een cyclustijd vertraagd wordt. Het tijdstip waarop de informatie via de registerkeuze (zie hoofdstuk IV) in het kerngeheugen wordt teruggeschreven, wordt bepaald door het kloksignaal  $K_{16}$ . Voor de optellingen  $y_{i-1} + \Delta y_i$  en  $r_{i-1} + y_i$  worden twee afzonderlijke optellers gebruikt zoals die hier zijn beschreven.



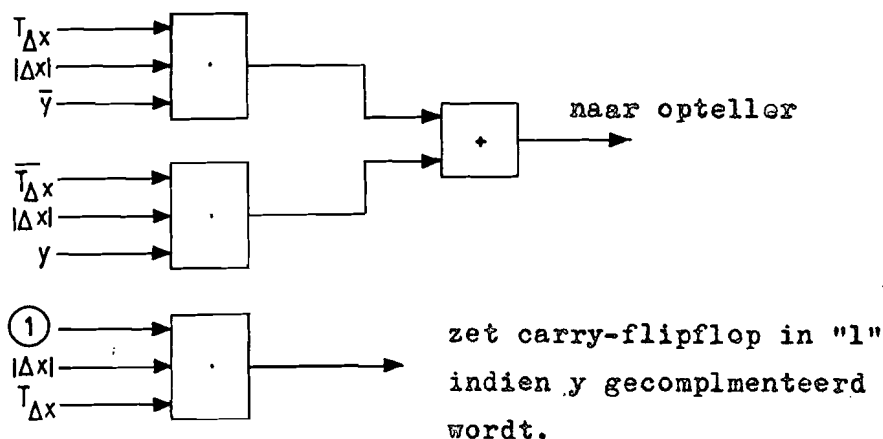
Figuur III-3  
Pulsdiagram voor de serie-opteller.

C. De beslissingseenheid.

In hoofdstuk I-B is beschreven dat de waarde van de ingangsgrootheid  $x$  bepaalt of en hoe  $y$  naar het R-register getransporteerd moet worden. Indien echter  $y$  afgetrokken moet worden, tellen we toch op maar complementeren dan eerst. Dit betekent dan in feite dat de waarde van  $\Delta x$  moet bepalen of  $y$  gecomplementeerd moet worden of niet. Samenvattend moet  $\Delta x$  voor de volgende bewerkingen zorgdragen:

$$\Delta x = \begin{cases} 01 & : y \text{ onveranderd bij } r \text{ optellen} \\ 00 & : \text{voor } y \text{ de waarde } 0 \text{ doorgeven} \\ 11 & : y \text{ complementeren en optellen} \end{cases}$$

Deze functies kunnen we eenvoudig realiseren met behulp van een poorten-systeem: we inhiberen  $y$  als  $\Delta x = 0$ , laten  $y$  door als  $\Delta x = 1$  en inverteren  $y$  als  $\Delta x = -1$ . Als we  $y$  inverteren moeten we echter nog een 1 bij het resultaat optellen. Dit doen we door dan de carry-flipflop alvast in de "1"-stand te zetten.



Figuur III-4.

De beslissingseenheid.

Voordat  $y$  de beslissingseenheid passeert, geven we de pulsen door middel van twee one-shots een pulsbreedte van  $3 \mu s$  en een vertraging, dusdanig dat de totale vertraging gelijk is aan één elementaire cyclustijd. De optellingen  $y + \Delta y$  en  $y + r$  worden dan in dezelfde tijd uitgevoerd, zoals reeds eerder is uitgelegd.

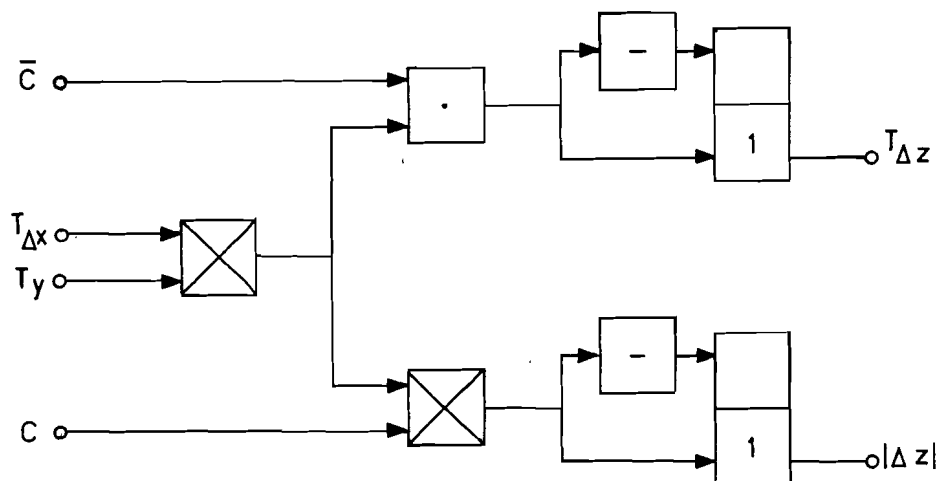
D. De output-detector.

In hoofdstuk I-E zijn de formules afgeleid voor het uitgangssignaal:

$$T_{\Delta z} = (T_{\Delta x} \otimes T_y) \bar{C}$$

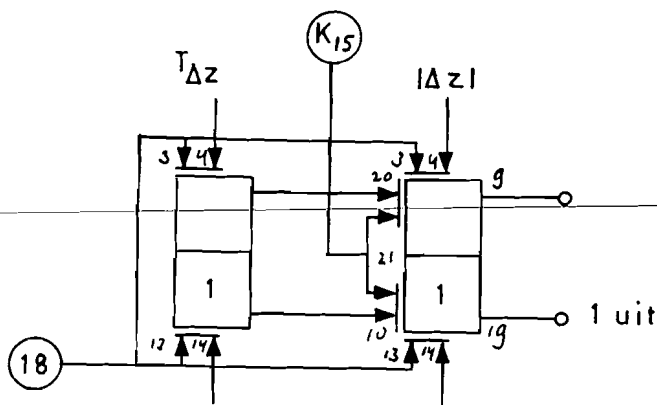
$$|\Delta z| = T_{\Delta x} \otimes T_y \otimes \bar{C}$$

Deze waarden moeten aan het eind van iedere integratiestap bepaald worden en in een register geplaatst worden, om ze tijdens de volgende integratiestap te kunnen gebruiken. Omdat  $\Delta z$  drie waarden kan aan-



Figuur III-5  
Logische schakeling voor  
outputdetectie.

nemen zijn er minimaal twee binaire elementen nodig om deze waarden eenduidig te kunnen onthouden. We gebruiken daarvoor een flipflopregister dat te splitsen is in een teken-flipflop en een modulusflipflop (zie fig. III-5). Om dit register te kunnen gebruiken als  $\Delta y$ -register van een andere integrator zijn de flipflops geschakeld als schuifregister. Op tijdstip 18 wordt de nieuwe informatie ingelezen, op tijdstip 2 van de volgende integratiestap krijgt de modulus-flipflop een schuifpuls zodat het teken in de modulus-flipflop staat. Als we dit register gebruiken als  $\Delta y$ -register van een integrator, dan krijgt de inhoud van het register automatisch een schaal-factor mee die correspondeert met de minst significante plaats van het Y-register van de betreffende integrator. Deze significantie kan zonder meer niet gewijzigd worden.

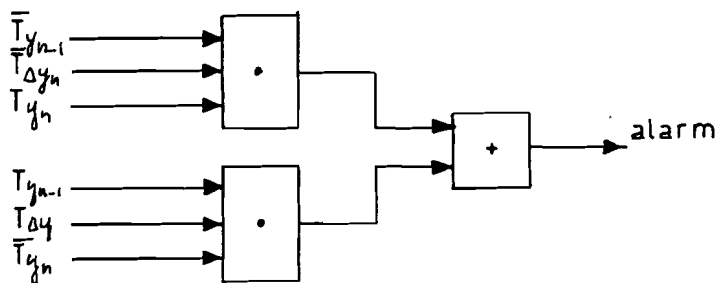


Figuur III-6  
Het output-register.

E. De Overloop-Detector.

Bij de behandeling van een probleem is het een belangrijke zaak om de voorkomende grootheden correcte schaalwaarden toe te kennen, Immers, alle getallen moeten binnen de registrarcapaciteit vallen. In hoofdstuk I.F is uitgelegd dat we een overloop van het Y-register kunnen constateren aan de hand van een tekencontrole:  $\bar{T}_{y_{n-1}} \cdot \bar{T}_{\Delta y_n} \cdot T_{y_n} + T_{y_{n-1}} \cdot T_{\Delta y_n} \cdot \bar{T}_{y_n} =$  alarm.

Deze functie wordt gerealiseerd door een poortschakeling als in figuur III.7. Met het signaal kan men dan een alarm instalatie bedienen die aangeeft welke integrator overloopt.



Figuur III-7  
De Overloop-Detector.

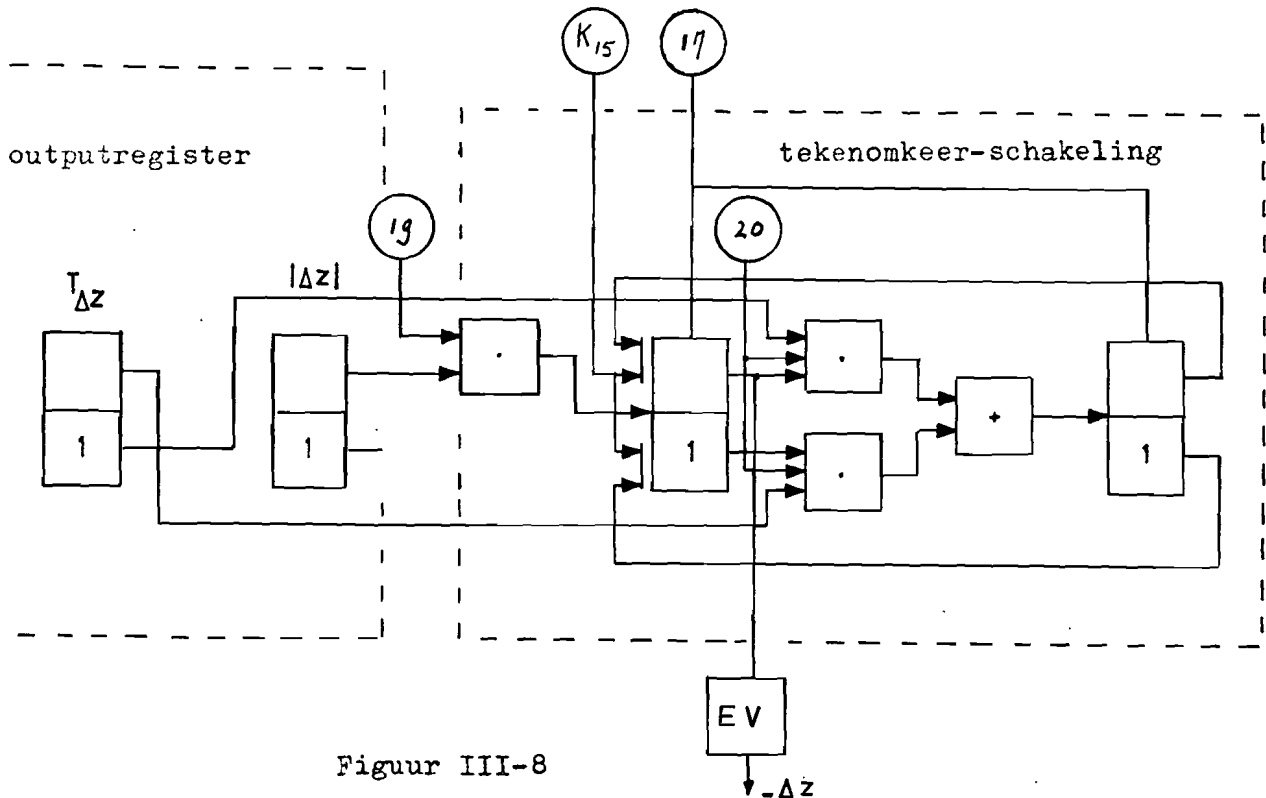
F. De inversie schakeling.

