

MASTER

Videorecording met ADM en ADPCM

Hadderingh, J.N.

Award date:
1978

[Link to publication](#)

Disclaimer

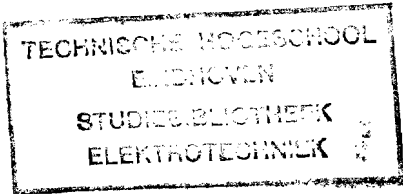
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository.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

3030 bse



Vakgroep Informatie en Communicatie Theorie

Vakgroepsvoorzitter Dr.ir.J.P.M.Schalkwijk

Videocodering met ADM en

ADPCM

door J.N.Hadderingh

Verslag van het afstudeer-
werk verricht in de perio-
de van april 1977 t/m sep-
tember 1978

Coaches:ir.J.A.M.de Brouwer

ir.J.E.Rooyackers

Inhoud

- 1 Inleiding
- 2 Coderen van videosignalen
- 3 Niet adaptieve coderingssystemen
 - 3.1 Deltamodulatie
 - 3.2 PCM
 - 3.3 DPCM
- 4 Adaptieve coderingssystemen
 - 4.1 ADM
 - 4.2 ADPCM
- 5 Adaptieve predictie en quantisatie
 - 5.1 Adaptieve predictie
 - 5.2 Voorstel voor een ander type adaptieve quantizer
 - 5.3 De invloed van quantisatie op predictie
 - 5.4 De invloed van een drempel in de predictie
 - 5.5 Driedimensionale systemen
 - 5.6 Samenvatting
 - 5.7 Conclusies
- 6 Het ADM programma
 - 6.1 Het programma
 - 6.2 Modificaties
- 7 Het ADPCM programma

8 Subroutines voor het ADPCM systeem

8.1 BEPMI, BEPNIV en LEESDR

8.2 BEPMSE en OUTPUT

8.3 BEPBZ, TABUIT, VEEG en DSHR

8.4 TABOUT, TABIN en AFLOOP

8.5 NEEMS en STUURS

8.6 Opbouw NEEMS en STUURS

8.7 Uitbreidingen en modificaties van NEEMS en STUURS

8.8 Het programma BEWEEG

9 Metingen en conclusies

9.1 Metingen en conclusies ADM

9.2 Metingen en conclusies ADPCM

Literatuur

Appendix A

Appendix B

Appendix C

Bijlage A

Bijlage B

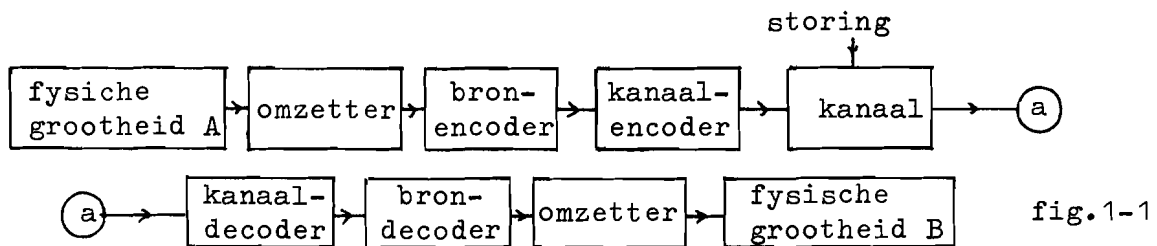
Bijlage C

Bijlage D

Bijlage E

Inleiding

In de communicatietechniek geeft men er tegenwoordig de voorkeur aan om van nature analoge signalen digitaal over te zenden. Bij het digitaliseren van audio- en videosignalen ontstaat een redundante bron. Er bestaan twee soorten redundantie nl. statistische en perceptuele redundantie. Statistische redundantie is een gevolg van het feit dat men uit het reeds ontvangen gedeelte van het bericht of door bepaalde voorkennis een deel van de informatie kan voorspellen. Perceptuele redundantie ontstaat doordat onze zintuigen of apparatuur maar voor een beperkt spectrum van prikkels gevoelig is. Bevinden zich in het signaal componenten die prikkels veroorzaken die buiten het spectrum vallen waarvoor onze zintuigen of apparaten gevoelig zijn, dan kunnen deze componenten worden weggelaten. Fig. 1-1 geeft het blokschema van een communicatie systeem.



De omzetter zet een fysische grootheid om in een elektrisch signaal. Daarna volgt de bronencoder die dit elektrische signaal omzet in een digitaal met een minimum aan bits.

De kanaalencoder voegt weer redundantie toe om ondanks storingen toch correcte decodering mogelijk te maken. De brondecoder zal uit de bitstroom van de kanaaldecoder een elektrisch signaal genereren dat "ongeveer" overeenkomt met het signaal dat de omzetter van de zender geleverd heeft. Vervolgens wordt dit signaal weer in een fysische grootte omgezet. Dat de brondecoder geen exacte copie van het input signaal kan leveren komt omdat er onderweg bewerkingen zijn uitgevoerd waarbij informatie verloren is gegaan. Het algemene streven zal nu zijn om met een minimaal aantal bits een zodanige copie van het origineel te ontvangen dat deze aanvaardbaar is. Dit "aanvaardbaar" is afhankelijk van het doel waarvoor het signaal gebruikt wordt.

Het bestaan van twee soorten redundantie maakt het mogelijk om de bronencoder in twee delen te splitsen. Het eerste deel verwijdert de perceptuele en het tweede deel de statistische redundantie. Voor het verwijderen van perceptuele redundantie kan gebruik worden gemaakt van de resultaten van de rate distortion theorie. Op de aldus verkregen bron kan dan noiseless source coding worden toegepast om de statistische redundantie te verwijderen. Door de in dit verslag beschreven ADPCM systemen wordt perceptuele redundantie en een deel van de statistische redundantie verwijderd.

Codering van Videobeelden

Een videobeeld is opgebouwd uit punten. Deze zijn gegroepeerd in lijnen en rasters. De correlatiefunctie van een reeks videobeelden is een functie van de drie plaatscoördinaten x, y en z . Er zijn nu verschillende mogelijkheden om de volgorde te kiezen waarin de beeldpunten door het coderingssysteem worden behandeld. Als men hiervoor x en/of y richting kiest dan spreekt men van intraframe, in de z richting van interframe systemen. In het laatste geval moet men het beeld opgebouwd denken als een veld van coderingssytemen. In de praktijk zal slechts een of een klein aantal systemen worden gebruikt met een geheugen voor de gegevens van de bronnen die niet bewerkt worden.

Nu heeft (A)DM een hogere samplefrequentie nodig dan een in kwaliteit vergelijkbaar (A)DPCM systeem. Dit maakt het simuleren van intraframe (A)DM op de huidige installatie (1978) tot een bijzonder omslachtige zaak daar de samplefrequentie vastligt, nl. 12.5 Mhz. Verhogen van deze frequentie naar bv. 25 Mhz is niet mogelijk. Men dient dus de tussenliggende samplewaarden te berekenen. Volgens het sampletheorema is dit mogelijk. Men maakt dan van de volgende formule gebruik:

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT) \frac{\sin\left(\frac{\pi t}{T} - n\pi\right)}{\frac{\pi t}{T} - n\pi}$$

T is de periodetijd tussen de samplewaarden

n is een geheel getal

f(x) is de samplewaarde op tijdstip x

Een probleem hierbij is dat deze reeks niet snel convergeert. Men moet dus veel termen meenemen voor de gewenste nauwkeurigheid bereikt is. Het hiervoor benodigde rekenwerk is dermate tijdrovend dat als men het op IMP16 microcomputers implementeert dit niet meer praktisch uitvoerbaar is. Het is niet mogelijk hier met een simpele lineaire interpolatie te volstaan. Uit het werk van Schmitz lit. (1) blijkt dat het grote probleem van ADM met videobeelden edgebusiness is. En juist door lineaire interpolatie zouden op de flanken van de signalen waar de edgebusiness ontstaat fouten worden gemaakt. Treedt er nu edgebusiness op dan is er niet meer na te gaan of dit een gevolg is van het ADM systeem of van foute interpolatie.

Bij interframe (A)DM ligt het probleem anders. Als er weinig of geen beweging is dan kan men wel (A)DM toepassen. Bij heftige beweging zal de deltamodulator deze niet kunnen volgen. Toch ligt hier een interessante mogelijkheid om redundantie reductie te verkrijgen. Deltamodulatie gebruikt slechts 1bit/pel en levert stilstaande en langzaam bewegende beelden van hoge kwaliteit.

Nu blijkt het oog niet in staat om beweging en de vorm van het bewegende voorwerp tegelijkertijd goed waar te nemen. Bv. beweegt een cirkelvormig voorwerp zich snel over het scherm dan is het oog niet in staat om die cirkelvorm te detecteren. Dit betekent dat bij snel bewegende beelden vervorming toelaatbaar is die door de kijker niet wordt waargenomen.

Een ADM systeem blijkt zich niet snel genoeg te kunnen aanpassen om een bevredigende weergave van snel bewegende beelden op te leveren. (A) DPCM systemen kunnen wel een goede weergave van snel bewegende beelden leveren. De prijs die hiervoor betaald moet worden is het gebruik van meerdere bits per pel. Wat nu moet worden onderzocht is met hoeveel niveaus kan worden volstaan om een bevredigende weergave te verkrijgen. Daar (A) DM systemen een goede weergave leveren bij stilstaande en langzaam bewegende beelden is het duidelijk dat het niet efficiënt is om een DPCM systeem toe te passen. Want om een goede werking bij weinig beweging te verzekeren is een fijne quantizer nodig die slechts een klein bereik hoeft te bezitten. Voor het volgen van een snelle beweging is daarentegen een quantizer met een groot bereik nodig. Het combineren van deze eisen leidt tot een systeem dat veel bits per pel gebruikt. Echter als de quantizer fijn moet zijn dan is dat grote bereik niet nodig terwijl bij heftige beweging vervorming toelaatbaar is en dus een grove quantizer.

Door de quantizer adaptief te maken moet het mogelijk zijn om aan beide eisen te voldoen zonder dat een groot aantal bits per pel wordt gebruikt.

Voor het coderen van bewegende beelden lijkt het dus aantrekkelijk om een ADPCM systeem toe te passen i.p.v. ADM of DPCM. Daar het geheugen van de simulatie apparatuur beperkt is kan er slechts een klein bewegend beeld gedurende korte tijd worden gerealiseerd. Hierdoor is het niet goed mogelijk om ADPCM systemen te simuleren die interframe werken. Door de korte reeksen beeldpunten die zo'n systeem dan te verwerken krijgt zullen inschakelverschijnselen een storende factor vormen. Verder kan men door beeldafmeting en tijdsduur slechts een indicatie van de kwaliteit van het systeem verkrijgen. Door intraframe ADPCM systemen te bestuderen kan aan deze beperkingen worden ontkomen. De opbouw van de simulatie apparatuur maakt het overigens mogelijk om met dezelfde programmatuur interframe i.p.v. intraframe ADPCM systemen te simuleren.

3.1 Delta Modulatie

Fig.3-1 geeft het blokschema van een delta modulator. Dit is in zijn grondvorm een eenvoudig te implementeren vorm van AD conversie.

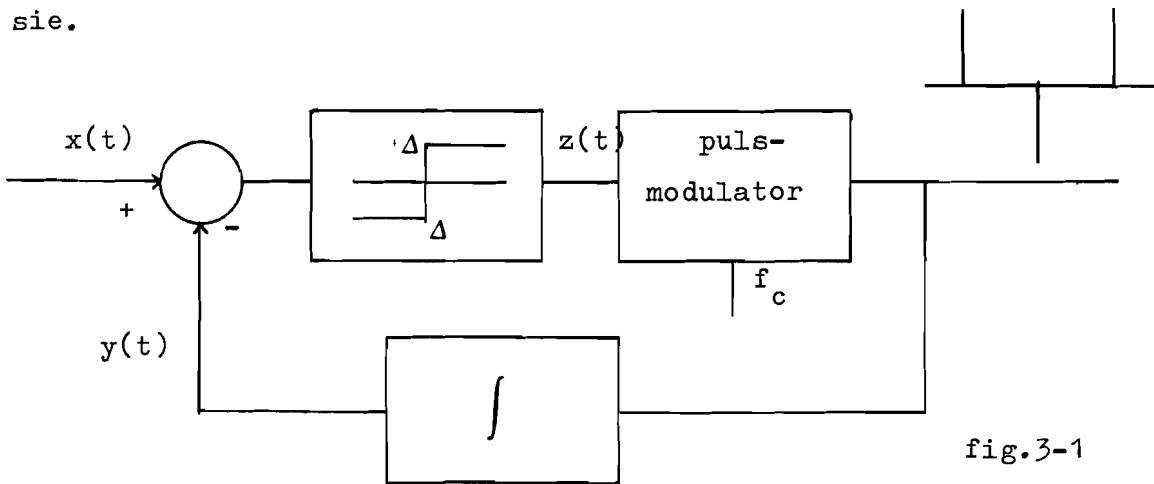


fig.3-1

Het analoge signaal $x(t)$ wordt vergeleken met het niveau van het gediscretiseerde signaal $y(t)$.

$$\text{als } x(t) \geq y(t) \text{ dan } z(t) = \Delta$$

$$\text{als } x(t) < y(t) \text{ dan } z(t) = -\Delta$$

Fig.3-2 geeft het verband tussen $x(t)$ en $y(t)$.

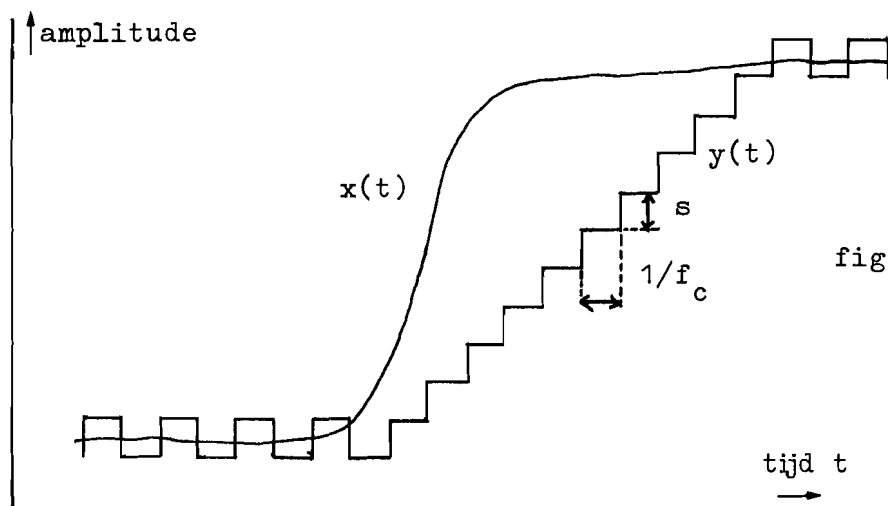


fig.3-2

Bij deze vorm van AD conversie worden twee soorten fouten gemaakt, korrelfouten en oversturingsfouten. Deze fouten veroorzaken ruis bij de ontvanger.

Korrelruis is een gevolg van de eindige afmetingen van de stapjes s waarmee het analoge signaal wordt benaderd.

Bij oversturingsruis kan de helling van het analoge signaal $x(t)$ door $y(t)$ niet worden gevolgd.

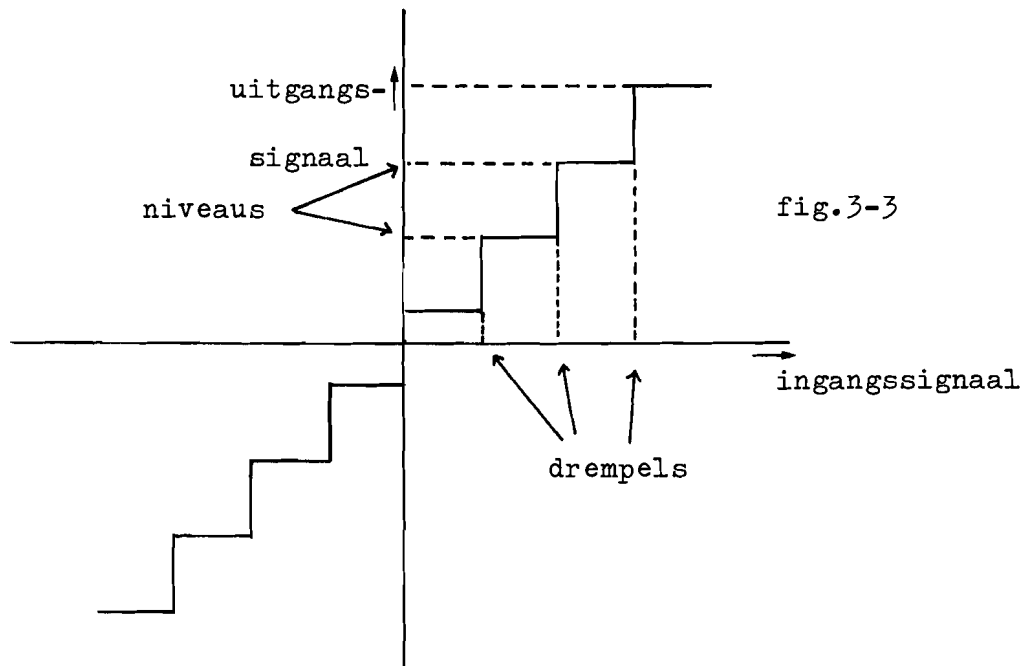
De remedie tegen korrelruis is het gebruik van een kleinere stap s . Die tegen oversturingsruis een grotere stap of het verhogen van f_c . Het gevolg is dat er voor ieder lineair DM systeem een optimale stapgrootte en samplefrequentie is waarbij de bijdrage van korrel- en oversturingsruis gelijkwaardig zijn. Het bepalen van deze optimale stapgrootte bij gegeven samplefrequentie is het probleem dat men bij het ontwerpen van een DM systeem dient op te lossen.

Kent men zijn ingangssignaal dan is dit mogelijk.

Als het ingangssignaal niet stationair is dan werkt een lineaire DM niet altijd optimaal. Om hier verbetering in te brengen moet de stapgrootte aangepast worden aan het ingangssignaal. Dan is er sprake van adaptieve delta modulatie.

3.2 Puls Code Modulatie

Bij puls code modulatie (PCM) wordt de amplitude van het ingangssignaal gequantiseerd. Fig. 3-3 geeft de overdrachtsfunctie van de quantizer. Bij elk niveau hoort een codewoord van



$n = 2^m$ bits met m het aantal drempels. Die n bits vormen het digitale uitgangssignaal. De quantizer is door twee series grootheden bepaald. Dit zijn de drempels en de niveaus. De drempels zijn de grenzen van de intervallen die op het zelfde niveau worden afgebeeld. Door Max lit. (2) is een methode ontwikkeld om een optimale keuze van drempels en niveaus te kunnen maken bij gegeven verdelingsfunctie van het ingangssignaal met het aantal niveaus als parameter. Stel:

$$D = E \{ f(\epsilon) \} = E \{ f(s_i - s_u) \} \text{ waarbij}$$

$f(x)$ een differentieerbare functie

ϵ de quantiseringsfout

D de vervorming

Kies N drempels $x_1 \dots x_N$ en $x_{N+1} = \infty$ Dan geldt:

$$D = \sum_{i=1}^N \int_{x_i}^{x_{i+1}} f(x-y_i) p(x) dx \quad (1)$$

waarbij y_i het uitgangsniveau voorstelt dat bij x_i hoort.

nu worden $\frac{\partial D}{\partial x_i} = f(x_i - y_{i-1})p(x_i) - f(x_i - y_i)p(x_i) \quad i=1 \dots N$

$$\frac{\partial D}{\partial y_i} = - \int_{x_i}^{x_{i+1}} f'(x-y_i) p(x) dx \quad i=1 \dots N$$

Nul stellen van $\frac{\partial D}{\partial x_i}$ en $\frac{\partial D}{\partial y_i}$ levert:

$$f(x_i - y_{i-1})p(x_i) = f(x_i - y_i)p(x_i) \quad (2)$$

$$\int_{x_i}^{x_{i+1}} f'(x-y_i) p(x) dx = 0 \quad (3)$$

Meestal kiest men voor $f(x) = x^2$. Dan is er sprake van Mean Squara Error. Daar $f(x) = -f(x)$ geldt nu voor de niet triviale oplossing van (2) en (3):

$$(2) \quad y_i = 2x_i - y_{i-1} \quad (4)$$

$$(3) \quad \int_{x_i}^{x_{i+1}} (x-y_i) p(x) dx = 0 \quad (5)$$

De MSE norm mag dan mathematisch plezierige eigenschappen hebben, hij stemt niet overeen met de norm die ons oog gebruikt. Uit perceptieve overwegingen heeft $f(x) = x^6$ lit. (3) de voorkeur.

$$(2) \quad y_i = 2x_i - y_{i-1} \quad (6)$$

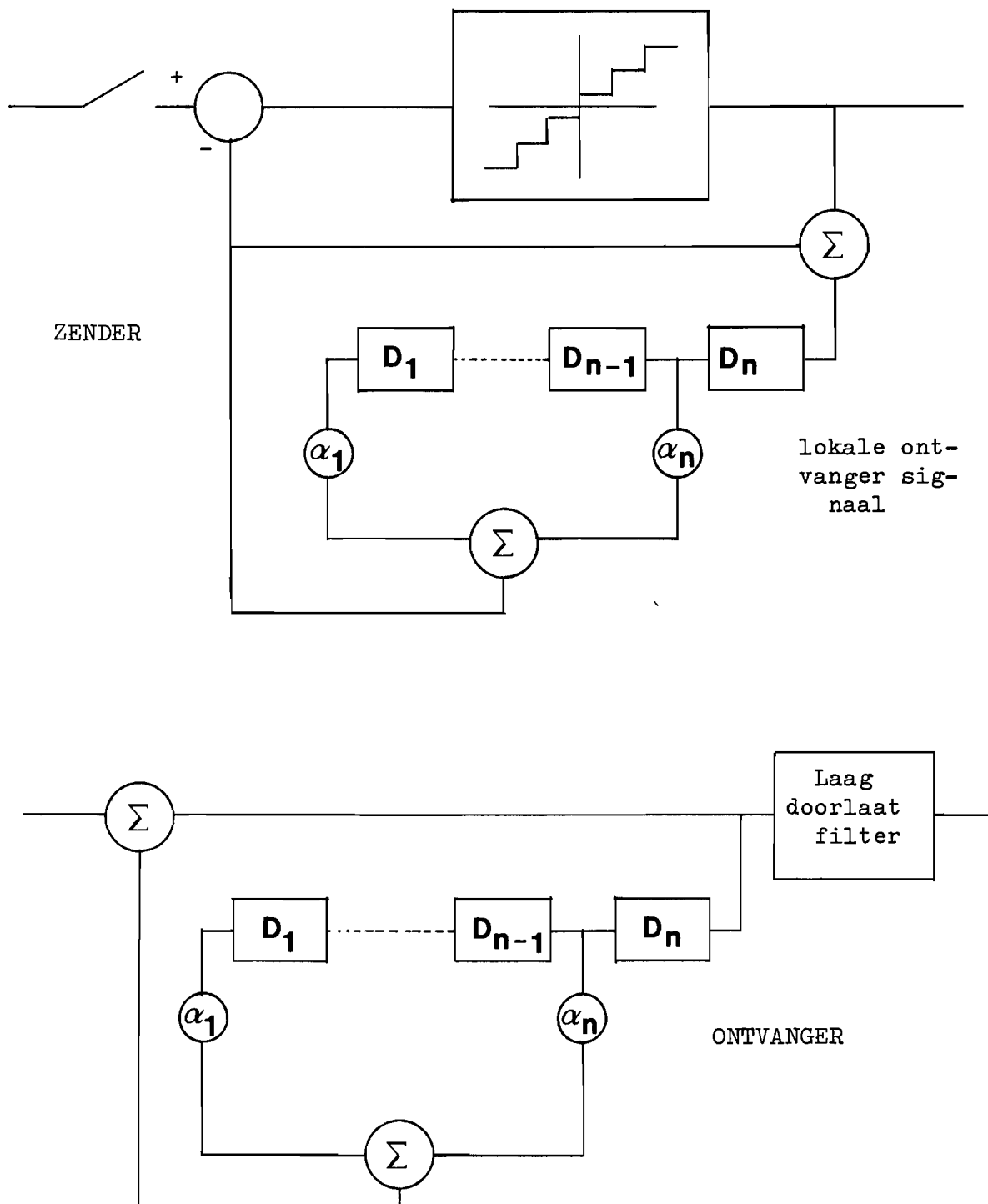
$$(3) \quad \int_{x_i}^{x_{i+1}} (x-y_i)^5 p(x) dx = 0 \quad (7)$$

Het is meestal niet mogelijk om voor de stelsels (4), (5) en (6), (7) expliciete oplossingen te vinden bij gegeven verdelingsfunctie $p(x)$. Men kiest daarom een y_1 rekest x_2 uit, bepaalt y_2 enz. en controleert of y_N voldoet aan (5) of (7). $x_N = \infty$. Is dit correct dan was y_1 correct en dus alle y_i 's en x_i 's. Zoniet kies andere y_1 . Voor de normale verdeling heeft Max lit. (2) dit gedaan. Voor de Laplace verdeling zie Paez en Glisson lit (4). Netravali lit. (5) heeft experimenteel quantizers bepaald zo dat de waargenomen fout minimaal was. Zijn resultaten vertonen veel overeenkomst met de aanname $f(x) = x^6$.

3.3 Differentiële Puls Code modulatie (DPCM)

Fig.4-1 geeft het blokschema van een DPCM systeem. Het is een predictieve methode van codering en het systeem bestaat uit een quantizer en een predictor. Uit de amplitude van een of meer voorgaande samples bepaalt de predictor een schatting voor het te coderen sample. Aan de quantizer wordt nu het verschil tussen deze schatting en het sample toegevoerd. Meestal zal de verdelingsfunctie van het aan de quantizer toegevoerde signaal een kleinere spreiding hebben dan de verdelingsfunctie van het te coderen signaal. De quantizer kan dan met minder niveaus werken om het signaal met een zelfde hoeveelheid vervorming te coderen. Daar dit kleinere aantal niveaus met minder bits kan worden gecodeerd geeft zo'n systeem zowel perceptuele als statistische redundantie reductie. Dit in tegenstelling tot een PCM systeem dat slechts perceptuele redundantie kan verminderen. Door de differentiele werking is het systeem gevoeliger geworden voor transmissie fouten. Het ontwerp van een DPCM systeem kan in twee delen worden gesplitst, te weten het ontwerp van de quantizer en dat van de predictor. Door eerste de quantizer kort te sluiten kan bij gegeven ingangssignaal de predictor bepaald worden. Dit gebeurt met de theorie van de lineair mean square estimation die gegeven een stationair signaal een optimaal resultaat levert. Is de predictor berekend dan wordt de verdelingsfunctie van het ingangssignaal

fig.3-4



van de quantizer bepaald. Nu kan met de theorie van Max, zie 3.2 de quantizer gedimensioneerd worden. Deze procedure kan nu recursief worden gemaakt. Men berekent dan het gequantiseerde signaal waarmee de predictor werkt. Door deze quantisatie is dit signaal veranderd en moet de predictor worden veranderd. Deze andere predictor levert weer een veranderd ingangssignaal aan de quantizer enz. Door nu net zo lang te rekenen totdat predictor en quantizer niet meer veranderen heeft men een optimaal systeem ontworpen. Indien het ingangssignaal van het DPCM systeem niet stationair is dan kan deze procedure niet zonder meer worden gevolgd daar de bereikte resultaten niet langer optimaal zijn. Om het DPCM systeem dan beter te laten functioneren moet het adaptief worden gemaakt. Hierop wordt in 4.2 en 5 dieper ingegaan.

4.1 Adaptieve Delta Modulatie (ADM)

Als men bij een lineaire DM de oversturingsruis wil verkleinen zonder de samplefrequentie te verhogen dan moet men de stapgrootte aanpassen aan de helling van het signaal. Zodra echter DE oversturing voorbij is dient de oorspronkelijke stapgrootte te worden hersteld ter vermijding van extra korrelruis. Deze verandering van stapgrootte moet aan de ontvanger worden meegegeeld. De ontvanger krijgt slechts de uitgangsbits van de zender. Uit dit signaal zal dan ook de informatie over het al dan niet vergrooten van de stap moeten worden gehaald. Wordt een reeks van uitsluitend enen of nullen ontvangen dan wordt de DM waarschijnlijk overstuurd. Het regelen van de stapgrootte kan gebeuren door naar enkele uitgangsbits te kijken lit.(1), of naar vele bv. 100 of meer lit.(6). Dit laatste noemt men DCDM. (Digital Controlled Delta Modulation). Hier zal verder ingegaan worden op het systeem, dat naar enkele voorgaande uitgangsbits kijkt. Van de verschillende varianten die hiervan bestaan is het systeem gekozen dat na elk bit de stapgrootte aanpast afhankelijk van de vier voorgaande bits. Dit systeem wordt verder met ADM aangeduid en is door Schmitz uitgebreid behandeld. Daarom zal er hier niet te diep op worden ingegaan.

Bij een ADM treden enkele verschijnselen op die bij een DM niet voorkomen of niet opgemerkt worden. Dit zijn convergentie, overshoot, stabiliteit en edgebusiness. Als de oversturing voorbij is moet de ADM weer naar zijn oorspronkelijke stapgrootte terugkeren. Dit wordt convergeren genoemd. Gebeurt dit langzaam dan is de convergentie slecht en als hij niet convergeert dan is de ADM instabiel geworden. Verder kan bij een ADM overshoot optreden. De gesimuleerde ADM is daarom voorzien van een eenvoudige corrector die deze overshoot onderdrukt. Door het adaptief maken wordt ook de edgebusiness hinderlijk. Edgebusiness ontstaat door het niet synchroon zijn van het sample tijdstip met het optreden van een flank in het signaal. Op het scherm ziet men dat stakke contouren rafelig worden weergegeven. Bij lineaire DM wordt edgebusiness door de oversturing onderdrukt. Bij ADM kan het erg hinderlijk zijn. De remedies zijn:

verhoging samplefrequentie

minder sterke stapvergroting

meerdimensionale prediktie

Dit laatste wordt in het hoofdstuk metingen ADM nader toegelicht.

Het regelmechanisme werkt als volgt: De laatste vier uitgangsbits D_k t/m D_{k-3} zijn in een geheugen opgeslagen. De stapgrootte S_{k+1} wordt $P \times S_k$ waarbij P een faktor is die bepaald wordt

door D_k t/m D_{k-3} . Welke P bij welke combinatie van D_k t/m D_{k-3} hoort is vastgelegd in het bewerkingsschema. Door deze P groter of kleiner dan 1 te kiezen wordt stapvergroting of verkleining verkregen. Uit de analyse van twee bits ADM systemen blijkt dat er gebieden zijn aan te geven waarbinnen de factoren P_1 en P_0 moeten liggen die voor stapvergroting resp. stapverkleining zorgen zo dat de deltamodulator convergeert. Nu is het niet mogelijk om P_1 en P_0 zo te kiezen dat convergentiesnelheid en flankresponsie optimaal zijn. Daarom is men van twee naar vier bits systemen overgegaan. De factoren zijn nu P_0 t/m P_3 . Deze zijn zo gekozen dat tijdens oversturing P_2 en P_3 en voor convergentie P_1 en P_0 worden gebruikt. Nu kunnen door de keuze van P_0 t/m P_3 convergentiesnelheid en flankresponsie wel optimaal zijn.

Gesimuleerd is de ADM van Schmitz met het volgende bewerkingsschema:

$$S_{k+1} = \begin{cases} 0.57 \times S_k & \text{als } D_k \neq D_{k-1} \\ S_k & \text{als } D_k = D_{k-1} \neq D_{k-2} \\ 2 \times S_k & \text{als } D_k = D_{k-1} = D_{k-2} \neq D_{k-3} \\ 4 \times S_k & \text{als } D_k = D_{k-1} = D_{k-2} = D_{k-3} \end{cases}$$

Om te voorkomen dat de ADM een slechte flankresponsie door te kleine stapgrootte zou krijgen is een minimale stapgrootte ingebouwd.

4.2 ADPCM

Een methode om de quantizer adaptief te maken gaat ervan uit dat er zoveel correlatie tussen de amplitudes van de samples van hetingangssignaal bestaat dat het zinnig is om de quantizer aan de amplitude van dit signaal aan te passen. Als maat voor de amplitude wordt het niveau van het vorige sample genomen. Door aan ieder niveau een factor te koppelen waarmee drempels en niveaus worden vermenigvuldigd is de quantizer adaptief gemaakt. Het streven is dat door een juiste keuze van deze factoren de quantizer steeds zo wordt bijgeregeld dat de dimensionering van deze quantizer onafhankelijk wordt van de statistiek van hetingangssignaal. Dit zou in een betere quantisatie van niet stationaire signalen moeten resulteren. Uit de literatuur zijn enige adaptieve quantizers bekend. Hier van zullen de systemen van Mitra lit. (7) en van Goldstein en Liu lit. (8) worden besproken. Het door Goodman en Gerscho lit. (9) voorgestelde systeem komt overeen met dat van Mitra. Waar Mitra de quantizer comprimeert en expandeert gebruiken Goodman en Gerscho een vaste quantizer en comprimeren en expanderen hetingangssignaal van deze quantizer. Het belangrijkste verschil is dat het gebied waarbinnen de quantizer kan variëren bij Mitra begrensd is en bij Goodman en Gerscho niet.

zowel Mitra als Goldstein en Liu vermenigvuldigen drempels en niveaus met een factor $\Delta(n)$.

Voor $\Delta(n)$ geldt:

$$0 < KMIN \leq \Delta(n) \leq MMAX > 1$$

Goldstein en Liu kozen voor het volgende systeem:

$$\Delta(n+1) = \begin{cases} MMAX & \text{als } \gamma \Delta(n) > MMAX \text{ en } |E_n| > \text{grens} \\ \gamma \Delta(n) & \text{als } \gamma \Delta(n) \leq MMAX \text{ en } |E_n| > \text{grens} \\ \frac{\Delta(n)}{\gamma} & \text{als } \frac{\Delta(n)}{\gamma} \geq KMIN \text{ en } |E_n| < \text{grens} \\ KMIN & \text{als } \frac{\Delta(n)}{\gamma} < KMIN \text{ en } |E_n| < \text{grens} \end{cases}$$

Grens is een getal zodanig dat steeds de helft van de uitgangsniveaus boven en de andere helft onder grens ligt. Er wordt een uniforme quantizer gebruikt. E_n is hetingangssignaal van de quantizer. De factor γ bepaald de mate van adaptie van de quantizer. KMIN en MMAX zijn de grenzen van het regelgebied.

Mitra realiseerde een iets ander systeem.

$$\Delta(n+1) = \begin{cases} MMAX & \text{als } MMAX < M_i \Delta(n) \\ M_i \Delta(n) & \text{als } KMIN \leq M_i \Delta(n) \leq MMAX \\ KMIN & \text{als } KMIN > M_i \Delta(n) \end{cases}$$

waarbij $M_i = \gamma^{m_i}$ met $i=1,2, \dots, N$

en $m_1 \leq m_2 \dots \leq m_N$, $0 > m_1$ en $m_N > 0$

Bij ieder uitgangsniveau van de quantizer behoort een factor m_i . De keuze van deze m_i 's en γ bepaalt het regelsysteem. In Mitra's ADPCM worden uniforme en niet uniforme quantizers gebruikt.

De ADPCM van Goldstein en Liu is een bijzondere uitvoering van Mitra's systeem. Gebruikt men bij Mitra een uniforme quantizer en kiest men $-1, -1, \dots, -1, 1, \dots, 1$ voor de m_i 's dan zijn beide systemen identiek.

