

MASTER

Special purpose hybride systeem voor simulaties van wereldmodellen

Monen, M.J.B.M.

Award date:
1973

[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.

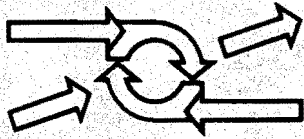
General rights

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.



GROEP METEN EN REGELEN
AFDELING DER TECHNISCHE NATUURKUNDE
T.H. Eindhoven, Postbus 513
Telefoon 040 - 472885

Rapport van het afstudeerwerk, uitgevoerd van mei 1972 tot april 1973.

Datum rapport: 18 april 1973. Begeleider: Ir. J.G.M. Cuypers.

Special purpose hybride systeem voor
simulaties van wereldmodellen.

M.J.B.M. Monen

Samenvatting:

Teneinde analoge simulatiestudies te kunnen verrichten aan systemen (in het bijzonder wereldmodellen) die qua grootte de capaciteit van de analoge rekenmachine overtreffen is een special purpose hybride systeem ontworpen bestaande uit een analoge rekenmachine (bijvoorbeeld de EAI 680) en een kleine procescomputer (de PDP8).

Aan de hand van een aantal eisen welke deels op grond van de uit te voeren simulaties deels op grond van de algemene bruikbaarheid van het systeem geformuleerd kunnen worden, is een keuze gemaakt voor een systeemopzet. De realisatie van deze systeemopzet is geresulteerd in een systeem dat getest op zijn bruikbaarheid aan de gestelde eisen beantwoordt.

I N H O U D S O P G A V E .

		blz.
I	Inleiding	1
I-1	Algemene Problematiek	1
I-2	Het Wereldmodel 3	1
I-3	Rekenfaciliteiten	2
I-4	Probleemstelling	3
II	De probleemstelling nader bezien	4
II-1	Eisen en wensen	4
II-2	Problemen bij de koppeling van analoge en digitale machine	6
III	Motivatie van de systeemopzet	8
III-1	De verdeling van taken	8
III-2	Het digitale hulpprogramma	8
III-3	De keuze van machineeenheid	9
III-4	Keuze van Schaling	10
III-5	Keuze van signaalverwerking in de PDP8	11
IV	Hybride simulatie van Wereldmodel 3	12
IV-1	De verdeling van rekentaken	12
IV-2	Het digitale rekengedeelte	13
IV-3	Subroutines voor het uitvoeren van hybride bewerkingen	17
IV-3-1	De digitale funktiegenerator	17
IV-3-2	De digitale Multiplier	21
IV-3-3	De digitale Opteller	22
IV-3-4	De digitale Potentiometer	22

		blz.
V	Realisatie van de systeemopzet	23
V-1	Globaal overzicht van het systeem	23
V-2	De programmamonitor	24
V-2-1	Uitvoering van de programmamonitor	25
V-3	Het inputgedeelte	27
V-3-1	De verwerking van de input	27
V-3-2	Inlezen van tabellen	30
V-3-3	De verwerking van tabelgegevens	32
V-3-4	Het inlezen en verwerken van de extreme waarden behorende bij de funktietabellen	34
V-3-5	Het inlezen en verwerken van potentiometer- standen	35
V-3-6	Het inlezen van een programma	37
V-4	Het rekengedeelte	37
V-4-1	Organisatie initialisatie van het hybride rekenprogramma	37
V-4-2	Het hybride rekenprogramma	40
VI	Testsimulatie met het gekoppelde systeem	45
VI-1	Simulatieresultaten	47
Bijlagen:	a. Lijst van subroutines	
	b. Lijst van adressen	
	c. Literatuurlijst.	

I INLEIDING.

I-1 Algemene Problematiek.

Een der werkeenheden van de vakgroep Meten en Regelen i.o. bestudeert de dynamica en regeling van niet-technische systemen. De projekten binnen deze werkeenheden maken deel uit van de landelijke projektgroep GLOBALE DYNAMICA, welke zich bezig houdt met de problematiek die de "Club van Rome" ¹⁾ indertijd op tafel heeft gelegd. De bestudering van deze problematiek in de vakgroep Meten en Regelen geschiedt door in de systeemkunde bekende technieken toe te passen op modellen van de wereld ²⁾. Het zogenaamde Wereldmodel 3 is tot op heden het meest volmaakte wereldmodel en is samengesteld door een team van het M.I.T. ^{*}) onder leiding van D.L. Meadows ³⁾. De bedoeling van deze modelstudies is om naast het verkrijgen van meer inzicht in de onderlinge relaties tussen de belangrijke grootheden in de wereld (zoals bijvoorbeeld bevolking en natuurlijke hulpbronnen) tevens het verloop van deze grootheden voor de komende decennia te leren kennen om ze zo nodig bij te kunnen sturen.

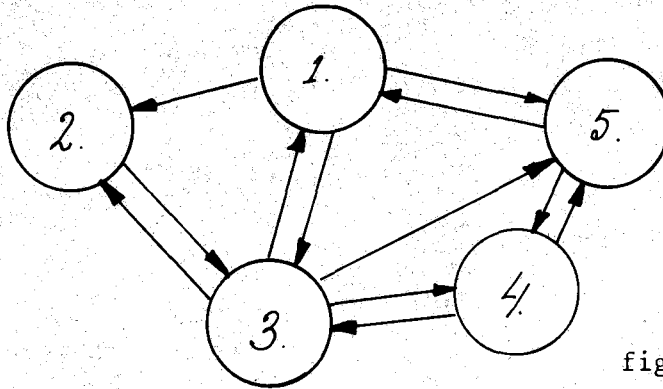
I-2 Het Wereldmodel 3.

Het Wereldmodel 3 is een model van de wereld dat een zo goed mogelijke mathematische beschrijving beoogt te geven van de bestaande wereld, voor zover dit een aantal belangrijke grootheden betreft. Het model is opgebouwd uit sectoren, die elkaar in meerdere of mindere mate beïnvloeden. Deze sectoren zijn:

1. De bevolkingssector
2. De grondstoffensektor
3. De kapitaal- en dienstensektor
4. De landbouwsector
5. De vervuilingsector

^{*}) Massachusetts Institute of Technology.

Een globaal beïnvloedingsschema van deze sectoren op elkaar wordt weergegeven in onderstaande figuur. Hierbij is vorenstaande nummering van toepassing.



figuur I-2-1

In het tot stand komen van het model van een deelsektor zijn twee stadia te onderscheiden. Het eerste stadium heeft een kwalitatief karakter waarin de beïnvloedingen van de diverse grootheden binnen een sektor worden vastgelegd. In het tweede stadium worden deze invloeden gekwantificeerd met behulp van statistische gegevens, behoudswetten en de nodige dosis (on)gezond verstand. Indien alle deelsektoren die op deze wijze verkregen zijn worden samengevoegd ontstaat een op zich autonoom systeem, Wereldmodel 3. Met dit model zijn door het M.I.T. een aantal berekeningen uitgevoerd met behulp van een digitaal programma geschreven in de simulatietaal DYNAMO⁴⁾. De dimensionering van het model is zodanig door het M.I.T. aangebracht dat het verloop van de modelgrootheden van de periode 1900 - 1970 zo goed mogelijk overeenstemt met het verloop van de werkelijke wereldgrootheden in deze periode. Bovendien is de berekening voortgezet tot het jaar 2100. Deze berekening wordt de "global standard run" genoemd.

I-3 Rekenfaciliteiten.

Het oorspronkelijk in DYNAMO geschreven model is omgezet in ALGOL voor gebruik op digitale rekenmachines zonder DYNAMO compiler.

Dit programma geeft exakt dezelfde resultaten als het oorspronkelijke DYNAMO programma (standard run) en is daarom in principe geschikt voor het verkrijgen van de resultaten van een aantal systeem- of regeltechnische analyses. Om echter snel inzicht in het systeem als functie van een aantal belangrijke parameters of om snel de mate waarin belangrijke grootheden elkaar beïnvloeden te achterhalen, is deze simulatie minder geschikt, aangezien de omlooptijd voor het verwerken van een programma bijzonder groot is. Dit laatste is onvermijdelijk aangezien een groot aantal lieden van dezelfde rekenfaciliteiten gebruik moet maken.

Om deze moeilijkheden te omzeilen is het wenselijk om naast de digitale simulatie te kunnen beschikken over analoge simulaties. Het grote voordeel van het werken met analoge rekenmachines is dat de tijd die nodig is om het model voor een honderdtal jaren door te rekenen tot enkele seconden beperkt kan worden. Binnen die rekestijd is ook de output van de analoge rekenmachine beschikbaar. Daardoor en omdat bij een analoge simulatie op eenvoudige wijze veranderingen zijn aan te brengen is analoog rekenen uitermate geschikt om op snelle en doeltreffende wijze inzicht te krijgen in het gedrag van de belangrijke grootheden uit het model als functie van de aangebrachte veranderingen.

De capaciteit van de beschikbare analoge rekenmachines is echter niet voldoende om het volledige Wereldmodel 3 te simuleren. Wel is er een digitale rekenmachine (PDP8) beschikbaar die met behulp van de nodige interface aan een analoge machine gekoppeld kan worden voor het overnemen van bepaalde - op de analoge machine ruimterovende - taken.

I-4

Probleemstelling.

De analoge en digitale machine moeten op een zodanige wijze met elkaar samenwerken dat de voordelen van analoog rekenen volledig behouden blijven. Dit betekent dat er met het gekoppelde systeem op dezelfde wijze gemanipuleerd moet kunnen worden als bij een volledig analoge simulatie.

II DE PROBLEEMSTELLING NADER BEZIEN.

Doelstelling:

Het gekoppeld systeem van analoge rekenmachine en digitale rekenmachine (PDP8) dient in eerste instantie voor het realiseren van de mogelijkheid tot het verrichten van analoge simulatiestudies aan Wereldmodel 3. Daarnaast is het wenselijk dat het systeem ook gebruikt kan worden voor andere soortgelijke problemen zonder dat de gehele digitale programmering veranderd moet worden.

II-1 Eisen en wensen.

Vanuit een aantal gezichtspunten kunnen verschillende eisen worden geformuleerd waaraan het gekoppelde systeem zal moeten voldoen.

Eisen die het verrichten van *analoge studies* aan het systeem stelt betreffen:

- rekentijd:* De rekentijd moet, binnen de rekennauwkeurigheid die voor een oriënterend onderzoek vereist is, kort gemaakt kunnen worden (dat wil zeggen orde van grootte: sekonde);
- Output:* Binnen deze rekentijd moet het gedrag als elektrische spanningen beschikbaar zijn van die grootheden waarvan het verloop bestudeerd moet worden;
- Wijzigingen:* Invloeden van verschillende grootheden op elkaar moeten snel gewijzigd kunnen worden.

Eisen waaraan het systeem omwille van de specifieke eisen die *wereldmodel 3* stelt zal moeten voldoen betreffen:

- Funktiegenereren:* De mogelijkheid om een veertigtal funktietabellen^{*)} in te bouwen moet aanwezig zijn;

*) zie Hoofdstuk IV-1.

Funktiewij-

zigingen:

Een funktietabel moet op eenvoudige wijze ten behoeve van gevoeligheidsonderzoek veranderd kunnen worden;

Funktievol-

gorde:

Ten behoeve van deelonderzoeken aan Wereldmodel 3 moet van een beperkt aantal funktietabellen gebruik gemaakt kunnen worden. Dit betekent dat de funktietabellen door elkaar (onafhankelijk van de inleesvolgorde) gebruikt moeten kunnen worden.

De eisen waaraan het systeem omwille van de *algemene bruikbaarheid* zal moeten voldoen worden in het onderstaande aan de hand van een tweetal aspecten verduidelijkt:

Overzichtelijkheid : Willen derden gebruik maken van de mogelijkheden van het gekoppelde systeem dan zal het systeem overzichtelijk (bijvoorbeeld modulair) opgebouwd moeten zijn. Daarnaast moet de bediening van en communicatie met de machine eenvoudig en eenduidig zijn.

Flexibiliteit

: Om doeltreffend met het systeem te kunnen werken moeten parameters en programmaonderdelen op eenvoudige wijze veranderd kunnen worden. Analoog is dit het veranderen van een potentiometer en het verwisselen van een patchpanel. Digitaal moet dit worden het intypen van een boodschap of inlezen van een eenvoudig programma (ponsband).

Naast bovengenoemde eisen is het wenselijk om de mogelijkheden die de digitale machine biedt zoveel mogelijk te benutten. Deze mogelijkheden betreffen:

Reken nauwkeurigheid: In principe is het mogelijk om in een digitale machine berekeningen met iedere gewenste nauwkeurigheid uit te voeren door van een voldoende aantal woorden gebruik te maken.

Logika : Doordat de digitale machine operaties met bitrijen uitvoert is deze zeer geschikt voor het verwerken van logika.

II-2 Problemen bij de koppeling van analoge en digitale machine.

Rekeneenheid.

Voor het uitvoeren van rekenkundige bewerkingen is het noodzakelijk dat er met een en slechts een rekeneenheid gewerkt wordt. Deze rekeneenheid kan men als volgt definiëren:

De grootte e is een rekeneenheid als geldt : $e^2 = e$.

Voor de analoge machine is deze rekeneenheid 10 volt. De normale rekeneenheid van de digitale machine wordt weergegeven door het oktale getal 0001. In het gehele systeem van analoge en digitale machine zal met één eenheid gewerkt moeten worden dat wil zeggen dat de analoge eenheid in een digitaal signaal ter grootte van één digitale eenheid omgezet moet worden (evenzo van digitaal naar analoog). De A/D- en D/A-omzeters zetten een analoog signaal van 10 volt om in een oktaal getal 3777 ($\approx 4000_8$). Dit betekent dat de analoge rekeneenheid niet zonder meer in de digitale rekeneenheid wordt omgezet.

Schaling der rekenmachines.

Bij het oplossen van een probleem met een analoge rekenmachine worden de grootheden uit een fysisch systeem door analoge spanningen weergegeven.

Het bereik dat deze spanningen kunnen doorlopen, wordt bepaald door het bereik waarin de elementen van de analoge machine normaal werken. De grenzen van dit bereik worden uitgedrukt in machineeenheden (bijvoorbeeld plus en min 10 volt voor de EAI 680).

De vertaling (schaling) van fysische grootheden naar spanningen moet zodanig zijn dat de bereikgrenzen niet overschreden worden. Daarnaast is het wenselijk dat kleine spanningen met het oog op de (relatieve) rekennauwkeurigheid van de analoge rekenelementen zoveel mogelijk vermeden worden.

In tegenstelling tot de analoge rekenmachine is het in principe niet nodig om een digitale machine met geschaalde signalen te laten werken. In de PDP8 bestaat een woord weliswaar uit 12 bits, maar men kan om een grootheid weer te geven gebruik maken van meerdere woorden.

Tijdsynchronisatie.