Voor sommige problemen is het noodzakelijk om het uitgangssignaal van teken te doen omkeren, b.v. voor het oplossen van de vergelijking  $y'' + y = 0$ . De tekenomkeerschakeling moet de volgende functies ver-



|          |                 |                  |
|----------|-----------------|------------------|
| richten: | $\Delta z = 01$ | $-\Delta z = 11$ |
|          | $\Delta z = 00$ | $-\Delta z = 00$ |
|          | $\Delta z = 11$ | $-\Delta z = 01$ |

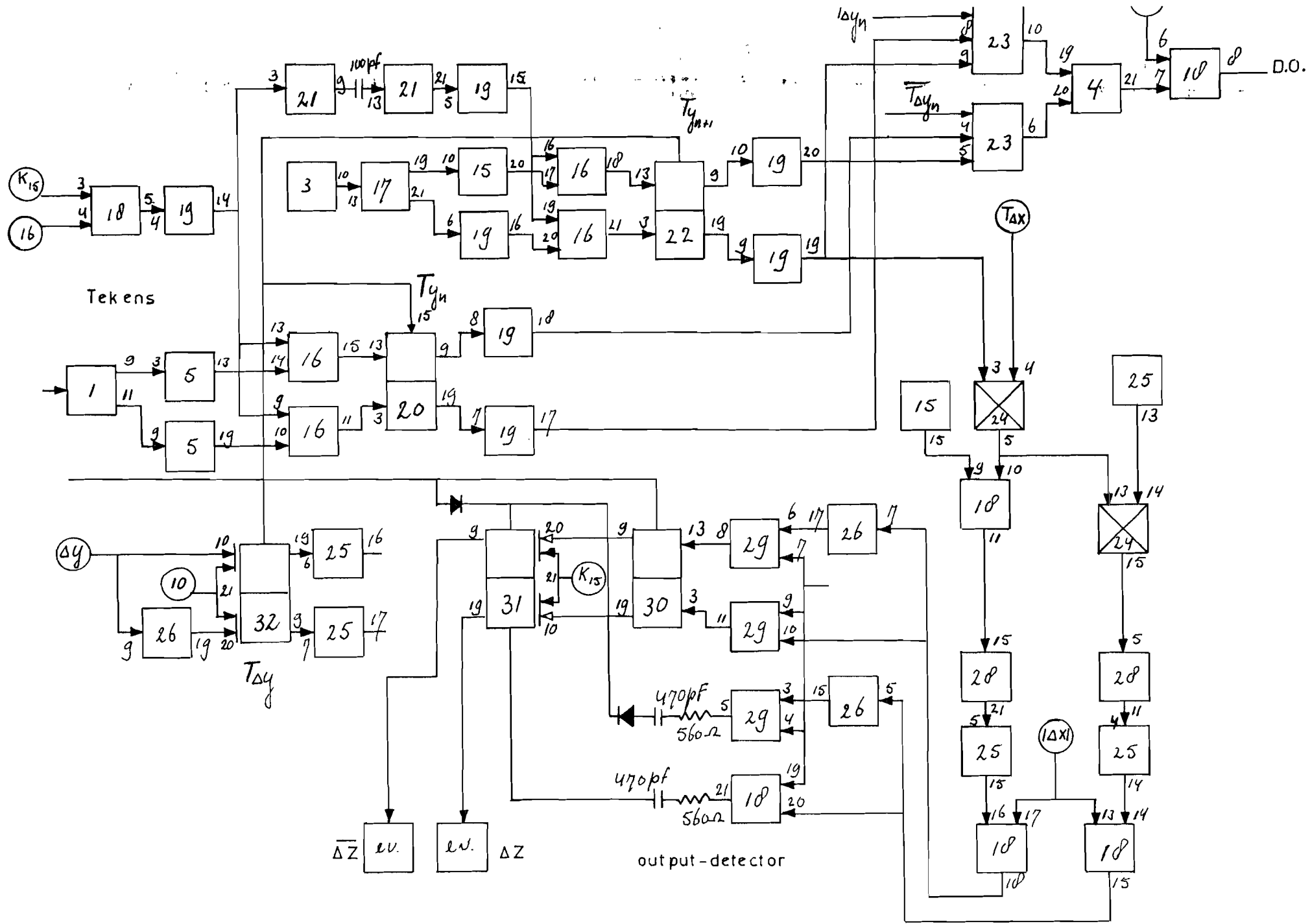
Dit betekent dat we  $\Delta z$  eerst moeten inverteren en er dan 1 bij op moeten tellen. De bewerking voeren we uit met behulp van de schakeling van figuur III-8.



Figuur III-8  
Teksomkering van  
binair getal.

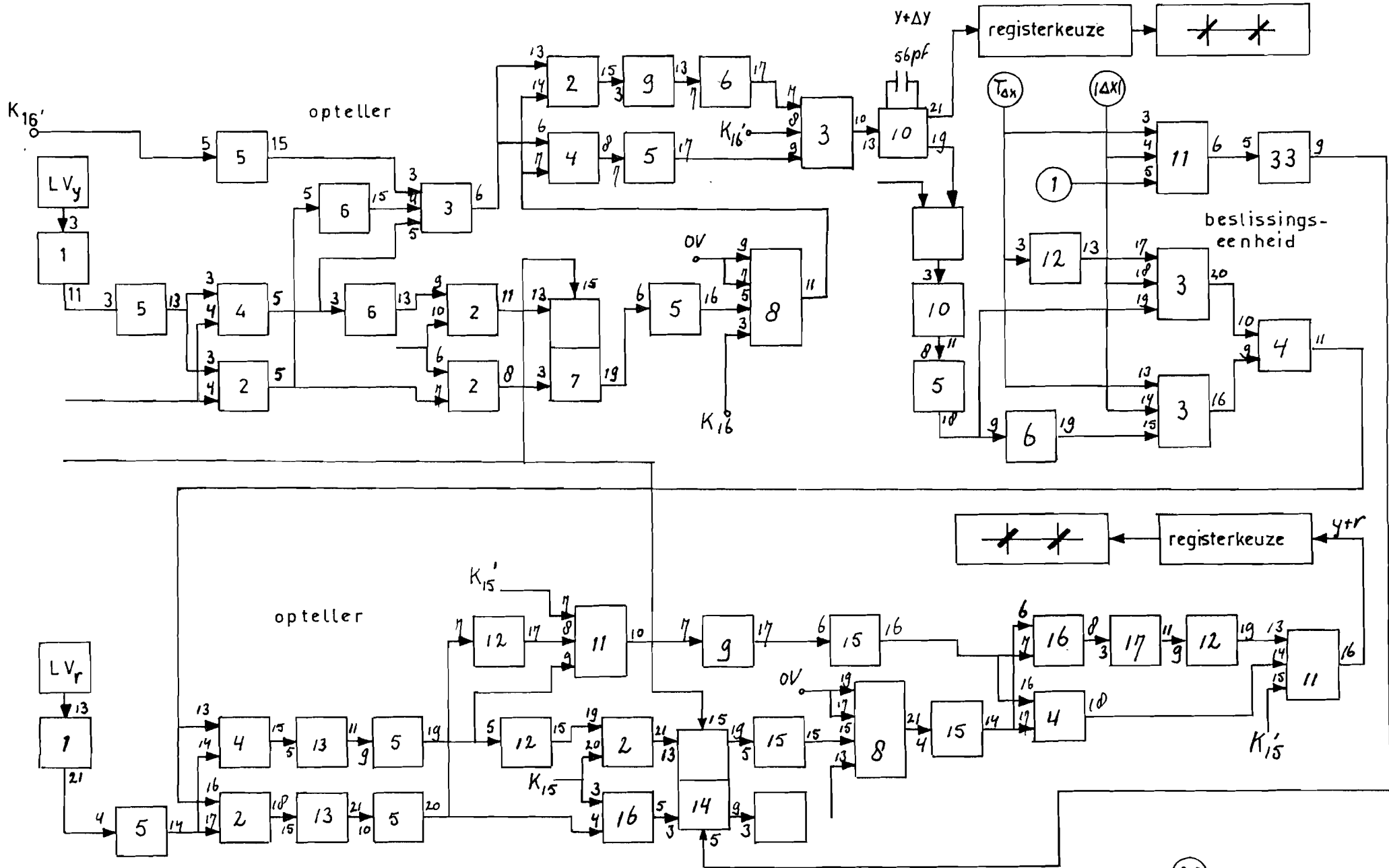
G. Het lezen van de tekens.

Voor het bepalen van de output en de overlooptdetectie dienen we te beschikken over de tekens van  $\Delta x$ ,  $y_{n-1}$ ,  $y_n$ ,  $\Delta y_n$  en de carry.  $T_{\Delta x}$  en de carry zijn reeds in flipflops aanwezig, de andere moeten apart uit het kerngeheugen gelezen worden. Aan de ingang van de optelschakeling kunnen we  $T_{y_{n-1}}$  lezen gedurende tijdstip (16). Aan de uitgang van de optelschakeling lezen we  $T_{y_n}$  tijdens hetzelfde tijdsinterval. Al deze informatie wordt in afzonderlijke flipflops gezet. Voor de schakeling hiervoor wordt verwezen naar het montageschema van de integrator figuur III-9.