Beide systemen zijn saturating adaptive quantizers. Er geldt nl: $0 < KMIN \leq \Delta(n) \leq MMAX > 1$. Zou $\Delta(n)$ niet begrensd zijn dan kan de stapresponsie van de quantizer slecht worden en zou door afronding $\Delta(n)$ 0 kunnen worden. Dan werkt het regelmechanisme niet meer. Tussen $KMIN, MMAX$ en de grootste en kleinste niveaus bestaan verbanden nl.:

$$\eta_0 \Delta(n) \geq \alpha \quad \text{voor } \Delta(n) \text{ minimaal} \quad (1)$$

$$\eta_{\max} \Delta(n) \leq \beta \quad \text{voor } \Delta(n) \text{ maximaal} \quad (2)$$

η_0 = kleinste uitgangsniveau

η_{\max} = grootste uitgangsniveau

α = kleinste ingangssignaal $\neq 0$

β = grootste ingangssignaal

$\Delta(n)$ minimaal = $KMIN$

$\Delta(n)$ maximaal = $MMAX$

$$(1) \implies KMIN \geq \frac{\alpha}{\eta_0} \quad \text{en} \quad (2) \implies MMAX \leq \frac{\beta}{\eta_{\max}}$$

Mitra en Goldstein en Liu gebruikten onafhankelijke uniform verdeelde ruis als ingangssignaal voor hun ADPCM's. Voor ruis

met een uniforme, gaussische of laplace verdeling berekenen Goldstein en Liu de verdeling van $\Delta(n)$ en van het ingangssignaal van de quantizer. Bij computer simulaties van hun ADPCM systemen blijkt vooral de gemeten verdelingsfunctie van het ingangssignaal prachtig met de theoretisch voorspelde curve overeen te stemmen. De S/N metingen die zij verricht hebben zijn hier van minder belang. Zij werken nl. met een variabele samplefrequentie die varieert van 3 tot 50 x de bandbreedte van het ingangssignaal van het ADPCM systeem. Daar simulatie van deze variabele bandbreedte niet praktisch uitvoerbaar is op de beschikbare apparatuur, zie hoofdstuk coderen video-beelden, heeft het niet veel zin daar verder op in te gaan. Mitra gebruikt bij zijn simulaties gaussische ruis als ingangssignaal en kijkt dan naar de S/N verhouding en de stapverdeling. Verder bekijkt hij de stapresponsie van zijn ADPCM als functie van ν . Mitra vergelijkt alleen de stapverdeling van het adaptieve en niet adaptieve geval. Er blijkt uit dat bij adaptieve systemen de verdelingsfunctie vlakker verloopt dan bij niet adaptieve systemen hetgeen voor noiseless source coding van het uitgangssignaal een nadeel is. Verder worden door hem geen ADPCM en DPCM systemen met elkaar vergeleken. Goldstein en Liu geven geen directe vergelijkingen van ADPCM met DPCM.

Het regelmechanisme van Mitra is verfijnder dan dat van Goldstein en Liu. Verder is het systeem van Goldstein en Liu uit een iets ander gezichtspunt ontwikkeld dan dat van Mitra. Zij stellen: gegeven een x bits adaptieve quantizer. Hoeveel kan dan de samplefrequentie dalen om dezelfde S/N verhouding te verkrijgen als met een niet adaptieve quantizer en hoe is dat voor verschillende factoren x ? De situatie op de simulatie apparatuur is precies andersom. Gegeven een bepaalde samplefrequentie, met een hoeveel bits adaptieve quantizer wordt een acceptabel beeld verkregen? In dat laatste geval zal een verfijnder regelmechanisme zoals dat van Mitra waarschijnlijk beter voldoen dan het systeem van Goldstein en Liu. Daarom is een systeem volgens Mitra gesimuleerd met opeenvolgende gehele getallen voor de m_i 's. Bv. -2, -1, 0, 1, 2 voor een quantizer met 10 niveaus. De programma's zijn overigens zo geschreven dat verschillende systemen gesimuleerd kunnen worden.

Als men ADPCM systemen gaat gebruiken om videobeelden te coderen dan is niet meer aan alle voorwaarden voldaan die door Mitra en Goldstein en Liu gesteld zijn. Doordat videobeelden een meerdimensionale correlatiefunctie bezitten zal de predictor uitgebreid moeten worden. Verder is het door Mitra en Goldstein en Liu gebruikte Mean Square Error foutcriterium niet geschikt en moet er een uit de perceptie leer afkomstig fout-

criterium worden gebruikt. De verschillende dimensioneringsproblemen die bij het ontwerp van een dergelijk systeem ontstaan lijken dan het meest efficiënt experimenteel opgelost te kunnen worden.

5.1 ADPCM met adaptieve predictie en quantisatie

Een niet adaptieve predictor vermenigvuldigt voorgaande samples met vaste waarden. Deze waarden zijn bepaald uit de correlatiecoëfficiënten van het gehele beeld. In de praktijk is dit slechts te realiseren door van een of meerdere beelden waarvan men weet dat ze representatief zijn voor de beelden waarvoor het systeem gaat worden gebruikt de correlatiefunctie te meten. Lokaal kunnen in een beeld aanzienlijke afwijkingen in deze correlaties optreden. Een predictor zou met deze afwijkingen rekening moeten houden. Stel de veel voorkomende situatie van een zwart-wit overgang in x of y richting. Een normale tweedimensionale predictor zal een grijsniveau voorspellen dat ver van de werkelijke waarde aflight. Een predictor die met deze overgang rekening houdt zal een betere voorspelling kunnen maken nl. alleen het x of y sample hetgeen in deze overgang het enig juiste is. Een en ander laat zich uitbreiden naar drie dimensies. Door de betere predictie zal de verdelingsfunctie van het ingangssignaal van de quantizer een kleinere spreiding bezitten. Deze quantizer kan dan met minder niveaus werken bij eenzelfde mate van vervorming. Bij de nu volgende bespreking van de adaptieve predictor zullen quantisatiefouten in eerste instantie worden verwaarloosd.

De tweedimensionale predictor. Gegeven de pels A t/m A_4 .

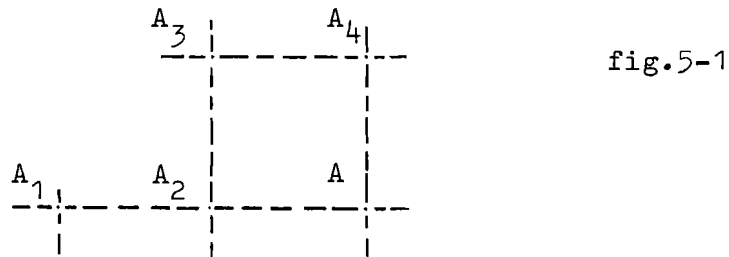


Fig.5-1 geeft aan hoe men zich de ligging van deze pels ergens in het beeld moet voorstellen. A moet worden geschat.

noem: $|A_2 - A_1| = V_1$ en $|A_2 - A_3| = V_2$

V_1 en V_2 vormen een maat voor het te verwachten succes van de predictie in resp. x en y richting.

nu kies voor \tilde{A} dit is de schatting voor A :

$$\tilde{A} = \frac{A_2 V_2 + A_4 V_1}{V_1 + V_2} \quad (3)$$

en $\tilde{A} = \frac{A_2 + A_4}{2} \quad (4)$ als $V_1 = V_2 = 0$

vb. Stel $A_3 = A_4 = a+b$ en $A_1 = A_2 = a$ dan (3) $\tilde{A} = a$ en $\tilde{A} \neq \tilde{A}(b)$.

Uit dit voorbeeld volgt dat sprongen veroorzaakt door horizontaal verlopende contouren correct worden voorspeld.

Bij verticaal verlopende contouren wordt er 1 beeldpunt fout geschat hoewel de predictie beter is dan eendimensionale predictie en niet slechter dan tweedimensionale niet adaptieve predictie. Het volgende voorbeeld maakt dit duidelijk.

vb. Gegeven: zie fig.5-2 A_1 t/m $A_3 = a$ en A_4 t/m $A_6 = a+b$

Nu doet zich de situatie voor dat $V_1 = V_2 = 0$. D.w.z. verwacht wordt dat zowel de x als de y richting de juiste

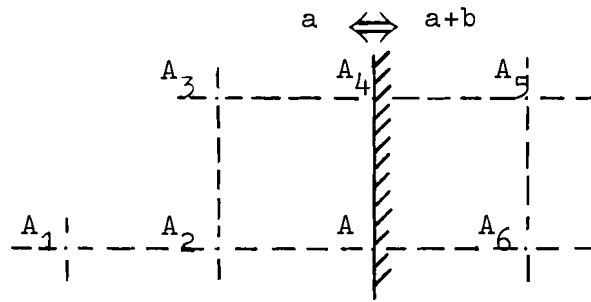


fig.5-2

schatting leveren. Nu

$$(4) \quad \tilde{A} = \frac{A_4 + A_2}{2} = \frac{a+b+a}{2} = a + \frac{b}{2} \neq a+b \text{ en}$$

$$(3) \quad \tilde{A}_6 = \frac{(a+b)b}{b} = a+b \text{ hetgeen correct is. Eendimensionale}$$

predictie zou $\tilde{A}=a$ geven.

Bij beelden waarbij de intensiteit slechts een functie is van de x of y richting lvert deze predictor altijd een juiste schatting. Voor een driedimensionaal systeem wordt voor \tilde{A}

de volgende uitdrukking gekozen:

$$\tilde{A} = \frac{A_2 V_2 V_3 + A_6 V_1 V_3 + A_5 V_1 V_2}{V_1 V_2 + V_2 V_3 + V_1 V_3} \quad (5)$$

$$\text{en } \tilde{A} = \frac{A_2 + A_5 + A_6}{3} \quad (6) \text{ als } V_1=V_2=V_3=0$$

Fig.5-3 geeft de ligging van de pels aan en

$$V_1 = |A_1 - A_2| \quad V_2 = |A_2 - A_3| \quad V_3 = |A_4 - A_2|$$

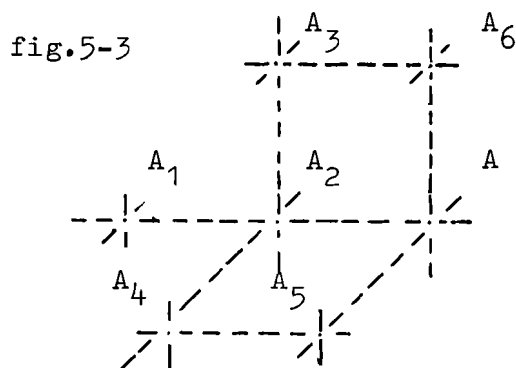
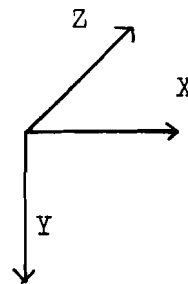


fig.5-3



vb. Stel A_1 t/m $A_4=a$ en A_5 t/m $A_6=a+b=A$

$$\text{dan (6)} \quad \tilde{A} = \frac{A_2 + A_5 + A_6}{3} = \frac{a+a+b+a+b}{3} = a + \frac{2}{3}b \text{ en dit is be-}$$

ter dan bij de tweedimensionale predictor die $a + \frac{b}{2}$ levert.

Overigens levert dit driedimensionale systeem in alle overige situaties bij stilstaande beelden een correcte schatting.

Voor systemen met een 3 of 4 bit (adaptieve) quantizer lijkt het weinig zinvol om driedimensionale predictie toe te passen met het doel een betere weergave te krijgen. De quantizer is dan meestal fijn genoeg om het verschil tussen predictie en werkelijke signaal zo te quantiseren dat de fout voor het oog niet waarneembaar is. In deze situaties heeft driedimensionale predictie alleen zin om de verdelingsfunctie van het uitgangssignaal te beïnvloeden en dit meer geschikt te maken voor verdere noiseless source coding. Door een betere predictie is nl. een kleinere spreiding van hetingangssignaal te verwachten en daardoor een frequenter gebruik van de onderste niveaus. Bij het gebruik van quantizers met minder dan 8 niveaus kan het noodzakelijk zijn om driedimensionale predictie toe te passen omdat de quantizer dan niet altijd meer in staat is het verschilsignaal zo te quantiseren dat de fout voor het oog niet waarneembaar is. Dit zal later in dit hoofdstuk nog nader worden toegelicht.

5.2 Voorstel voor een ander type adaptieve quantizer

De factoren V_1 en V_2 die worden gebruikt om de kwaliteit van de schatting aan te geven kunnen ook worden gebruikt voor het adaptief maken van de quantizer. Motivatie: Stel dat V_1 en V_2 groot zijn. Dan was bij het voorgaande sample de schatting in x en y richting weinig succesvol en trad een grote afwijking op. Om deze grote afwijking goed te kunnen quantiseren moet de quantiser een groot bereik hebben. Is de schatting nauwkeurig dan zal een fijne quantizer het best voldoen. De predictor kan dus een regelsignaal leveren voor de adaptie van de quantizer. Dit is een andere mogelijkheid om de quantizer adaptief te maken dan de methodes van Mitra, Goldstein en Liu. T.o.v. deze methodes heeft het nieuwe voorstel het voordeel dat het momentaan werkt.

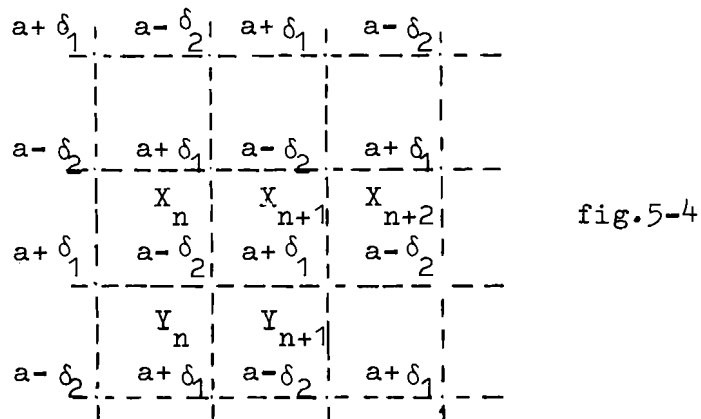
Het systeem werkt als volgt. Voor ieder beeldpunt wordt een quantizer berekend door een vooraf ingestelde quantizer met een factor $\Delta(n) \geq 1$ te vermenigvuldigen. Deze factor wordt als volgt bepaald: als $V_1 < V_2$ dan $V := V_1$ anders $V := V_2$ en

$$\Delta(n) := 1 + qV \quad (7)$$

waarbij q een instelbare adaptiefactor is die de mate van adaptie bepaald.

5.3 De invloed van de kwantisatie op de predictie

In het voorgaande is er steeds vanuit gegaan dat de fouten die door kwantisatie worden gemaakt op de predictie weinig invloed hebben. Voor fijne quantizers is dit een redelijke veronderstelling. Bij de analyse van 2 en 3 bit systemen is dit niet meer het geval. Daarom volgt nu een analyse van het systeemgedrag in verschillende situaties met inachtneming van kwantisatie. Hiervoor wordt een gebied van constante intensiteit en drie soorten sprongen in de intensiteit genomen. Stel constante intensiteit a . Dit wordt door de ADPCM als volgt weergegeven: zie fig.5-4



notaties: X_n = de waarde die het beeldpunt zou moeten hebben
 \tilde{X}_n = de schatting van X_n
 \bar{X}_n = de waarde die uiteindelijk aan X_n wordt toegerekend.

$$(3) \Rightarrow \tilde{X}_n = \frac{(\delta_1 + \delta_2)(a - \delta_2) + (\delta_1 + \delta_2)(a - \delta_2)}{(\delta_1 + \delta_2) + (\delta_1 + \delta_2)} = a - \delta_2 \quad X_n = a$$

Stel dat de quantizer het kleinste uitgangsniveau gebruikt

en noem dit δ_1 dan

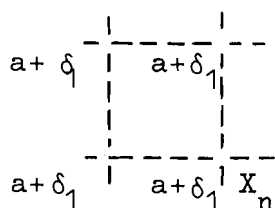
$$(7) \Rightarrow \bar{X}_n = (a - \delta_2) + \delta_1(1 + q(\delta_1 + \delta_2)) = a + \delta_1 \quad (8)$$

$$(8) \Rightarrow \delta_1 + \delta_2 = \eta_1(1 + q(\delta_1 + \delta_2)) \quad \text{met } \delta_1 + \delta_2 = \delta \quad \text{dan:}$$

$$\delta = \eta_1(1 + q\delta) \quad (9) \quad \text{en} \quad \delta = \frac{\eta_1}{1 - \eta_1 q} \quad (10)$$

Gegeven η_1 en q kan de quantisatiefout δ in vlakke gebieden worden bepaald. In fig. 5-5 is δ als functie van q met η_1 als parameter uitgezet. Uit deze figuur blijkt dat door het adaptief maken de korrelruis in vlakke gebieden is vergroot t.o.v. een niet adaptief systeem met eenzelfde kleinste niveau η_1 .

Dat het voor deze afleiding gebruikte patroon een reële aanname is blijkt niet alleen uit de metingen maar kan ook worden verklaard. Daar het vlak gemiddeld een constante intensiteit moet bezitten zal bij verschuiving over p roosterpunten in de x en q in de y richting dezelfde configuratie optreden. Wat zijn de minimale p en q die hieraan voldoen? $p=q=1$ gaat niet want

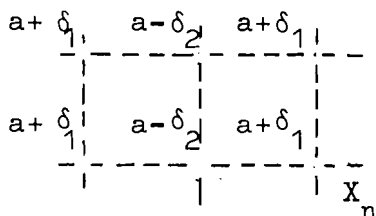


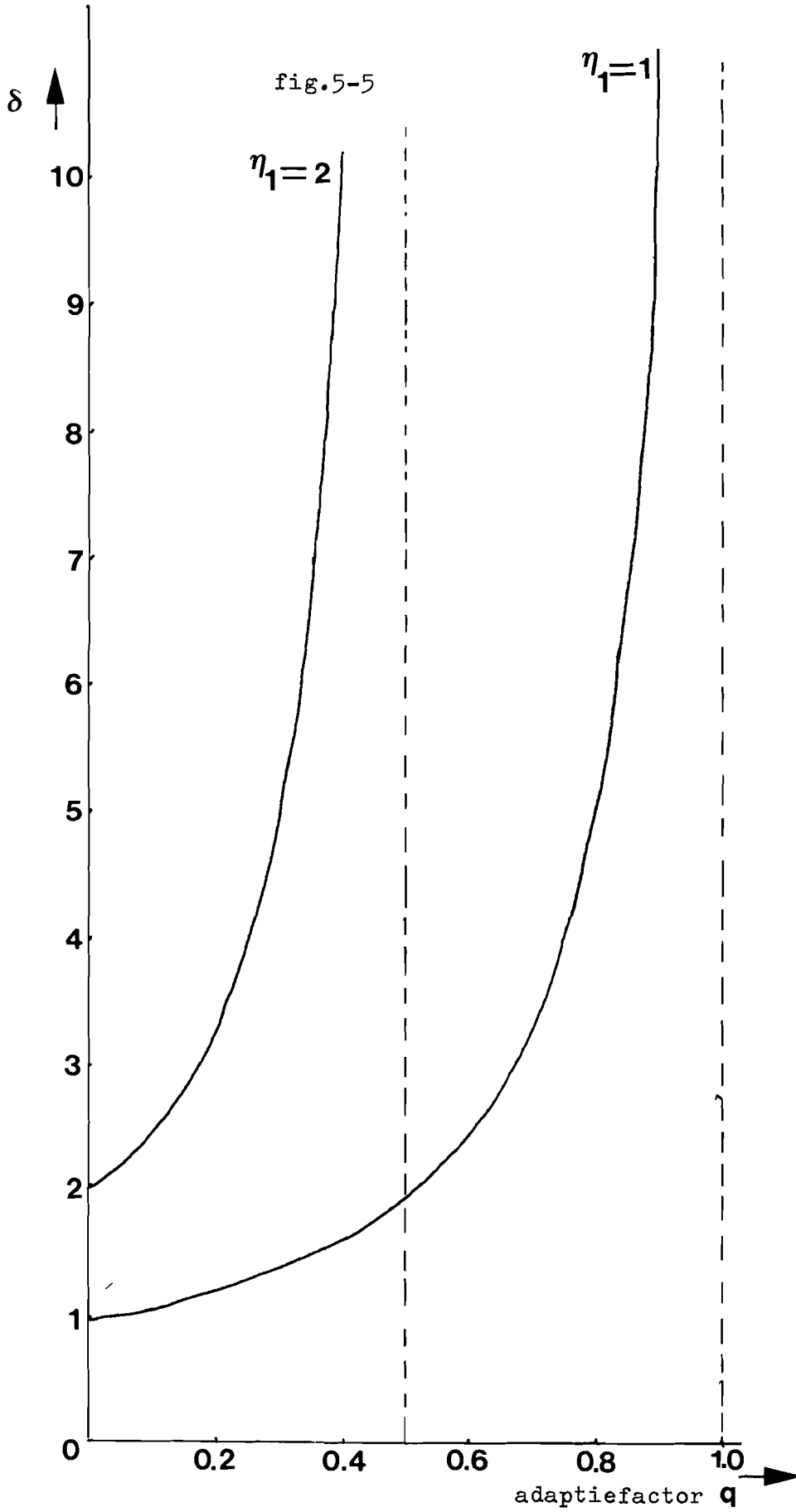
$$(3) \quad \tilde{X}_n = a + \delta_1$$

$$\text{en } \bar{X}_n = a + \delta_1 - \eta_1(1 + q\delta)$$

$$= a + \delta_1 - \eta_1 \neq a - \delta_2$$

probeer dan $p=2$ en $q=1$





$$(3) \quad \tilde{X}_n = a + \delta_1 \quad \bar{X}_n = a + \delta_1 - \eta_1 \neq a + \delta_1$$

dus $p=2, q=1$ voldoet niet. Evenzo $p=1, q=2$ en de eerstvolgende combinatie is $p=q=2$ met het eerder beschreven patroon.
sprongresponsie

a) sprong in x richting

Het te verwachten patroon als responsie op een sprong van a naar $a+b$ is

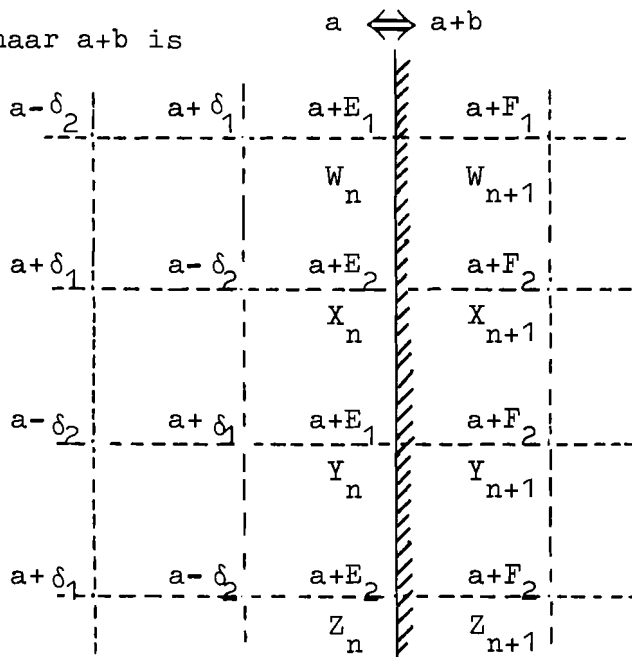


fig.5-6

$$(3) \quad \tilde{X}_n = \frac{(\delta_1 + \delta_2)(a + E_1) + (a - \delta_2)(\delta_1 + \delta_2)}{2(\delta_1 + \delta_2)} = a + \frac{E_1 - \delta_2}{2}$$

$$(7) \quad \bar{X}_n = a + \frac{E_1 - \delta_2}{2} + \eta_m(1 + q\delta) \text{ met } \delta = \delta_1 + \delta_2 \text{ en } \delta_m \text{ het maximale}$$

uitgangsniveau van de quantizer.

$$\text{idem } \tilde{Y}_n = a + \frac{E_2 + \delta_1}{2} \text{ en } \bar{Y}_n = a + \frac{E_2 + \delta_1}{2} + \eta_m(1 + q\delta) = a + E_1$$

In appendix A wordt afgeleid dat voor E_1 en E_2 geldt

$$E_1 = \frac{2\delta_1 - \delta_2}{3} + 2\eta_m(1 + q\delta) \text{ of met (9) } E_1 = \frac{2\delta_1 - \delta_2}{3} + 2\frac{\eta_m}{\eta_1} \quad (11)$$

$$\text{en } E_2 = \frac{\delta_1 - 2\delta_2}{3} + 2\eta_m(1 + q\delta) \text{ of met (9) } E_2 = \frac{\delta_1 - 2\delta_2}{3} + 2\frac{\eta_m}{\eta_1} \quad (12)$$

Voor F_1 en F_2 geldt (zie appendix A)

$$F_1 = \frac{E_2 P_1 + E_1 (P_2 + E)}{P_1 + P_2 + E} + \eta_i (1 + qE) \frac{(2P_1 + E)(P_2 + E)}{E(P_1 + P_2 + E)} \quad (13)$$

$$F_2 = \frac{E_1 P_2 + E_2 (P_1 + E)}{P_1 + P_2 + E} + \eta_i (1 + qE) \frac{(2P_2 + E)(P_1 + E)}{E(P_1 + P_2 + E)} \quad (14)$$

Als $\eta_m \gg \eta_1$ kan men stellen dat $F_1 = F_2 = F$ en dat (zie appendix A)

$$F \approx 2\eta_m (1 + q\delta) + \eta_i (1 + qE) \frac{6\eta_m}{\eta_1}$$

Men ziet dat voor bv. $\eta_m = 6, \eta_1 = 2$ en $q = 0.27$ het laagste niveau van de aangepaste quantizer al op ≈ 50 komt te liggen. Hier heeft de waarde van q niet veel invloed op daar E klein is. Als men bovenstaande getallen in (11) en (12) invult dan

$$E_1 \approx 24 \quad \text{en} \quad E_2 \approx 24$$

Dwz. voor b (=de sprongamplitude) groot treedt er een beeldpunt vertraging in de sprong op.

b) sprong in x en y richting ter grootte b . Fig 5-7 geeft het patroon dat men als responsie op een sprong van a naar $a+b$ kan verwachten

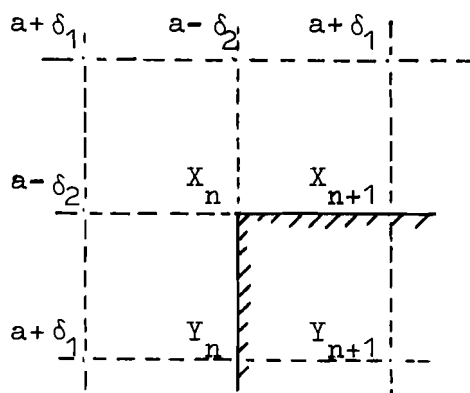


fig.5-7

$$(3) \Rightarrow \tilde{X}_n = a - \delta_2 \quad (7) \Rightarrow \bar{X}_n = a - \delta_2 + \eta_m (1 + q\delta)$$

$$\text{idem: } \bar{Y}_n = a + \frac{\delta_1 - \delta_2}{2} + \frac{3}{2} \eta_m (1 + q\delta)$$

$$\text{en } \bar{X}_{n+1} = a + \frac{\delta_1 - \delta_2}{2} + \frac{1}{2} \eta_m (1 + q\delta) + \eta_i (1 + q\eta_m (1 + q\delta)).$$

men ziet dat met name de rij $\bar{X}_n, \bar{X}_{n+1}, \dots$ snel stijgt terwijl het in de y richting iets minder vlug gaat.

c) een sprong volgens de contour $x=y$ met amplitude b

Fig.5-8 en 5-9 geven de responsies weer.

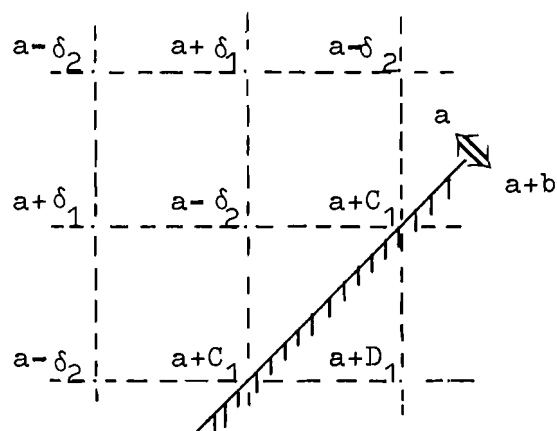


fig.5-8

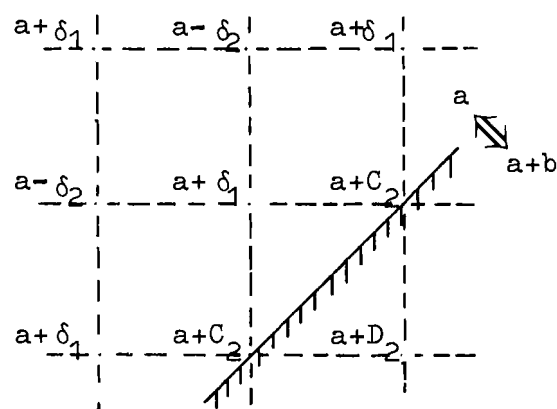


fig.5-9

m.b.v.(3) en (7) kan men afleiden dat

$$C_1 = \delta_1 + \eta_m (1+q \delta) \quad \text{en} \quad C_2 = -\delta_2 + \eta_m (1+q \delta)$$

$$D_1 = \delta_1 + \eta_m (1+q \delta) + \eta_i (1+q(\delta + \eta_m (1+q \delta)))$$

$$D_2 = -\delta_2 + \eta_m (1+q \delta) + \eta_i (1+q(-\delta + \eta_m (1+q \delta)))$$

Hieruit volgt dat $C_1 - C_2 = \delta$ en $D_1 - D_2 = \delta(1+2\eta_i q)$

Dit zijn de verschillen die tussen de beelden kunnen optreden.

Voor een sprong in de x richting is dit

$$(11)-(12) = E_1 - E_2 = \frac{\delta}{3} \quad \text{en}$$

$$(13)-(14) = F_1 - F_2 = \frac{1+2\eta_i(1+q\frac{\delta}{3})}{1+12\eta_m/\eta_1} \quad \text{zie appendix B}$$

Uit deze berekeningen volgt dat vooral bij schuine contouren flikkeringen zijn te verwachten bij een tweedimensionaal systeem. Men ziet verder uit deze situaties dat het systeem vooral bij sprongen in x en y richting en volgens schuine contouren een te kleine waarde geeft voor de amplitude van de sprong. Dit betekent dat bij dit soort overgangen een veegeffect zal optreden. Om dit effectief te onderdrukken moet men voor quantizers met een gering aantal niveaus te grote waarden van q en η_m gebruiken waardoor de korrelruis bij x of y overgangen en in de vlakke gebieden te groot wordt.

5.4 De invloed van een drempel in de adaptie

Een remedie hiertegen is het inbouwen van een drempel in het regelmechanisme voor de adaptie. Hoe moet deze worden gekozen

opdat er in gebieden met constante intensiteit geen hinderlijke korrelruis ontstaat. Stel weer fig. 5-4 als responsie.

nu $\tilde{X}_n = a - \delta_2$ en

$$\bar{X}_n = (a - \delta_2) + \eta_1 (1 - \alpha + q(\delta_1 + \delta_2)) = a + \delta_1 \text{ met } \alpha \text{ de drempel.}$$

Dan $\delta = \eta_1 (1 - \alpha + q\delta)$. Voor een volledige eliminatie van het effect zou moeten gelden $\delta = \eta_1 \Rightarrow \alpha = \eta_1 q$. In vlakke gebieden is dan het regelmechanisme uitgeschakeld en wordt met de kleinste stap gewerkt. Wat voor invloed heeft deze drempel echter op de stapresponsie? Neem weer stap ter grootte b in de x richting.

Zie fig. 5-6 voor de responsie.

$$\text{Nu } E_1 = \frac{2\delta_1 - \delta_2}{3} + 2\eta_m \text{ en } E_2 = \frac{\delta_1 - 2\delta_2}{3} + 2\eta_m$$

Evenals in het geval zonder drempel kan men F_1 en F_2 bepalen

$$F_1 = K + C_1 + \eta_i \left(1 - \eta_1 q + \frac{\eta_1}{3} q\right) A_1 \text{ en } F_2 = K + C_2 + \eta_i \left(1 - \eta_1 q + \frac{\eta_1}{3} q\right) A_2$$

daar $1 - \eta_1 q + \frac{\eta_1}{3} q < 1$ wordt deze term door het regelmechanisme gelijk aan 1 gemaakt.

$$\text{Dan: } F_1 = K + C_1 + \eta_i A_1 \text{ en } F_2 = K + C_2 + \eta_i A_2$$

Als men deze uitdrukkingen met (13) en (14) vergelijkt dan valt het op dat er een kleinere waarde voor de sprongamplitude wordt berekend. Vooral voor een sprong in de x en y richting en langs een contour $x=y$ zal dit effect het veeffect doen toenemen en een grotere q noodzakelijk maken. Dit is experimenteel bevestigd. Een kleine drempel die de invloed van adaptie in vlakke gebieden niet volledig uitschakeld kan de prestatie van het systeem verbeteren. Bij een volledige com-

pensatie wordt de verslechterde flankresponsie zo storend dat het systeem een minder goede beeldkwaliteit geeft dan een systeem zonder drempel. De flankresponsie wordt nl. door twee elkaar versterkende effecten nadelig beïnvloed.

- 1) i.p.v. $\eta_i(1-qV)$ nu $\eta_i(1+q(V-\eta_1))$
- 2) doordat kleiner wordt zal V , dit is het kleinste verschil tussen de beeldpunten dat de adaptie bepaald, kleiner zijn.

5.5 Driedimensionale systemen

Tot nu toe zijn alleen tweedimensionale systemen bekeken. Daarom zullen nu enige eigenschappen van het driedimensionale systeem zonder drempel worden bekeken. Aan de weergave van gebieden met een constante intensiteit verandert driedimensionale predictie niets dus hier zal verder niet op ingegaan worden. Wat wel verandert is de sprongresponsie en daarom zullen de drie sprongen die in het voorgaande voor tweedimensionale predictie zijn bekeken nu voor de driedimensionale situatie worden doorgerekend.

a) sprong ter grootte b in de x richting. Zie fig. 5-10

$$(5) \Rightarrow \tilde{X}_n = \frac{\delta^2(a+E_1) + \delta^2(a-\delta_2) + \delta^2(a+E_1)}{3\delta^2}$$

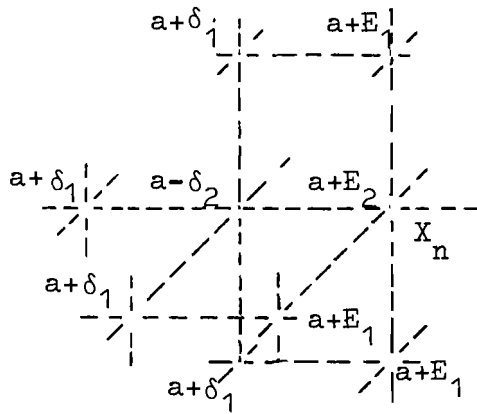
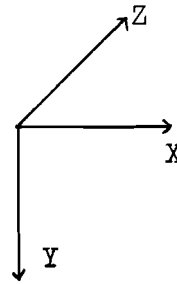


fig.5-10



$$\bar{X}_n = a + \frac{2E_1 - \delta_2}{3} + \eta_m(1 + q\delta) = a + E_2 \quad (15)$$

$$\tilde{Y}_n = \frac{\delta^2(a + E_2) + \delta^2(a + \delta_1) + \delta^2(a + E_2)}{3\delta^2}$$

$$\bar{Y}_n = a + \frac{2E_2 + \delta_1}{3} + \eta_m(1 + q\delta) = a + E_1 \quad (16)$$

Uit (15) en (16) kunnen E_1 en E_2 worden bepaald

$$E_1 = \frac{3\delta_1 - 2\delta_2}{5} + 3\eta_m(1 + q\delta)$$

$$E_2 = \frac{2\delta_1 - 3\delta_2}{5} + 3\eta_m(1 + q\delta)$$

b) sprong met amplitude b in x en y richting. Zie fig.5-11

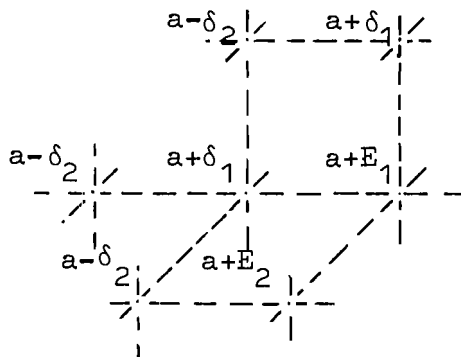
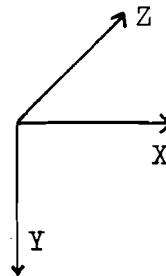


fig.5-11

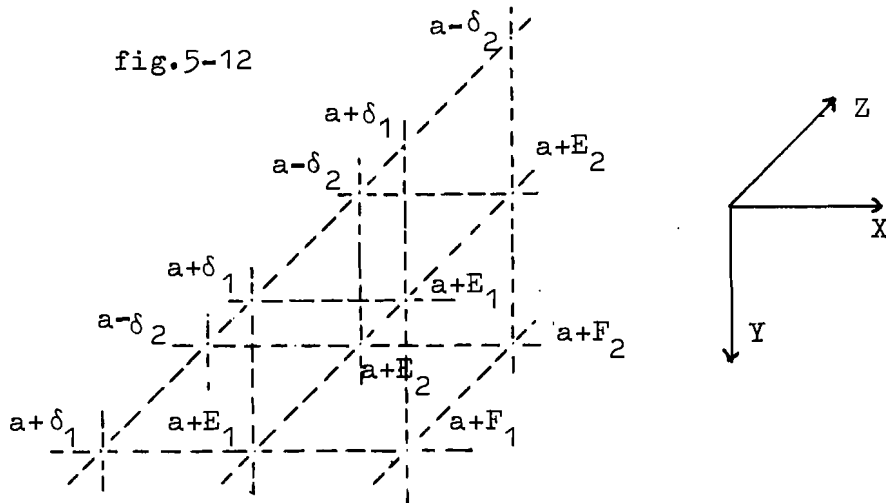


M.b.v.(5) en (7) wordt na soortgelijk rekenwerk als bij a)

verkregen dat:

$$E_1 = \frac{3\delta_1 - \delta_2}{4} + \frac{3}{2}\eta_m(1+q\delta) \quad (17) \quad E_2 = \frac{\delta_1 - 3\delta_2}{4} + \frac{3}{2}\eta_m(1+q\delta) \quad (18)$$

c) sprong met amplitude b volgens contour $x=y$. Zie fig.5-12



Voor E_1 en E_2 gelden (17) en (18). Hiermee wordt dan samen met (5) en (7) de vergelijkingen voor F_1 en F_2 opgesteld.

$$F_1 = \frac{E_2(E_1 - \delta_1) + 2EE_1}{2E + E_1 - \delta_1} + \eta_i(1+q\delta)$$

$$F_2 = \frac{E_1(E_2 + \delta_2) + 2EE_2}{2E + E_2 - \delta_2} + \eta_i(1+q\delta)$$

In zowel a), b) als c) blijkt er bij dezelfde η_m, η_1 en q een verbetering van $\pm 50\%$ in de sprongresponsie op te treden t.o.v. tweedimensionale systemen. Verder is bij a), b) en c) $E_1 - E_2$ gelijk aan resp. $\delta/5, \delta/2$ en $\delta/2$ i.p.v. resp. $\delta/3, \delta$ en δ zoals in het tweedimensionale geval. In situatie c) is $F_1 - F_2 \neq f(q)$ maar slechts een verschil tussen twee vrijwel identieke termen terwijl dit bij het tweedimensionale systeem $\delta(1+2\eta_i q)$ was hetgeen eliminatie van flikkerende con-

touren betekent. Aan de weergave van gebieden met een constante intensiteit verbetert driedimensionale predictie niet direct iets. Wel indirect daar een kleinere factor q kan worden gekozen t.o.v. tweedimensionale predictie. Door het niet lineaire verband tussen q en δ zie (10) en fig. 5-5 kan dit tot een aanzienlijk kleinere δ aanleiding geven.