Indien in beide machines bewerkingen uitgevoerd worden welke afhangen van de procestijd dan moet de verhouding rekentijd/procestijd van beide machines op elkaar zijn afgestemd.

Kapaciteit.

Niet alleen met de capaciteit van de analoge rekenmachine maar ook in zekere mate met de capaciteit van de digitale machine zal rekening gehouden moeten worden. Het is zaak om een efficiënt gebruik van het geheugen te maken.

Het belangrijkste capaciteitsprobleem betreft de A/D- en D/A-omzetter. Bij het opzetten van een simulatie zal hiermede terdege rekening gehouden moeten worden.

III MOTIVATIE VAN DE SYSTEEMOPZET.

III-1 De verdeling van taken.

Om aan de eisen, die het verrichten van analoge studies aan het systeem stelt (zie paragraaf II-1), te kunnen voldoen wordt een taakverdeling tussen analoge en digitale machine aangebracht. De digitale machine is daarbij te beschouwen als een uitbreiding van de analoge machine. De taken die de analoge machine om capaciteitsredenen niet aan kan en welke de digitale machine evengoed zo niet beter (nauwkeuriger) kan verrichten worden door de digitale machine verricht.

In concreto is voor de volgende taakverdeling gekozen:

- analoog* : 1. Alle tijdafhankelijke bewerkingen worden analoog gerealiseerd;
2. Alle grootheden, waarvan het verloop gedurende een run vastgelegd moet worden worden analoog uitgevoerd;
3. Die programmagedeelten waarvan men de invloed op de rest van het systeem door het aanbrengen van wijzigingen wil nagaan, worden analoog uitgevoerd;
- digitaal* : 4. Het overgrote deel van de algebraïsche bewerkingen, welke door beperkte capaciteit niet analoog te realiseren zijn, wordt digitaal uitgevoerd.

Door alle tijdafhankelijke bewerkingen analoog uit te voeren, wordt het probleem van (proces)tijd synchronisatie omzeild. Dit houdt onder meer in dat met de rekentijd van de digitale machine geen rekening gehouden hoeft te worden mits deze verwaarloosbaar klein is ten opzichte van de rekentijd van de analoge machine. De rekentijd voor de simulatie wordt naar onderen begrensd door de rekentijd van de digitale machine.

III-2 Het digitale Hulpprogramma.

Om aan de eisen die omwille van de algemene bruikbaarheid aan het systeem gesteld worden te kunnen voldoen wordt een hulpprogramma gebruikt.

Door dit hulpprogramma is het mogelijk om de programmering van de PDP8 voor hybride simulaties eenvoudig te houden. Dit hybride rekenprogramma zelf hoeft dan in principe slechts te bestaan uit een opsomming van de digitaal uit te voeren bewerkingen.

Ievens maakt het hulpprogramma de bediening van de PDP8 eenvoudig. Het inlezen of veranderen van gegevens kan door intypen van een eenvoudige kode op de teletype uitgevoerd worden.

Met behulp van het hulpprogramma is het bovendien mogelijk om de rekkentijd van een simulatie kort te houden. Dit gebeurt door een gedeelte van de rekenorganisatie, die verricht moet worden, reeds voor de eigenlijke simulatie uit te voeren.

III-3 De keuze van machineeenheid.

Mogelijkheden om het probleem dat de analoge machineeenheid niet in de digitale machineeenheid wordt omgezet aan te pakken zijn:

1. Bij uitwisseling van signalen tussen beide rekenmachines omrekenen van de eenheden door een signaal dat de digitale machine binnenkomt te vermenigvuldigen met 2^{-11} en door een signaal dat de digitale machine uitgaat te vermenigvuldigen met 2^{+11} .
2. Het invoeren van een hybride eenheid e_h .
Analoog is deze hybride eenheid tien volt; digitaal is deze hybride eenheid 4000_g .

Gekozen is voor de laatste aanpak omdat:

1. Het werken met één eenheid de bruikbaarheid van het systeem ten goede komt;
2. Het werken met een hybride eenheid bij het uitvoeren van de bewerkingen tijdsbesparend is.

Door met een hybride eenheid te werken moet er op gelet worden dat de resultaten van de digitale bewerkingen eventueel omgerekend moeten worden om het juiste aantal eenheden te verkrijgen. Dit probleem is bij iedere bewerking slechts eenmalig en komt ter sprake bij het uitwerken van de subroutines voor deze bewerkingen.

III-4 Keuze van Schaling.

Om de schaling van het gekoppelde systeem aan te pakken zijn er in principe de volgende mogelijkheden:

1. Alle signalen worden zodanig geschaald dat deze binnen het machinebereik blijven (± 1). Dit geldt zowel voor het analoge als het digitale gedeelte.
2. Het analoge gerealiseerde deel wordt op de normale wijze geschaald. De signalen die de PDP8 ingaan worden telkens naar hun oorspronkelijke waarden teruggerekend en verder als zodanig verwerkt in floating point (meerdere woorden per signaal). De eindresultaten worden analoge geschaald en via een D/A-omzetter aan de analoge machine afgegeven.
3. Het analoge gedeelte wordt weer op de normale wijze geschaald. Ieder signaal in de digitale machine is geschaald maar hoeft niet binnen het machinebereik (± 1) te blijven door met een semi-floating point notatie te werken. De signalen die de digitale machine in- en uitgaan worden herschaald.

De factoren die het al of niet of gedeeltelijk toepassen van schaling voor het digitale rekengedeelte bepalen zijn:

1. de snelheid waarmee het digitale gedeelte uitgerekend moet kunnen worden;
2. de nauwkeurigheid waarmee het digitale gedeelte uitgerekend moet kunnen worden.

Deze twee factoren zijn twee tegenstrijdige zaken. Het rekenen met grotere nauwkeurigheid houdt in werken met meerdere woorden per signaal. Het gebruik van meerdere woorden per signaal betekent een vermeerdering van de rekentijd welke in dezelfde orde van grootte ligt.

De grootte van de digitale rekentijd wordt voornamelijk bepaald door de tijdsduur die nodig is voor het uitvoeren van een subroutine, welke een analoge element vervangt.

De snelheid waarmee een van deze subroutines (de funktiegenerator) werkt hangt in sterke mate af van de schaling van hetingangssignaal. Voor de andere elementen is voor de rekensnelheid schaling niet van belang.

Gekozen is daarom voor een schaling van het digitale gedeelte welke afgezien van de signalen van de analoge rekenmachine uitsluitend afhangt van de funktiegeneratorelementen.

III-5 Keuze van signaalverwerking in de PDP8.

De relatief kleinste onnauwkeurigheid waarmee een analoog signaal in een digitaal signaal kan worden omgezet is $1/2023$. Indien de signaalgrootte in 11 bit wordt weergegeven betekent dit dat aan de juistheid van het laatste bit getwijfeld moet worden. Berekeningen met een nauwkeurigheid van meer dan 11 bit zijn daarom weinig zinvol. Berekeningen uitgevoerd met een relatieve nauwkeurigheid (bijvoorbeeld 11 bit) zijn echter wel gewenst om er voor te zorgen dat zo min mogelijke informatie, door het grote bereik dat de signalen doorlopen, verloren gaat.

Het werken met relatieve nauwkeurigheden in een digitale machine gebeurt door getallen weer te geven in de vorm van een mantisse en een exponent. De mantisse kan afhankelijk van de gewenste nauwkeurigheid meerdere adressen beslaan; ook de exponent kan afhankelijk van het gewenste bereik meerdere adressen omvatten. In ons geval waarin een nauwkeuriger berekening dan met 11 bit niet nodig is en waarin het bereik van de getallen door een semi-schaling beperkt wordt kan volstaan worden met twee adressen. Deze twee adressen geven het getal in *semi-floating point* notatie weer; het eerste adres bevat de mantisse (teken + 11 significante bits), het tweede adres bevat de exponent van de bij de mantisse behorende macht van twee.

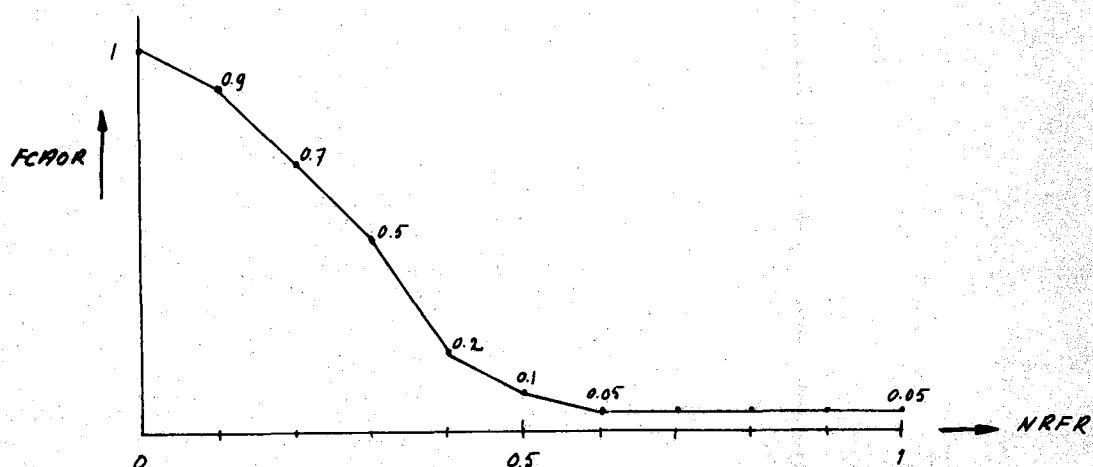
IV HYBRIDE SIMULATIE VAN WERELDMODEL 3.

IV-1 De verdeling van rekentaken.

Bij het maken van de taakverdeling tussen analoge en digitale rekenmachine (zie paragraaf III-1) voor hybride simulatie van Wereldmodel 3 moet met het onderstaande rekening gehouden worden.

1. De grootheden waarvan men het verloop gedurende een run wil registreren zijn:
 - a. de toestandsgrootheden (bijvoorbeeld bevolking, natuurlijke hulpbronnen en industrieel kapitaal);
 - b. die grootheden die behalve invloed op hun eigen deelsektor ook nog invloed op andere deelsektoren uitoefenen.
2. De wijzigingen die men in Wereldmodel 3 in eerste instantie aan wil brengen zijn:
 - a. veranderen van coëfficiënten door middel van potentiometers (beginwaarden, parameters);
 - b. het aanbrengen of verbreken van terugkoppelingen;
 - c. wijzigen van funktietabellen.

De algebraïsche bewerkingen die om capaciteitsredenen in het digitale gedeelte uitgevoerd moeten worden bestaan voornamelijk uit het genereren van funkties. Voor het hele Wereldmodel 3 zijn dit er een veertigtal. De funkties die gegenereerd moeten worden bestaan binnen het opgegeven funktiebereik uit een aaneenschakeling van lijnstukken, welke lineair zijn over gelijke intervallen. Buiten het funktiebereik zijn de funktiewaarden gelijk aan de opgegeven grenswaarde. Een dergelijk funktieverband wordt weergegeven in figuur IV-1-1.



figuur IV-1-1.

Ter illustratie is in figuur IV-1-2 de taakverdeling tussen analoge en digitale rekenmachine voor de kapitaal- en grondstoffensektor door middel van een rekenschema weergegeven.

IV-2 Het digitale rekengedeelte.

De analoge elementen welke de digitale rekenmachine (PDP8) moet vervangen voor simulatie van Wereldmodel 3 zijn:

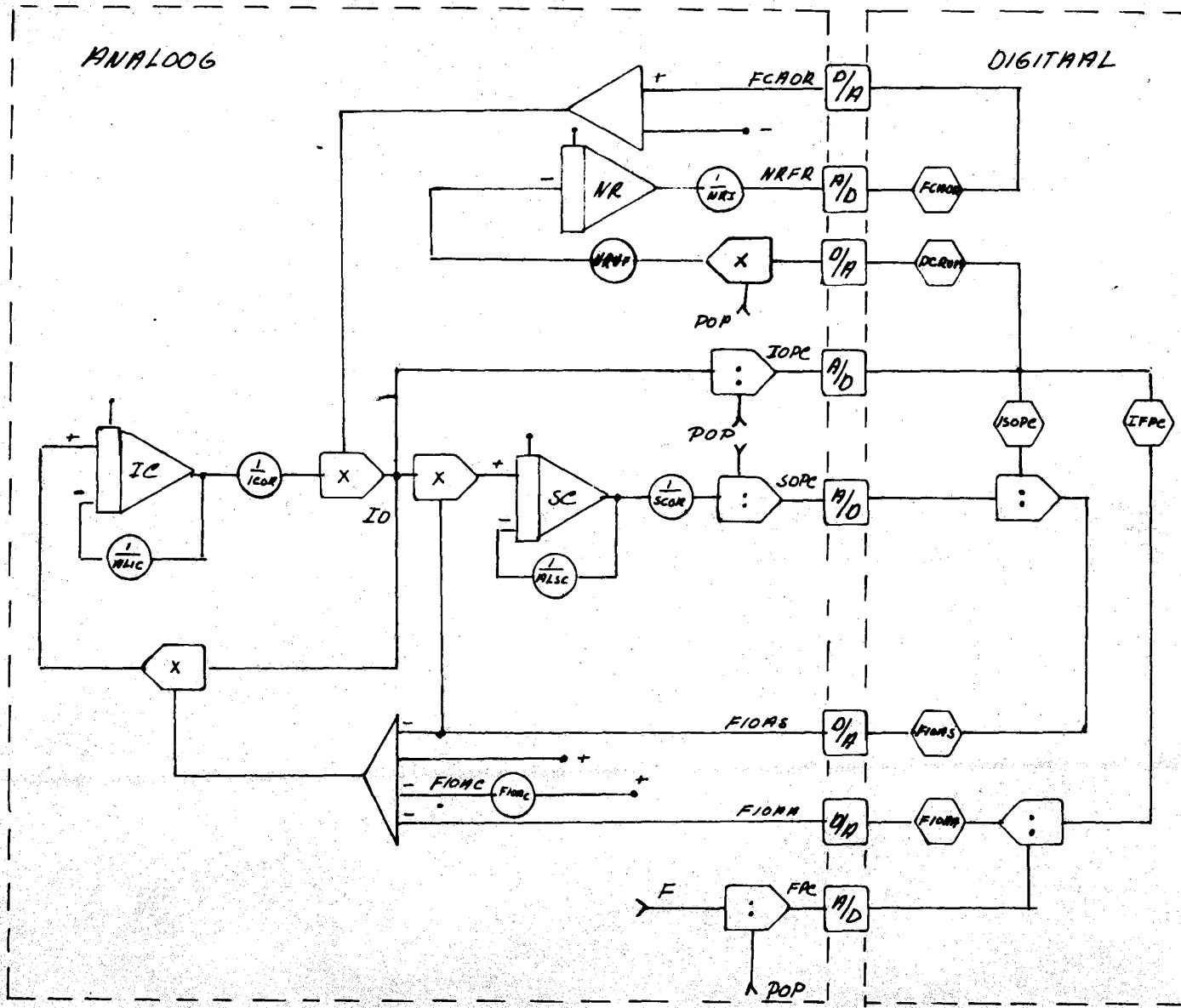
- funktiegeneratoren
- multipliers
- optellers
- potentiometers

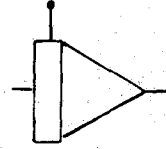
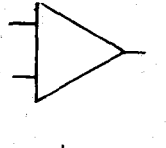
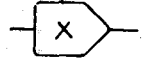


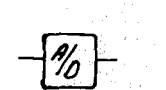
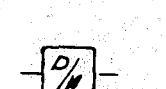

De elementen zijn in de digitale rekenmachine als subroutines uitgevoerd. Deze subroutines worden door het zogenaamde taakprogramma met de nodige parameters aangevraagd.