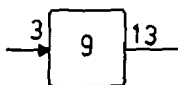
Figuur III-9.

Montageschema van de integrator.



Stuklijst bij figuur III-9.

|    |   |     |                         |
|----|---|-----|-------------------------|
| 1  | C | 020 | one shot                |
| 2  | C | 542 | en-poort tweev.         |
| 3  | C | 543 | en-poort drief.         |
| 4  | C | 502 | of-poort tweev.         |
| 5  | C | 300 | emittervolger PNP       |
| 6  | C | 240 | inverter                |
| 7  | C | 040 | flipflop                |
| 8  | C | 443 | puls-en-poort drief.    |
| 9  | C | 100 | pulsversterker          |
| 10 | C | 020 | one shot                |
| 11 | C | 543 | en-poort drief.         |
| 12 | C | 240 | inverter                |
| 13 | C | 200 | niveau-regenerator      |
| 14 | C | 040 | flipflop                |
| 15 | C | 300 | emittervolger PNP       |
| 16 | C | 542 | en-poort tweev.         |
| 17 | C | 020 | one shot                |
| 18 | C | 542 | en-poort tweev.         |
| 19 | C | 300 | emittervolger           |
| 20 | C | 040 | flipflop                |
| 21 | C | 020 | one shot                |
| 22 | C | 040 | flipflop                |
| 23 | C | 543 | en-poort drief.         |
| 24 | C | 900 | exclusive-or            |
| 25 | C | 300 | emittervolger           |
| 26 | C | 240 | inverter                |
| 27 | C | 503 | of-poort drief.         |
| 28 | C | 200 | niveau-regenerator      |
| 29 | C | 542 | en-poort tweev.         |
| 30 | C | 040 | flipflop                |
| 31 | C | 080 | schuifregister-flipflop |
| 32 | C | 080 | schuifregister "        |
| 33 | C | 340 | reset-generator         |
| 34 | C | 110 | inversie pulsversterker |
| 35 | C | 200 | niveau-regenerator.     |



9 = nummer van de printplaat  
3 = nummer van de ingangsklem  
13 = nummer van de uitgangsklem

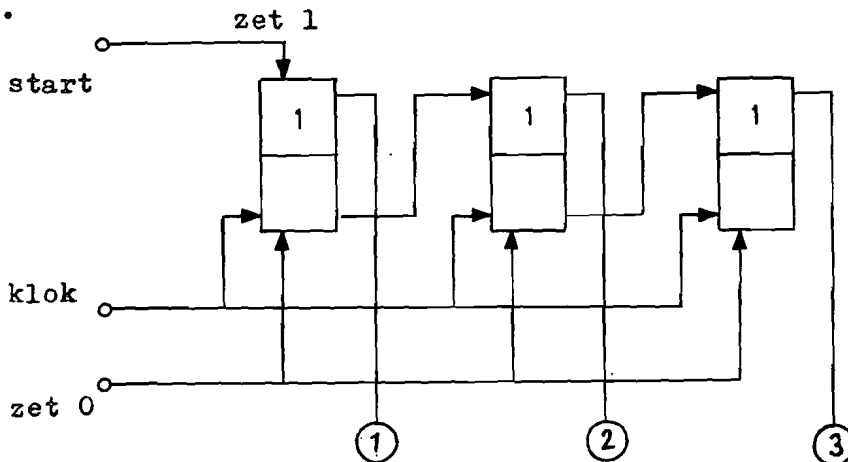
HOOFDSTUK IV

Besturing en selectie van integrator en registers.

A. Besturing.

Om de rekenkundige operaties, die de machine moet verrichten te kunnen leiden, zal het noodzakelijk zijn om over een besturend orgaan te beschikken. We moeten de berekening namelijk kunnen starten en stoppen, de beginwaarden moeten in de registers gezet kunnen worden terwijl men alle informatie indien gewenst moet kunnen verwijderen: men moet alle geheugenelementen in de "0" stand kunnen zetten. Ook moeten de geheugenkernen geselecteerd worden. Men dient dan in ieder geval over een paneel te beschikken met een stop-, start- en inleestoets. De schoon-toets is pas in een later stadium van belang.

Centraal in de besturingseenheid staat een zogenaamde lineaire teller. Deze wordt gevormd door een aantal flipflops. Van een flipflop wordt de "0"-uitgang verbonden met de "1"-zet ingang van de volgende flipflop. Alle flipflops ontvangen een kloksignaal op de "0"-zet ingang (zie figuur IV-1).

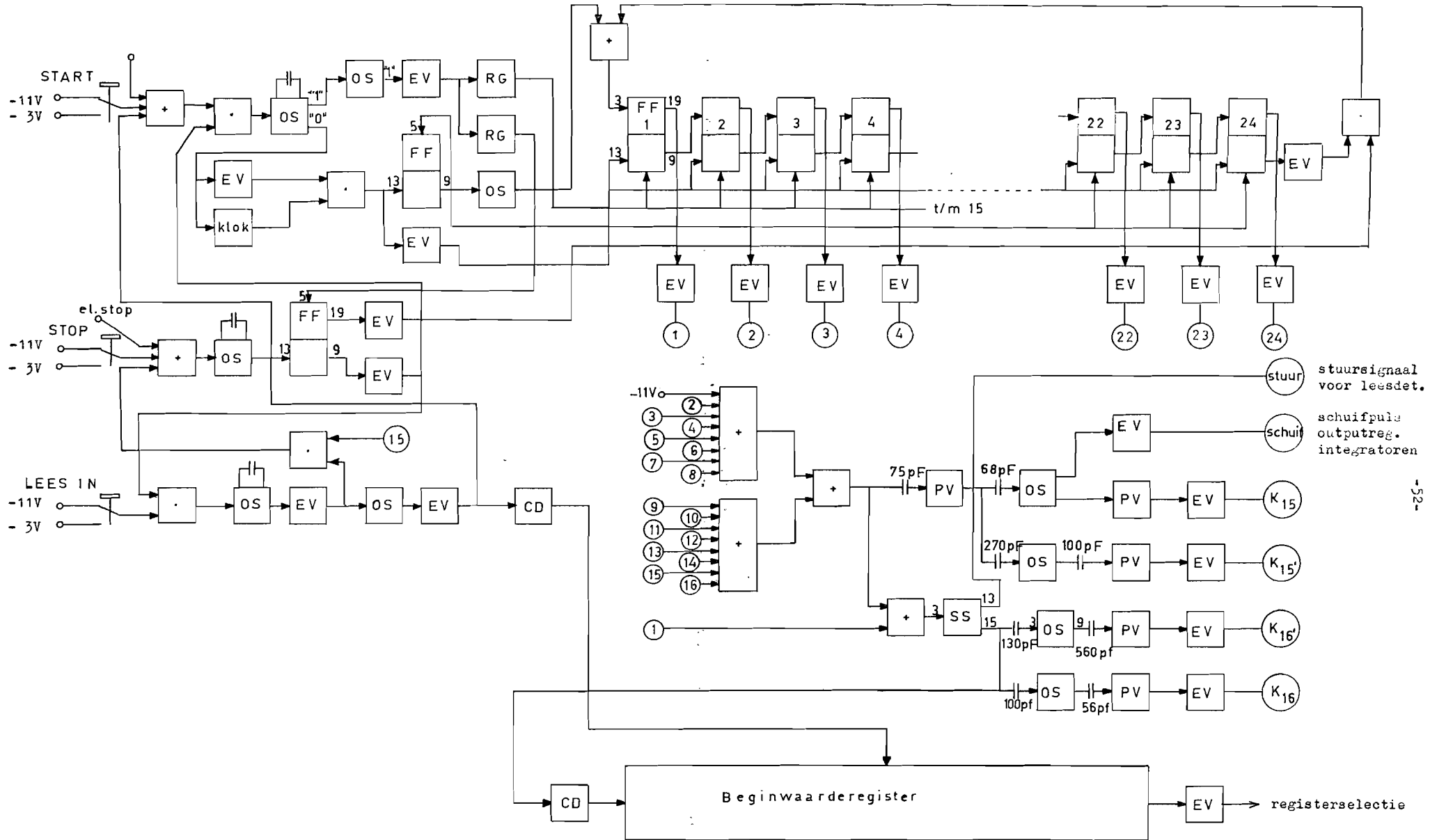


Figuur IV-1  
Een lineaire teller.

We gaan uit van de toestand dat alle flipflops in de "0"-stand staan. We zetten door middel van een startsignaal de eerste flipflop in de "1"-stand. Door de eerstvolgende klokpuls wordt deze flipflop weer in de "0"-stand teruggezet. Dit geeft echter een positief gaand signaal op de "0"-uitgang, zodat de tweede flipflop in "1" gezet wordt. Volgende

klokpulsen schuiven de "1" stand telkens een plaats op, zodat het rangnummer van de flipflop die in de "1" stand staat het aantal klokpulsen aangeeft dat is binnengekomen na het startsignaal. Is dit aantal groter dan het aantal flipflops, dan staan alle elementen weer in de "0" stand. Als we echter de "0" uitgang van de laatste flipflop koppelen met de "1"-zet ingang van de eerste, dan circuleert de "1" in de teller en verkrijgen we een zogenaamde ringteller. Deze ringteller gebruiken we nu als scanner, d.w.z. de signalen die deze flipflops afgeven, gebruiken we voor de besturing, van kerngeheugen en integrator. Een blokschema van de besturingseenheid vinden we in figuur IV-2. Met de signalen 1 tot en met 16 selecteren we de drijvers van het kerngeheugen, zodat de woordlijnen achtereenvolgens afgetast worden (vergelijk figuur II-7). Via of-poorten gesommeerd geven deze signalen ons pulstreinen bestaande uit vijftien en resp. zestien pulsen die de optellingen en het terugschrijven van de nieuwe informatie synchroniseren.