5.6 Samenvatting

nu is men in staat om de gevolgen van quantisatie voor de predictie te bekijken voor de drie soorten sprongen.

a) sprong in x richting. Zonder quantisatie wordt $a + \frac{1}{2}b$

voorspeld voor \tilde{X}_n, \tilde{Y}_n en \tilde{Z}_n . Met quantisatie wordt dit $a + \frac{\delta_1 - 2\delta_2}{3} + \eta_m(1+q\delta)$ of $a + \frac{2\delta_1 - \delta_2}{3} + \eta_m(1+q\delta)$

en zonder quantisatie $a+b$ voor X_{n+1}, Y_{n+1} en Z_{n+1} en $a + K + C_1 + \eta_i(1+qE)A_1$ of $a + K + C_2 + \eta_i(1+qE)A_2$ met quantisatie.

Met driedimensionale predictie worden \tilde{X}_n, \tilde{Y}_n en \tilde{Z}_n $a + \frac{2}{3}b$ zonder quantisatie. Met quantisatie wordt dit

$a + \frac{2}{15}(3\delta_1 - 7\delta_2) + 2\eta_m(1+q\delta)$ of $a + \frac{2}{15}(7\delta_1 - 3\delta_2) + 2\eta_m(1+q\delta)$.

b) sprong in x en y richting. Zonder quantisatie wordt $a + \frac{1}{2}b$

voorspeld voor \tilde{X}_{n+1} en \tilde{Y}_n . $\tilde{X}_n = a$. Met quantisatie wordt dit $\tilde{X}_{n+1} = a + \frac{\delta_1 - \delta_2}{2} + \frac{1}{2}\eta_m(1+q\delta)$, $\tilde{Y}_n = a + \frac{\delta_1 - \delta_2}{2} + \frac{1}{2}\eta_m(1+q\delta)$ en $\tilde{X}_n = a - \delta_2$

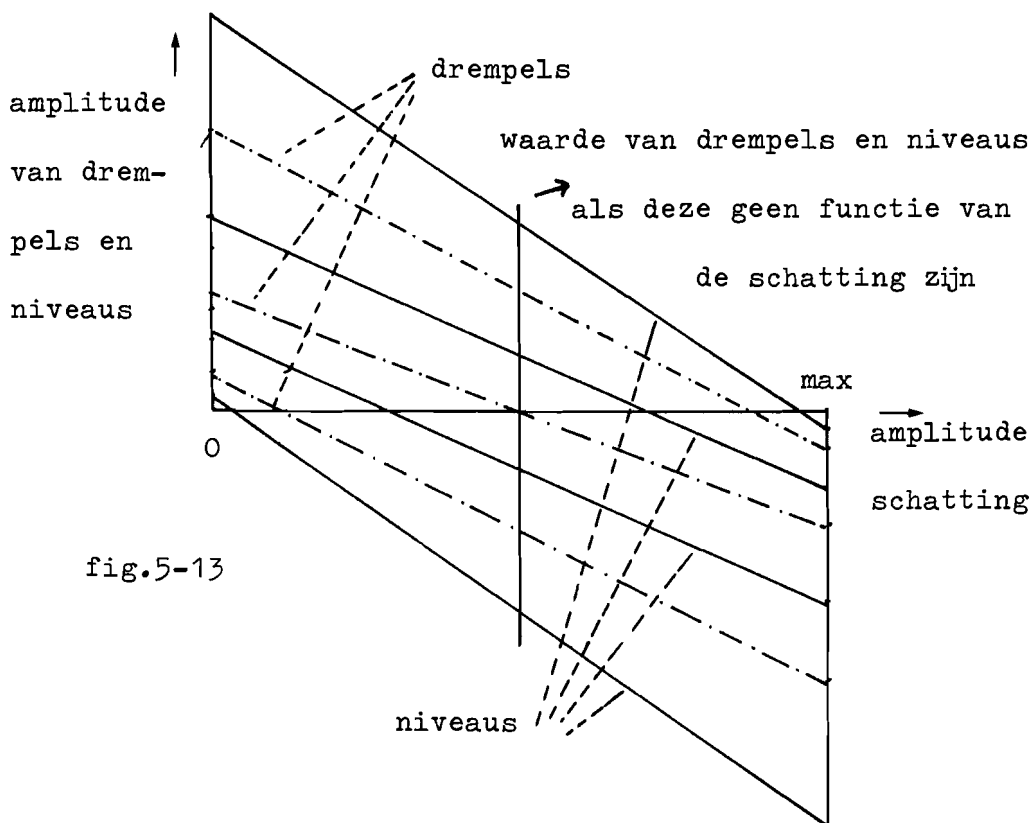
c) sprong langs contour $x=y$. Hier loopt zowel met als zonder

quantisatie de schatting 1 beeldpunt achter bij de werkelijke waarde. Door quantisatie wordt deze fout over meerdere beeldpunten uitgesmeerd afhankelijk van de dimensionering van het systeem. (veegeffect)

5.7 Conclusies

- 1) Driedimensionale predictie is te preferen boven tweedimensionale predictie. Dit geldt ook bij beweging. Zie appendix C waar wordt aangetoond dat het systeem dan overgaat in tweedimensionaal systeem.
- 2) Het invoeren van een drempel zodanig dat in gebieden met een constante intensiteit de quantizer niet adaptief is, levert een aanzienlijke degradatie van de sprongresponsie op. Een kleine drempel die het regelmechanisme niet volledig uitschakelt werkt beter.
- 3) Door quantisatie wordt soms een aanzienlijke afwijking in de predictie verkregen.
- 4) Door voor de predictie meer dan twee verschillen tussen de al ontvangen beeldpunten te gebruiken en ook de sturing van de adaptieve quantizer van meerdere verschillen afhankelijk te maken is waarschijnlijk verdere verbetering haalbaar. Men kan dan bij een lagere bitrate dezelfde fout verkrijgen of bij dezelfde bitrate een kleinere fout .

- 5) Het voorgestelde systeem van adaptieve predictie en adaptieve quantisatie levert bij lage bitrates een betere S/N verhouding en een betere beeldkwaliteit bij lage bitrates dan het systeem van Mitra.
- 6) Bij 3 bit systemen is er visueel geen verschil tussen beide methodes maar levert het voorgestelde systeem een uitgangssignaal dat beter geschikt is voor noiseless source coding. Bij de voorgestelde quantizer is de ligging van de drempels en de niveaus geen functie van de amplitude van de schatting. Hierdoor worden soms niveaus gereserveerd voor situaties die niet kunnen voorkomen. Door deze ligging wel een functie van de schatting te maken wordt de quantisatiefout verkleind. Zie fig.5-13.



Het ADM Programma

Het ADM systeem dat is gesimuleerd komt overeen met dat van Schmitz. Fig. (6-1) geeft het principecircuit. Een van de door Schmitz voorgestelde modificaties is er in aangebracht nl. de overshoot onderdrukking. Gekozen is voor de meest eenvoudige vorm van overshoot onderdrukking waarbij alleen de voorgaande stap wordt gebruikt. Was dit een stap waarvoor de maximale stapvergrotingsfactor is geselecteerd en wordt deze door een stap gevolg die kleiner is dan wordt het uitgangssignaal dat bij de vorige stap behoort gelijk gemaakt aan het niveau dat ontstaan zou zijn bij gebruik van deze kleinere stap. Zie fig. (6-2).

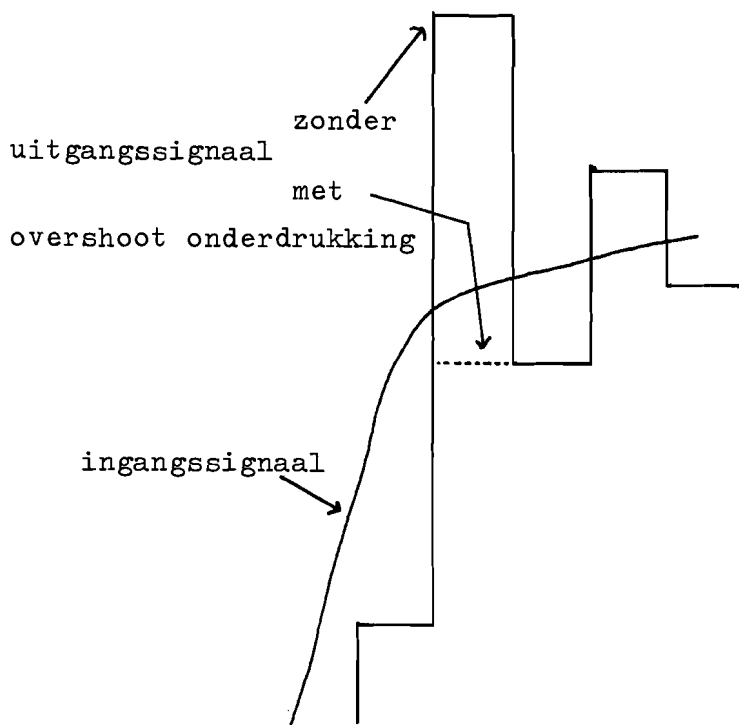


fig.6-2

6.1 Het programma

Het ADM programma bestaat slechts uit enkele regels en doet niets anders dan het aanroepen van een subroutine die de simulatie uitvoert. Deze subroutine is universeel bruikbaar en kan in ieder programma via een globale naam aangeroepen worden. Het hoe en waarom van globale namen wordt in lit.(10) uitgelegd. De flowchart (bijlage A) geeft schematisch aan hoe de subroutine is opgebouwd. Hij bestaat uit twee hoofddelen. Het eerste verzorgt de initialisatie, het tweede de simulatie. Het simulatie gedeelte levert van een aantal rijen samples naar keuze:

- a) het uitgangssignaal type:=0
- b) de stapgrootte type:=1
- c) de gesimuleerde ontvanger type:=2

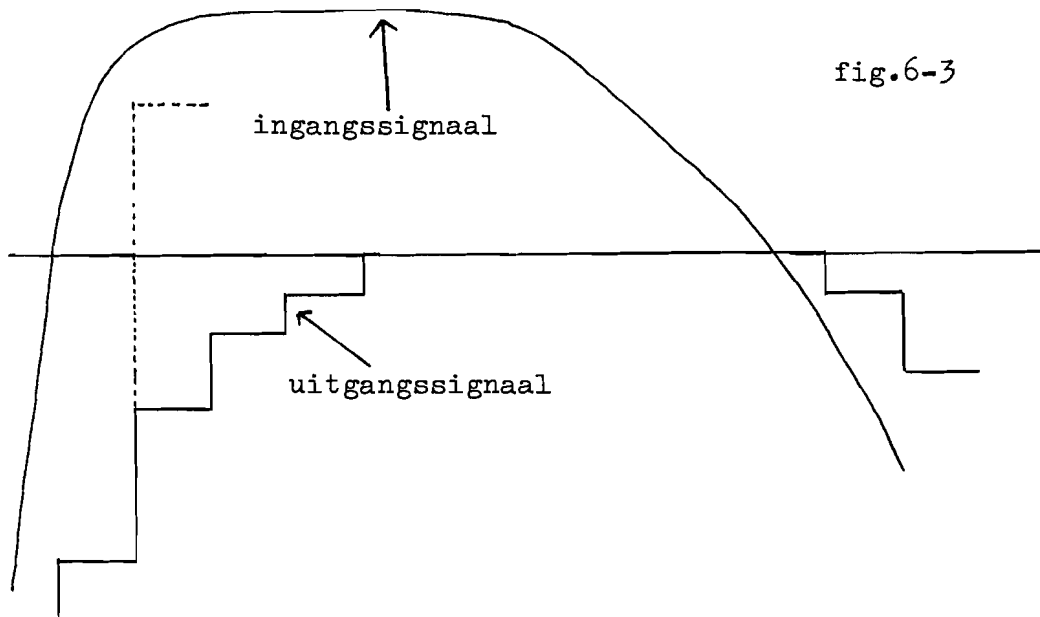
De in- en output zijn naar keuze disk of videogeheugen.

Register 1 bevat de stapgrootte. Geheugenplaatsen \$DEKA t/m \$DEK3 bevatten het uitgangssignaal. Het eerste ingelezen sample van een reeks wordt als schatting voor het volgende sample genomen. Er wordt van uitgegaan dat de samples binaire waarden hebben die tussen 0 en A liggen. A is de maximale amplitude die men bij de initialisatie opgeeft. Deze mag niet groter zijn dan 255. Grotere samples worden niet door de subroutines NEEMS

en STUURS geaccepteerd daar deze ontworpen zijn om te werken met datawoorden van 8 bit. Elke keer dat een nieuw sample moet worden gehaald springt de subroutine naar \$ADM2. Daarna wordt het nieuwe sample met de schatting \$SCHA vergeleken. Als de schatting groter is dan het sample wordt \$DEKA 0, anders 1. Vervolgens wordt gekeken of het uitgangsbite \$DEKA gelijk is aan het vorige \$DEK1, zo ja of ze gelijk zijn aan \$DEK2, zo ja of ze gelijk zijn aan \$DEK3. Afhankelijk van de uitkomst van deze vergelijkingen wordt de inhoud van AC(1), de stapgrootte, aangepast. Nu wordt afhankelijk van het 0 of 1 zijn van \$DEKA de stapgrootte van de oude schatting \$OSCH afgetrokken of opgeteld en dit wordt dan de nieuwe schatting \$SCHA. Als \$SCHA kleiner dan 0 of groter dan \$MAX is treedt de overflow beveiliging in werking die ervoor zorgt dat de stapgrootte wordt aangepast. Deze werkt als volgt: Het verschil tussen het vorige sample en de grens wordt bepaald. Het halve verschil wordt nu als nieuwe stapgrootte genomen. Hierdoor wordt voorkomen dat door een eenmalige piek de ADM weer met de kleinste stapgrootte gaat werken. Zie fig. (6-3)

6.2 Modificaties

Wenst men een ander bewerkingsschema in te stellen dan is dat mogelijk. De faktor 0.57 kan in elk getal tussen de 0 en



1 worden veranderd door 57 een waarde tussen 0 en 100 te geven. Evenals verderop in dit verslag bij het ADPCM programma gebeurd is zijn de samples om rekentechnische redenen, IMP16 heeft geen floating point, met 100 vermenigvuldigd. De faktor 1 wordt gerealiseerd door niets te doen en de factoren 2 en 4 door het resp. 1 en 2 plaatsen naar links schuiven van AC(1). Wil men in plaats van 2 of 4 andere factoren dan moet het programma tussen de labels \$A3 en \$A4 gewijzigd worden. Stel bv. 1.5 i.p.v. 2 dan wordt SHL 1,1 vervangen door:

MPY	\$150
DIV	\$100
SKG	0,\$50

```

      JMP      $XX
      AISZ    1,1
$XX:   JMP      $A3

```

Door i.p.v. 150 een ander getal tussen 0 en 999 te nemen kunnen alle factoren tussen 0.01 en 9.99 worden gekozen. Op precies dezelfde manier kan de factor 4 worden veranderd. De laatste regel wordt dan:

```
$XX:   SETST  5
```

De factor 1 kan gewijzigd worden door het stukje programma tussen de regels 87 en 88 in te voegen. Regel 88 `JMP $A3` vervalt dan daar in de laatste regel al staat:

```
$XX:   JMP      $A3
```

Grotere amplitudes dan 255 vereisen andere subroutines `STUURS` en `NEEMS`. Voor modificaties hiervan zie hoofdstuk 8. Trouwens, als men grotere amplitudes wenst toe te passen dan zal het rekenwerk meestal niet meer in een register passen door het ontbreken van floating point. Men moet dan dus veel meer veranderen. Wil men de `ADM` subroutine niet op data uit het videogeheugen of van de disk laten werken maar op data die elders in het geheugen van de `IMP` staat dan moeten `NEEMS` en `STUURS` vervangen worden.

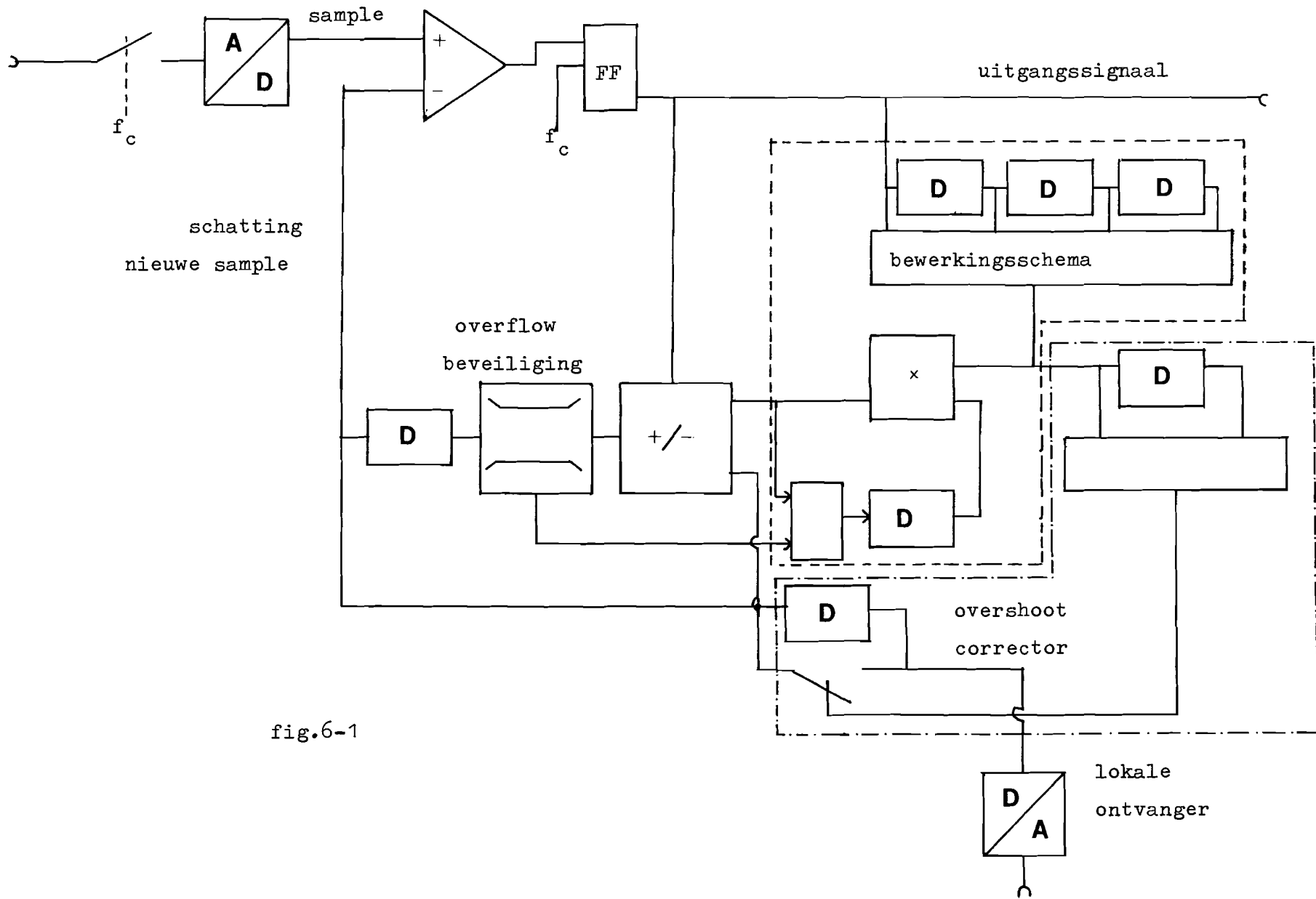


fig.6-1

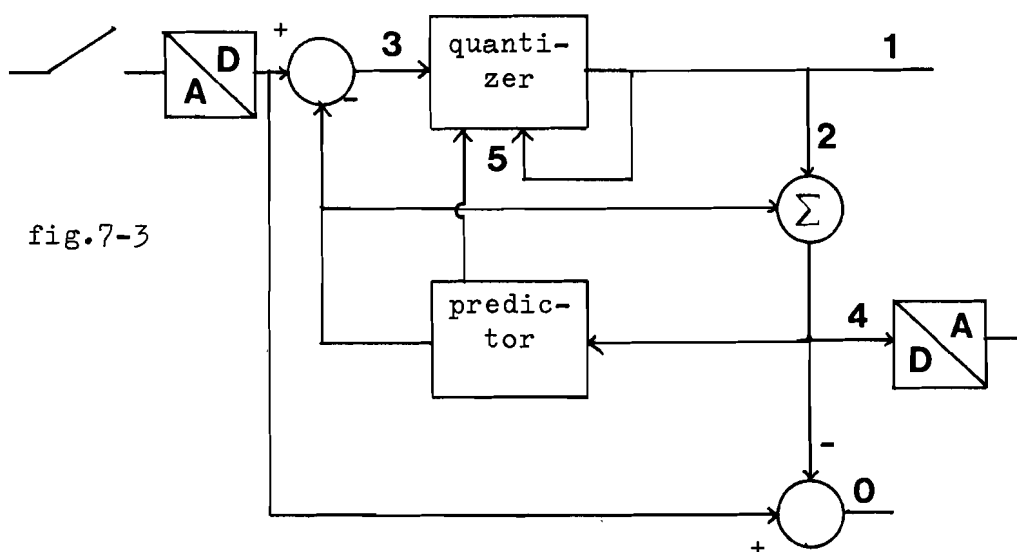
Toelichting bij fig.6-2

- a) Trek van het sample de schatting af en bepaal het teken van het verschil.
- b) Afhankelijk van dit teken wordt een positieve of negatieve puls als uitgangssignaal verstuurd
- c) Bepaal uit de laatste vier uitgangsbits een stapvergrotingsfactor.(zie 4.1)
- d) bereken nieuwe stapgrootte.
- e) Voor positief of negatief teken (zie a) wordt bij de oude schatting een stap opgeteld resp.afgetrokken.
- f) Kijk of de schatting binnen de grenzen ligt.Zoniet halveer het verschil tussen oude schatting en de grens en neem dit als stapgrootte.
- g) Was de voorgaande stap er een van maximale grootte en is de nieuwe stap kleiner?Gebruik dan de nieuwe schatting als ontvangersignaal anders oude schatting.

7 Het ADPCM programma

Fig.7-1 geeft het principe circuit van het systeem volgens Mitra zoals dat is gesimuleerd. Fig.7-2 geeft het principe circuit van het voorgestelde systeem waarbij de predictor de quantizer stuurt. Uit de figuren blijkt een grote mate van overeenkomst in de opbouw van beide systemen. Het enige verschil zit in het regelsignaal van de quantizer. Daarom zal het programma voor het systeem van Mitra besproken. Daarna zullen de verschillen met het andere systeem worden toegelicht. Bijlage B geeft de flowcharts van beide systemen. Het ADPCM programma is uit drie delen opgebouwd. Het eerste stuk verzorgt het inlezen van constanten, het geven van de juiste startwaarden en het schoonvegen van sommige geheugenplaatsen. Daarna volgt het programmadeel dat de feitelijke simulatie uitvoert. Is de simulatie voltooid dan wordt deze afgesloten met een stuk programma dat voor het uitprinten van het signaal- en ruisvermogen en van de verdelingsfunctie van het uitgangssignaal zorg draagt. Het simulatiedeel kan vijf verschillende outputs leveren. In fig.7-3 is aangegeven waar in het systeem deze signalen optreden. De gebruiker kiest een uitgangssignaal door het intypen van een getal. Welk getal bij welk uitgangssignaal behoort staat in de nu volgende tabel.

type	output
0	foutsignaal
1	uitgangssignaal ADPCM
2	correctieterm
3	XN
4	simulatie ontvanger
5	$\Delta(n)$



De opbouw van dit programma wijkt iets af van de andere programma's die voor de IMP geschreven zijn. Normaal staan alle subroutines in de TSECT evenals stukken van het hoofdprogramma tenzij dit niet te lang is, bv. 0-40 regels, want dan staat het in zijn geheel in de BSECT. De grens hiervoor wordt oa. bepaald door het verstoren van de werking van de globale variabelen. Om echter het programma overzichtelijk te houden

zijn nu een stel geheugenplaatsen in de ASECT gereserveerd. Deze zijn zowel in het hoofdprogramma als in de subroutines met dezelfde namen aangeduid. Doordat deze delen gescheiden zijn geassembleerd kon dat. De reden dat hiervoor geen globale namen gekozen zijn is dat anders de adressering in sommige subroutines minder overzichtelijk en gecompliceerder zou zijn. Verder is de in andere programma's gebruikte constructie aangehouden.

Aangenomen is steeds dat de drempels en uitgangsniveaus van de quantizer symmetrisch om de 0 liggen. Dit maakt het mogelijk om met een quantizer voor positieve waarden te volstaan en daar de modulus van hetingangssignaal aan toe te voeren. Dit is in de gerealiseerde systemen ook gebeurd. Om verschillende systemen te kunnen simuleren is het aantal drempels en de ligging ervan instelbaar gemaakt. Dan zijn twee mogelijkheden.

a) de machine berekent zelf de niveaus halverwege de drempels zoals Max dat aangeeft.

b) de machine laat de gebruiker de niveaus intypen.

Om dit te realiseren zijn er twee verschillende subroutines BEPNIV geschreven. Tijdens het loaden kan men bepalen welke van de twee subroutines men wenst. Voor het ADPCM programma maakt dit geen verschil daar in beide gevallen dezelfde subroutine aanroep JSR BEPNIV wordt gebruikt. De gren-

zen waartussen DELTAN kan variëren zijn instelbaar. Achter DELTAMIN en DELTAMAX kan men de waarden opgeven. Bij de keuze van deze waarden moet men rekening houden met het kleinste niveau. $\text{DELTAMIN} \times \text{kleinste niveau} \geq 100$. Waarom dit 100 en niet 1 moet zijn wordt duidelijk als men weet dat in het programma sommige getallen met een factor 100 zijn vermenigvuldigd om problemen met breuken te vermijden. De grens voor DELTAMAX is van veel minder belang daar de overflow beveiliging ervoor zorgt dat de gevolgen van te grote stappen, nl. overflow, niet worden gedaan. De subroutine HALV halveert het door de gebruiker opgegeven aantal drempels en zorgt voor een correcte afronding bij een oneven aantal. Met de keuze van de factor gamma kan men samen met deltamin en deltamax de mate van adaptie regelen. Hoe dit werkt is in hoofdstuk 4.2 beschreven. Heeft het programma gamma ingelezen dan rekent BEPMI de MI's uit. Zowel gamma als deltamin en deltamax zijn met 100 vermenigvuldigd. Wil men bv. $\gamma = 1.05$ dan moet men 105 intypen en $\text{deltamin} := 0.20$ dan 20 intypen. Nu worden de lengte en het aantal reeksen waarop de simulatie moet worden uitgevoerd ingelezen en opgeslagen. De teller voor de lengte van de reeks wordt met een verlaagd daar er naar verschillen tussen beeldpunten wordt gekeken en er altijd een verschil minder is dan er beeldpunten zijn. Door het al dan niet zetten van statusflag 6 wordt gekozen voor een-

of tweedimensionale correlatie. De feitelijke keuze wordt door statusflag 7 gemaakt. Bij de eerste reeks zijn er nog geen beeldpunten van de vorige reeks bekend zodat men dan altijd eendimensionaal werkt. Statusflag 7 is dan gecleard. Na afloop van de eerste reeks wordt afhankelijk van het al of niet geset zijn van flag 6 flag 7 wel of niet geset. Hiermee is de keuze tussen een- en tweedimensionaal gerealiseerd. Het label LOOPA geeft het begin van de simulatie van een reeks aan. Er wordt een beeldpunt ingelezen, met 100 vermenigvuldigt en als schatting voor het volgende beeldpunt genomen. In het geval van tweedimensionale correlatie wordt de nieuwe schatting bepaald volgens:

$$\text{COMBSC} := \frac{\text{SAMX CORY} + \text{SAMY CORX}}{\text{CORT}}$$

Zie theorie adaptieve predictie voor verklaring. De samples SAMY zijn in een tabel aan het eind van het programma opgeslagen. Hiervoor is achter het label TAB ruimte gereserveerd. De inhoud van POIN geeft het adres aan van de geheugenplaats in de tabel die aan de beurt is. Voor het eerste beeldpunt van een reeks zijn CORX en CORY niet gedefinieerd en worden ze 1 gemaakt evenals CORT. Een bijzondere situatie doet zich voor als CORX=CORY=0 want dan is ook CORT 0 en zou het programma vastlopen. Fysisch wil het 0 zijn van CORX en CORY zeggen dat

$$\text{COMBSC} := \frac{\text{SAMX} + \text{SAMY}}{2}$$

. Door op CORT=0 te testen en dan CORX:=CORY:=CORT:=1 te maken (JSR MAAKN) is deze situatie bereikt. De subroutine DSHR zorgt ervoor dat

SAMX CORY + SAMY CORX door twee worden gedeeld. CORT is altijd al door twee gedeeld. Dit moet omdat de IMP slechts door positieve getallen kan delen. CORX en CORY kunnen maximaal 25500 zijn en hun som dus 51000. Daar dit groter is dan 2^{15} is dit voor de IMP een negatief getal. Met deze deling door twee is dit verholpen. Als voor eendimensionale correlatie is gekozen dan springt het programma naar NSOM en wordt het gedeelte dat COMBSC berekent overgeslagen. De schatting die nog in AC(1) stond wordt dan in COMBSC opgeslagen. Nu kan het feitelijke ADPCM proces beginnen. Er wordt een beeldpunt ingelezen, met 100 vermenigvuldigt en de schatting wordt er vanaf getrokken. Dit resultaat heet XN. Hiervan wordt de modulus bepaald die MODXN heet. Het centrale deel van het ADPCM programma wordt gevormd door het stukje na het label VERGL. Hier wordt MODXN met de drempels vergeleken. Zodra een drempel groter is dan deze modulus wordt het vergelijken stopgezet. Daarom moet er altijd een drempel van 256 of groter zijn. Doordat de drempels met factoren groter of kleiner dan 1 worden vermenigvuldigd kan deze maximale drempel te klein worden. Hiertegen is een beveiliging ingebouwd. Dit gebeurt door te kijken of het maximale aantal drempels is bereikt. Is dit het geval dan wordt deze drempel altijd 256. Afhankelijk van het al dan niet positief zijn van XN wordt het uitgangsniveau van teken verwisselt en bij de voorgaande schatting opgeteld. Daarna wordt

gekeken of het zo verkregen getal groter of kleiner is dan resp. de boven- of ondergrens waarbinnen het gesimuleerde ontvangersignaal moet liggen. (0-255). Voor registeroverflow hoeft men niet bang te zijn. Om overflow te krijgen moet men minstens ± 70 stappen boven de 255 zitten. Hoe komt men daar? dan moet het verschil tussen schatting en sample groter of gelijk aan de grootste drempel zijn. In het geval dat het sample zijn maximale waarde heeft, 255, en het verschil tussen schatting en sample minimaal dan heeft men nog altijd 70 stappen voor overflow optreedt. Stel dat de grootste drempel op bv. 120 ligt dan zal het uitgangsniveau tussen 120-195 liggen. Nu blijkt dat bij grote drempels men het uitgangsniveau beter dicht bij de ondergrens dan bij de bovengrens van het interval kan leggen. Er hoeft dus geen vrees voor overflow te bestaan. Door het verschil tussen het nieuwe sample en het voorgaande in x en y richting te bepalen en hiervan de modulus te nemen worden CORX en CORY bepaald. De som van CORX en CORY heet CORT. In de subroutines BEPMSE en wordt het kwadraat van het signaal en van het foutsignaal bepaald. BEPBEZ zorgt ervoor dat de teller die de index van het uitgangsniveau aangeeft met een verhoogd wordt en OUTPUT stuurt het gewenste outputsignaal naar het scherm of de disk. Daarna wordt de nieuwe DELTAN bepaald, en gekeken of deze binnen de grenzen ligt. Hiermee is het ADPCM proces eenmaal doorlopen. Afhankelijk van het al dan niet 0 zijn van TEL wordt er naar

gekeken of het zo verkregen getal groter of kleiner is dan resp.de boven-of ondergrens waarbinnen het gesimuleerde ontvangersignaal moet liggen.(0-255).Voor registeroverflow hoeft men niet bang te zijn.Om overflow te krijgen moet men minstens ± 70 stappen boven de 255 zitten.Hoe komt men daar?dan moet het verschil tussen schatting en sample groter of gelijk aan de grootste drempel zijn.In het geval dat het sample zijn maximale waarde heeft,255,en het verschil tussen schatting en sample minimaal dan heeft men nog altijd 70 stappen voor overflow optreedt.Stel dat de grootste drempel op bv.120 ligt dan zal het uitgangsniveau tussen 120-195 liggen.Nu blijkt dat bij grote drempels men het uitgangsniveau beter dichterbij de ondergrens dan bij de bovengrens van het interval kan leggen.Er hoeft dus geen vrees voor overflow te bestaan.Door het verschil tussen het nieuwe sample en hetvoorgaande in x en y richting te bepalen en hiervan de modulus te nemen worden CORX en CORY bepaald.De som van CORX en CORY heet CORT.In de subroutines BEPMSE en wordt het kwadraat van het signaal en van het foutsignaal bepaald.BEPBEZ zorgt ervoor dat de teller die de index van het uitgangsniveau aangeeft met een verhoogd wordt en OUTPUT stuurt het gewenste outputsignaal naar het scherm of de disk.Daarna wordt de nieuwe DELTAN bepaald.en gekeken of deze binnen de grenzen ligt.Hiermee is het ADPCM proces eenmaal doorlopen.Afhankelijk van het al dan niet 0 zijn van TEL wordt er naar

LOOP teruggesprongen. Is TEL 0 dan wordt er nogmaals output gegeven. Dit komt omdat er per reeks eenmaal minder output wordt gegeven dan input gevraagd. Aangezien de lengte van deze reeksen overeenkomt met een afmeting van een video-beeld of de lengte van een reeks beelden moet er extra output worden gegeven om problemen bij het zichtbaar maken te voorkomen. Als alle reeksen bewerkt zijn komt het derde stuk van het programma in actie. De verdelingsfunctie van het uitgangssignaal, het signaal- en het ruisvermogen worden afgedrukt. Hiermee is het ADPCM programma klaar. AFLOOP zorgt er nu voor dat men het programma opnieuw kan uitvoeren of dat er een ander programma van de disk wordt gehaald.