Het digitale taakprogramma moet zo snel uitgevoerd kunnen worden dat de hiervoor benodigde rekentijd ook bij hoge simulatiesnelheden geen invloed heeft op de resultaten.

Een bruikbare methode om deze rekentijd (monsterperiode) te verkorten bestaat uit het van te voren uitvoeren van een gedeelte van de rekenorganisatie (zie paragraaf III-2). Daarnaast moet er naar gestreefd worden om de subroutines, die de betreffende bewerkingen uit het digitale taakprogramma moeten uitvoeren zo efficiënt mogelijk te laten werken, dat wil zeggen dat ze in de korst mogelijke tijd het resultaat met de gewenste nauwkeurigheid berekenen.

De subroutines, die voor het uitvoeren van deze taken geschreven zijn, werken in genoemde semi-floating point (zie paragraaf III-5). Deze subroutines zijn geschreven voor zowel positieve als negatieve getallen. Dit is gedaan voor de algemene bruikbaarheid van het systeem; voor simulatie van Wereldmodel 3 kan in principe met uitsluitend positief werkende subroutines volstaan worden. De enkele uitzonderingen in Wereldmodel 3 waarin ook negatieve signalen voorkomen kunnen dan ad hoc behandeld worden.



-  integrator
-  opteller
-  vermenigvuldiger
-  deler
-  potentiometer
-  A/D-omzetter
-  D/A-omzetter
-  beïnvloeding uit andere sektor

Figuur IV-1-2. Rekenschema van Kapitaal- en Grondstoffensektor.

Tabel IV-1. Lijst met symbolen.

ALIC	:	Average Lifetime of Industrial Capital
ALSC	:	Average Lifetime of Service Capital
FCAOR	:	Fraction of Capital Allocated to Obtaining Resources
FPC	:	Food Per Capita
FIOAA	:	Fraction of Industrial Output Allocated to Agriculture
FIOAC	:	Fraction of Industrial Output Allocated to Consumption
FIOAS	:	Fraction of Industrial Output Allocated to Services
IC	:	Industrial Capital
ICOR	:	Industrial Capital Output Ratio
IFPC	:	Indicated Food Per Capita
IO	:	Industrial Output
IOPC	:	Industrial Output Per Capita
ISOPC	:	Indicated Service Output Per Capita
NR	:	Nonrenewable Resources
NRFR	:	Nonrenewable Resources Fraction Remaining
NRI	:	Nonrenewable Resources, Initial
NRUF	:	Nonrenewable Resources Usage Factor
PRCUM	:	Per Capita Resource Usage Multiplier
POP	:	Population
SC	:	Service Capital
SCOR	:	Service Capital Output Ratio
SO	:	Service Output
SOPC	:	Service Output Per Capita

Tabel IV-2. De modelvergelijkingen.

Toestandsvergelijkingen:

$$IC = (1 - FIOAC - FIOAA - FIOAS) \times IO - IC/ALIC$$

$$SC = FIOAS \times IO - SC/ALSC$$

$$NR = NRUF \times PCRUM \times POP$$

Beginwaarden:

$$IC = 2,1 \cdot 10^{11}$$

$$SC = 1,2 \cdot 10^{11}$$

$$NR = 1 \cdot 10^{12}$$

Koppelvergelijkingen:

$$IO = IC \times (1 - FCAOR)/ICOR$$

$$IOPC = IO/POP$$

$$SO = SC/SCOR$$

$$SOPC = SO/POP$$

$$NRFR = NR/NRI$$

$$FPC = F/POP$$

Tabeljunkties:

$$ISOPC = F(IOPC)$$

$$FIOAS = F(SOPC/ISOPC)$$

$$PCRUM = F(IOPC)$$

$$FCAOR = F(NRFR)$$

$$IFPC = F(IOPC)$$

$$FIOAA = F(FPC/IFPC)$$

Constanten:

$$ALIC = 14$$

$$ALSC = 18$$

$$FIOAC = 0,43$$

$$ICOR = 3$$

$$NRI = 1 \cdot 10^{12}$$

$$NRUF = 1$$

$$SCOR = 1$$

Beïnvloedingen van buiten:

POP (bevolkingssector)

F (landbouwsector)

IV-3 Subroutines voor het uitvoeren van hybride bewerkingen.

Om van de subroutines voor het uitvoeren van hybride bewerkingen gebruik te kunnen maken moeten deze in het taakprogramma aangevraagd worden waarbij de bij de aanroep behorende signaalparameters in floating accumulators te vinden zijn welke door het taakprogramma zelf geïnitieerd worden. Daarnaast worden die parameters die bij een subroutine-aanroep nodig zijn en die niet van een signaalgrootte afhangen uit een van te voren door het organisatieprogramma geïnitieerde lijst gehaald (zie paragraaf III-2). De principes en de belangrijkste punten in de uitvoering van deze subroutines worden in de navolgende paragrafen beschreven. Een flowdiagram en een volledige listing van deze subroutines is te vinden in Appendix A. Voor gebruik van deze subroutines kan volstaan worden met de in de Handleiding gegeven beschrijving van deze subroutines.

IV-3-1 De digitale funktiegenerator.

Principe.

De uitgangswaarde y welke bij een ingangswaarde x door de digitale funktiegenerator gegenereerd moet worden^{*}) voldoet aan onderstaand rekenvoorschrift

$$y = y_n + \frac{y_{n+1} - y_n}{x_{n+1} - x_n} * (x - x_n) \quad (IV-3-1-a)$$

Aangezien $x_{n+1} - x_n = h$ is dit te herschrijven tot

$$y = y_n + (y_{n+1} - y_n) * \frac{x - x_n}{h} \quad (IV-3-1-b)$$

De funktiewaarden y_n worden in tabelvorm in het geheugen opgeslagen. Een funktiewaarde wordt daarbij in één woord weergegeven^{**}). Indien de grootte van een funktiewaarde niet met een woordlengte is weer te

*) Zie paragraaf IV-1-I en figuur IV-3-1.

* *) Het opbergen in een geheugenwoord heeft naast sneller rekenen het voordeel dat de geheugenbelasting van de PDP8 met een faktor 2 beperkt wordt.

geven dan wordt de gehele tabel met eenzelfde schaalfactor geschaald^{*)}. De volgorde waarin de funktiewaarden in deze funktietabellen staan is zodanig dat geldt:

$$\text{adres } (y_n) = \text{adres } (y_0) + n \quad \text{met } n \in \mathbb{Z} \quad (\text{IV-3-1-c})$$

Voor het ingangssignaal x van de funktiegenerator geldt:

$$x = x - x_n + x_n$$

Voor $x_0 = 0$ volgt hieruit met $x_{n+1} - x_n = h$ dat

$$x = (x - x_n) + n \cdot h$$

Delen van x door h levert:

$$x/h = (x - x_n)/h + n$$

De termen in het linkerlid van deze vergelijking zijn voor het berekenen van de uitgangswaarde y met formule (IV-3-1-b) van belang. De eerste term $(x - x_n)/h$ is in deze formule direkt nodig, terwijl met de tweede term n de funktiewaarden y_n en y_{n+1} gevonden kunnen worden. Dit opzoeken van y_n en y_{n+1} in de funktietabel gebeurt met behulp van het referentieadres $\text{adres } (y_0)$ en de formule (IV-3-1-c).

Door het ingangssignaal x van de funktiegenerator zodanig te schalen dat het getal voor h een zo groot mogelijke macht^{**)} van twee is, zijn de termen $(x - x_n)/h$ en n met de PDP8 zeer snel uit x te bepalen. De minst significante bits van x vormen door deze schaling het getal $x - x_n$, terwijl de meest significante bits n vormen. Deling van de op deze wijze geschaalde x door h met de PDP8 (schuiven van bits in het rekenregister) heeft tegelijkertijd de splitsing van x/h in de termen n en $(x - x_n)/h$ tot gevolg.

*) De funkties die voor Wereldmodel 3 nodig zijn kunnen op deze wijze zonder informatieverlies weergegeven worden. De schaalfactor is daarbij een macht van tien.

***) Het getal dat de intervalafstand h weergeeft moet zo groot zijn dat het maximale ingangssignaal x dat voorkomt in 11 bits kan worden weergegeven.

Uitvoering.

De ingangswaarde van de digitale funktiegenerator wordt ten behoeve van de rekennauwkeurigheid in semi-floating point notatie opgegeven. De schaling, waarop het funktiegeneratieprincipe berust, houdt in dat de waarde van de exponent van een funktiegeneratoringang, die binnen het bereik van de funktiegenerator valt, kleiner of gelijk aan nul is. Is deze exponent positief dan wordt ofwel een foutmelding gegeven ofwel de grenswaarde van de ingelezen funktietabel als uitgang genomen^{*}).

Voor het bepalen van de termen n en $(x_n - x)/h$ wordt de mantisse van de funktiegeneratoringang met een ASR instructie^{**}) in het rekenregister verschoven. Het aantal plaatsen waarover geschoven wordt is afhankelijk van de bij de mantisse behorende exponent en het aantal intervallen in de funktiegeneratortabel.

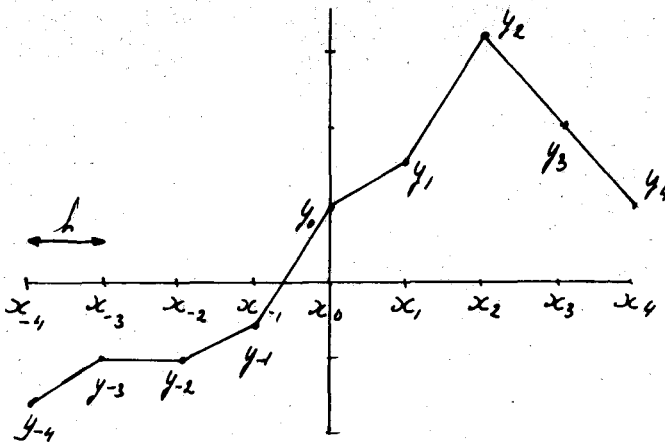
Het aantal plaatsen, waarover op grond van het aantal intervallen geschoven moet worden staat gevolgd door het referentieadres (adres (y_0)) op een geheugenplaats welke met een pointer wordt aangeduid. Omwille van de rekensnelheid staat deze pointer op een autoindexadres.

De berekende uitgangswaarde van de funktiegenerator wordt genormaliseerd en in semi-floating point notatie afgegeven.

In figuur IV-3-1 wordt de werking en het principe van de digitale funktiegenerator nader geïllustreerd.

-
- ^{*}) Een foutmelding wordt gegeven indien omwille van de rekennauwkeurigheid slechts een gedeelte van de ingelezen funktietabel gebruikt wordt; wordt de gehele ingelezen funktietabel gebruikt dan wordt de grenswaarde als uitgang genomen (zie paragraaf V-4-1).
- ^{**}) Bij gebruik van een ASR instructie is het mogelijk om ook met negatieve ingangswaarden te werken, aangezien de door het schuiven vrijgekomen plaatsen met het tekenbit worden gevuld⁵⁾. Dit betekent dat voor negatieve ingangen n negatief wordt.

FUNCTIEWEERGAVE IN GRAFIEK

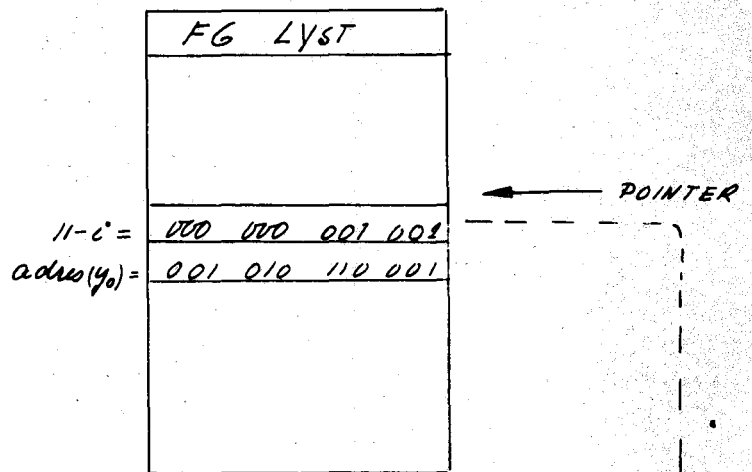


- y₁ y₋₄ = - 16
- y₂ y₋₃ = - 10
- y₃ y₋₂ = - 10
- y₄ y₋₁ = - 5
- y₀ = 10
- y₁ = 16
- y₂ = 32
- y₃ = 20
- y₄ = 10
- h = 10

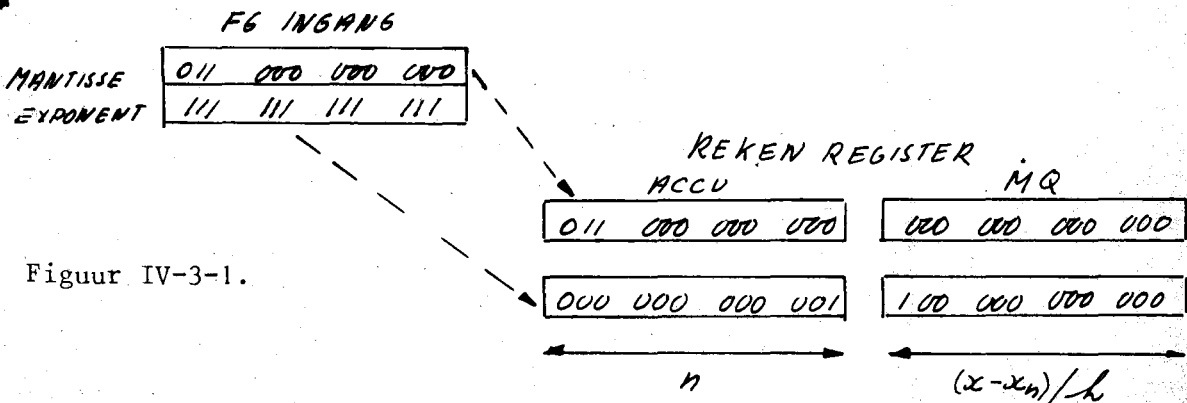
FUNCTIE WEERGAVE IN DE PDPD

	FUNCTIE TABEL
y ₋₄ =	111 111 100 000
y ₋₃ =	111 111 110 110
y ₋₂ =	111 111 110 110
y ₋₁ =	111 111 111 011
y ₀ =	000 000 001 010
y ₁ =	000 000 010 000
y ₂ =	000 000 100 000
y ₃ =	000 000 010 100
y ₄ =	000 000 001 010
h =	001 000 000 000

$\overleftrightarrow{i=2}$



PRINCIPE FUNCTIEGENERATOR



Illustratie van het Functiegeneratorprincipe.