We moeten dus de ringteller op een willekeurig moment kunnen laten stoppen en starten, hetzij door middel van een druktoets, hetzij met behulp van een elektrische puls. Het eenvoudigst werkt de stopfunctie. We onderbreken voor het stoppen de terugkoppeling van flipflop 24 naar flipflop 1 door in de terugkoppellijn een en-poort op te nemen, waarvan we een ingang laag maken, zodat de poort spert. (We kunnen dus alleen stoppen nadat een elementaire integratiestap volledig is uitgevoerd.) We zetten daartoe een flipflop in de "0" stand door op de stopstoets te drukken. Achter deze toets is een one-shot geschakeld, die het prellen van de schakelaar opvangt en slechts één puls afgeeft. Daartoe moet de puls voldoende lang aanhouden. Via de of-poort kunnen we de berekening met een elektrische puls doen stoppen. Starten is slechts mogelijk nadat de machine eerst gestopt is. Hiermede voorkomen we dat door een foute bediening een integratiestap niet volledig uitgevoerd wordt terwijl de machine opnieuw start. De starttoets slaat eveneens een one-shot aan die het prellen opvangt. Een volgende one-shot geeft een korte puls en zet daarmee via twee reset-generatoren alle elementen van de ringteller in "0", terwijl een andere flipflop in "1" gezet wordt. Op de "0"-zet ingang van deze flipflop staat het kloksignaal, dat tijdens het "0" zetten geïnhibeerd wordt. Tijdens het afvallen van de startpuls wordt de klok gesynchroniseerd, waarop de flipflop in "1" gezet wordt en de eerste flipflop van de ringteller in "1" gezet wordt, zodat de teller gestart is. Een andere



Figuur VI-2.  
De besturingseenheid.

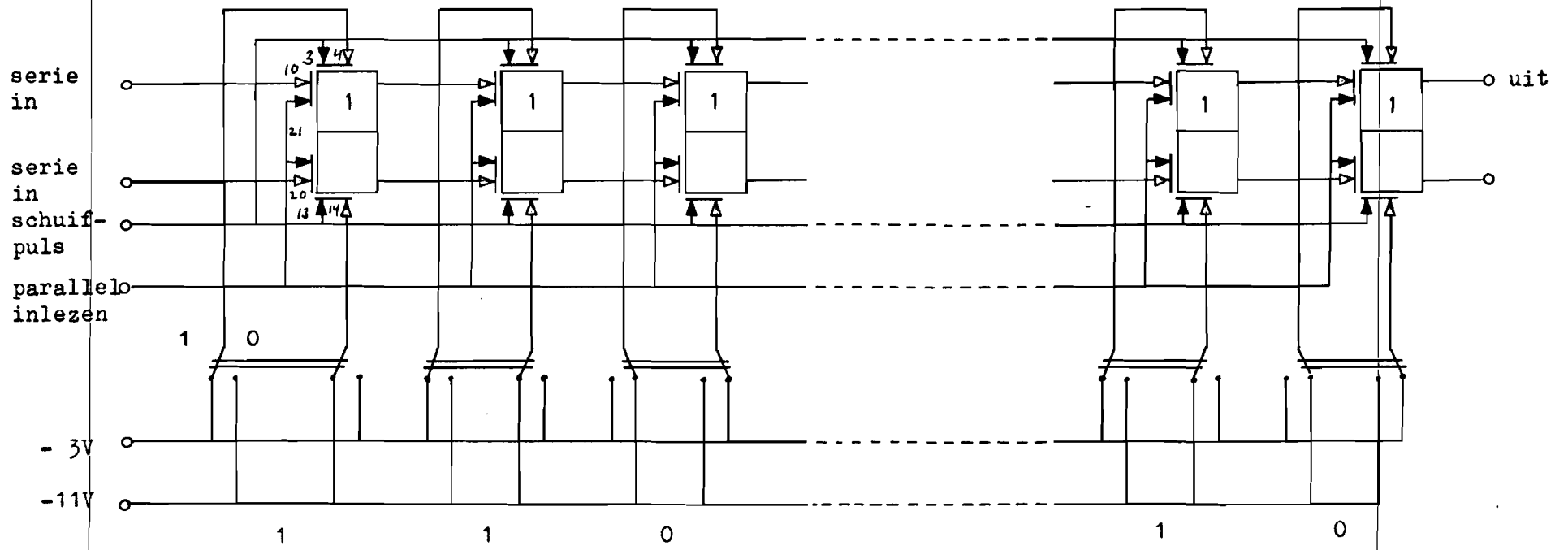
procedure is het inlezen van de beginwaarden.

Evenals bij een analoge rekenmachine moeten we de waarden van de verschillende variabelen op het tijdstip  $t = 0$  kunnen vastleggen. Deze waarden schrijven we in de Y-registers van de integrator waar de grootheid geaccumuleerd wordt. De beginwaarden worden daartoe eerst in een beginwaarderegister (i.c. een flipflop-schruifregister) genoteerd dat via een selectie aan betreffende integrator gekoppeld. We kunnen dan de informatie in de opteller schuiven (dus als  $y$  invoeren), zodat het getal als eindresultaat van een optelling wordt beschouwd en daarom in het kerngeheugenregister wordt opgeslagen. De beginwaarden kunnen vanuit een andere eenheid (b.v. een decimaal-binair-omzetter of een geheugen) via de parallelgangen van de schuifregister flipflops in het beginwaarderegister gevoerd worden. Bij wijze van tijdelijke oplossing kiezen we de getallen met behulp van zestien schakelaars die op het paneel zijn aangebracht. We kennen dan in volgorde aan elke schakelaar een significantie toe volgens de binaire code en kunnen de schakelaar in de "0" of in de "1" stand zetten (zie figuur IV-3).

Verder beschikken we over de kloksignalen  $K_{16}$ ,  $K_{16}'$ ,  $K_{15}$  en  $K_{15}'$ , die voor zover het  $K_{16}$  en  $K_{16}'$  betreft, afkomstig zijn van de schakeling SS (zie figuur II-21).  $K_{16}'$  is ca.  $0,5 \mu s$  vertraagd t.o.v.  $K_{16}$  en overigens identiek. Hetzelfde geldt voor de relatie tussen  $K_{15}'$  en  $K_{15}$ . Met de one shots kiezen we de juiste vertragingen, met de pulsversterkers regelen we de pulsbreedte op  $0,5$  à  $1 \mu s$ . Daartoe zijn capaciteiten van in de tekening aangegeven waarden aan de ingangen van deze schakelingen aangebracht. Het tijdstip van de kloksignalen  $K_{16}'$  en  $K_{15}'$  is dusdanig gekozen dat de lees-schrijfcyclustijd in twee gelijke stukken wordt verdeeld. Dit houdt in dat zij ca.  $2,5 \mu s$  na de voorflank van de leesstroompuls komen. Met  $K_{16}$  schuiven we ook de beginwaarden uit het register in de integrator. Om het inschrijven in 't kerngeheugen mogelijk te maken, zullen we ook de scanner moeten starten. De "lees-in" puls bedient dus ook de startschakeling. Het herhalen van de inschrijfcyclus heeft geen zin. De inleesschakeling geeft dan ook een stopsignaal na het beëindigen van de procedure.

Wat het schoonmaken van de registers en andere informatie bewarende elementen betreft kan men opmerken dat men bij bestaande machines een keuze heeft uit tal van combinaties. Het is in dit stadium nog niet te overzien welke combinaties voor een centrale bediening in aanmerking komen. Voorlopig wordt het "schoonvegen" dan ook verricht door in elke integrator "0" te schrijven met behulp van het beginwaarderegister. Alle andere ge-





Figuur IV-3.

Het beginwaarderegister.

De getallen worden gekozen met behulp van de standenschakelaars.

heugenelementen registeren dan óók de waarden 0.

De gebouwde machine wordt geprogrammeerd door draadverbindingen op een z.g. "patch-panel" aan te brengen.

Als men echter de machinetijd ten volle wil benutten, kan men het apparaat beter zó maken, dat het programma in codevorm kan worden medegedeeld. Bij moderne D.D.A.'s is dit meestal wel mogelijk.

#### B. De integratorselectie.

Met de integratorselectie kiezen we de integrator waar een betreffende beginwaarde in het Y-register moet worden geschreven.

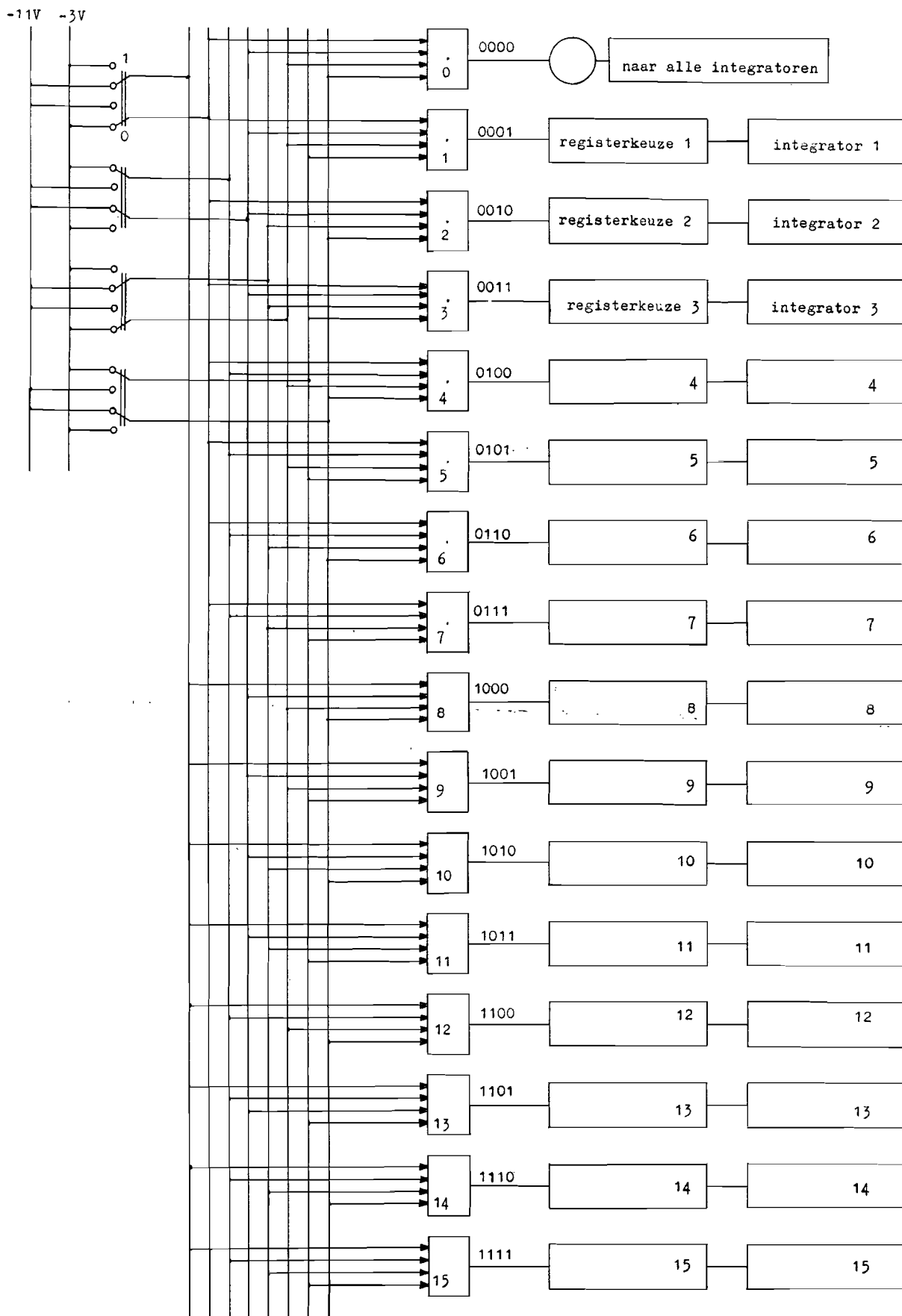
We nummeren dan de integratoren van 1 tot een gewenst aantal en coderen deze rangnummers binair. We kunnen de nummers dan kiezen met behulp van standen-schakelaars, die op de ingangsklemmen van met de integratoren corresponderende en-poorten zijn aangesloten. In figuur IV-4 is een selectie getekend voor vijftien integratoren. Bij een bepaalde standencombinatie van de schakelaars kan slechts één poort hoog signaal afgeven. De andere poorten geven laag signaal. Elke integrator beschikt ook nog over een registerkeuze; we moeten immers of het beginwaarderegister of het Y-register aan de reken-eenheid koppelen terwijl we moeten bepalen of de inhoud van het Y-register bij die van het R-register geteld moet worden of dat de beginwaarde 0,5 moet worden geschreven. Het niveau van het signaal dat de selectiepoort afgeeft beslist hierover.