Zoals eerder vermeldt is het programma voor het systeem waarbij de predictor de quantizer stuurt grotendeels identiek met het zojuist beschreven programma. Het maakt van dezelfde subroutines gebruik. Alleen heeft het BEPMI niet nodig. De geheugenplaatsen die deze subroutine gebruikt zijn wel gereserveerd om met dezelfde load module voor de subroutines van beide systemen te kunnen volstaan. De statements voor het inlezen van KMIN, MMAX en het bepalen van een- of tweedimensionale correlatie zijn vervallen. In plaats daarvan worden de adaptiefactor en de bovengrens ervan ingelezen. Deze adaptiefactor en bovengrens zijn met 255 vermenigvuldigd. Voor bv. een q van 0.20 moet men 51 intypen. Na het versturen van het out-

put signaal (JSR OUTPUT) zit een wijziging. Van de getallen CORX en CORY wordt de kleinste bepaald. Is dit nul dan wordt DELTAN 100 gemaakt. Als dit ongelijk nul is dan wordt het met XMINE vermenigvuldigd en gedeeld door 255. Is DELTAN groter dan DELMAX dan wordt DELTAN gelijk gemaakt aan DELMAX.

Tot slot volgen nog de samenstelling van de beide load modules van de systemen en een voorbeeld van een simulatie met elk van de systemen.

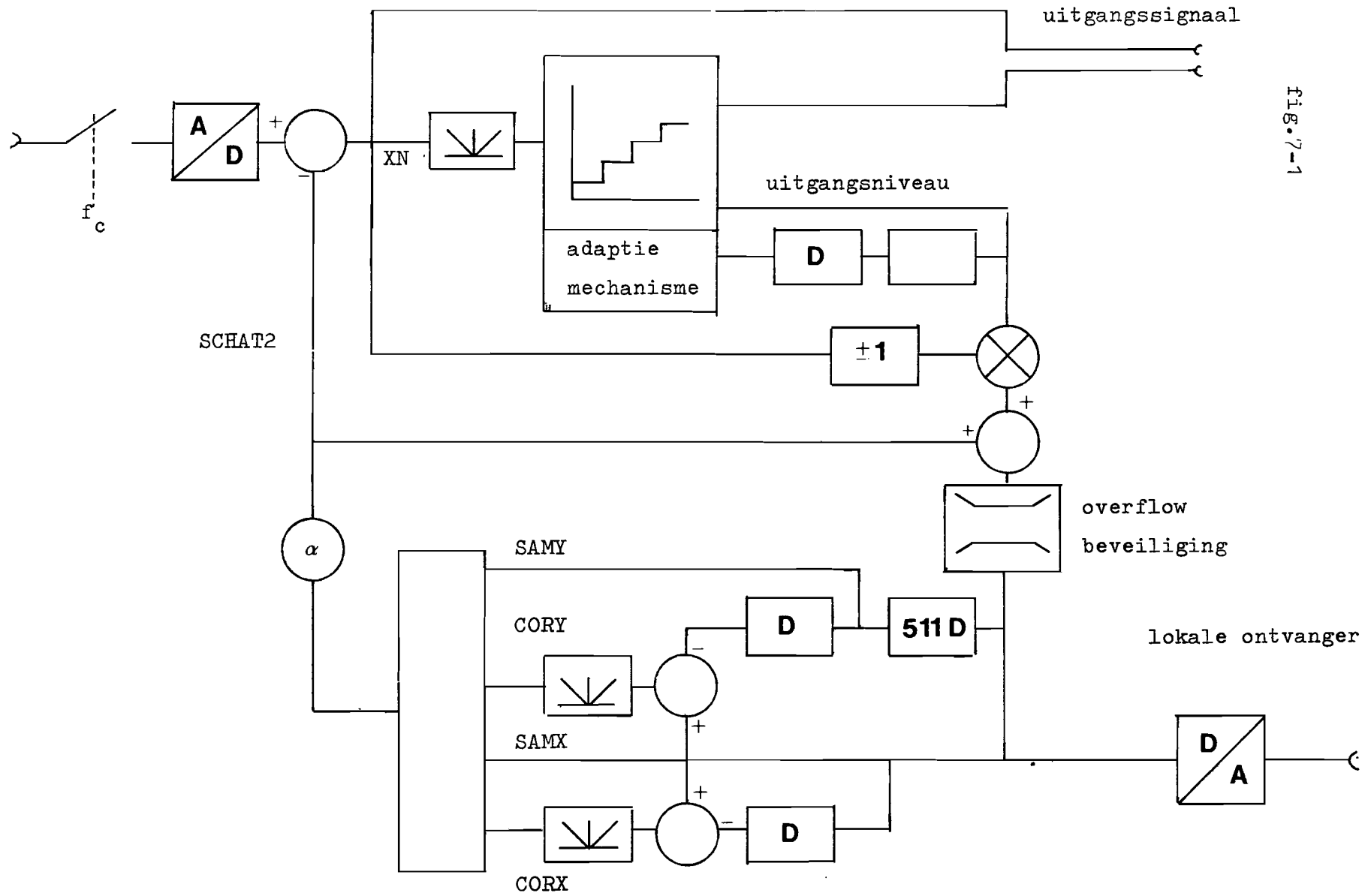


fig. 7-1

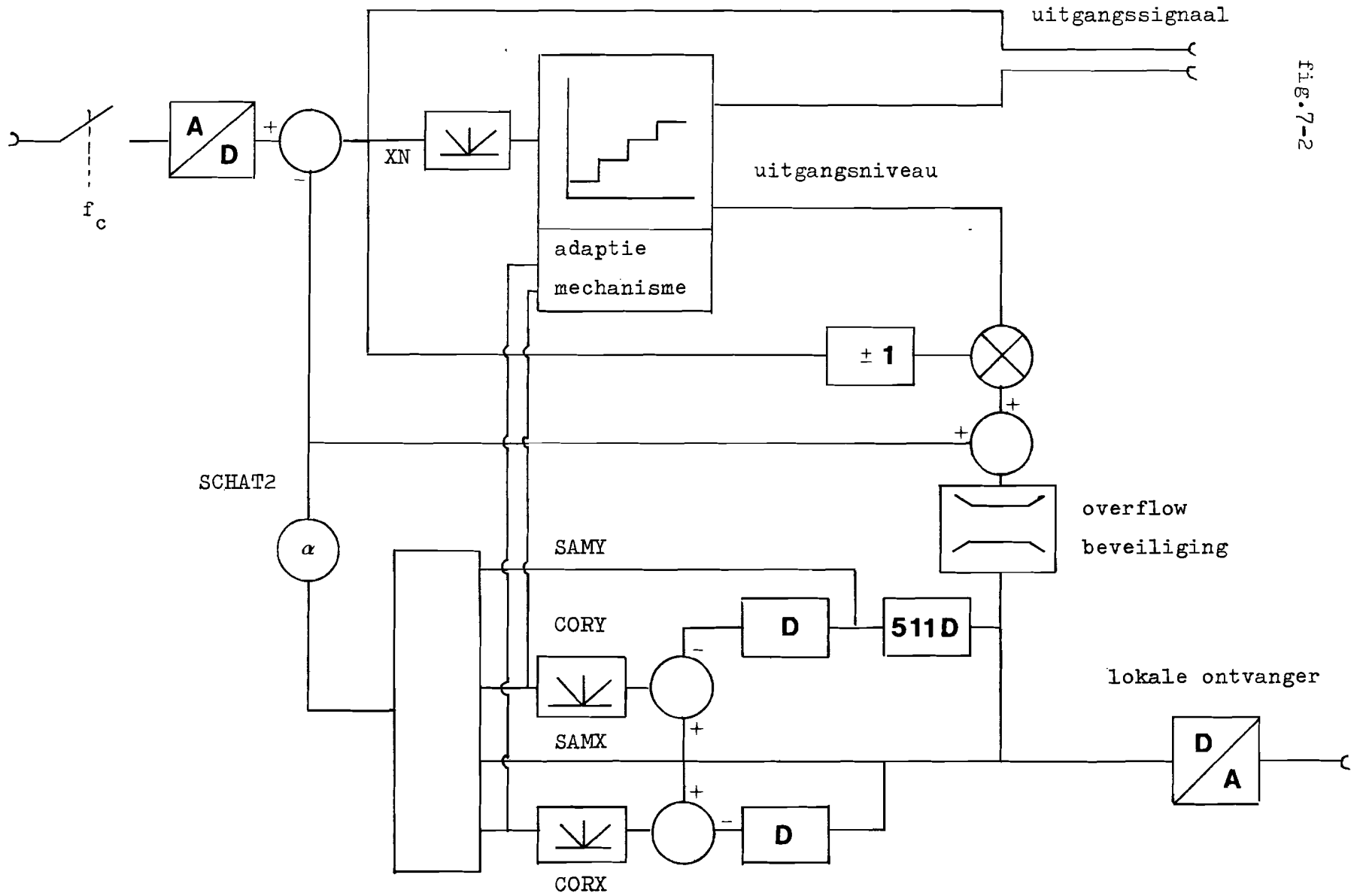


fig. 7-2

Toelichting bij de figuren 7-1 en 7-2.

- a) Van het sample wordt een schatting (SCHAT2) afgetrokken.
- b) Van dit verschil wordt de modulus naar een quantizer toegestuurd.
- c) Deze quantizer wordt voor ieder beeldpunt berekend door een vooraf ingestelde quantizer met een factor DELTAN te vermenigvuldigen.
- d) Het uitgangsniveau van de quantizer wordt voor het teken gecorrigeerd (zie b) en opgeteld bij SCHAT2.
- e) Gekeken wordt of deze som binnen de grenzen 0-255 valt. Anders maakt de overflowbeveiliging de som gelijk aan de grens die wordt overschreden.
- f) Het nu verkregen signaal is het lokale ontvanger signaal.
- g) Uit dit lokale ontvangersignaal wordt de schatting berekend volgens:
$$\text{COMBSC} = \frac{\text{SAMY CORX} + \text{SAMX CORY}}{\text{CORX} + \text{CORY}} \quad (\text{Zie } 5.1)$$
- h) DELTAN wordt bij het systeem volgens Mitra bepaald door DELTAN met een getal ν^{m_i} te vermenigvuldigen waarbij m_i door het uitgangsniveau van de quantizer wordt bepaald. Bij het systeem waarbij de predictor de quantizer stuurt wordt DELTAN gelijk aan $1+qV$ met $V=\min(\text{CORX},\text{CORY})$. In beide systemen is DELTAN begrensd. (Zie verder 4.2 en 5.2)
- i) $\text{SCHAT2} := \text{COMBSC}$

ANTAL NIVEAUS - 4

PREMPELS - 4

NIVEAUS - 2,

6

ADAPTIE FAKTOR - 70

LEARNPARAMETER - 100

LENGTE REEKS - 512

ANTAL REEKSEN - 256

SOORT OUTPUT - 4

DELTA MAX - 42

INPUT VAN DISK OF VIDEOMEM.? D

FIRST INPUT SECTOR HEX - 269

OUTPUT NAAR DISK OF VIDEOMEM? V

SIGNAL= 13830,49096

NOISE= 111,95494

STAPVERDELING:

1,02218 0,28598

DISK? (Y OR N)

AANTAL NIVEAUS - 8

DREMPELS - 18

40
100

NIVEAUS - 9

27
70
120

DELTAN MIN - 12

DELTAN MAX - 200

GAMMA - 130

MI WORDT 59 76 130 169

LEAKPARAMETER - 100

LENGTE REEKS - 512

AANTAL REEKSEN - 256

SOORT OUTPUT - 4

TWEEDIM. COR.? (Y OR N) - N

INPUT VAN DISK OF VIDEOMEM.? D

FIRST INPUT SECTOR HEX - 269

OUTPUT NAAR DISK OF VIDEOMEM? V

SIGNAL= 13830,49096

NOISE= 108,03632

STAPVERDELING:

 0,56527 0,39162 0,24352 0,10775

DISK? (Y OR N)

REC 040

FILE NAME: DSCLDR

DISC LOADER (REV. F) READY.

!LM

!MP 0420

!RLM 0467

INOUT	INPUT EN OUTPUT ROUTIN	0618	F2EE
	TS =0120:01E2		

!RLM 02A8

INOUT	DISK OF VIDEOMEM	5FB8	4810
	TS =01E3:0502		

!RLM 0403

SUBR	TABEL LEZEN EN PRINTEN	0056	EEB8
	TS =0503:0545 PTR=00FE:00FF		

!RLM 01F5

AFLOOP	AFLOOP ZONDER LESEC	88CE	726F
	TS =0546:056C PTR=00FD:00FD		

!RLM 026C

BINASC	24/6/77	1CE2	EE12
	TS =056D:05BF		

!RLM 03BA

SUBR	INLEZEN NIVEAUS	6C9A	8552
	TS =05C0:06EF PTR=00F9:00FC		

!OBS0

!RLM 0419

ADPCM	PREDIC STUURT QUANT	52E6	91FD
	BS =0000:008B TS =06F0:0A16 PTR=00EE:00F8 ENT=073E		

!ER

!GO

	BS =0000:008B TS =0120:0A16 PTR=00EE:0100 ENT=073E		
--	--	--	--

PGM WRITTEN TO SECTORS 0420:0429

!

DSCLDR

DSCLDR

7-16

DISC LOADER (REV. F) READY.

U M

IMP 03F0

IRLM 0467

INOUT INPUT EN OUTPUT ROUTIN 0618 F2EE
TS =0120:01E2

IRLM 02AB

INOUT DISK OF VIDEOMEM 5FB8 4810
TS =01E3:0502

IRLM 0403

SUBR TABEL LEZEN EN PRINTEN 0D56 EEB8
TS =0503:0545 PTR=00FE:00FF

IRLM 01F5

AFLOOP AFLOOP ZONDER LESEC 88CE 726F
TS =0546:056C PTR=00FD:00FD

IRLM 026C

BINASC 24/6/77 1CE2 EE12
TS =056D:05BF

IRLM 03C8

SUBR ALLES INLEZEN 9AB0 4E1F
TS =05C0:06B0 PTR=00FA:00FC

!OBSO

IRLM 03D9

ADPCM EEN OF TWEEDIM. COR. 72F2 17A9
BS =0000:008C TS =06B1:09E6 PTR=00EE:00F9 ENT=070D

!ER

!GO

BS =0000:008C TS =0120:09E6 PTR=00EE:0100 ENT=070D

PGM WRITTEN TO SECTORS 03F0:03F9

!

FILE NAME: DSCLDR

DISC LOADER (REV. F) READY.
ILM

MP 03E0

IRLM 0467

INOUT INPUT EN OUTPUT ROUTIN 0618 F2EE
TS =0120:01E2

IRLM 02A8

INOUT DISK OF VIDEOMEM 5FB8 4810
TS =01E3:0502

IRLM 0403

SUBR TABEL LEZEN EN PRINTEN 0D56 EEB8
TS =0503:0545 PTR=00FE:00FF

IRLM 01F5

AFLOOP AFLLOOP ZONDER LESEC 88CE 726F
TS =0546:056C PTR=00FD:00FD

IRLM 026C

BINASC 24/6/77 1CE2 EE12
TS =056D:05BF

IRLM 03BA

SUBR INLEZEN NIVEAUS 6C9A 8552
TS =05C0:06EF PTR=00F9:00FC

!0BS0

IRLM 03D9

ADPCM EEN OF TWEEDIM. COR. 72F2 17A9
BS =0000:008C TS =06F0:0A25 PTR=00ED:00F8 ENT=074C

!ER

!G0

BS =0000:008C TS =0120:0A25 PTR=00ED:0100 ENT=074C

PGM WRITTEN TO SECTORS 03E0:03E9

!

Subroutines voor het ADPCM programma

In dit hoofdstuk worden de subroutines behandeld die speciaal voor het ADPCM programma geschreven zijn. Een aantal hiervan is algemeen toepasbaar. Veel subroutines werken met variabelen die ook elders gebruikt worden of gedefinieerd zijn. Nu zijn er verschillende methodes om variabelen aan een subroutine over te dragen. Zie lit.(11).

- a) Men kan in de registers bepaalde waarden zetten.
dit gaat slechts voor maximaal 4 variabelen.
- b) Men zou ze op de stack kunnen zetten. Dit is niet aan te bevelen daar de stack ook gebruikt wordt om terugkeer adressen bij subroutinesprongen te bewaren. Daar de stack slechts 16 plaatsen diep is zou overflow optreden.
- c) Men zet de variabelen op geheugenplaatsen. Deze methode is niet handig voor subroutines voor algemeen gebruik. De programmeur wordt nl. sterk in de keuze van zijn labels beperkt.
- d) men zet de variabelen in een tabel en geeft de subroutine het adres van de tabel, bv. in een register, mee.

Door de constructie van de subroutinesprongen bij de IMP is het een handige methode om de tabel direct onder de subroutine te plaatsen. Op de bovenste plaats van de stack staat dan automatisch het eerste adres van de tabel.

Wel moet aan het eind van de subroutine RTS vervangen worden door RTS X met X de lengte van de tabel. Deze methode is de meest geschikte voor de overdracht van meerdere variabelen in subroutines voor algemeen gebruik. De programmeur wordt op geen enkele manier in de keuze van zijn labels beperkt. Ook de fabrikant gebruikt deze methode vaak bv. DISKIO, LMSG etc. In dit verslag worden oa. TABIN en TABOUT beschreven die zo zijn opgebouwd.

8.1 BEPMI, BEPNIV en LEESDR

Van de subroutines BEPMI, BEPNIV en LEESDR bestaan verschillende versies. De reden daarvoor is bij de beschrijving van het ADPCM programma al gegeven. In de eenvoudigste vorm bestaan ze alle drie uit weinig meer dan het bepalen van het aantal in te lezen getallen door de subroutine HALV. Het feitelijke inlezen gebeurt dan door TABIN. Bij LEESDR wordt het aantal in te lezen getallen met een verlaagd. Dit kan omdat de grootste drempel altijd 256 is en al in het ADPCM programma is ingebouwd. Van BEPMI bestaat er een versie waarbij een factor ν wordt ingelezen. Met behulp van deze ν wordt dan de reeks:

$$\nu^{-N}, \dots, \nu^{-1}, (\nu^0), \nu^{+1}, \dots, \nu^N \text{ gevormd.}$$

Het al of niet aanwezig zijn van de term ν^0 hangt af van het even of oneven zijn van het aantal termen van de reeks zoals dat door HALV berekend wordt. De subroutine leest eerst ν in.

Met behulp van HALV wordt het aantal termen dat uitgerekend moet worden bepaald. BEPMI is wat gecompliceerd van opbouw door het feit dat de volgorde waarin de termen het handigst uitgerekend worden niet overeenstemt met de volgorde waarin ze in het geheugen moeten staan. Door de vorige term steeds met V te vermenigvuldigen en dan door 100 te delen wordt de volgende term van het deel van de reeks met de positieve machten bepaald. De negatieve machten worden dan door het delen van 10000 door de positieve machten uitgerekend. Als de gehele reeks bepaald is zorgt TABOUT voor het uitprinten.

8.2 BEPMSE en OUTPUT

De subroutine BEPMSE berekent het kwadraat van het ingangssignaal en dat van het foutsignaal. Beide worden door 2^1 gedeeld om register verflow bij sommatie te voorkomen. Na ieder bewerkt beeldpunt worden deze kwadraten bepaald en bij de voorgaande opgeteld. Dit gebeurt op de geheugenplaatsen SIGNAL en NOISE in dubbel precisie.

De subroutine OUTPUT zet het gewenste outputsignaal in AC(0), zodat STUURS dit signaal naar de disk of het videogeheugen kan sturen. In SOORT staat een getal dat de soort output aangeeft. Door de SKNE instructies worden de subroutines van de niet gewenste outputsymbolen overgeslagen en slechts die van

het gewenste signaal uitgevoerd. Dit maakt het mogelijk om andere signalen aan de bestaande toe te voegen. Door voor de

regel: JSR STUURS de regels:

SKNE 1,\$ZES

LD 0,NIEUW

of: SKNE 1,\$ZES

JSR \$NIEU

toe te voegen is een extra outputsignaal gerealiseerd. Wel dient men er voor te zorgen dat \$ZES en NIEUW of \$NIEU worden gedefinieerd. Dit zal in de meeste gevallen betekenen dat via een globale naam een extra variabele van het ADPCM programma naar de subroutine moet worden overgedragen. In de verschillende subroutines die OUTPUT gebruikt worden de getallen die van het ADPCM programma komen in een zodanige vorm gebracht dat ze door STUURS kunnen worden verwerkt. SCHAT en TERM moeten bv. door 100 worden gedeeld om tussen 0-255 te liggen. Bij de index moet meer gebeuren. Daar de quantizer slechts naar de modulus van XN kijkt moet afhankelijk van het teken van XN index in orde gemaakt worden. Dit corrigeren gebeurt als volgt: Er wordt gekeken of het aantal niveaus even of oneven is. Is dit oneven dan wordt \$HULP 1 anders 0. Daar de index bij de loop VERGL altijd met 1 verhoogd wordt moet van INDEX 1 afgetrokken worden. De bits van AC(0) worden dan afhankelijk van het positief of negatief zijn van

XN gecomplementeerd. Door er nu \$HULP bij op te tellen is de juiste index verkregen. Dit getal wordt nog met 16 vermenigvuldigt (SHL 0,4) om het op het scherm beter zichtbaar te maken.

8.3 BEPBZ, TABUIT, VEEG en DSHR

De subroutine BEPBZ bekijkt de index van het uitgangsniveau. De bij deze index behorende teller wordt in dubbel precisie met een opgehoogd. Deze tellers staan op de geheugenplaatsen achter het label TAB. De subroutine VEEG zorgt ervoor dat deze tellers 0 gemaakt worden voor de start van de simulatie. Na het beëindigen van de simulatie zorgt TABUIT voor het uitprinten van de tellers. De opzet van deze subroutine vertoont veel overeenkomst met TABOUT. Nu wordt echter BINASC i.p.v. TODEC gebruikt omdat dubbel precisie woorden moeten worden uitgeprint. BINASC is een voor het globl systeem geschikt gemaakte subroutine van Jan Janssen lit.(12). De subroutine DSHR zorgt voor het 1 plaats naar rechts schuiven van AC(0) en AC(1) waarbij het minst significante bit van AC(0) het meest significante bit van AC(1) wordt.

8.4 TABOUT, TABIN en AFLOOP

De subroutine TABOUT is geschreven op tabellen uit te printen. Men dient op te geven waar de tabel in het geheugen staat, hoe lang hij is en hoeveel elementen men per regel afgedrukt wil hebben. De aanroep is als volgt:

```
JSR    TABOUT
      .WORD  ADRES
      .WORD  LENGTE
      .WORD  AANTAL
```

De subroutine TABIN is geschreven om tabellen in te lezen. Men dient op te geven waar de tabel in het geheugen moet komen te staan en hoe lang hij is. De aanroep wordt dan:

```
JSR    TABIN
      .WORD  ADRES
      .WORD  LENGTE
```

TABIN en TABOUT lezen en leveren decimale getallen. De getalconversie gebeurt door de subroutines TIDEC en TODEC. Zou men in plaats van het decimale een ander talstelsel willen gebruiken dan is dat mogelijk. Men moet dan overal in het programma de woorden TODEC en/of TIDEC vervangen door TOHEX en/of TIHEX, TOOCT en/of TIOCT of TOBIN en/of TIBIN als men resp. hexadecimale, octale of binaire getallen wenst. TABIN en TABOUT werken als volgt: Eerst worden de registers gered.

Dit gebeurt door het aanroepen van \$SAVE. Daarna worden met behulp van het terugkeeradres dat op de top van de stack staat de gegevens over adres en lengte van de tabel ingelezen. Tot zover zijn beide identiek. Doordat beide identiek zijn is dit deel van de subroutines zelf weer als een subroutine (\$AANT) geschreven. De eerste drie regels, het verwisselen van de top van de stack met AC(3), kan niet in \$AANT. Dan zou het terugkeeradres van \$AANT i.p.v. TABIN of TABOUT op de top van de stack staan. Na de aanroep van \$AANT staat in AC(2) het terugkeeradres+1. Het verschil tussen TABIN en TABOUT zit in de loop die nu moet worden doorlopen. Bij TABOUT is een extra loop opgenomen om het aantal getallen per regel bij te houden. Is dit aantal bereikt dan wordt opnieuw naar \$A gesprongen en dit aantal weer in AC(1) ingelezen. \$LOC is de pointer die bijhoudt welk element van de tabel aan de beurt is.

De subroutine AFLOOP maakt het mogelijk om na het uitvoeren van een programma terug te springen naar de disk om een volgend programma van de schijf te halen of om het programma opnieuw te starten. Men roept de subroutine als volgt aan:

```
JSR    AFLOOP
JMP    START
```

waarbij START het label van het entrypoint is. Na aanroep ver-

schijnt DISK? (Y OR N) op de TTY. Typt men N in dan wordt het programma herstart. Typt men Y in dan leest men:

SECTOR HEX - en men typt hierachter drie hexadecimale digits in die het sectornummer van het verlangde load module aangeven. AFLOOP heeft om dit sectornummer in te lezen de subroutine LESEC nodig. Zie hiervoor NEEMS, STUURS.

8.5 NEEMS en STUURS

De subroutines NEEMS en STUURS halen en brengen data van of naar diskette of videogeheugen. De keuze tussen diskette of videogeheugen wordt pas gemaakt tijdens het uitvoeren van een programma door de gebruiker. Hierdoor is het niet meer noodzakelijk om beelden van de diskette in het videogeheugen te plaatsen. Tevens is men in staat om zonder gebruik te maken van het videogeheugen beelden te bewerken. De subroutines gebruikt men als volgt. Voordat de eerste data wordt bewerkt neemt men de volgende regels in zijn programma op:

```
LD      0,X8000
JSR     NEEMS
JSR     STUURS
```

Door nu telkens wanneer men data wenst te ontvangen of versturen NEEMS resp. STUURS in het programma op te nemen wordt een 8 bits datawoord in AC(0) gezet of uit AC(0) ge-

haald.(8 minst significante bits).Attentie! Tijdens data-transport moeten bit 14 en 15 van AC(0) 0 zijn.Wenst men de zelfde data opnieuw te bewerken dan moeten de volgende regels in het programma worden opgenomen.

```
LD      0,XCOOO
```

```
JSR     NEEMS
```

NEEMS zal nu weer op de eerste sector die is ingelezen zijn data gaan halen.Daar het videogeheugen cyclisch is opgebouwd wordt dit na 2^{17} beeldpunten automatisch gereset.Voor minder dan 2^{17} beeldpunten dient dit resetten met de TRANSMITTER/RECEIVER schakelaar te gebeuren.Deze is niet via software te besturen.Na het laatste datatransport is het mogelijk dat er een hoeveelheid data in het buffer staat die nog op de diskette moet worden geschreven.Daarom voegt men:

```
LD      0,XCOO
```

```
JSR     STUURS
```

aan het programma toe.Bij het uitvoeren van het programma typt men achter INPUT VAN DISK OF VIDEOMEM?

```
en OUTPUT NAAR DISK OF VIDEOMEM?
```

D (=diskette) of V (=videogeheugen) in .Achter

```
FIRST INPUT SECTOR HEX -
```

en FIRST OUTPUT SECTOR HEX - typt men de sectornummers van de eerste input- resp. outputsector in.Dit moeten altijd 3 hexadecimale karakters zijn.Een fout in de eerste

twee karakters kan men corrigeren door een spatie in te typen. Oude programma's die nog met VTRANS en VREC werken kunnen aangepast worden door VTRANS door NEEMS en VREC door STUURS te vervangen. Verder moeten de extra regels worden toegevoegd die in het voorgaande zijn genoemd.

8.6 Opbouw NEEMS en STUURS

Uit de flowcharts, bijlage C voor NEEMS en bijlage D voor STUURS, blijkt dat de subroutines uit twee delen zijn opgebouwd. Het ene deel verzorgt het feitelijke datatransport, (tweede pag. van de bijlages), terwijl het andere deel de keuze tussen diskette en videogeheugen bepaald en de juiste startwaarden levert. De keuze tussen beide delen wordt gemaakt door het al of niet 1 zijn van bit 15 van AC(0). \$STA1 en \$STA2 bevatten de informatie of er met disk, \$STA1=\$STA2=2, videogeheugen, \$STA1=\$STA2=0 of een combinatie, \$STA1=0 en \$STA2=2 of \$STA1=2 en \$STA2=0 wordt gewerkt. Bij het transport van en naar de disk zorgen de instructies STB en LDB ervoor dat 8 bits van AC(0) in en uit het databuffer dat aan het eind van de subroutines staat worden gehaald. AC(3) fungeert dan als indexregister. Stel AC(3) bevat X'4444. Het low-order byte van AC(0) wordt dan in het high-order byte van geheugenplaats X'2222 opgeslagen. Als AC(3) X'4445 bevat dan wordt het low-order byte van AC(0) in het low-order byte van X'2222 opgeslagen. Voor het laden van

AC(O) geldt hetzelfde.

8.7 Uitbreidingen en modificaties van NEEMS en STUURS

Indien de TRANSMITTER/RECEIVER schakelaar van het videogeheugen software bestuurbaar wordt gemaakt dan kan dat in NEEMS en STUURS worden opgenomen. Hierdoor zou het mogelijk zijn om minder dan 2^{17} beeldpunten meerdere malen uit het videogeheugen te halen zonder menselijk ingrijpen. Als in de huidige opzet het videogeheugen zou worden vervangen dan betekent dit dat ook de besturing zou worden gewijzigd. Om dan bestaande software zonder ingrijpende modificaties te kunnen gebruiken zou men slechts NEEMS en STUURS hoeven te veranderen. Verder zijn bewerkingen denkbaar waarvoor geen handige interface instellingen kunnen worden gevonden. Doordat 1 sector van de diskette overeenkomt met 1 samengesteld blok van het videogeheugen kan men door met enkele extra tellers de sectornummering tijdens het opslaan met STUURS te veranderen over grotere samengestelde blokken beschikken. Zo kunnen beelden worden opgeborgen op een manier dat men ze efficiënter kan bewerken.

8.8 Het programma BEWEEG

Dit programma verzorgt een lineaire eenparige beweging met een beperkt instelbare snelheid. De werking is als volgt. Er wordt een lijn van het videogeheugen ingelezen en deze wordt 32 x teruggezet over steeds 0,2 of 4 plaatsen verschoven. Dit verschuiven gebeurt door het beeldpunt uit een buffer te lezen met behulp van een indexregister AC(3). Door nu telkens bij de beginwaarde de snelheid op te tellen en in AC(3) modulo de lijnlengte te rekenen wordt de beweging verkregen. Voor het andere raster dient de beginwaarde over de halve snelheid verschoven te liggen. Het beeld zit als volgt in het videogeheugen. Zie fig.8-1

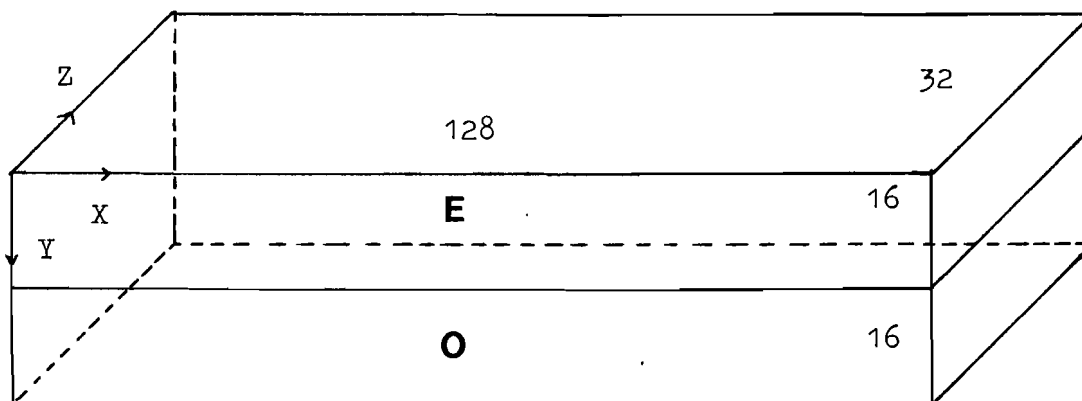


fig.8-1

9.1 Metingen en conclusies ADM

Het ADM programma is gebruikt om reeksen van 32 beeldjes te bewerken. Bij stilstand en lage snelheden werkt de ADM goed. Zodra de snelheid groter wordt of er sprongen optreden dan duurt het enige beeldjes voordat de ADM zich aan de nieuwe situatie heeft aangepast. Dit was overigens geheel volgens verwachting. Bij metingen in het vlak werkte de ADM beter dan in eerste instantie verwacht mocht worden. Vlakken werden redelijk goed weergegeven. Alleen verticale overgangen werden zeer rafelig door edgebusiness. Het vermoeden bestond dat door in twee richtingen te kijken deze edgebusiness onderdrukt zou kunnen worden. Om dit te onderzoeken is het ADM programma gewijzigd. I.p.v. de schatting waarmee werd vergeleken is nu bij deze waarde de waarde van het pel dat op de vorige lijn ligt opgeteld. en deze som is door twee gedeeld en als schatting gebruikt. Inderdaad werden de verticale overgangen nu correct weergegeven. De horizontale overgangen waren behoorlijk in kwaliteit achteruit gegaan. Nu hoeft dit laatste ook geen verbazing te wekken. Terwijl voor het bepalen van de amplitude van de schatting in de x richting 4 voorgaande samples worden gebruikt telt het resultaat van het bovenliggende pel voor 50% van zijn amplitude mee in de uiteindelijke schatting. Dit is iets

wat zou moeten worden verbeterd. Wel heeft dit experiment als resultaat opgeleverd dat het voor de reductie van edgebusiness niet noodzakelijk is om de samplefrequentie te verhogen. D.m.v. een goed predictie schema in zowel x als y richting kan edgebusiness worden onderdrukt.

9.2 Metingen en conclusies ADPCM

Uit de metingen met het ADPCM systeem blijkt dat bij een gelijk aantal niveaus ADPCM een verbetering oplevert t.o.v. DPCM en deze verbetering is groter naarmate het aantal niveaus kleiner is. Deze verbetering is zowel objectief als subjectief. De S/N verhouding van het adaptieve systeem is beter dan die van het niet adaptieve. Nu moet men zich van dat beter zijn van de S/N verhouding geen al te grote verwachtingen maken. Het gaat om verbeteringen in de grootte orde van 0.1 - 1 dB. Subjectief is de verbetering veel groter dan op grond van de verbetering in de S/N verhouding mocht worden verwacht. Door met een gering aantal niveaus te quantiseren ontstaan door het DPCM systeem in egale gebieden contouren. Het ADPCM systeem vervaagt deze contouren volledig bij bitrates waar ze bij het DPCM systeem duidelijk zichtbaar zijn. Hieruit kan men concluderen dat de S/N verhouding geen geschikt kwaliteitscriterium is voor dit

soort systemen. Het nut ervan is slechts dat het als indicatie dienst kan doen omdat subjectief betere systemen meestal ook een betere S/N verhouding bezitten. De mate van verbetering in beeldkwaliteit behoeft echter totaal niet met de verbetering in S/N verhouding overeen te stemmen. De adaptatiesnelheid die door de parameter wordt bepaald blijkt niet veel invloed op de kwaliteit van het beeld te hebben. Het is meer een kwestie van wel of niet adaptief dan van hoe sterk adaptief. De optimale ligging van niveaus en drempels blijkt niet aan de theorie van Max te voldoen. Het is gunstiger om de hogere niveaus dichterbij de benedendrempels van de intervallen te leggen. T.o.v. DPCM blijkt ADPCM vaker de hogere niveaus te gebruiken. Dit is voor verdere redundantiereductie een nadeel. Als men de stapverdeling van de ADPCM's bekijken blijkt dat de kleinste stap veruit het meest wordt gebruikt. Aangezien bij de kleinste stappen niveauverlaging optreedt moet het ADPCM systeem regelmatig op de ondergrens van zijn regelbereik vastlopen. Experimenteel is dit bevestigd. Daarbij is ook gebleken dat dit veel vaker het geval is dan uit de resultaten van Mitra en Goldstein en Liu zou zijn te verwachten. Dit is te verklaren door aan te nemen dat videosignalen veel meer stukken met een constant niveau bezitten dan het geval is bij de gesimuleerde signalen die in de literatuur worden gebruikt. De adaptieve predictie blijkt

De adaptieve predictie voldoet goed in de praktijk. Van het veegeffect dat bij niet adaptieve meerdimensionale predictie optreedt is niet te zien. Figuren met scherpe overgangen worden door dit systeem goed weergegeven. Op de verbetering van het uitgangssignaal met betrekking tot de mogelijkheden voor verdere redundantiereductie met behulp van noiseless source coding wordt in hoofdstuk 9.3 dieper ingegaan.