IV-3-2 De digitale Multiplier.

Principe.

In de digitale rekenmachine worden produkten en quotiënten van signalen uitgerekend. De rekeneenheid waarin deze getallen uitgedrukt worden is de hybride rekeneenheid (e_h). Het berekenen van een quotiënt of produkt van twee getallen in de PDP8 gebeurt in principe op de normale digitale wijze, waarbij gebruik gemaakt wordt van een 24 bits rekenregister. De rekeneenheid is de digitale rekeneenheid (e_d). Het produkt of quotiënt uitgedrukt in digitale eenheden moet met een faktor gekorrigeerd worden om het produkt of quotiënt uitgedrukt in hybride eenheden te verkrijgen. Deze factoren zijn 2^{-11} voor het produkt en 2^{+11} voor het quotiënt. Dit is op eenvoudige wijze na te gaan met de formule $e_h = 10^{11} * e_d$ welke het verband tussen de beide rekeneenheden weergeeft.

Uitvoering.

De signalen worden in semi-floating point verwerkt, waardoor het bereik van de getallen aanzienlijk vergroot wordt ^{*}). Daarom en omdat de signalen ten behoeve van de funktiegeneratoren regelmatig op een hybride machineeenheid geschaald worden bestaat er geen gevaar voor overflow. Bij het schalen van het digitale gedeelte hoeft daarom met multipliers geen rekening gehouden te worden. De eigenlijke digitale vermenigvuldiging en deling kan slechts met positieve getallen uitgevoerd worden, waardoor extra maatregelen getroffen worden met betrekking tot het teken. Beide ingangssignalen worden op teken getest en zo nodig van teken omgekeerd. Het teken van het resultaat wordt onthouden en na de berekening van de grootte aan het uitgangssignaal toegevoegd.

In de exponent van het uitgangssignaal wordt de korrektiefaktor voor de overgang van digitale naar hybride eenheden op eenvoudige wijze verwerkt.

^{*}) grootste getal in e_h is 2^{2023}

kleinste getal in e_h is 2^{-2024}

IV-3-3 De digitale Opteller.

Het optellen en aftrekken van hybride signalen in de digitale machine gebeurt volgens een normale digitale optelling. Dit brengt geen extra problemen met betrekking tot de verschillende rekeneenheden met zich mee.

Aangezien de signalen in semi-floating point verwerkt worden moet er zorg voor gedragen worden dat bij de optelling de komma's van beide signalen onder elkaar staan. Hieraan is voldaan als de exponenten van beide getallen gelijk zijn. Deze exponenten worden gelijk gemaakt door bij de kleinste exponent een passend aantal op te tellen, terwijl de bij deze exponent behorende mantisse een gelijk aantal malen door twee gedeeld wordt.

Bij optellen van twee getallen met hetzelfde teken kan overflow optreden. Hier wordt op getest en eventueel wordt het resultaat gekorrigeerd.

IV-3-4 De digitale Potentiometer.

Principe.

De functie van een potentiometer is het vermenigvuldigen van signalen met een vaste faktor. Het vermenigvuldigen van een signaal in hybride eenheden met een konstante faktor kan in de PDP8 met digitale eenheden gebeuren. Het op deze wijze ontstane produkt hoeft niet naar hybride eenheden omgerekend te worden, wat op eenvoudige wijze met de formule $e_h = 10^{11} * e_d$, welke het verband tussen beide rekeneenheden weergeeft, is na te gaan.

Uitvoering.

De konstante faktor waarmee het ingangssignaal vermenigvuldigd moet worden staat op een geheugenplaats in een lijst met potentiometerwaarden welke met een pointer wordt aangewezen. Omwille van de rekensnelheid staat deze pointer op een autoindexadres.

Het produkt van hetingangssignaal en de konstante faktor wordt met behoud van teken op normale wijze in semi-floating point berekend.

V REALISATIE VAN DE SYSTEEMOPZET.

V-1 Globaal overzicht van het systeem.

Het systeem is opgebouwd uit een aantal min of meer op zich zelf staande taakprogramma's. Deze taakprogramma's kunnen in opdracht van de operator uitgevoerd worden. Voor het geven van deze opdrachten wordt gebruik gemaakt van een teletype. Het verwerken van deze opdrachten gebeurt in eerste instantie door de programmamonitor. De programmamonitor is dat gedeelte van de programmatuur dat de kommunikatie tussen operator en PDP8 verzorgt; in de programmamonitor beginnen en eindigen alle deeltaken (zenuwcentrum).

De taken die via de programmamonitor uitgevoerd kunnen worden zijn te verdelen in twee gedeelten:

1. een inputgedeelte;
2. een rekengedeelte;

Het inputgedeelte omvat:

- a. Het inlezen van tabellen.

Naast funktietabellen, die de informatie over de te genereren funkties bevatten, worden programmatabellen, met informatie over het uit te voeren programma, ingelezen;

- b. Het inlezen van extreme waarden.

Deze extreme waarden, die nodig zijn voor de schaling, behoren bij de funktietabellen;

- c. Het inlezen van potentiometerstanden.

De potentiometerstanden worden in formulevorm opgegeven, die aangeeft welke bewerkingen er met welke waarden uitgevoerd moeten worden;

- d. Het inlezen van het hybride rekenprogramma.

Het rekengedeelte omvat:

a. Een initialisatiedeel.

Dit initialisatiedeel verwerkt de ingelezen data tot bruikbare informatie voor het hybride programma;

b. Het hybride rekendeel.

Dit bestaat naast een vast deel uit een deel dat specifiek is voor het te bestuderen probleem.

Het hybride rekenprogramma is de kern waaromheen het gehele systeem is opgezet. De andere taken dienen enerzijds voor het sneller laten werken van dit programma anderzijds om de gebruiker een aantal taken, die voor het initialiseren van het programma noodzakelijk zijn, efficiënter te laten uitvoeren.

In dit Hoofdstuk worden de principes en de voornaamste onderdelen van de uitvoering der verschillende taken nader uiteengezet. De flowdiagrammen en de volledige listing van de verschillende onderdelen zijn te vinden in Appendix B.

V-2 De programmamonitor.

Het Hulpprogramma wordt gestart op adres 2000 voor initialisatie, waarna de PDP8 in de programmamonitor komt. Dit wordt op de teletype gemeld door het afdrukken van een PUNT (.).

Via de programmamonitor kunnen met de teletype taken aangevraagd worden door het indrukken van een van de *toetsen 1 tot en met 6*. De PDP8 reageert op deze aanvragen door op de teletype het herkenningwoord van de aangevraagde taak af te drukken. De herkenningwoorden van de taken zijn:

1. TABELLEN
2. MAXIMA
3. POTSTANDEN
4. PROGRAMMA
5. INITIALISATIE
6. STATTEST

De aangevraagde taak kan geëffektueerd worden door het indrukken van de *returntoets*, ieder ander kommando annuleert de aangevraagde taak. Indien de PDP8 een aangevraagde geëffektueerde taak niet kan verwerken (bijvoorbeeld door te weinig of foutieve gegevens) dan wordt dit door het afdrukken van een VRAAGTEKEN (?) op de teletype gemeld. In het geval dat de PDP8 een taak wel kan uitvoeren wordt deze uitgevoerd waarna een STER (★) op de teletype wordt afgedrukt.

Na annuleren en al of niet korrekt uitvoeren van een geëffektueerde taak komt de PDP8 terug in de programmamonitor door het indrukken van de *CTRL-C toets*.

In figuur V-1 is een overzichtsschema gegeven.

V-2-1 Uitvoering van de Programmamonitor.

Melding op teletype.

Door het uittypen van mededelingen op de teletype wordt van een eenvoudige typeroutine (*PRINT en PRINLR*) gebruik gemaakt, welke ook bij andere taken gebruikt worden.

Taakaanvraag.

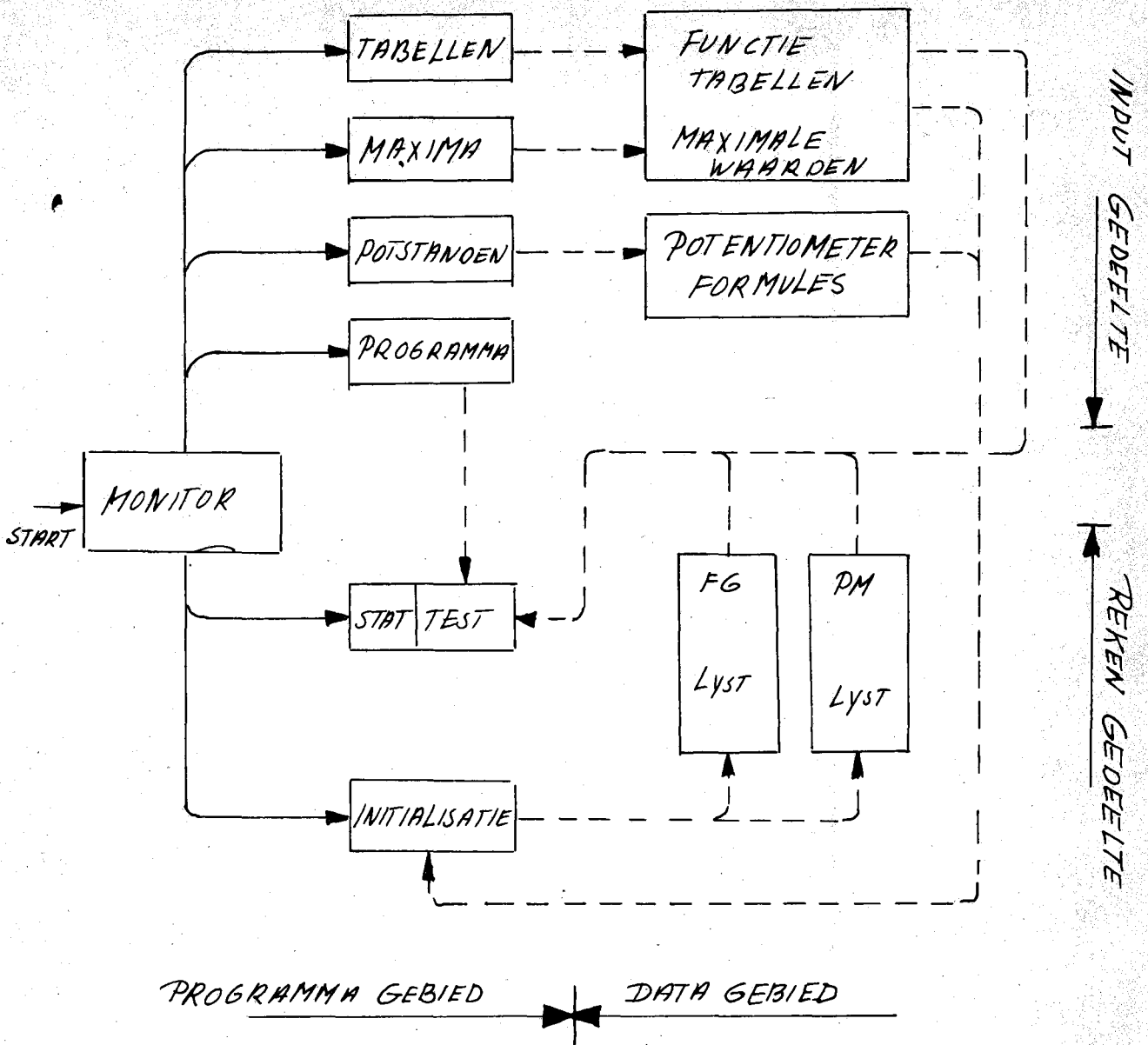
Een door de operator ingetypt karakter voor een taakaanvraag wordt door de PDP8 ingelezen en gestript op de laatste drie bits na. Vormen deze drie bits de getallen nul of zeven dan wordt om een nieuwe opdracht gevraagd; in de overige gevallen treedt navolgende procedure in werking. Het op bovenstaande wijze verkregen getal wordt opgeteld bij het begin adres (BAML) van de monitorlijst, waardoor het adres gevonden wordt waarin de nodige informatie staat over de uit te voeren taak. De inhoud van dit adres bestaat uit een getal waarvan de eerste bits het beginadres van een buffer ten opzichte van een vast adres (BAAML) en de laatste bits de grootte van dit buffer weergeven.

Herkenningswoord.

Dit buffer bevat het herkeningswoord en het beginadres van de uit te voeren taak. In dit buffer zijn twee ASCII karakters per woord opgeslagen.

Effektueren.

Na afdrukken van het herkeningswoord wacht de PDP8 op een teletypekarakter.



Figuur V-1.

Overzichtsschema van de systeemprogrammatuur.

- = stuurstroom
- - -→ = informatiestroom

Als dit karakter een return is wordt naar het beginadres van de betreffende taak, dat in vorenstaand buffer staat, gesprongen. Als dit karakter geen return is wordt om een nieuwe opdracht gevraagd.

Na een melding dat een taak al of niet uitgevoerd is wordt op het CTRL-C karakter van de teletype gewacht, waarna naar het monitorbegin gesprongen wordt. Ieder ander karakter wordt genegeerd. Dit is gedaan om foutmeldingen bij het werken met ponsband te voorkomen.

V-3 Het inputgedeelte.

De gegevens die nodig zijn voor het juist uitvoeren van het rekenprogramma, worden in ASCII-code met de teletype door de taken 1, 2 en 3 in de PDP8 ingelezen.

Het inlezen van deze gegevens geschiedt in de vorm van informatieblokken, die beginnen met een rangnummer en eindigen met een *puntkomma* (;). In deze blokken bestaan de getallen uit vier cijfers welke onmiddellijk worden voorafgegaan door een *punt* (.). Deze getallen mogen in grootte variëren van 0 tot en met 2023. In die gevallen waarin de getallen niet geheel (reals) zijn, komt deze punt overeen met de decimale komma. Een negatief getal wordt in deze blokken weergegeven door direkt voor deze punt een *minteken* (-) te plaatsen. Ieder getal mag eventueel door commentaar gevolgd worden, mits daarin *geen* der symbolen punt (.), puntkomma (;) of dollarteken (\$) voorkomt.

Wanneer alle gewenste informatieblokken door een taak zijn ingelezen moet dit aan de PDP8 medegedeeld worden door het indrukken van de *dollartoets* (\$), waardoor de machine weer naar de programmamonitor terugspringt.