Hoog niveau betekent dat het beginwaarderegister aan de integrator gekoppeld wordt en voor r de waarde 0,5 geschreven wordt. Bij laag niveau wordt de integrator weer aan zijn eigen registers gekoppeld.

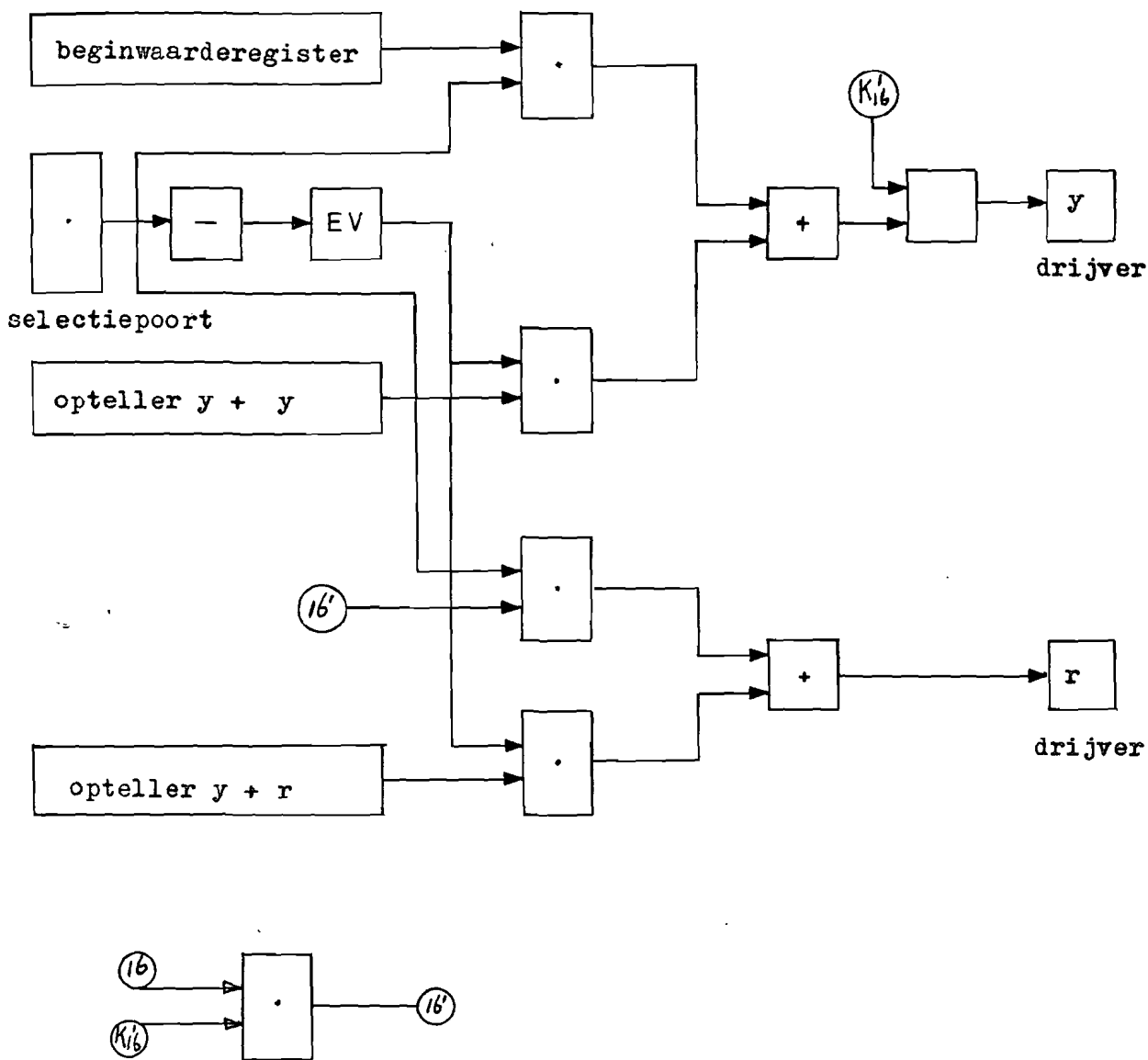
Daar de getallen in het R-register per definitie kleiner zijn dan 1 en dus geschreven kunnen worden als

$$\sum_{i=1}^n a_i 2^{-i}$$

komt 0,5 overeen met een 1 op de meest significante plaats in het register. De schakeling is gegeven in figuur IV-5. Het schrijven in het geheugen synchroniseren we met  $K_{16}'$ .



Figuur IV-4.  
Integratorselectie.



Figuur IV-5  
De registerkeuze.

HOOFDSTUK V

Output-apparatuur.

A. Digitaal-analoog omzetter.

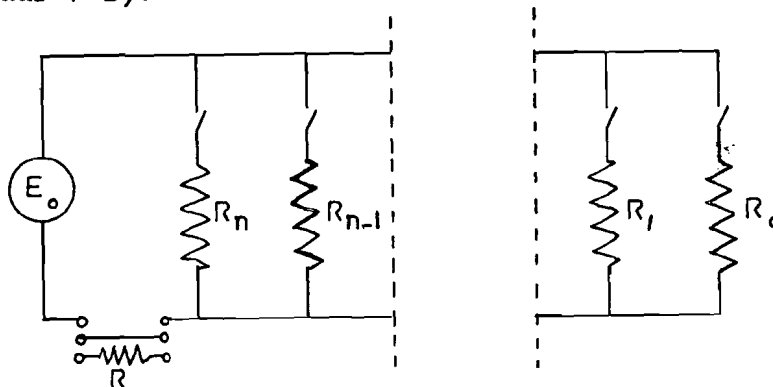
Om tegemoet te komen aan de behoefte om snel een overzicht te kunnen krijgen van het verloop van een berekening is de mogelijkheid geschapen om dit verloop te registreren m.b.v. een oscillograaf of een schrijver. Daar de grootheden in de machine uitsluitend in digitale vorm aanwezig kunnen zijn, zal hiertoe een apparaat vereist zijn, dat de omzetting van informatie in getalvorm naar informatie in de vorm van continue signalen bewerkstelligt: een digitaal-analoog omzetter.

Er zijn verschillende manieren waarop men een omzetting kan realiseren. Alle omzeters echter produceren een stroom of een spanning die evenredig is met het te converteren getal. Dit getal is afkomstig uit een digitale machine, die enige tijd nodig heeft om een nieuwe getalwaarde te produceren. We krijgen dus gedurende een cyclustijd een signaal dat correspondeert met het vorige getal. De functie wordt dan door rechthoeken benaderd.

Het systeem waarop de keuze is gevallen is direct aan te passen op de standaard flip-flops en brengt geen enkele beperking met zich mede ten aanzien van de herhalingsfrequentie waarmee de flip-flops bedreven worden. De omzetter aan beide gekozen dimensionering een maximale output spanning cab ca. 12 mV volt leveren.

Voor de beschrijving van het circuit gaan we uit van een constante spanningsbron  $E_0$ .

We bouwen een weerstandsnetwerk op waarvan de elementen  $R_j$  met behulp van een schakelaar  $S_j$  aan de spanningsbron kunnen worden gekoppeld (zie figuur V-1).



Figuur V-1  
De conversie kan plaats vinden  
m.b.v. dit weerstandsnetwerk.

We kiezen de weerstandswaarden zo dat de stroom die door een ingeschakelde weerstand vloeit in verhouding tot de stromen door de andere weerstanden in grootte correspondeert met de significante van de te converteren bit.

Stel dat we het getal  $p = \sum_0^n a_j 2^j$  willen converteren. Als nu  $a_j = 0$ , dan staat de schakelaar  $S_j$  open, is  $a_j = 1$ , dan wordt hij gesloten. Stel nu dat de minst significante bit gerepresenteerd wordt door een stroom  $I_0$ , dan kiezen we  $I_j = \frac{E_0}{R_0}$ ,

$$\text{dan is } I_{\text{tot}} = \sum_0^n \frac{E_0 a_j 2^j}{R_0} = \frac{E_0}{R_0} \sum_0^n a_j 2^j = \frac{E_0}{R} \cdot p.$$

Hieruit zien we dat de totale stroom evenredig is met het te converteren getal  $p$ . Als we aan deze stroom een spanning willen koppelen, dan kunnen we de kortsluiting tussen de klemmen A en B vervangen door een weerstand  $R$ . Hierdoor wordt een fout geïntroduceerd die klein is als deze weerstand klein is t.o.v. de minimale resulterende weerstand van het netwerk (alle schakelaars gesloten). Deze is bij de gekozen waarden van  $R_j$  bij benadering gelijk aan  $\frac{1}{2}R_n$  ( $R_n$  is de kleinste weerstand in het netwerk). Als we  $R = 10\Omega$  nemen, dan is de fout die hierdoor ontstaat kleiner dan  $\frac{1}{4}\%$ , want in fig. V-2, die het schema van de complete omzetter geeft, zien we dat de waarde van  $R_n$  ca.  $8k\Omega$  bedraagt.

De gebouwde omzetter kan een binair getal van maximaal zeven bits omzetten. De minst significante bit vertegenwoordigt dan minder dan  $1\%$  van de maximale totale stroom door de weerstand  $R$ . Het heeft dus in verband met de bereikbare nauwkeurigheid geen zin om meer dan zeven bits om te zetten. Wel kunnen dan de stapjes door de schrijver geregistreerd worden. Indien men dit bezwaarlijk acht, kan men er toe overgaan om het aantal geconverteerde bits uit te breiden.