9.3 Metingen en conclusies van het nieuwe type ADPCM

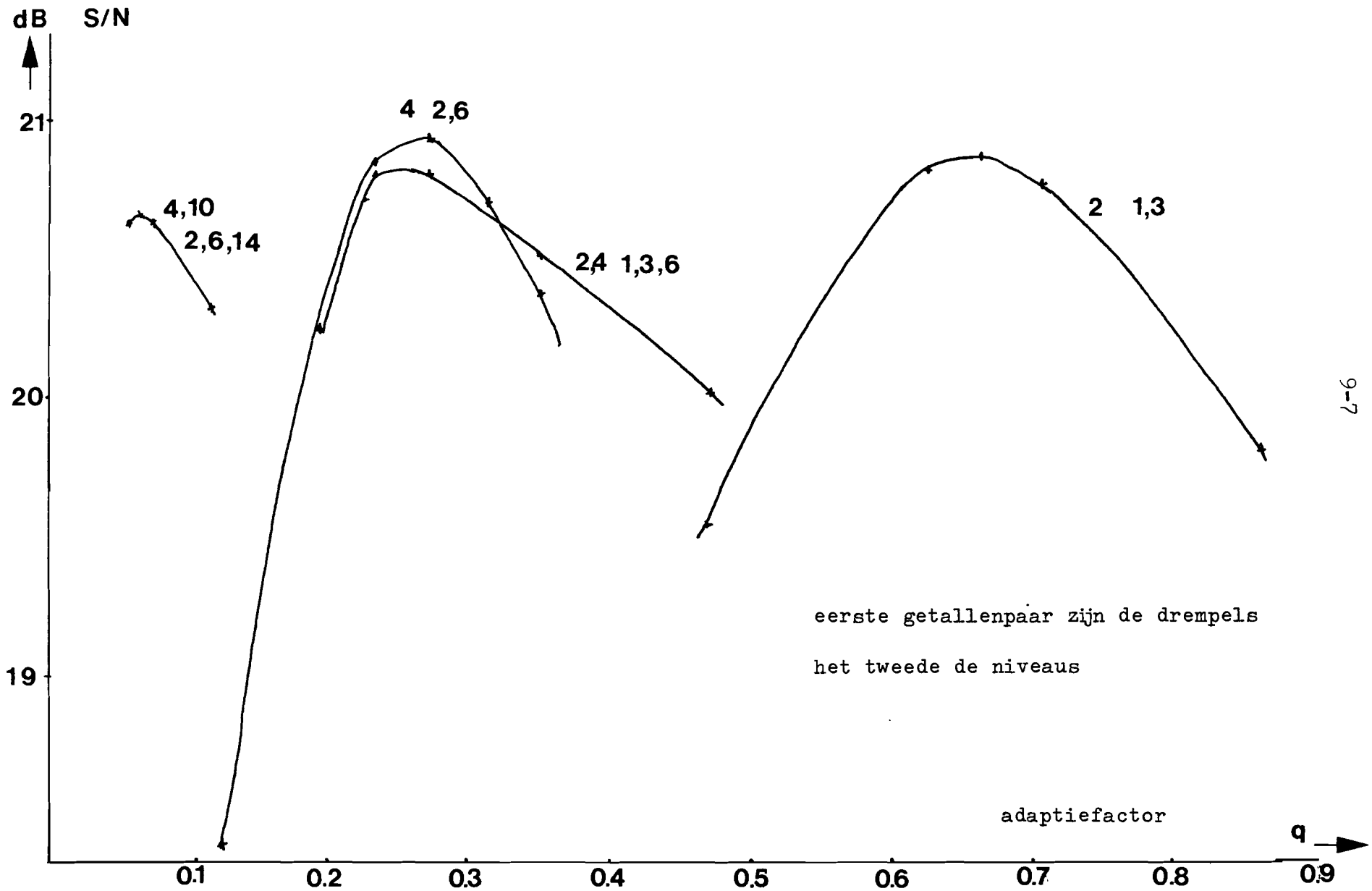
Het ADPCM systeem waarbij de prediktor de quantizer stuurt levert bij lagere bitrates een betere beeldkwaliteit, S/N verhouding en een beter uitgangssignaal dan het systeem van Mitra. Dit is ook verklaarbaar daar bij dit systeem predictie en quantisatie zijn gekoppeld. Dit is bij een ADPCM volgens Mitra niet het geval zodat bij lage bitrates quantisatiefouten de beeldkwaliteit behoorlijk kunnen beïnvloeden. De gemeten afhankelijkheid tussen adaptiefactor en S/N verhouding met verschillende quantizers als parameter fig. 9-1 laat zien dat er voor een quantizer een optimale adaptiefactor bestaat. Voor kleinere adaptiefactoren wordt de beeldkwaliteit door het veegeffect gedegradeerd terwijl grotere adaptiefactoren de korrelruis doen toenemen en de flank-

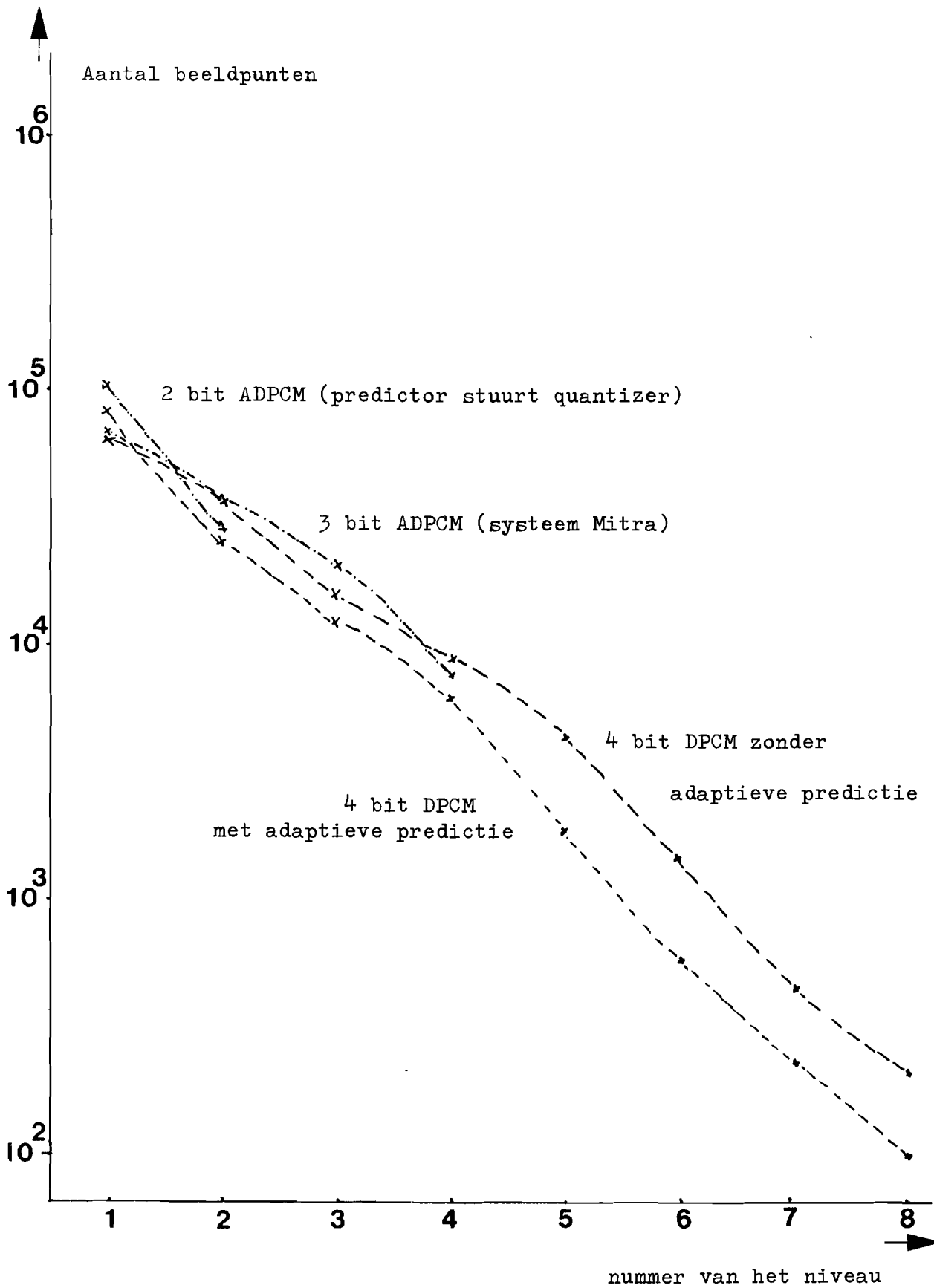
responsie door overshoot effecten verslechteren. Overigens moet bij dit soort systemen aan de S/N verhouding niet te veel waarde worden gehecht want systemen met 6 en 8 niveaus geven een visueel betere weergave dan systemen met 4 niveaus hetgeen niet uit de S/N verhouding blijkt. Het moet mogelijk zijn om met dit systeem een beeld met 2 bit per pel te coderen met een aanvaardbare beeldkwaliteit. Wel zullen dan enkele van de in hoofdstuk 5.7 voorgestelde verfijningen moeten worden toegepast. Het systeem blijkt ook betere mogelijkheden tot verdere redundantie reductie te bezitten dan dat van Mitra aangezien de verdelingsfunctie van het uitgangssignaal gepiekt is. Het moet mogelijk zijn met zo'n systeem onder de 1 bit per pel te komen als men op het uitgangssignaal noiseless source coding toepast. Daarvoor moet er dan waarschijnlijk wel gebruik worden gemaakt van driedimensionale predictie, een verfijnder schema voor predictie en quantizer sturing en een quantizer die efficiënter werkt doordat drempels en niveaus een functie zijn van de amplitude van de schatting.

Het toepassen van meerdimensionale adaptieve predictie kan zelfs bij DPCM systemen zinvol zijn omdat de quantizer dan een uitgangssignaal kan leveren dat zich beter noiseless laat coderen. In deze situatie dient men niet alleen perceptuele criteria te hanteren om het al dan niet toepassen van

meerdimensionale adaptieve predictie te beoordelen. Hoewel er visueel soms nauwelijks of geen verbeteringen hoeven op te treden kan de te bereiken mate van statistische redundantie reductie op het uitgangssignaal zodanig toenemen dat dit het toepassen van meerdimensionale predictie volledig rechtvaardigt. Een iets minder geraffineerd 3 bits ADPCM systeem kan misschien aantrekkelijk zijn in die situaties waar de relatieve eenvoud van het systeem te verliezen is boven maximale redundantie reductie.

Fig. 9-2 geeft de verdelingsfunctie van het uitgangssignaal voor verschillende (A)DPCM systemen. Langs de verticale as staat het aantal beeldpunten uit dat op een bepaald niveau wordt afgebeeld. Langs de horizontale as staan de niveaus uitgezet. Daar de quantizer de modulus van het verschilsignaal quantizeert vallen in deze figuur de positieve en negatieve niveaus samen.





Literatuur

- (1) H.J.R.Schmitz Deltamodulatie van videosignalen, een onderzoek naar datareductie.
Afstudeerverslag T.H.E. augustus 1976
- (2) J.Max Quantizing for minimum distortion.
I.R.E.Trans. on Info.Theory II-6 7-12 1960
- (3) D.J.Sakrison On the role of the observer and a distortion measure in image processing.
IEEE Trans.on Communications COM-25 no 11 november 1977
- (4) M.D.Paez and T.H.Glisson Minimum mean squared error quantization in speech PCM and DPCM systems.
IEEE Trans.on Communications COM-20 april 1972
- (5) A.N.Netravali On quantizers for DPCM coding of picture signals.
IEEE Trans.on Info.Theory IT-23 no 3 may 1977
- (6) J.A.Greefkes en K.Riemens Codemodulatie met digitaal geregelde compansie voor transmissie van spraak.
Philips Technisch Tijdschrift jaargang 31
1970 no 11/12 pp 349-368
- (7) D.Mitra New results from a mathematical study of an adaptive quantizer.
Bell System Technical Journal February 1975 pp 335-368

- (8) L.H.Goldstein and B.Liu Quantization noise in ADPCM systems.
IEEE Trans.on communications COM-25 no 2 february 1977
L.H.Goldstein and B.Liu Quantization error and step size distributions in ADPCM.
IEEE Trans.on Info.Theory IT-23 no 2 march 1977
- (9) D.J.Goodman and A.Gersho Theory of an adaptive quantizer.
IEEE Trans.on Communications COM-22 no 8 August 1975
- (10) J.E.Rooyackers (red.) Handleiding programmeren IMP
Vakgroep EI THE
- (11) A.J.W.Duijvenstein en H.Alblas Inleiding in de systeem-programmatuur deel 2
Kluwer/Deventer 1973
- (12) J.A.M.Jansen Simulatie van een Hademard DPCM systeem voor videocodering.
Afstudeerverslag THE oktober 1977

Appendix A

Sprong in de x richting. Uit de uitdrukkingen voor X_n en Y_n

volgt dat voor E_1 en E_2 moet gelden:

$$E_1 = \frac{E_2 + \delta_1}{2} + \eta_m (1 + q\delta) \quad \text{en}$$

$$E_2 = \frac{E_1 + \delta_2}{2} + \eta_m (1 + q\delta)$$

Oplossen van dit stelsel levert voor E_1 en E_2

$$E_1 = \frac{2\delta_1 - \delta_2}{3} + 2\eta_m (1 + q\delta) \quad \text{of met (9)} \quad E_1 = \frac{2\delta_1 - \delta_2}{3} + 2 \frac{\eta_m}{\eta_1} \delta$$

$$E_2 = \frac{\delta_1 - 2\delta_2}{3} + 2\eta_m (1 + q\delta) \quad \text{of met (9)} \quad E_2 = \frac{\delta_1 - 2\delta_2}{3} + 2 \frac{\eta_m}{\eta_1} \delta$$

Men ziet dat $b - E_1$ en $b - E_2$ de fout in de flankresponsie voorstellen.

noem $a + \delta_1 - a - E_1 = P_1 = \delta/3 + 2\eta_m (1 + q\delta)$

$$a - \delta_2 - a - E_2 = P_2 = -\delta/3 + 2\eta_m (1 + q\delta)$$

$$E_1 - E_2 = E = \delta/3$$

dan $\tilde{X}_{n+1} = a + \frac{P_2 F_1 + E E_2}{P_2 + E}$ en dus

$$X_{n+1} = a + \frac{P_2 F_1 + E E_2}{P_2 + E} + \eta_i (1 + qE) = a + F_2 \quad (a1)$$

$$\tilde{Y}_{n+1} = a + \frac{P_1 F_2 + E E_1}{P_1 + E}$$

$$\bar{Y}_{n+1} = a + \frac{P_1 F_2 + E E_1}{P_1 + E} + \eta_i (1 + qE) = a + F_1 \quad (a2)$$

waarbij η_i het i de uitgangsniveau van de quantizer.

Uit (a1) en (a2) kunnen F_1 en F_2 worden opgelost.

$$F_1 = \frac{E_2 P_1 + E_1 (P_2 + E)}{P_1 + P_2 + E} + \eta_i (1 + qE) \frac{(2P_1 + E)(P_2 + E)}{E(P_1 + P_2 + E)} \quad (13)$$

$$F_2 = \frac{E_1 P_2 + E_2 (P_1 + E)}{P_1 + P_2 + E} + \eta_i (1 + qE) \frac{(2P_2 + E)(P_1 + E)}{E(P_1 + P_2 + E)} \quad (14)$$

$$\text{of: } F_1 = K + C_1 + \eta_1 (1 + qE) A_1$$

$$F_2 = K + C_2 + \eta_2 (1 + qE) A_2$$

$$\text{met } K = \frac{E_2 P_1 + E_1 P_2}{P_1 + P_2 + E}$$

$$A_1 = \frac{(2P_1 + E)(P_2 + E)}{E(P_1 + P_2 + E)}$$

$$A_2 = \frac{(2P_2 + E)(P_1 + E)}{E(P_1 + P_2 + E)}$$

$$C_1 = \frac{E_1 E}{P_1 + P_2 + E}$$

$$C_2 = \frac{E E_2}{P_1 + P_2 + E}$$

invullen van E_1, E_2, E, P_1 en P_2 en toepassen van (9) levert:

$$A_1 = \frac{6 \eta_m (3\eta_1 + 12\eta_m)}{\eta_1 (\eta_1 + 12\eta_m)}, \quad A_2 = \frac{(\eta_m + 2\eta_1)(12\eta_m - \eta_1)}{\eta_1 (\eta_1 + 12\eta_m)}$$

$$K = \frac{-\eta_1^2 \delta + 18\eta_m \eta_1 \delta (\delta_1 - \delta_2) + 72\eta_m^2 \delta}{3\eta_1^2 + 36\eta_1 \eta_m}$$

$$C_1 = \frac{(2\delta_1 - \delta_2)\eta_1 + 6\eta_m}{3\eta_1 + 36\eta_m \delta} \quad \text{en} \quad C_2 = \frac{(\delta_1 - 2\delta_2)\eta_1 + 6\eta_m}{3\eta_1 + 36\eta_m \delta}$$

Appendix B

$$(13)-(14) = F_1 - F_2 + \frac{E_1 E - E_2 E}{E + P_1 + P_2} + \eta_i (1 + qE) \frac{P_1 - P_2}{P_1 + P_2 + E}$$

invullen van P ,P en E (zie appendix A)

$$F_1 - F_2 = \frac{\delta/3}{\sqrt[3]{3} + 4\eta_m (1 + q\delta)} + \eta_i (1 + qE) \frac{2\sqrt[3]{3}}{\sqrt[3]{3} + 4\eta_m (1 + q)}$$

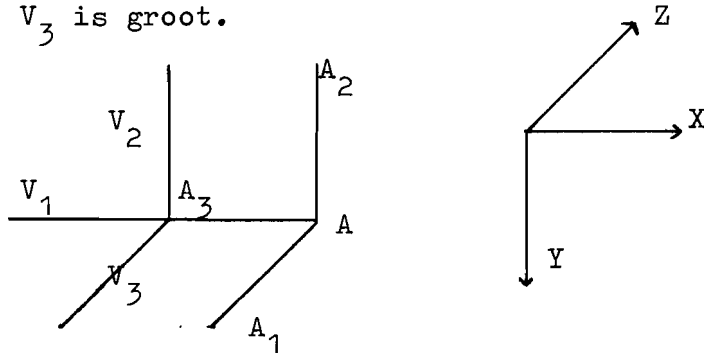
met (10) wordt dit:

$$F_1 - F_2 = \frac{1 + 2\eta_i (1 + q\delta/3)}{1 + 12\eta_m/\eta_1}$$

Appendix C

De invloed van beweging op driedimensionale predictie.

In de gedeelten van het beeld waar de beweging optreedt zal er tussen de punten in de z richting geen correlatie bestaan. Dwz. V_3 is groot.



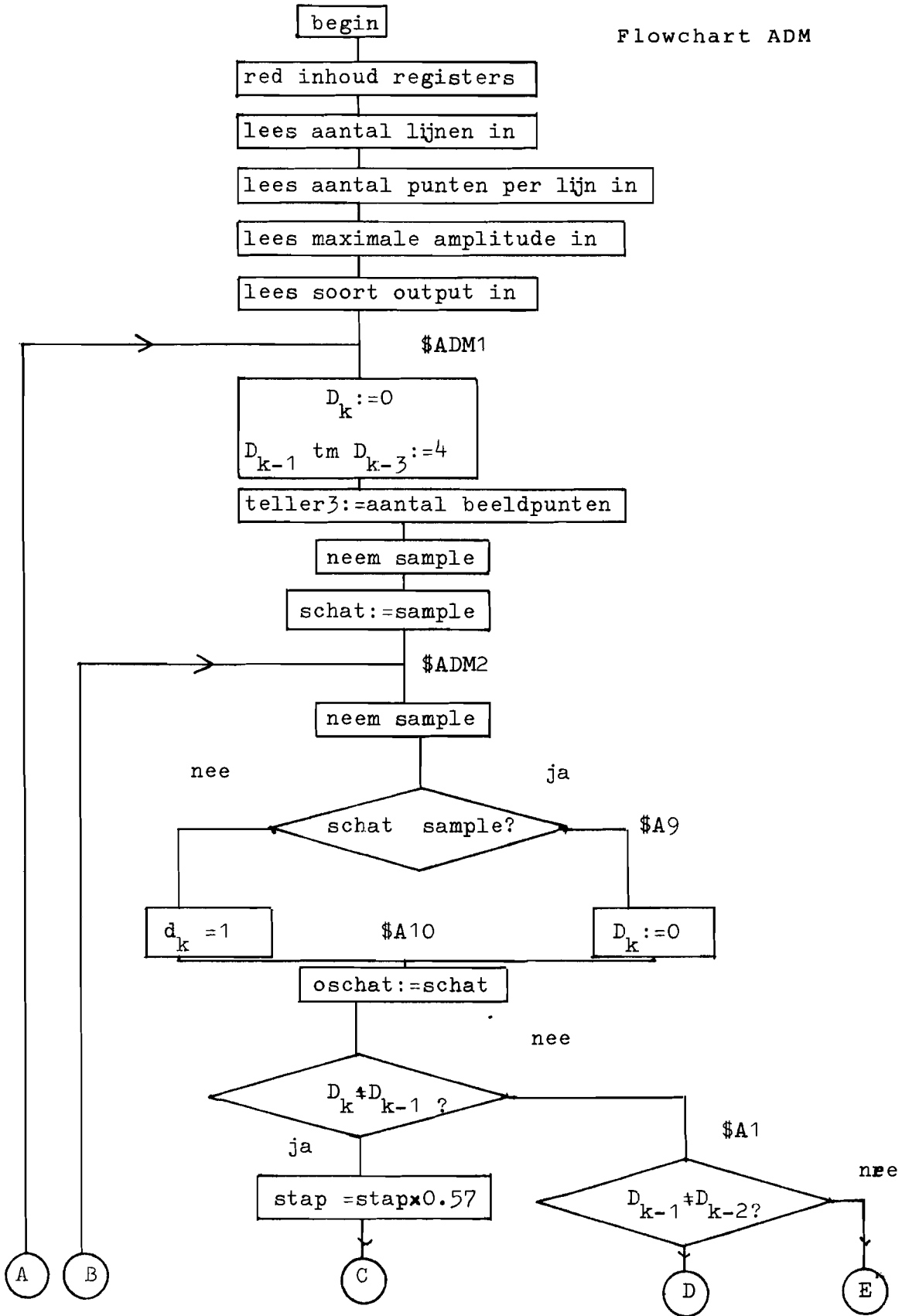
$$A = \frac{A_1 V_1 V_2 + A_2 V_1 V_3 + A_3 V_2 V_3}{V_1 V_2 + V_1 V_3 + V_2 V_3}$$

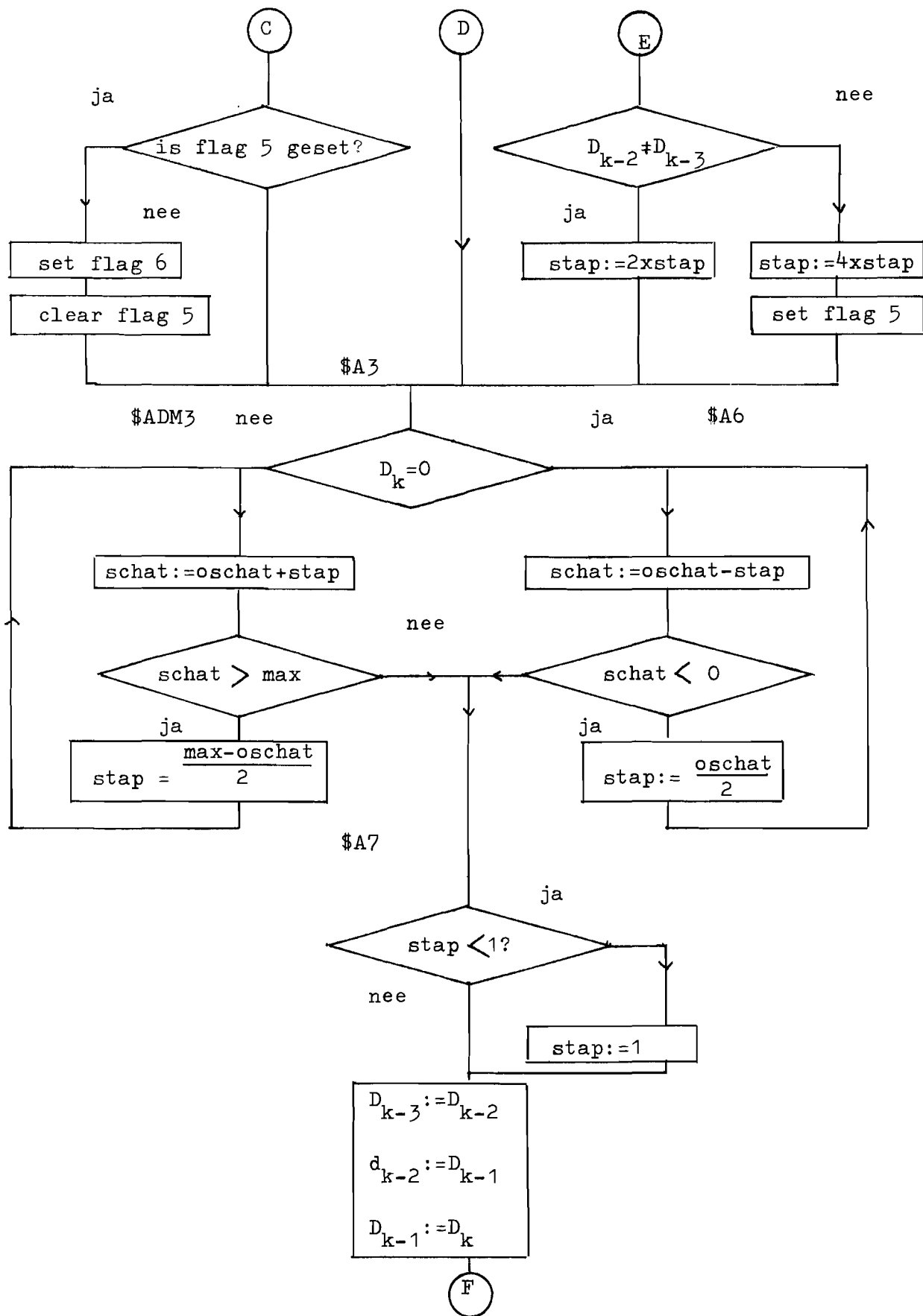
Stel $V_3 \gg V_1$ en $V_3 \gg V_2$ dan

$$A = A_1 \frac{V_1 V_2}{V_1 V_3 + V_2 V_3} + \frac{A_2 V_1 + A_3 V_2}{V_1 + V_2}$$

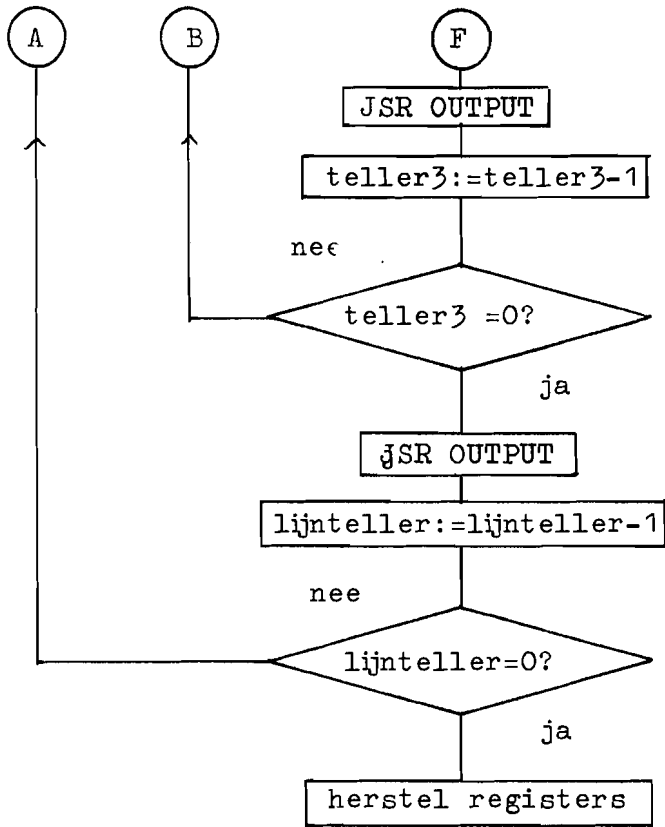
Dwz. A wordt volgens het tweedimensionale systeem geschat plus een stoorterm die in vrijwel alle gevallen klein is t.o.v. de andere term. Als $V_3 \not\gg V_1$ en $V_3 \not\gg V_2$ dan is de schatting in x en y richting ook slecht en levert de derde dimensie nauwelijks verdere degradatie. Bij beweging zullen de bewegeende delen dus volgens een tweedimensionaal systeem worden weergegeven. Door de q (=adaptiefactor) een functie van V_3 te maken is het mogelijk om zowel het stilstaande driedimensionale systeem als het bewegeende tweedimensionale elk hun eigen optimale q te geven.

Flowchart ADM





A-3



END PASS 1

```
1          .TITLE  TESTADM
2          .EXTD
3          .GLOBL  ADM,AFL00P
4          0000   .BSECT
5          000A   .=.+10
6 000A 2801 X   START: JSR    ADM
7 000B 2802 X   JSR    AFL00P
8 000C 200A B   JMP    START
9          000A   .END   START
```

```
ADM      0001  X          AFL00P  0002  X          START  000A  B
```

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=2347

OBJECT CHECKSUM=6418

DISC SECTORS USED

```
FIRST INPUT SECTOR HEX - 01D0
FINAL INPUT SECTOR HEX - 01D0
```

```
FIRST OBJECT SECTOR HEX - 01D1
FINAL OBJECT SECTOR HEX - 01D1
```

NEXT ASSEMBLY

*.ASM

END PASS 1

1			.TITLE	ADM
2			.LOCAL	
3			.GLOBL	NEEMS,STUURS,TIDEC,ADM
4			.EXTD	
5	0000		.TSECT	
6	0000	0D0A A	\$T1: .WORD	0DOA
7	0001	4C45 A	.ASCII	'LENGTE REEKS - '
	0002	4E47 A		
	0003	5445 A		
	0004	2052 A		
	0005	4545 A		
	0006	4B53 A		
	0007	202D A		
	0008	2020 A		
8	0009	0000 A	.WORD	0
9	000A	0D0A A	\$T2: .WORD	0DOA
10	000B	4141 A	.ASCII	'AANTAL REEKSEN - '
	000C	4E54 A		
	000D	414C A		
	000E	2052 A		
	000F	4545 A		
	0010	4B53 A		
	0011	454E A		
	0012	202D A		
	0013	2020 A		
11	0014	0000 A	.WORD	0
12	0015	0D0A A	\$T3: .WORD	0DOA
13	0016	5459 A	.ASCII	'TYPE OUTPUT - '
	0017	5045 A		
	0018	204F A		
	0019	5554 A		
	001A	5055 A		
	001B	5420 A		
	001C	2D20 A		
14	001D	0000 A	.WORD	0
15	001E	0D0A A	\$T4: .WORD	0DOA
16	001F	4D41 A	.ASCII	'MAX. AMPLITUDE - '
	0020	5B2E A		
	0021	2041 A		
	0022	4D50 A		
	0023	4C49 A		
	0024	5455 A		
	0025	4445 A		
	0026	202D A		
	0027	2020 A		
17	0028	0000 A	.WORD	0
18		002A	\$TEL1: .	+.1
19	002A	FFFF A	\$MIN1: .WORD	-1
20	002B	0000 A	\$NUL: .WORD	0
21	002C	0032 A	\$50: .WORD	50
22	002D	0039 A	\$57: .WORD	57
23	002E	0064 A	\$100: .WORD	100
24	002F	7EC3 A	\$MSG: .WORD	X'7EC3
25	0030	7E73 A	\$GECO: .WORD	X'7E73

26	0031	000F	A	\$XF:	.WORD	X'F
27	0032	8000	A	\$8000:	.WORD	X'8000
28	0033	2D00	A	ADM:	JSR	@.+1
29	0034	00BF	T		.WORD	\$SAVE
30	0035	2DF9	A		JSR	@\$MSG
31	0036	0000	T		.WORD	\$T1
32	0037	2803	X		JSR	TIDEC
33	0038	A1F0	A		ST	0,\$TEL1
34	0039	7DEF	A		DSZ	\$TEL1
35	003A	2DF4	A		JSR	@\$MSG
36	003B	000A	T		.WORD	\$T2
37	003C	2803	X		JSR	TIDEC
38	003D	A135	A		ST	0,\$TEL2
39	003E	2DF0	A		JSR	@\$MSG
40	003F	0015	T		.WORD	\$T3
41	0040	2DEF	A		JSR	@\$GECO
42	0041	61EF	A		AND	0,\$XF
43	0042	A131	A		ST	0,\$TYPE
44	0043	2DEB	A		JSR	@\$MSG
45	0044	001E	T		.WORD	\$T4
46	0045	2803	X		JSR	TIDEC
47	0046	A12E	A		ST	0,\$MAX
48	0047	81EA	A		LD	0,\$8000
49	0048	2801	X		JSR	NEEMS
50	0049	2802	X		JSR	STUURS
51	004A	4C00	A	\$ADM1:	LI	0,0
52	004B	A12A	A		ST	0,\$DEKA
53	004C	4C04	A		LI	0,4
54	004D	A129	A		ST	0,\$DEK1
55	004E	A129	A		ST	0,\$DEK2
56	004F	A129	A		ST	0,\$DEK3
57	0050	81D8	A		LD	0,\$TEL1
58	0051	A120	A		ST	0,\$TEL3
59	0052	4D01	A		LI	1,1
60	0053	2801	X		JSR	NEEMS
61	0054	A126	A		ST	0,\$SCHA
62	0055	2801	X	\$ADM2:	JSR	NEEMS
63	0056	E124	A		SKG	0,\$SCHA
64	0057	2103	A		JMP	\$A9
65	0058	4E01	A		LI	2,1
66	0059	A91C	A		ST	2,\$DEKA
67	005A	2102	A		JMP	\$A10
68	005B	4E00	A	\$A9:	LI	2,0
69	005C	A919	A		ST	2,\$DEKA
70	005D	811D	A	\$A10:	LD	0,\$SCHA
71	005E	A11B	A		ST	0,\$OSCH
72	005F	8916	A		LD	2,\$DEKA
73	0060	F916	A		SKNE	2,\$DEK1
74	0061	210C	A		JMP	\$A1
75	0062	0480	A		MPY	\$57
	0063	002D	T			
76	0064	0490	A		DIV	\$100
	0065	002E	T			
77	0066	E1C5	A		SKG	0,\$50
78	0067	2101	A		JMP	\$A2