V-3-1 De verwerking van de input.

De voornaamste deeltaken die bij het verwerken van de via de teletype in te lezen gegevens aan de orde komen worden uitgevoerd door subroutines. Deze deeltaken zijn:

1. Het opslaan van de ingelezen ASCII-karakters in een buffer.
2. Het omzetten der gegevens van ASCII naar machinecode.
3. Het opbergen van de in machinetaal gecodeerde gegevens in het datagebied.

Het bufferen van ASCII-karakters.

Om in een later stadium gegevens via de snelle lezer in te kunnen lezen is het gewenst dat ingelezen gegevens gebufferd worden. Dit bufferen van inleesgegevens wordt verzorgd door de *subroutine LBIN*. Deze subroutines drukt het ingelezen karakter ter controle af op de teletype met de *subroutine PRINT* en bergt daarna het karakter in een woord in een buffer op. Indien dit buffer vol is of indien het laatst gebufferde karakter een dollarteken (\$) is (einde input) dan wordt deze subroutine beëindigd.

Het decoderen van ASCII-karakters.

De werking van de subroutines die de ASCII-karakters uit bovengenoemd buffer omzetten in machinetaal wordt bij de betreffende taken nader behandeld. In deze subroutines wordt voor ophalen van de gebufferde ASCII-karakters gebruik gemaakt van de *subroutine LBUIT*. Deze subroutine haalt bij aanroep een karakter op en leest, indien dit het laatste karakter uit het buffer was, het buffer met de *subroutine LBIN* opnieuw in.

Voor het omzetten van een positief of negatief decimaal getal, waarvan de grootte door vier opeenvolgende adressen van het inleesbuffer in ASCII code wordt weergegeven, in een binair getal van een woordlengte, dient de *subroutine OKTA*. Het omzetten van een decimaal in een binair getal door deze subroutine gaat als volgt in zijn werk. In het eerste van de vier adressen uit het buffer staat in ASCII code het aantal duizendtallen, in het tweede het aantal honderdtallen, in het derde het aantal tientallen en in het vierde het aantal eenheden.

Door zoveel malen als in deze adressen aangegeven staat het getal duizend (= 1750₈), het getal honderd (= 144₈) het getal tien (= 12₈) en getal één (=1₈) in binaire code bij elkaar op te tellen is het decimale getal in een binair getal omgezet. Als het getal is omgezet wordt het van het gewenste teken voorzien. Indien in het buffer getallen niet door vier adressen worden weergegeven volgt een foutmelding.

Het opbergen van gegevens in het datagebied.

Het toewijzen van de ruimte in het datagebied en het opbergen van het in machinetaal gecodeerde informatieblok wordt door de *subroutine* *OBUS* verzorgd. De gedecodeerde informatieblokken worden op nummervolgorde geplaatst (NUMMER). Hierbij wordt gebruik gemaakt van een adressenlijst ter grootte van een halve pagina, waarin de beginadressen van de reeds geplaatste blokken in het datagebied staan. De plaats in de adressenlijst komt overeen met het nummer van het informatieblok. Indien een adres in deze adressenlijst nul is betekent dit dat het betreffende blok nog niet in het datagebied geplaatst is^{*}). Het beginadres van deze adressenlijst staat op adres LBA.

Naast bovengenoemde adressenlijst wordt het laatst gevulde adres in het datagebied bijgehouden op adres LGA^{**}).

De ruimte die voor herplaatsen of tussenvoegen van informatieblokken nodig is, wordt gekreeerd door het opschuiven van de hoger genummerde blokken. Ruimte die door herplaatsen van blokken ontstaat wordt opgevuld door terugschuiven van de hoger genummerde blokken.

Bij het op- en terugschuiven in het datagebied worden de beginadressen in de adressenlijst (LBA) aangepast. 1

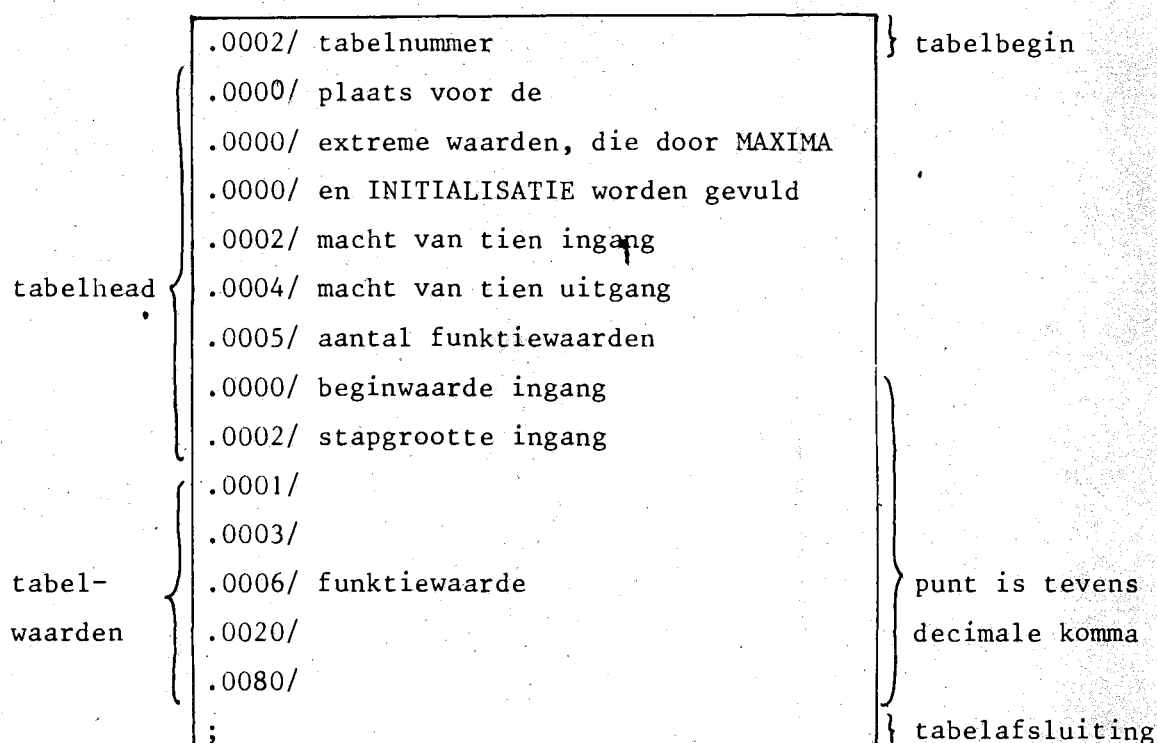
*) Bij het initialiseren van het systeem worden alle adressenlijsten schoongeveegd.

**) Bij het initialiseren van het systeem worden de adressen waarop de laatst gevulde adressen van de datagebieden worden onthouden gelijk gemaakt aan de beginadressen van deze datagebieden.

V-3-2 Inlezen van tabellen.

Funktietabellen.

De funktietabellen worden met de teletype door taak 1 (TABELLEN) in de PDP8 ingelezen. De tabelnummers van de funktietabellen mogen de waarden van 0001 tot 0059 doorlopen. Iedere funktietabel moet van een tabelhead, die in onderstaand voorbeeld wordt toegelicht, zijn voorzien.



Binnen de omlijsting staat de informatie zoals deze de PDP8 ingevoerd moet worden.

Indien slechts met positieve waarden van de funktiegeneratoringang gewerkt wordt, moet de beginwaarde van de funktiegeneratoringang nul zijn.

Indien ook met negatieve waarden van de funktiegeneratoringang gewerkt wordt is de beginwaarde negatief. Daarnaast dient het aantal funktiewaarden dat bij een negatieve ingang behoort gelijk te zijn aan het aantal funktiewaarden dat bij een positieve ingang behoort. Met de funktiewaarde van de nulingang erbij is het aantal funktiewaarden in deze funktietabellen dus oneven.

Programmatabelen.

Een aantal gegevens die van belang zijn voor het juist functioneren van het rekenprogramma moeten met de teletype door taak 1 (TABELLEN) in de PDP8 ingelezen worden.

In de tabellen genummerd 0060 en 0061 moet opgegeven worden in welke volgorde (nummer) respektievelijk de funktiegeneratoren en potentiometers uit het digitale rekenschema in het rekenprogramma gebruikt worden. Dit is nodig om er voor te kunnen zorgen dat de juist funktietabellen en de juiste potentiometerwaarden in het rekenprogramma gebruikt worden.

De extreme waarden van de grootheden die de analoge rekenmachine uitgaan en ingaan moeten respektievelijk in tabel nummer 0062 en tabel nummer 0063 op nummervolgorde opgeslagen worden. Deze gegevens zijn nodig voor het schalen van de twee rekenmachines op elkaar. Konstanten die in het digitale gedeelte verwerkt worden zijn in tabel 0064 opgeslagen. Deze tabel is op dezelfde wijze opgebouwd als de tabellen 0062 en 0063. Van beide soorten programmatabelen volgt hierna een voorbeeld.

head	}	.0060/ tabelnummer	}	tabelbegin
		.0004/ aantal te gebruiken FG's		
tabel- waarden	}	.0002/ tabelnummer voor 1e aanroep	}	tabelafsluiting
		.0003/ tabelnummer voor 2e aanroep		
		.0001/ tabelnummer voor 3e aanroep		
		.0004/ tabelnummer voor 4e aanroep		
		;		

tabelhead	}	.0062/ tabelnummer	}	tabelbegin
		.0003/ aantal A/D-omzetters		
tabel- waarden	}	.0010/ mantisse AD0001 (decimaal	}	tabelafsluiting
		.0002/ exponent AD0001 (decimaal)		
		.0020/ mantisse AD0002 (decimaal)		
		.0002/ exponent AD0002 (decimaal)		
		.0080/ mantisse AD0003 (decimaal)		
		.0006/ exponent AD0003 (decimaal)		
		;		

Binnen de omlijsting staat de informatie zoals deze de PDP8 ingevoerd kan worden.

V-3-3 De verwerking van tabelgegevens.

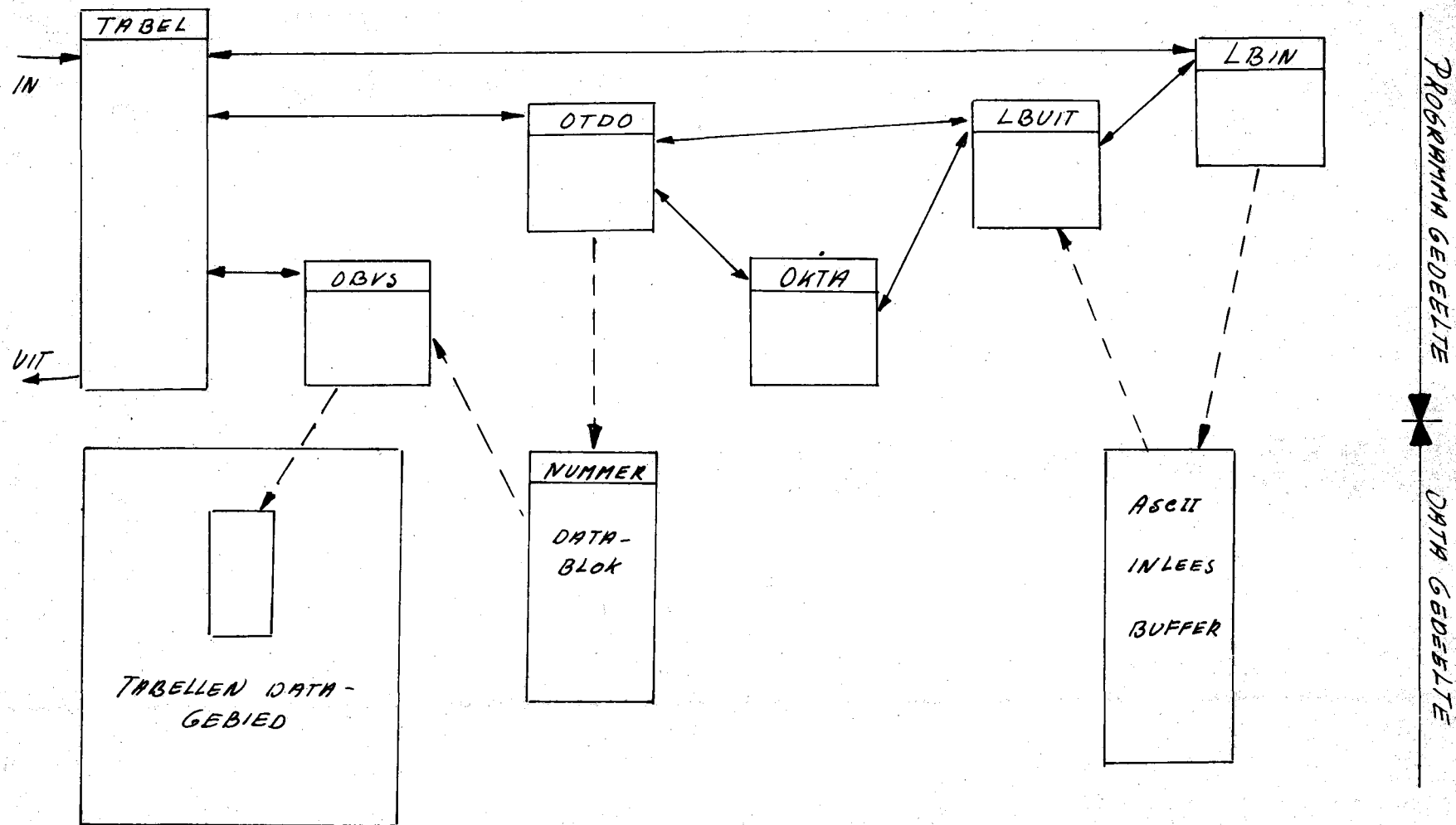
Het programmadeel dat de tabelinput verwerkt begint op het adres gelabeld TABEL.

De initialisatie van dit programmadeel betreft het aangeven van het datagebied (LBA en LGA) waarin de tabellen opgeborgen moeten worden. Deze adressen worden respectievelijk gelijk aan BAALFT en LGAFG, welke het beginadres van de funktietabellenadressenlijst en het laatstgevolde adres in het funktietabellendatagebied bevatten.

Het decoderen van het ingelezen buffer gebeurt met de *subroutine OTDO* door het testen van de uit het buffer opgehaalde karakters (*LBUTT*). Getest wordt daarbij of het karakter een dollarteken, puntkomma, punt of minteken is. Is het karakter geen van deze symbolen dan wordt het volgende karakter getest. Is het karakter een dollar dan is de taak uitgevoerd en wordt naar de programmamonitor teruggekeerd. Is het karakter een puntkomma dan is een volledig informatieblok gedecodeerd en moet dit opgeborgen worden in het datagebied (*OBUS*). Is het karakter een punt dan betekent dit dat de vier volgende karakters een getal bevatten wat gedecodeerd moet worden (*OKTA*). Dit getal wordt negatief indien het vorige geteste karakter een minteken was.