Voor de constante spanningsbron  $E_0$  in combinatie met de schakelaar  $S_j$  is een afzonderlijke flipflop voor elke plaats gekozen, waarbij het  $-3V$ -niveau als 1 is gedefinieerd en het  $-11V$ -niveau als 0 (register 2 in fig. V-2). Als de flipflop een 0 bevat, mag er geen stroom door de weerstand lopen. Daarom is een diode met grote sperweerstand aan de uitgang van de flipflop geschakeld, terwijl hij tevens met de kathode indirect verbonden is aan een niveau dat hoger is dan  $-11V$ . Dit niveau wordt verkregen met behulp van een zenerdiode OAZ 213 (punt  $y = -10,5V$ ).

Ook het 1-niveau van de flipflop wordt constant gehouden. Daartoe wordt het niveau afgesneden m.b.v. een germanium-diode die met de kathode aan een constante spanningsbron is gelegd. Deze wordt eveneens

m.b.v. een zenerdiode verkregen (punt  $x = 4V$ . De gebruikte zenerdiodes zijn ingesteld op 10 mA. Een voldoende stabiel niveau is dan gewaarborgd.

---

De converter-schakeling en de clipper zijn tezamen op een printplaatje gemonteerd. De nummering van de aansluitingen is in de figuur aangegeven.

Het inlezen van een getal geschiedt vanuit een schuifregister (register 1), waar de informatie inschuift. Aan het eind van iedere integratieperiode staat de informatie op de juiste plaats, d.w.z. de minst significante bit staat in de meest rechtse flip-flop en de meest significante bit in de meest linkse. Dan kan een puls B gegeven worden zodat de informatie in register 2 schuift.

De omzetter dient ook geschikt te zijn voor het omzetten van negatieve getallen. De tekenbit staat (op het juiste ogenblik) in de meest linkse flip-flop van register 1. Als daar een 0 staat en het getal dus positief is, dienen geen speciale maatregelen genomen te worden voor de conversie plaatsvindt. Is het getal echter negatief, dan kan men de absolute waarde van dit getal verkrijgen door een inversie toe te passen en een 1 bij de minst significante plaats van het getal in de rekenmachine, dat uit meer dan 7 bits kan bestaan, op te tellen. Als men dit laatste nalaat, dan zal de maximale fout die hierdoor kan ontstaan de waarde van de minst significante bit hebben. Dat gebeurt alleen als de carry die hierdoor zou moeten optreden net tot de zevende plaats of verder zou doorlopen. Men kan verwachten dat dit zelden voorkomt. In andere gevallen treedt geen extra fout op. We kunnen dus een invertteerpuls C aan register 1 geven voordat puls B gegeven wordt. Puls C wordt via de en-poort slechts doorgelaten als het getal negatief is, hetgeen gecontroleerd wordt door het niveau van de teken-flip-flop. Verder wordt door de teken-flip-flop en relais gestuurd dat, indien bekrachtigd, de uitgangsklemmen verwisselt, zodat het teken van de uitgangsspanning dan omkeert.

---

|       |         |
|-------|---------|
| DATUM |         |
| U     | AKKOORD |
| BLZ   | VAN     |

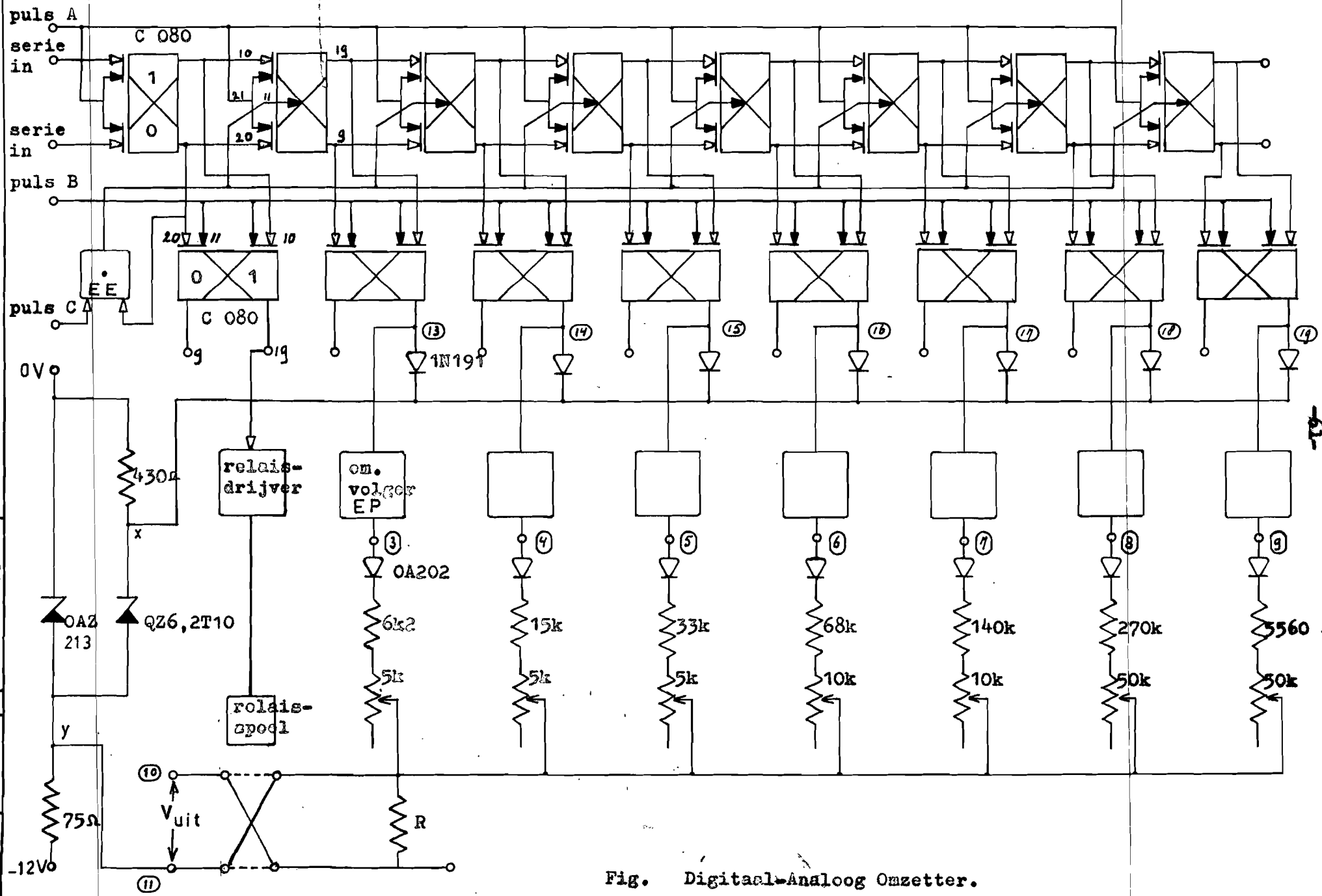


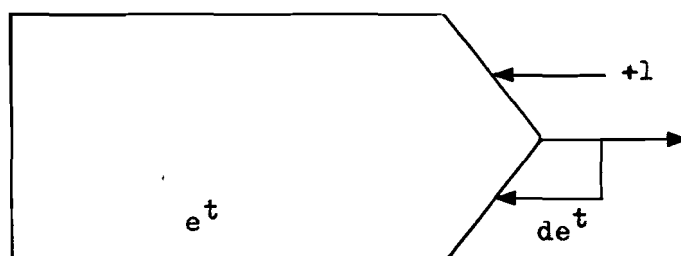
Fig. Digitaal-Analoog Omzetter.



HOOFDSTUK VI

Eenvoudige toepassingen van de D.D.A.

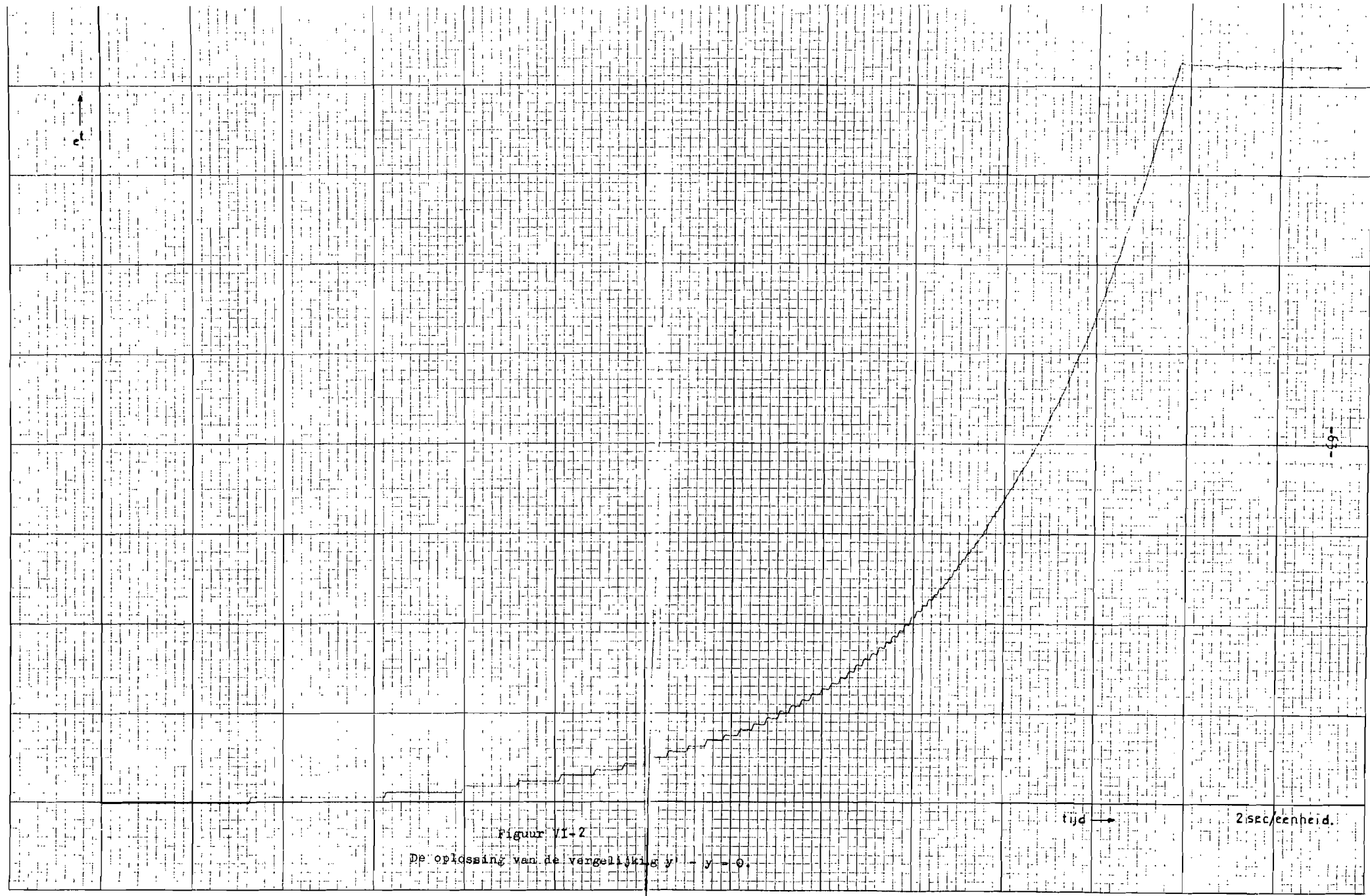
A. Het oplossen van de vergelijking  $y' - y = 0$ .