79	0068	4901	A		AISZ	1,1
80	0069	0745	A	\$A2:	SKSTF	5
81	006A	2117	A		JMP	\$A3
82	006B	0706	A		SETST	6
83	006C	0715	A		CLRST	5
84	006D	2114	A		JMP	\$A3
85	006E	8909	A	\$A1:	LD	2,\$DEK2
86	006F	F907	A		SKNE	2,\$DEK1
87	0070	210B	A		JMP	\$A4
88	0071	2110	A		JMP	\$A3
89		0073		\$TEL3:	.=.+1	
90		0074		\$TEL2:	.=.+1	
91		0075		\$TYPE:	.=.+1	
92		0076		\$MAX:	.=.+1	
93		0077		\$DEKA:	.=.+1	
94		0078		\$DEK1:	.=.+1	
95		0079		\$DEK2:	.=.+1	
96		007A		\$DEK3:	.=.+1	
97		007B		\$OSCH:	.=.+1	
98		007C		\$SCHA:	.=.+1	
99	007C	F9FC	A	\$A4:	SKNE	2,\$DEK3
100	007D	2102	A		JMP	\$A5
101	007E	5D01	A		SHL	1,1
102	007F	2102	A		JMP	\$A3
103	0080	5D02	A	\$A5:	SHL	1,2
104	0081	0705	A		SETST	5
105	0082	4C00	A	\$A3:	LI	0,0
106	0083	F1F2	A		SKNE	0,\$DEKA
107	0084	2109	A		JMP	\$A6
108	0085	8DF4	A	\$ADM3:	LD	3,\$OSCH
109	0086	3700	A		RADD	1,3
110	0087	ADF3	A		ST	3,\$SCHA
111	0088	EDEC	A		SKG	3,\$MAX
112	0089	210F	A		JMP	\$A7
113	008A	85EA	A		LD	1,\$MAX
114	008B	D5EE	A		SUB	1,\$OSCH
115	008C	5DFF	A		SHR	1,1
116	008D	21F7	A		JMP	\$ADM3
117	008E	8DEB	A	\$A6:	LD	3,\$OSCH
118	008F	3681	A		RCPY	1,2
119	0090	5201	A		CAI	2,1
120	0091	3B00	A		RADD	2,3
121	0092	ADE8	A		ST	3,\$SCHA
122	0093	ED96	A		SKG	3,\$MIN1
123	0094	2101	A		JMP	\$A8
124	0095	2103	A		JMP	\$A7
125	0096	85E3	A	\$A8:	LD	1,\$OSCH
126	0097	5DFF	A		SHR	1,1
127	0098	21F5	A		JMP	\$A6
128	0099	E591	A	\$A7:	SKG	1,\$NUL
129	009A	4D01	A		LI	1,1
130	009B	290F	A		JSR	\$OUTP
131	009C	81DB	A		LD	0,\$DEK2
132	009D	A1DB	A		ST	0,\$DEK3
133	009E	81DB	A		LD	0,\$DEK1


```

134 009F A1D8 A      ST      0,$DEK2
135 00A0 81D5 A      LD      0,$DEKA
136 00A1 A1D5 A      ST      0,$DEK1
137 00A2 7DCF A      DSZ    $TEL3
138 00A3 21B1 A      JMP    $ADM2
139 00A4 2906 A      JSR    $OUTP
140 00A5 7DCD A      DSZ    $TEL2
141 00A6 21A3 A      JMP    $ADM1
142 00A7 8127 A      LD      0,$C000
143 00A8 2802 X      JSR    STUURS
144 00A9 291D A      JSR    $REST
145 00AA 0200 A      RTS
146 00AB 4E00 A      $OUTP: LI      2,0
147 00AC F9C7 A      SKNE   2,$TYPE
148 00AD 2107 A      JMP    $OUT1
149 00AE 4E01 A      LI      2,1
150 00AF F9C4 A      SKNE   2,$TYPE
151 00B0 3481 A      RCPY   1,0
152 00B1 4E02 A      LI      2,2
153 00B2 F9C1 A      SKNE   2,$TYPE
154 00B3 2104 A      JMP    $OUT
155 00B4 2108 A      JMP    $UIT
156 00B5 81C0 A      $OUT1: LD      0,$DEKA
157 00B6 5C07 A      SHL    0,7
158 00B7 2105 A      JMP    $UIT
159 00B8 81C1 A      $OUT:  LD      0,$OSCH
160 00B9 0746 A      SKSTF  6
161 00BA 2102 A      JMP    $UIT
162 00BB 81BF A      LD      0,$SCHA
163 00BC 0716 A      CLRST  6
164 00BD 2802 X      $UIT:  JSR    STUURS
165 00BE 0200 A      RTS
166 00BF A110 A      $SAVE: ST      0,$RED
167 00C0 A510 A      ST      1,$RED+1
168 00C1 A910 A      ST      2,$RED+2
169 00C2 AD10 A      ST      3,$RED+3
170 00C3 0080 A      PUSHF
171 00C4 4400 A      PULL
172 00C5 A10E A      ST      0,$RED+4
173 00C6 0201 A      RTS
174 00C7 810C A      $REST: LD      0,$RED+4
175 00C8 4000 A      PUSH
176 00C9 0280 A      PULLF
177 00CA 8105 A      LD      0,$RED
178 00CB 8505 A      LD      1,$RED+1
179 00CC 8905 A      LD      2,$RED+2
180 00CD 8D05 A      LD      3,$RED+3
181 00CE 0200 A      RTS
182 00CF C000 A      $C000: .WORD  X'C000
183      00D5      $RED:  . = .+5
184      0000      .END

```

```

ADM      0033 GT*      NEEMS      0001 X      STUURS      0002 X
TIDEC    0003 X      $100      002E T      $50         002C T
$57      002D T      $8000     0032 T      $A1         006E T
$A10     005D T      $A2       0069 T      $A3         0082 T

```

\$A4	007C	T	\$A5	0080	T	\$A6	008E	T
\$A7	0099	T	\$A8	0096	T	\$A9	005B	T
\$ADM1	004A	T	\$ADM2	0055	T	\$ADM3	0085	T
\$C000	00CF	T	\$DEK1	0077	T	\$DEK2	0078	T
\$DEK3	0079	T	\$DEKA	0076	T	\$GECO	0030	T
\$MAX	0075	T	\$MMSG	002F	T	\$MIN1	002A	T
\$NUL	002B	T	\$OSCH	007A	T	\$OUT	00B8	T
\$OUT1	00B5	T	\$OUTP	00AB	T	\$RED	00D0	T
\$REST	00C7	T	\$SAVE	00BF	T	\$SCHA	007B	T
\$T1	0000	T	\$T2	000A	T	\$T3	0015	T
\$T4	001E	T	\$TEL1	0029	T	\$TEL2	0073	T
\$TEL3	0072	T	\$TYPE	0074	T	\$UIT	00BD	T
\$XF	0031	T						

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=B518

OBJECT CHECKSUM=F3B7

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 01C5

FINAL INPUT SECTOR HEX - 01CA

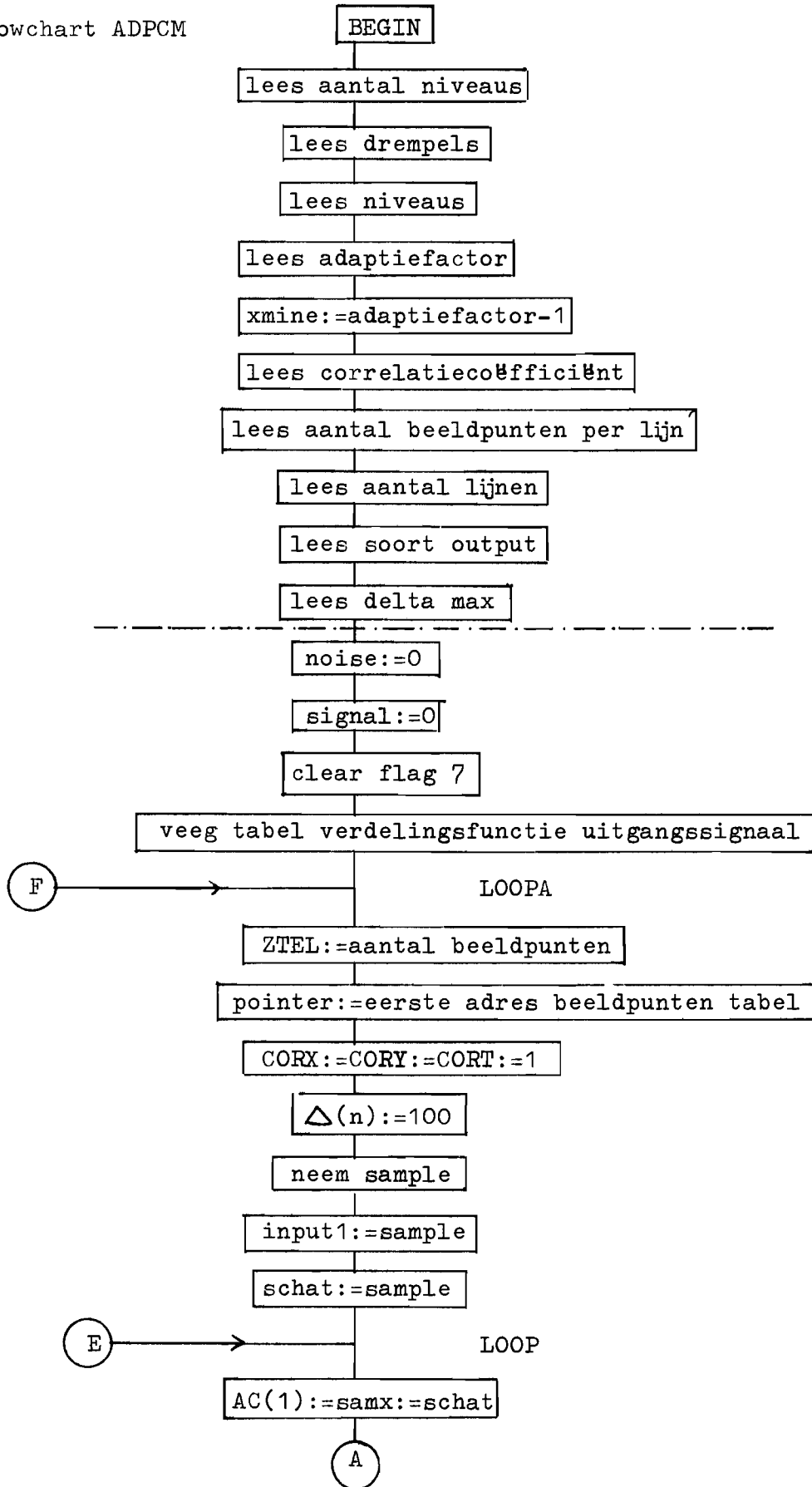
FIRST OBJECT SECTOR HEX - 01CB

FINAL OBJECT SECTOR HEX - 01CC

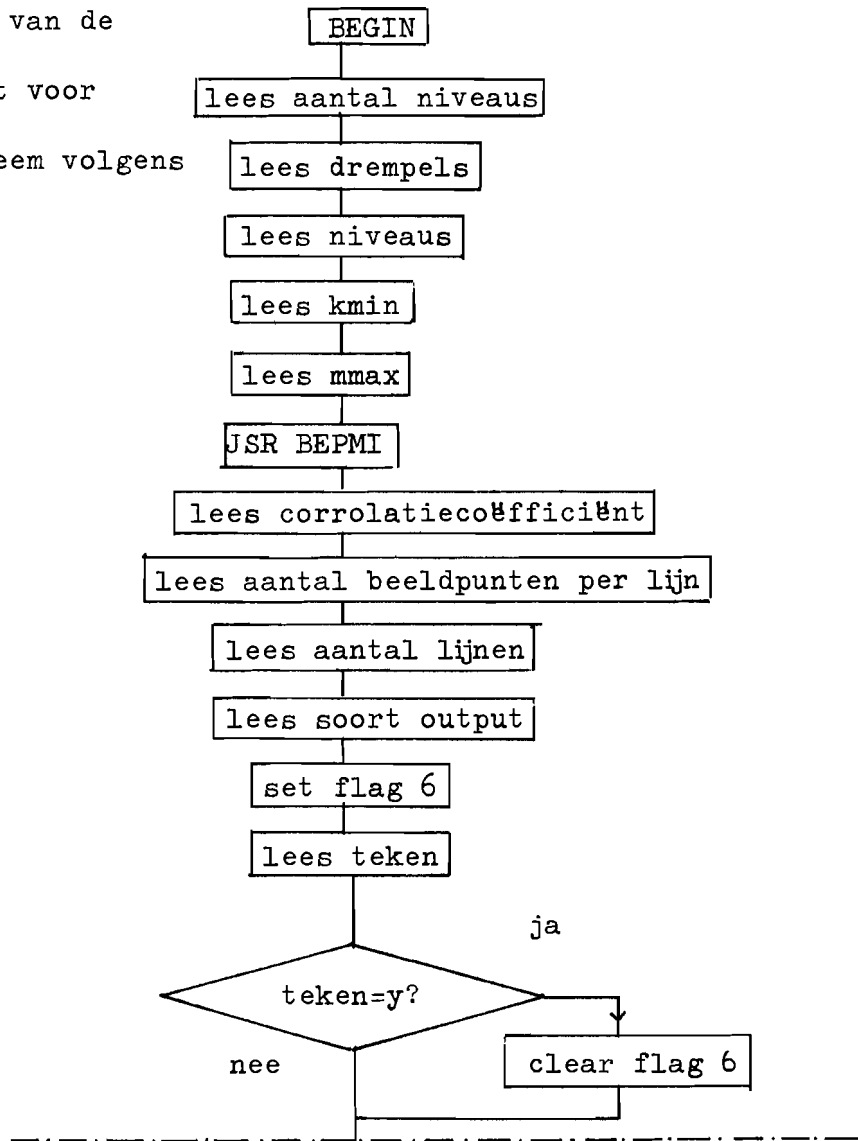
NEXT ASSEMBLY

*.ASM

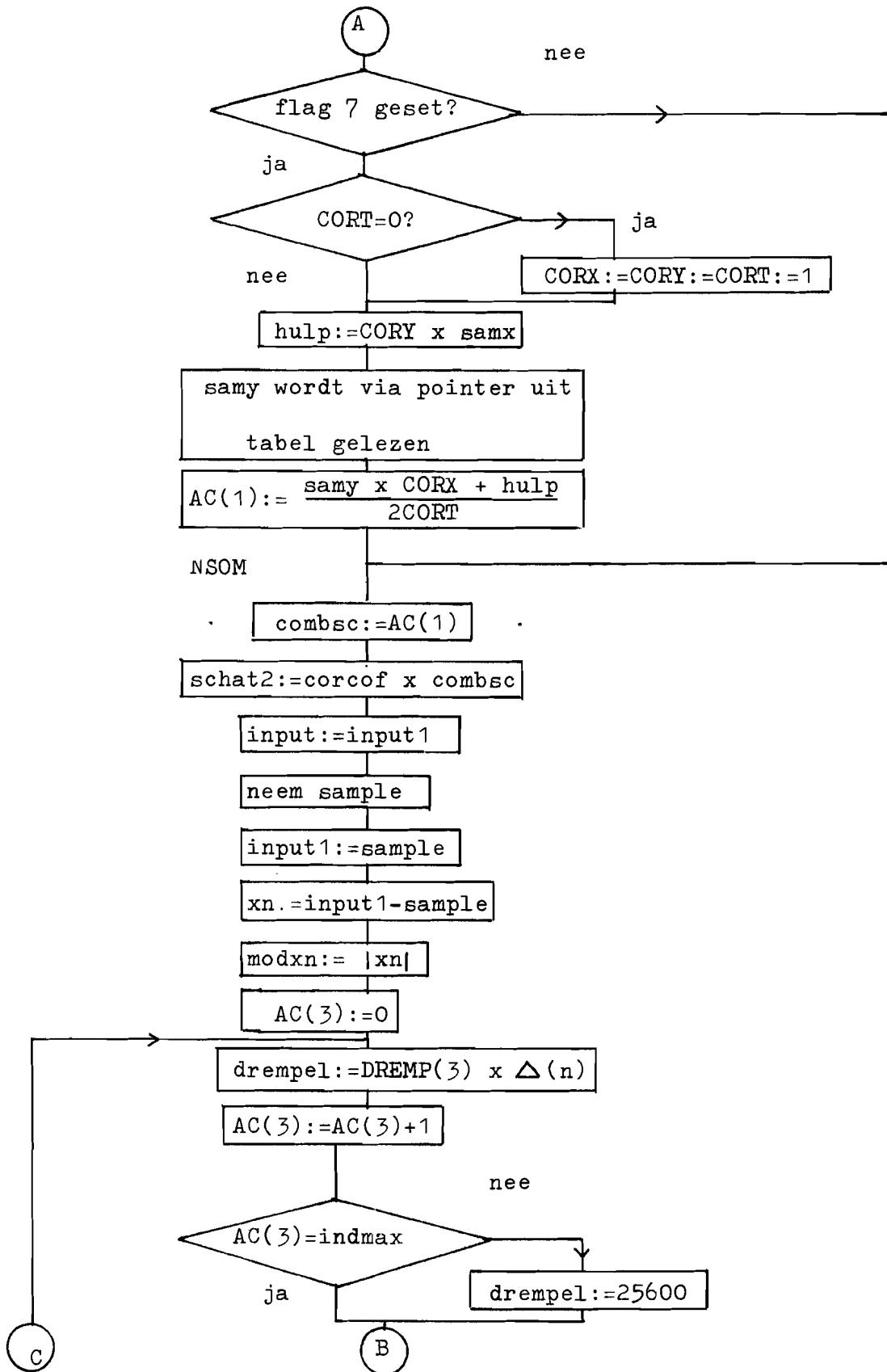
Flowchart ADPCM

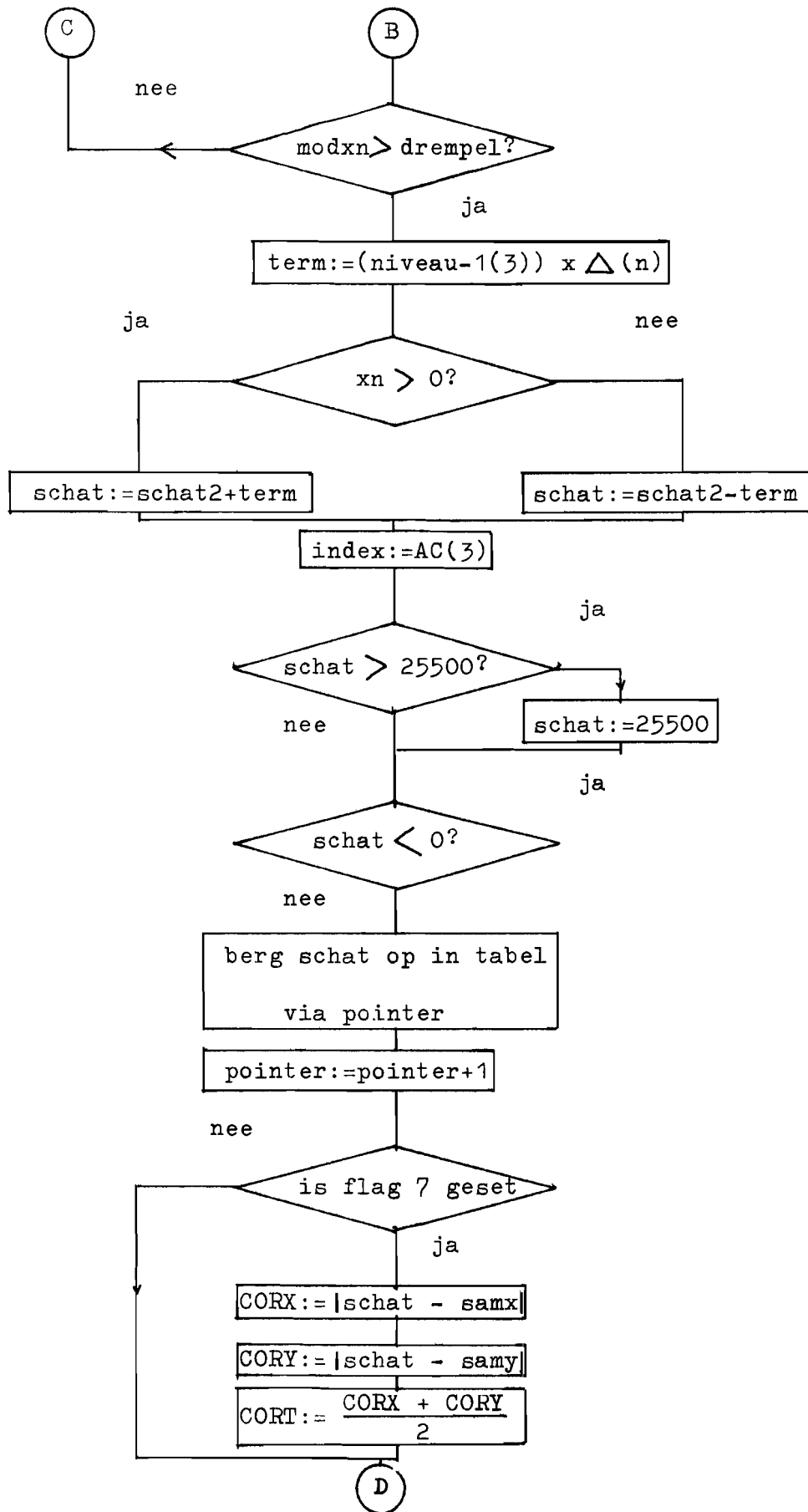


wijziging van de
flowchart voor
het systeem volgens
Mitra

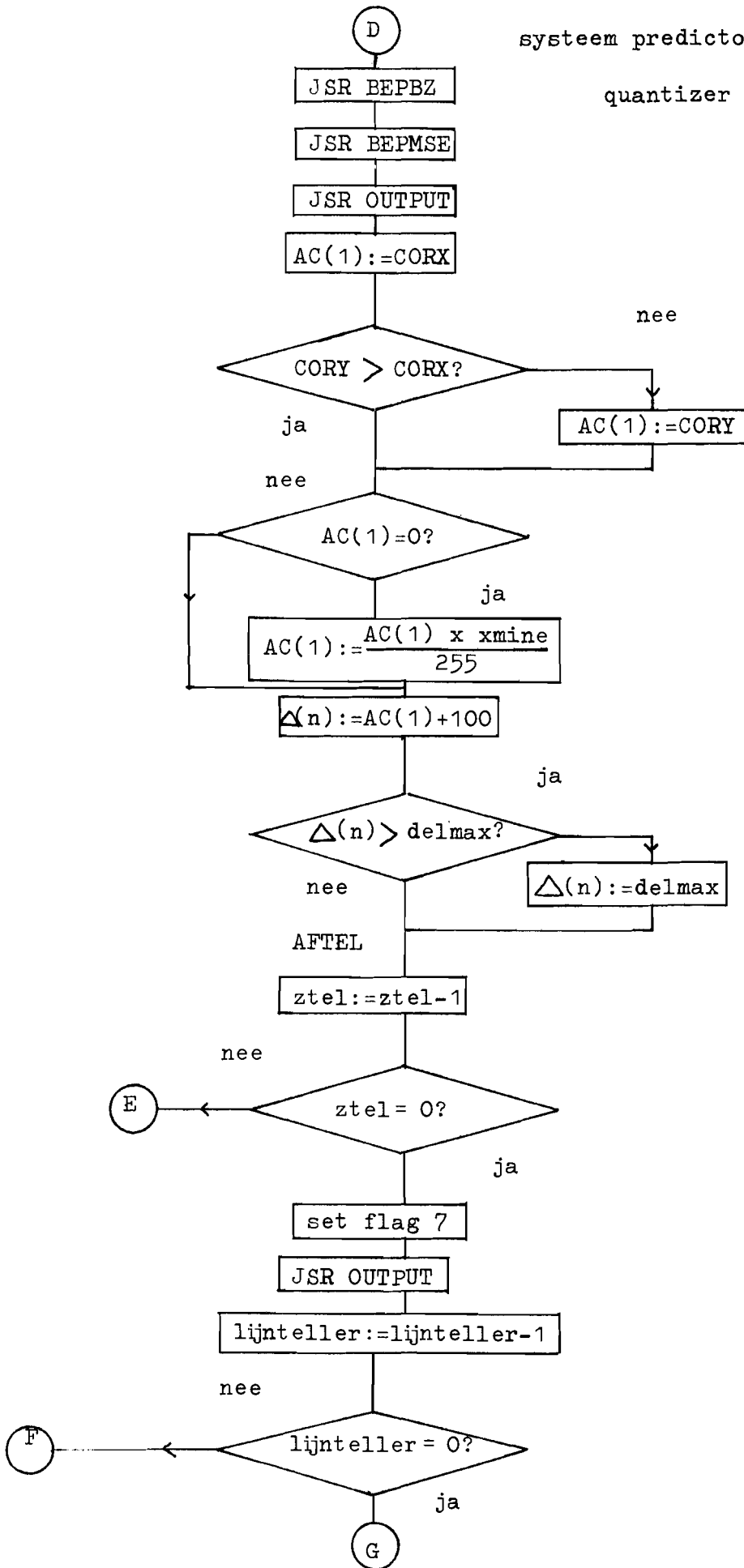


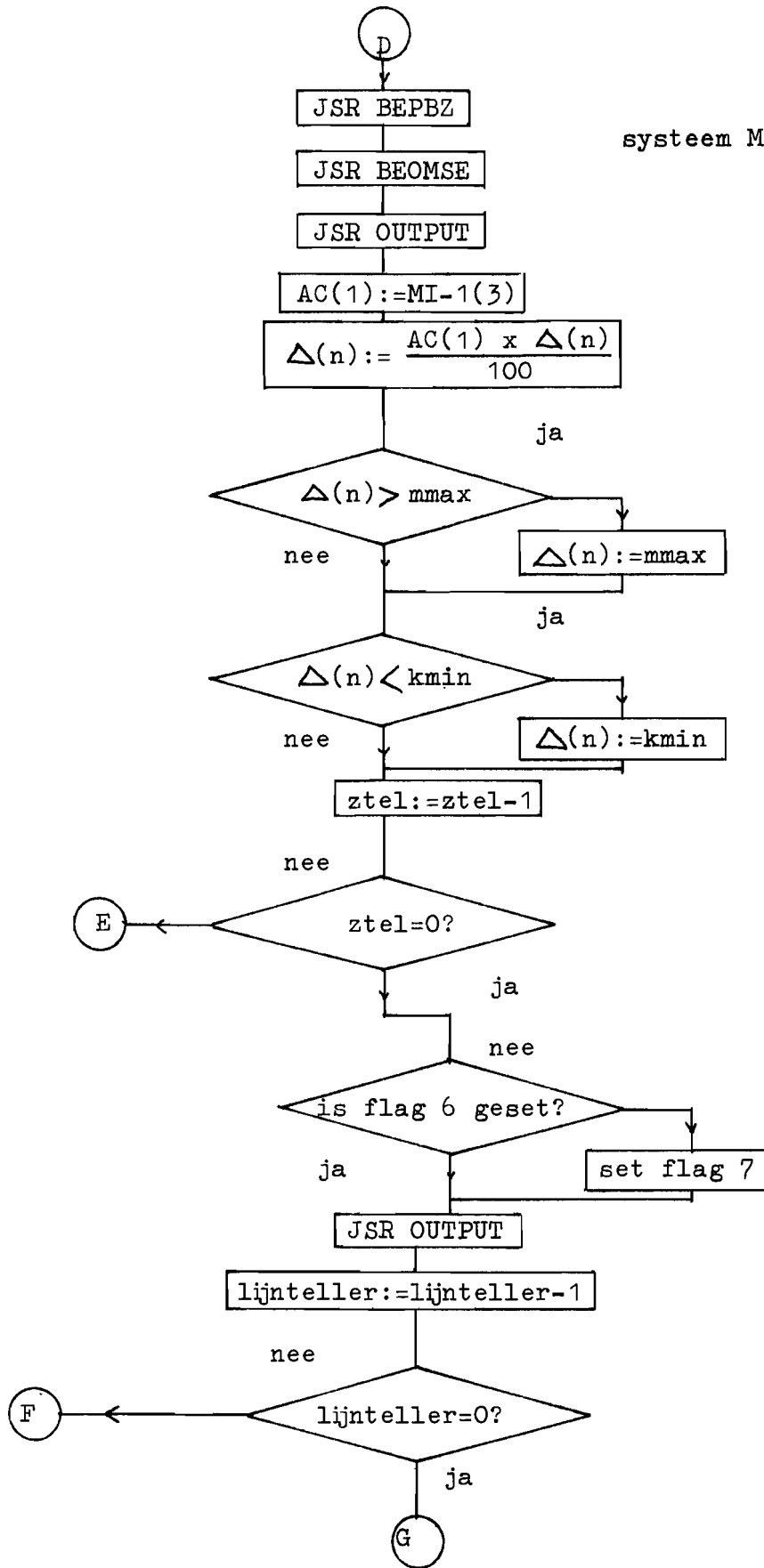
zie verder B-1



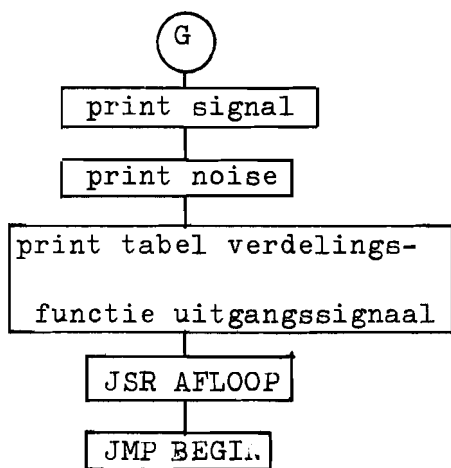


stroom predictor stuurt
quantizer





B-7



END PASS 1

1			.TITLE	ADPCM, ' PREDIC STUURT QUANT '
2			.EXTD	
3	0000		.BSECT	
4			.GLOBL	STUURS,NEEMS,AFLOOP,TIDEC,DSHR
5			.GLOBL	OUTPUT,BEFMSE,BINASC
6			.GLOBL	LEESDR,BPNIV,BEPBZ,VEEG,TARUIT,HALV
7	0002		NOISE:	.=.+2
8	0004		SIGNAL:	.=.+2
9	0018		DREMP:	.=.+20
10	002C		NIVEAU:	.=.+20
11	0040			.=.+20
12	0041		DELTAN:	.=.+1
13	0042		INDEX:	.=.+1
14	0043		TERM:	.=.+1
15	0044		XN:	.=.+1
16	0045		SCHAT:	.=.+1
17	0046		AANTAL:	.=.+1
18	0047		SOORT:	.=.+1
19	0048		INPUT:	.=.+1
20	0070			.=.+40
21	0071		XMINE:	.=.+1
22	0072		DELMAX:	.=.+1
23	0073		MODXN:	.=.+1
24	0074		CURCOF:	.=.+1
25	0075		SCHAT2:	.=.+1
26	0076		INPUT1:	.=.+1
27	0076	0000	A NUL:	.WORD 0
28	0077	0001	A EEN:	.WORD 1
29	0079		TEL1:	.=.+1
30	007A		TEL2:	.=.+1
31	007B		ZTEL:	.=.+1
32	007C		INDMAX:	.=.+1
33	007C	0032	A G50:	.WORD 50
34	007D	0064	A G100:	.WORD 100
35	007E	639C	A G25500:	.WORD 25500
36	007F	6400	A G25600:	.WORD 25600
37	0080	7EC3	A LMSG:	.WORD X'7EC3
38	0081	8000	A X8000:	.WORD X'8000
39	0082	C000	A XC000:	.WORD X'C000
40	0084		POIN:	.=.+1
41	0085		COMBSC:	.=.+1
42	0087		HULP:	.=.+2
43	0088		SAMX:	.=.+1
44	0089		SAMY:	.=.+1
45	008A		CORX:	.=.+1
46	008B		CORY:	.=.+1
47	008C		CORT:	.=.+1
48	0000		.TSECT	
49	0000	0D0A	A TEKST1:	.WORD 0D0A
50	0001	4141	A	.ASCII 'AANTAL NIVEAUS - '
	0002	4E54	A	
	0003	414C	A	
	0004	204E	A	
	0005	4956	A	

0005	4541	A			
0007	5553	A			
0008	202D	A			
0009	2020	A			
51	000A	0000	A	.WORD	0
52	000B	0D0A	A	TEKST4: .WORD	0D0A
53	000C	4C45	A	.ASCII	'LEAKPARAMETER - '
	000D	414B	A		
	000E	5041	A		
	000F	5241	A		
	0010	4D45	A		
	0011	5445	A		
	0012	5220	A		
	0013	2D20	A		
54	0014	0000	A	.WORD	0
55	0015	0D0A	A	TEKST5: .WORD	0D0A
56	0016	4C45	A	.ASCII	'LENGTE REEKS - '
	0017	4E47	A		
	0018	5445	A		
	0019	2052	A		
	001A	4545	A		
	001B	4B53	A		
	001C	202D	A		
	001D	2020	A		
57	001E	0000	A	.WORD	0
58	001F	0D0A	A	TEKST6: .WORD	0D0A
59	0020	4141	A	.ASCII	'AANTAL REEKSEN - '
	0021	4E54	A		
	0022	414C	A		
	0023	2052	A		
	0024	4545	A		
	0025	4B53	A		
	0026	454E	A		
	0027	202D	A		
	0028	2020	A		
60	0029	0000	A	.WORD	0
61	002A	0D0A	A	TEKSTA: .WORD	0D0A
62	002B	534F	A	.ASCII	'SOORT OUTPUT - '
	002C	4F52	A		
	002D	5420	A		
	002E	4F55	A		
	002F	5450	A		
	0030	5554	A		
	0031	202D	A		
	0032	2020	A		
63	0033	0000	A	.WORD	0
64	0034	0D0A	A	TEKSTE: .WORD	0D0A
65	0035	4445	A	.ASCII	'DELTAN MAX - '
	0036	4C54	A		
	0037	414E	A		
	0038	204D	A		
	0039	4158	A		
	003A	202D	A		
	003B	2020	A		
66	003C	0000	A	.WORD	0

67	003D	0D0A	A	TEKSTF:	.WORD	0D0A
68	003E	4144	A		.ASCII	'ADAPTIE FAKTOR - '
	003F	4150	A			
	0040	5449	A			
	0041	4520	A			
	0042	4641	A			
	0043	4B54	A			
	0044	4F52	A			
	0045	2020	A			
	0046	2020	A			
69	0047	0000	A		.WORD	0
70	0048	4D01	A	MAAKN:	LI	1,1
71	0049	A489	B		ST	1,CORX
72	004A	A48A	B		ST	1,CORY
73	004B	A48B	B		ST	1,CORT
74	004C	0200	A		RTS	
75	004D	0126	T	ADRES:	.WORD	TAB
76	004E	2C80	B	START:	JSR	@LMESG
77	004F	0000	T		.WORD	TEKST1
78	0050	2804	X		JSR	TIDEC
79	0051	A045	B		ST	0,AANTAL
80	0052	2809	X		JSR	LEESDR
81	0053	280A	X		JSR	BNIV
82	0054	2C80	B		JSR	@LMESG
83	0055	003D	T		.WORD	TEKSTF
84	0056	2804	X		JSR	TIDEC
85	0057	D077	B		SUB	0,EEN
86	0058	A070	B		ST	0,XMINE
87	0059	2C80	B		JSR	@LMESG
88	005A	000B	T		.WORD	TEKST4
89	005B	2804	X		JSR	TIDEC
90	005C	A073	B		ST	0,CORCOF
91	005D	2C80	B		JSR	@LMESG
92	005E	0015	T		.WORD	TEKST5
93	005F	2804	X		JSR	TIDEC
94	0060	A078	B		ST	0,TEL1
95	0061	7C78	B		DSZ	TEL1
96	0062	2C80	B		JSR	@LMESG
97	0063	001F	T		.WORD	TEKST6
98	0064	2804	X		JSR	TIDEC
99	0065	A079	B		ST	0,TEL2
100	0066	2C80	B		JSR	@LMESG
101	0067	002A	T		.WORD	TEKSTA
102	0068	2804	X		JSR	TIDEC
103	0069	A046	B		ST	0,SOORT
104	006A	2C80	B		JSR	@LMESG
105	006B	0034	T		.WORD	TEKSTE
106	006C	2804	X		JSR	TIDEC
107	006D	3181	A		RCFY	0,1
108	006E	0480	A		MPY	G100
	006F	007D	B			
109	0070	A471	B		ST	1,DELMAX
110	0071	280E	X		JSR	HALV
111	0072	A07B	B		ST	0,INDMAX
112	0073	8081	B		LD	0,X8000