Na het opbergen van een informatieblok wordt het laatstgevolde adres van het funktietabellen datagebied (LGAFG) onthouden, dat wil zeggen LGAFG wordt gelijk aan LGA. Vervolgens wordt getest of het datagebied niet overschreden wordt. Indien dit het geval is wordt een foutmelding (monitor) gegeven, anders wordt tot het decoderen van de volgende tabel overgegaan.

In figuur V-2 is een overzichtsschema van het inleesprogramma voor tabellen gegeven.



Figuur V-2.

Overzichtsschema van het inleesprogramma van tabellen.

V-3-4 Het inlezen en verwerken van de extreme waarden behorende bij de funktie-
tabellen.

Inlezen.

Om de schaling en eventueel herschaling van het digitale rekenschema in de PDP8 door te kunnen voeren moeten (geschatte) extreme waarden van de funktiegeneratoringen aan de PDP8 opgegeven worden. Op grond van deze opgegeven waarde worden dan de extreme waarde van de digitale funktiegeneratoringang en de extreme waarde van de digitale funktiegeneratoruitgang bepaald.

Het opgeven van de absoluut genomen extreme waarde aan de PDP8 gaat door taak 2 (MAXIMA) met de teletype volgens onderstaand voorbeeld.

```
.0008/ tabelnummer  
.0006/ extreme waarde FG ingang  
;
```

Er moet op gelet worden dat de plaats van de punt (decimale komma) overeenstemt met de opgegeven macht van tien van de funktiegeneratoringang die in de betreffende tabel is opgegeven.

Verwerken.

Het programmadeel dat het inlezen van extreme waarden verzorgt begint op adres MAX.

Het bufferen (*LBIN*) en decoderen (*OTDO*) is in paragraaf V-3-1 en paragraaf V-3-3 reeds uiteengezet. De extreme waarden worden op het eerste adres van de tabelhead waarop de extreme waarde betrekking heeft opgeborgen. Dit adres wordt uit de funktietabellenadressenlijst (*BAALFT*) gehaald. Is dit adres nul dat wil zeggen dat de tabel nog niet geplaatst is dan volgt een foutmelding.

Na opbergen van de extreme waarde wordt met decoderen van de overige informatie uit het inleesbuffer verder gegaan.

V-3-5 Het inlezen en verwerken van de potentiometerstanden.

Inlezen.

Met de teletype kunnen met taak 3 (POTSTANDEN) de gegevens, volgens welke de potentiometerwaarden berekend moeten worden, ingelezen worden. Bij het inlezen van potentiometerstandengegevens wordt van onderstaande symbolische notatie gebruik gemaakt:

- PM0001 (= potentiometer nummer één)
- AU0001 (= Maximale waarde Analoge Uitgang AD nummer één)
- AI0001 (= Maximale waarde Analoge Ingang DA nummer één)
- MI0001 (= Maximale waarde Ingang FG0001)
- MU0001 (= Maximale waarde Uitgang FG0001)
- CO0001 (= Constante nummer 1)

Dit laatste label wordt gebruikt voor de in tabel 0064 opgeslagen constanten.

De wijze waarop het inlezen van potentiometerstanden moet gebeuren is het eenvoudigst aan de hand van een voorbeeld toe te lichten.

PM0001 = AU0001/MI0002;
PM0002 = AU0002/MI0002;
PM0003 = AU0003/MI0003;
PM0004 = MU0002/MU0003;
PM0005 = MU0001* MU0002 /MU0003/MI0004;
PM0006 = MU0004/AI0001;

Binnen de omlijsting staat de informatie zoals deze de PDP8 ingevoerd moet worden. Daarbij moet opgemerkt worden dat alle symbolen achter elkaar (zonder spaties of nieuwe regel) moeten staan. Alleen achter de puntkomma mag kommentaar volgen. De delingen (/) en vermenigvuldigingen (*) worden zonder onderscheid in volgorde van aanvraag afgehandeld, waarbij de bewerking wordt toegepast op de eerste term of op het verkregen tussenresultaat.

verwerken.

Het programmadeel dat de potentiometerstandeninput verwerkt begint op het adres gelabeld POT.

De initialisatie van het datagebied (LGA en LBA) gebeurt door LBA gelijk aan BALPM en LGA gelijk aan LGAPM te maken. Deze adressen bevatten het begin-adres van de potentiometerstandenadressenlijst en het laatstgevulde adres in het potentiometerdatagebied.

Het decoderen van het ingelezen buffer gebeurt met de *subroutine OPDO*.

Het decoderen gebeurt door het testen van de karakters of combinatie van karakters uit het inleesbuffer. Is het karakter een dollar dan is de taak klaar en wordt naar de monitor gesprongen.

De getal identifiers waar op getest wordt zijn de letterkombinaties van de gebruikte labels. Nadat de identifier voor het nummer van de potentiometer gevonden is volgt voor een afwijking van de beschreven input een foutmelding tenzij de potentiometerafsluiter (;) reeds geïdentificeerd is. De labels die op een funktietabel betrekking hebben (bijvoorbeeld MU002 en MI0001) worden in een woord gedecodeerd. De eerste 9 bits van een woord bevatten het nummer; de laatste 3 bits vormen het getal nul voor MI en één voor MU. De labels die op programmatabellen betrekking hebben (bijvoorbeeld AI0001, AU0002 en CO0002) worden eveneens in een woord gedecodeerd. De inhoud van de eerste 9 bits van dit woord geeft het nummer weer, de laatste 3 bits vormen het getal nul voor AI, één voor AU en twee voor CO. Ter onderscheid met de vorige woorden deze woorden negatief in het gedecodeerde informatieblok gezet.

Voor de bewerking delen (/) wordt niets voor de bewerking vermenigvuldigen min één (7777_8) in het informatieblok geplaatst. Ter afsluiting (;) van de potentiometerstand wordt nul (0000_8) in het gedecodeerde informatieblok geplaatst, waarna de decodering beëindigd wordt.

Na het opbergen van het informatieblok wordt het laatstgevulde adres (LGA) van het potentiometerstandendatagebied (LGAPM) onthouden dat wil zeggen LGAPM wordt gelijk aan LGA. Vervolgens wordt getest of het datagebied niet overschreden is. Is dit het geval dan volgt een foutmelding anders wordt tot het decoderen van de rest van het inleesbuffer overgegaan.

V-3-6 Het inlezen van een programma.

Voor het inlezen van programmabanden in binaire code wordt gebruik gemaakt van de BIN LOADER (een DEC programma). Deze BIN LOADER is zodanig gewijzigd*) dat deze vanuit de monitor met taak 4 (PROGRAMMA) kan worden aangeroept. Na inlezen van dat programma wordt naar de monitor teruggesprongen waar het al of niet korrekt inlezen van de programmaband (pariteitscheck) gemeld wordt. Het beginadres van het programmainput-gedeelte is het adres gelabeld LOAD.

V-4 Het rekengedeelte.

Het rekengedeelte van het hybride systeem bestaat uit een tweetal afzonderlijk met de monitor aan te roepen taken.

De eerste van deze taken, taak 5 (INITIALISATIE) voert die berekeningen voor het hybride rekenprogramma uit waarvoor alle gegevens tevoren bekend zijn (bijvoorbeeld potentiometerwaarden).

Daarnaast wordt door deze taak een deel van de organisatie voor het hybride rekenprogramma verricht.

De tweede van bovengenoemde taken, taak 6 (STATTEST) zorgt voor het hybride rekenwerk. Dat wil zeggen dat bij signalen die van de analoge rekenmachine komen bijbehorende signalen voor de analoge machine worden uitgerekend.

V-4-1 Organisatie initialisatie van het hybride rekenprogramma.

De ingelezen gegevens moeten verwerkt worden tot bruikbare informatie voor het hybride rekenprogramma. Deze verwerking houdt onder meer in:

1. Het initialiseren van de Functie Generator lijst.
2. Het initialiseren van de potentiometerwaardenlijst.
3. Het initialiseren van het aantal nodige A/D- en D/A-omzetter.

*) De wijzigingen in de BIN LOADER worden aangebracht tijdens de initialisatie van het systeem nadat de systeempromatuur door de ongewijzigde BIN LOADER is ingelezen. Deze wijzigingen betreffen twee adressen het startadres en het eindadres van de BIN LOADER;

Funktiegeneratorlijst.

De funktiegeneratorlijst bevat voor iedere funktiegeneratoraanvraag in het hybride rekenprogramma twee parameters. Deze parameters zijn het aantal plaatsen waarover de ingang van de betreffende funktiegenerator geschoven moet worden en het adres, waarin de funktiewaarde, die bij de nulingang van de betreffende funktiegenerator behoort, staat. De volgorde, waarin de parameterparen voor de verschillende funktiegeneratoren in de funktiegeneratorlijst geplaatst zijn is gelijk aan de volgorde waarin de verschillende funktiegeneratoren in het hybride rekenprogramma verwerkt worden (tabel 60).

Potentiometerwaardenlijst.

De potentiometerwaardenlijst bevat de waarden van de diverse potentiometers. In deze lijst wordt iedere potentiometerwaarde door twee opeenvolgende adressen in semi-floating point weergegeven. De volgorde waarin de potentiometerwaarden in de potentiometerwaardenlijst staan, is gelijk aan de volgorde waarin de betreffende potentiometers in het hybride rekenprogramma worden aangeroepen (tabel 61).

De taak die de vóórinitialisatie van het hybride rekenprogramma uitvoert is taak 5 (INITIALISATIE). Het beginadres van deze taak is het adres gelabeld INIT. Het plaatsen van een parameterpaar in de FGlijst verzorgt de *subroutine FGTA*. Deze subroutine berekent tevens de maximale waarden van de funktiegenerator in- en uitgang die bij het berekenen van de potentiometerstanden nodig zijn. Het berekenen en plaatsen in de potentiometerwaardenlijst van een potentiometer wordt verzorgd door de *subroutine PMSET*. De gegevens die voor het juist functioneren van deze subroutines nodig zijn, zijn het nummer (tabel of PM) en de plaats (pointer) waar de parameters in de lijst (FG of PM) opgeborgen moeten worden. Deze pointer wordt bij de eerste aanroep van een van deze subroutines op het beginadres van de betreffende lijst gezet. Bij iedere volgende aanroep van dezelfde subroutine is deze pointer twee plaatsen verhoogd. De volgorde van de nummers waarmee de subroutines *FGTA* en *PMSET* worden aangeroepen is vastgelegd door de tabellen 60 en 61.

Het aantal A/D- en D/A-omzetters, dat in het hybride rekenprogramma omgezet wordt, staat op de adressen gelabeld AANAD en AANDA. Deze adressen worden gevuld met de aantallen A/D-en D/A-omzetters welke door de gebruiker in de tabellen 62 en 63 zijn opgegeven.

De subroutine FGTA.

De tabelhead van de funktietabel wordt van het datagebied overgeplaatst naar een ruimte op bladzijde nul, zodat de tabelheadwaarden direkt beschikbaar zijn.

Uitgemaakt wordt nu op grond van de opgegeven geschatte extreme waarde van de funktiegeneratoringang hoeveel funktieintervallen nodig zijn. Het aantal funktieintervallen wordt vervolgens aangevuld tot de eerstvolgende macht van twee zodat iedere interval één éénduidig door een geschikt aantal bits (i) kan worden weergegeven.

Dit aantal bits bepaald het aantal shifts dat in de FG lijst wordt opgeborgen ($11-i$).

Hierna wordt bepaald of het aantal intervallen in de bestaande funktietabel toereikend is. Is dit het geval dan wordt het adres de funktiewaarde, die behoort bij de nulingang, in de FG lijst geplaatst.

Indien het aantal intervallen niet toereikend is dan wordt in het funktietabellendatagebied, aansluitend aan de laatst geplaatste funktietabel (LGAFG) een nieuwe funktietabel gemaakt (zonder tabelhead) die is aangevuld met de laatste funktiewaarde (voor negatieve ingangen ook met de eerste funktiewaarde) totdat het aantal intervallen toereikend is.

Vervolgens wordt het adres van de funktiewaarde die behoort bij de nulingang van deze nieuwe tabel in de FG lijst geplaatst ^{*}). In de tabelhead worden de maximale extreme waarden van de ingang (= aantal intervallen * stapgrootte) en uitgang (= 3777_8) opgeborgen, welke nodig zijn voor het berekenen van de potentiometerwaarden.

^{*}) Aanvullen en verplaatsen van de funktietabellen wordt gedaan om bij bereikoverschrijding van de funktietabel uit te kunnen maken wat er moet gebeuren. Hetzij een foutmelding geven; hetzij de grenswaarde als uitgang nemen (zie paragraaf IV-3).

De subroutine PMSET.

Met het potentiometernummer wordt het beginadres van het blok, waarin de potentiometerstand in machinecode staat weergegeven, uit de potentiometeradressenlijst opgehaald.

Voor het ophalen van een extreme waarde uit het datagebied zorgt de *subroutine NORPM*. Deze subroutine decodeert de code voor de extreme waarde en haalt via de funktietabellenadressenlijst de betreffende extreme waarde uit de tabelhead van de funktietabellen of uit de programmatabellen (tabel 63 tot en met 64) op.

De opgehaalde extreme waarde bestaat uit een getal met bijbehorende macht van tien. Van de opgehaalde macht van tien wordt vier afgetrokken, dit vindt zijn oorzaak in de gebruikte omzetting van decimale naar binaire getallen (*OKTA*). De opgehaalde getallen worden in semi-floating point notatie omgezet.