Figuur VI-1  
Het genereren van  $e^t$ .

Hiervoor hebben we slechts één integrator nodig waarvan we de uitgang terugkoppelen naar de ingang. Daar we naar de tijd integreren nemen we  $dX = +1$ . We schrijven een beginvoorwaarde in ongelijk aan nul en starten de machine. De oplossing wordt m.b.v. een schrijver geregistreerd (zie figuur VI-2). Dit blijkt een exponentiële functie te zijn.

Uit de kromme blijkt duidelijk het digitale karakter van de benadering. We kunnen de stapjes "onzichtbaar" maken door meer bits te converteren (zie hoofdstuk V).



Figuur VI-2

De oplossing van de vergelijking  $y' - y = 0$ .

tijd →

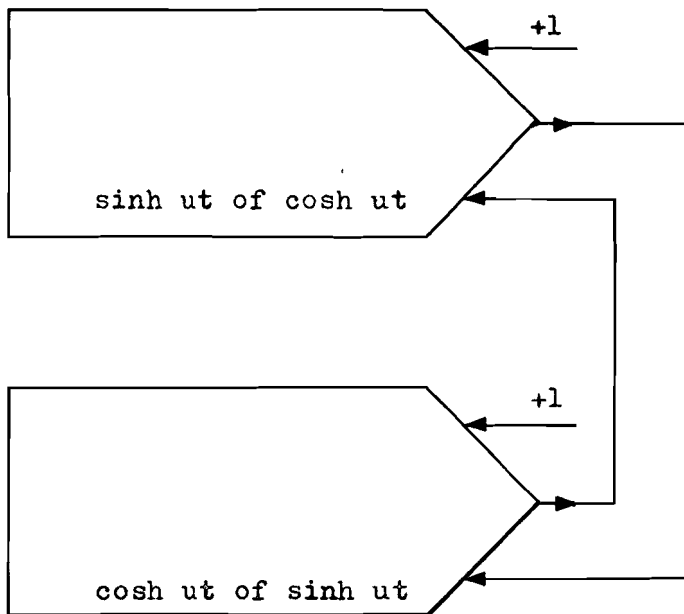
2 sec/eenheid.

B. Het oplossen van de vergelijking  $y'' - y = 0$ .

Om tot een oplossing hiervan te komen, moeten we  $y''$  twee maal integreren. Vervolgens identificeren we  $y''$  met  $y$ :  $y'' = y$ .

~~De schakeling wordt dan als in figuur VI-3. De algemene oplossing van de vergelijking is  $A \sinh ut + B \cosh ut$ .~~

De  $\sinh$  is met behulp van de schrijver geregistreerd (zie fig. VI-4).

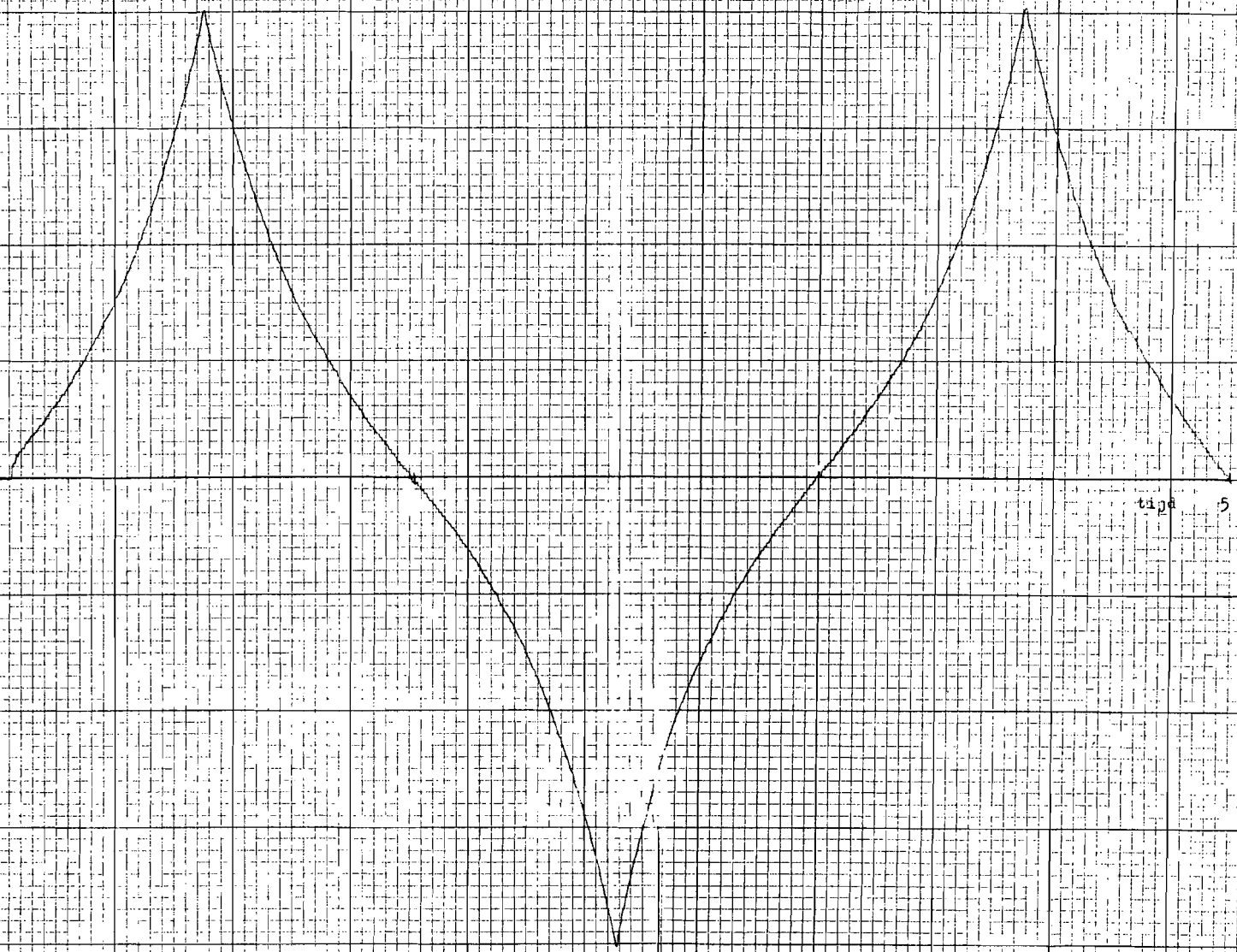


Figuur VI-3.

Het genereren van de  
hyperbolische functies.

Daar we naar de tijd integreren, nemen we voor beide integratoren  $dx = +1$ . De integratoren zijn volkomen symmetrisch geschakeld. Het hangt dan ook van de keuze der beginwaarden af in welke integrator welke hyperbolische functie komt.

sinh ut

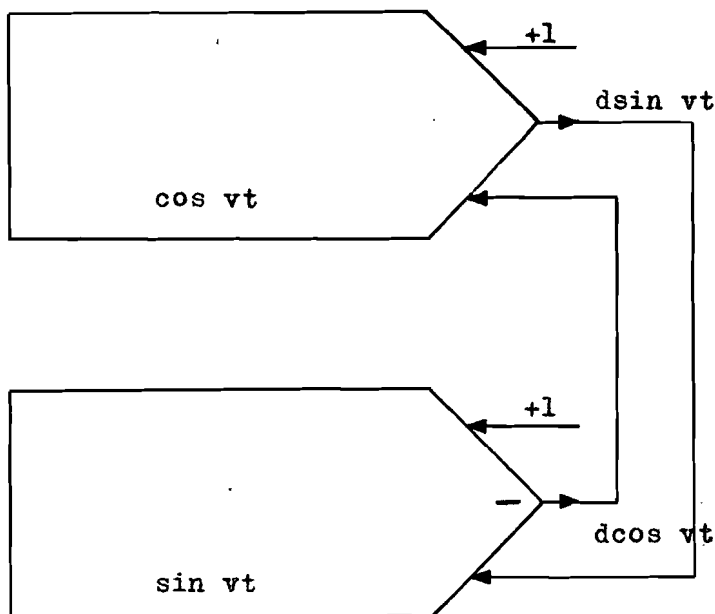


-65-

Figuur VI-4.  
Oplossing van de vergelijking  $y'' - y = 0$ .