113	0074	2802	X		JSR	NEEMS
114	0075	2801	X		JSR	STUURS
115	0076	4000	A		LI	0,0
116	0077	A000	B		ST	0,NOISE
117	0078	A001	B		ST	0,NOISE+1
118	0079	A002	B		ST	0,SIGNAL
119	007A	A003	B		ST	0,SIGNAL+1
120	007B	280C	X		JSR	VEEG
121	007C	0717	A		CLRST	7
122	007D	8478	B	LOOPA:	LD	1,TEL1
123	007E	A47A	B		ST	1,ZTEL
124	007F	85CD	A		LD	1,ADRES
125	0080	A483	B		ST	1,POIN
126	0081	29C6	A		JSR	MAAKN
127	0082	847D	B		LD	1,G100
128	0083	A440	B		ST	1,DELTAN
129	0084	2802	X		JSR	NEEMS
130	0085	A075	B		ST	0,INFUT1
131	0086	3181	A		RCPY	0,1
132	0087	0480	A		MPY	G100
	0088	007D	B			
133	0089	A444	B		ST	1,SCHAT
134	008A	8444	B	LOOP:	LD	1,SCHAT
135	008B	A487	B		ST	1,SAMX
136	008C	0747	A		SKSTF	7
137	008D	2113	A		JMP	NSOM
138	008E	808B	B		LD	0,CORT
139	008F	E076	B		SKG	0,NUL
140	0090	29B7	A		JSR	MAAKN
141	0091	0480	A		MPY	CORY
	0092	008A	B			
142	0093	A085	B		ST	0,HULP
143	0094	A486	B		ST	1,HULP+1
144	0095	9483	B		LD	1,@POIN
145	0096	A488	B		ST	1,SAMY
146	0097	0480	A		MPY	CORX
	0098	0089	B			
147	0099	04A0	A		DADD	HULP
	009A	0085	B			
148	009B	2805	X		JSR	DSHR
149	009C	0490	A		DIV	CORT
	009D	008E	B			
150	009E	E07C	B		SKG	0,G50
151	009F	2101	A		JMP	.+2
152	00A0	C477	B		ADD	1,EEN
153	00A1	A484	B	NSOM:	ST	1,COMBSC
154	00A2	8473	B		LD	1,CORCOF
155	00A3	0480	A		MPY	COMBSC
	00A4	0084	B			
156	00A5	0490	A		DIV	G100
	00A6	007D	B			
157	00A7	E07C	B		SKG	0,G50
158	00A8	2101	A		JMP	.+2
159	00A9	C477	B		ADD	1,EEN
160	00AA	A474	B		ST	1,SCHAT2

161	00A8	8075	B	LD	0, INPUT1
162	00AC	A047	B	ST	0, INFUT
163	00AD	2802	X	JSR	NEEMS
164	00AE	A075	B	ST	0, INPUT1
165	00AF	3181	A	RCPY	0, 1
166	00B0	0480	A	MPY	G100
	00B1	007D	B		
167	00B2	D474	B	SUB	1, SCHAT2
168	00B3	A443	B	ST	1, XN
169	00B4	E476	B	SKG	1, NUL
170	00B5	5101	A	CAI	1, 1
171	00B6	A472	B	ST	1, MODXN
172	00B7	4F00	A	LI	3, 0
173	00B8	8704	B	VERGL:	LD 1, DREMP(3)
174	00B9	0480	A	MPY	DELTAN
	00BA	0040	B		
175	00BB	4B01	A	AISZ	3, 1
176	00BC	FC7B	B	SKNE	3, INDMAX
177	00BD	847F	B	LD	1, G25600
178	00BE	E472	B	SKG	1, MODXN
179	00BF	21F8	A	JMP	VERGL
180	00C0	8717	B	LD	1, NIVEAU-1(3)
181	00C1	0480	A	MPY	DELTAN
	00C2	0040	B		
182	00C3	A442	B	ST	1, TERM
183	00C4	8043	B	LD	0, XN
184	00C5	E076	B	SKG	0, NUL
185	00C6	5101	A	CAI	1, 1
186	00C7	C474	B	ADD	1, SCHAT2
187	00C8	A444	B	ST	1, SCHAT
188	00C9	AC41	B	ST	3, INDEX
189	00CA	4C00	A	LI	0, 0
190	00CB	847E	B	LD	1, G25500
191	00CC	E444	B	SKG	1, SCHAT
192	00CD	A444	B	ST	1, SCHAT
193	00CE	8844	B	LD	2, SCHAT
194	00CF	E876	B	SKG	2, NUL
195	00D0	A044	B	ST	0, SCHAT
196	00D1	8044	B	LD	0, SCHAT
197	00D2	B083	B	ST	0, @POIN
198	00D3	7883	B	ISZ	POIN
199	00D4	0747	A	SKSTF	7
200	00D5	210C	A	JMP	NBEF
201	00D6	D087	B	SUB	0, SAMX
202	00D7	E076	B	SKG	0, NUL
203	00D8	5001	A	CAI	0, 1
204	00D9	A089	B	ST	0, CORX
205	00DA	8044	B	LD	0, SCHAT
206	00DB	D088	B	SUB	0, SAMY
207	00DC	E076	B	SKG	0, NUL
208	00DD	5001	A	CAI	0, 1
209	00DE	A08A	B	ST	0, CORY
210	00DF	C089	B	ADD	0, CORX
211	00E0	5CFF	A	SHR	0, 1
212	00E1	A08B	B	ST	0, CORT

```

213 00E2 2806 X NBEP: JSR BEPBZ
214 00E3 2807 X JSR BEPMSE
215 00E4 2806 X JSR OUTPUT
216 00E5 8489 B LD 1,CORX
217 00E6 808A B LD 0,CORY
218 00E7 E48A B SKG 1,CORY
219 00E8 2101 A JMP X
220 00E9 3181 A RCPY 0,1
221 00EA E476 B X: SKG 1,NUL
222 00EB 2104 A JMP GEENDN
223 00EC 0480 A MPY XMINE
    00ED 0070 B
224 00EE 0490 A DIV 6255
    00EF 010E T
225 00F0 C47D B GEENDN: ADD 1,G100
226 00F1 A440 B ST 1,DELTAN
227 00F2 E471 B SKG 1,DELMAX
228 00F3 2102 A JMP AFTEL
229 00F4 8471 B LD 1,DELMAX
230 00F5 A440 B ST 1,DELTAN
231 00F6 7C7A B AFTEL: DSZ ZTEL
232 00F7 2192 A JMP LOOP
233 00F8 0707 A SETST 7
234 00F9 2806 X JSR OUTPUT
235 00FA 7C79 B DSZ TEL2
236 00FB 2513 A JMP @LOOPJE
237 00FC 2C80 B JSR @LMESG
238 00FD 0110 T .WORD TEKST8
239 00FE 8002 B LD 0,SIGNAL
240 00FF 8403 B LD 1,SIGNAL+1
241 0100 2808 X JSR BINASC
242 0101 2C80 B JSR @LMESG
243 0102 0116 T .WORD TEKST9
244 0103 8000 B LD 0,NOISE
245 0104 8401 B LD 1,NOISE+1
246 0105 2808 X JSR BINASC
247 0106 2C80 B JSR @LMESG
248 0107 011C T .WORD TEKSTC
249 0108 280D X JSR TABUIT
250 0109 8082 B LD 0,XC000
251 010A 2801 X JSR STUURS
252 010B 2803 X JSR AFLOOP
253 010C 2500 A JMP @.+1
254 010D 004E T .WORD START
255 010E 00FF A 6255: .WORD 255
256 010F 007D T LOOPJE: .WORD LOOPA
257 0110 0D0A A TEKST8: .WORD 0D0A
258 0111 5349 A .ASCII 'SIGNAL='
    0112 474E A
    0113 414C A
    0114 3D20 A
259 0115 0000 A .WORD 0
260 0116 0D0A A TEKST9: .WORD 0D0A
261 0117 4E4F A .ASCII 'NOISE='
    0118 4953 A

```

```

0119 453D A
011A 2020 A
262 011B 0000 A      .WORD      0
263 011C 0D0A A      TEKSTC:   .WORD      0D0A
264 011D 5354 A      .ASCII    'STAPVERDELING: '
011E 4150 A
011F 5645 A
0120 5244 A
0121 454C A
0122 494E A
0123 473A A
0124 2020 A
265 0125 0000 A      .WORD      0
266      0327      TAB:      .=.+513
267      004E      .END      START

```

AANTAL	0045	B	ADRES	004D	T	AFLOOP	0003	X
AFTEL	00F6	T	BEPBZ	000B	X	BEPMSE	0007	X
BINASC	0008	X	BPNIIV	000A	X	COMBSC	0084	B
CORCDF	0073	B	CORT	008B	B	CORX	0089	B
CORY	008A	B	DELMAX	0071	B	DELTAN	0040	B
DREMP	0004	B	DSHR	0005	X	EEN	0077	B
G100	007D	B	G255	010E	T	G25500	007E	B
G25600	007F	B	G50	007C	B	GEENDN	00F0	T
HALV	000E	X	HULP	0085	B	INDEX	0041	B
INDMAX	007B	B	INPUT	0047	B	INPUT1	0075	B
LEESDR	0009	X	LMESG	0080	B	LOOP	008A	T
LOOPA	007D	T	LOOPJE	010F	T	MAAKN	0048	T
MODXN	0072	B	NBEP	00E2	T	NEEMS	0002	X
NIVEAU	0018	B	NOISE	0000	B	NSDM	00A1	T
NUL	0076	B	OUTPUT	0006	X	POIN	0083	B
SANX	0087	B	SAMY	0088	B	SCHAT	0044	B
SCHAT2	0074	B	SIGNAL	0002	B	SOORT	0046	B
START	004E	T	STUURS	0001	X	TAB	0126	T
TABUIT	000D	X	TEKST1	0000	T	TEKST4	000B	T
TEKST5	0015	T	TEKST6	001F	T	TEKST8	0110	T
TEKST9	0116	T	TEKSTA	002A	T	TEKSTC	011C	T
TEKSTE	0034	T	TEKSTF	003D	T	TEL1	0078	B
TEL2	0079	B	TERM	0042	B	TIDEC	0004	X
VEEG	000C	X	VERGL	008B	T	X	00EA	T
XE000	0081	B	XC000	0082	B	XMINE	0070	B
XN	0043	B	ZTEL	007A	B			

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=52E6

OBJECT CHECKSUM=91FD

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 0410

FINAL INPUT SECTOR HEX - 041B

FIRST OBJECT SECTOR HEX - 0419

FINAL OBJECT SECTOR HEX - 041B

NEXT ASSEMBLY

*.ASM

END PASS 1

1			.TITLE	ADPCM, ' EEN OF TWEEDIM. COR. '
2			.EXTD	
3	0000		.BSECT	
4			.GLOBL	STUURS, NEEMS, AFLOOP, TIDEC, DSHR
5			.GLOBL	OUTPUT, BEPMI, BEPMSE, BINASC
6			.GLOBL	LEESDR, BPNIV, BEPBZ, VEEG, TABUIT, HALV
7	0002	NOISE:	.=,+2	
8	0004	SIGNAL:	.=,+2	
9	0018	DREMP:	.=,+20	
10	002C	NIVEAU:	.=,+20	
11	0040	MI:	.=,+20	
12	0041	DELTAN:	.=,+1	
13	0042	INDEX:	.=,+1	
14	0043	TERM:	.=,+1	
15	0044	XN:	.=,+1	
16	0045	SCHAT:	.=,+1	
17	0046	AANTAL:	.=,+1	
18	0047	SOORT:	.=,+1	
19	0048	INFUT:	.=,+1	
20	0070		.=,+40	
21	0071	KMIN:	.=,+1	
22	0072	MMAX:	.=,+1	
23	0073	MODXN:	.=,+1	
24	0074	CORCOF:	.=,+1	
25	0075	SCHAT2:	.=,+1	
26	0076	INFUT1:	.=,+1	
27	0076	0000	A NUL:	.WORD 0
28	0077	0001	A EEN:	.WORD 1
29	0079		TEL1:	.=,+1
30	007A		TEL2:	.=,+1
31	007B		ZTEL:	.=,+1
32	007C		INDMAX:	.=,+1
33	007C	0032	A G50:	.WORD 50
34	007D	0064	A G100:	.WORD 100
35	007E	639C	A G25500:	.WORD 25500
36	007F	6400	A G25600:	.WORD 25600
37	0080	7EC3	A LMESG:	.WORD X'7EC3
38	0081	7E73	A LGECO:	.WORD X'7E73
39	0082	8000	A X8000:	.WORD X'8000
40	0083	C000	A XC000:	.WORD X'C000
41	0085		FOIN:	.=,+1
42	0086		COMBSC:	.=,+1
43	0088		HULP:	.=,+2
44	0089		SAMX:	.=,+1
45	008A		SAMY:	.=,+1
46	008B		CDRX:	.=,+1
47	008C		CORY:	.=,+1
48	008D		CORT:	.=,+1
49	0000		.TSECT	
50	0000	0D0A	A TEKST1:	.WORD 0D0A
51	0001	4141	A	.ASCII 'AANTAL NIVEAUS - '
	0002	4E54	A	
	0003	414C	A	
	0004	204E	A	

```

0005 4956 A
0006 4541 A
0007 5553 A
0008 202D A
0009 2020 A
52 000A 0000 A      .WORD      0
53 000B 0D0A A      TEKST4:   .WORD      0D0A
54 000C 4C45 A      .ASCII    'LEAKPARAMETER - '
      000D 414B A
      000E 5041 A
      000F 5241 A
      0010 4D45 A
      0011 5445 A
      0012 5220 A
      0013 2D20 A
55 0014 0000 A      .WORD      0
56 0015 0D0A A      TEKST5:   .WORD      0D0A
57 0016 4C45 A      .ASCII    'LENGTE REEKS - '
      0017 4E47 A
      0018 5445 A
      0019 2052 A
      001A 4545 A
      001B 4B53 A
      001C 202D A
      001D 2020 A
58 001E 0000 A      .WORD      0
59 001F 0D0A A      TEKST6:   .WORD      0D0A
60 0020 4141 A      .ASCII    'AANTAL REEKEN - '
      0021 4E54 A
      0022 414C A
      0023 2052 A
      0024 4545 A
      0025 4B53 A
      0026 454E A
      0027 202D A
      0028 2020 A
61 0029 0000 A      .WORD      0
62 002A 0D0A A      TEKSTA:   .WORD      0D0A
63 002B 534F A      .ASCII    'SOORT OUTPUT - '
      002C 4F52 A
      002D 5420 A
      002E 4F55 A
      002F 5450 A
      0030 5554 A
      0031 202D A
      0032 2020 A
64 0033 0000 A      .WORD      0
65 0034 0D0A A      TEKSTB:   .WORD      0D0A
66 0035 4445 A      .ASCII    'DELTAN MIN - '
      0036 4C54 A
      0037 414E A
      0038 204D A
      0039 494E A
      003A 202D A
      003B 2020 A

```

67	003C	0000	A		.WORD	0
68	003D	0D0A	A	TEKSTD:	.WORD	0D0A
69	003E	4445	A		.ASCII	'DELTA MAX - '
	003F	4C54	A			
	0040	414E	A			
	0041	204D	A			
	0042	4158	A			
	0043	202D	A			
	0044	2020	A			
70	0045	0000	A		.WORD	0
71	0046	0D0A	A	TEKSTE:	.WORD	0D0A
72	0047	5457	A		.ASCII	'TWEEDIM. COR.? (Y OR N) - '
	0048	4545	A			
	0049	4449	A			
	004A	4D2E	A			
	004B	2043	A			
	004C	4F52	A			
	004D	2E3F	A			
	004E	2028	A			
	004F	5920	A			
	0050	4F52	A			
	0051	204E	A			
	0052	2920	A			
	0053	2D20	A			
73	0054	0000	A		.WORD	0
74	0055	0059	A	X59:	.WORD	X'59
75	0056	4D01	A	MAAKN:	LI	1,1
76	0057	A48A	B		ST	1,CORX
77	0058	A48B	B		ST	1,CORY
78	0059	A48C	B		ST	1,CORT
79	005A	0200	A		RTS	
80	005B	0135	T	ADRES:	.WORD	TAB
81	005C	2C80	B	START:	JSR	@LMESG
82	005D	0000	T		.WORD	TEKST1
83	005E	2804	X		JSR	TIDEC
84	005F	A045	B		ST	0,AANTAL
85	0060	280A	X		JSR	LEESDR
86	0061	280B	X		JSR	BFNIV
87	0062	2C80	B		JSR	@LMESG
88	0063	0034	T		.WORD	TEKSTB
89	0064	2804	X		JSR	TIDEC
90	0065	A070	B		ST	0,KMIN
91	0066	2C80	B		JSR	@LMESG
92	0067	003D	T		.WORD	TEKSTD
93	0068	2804	X		JSR	TIDEC
94	0069	A071	B		ST	0,MMAX
95	006A	2807	X		JSR	BEFMI
96	006B	2C80	B		JSR	@LMESG
97	006C	000B	T		.WORD	TEKST4
98	006D	2804	X		JSR	TIDEC
99	006E	A073	B		ST	0,CORCDF
100	006F	2C80	B		JSR	@LMESG
101	0070	0015	T		.WORD	TEKST5
102	0071	2804	X		JSR	TIDEC
103	0072	A078	B		ST	0,TEL1

104	0073	7C78	B	DSZ	TEL1
105	0074	2C80	B	JSR	@LMESG
106	0075	001F	T	.WORD	TEKST6
107	0076	2804	X	JSR	TIDEC
108	0077	A079	B	ST	0,TEL2
109	0078	2C80	B	JSR	@LMESG
110	0079	002A	T	.WORD	TEKSTA
111	007A	2804	X	JSR	TIDEC
112	007B	A046	B	ST	0,SOORT
113	007C	0706	A	SETST	6
114	007D	2C80	B	JSR	@LMESG
115	007E	0046	T	.WORD	TEKSTE
116	007F	2C81	B	JSR	@LGECO
117	0080	F1D4	A	SKNE	0,X59
118	0081	0716	A	CLRST	6
119	0082	280F	X	JSR	HALV
120	0083	A07B	B	ST	0,INDMAX
121	0084	8082	B	LD	0,X8000
122	0085	2802	X	JSR	NEEMS
123	0086	2801	X	JSR	STUURS
124	0087	4C00	A	LI	0,0
125	0088	A000	B	ST	0,NOISE
126	0089	A001	B	ST	0,NOISE+1
127	008A	A002	B	ST	0,SIGNAL
128	008B	A003	B	ST	0,SIGNAL+1
129	008C	280D	X	JSR	VEEG
130	008D	0717	A	CLRST	7
131	008E	8478	B	LOOPA: LD	1,TEL1
132	008F	A47A	B	ST	1,ZTEL
133	0090	85CA	A	LD	1,ADRES
134	0091	A484	B	ST	1,POIN
135	0092	29C3	A	JSR	MAAKN
136	0093	847D	B	LD	1,G100
137	0094	A440	B	ST	1,DELTAN
138	0095	2802	X	JSR	NEEMS
139	0096	A075	B	ST	0,INPUT1
140	0097	3181	A	RCPY	0,1
141	0098	0480	A	MPY	G100
	0099	007D	B		
142	009A	A444	B	ST	1,SCHAT
143	009B	8444	B	LOOP: LD	1,SCHAT
144	009C	A488	B	ST	1,SAMX
145	009D	0747	A	SKSTF	7
146	009E	2113	A	JMP	NSOM
147	009F	808C	B	LD	0,CORT
148	00A0	E076	B	SKG	0,NUL
149	00A1	29B4	A	JSR	MAAKN
150	00A2	0480	A	MPY	CORY
	00A3	008B	B		
151	00A4	A086	B	ST	0,HULP
152	00A5	A487	B	ST	1,HULP+1
153	00A6	9484	B	LD	1,@POIN
154	00A7	A489	B	ST	1,SAMY
155	00A8	0480	A	MPY	CORX
	00A9	008A	B		

156	00AA	04A0	A		DADD	HULF
	00AB	0086	B			
157	00AC	2805	X		JSR	DSHR
158	00AD	0490	A		DIV	CORT
	00AE	008C	B			
159	00AF	E07C	B		SKG	0,G50
160	00B0	2101	A		JMP	.+2
161	00B1	C477	B		ADD	1,EEN
162	00B2	A485	B	NSOM:	ST	1,COMBSC
163	00B3	8473	B		LD	1,CORCOF
164	00B4	0480	A		MPY	COMBSC
	00B5	0085	B			
165	00B6	0490	A		DIV	G100
	00B7	007D	B			
166	00B8	E07C	B		SKG	0,G50
167	00B9	2101	A		JMP	.+2
168	00BA	C477	B		ADD	1,EEN
169	00BB	A474	B		ST	1,SCHAT2
170	00BC	8075	B		LD	0,INPUT1
171	00BD	A047	B		ST	0,INPUT
172	00BE	2802	X		JSR	NEEMS
173	00BF	A075	B		ST	0,INPUT1
174	00C0	3181	A		RCFY	0,1
175	00C1	0480	A		MPY	G100
	00C2	007D	B			
176	00C3	D474	B		SUB	1,SCHAT2
177	00C4	A443	B		ST	1,XN
178	00C5	E476	B		SKG	1,NUL
179	00C6	5101	A		CAI	1,1
180	00C7	A472	B		ST	1,MODXN
181	00C8	4F00	A		LI	3,0
182	00C9	8704	B	VERGL:	LD	1,DREMP(3)
183	00CA	0480	A		MPY	DELTAN
	00CB	0040	B			
184	00CC	4B01	A		AISZ	3,1
185	00CD	FC7B	B		SKNE	3,INDMAX
186	00CE	847F	B		LD	1,G25600
187	00CF	E472	B		SKG	1,MODXN
188	00D0	21F8	A		JMP	VERGL
189	00D1	8717	B		LD	1,NIVEAU-1(3)
190	00D2	0480	A		MPY	DELTAN
	00D3	0040	B			
191	00D4	A442	B		ST	1,TERM
192	00D5	8043	B		LD	0,XN
193	00D6	E076	B		SKG	0,NUL
194	00D7	5101	A		CAI	1,1
195	00D8	C474	B		ADD	1,SCHAT2
196	00D9	A444	B		ST	1,SCHAT
197	00DA	AC41	B		ST	3,INDEX
198	00DB	4C00	A		LI	0,0
199	00DC	847E	B		LD	1,G25500
200	00DD	E444	B		SKG	1,SCHAT
201	00DE	A444	B		ST	1,SCHAT
202	00DF	8844	B		LD	2,SCHAT
203	00E0	E876	B		SKG	2,NUL

204	00E1	A044	B	ST	0,SCHAT
205	00E2	8044	B	LD	0,SCHAT
206	00E3	B084	B	ST	0,@POIN
207	00E4	7884	B	ISZ	POIN
208	00E5	0747	A	SKSTF	7
209	00E6	210C	A	JMP	NBEP
210	00E7	D088	B	SUB	0,SAMX
211	00E8	E076	B	SKG	0,NUL
212	00E9	5001	A	CAI	0,1
213	00EA	A08A	B	ST	0,CORX
214	00EB	8044	B	LD	0,SCHAT
215	00EC	D089	B	SUB	0,SAMY
216	00ED	E076	B	SKG	0,NUL
217	00EE	5001	A	CAI	0,1
218	00EF	A08B	B	ST	0,CORY
219	00F0	C08A	B	ADD	0,CORX
220	00F1	5CFF	A	SHR	0,1
221	00F2	A08C	B	ST	0,CORT
222	00F3	280C	X	JSR	BEPBZ
223	00F4	2808	X	JSR	BEPMSE
224	00F5	2806	X	JSR	OUTPUT
225	00F6	872B	B	LD	1,MI-1(3)
226	00F7	0480	A	MPY	DELTAN
	00F8	0040	B		
227	00F9	0490	A	DIV	G100
	00FA	007D	B		
228	00FB	E07C	B	SKG	0,G50
229	00FC	2101	A	JMP	.+2
230	00FD	C477	B	ADD	1,EEN
231	00FE	A440	B	ST	1,DELTAN
232	00FF	8870	B	LD	2,KMIN
233	0100	8071	B	LD	0,MMAX
234	0101	E470	B	SKG	1,KMIN
235	0102	A840	B	ST	2,DELTAN
236	0103	E040	B	SKG	0,DELTAN
237	0104	A040	B	ST	0,DELTAN
238	0105	7C7A	B	DSZ	ZTEL
239	0106	2194	A	JMP	LOOP
240	0107	0746	A	SKSTF	6
241	0108	0707	A	SETST	7
242	0109	2806	X	JSR	OUTPUT
243	010A	7C79	B	DSZ	TEL2
244	010B	2512	A	JMP	@LOOPJE
245	010C	2C80	B	JSR	@LMESG
246	010D	011F	T	.WORD	TEKST8
247	010E	8002	B	LD	0,SIGNAL
248	010F	8403	B	LD	1,SIGNAL+1
249	0110	2809	X	JSR	BINASC
250	0111	2C80	B	JSR	@LMESG
251	0112	0125	T	.WORD	TEKST9
252	0113	8000	B	LD	0,NOISE
253	0114	8401	B	LD	1,NOISE+1
254	0115	2809	X	JSR	BINASC
255	0116	2C80	B	JSR	@LMESG
256	0117	012B	T	.WORD	TEKSTC

NBEP:

```

257 0118 280E X      JSR      TABUIT
258 0119 8083 B      LD       0,XC000
259 011A 2801 X      JSR      STUURS
260 011B 2803 X      JSR      AFLLOOP
261 011C 2500 A      JMP      @.+1
262 011D 005C T      .WORD   START
263 011E 008E T      LOOPJE: .WORD   LOOPFA
264 011F 0D0A A      TEKST8: .WORD   0D0A
265 0120 5349 A      .ASCII  'SIGNAL='
        0121 474E A
        0122 414C A
        0123 3D20 A
266 0124 0000 A      .WORD   0
267 0125 0D0A A      TEKST9: .WORD   0D0A
268 0126 4E4F A      .ASCII  'NOISE='
        0127 4953 A
        0128 453D A
        0129 2020 A
269 012A 0000 A      .WORD   0
270 012B 0D0A A      TEKSTC: .WORD   0D0A
271 012C 5354 A      .ASCII  'STAFVERDELING:'
        012D 4150 A
        012E 5645 A
        012F 5244 A
        0130 454C A
        0131 494E A
        0132 473A A
        0133 2020 A
272 0134 0000 A      .WORD   0
273          0336      TAB:     .=.+513
274          005C      .END    START
    
```

AANTAL	0045	B	ADRES	005B	T	AFLLOOP	0003	X
BEFBZ	000C	X	BEPMI	0007	X	BEPMSE	0008	X
BINASC	0009	X	BPNIIV	000B	X	COMBSC	0085	B
CORCOF	0073	B	CORT	008C	B	CORX	008A	B
CORY	008B	B	DELTAN	0040	B	DREMP	0004	B
DSHR	0005	X	EEN	0077	B	G100	007D	B
G25500	007E	B	G25600	007F	B	G50	007C	B
HALV	000F	X	HULF	0086	B	INDEX	0041	B
INDMAX	007B	B	INPUT	0047	B	INPUT1	0075	B
KNIN	0070	B	LEESDR	000A	X	LGECO	0081	B
LMESG	0080	B	LOOP	009B	T	LOOPFA	008E	T
LOOPJE	011E	T	MAAKN	0056	T	MI	002C	B
MMAX	0071	B	MODXN	0072	B	NBEP	00F3	T
NEEMS	0002	X	NIVEAU	0018	B	NOISE	0000	B
NSOM	00B2	T	NUL	0076	B	OUTPUT	0006	X
POIN	0084	B	SAMX	0088	B	SAMY	0089	B
SCHAT	0044	B	SCHAT2	0074	B	SIGNAL	0002	B
SOORT	0046	B	START	005C	T	STUURS	0001	X
TAB	0135	T	TABUIT	000E	X	TEKST1	0000	T
TEKST4	000B	T	TEKST5	0015	T	TEKST6	001F	T
TEKST8	011F	T	TEKST9	0125	T	TEKSTA	002A	T
TEKSTB	0034	T	TEKSTC	012B	T	TEKSTD	003D	T
TEKSTE	0046	T	TEL1	0078	B	TEL2	0079	B
TERM	0042	B	TIDEC	0004	X	VEEG	000D	X

VERGL	0009	T	X59	0055	T	X8000	0082	B
XC000	0083	B	XN	0043	B	ZTEL	007A	B

NO ERROR LINES
 END PASS 4
 SOURCE CHECKSUM=72F2
 OBJECT CHECKSUM=17A9

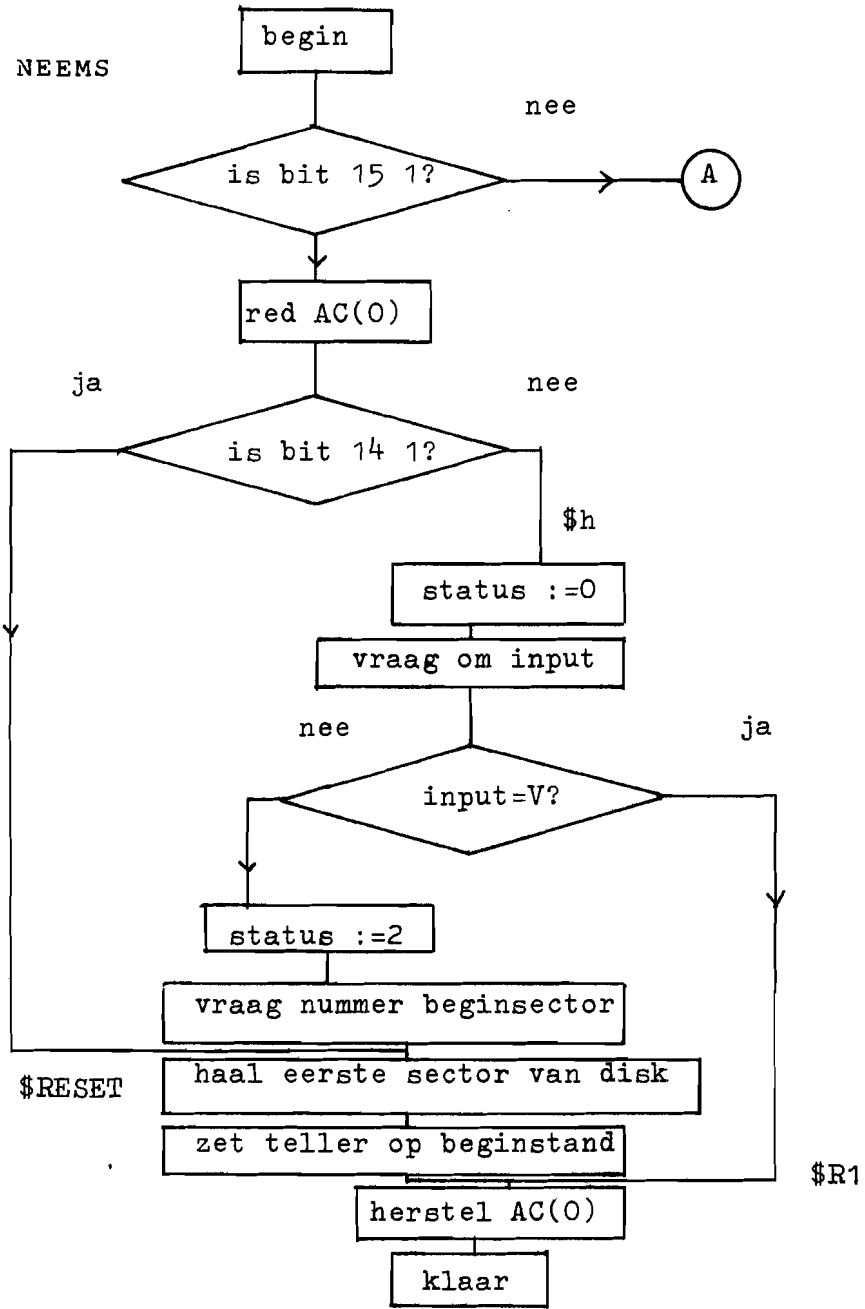
DISC SECTORS USED

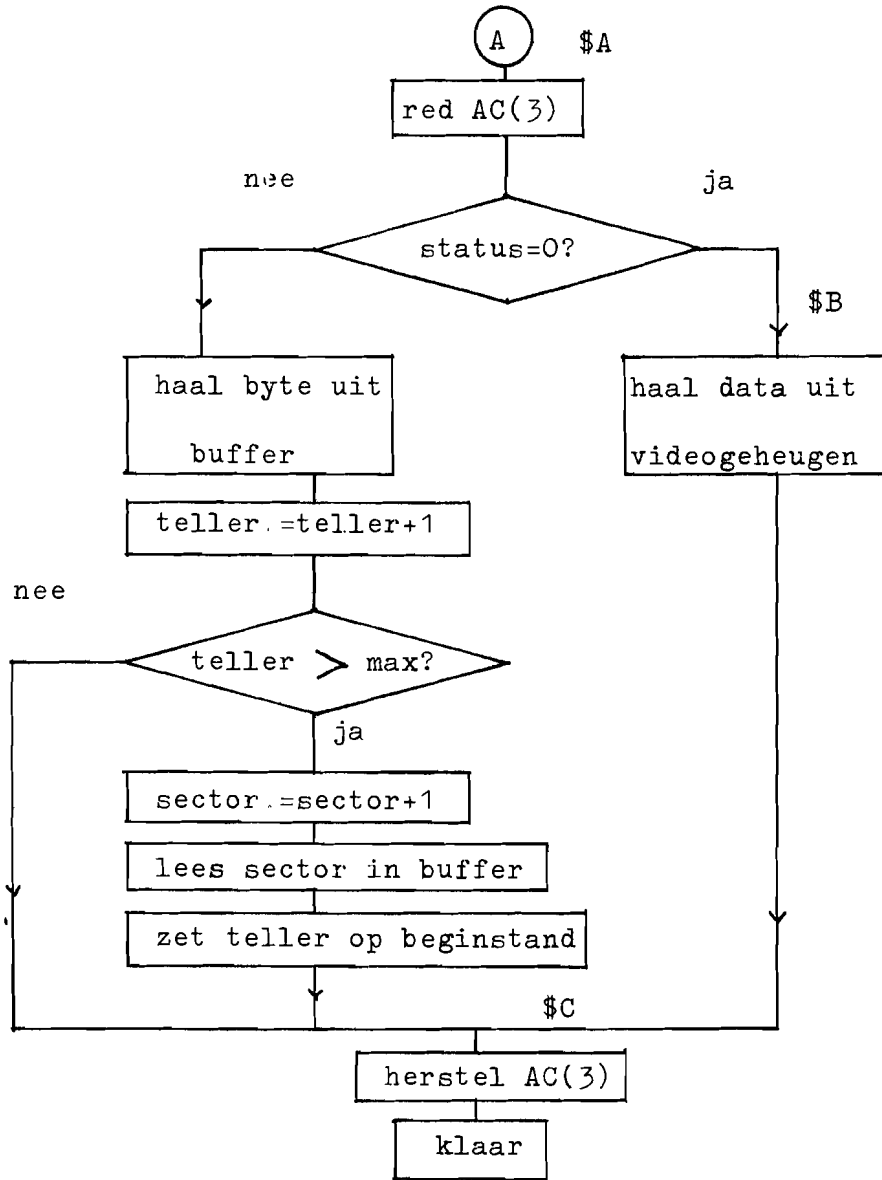
FIRST INPUT SECTOR HEX - 03D0
 FINAL INPUT SECTOR HEX - 03D8

FIRST OBJECT SECTOR HEX - 03D9
 FINAL OBJECT SECTOR HEX - 03DB

NEXT ASSEMBLY
 *.ASM

Flowchart NEEMS





DI02A0, D0

END PASS 1

1				.TITLE	INOUT, 'DISK OF VIDEOMEM '
2				.GLOBL	NEEMS, STUURS, LESEC
3				.LOCAL	
4		0000		.TSECT	
5				.EXTD	
6	0000	0D0A	A	\$1:	.WORD 0D0A
7	0001	494E	A		.ASCII 'INPUT VAN DISK OF VIDEOMEM.?'
	0002	5055	A		
	0003	5420	A		
	0004	5641	A		
	0005	4E20	A		
	0006	4449	A		
	0007	534B	A		
	0008	204F	A		
	0009	4620	A		
	000A	5649	A		
	000B	4445	A		
	000C	4F4D	A		
	000D	454D	A		
	000E	2E3F	A		
	000F	2020	A		
8	0010	0000	A		.WORD 0
9	0011	0D0A	A	\$2:	.WORD 0D0A
10	0012	4649	A		.ASCII 'FIRST INPUT SECTOR HEX - '
	0013	5253	A		
	0014	5420	A		
	0015	494E	A		
	0016	5055	A		
	0017	5420	A		
	0018	5345	A		
	0019	4354	A		
	001A	4F52	A		
	001B	2048	A		
	001C	4558	A		
	001D	202D	A		
	001E	2020	A		
11	001F	0000	A		.WORD 0
12		0021		\$HULP:	.=.+1
13	0021	075F	A	NEEMS:	SKBIT 15
14	0022	2124	A		JMP \$A
15	0023	4000	A		PUSH 0
16	0024	075E	A		SKBIT 14
17	0025	2104	A		JMP \$H
18	0026	3082	A		RXOR 0,0
19	0027	F13A	A		SKNE 0,\$STA1
20	0028	211C	A		JMP \$R1
21	0029	2114	A		JMP \$RESET
22	002A	4C00	A	\$H:	LI 0,0
23	002B	A136	A		ST 0,\$STA1
24	002C	2D43	A		JSR @\$MSG
25	002D	0000	T		.WORD \$1
26	002E	2D43	A		JSR @\$GECO
27	002F	613C	A		AND 0,\$X7F
28	0030	F13C	A		SKNE 0,\$X56