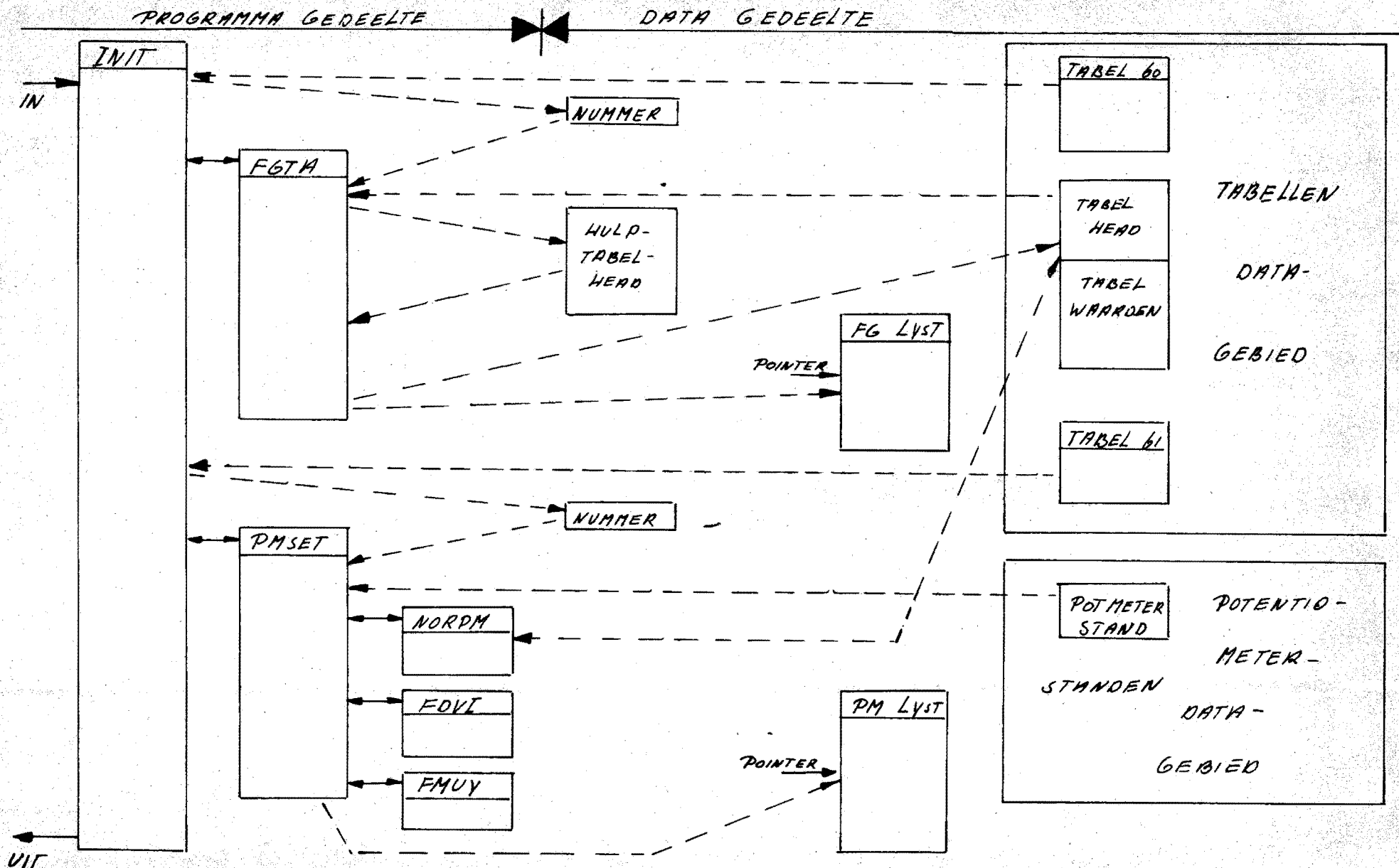
De berekening van de potentiometerwaarden gaat in semi-floating point. Hiervoor dienen de subroutines *FDUI* (deling) en *FMUY* (vermenigvuldiging) welke de resultaten in 11 bit nauwkeurig afronden. De berekening wordt in stappen uitgevoerd, waarbij het tussenresultaat in drie adressen genoteerd staat, een mantisse, een binaire exponent en een decimale exponent. Indien het eindresultaat op deze wijze berekend is wordt het omgerekend tot een getal bestaande uit een mantisse en een binaire exponent. Dit gebeurt door het eindresultaat (afhankelijk van de decimale exponent) een aantal keer met tien te vermenigvuldigen of door tien te delen. De op deze wijze verkregen potentiometerwaarde wordt opgeborgen in de potentiometerwaardenlijst.

Een overzichtsschema van de vóórinitialisatie is gegeven in figuur V-3.

V-4-2 Het hybride rekenprogramma.

Het digitale gedeelte van het hybride rekenprogramma wordt gestart met taak nummer 6 (STATTEST). Deze taak bestaat uit een viertal deeltaken:

1. Het inlezen van analoge signalen;
2. Het verwerken van deze signalen;
3. Het uitlezen van berekende waarden als analoge signalen;
4. Het synchroniseren van beide rekenmachines.



Figuur V-3.

Overzichtsschema van de vóórinitialisatie van het hybride rekenprogramma.

De tweede van bovengenoemde deeltaken bestaat deels uit harde deels uit zachte software. Onder de harde software vallen de subroutines welke de analoge elementen (zie paragraaf IV-3) vervangen, terwijl onder zachte software de wijze waarop en de volgorde waarin deze subroutines aangeroepen worden verstaan wordt. Deze zachte software die specifiek is voor het te bestuderen probleem wordt door de gebruiker geprogrammeerd in Assembler, vertaald en als binband met taak 4 (PROGRAMMA*) ingelezen. Het digitale rekenprogramma wordt in een gesloten lus doorlopen, dat wil zeggen na beëindigen van deeltaak 4 wordt weer aan deeltaak 1 begonnen.

Het inlezen van analoge signalen.

Het selekteren van A/D-omzetters gebeurt via de multiplexer. De omzetting van analoge naar digitale signalen vergt enige tijd. Om de hybride re-
kentijd zo efficiënt mogelijk te benutten worden gedurende het omzetten van de analoge signalen enige noodzakelijke handelingen verricht. Deze handelingen zijn:

1. Tijdens de eerste A/D-omzetting:

Het initialiseren van het verwerkingsgedeelte onder andere het re-
setten van pointers en tellers.

2. Tijdens de tweede en volgende A/D-omzettingen.

Het in semi-floating point notatie omzetten van het daarvoor inge-
lezen signaal.

De ingelezen waarden worden in semi-floating point opgeborgen in een buffer op bladzijde nul, zodat ze in willekeurige volgorde door direkte adres-
sering in het verwerkingsgedeelte aangeroepen kunnen worden.

Het verwerkingsgedeelte.

Het verwerkingsgedeelte bestaat voornamelijk uit subroutineaanroepen, die functioneren als analoge elementen.

*) Voor een voorbeeld zie Appendix C.

De rest van het rekengedeelte bestaat uit initialisatie van deze subroutines met de parameters die van de signaalgrootte afhangen. De initialisatie met de overige parameters is verzorgd door taak 5 (INITIALISATIE). Een geheel verwerkt signaal wordt van semi-floating point notatie omgezet in een analoog geschaald signaal en opgeborgen in een buffer. Dit gebeurt met de subroutine *FIXT*^{*}). Deze subroutine bergt de aangeboden signalen op volgorde van aanvraag in het uitleesbuffer op.

Uitlezen van D/A-omzetters.

Nadat alle signalen verwerkt zijn en het uitleesbuffer geheel gevuld is, worden in het uitleesbuffer opgeborgen signalen een voor een in analoge signalen omgezet; hiervoor is geen extra omzettijd nodig.

De betreffende converters worden door met één ophogen van de multiplexier geselecteerd. De volgorde waarin de signalen uitgelezen worden is gelijk aan de volgorde waarin ze in het uitleesbuffer opgeborgen zijn.

Synchronisatie.

Om de sampleperiode met de hardware klok te kunnen regelen, wordt de klok voor iedere digitale rekenslag gestart. Na het eindigen van een rekenslag wordt getest of de klokvlag op is; is deze op dan wordt deze ge-cleared en met het programma verder gegaan, zo niet dan wordt op de klokvlag gewacht.

Vervolgens wordt de interrupt faciliteit aangezet en gewacht op een interrupt.

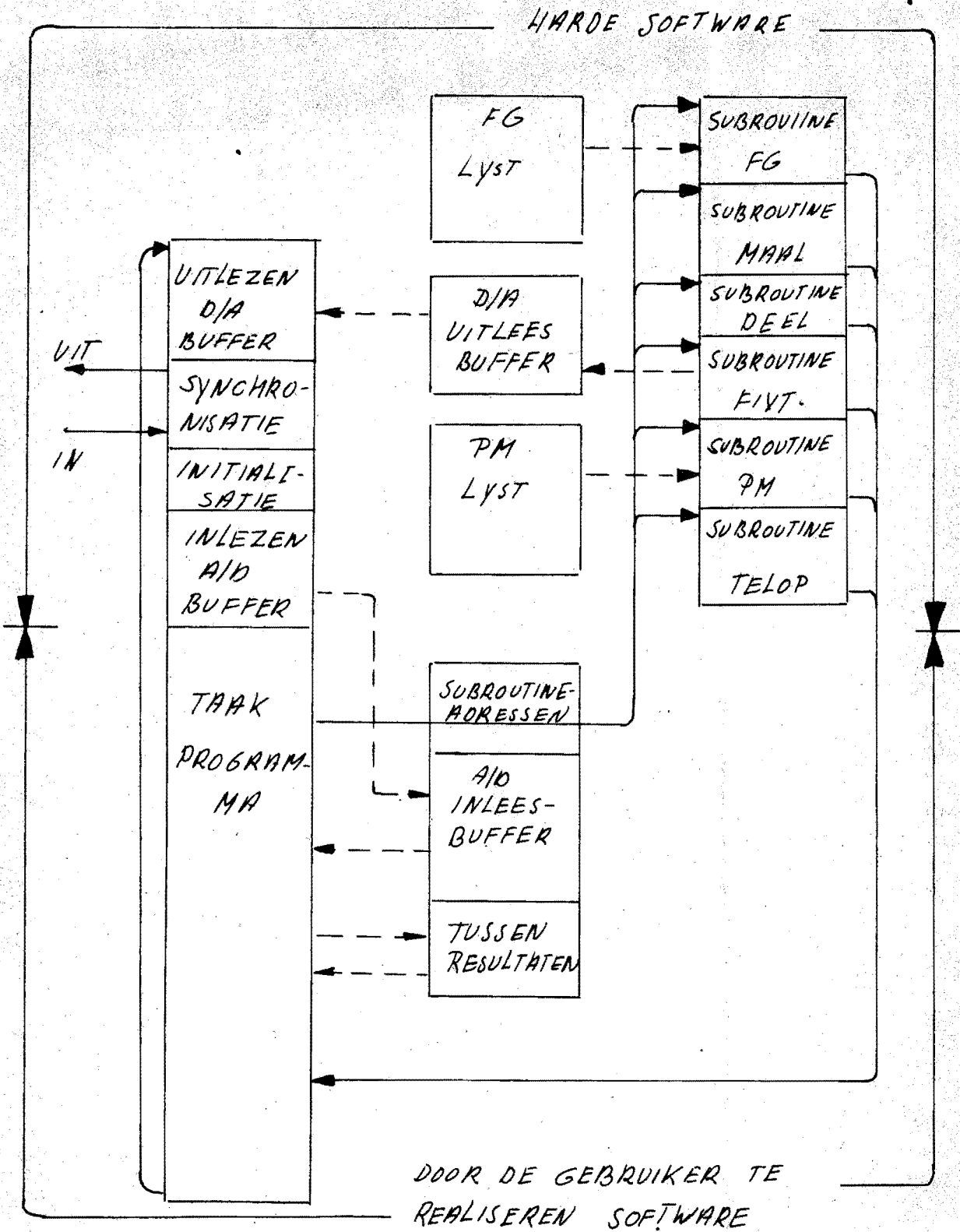
Een interrupt kan komen van:

1. De operate vlag van de analoge rekenmachine;
2. De keyboard vlag van de teletype.

Indien deze interrupt niet veroorzaakt is door de keyboardvlag dan wordt aan de volgende rekenslag begonnen. Is deze interrupt wel door de keyboardvlag veroorzaakt dan wordt naar de programmamonitor gesprongen waar medegedeeld wordt dat het rekenprogramma is uitgevoerd.

In figuur V-4 is een overzichtsschema van het hybride rekenprogramma gegeven.

^{*}) Zie Appendix A.



Figuur V-4.

Overzichtsschema van het hybride rekenprogramma.

VI TESTSIMULATIES MET HET GEKOPPELDE SYSTEEM.

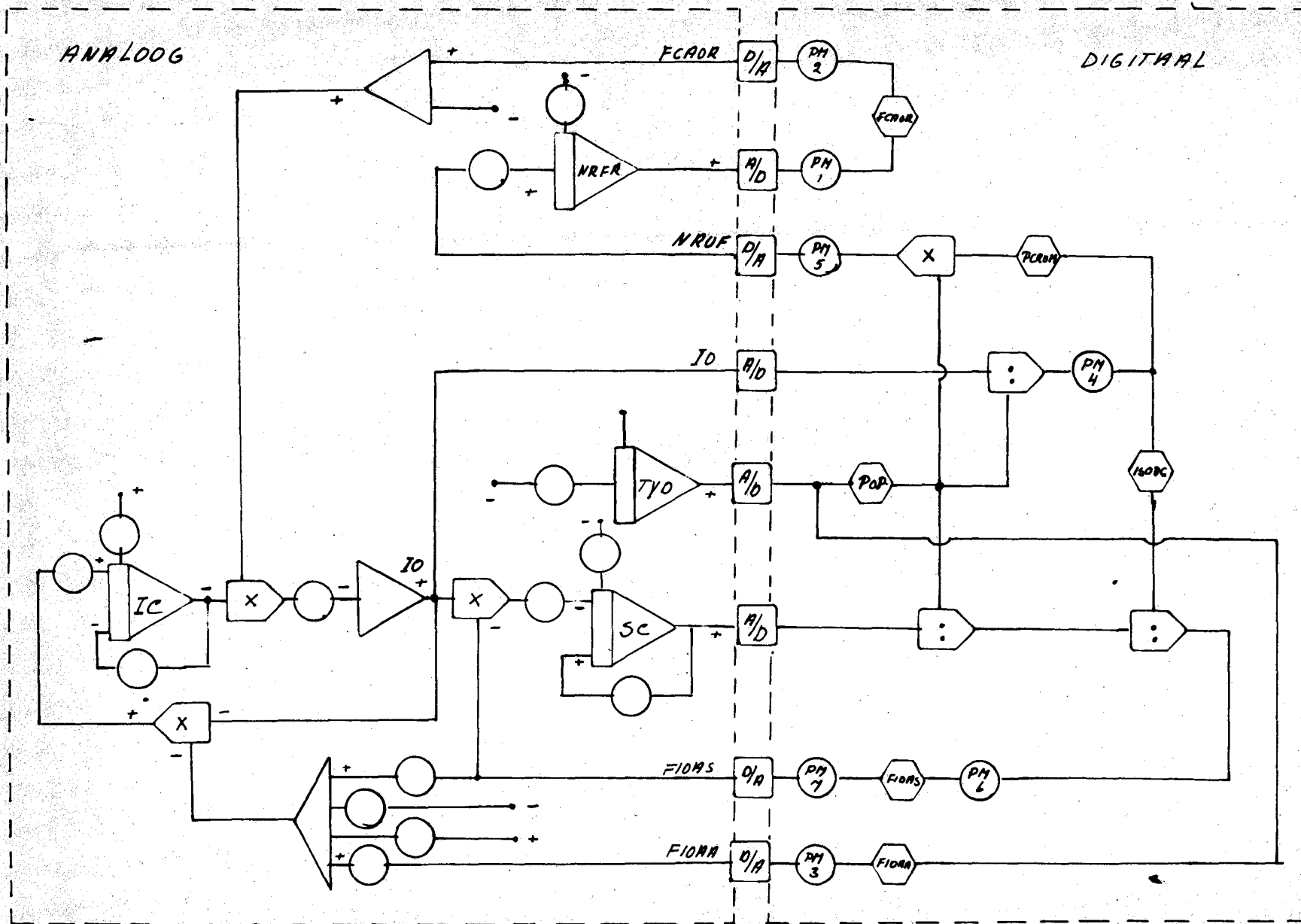
Om het gekoppelde systeem van analoge rekenmachine met de PDP8 op zijn bruikbaarheid te testen zijn voor een gedeelte van Wereldmodel 3 enige hybride rekensimulaties uitgevoerd.