C. Het oplossen van de vergelijking  $y'' + y = 0$ .

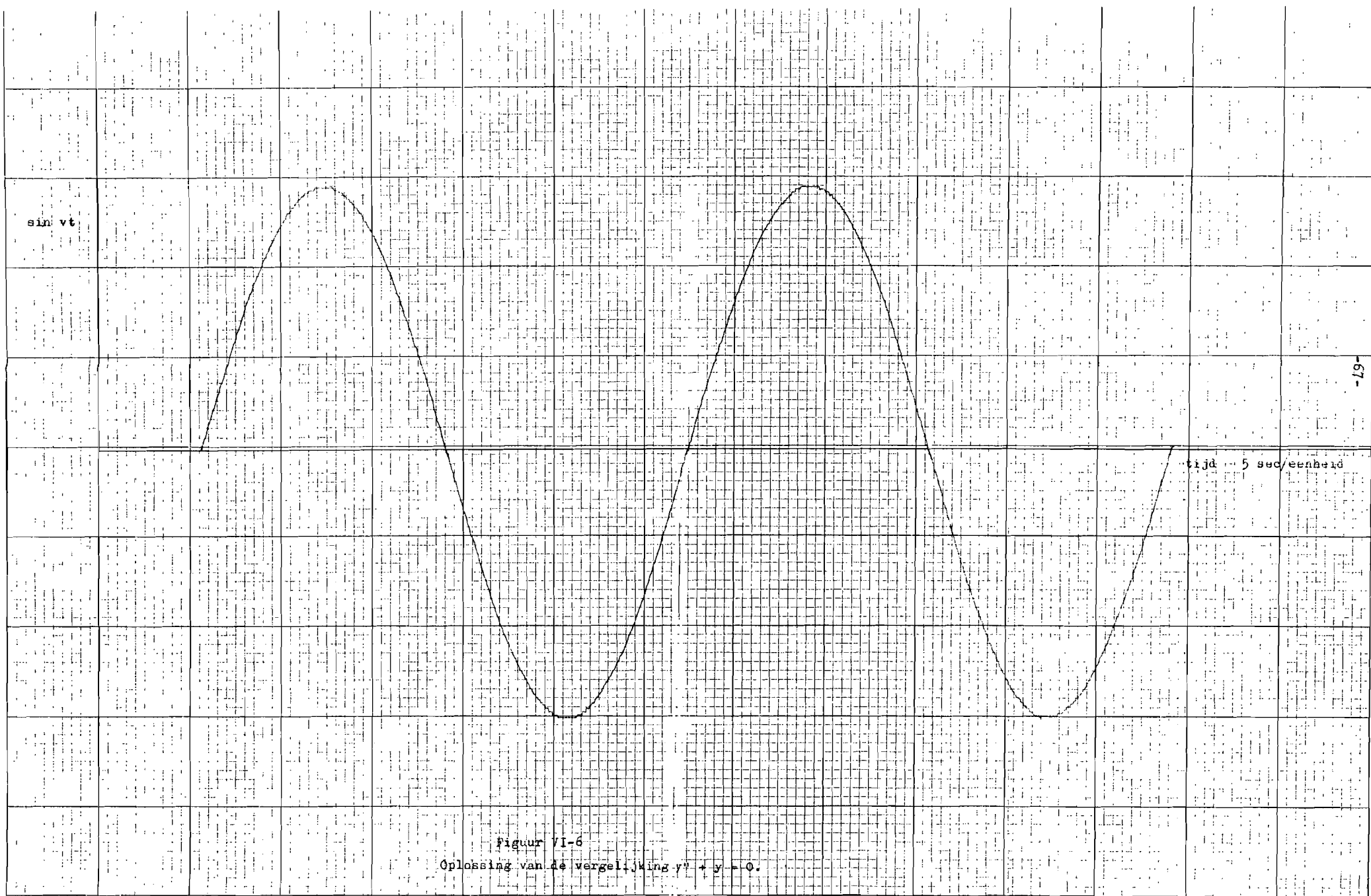
Deze vergelijking komt veel overeen met de voorgaande: we moeten eveneens twee maal integreren om de oplossing te vinden. Echter, nu is  $y'' = -y$ , dus we moeten eerst een min-teken voor de geïntegreerde functie plaatsen voordat we hem met  $y$  identificeren. Daarom schakelen we achter de dz-uitgang van de ene integrator de tekenomkeerschakeling van figuur III-8, en verbinden die met de dy-ingang van de andere. Dit is aangegeven in figuur VI-5.



Figuur VI-5.

Het genereren van de  
harmonische functies.

Daar we naar de tijd integreren, nemen we weer  $dx = +1$ . De algemene oplossing is  $A \sin vt + B \cos vt$ . Het verloop van de sinus is met de schrijver vastgelegd (zie figuur VI-6). Van de keuze der beginwaarden zal de amplitude van de sinus afhangen. Hoe groter deze zijn, des te nauwkeuriger is de benadering van de werkelijke functie. Zoals altijd bij de D.D.A. wordt hier een grens gesteld door de capaciteit van het Y-register. De hoeksnelheid van de harmonische functie kan echter zonder meer bij de gebouwde machine niet veranderd worden door een kleinere beginwaarde te kiezen, daar de significantie van het getal dz ongewijzigd blijft.



Figuur VI-6  
Oplossing van de vergelijking  $y'' + y = 0$ .

Besluit.

Uit dit rapport moge blijken dat hier de opzet gegeven wordt van een Digital Differential Analyser die meer dan achtduizend integratiestappen per seconde kan uitvoeren bij een woordlengte van vijftien significante bits.

Het aantal integratoren kan willekeurig worden uitgebreid, zonder dat dit beperkingen oplegt aan de snelheid van de machine.

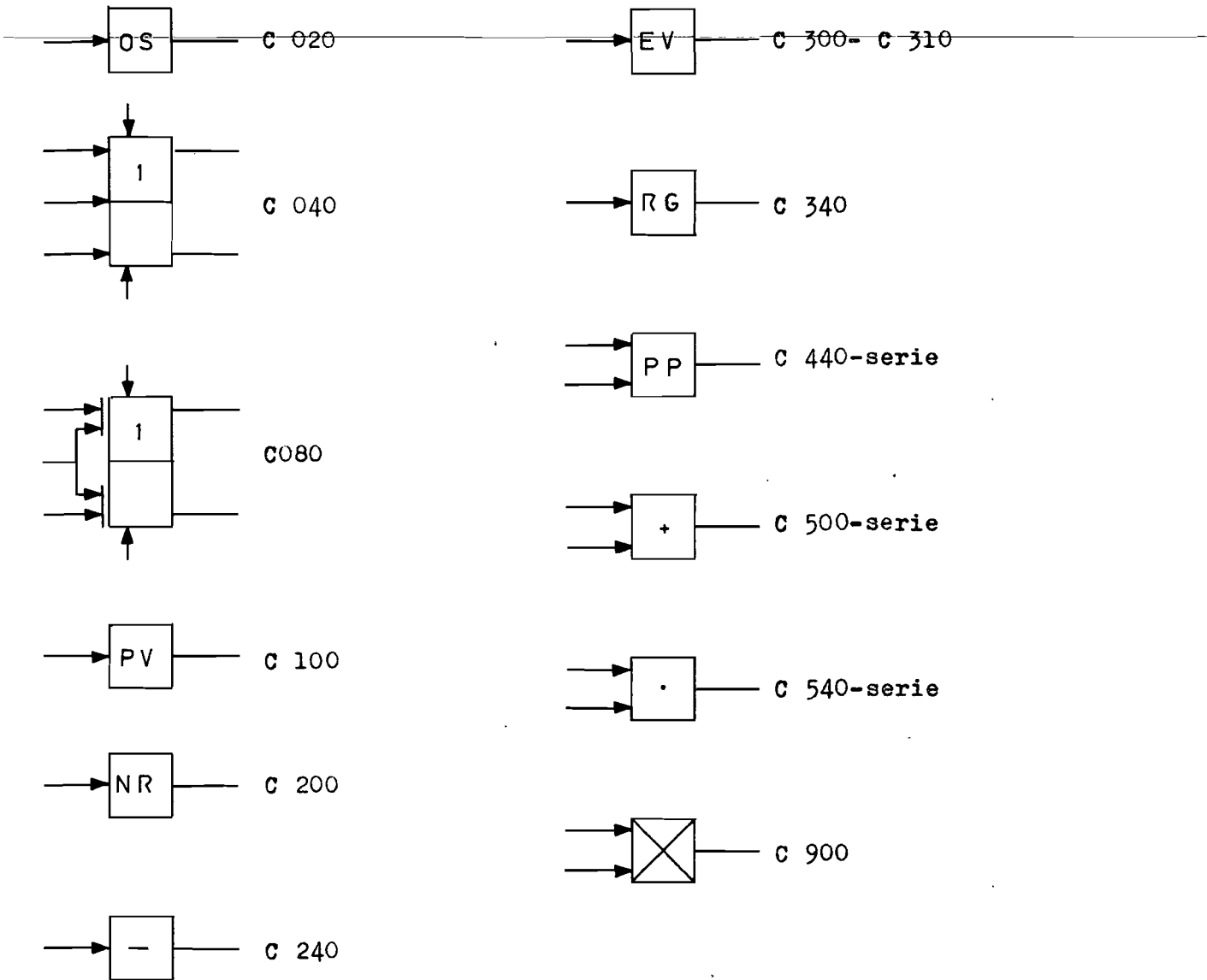
Eveneens is ruimte opengelaten voor een uitbreiding van de woordlengte die wel ten koste van de machinesnelheid zal gaan. Gebouwd zijn twee integratoren met bijbehorende apparatuur, waarbij hoofdzakelijk gebruik is gemaakt van logische schakelingen, die door de Technische Hogeschool gebouwd en beschikbaar gesteld zijn.

Het systeem is opgezet voor zestien integratoren.

Professor Heetman dank ik van harte voor de bijzonder interessante opdracht die ik van hem ter voltooiing van mijn studie aan de Technische Hogeschool te Eindhoven mocht ontvangen. Ook de heer Ir.Schnabel ben ik veel dank verschuldigd voor de wijze waarop hij mij bij het uitvoeren van de opdracht heeft geleid en de vruchtbare discussies die ik met hem mocht hebben. Verder dank ik allen die, hetzij direct, hetzij indirect tot de realisering van het te bouwen systeem hebben bijgedragen, in het bijzonder mijn collega de heer F.L.Engel waarmee ik zeer waardevolle discussies heb gehad.

N.J.Bos.

### Symbolen



Voor een verklaring der codenummers wordt verwezen naar de documentatie van logische schakelingen die uitgegeven is door de Technische Hogeschool Eindhoven afdeling Elektrotechniek Sectie Telecommunicatie B.



Literatur.

1. Huskey and Korn Computer Handbook. Section 19

---

  2. Hoffmann Digitale Informationswandler. p.160 o.v.
  3. Palevsky The Design of the Bendix Digital Differential Analyser. Proc. of the I.R.E., oct.
  4. Tootill The Incremental Digital Computer. Process Control and Automation, sept.1958
  5. Bradley and Genna Design of a one MC Iteration Rate D.D.A. Hazeltine Development Center, Inc. Indianapolis, India.
  6. Mitchell and Ruhman The Trice - a high speed incremental computer 1958. Packard Bell Computer Corporation. Los Angeles California.
  7. Owen, Partridge and Sizer The Differential Analyser and its realization in Digital Form. Electronic Engineering, oct.1960 - nov.1960.
  8. Owen, Partridge and Sizer CORSAIR, a Digital Differential Analyser. Electronic Engineering, dec.1960
  9. Owen, Partridge and Sizer The Address System and Increment Store for a Digital Differential Analyser. Electronic Engineering, oct.1960.
  10. Herring and Lamb The Digital Differential Analyser as a General Analogue Computer. Second International Analogue Computation Meetings, 1958.
  11. Mendelson The Decimal Digital Differential Analyser. Aeronautical Engineering Review, febr.1954.
-