Het gedeelte van Wereldmodel 3, dat voor deze testsimulatie gebruikt is, is de kapitaal-grondstoffensektor, waarvan in paragraaf IV de modelvergelijkingen en een principeschema voor de taakverdeling tussen analoge en digitale rekenmachine gegeven zijn.

De analoge rekenmachine, waarvan bij deze testsimulaties gebruik werd gemaakt, is een RA741 Telefunken, welke alleen geschikt is voor het analog simuleren van kleine systemen. Vanwege het gering aantal multipliers (vier) is het noodzakelijk om voor deze simulaties een geringe verschuiving in taakverdeling tussen analoge en digitale rekenmachine aan te brengen. Als invloeden van buiten de kapitaal-sektor zijn genomen POP en FIOAA (zie tabel IV-2). Met behulp van een digitale funktiegenerator, welke als ingang de procestijd*) heeft wordt het global standard run gedrag van ieder van deze grootheden gesimuleerd.

In figuur VI-1 is het geschaalde rekenschema voor de hybride simulatie van de kapitaal-grondstoffensektor weergegeven. Het op grond van dit geschaalde rekenschema samengestelde digitale taakprogramma is te vinden in Appendix C. Tevens staan hierin de gebruikte funktietabellen, programmatabellen, extreme waarden en digitale potentiometerstanden vermeld. Schattingen voor de extreme waarden van de verschillende grootheden zijn gemaakt op grond van de extreme waarden welke deze grootheden in de global standard run aannemen. Deze zijn met het in de Inleiding vermelde ALGOLprogramma van Wereldmodel 3 berekend.

*) Met de procestijd wordt hier bedoeld de tijd die tussen de jaren 1940 en 2100 verloopt.



Geschaald rekenschema van de kapitaal-grondstoffensektor.

Figuur VI-1.

I-1 Simulatieresultaten.

Om na te gaan in hoeverre de resultaten van het gekoppelde systeem betrouwbaar zijn zijn deze resultaten vergeleken met de resultaten van het ALGOLprogramma van Wereldmodel 3⁶⁾. Bij vergelijking van de met het ALGOLprogramma berekende krommen met de krommen, die door het gekoppelde systeem berekend zijn, zijn bij voldoende lange simulatietijd geen afwijkingen waar te nemen.

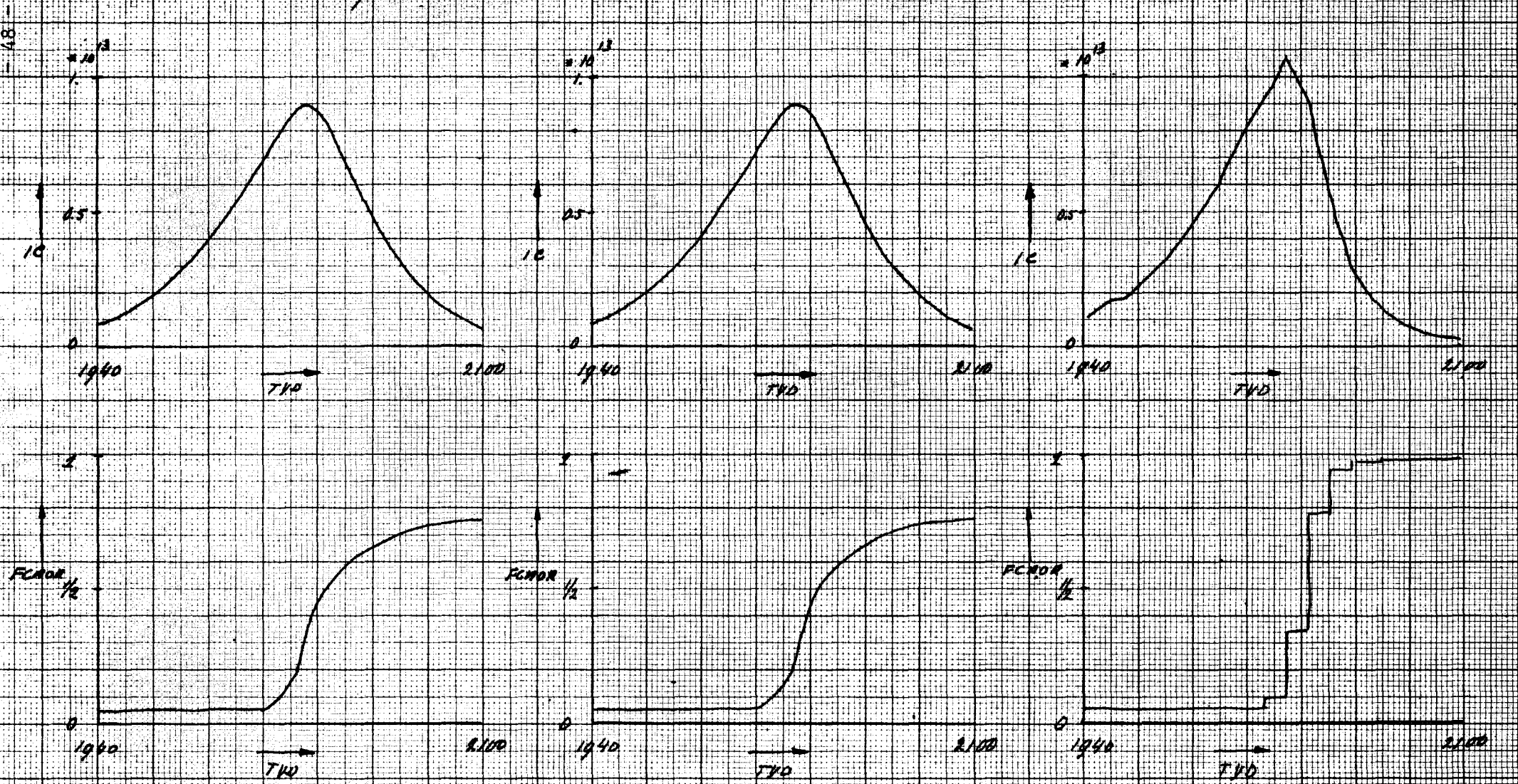
Voor dezelfde simulatietijden zijn bij verschillende samplefrequenties (die met de digitale klok ingesteld kunnen worden) een aantal krommen opgenomen om de invloed van het samplen op de resultaten na te gaan. Het verlagen van de maximale samplefrequentie met een faktor 50 had bij een simulatietijd van 16 seconden nauwelijks invloed op de resultaten. Bij verlaging van de samplefrequentie met een faktor 500 traden duidelijk dynamische afwijkingen op. In de figuren VI-2 en VI-3 zijn enige van deze simulatieresultaten weergegeven.

Op grond van deze inleidende metingen kan een schatting gemaakt worden van de tijd waarin het gehele Wereldmodel 3 door het hybride systeem met de gewenste nauwkeurigheid doorgerekend kan worden. Deze simulatietijd zal ongeveer vier seconden zijn, daar de verhouding van de rekentijden in dezelfde orde van grootte zal liggen als de verhouding van het aantal te genereren functies (1:7) van beide simulaties.

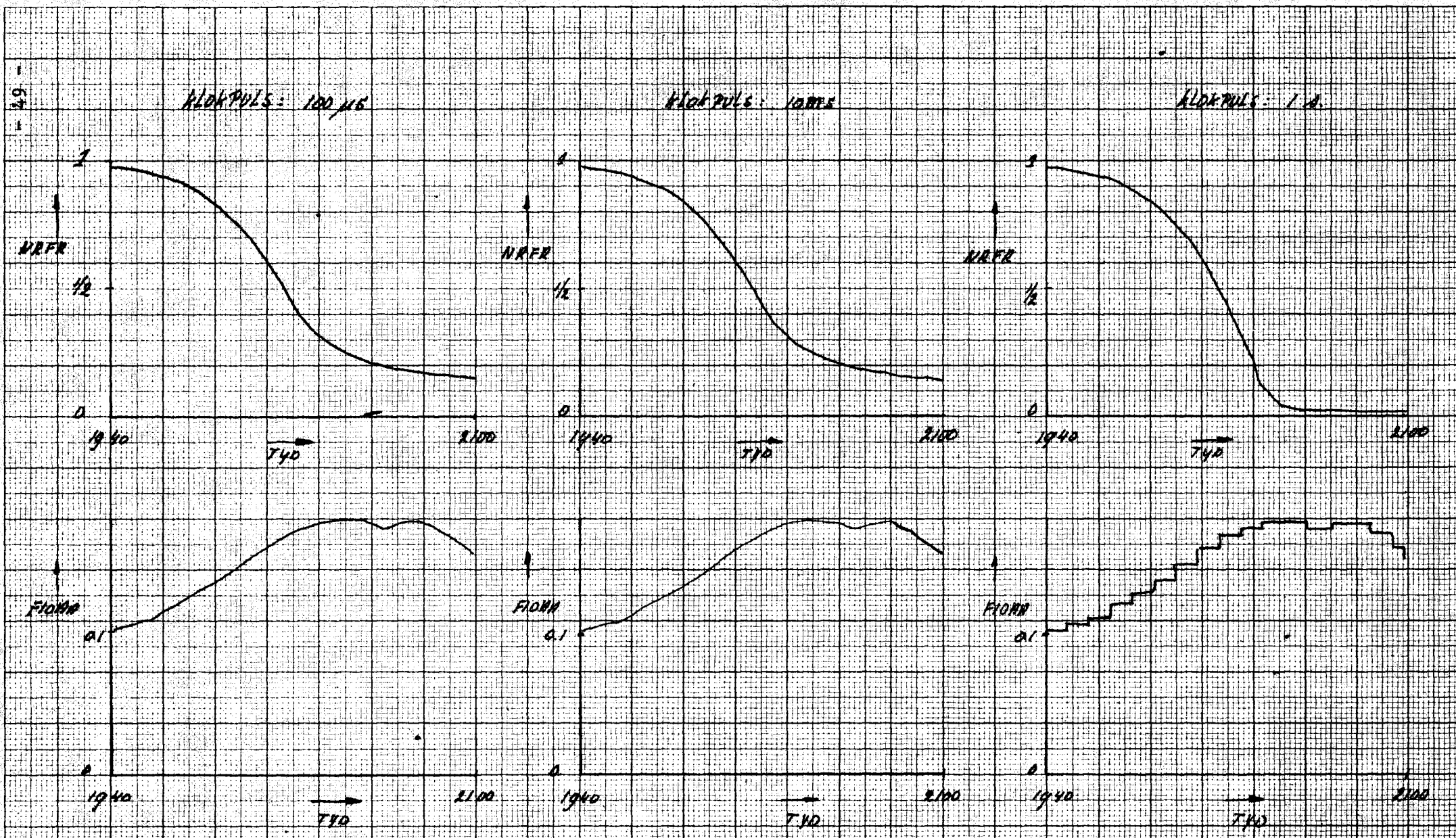
KLOK PULS: 100 μ s

KLOK PULS: 10 ms

KLOK PULS: 1 s



FIGUUR VI-2 : IC EN FCMOR IN 'GLOBAAL' STANDAARD RUN'
 BY VERSCHILLENDE SAMPLEREGENTIJS
 1 MINIMALE SAMPLEREG. IS 3 ms



FIGUUR II-3 HET 'GLOBAL STANDARD RUN' GEORIG VAN NRFR EN FIOMR BIJ VERSCHILLENDE SMPLEFREQUENTIE (MINIMALE SMPLE TUD IS 5 μs)

Lijst van subroutines:

- FDVI* : subroutine voor het uitvoeren van een deling.
- FGTA* : subroutines voor het initialiseren van de FG lijst.
- FIXT* : subroutine voor het vullen van het uitleesbuffer van het hybride rekenprogramma.
- FMUY* : subroutine voor het uitvoeren van een vermenigvuldiging.
- LBIN* : subroutine voor het inlezen van ASCII karakters in een buffer.
- LBUIT* : subroutine voor het ophalen van ASCII karakters uit het inleesbuffer.
- NORPM* : subroutine voor het ophalen van een extreme waarde.
- OKTA* : subroutine voor het omzetten van een decimaal getal in ASCII code naar een machinegetal.
- OBVS* : subroutine voor het plaatsen van een informatieblok in het datagebied.
- OPDO* : subroutine voor het decoderen van potentiometerstanden van ASCII naar machinecode.
- OTDO* : subroutine voor het decoderen van tabellen van ASCII naar machinecode.
- PMSET* : subroutine voor het initialiseren van de PM lijst.
- PRINLR* : subroutine voor het typen van een karakter op een nieuwe regel.
- PRINT* : subroutine voor het typen van een karakter.

Bijlage b.

Lijst van adressen:

AANAD	:	aantal nodige A/D-omzetters.
AANDA	:	aantal nodige D/A-omzetters.
BAALFT	:	beginadres adressenlijst van de funktietabellen.
BAAML	:	beginadres buffer herkenningwoorden.
BAML	:	beginadres monitorlijst.
BALPM	:	beginadres adressenlijst van de potentiometers.
INIT	:	beginadres van vóórinitialisatieprogramma.
LBA	:	beginadres van de lopende adressenlijst.
LGA	:	laatstgepulde adres van het lopende datagebied.
LGAFG	:	laatstgepulde adres van het funktietabellendatagebied.
LGAPM	:	laatstgepulde adres van het potentiometerstandendatagebied.
MAX	:	beginadres van het inleesprogramma voor de extreme waarden.
NUMMER	:	nummer van het lopende informatieblok.
POT	:	beginadres van het inleesprogramma voor potentiometerstanden.
TABEL	:	beginadres van inleesprogramma voor tabellen.

Literatuurlijst.

1. D.L. Meadows c.s.,
Rapport van de Club van Rome (AULA 500).
Het Spectrum N.V., Utrecht, 1972.
2. O. Rademaker, J.G.M. Cuypers,
Werkgroep Globale Dynamica Voortgangsverslagen 1 en 2.
Technische Hogeschool Eindhoven, 1972.
3. W.W. Behrens, D.L. Meadows,
The Dynamics of Growth in a Finite World (Preliminary Draft).
M.I.T., (1972-04-07).
4. A.L. Pugh,
Dynamo II, User's Manual,
The M.I.T. Press, 1970.
5. Small Computer Handbook.
Digital Equipment Corporation (1967).
6. M.J.B.M. Monen,
Het gedrag van de grootheden in de standaardrun van Wereld 3.
Informeel Notitie 59 Werkgroep Globale Dynamica (1972-07-11).