INOUT DISK OF VIDEOMEM

```

29 0031 2113 A      JMP      $R1
30 0032 8163 A      LD       0,$ADR1
31 0033 A145 A      ST       0,$TAB1
32 0034 5C01 A      SHL     0,1
33 0035 A13E A      ST       0,$BUF1
34 0036 C15E A      ADD     0,$511
35 0037 A12C A      ST       0,$MAX1
36 0038 4C02 A      LI      0,2
37 0039 A128 A      ST       0,$STA1
38 003A 2D35 A      JSR     @$MSG
39 003B 0011 T      .WORD  $2
40 003C 2944 A      JSR     LESEC
41 003D A1E2 A      ST       0,$HULP
42 003E 81E1 A      $RESET: LD       0,$HULP
43 003F A138 A      ST       0,$SEC1
44 0040 2D35 A      JSR     @$DISK
45 0041 0077 T      .WORD  $LEES
46 0042 21FD A      JMP     .-2
47 0043 8130 A      LD       0,$BUF1
48 0044 A121 A      ST       0,$TEL1
49 0045 4400 A      $R1:   PULL   0
50 0046 0200 A      RTS
51 0047 4300 A      $A:   PUSH   3
52 0048 3F82 A      RXOR   3,3
53 0049 FD18 A      SKNE   3,$STA1
54 004A 210E A      JMP     $B
55 004B 8D1A A      LD      3,$TEL1
56 004C 07C0 A      LDB    (3)
      004D 0000 A
57 004E 4B01 A      AISZ   3,1
58 004F AD16 A      ST     3,$TEL1
59 0050 ED13 A      SKG    3,$MAX1
60 0051 210E A      JMP     $C
61 0052 7925 A      ISZ   $SEC1
62 0053 2D22 A      JSR   @$DISK
63 0054 0077 T      .WORD  $LEES
64 0055 21FD A      JMP     .-2
65 0056 8D1D A      LD     3,$BUF1
66 0057 ADOE A      ST     3,$TEL1
67 0058 2107 A      JMP     $C
68 0059 0C00 A      $B:   SFLG   4
69 005A 0D00 A      SFLG   5
70 005B 0E80 A      PFLG   6
71 005C 4F48 A      LI     3,X'48
72 005D 1F01 A      BOC   15,+.2
73 005E 21FE A      JMP     .-1
74 005F 0400 A      RIN
75 0060 4700 A      $C:   PULL   3
76 0061 0200 A      RTS
77      0063      $STA1:  .,+.1
78      0064      $STA2:  .,+.1
79      0065      $MAX1:  .,+.1
80      0066      $MAX2:  .,+.1
81      0067      $TEL1:  .,+.1
82      0068      $TEL2:  .,+.1

```

```

83      0069      $VORG:  .=. +1
84 0069 0020 A  $X20:  .WORD  X'20
85 006A 0040 A  $X40:  .WORD  X'40
86 006B 000F A  $XF:   .WORD  X'F
87 006C 007F A  $X7F:  .WORD  X'7F
88 006D 0056 A  $X56:  .WORD  X'56
89 006E 0037 A  $X37:  .WORD  X'37
90 006F 0030 A  $X30:  .WORD  X'30
91 0070 7EC3 A  $MMSG:  .WORD  X'7EC3
92 0071 7E59 A  $PUTC:  .WORD  X'7E59
93 0072 7E73 A  $GECO:  .WORD  X'7E73
94      0074      $BUF2:  .=. +1
95      0075      $BUF1:  .=. +1
96 0075 0009 A  $9:    .WORD  9
97 0076 C008 A  $DISK:  .WORD  X'C008
98 0077 0002 A  $LEES:  .WORD  2
99 0078 0000 A  $SEC1:  .WORD  0
100     007C      $TAB1:  .=. +3
101 007C 0007 A  $SCHR:  .WORD  7
102 007D 0000 A  $SEC2:  .WORD  0
103     0081      $TAB2:  .=. +3
104 0081 4300 A  LESEC:  PUSH   3
105 0082 4F03 A  $LESE:  LI     3,3
106 0083 4C00 A          LI     0,0
107 0084 A1E3 A          ST     0,$VORG
108 0085 2DEC A  $VOLG:  JSR   @$GECO
109 0086 61E5 A          AND   0,$X7F
110 0087 F1E1 A          SKNE  0,$X20
111 0088 21F9 A          JMP   $LESE
112 0089 E1E0 A          SKG   0,$X40
113 008A 2101 A          JMP   $TTAL
114 008B C1E9 A          ADD   0,$9
115 008C 61DE A  $TTAL:  AND   0,$XF
116 008D C1DA A          ADD   0,$VORG
117 008E 4BFF A          AISZ  3,-1
118 008F 2102 A          JMP   $LUS
119 0090 4700 A          FULL  3
120 0091 0200 A          RTS
121 0092 5C04 A  $LUS:  SHL   0,4
122 0093 A1D4 A          ST     0,$VORG
123 0094 21F0 A          JMP   $VOLG
124 0095 01FF A  $511:  .WORD  511
125 0096 0120 T  $ADR1:  .WORD  $TB1
126 0097 0220 T  $ADR2:  .WORD  $TB2
127 0098 075F A  $TUURS: SKBIT  15
128 0099 211B A          JMP   $D
129 009A 4000 A          PUSH  0
130 009B 075E A          SKBIT  14
131 009C 2101 A          JMP   $G
132 009D 2132 A          JMP   $AFLP
133 009E 4C00 A  $G:    LI     0,0
134 009F A1C3 A          ST     0,$STA2
135 00A0 2DCF A          JSR   @$MMSG
136 00A1 00F1 T          .WORD  $3
137 00A2 2DCF A          JSR   @$GECO

```

138	00A3	61C8	A	AND	0,\$X7F
139	00A4	F1C8	A	SKNE	0,\$X56
140	00A5	2148	A	JMP	\$R2
141	00A6	4C02	A	LI	0,2
142	00A7	A1B8	A	ST	0,\$STA2
143	00A8	81EE	A	LD	0,\$ADR2
144	00A9	A1D4	A	ST	0,\$TAB2
145	00AA	5C01	A	SHL	0,1
146	00AB	A1C7	A	ST	0,\$BUF2
147	00AC	C1E8	A	ADD	0,\$511
148	00AD	A1B7	A	ST	0,\$MAX2
149	00AE	2DC1	A	JSR	@\$MESG
150	00AF	0102	T	.WORD	\$4
151	00B0	29D0	A	JSR	LESEC
152	00B1	A1CB	A	ST	0,\$SEC2
153	00B2	81C0	A	LD	0,\$BUF2
154	00B3	A1B3	A	ST	0,\$TEL2
155	00B4	2139	A	JMP	\$R2
156	00B5	4300	A	\$D:	PUSH 3
157	00B6	3F82	A	RXOR	3,3
158	00B7	FDAB	A	SKNE	3,\$STA2
159	00B8	210E	A	JMP	\$E
160	00B9	8DAD	A	LD	3,\$TEL2
161	00BA	07D0	A	STB	(3)
	00BB	0000	A		
162	00BC	4B01	A	AISZ	3,1
163	00BD	ADA9	A	ST	3,\$TEL2
164	00BE	EDA6	A	SKG	3,\$MAX2
165	00BF	210E	A	JMF	\$F
166	00C0	2DB5	A	JSR	@\$DISK
167	00C1	007C	T	.WORD	\$SCHR
168	00C2	21FD	A	JMF	,-2
169	00C3	8DAF	A	LD	3,\$BUF2
170	00C4	ADA2	A	ST	3,\$TEL2
171	00C5	79B7	A	ISZ	\$SEC2
172	00C6	2107	A	JMF	\$F
173	00C7	0C80	A	\$E:	PFLG 4
174	00C8	0D00	A	SFLG	5
175	00C9	0E80	A	PFLG	6
176	00CA	4F48	A	LI	3,X'48
177	00CB	1F01	A	BOC	15, +2
178	00CC	21FE	A	JMP	,-1
179	00CD	0600	A	ROUT	
180	00CE	4700	A	\$F:	PULL 3
181	00CF	0200	A	RTS	
182	00D0	4C00	A	\$AFLP:	LI 0,0
183	00D1	F191	A	SKNE	0,\$STA2
184	00D2	2118	A	JMP	\$R2
185	00D3	2D9C	A	JSR	@\$MESG
186	00D4	0111	T	.WORD	\$5
187	00D5	819D	A	LD	0,\$BUF2
188	00D6	F190	A	SKNE	0,\$TEL2
189	00D7	2104	A	JMP	\$I
190	00D8	2D9D	A	JSR	@\$DISK
191	00D9	007C	T	.WORD	\$SCHR

```

192 00DA 21FD A      JMP      .-2
193 00DB 2101 A      JMP      $J
194 00DC 7DA0 A      $I:     DSZ      $SEC2
195 00DD 4300 A      $J:     PUSH     3
196 00DE 4F03 A      LI      3,3
197 00DF 819D A      $VOLK: LD       0,$SEC2
198 00E0 5CF8 A      SHR     0,8
199 00E1 E193 A      SKG    0,$9
200 00E2 2102 A      JMP    $TIEN
201 00E3 C18A A      ADD    0,$X37
202 00E4 2101 A      JMP    $PRNT
203 00E5 C189 A      $TIEN: ADD    0,$X30
204 00E6 2D8A A      $PRNT: JSR    @ $PUTC
205 00E7 8195 A      LD     0,$SEC2
206 00E8 6107 A      AND    0,$XFF
207 00E9 5C04 A      SHL    0,4
208 00EA A192 A      ST     0,$SEC2
209 00EB 4BFF A      AISZ  3,-1
210 00EC 21F2 A      JMP    $VOLK
211 00ED 4700 A      PULL   3
212 00EE 4400 A      $R2:   PULL   0
213 00EF 0200 A      RTS
214 00F0 00FF A      $XFF: .WORD   X'FF
215 00F1 0D0A A      $3:   .WORD   0D0A
216 00F2 4F55 A      .ASCII 'OUTPUT NAAR DISK OF VIDEOMEM? '
      00F3 5450 A
      00F4 5554 A
      00F5 204E A
      00F6 4141 A
      00F7 5220 A
      00F8 4449 A
      00F9 534B A
      00FA 204F A
      00FB 4620 A
      00FC 5649 A
      00FD 4445 A
      00FE 4F4D A
      00FF 454D A
      0100 3F20 A
217 0101 0000 A      .WORD   0
218 0102 0D0A A      $4:   .WORD   0D0A
219 0103 4649 A      .ASCII 'FIRST OUTPUT SECTOR HEX - '
      0104 5253 A
      0105 5420 A
      0106 4F55 A
      0107 5450 A
      0108 5554 A
      0109 2053 A
      010A 4543 A
      010B 544F A
      010C 5220 A
      010D 4845 A
      010E 5820 A
      010F 2D20 A
220 0110 0000 A      .WORD   0
    
```

```

221 0111 0DOA A $5: .WORD 0DOA
222 0112 4649 A .ASCII 'FINAL OUTPUT SECTOR HEX - '
    0113 4E41 A
    0114 4C20 A
    0115 4F55 A
    0116 5450 A
    0117 5554 A
    0118 2053 A
    0119 4543 A
    011A 544F A
    011B 5220 A
    011C 4845 A
    011D 5820 A
    011E 2D20 A
223 011F 0000 A .WORD 0
224     0220 $TB1: .=.+X'100
225     0320 $TB2: .=.+X'100
226     .LOCAL
227     0000 .END
    
```

LESEC	0081	GT	NEEMS	0021	GT*	STUURS	0098	GT*
\$1	0000	T	\$2	0011	T	\$3	00F1	T
\$4	0102	T	\$5	0111	T	\$511	0095	T
\$9	0075	T	\$A	0047	T	\$ADR1	0096	T
\$ADR2	0097	T	\$AFLP	00D0	T	\$B	0059	T
\$BUF1	0074	T	\$RUF2	0073	T	\$C	0060	T
\$D	00B5	T	\$DISK	0076	T	\$E	00C7	T
\$F	00CE	T	\$G	009E	T	\$GECO	0072	T
\$H	002A	T	\$HULP	0020	T	\$I	00DC	T
\$J	00DD	T	\$LEES	0077	T	\$LESE	0082	T
\$LUS	0092	T	\$MAX1	0064	T	\$MAX2	0065	T
\$MSG	0070	T	\$PRNT	00E6	T	\$PUTC	0071	T
\$R1	0045	T	\$R2	00EE	T	\$RESE	003E	T
\$SCHR	007C	T	\$SEC1	0078	T	\$SEC2	007D	T
\$STA1	0062	T	\$STA2	0063	T	\$TAB1	0079	T
\$TAB2	007E	T	\$TB1	0120	T	\$TB2	0220	T
\$TEL1	0066	T	\$TEL2	0067	T	\$TIEN	00E5	T
\$TTAL	008C	T	\$VOLG	0085	T	\$VOLK	00DF	T
\$VORG	0068	T	\$X20	0069	T	\$X30	006F	T
\$X37	006E	T	\$X40	006A	T	\$X56	006D	T
\$X7F	006C	T	\$XF	006B	T	\$XFF	00F0	T

```

NO ERROR LINES
END PASS 4
SOURCE CHECKSUM=5FB8
OBJECT CHECKSUM=4810
    
```

DISC SECTORS USED

```

FIRST INPUT SECTOR HEX - 02A0
FINAL INPUT SECTOR HEX - 02A7
    
```

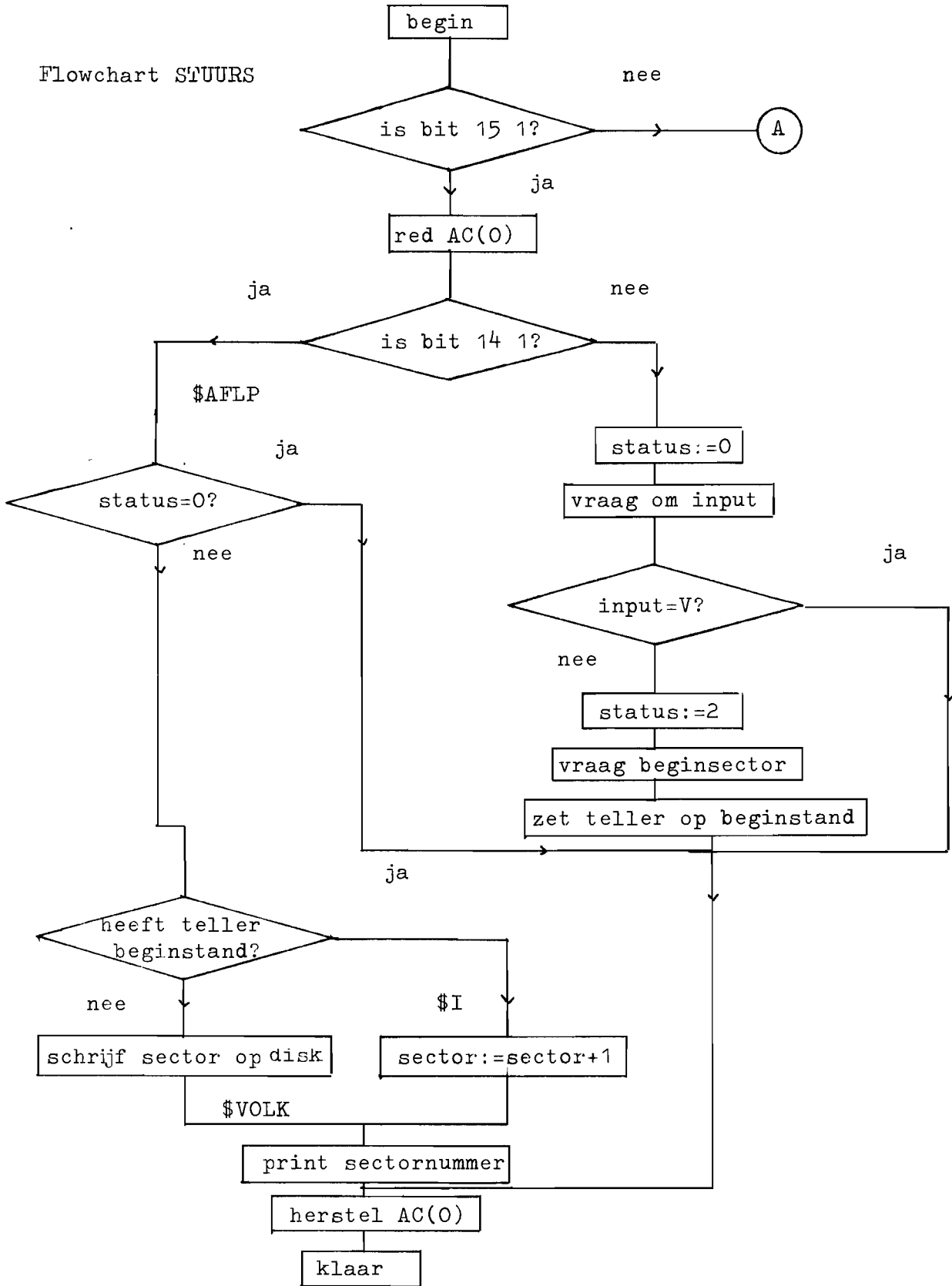
```

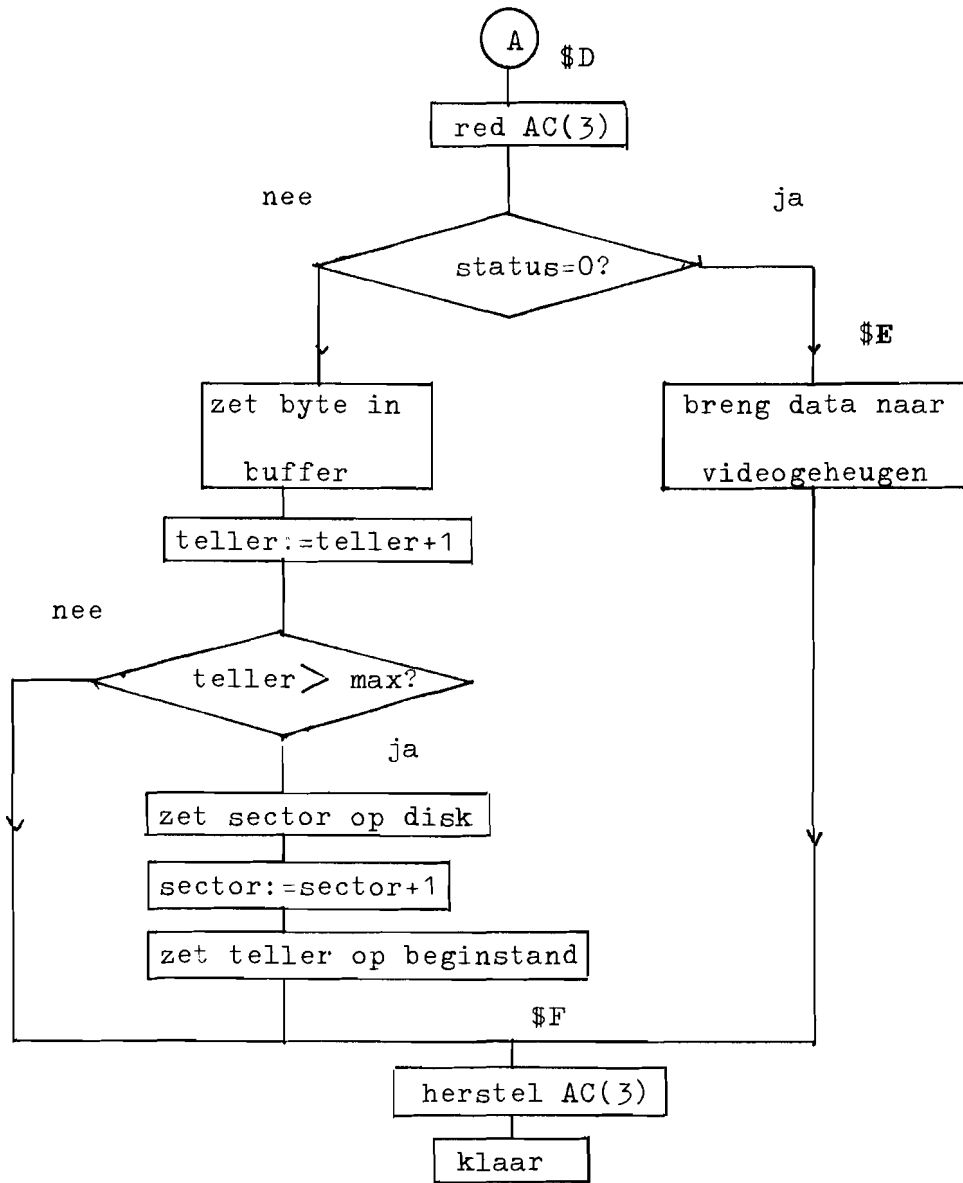
FIRST OBJECT SECTOR HEX - 02A8
FINAL OBJECT SECTOR HEX - 02A9
    
```

```

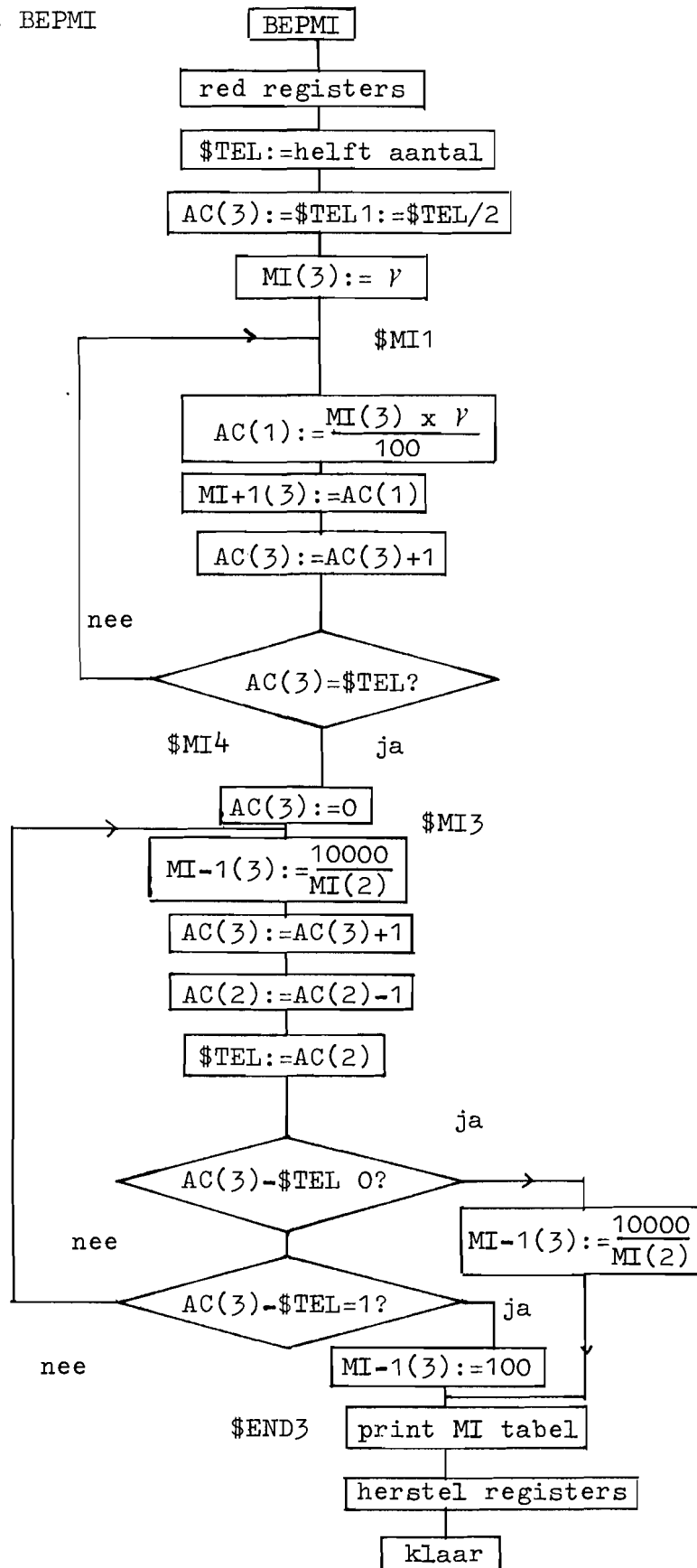
NEXT ASSEMBLY
*.ASM
    
```


Flowchart STUURS

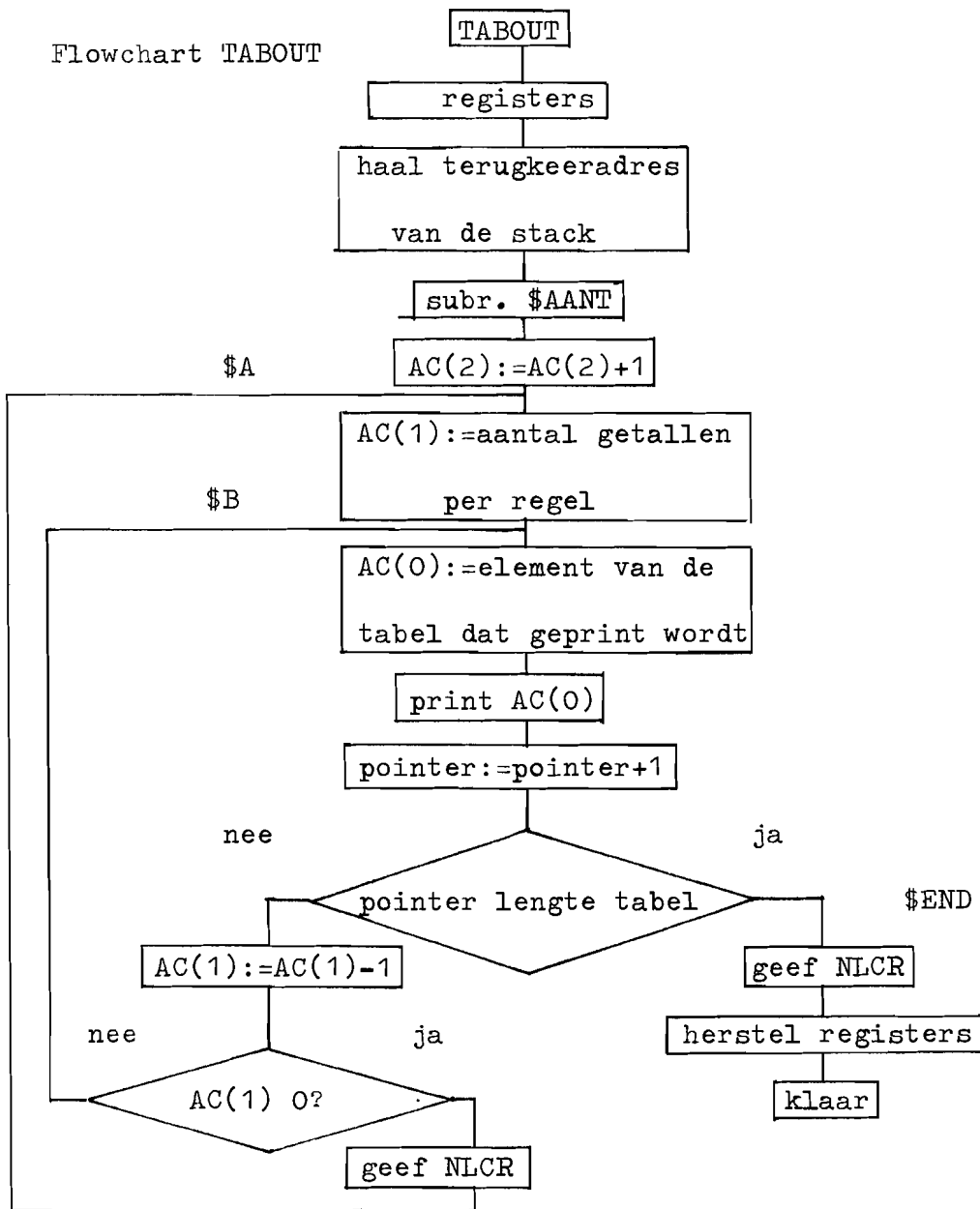




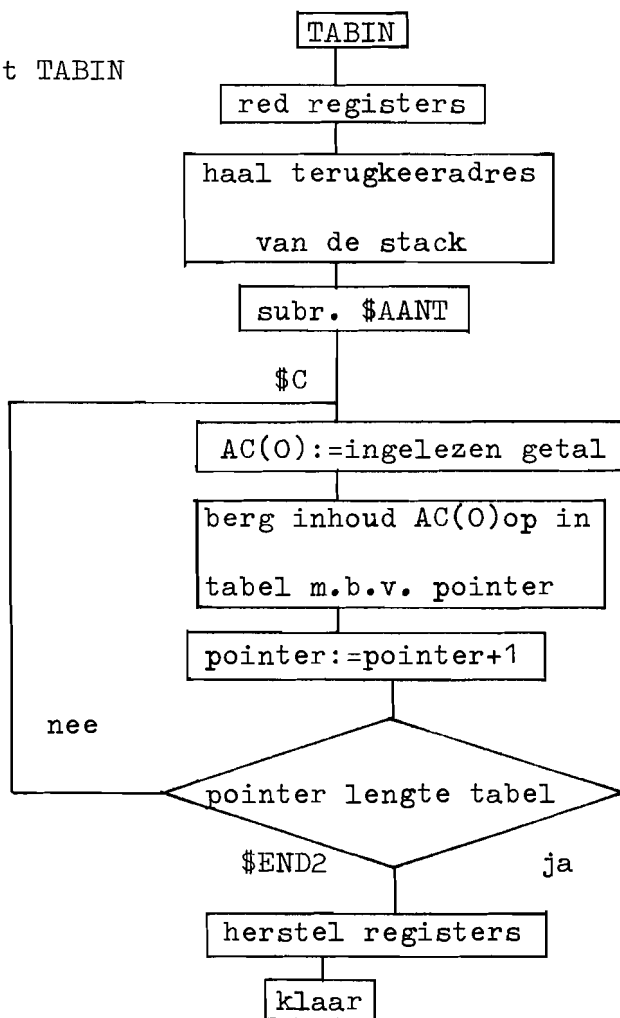
Flowchart BEPMI



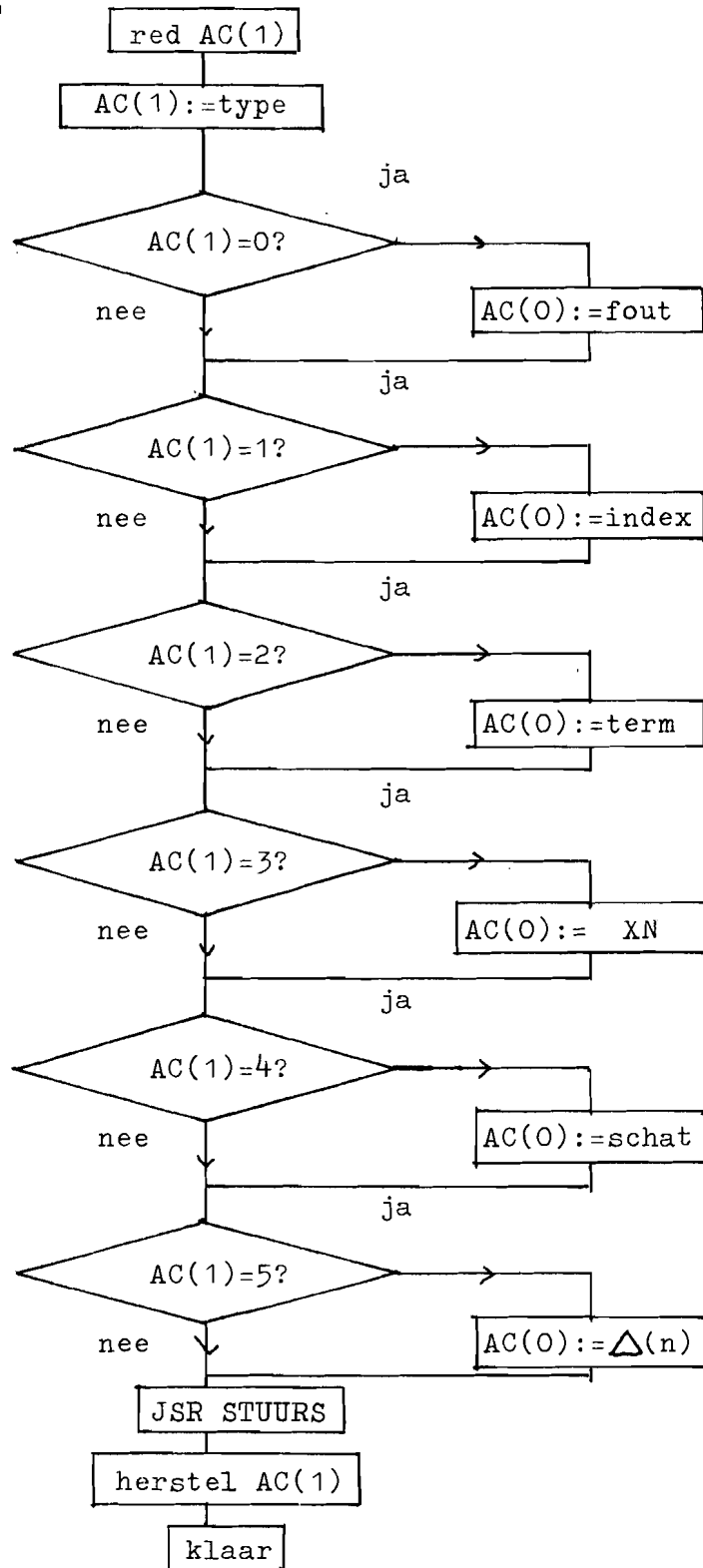
Flowchart TABOUT



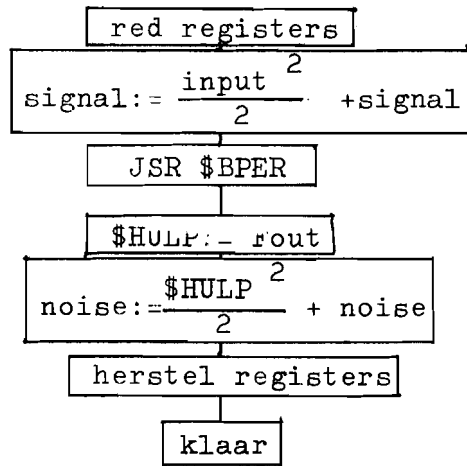
Flowchart TABIN



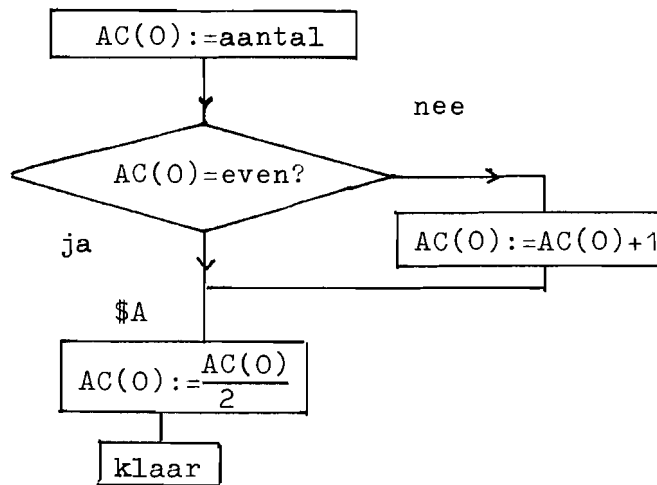
Flowchart OUTPUT



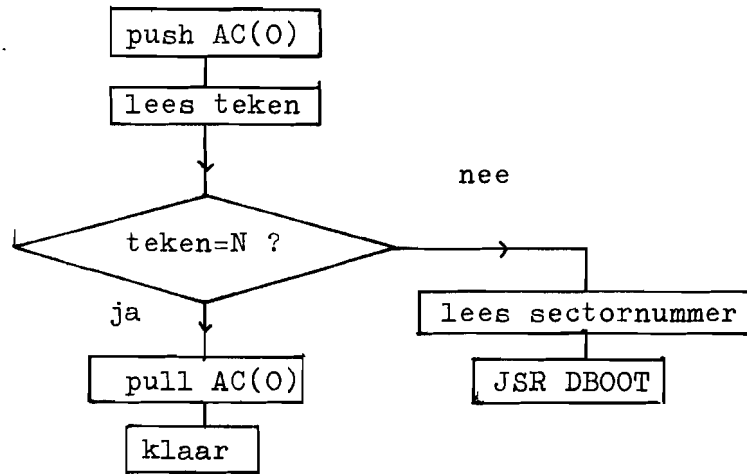
Flowchart BEPMSE



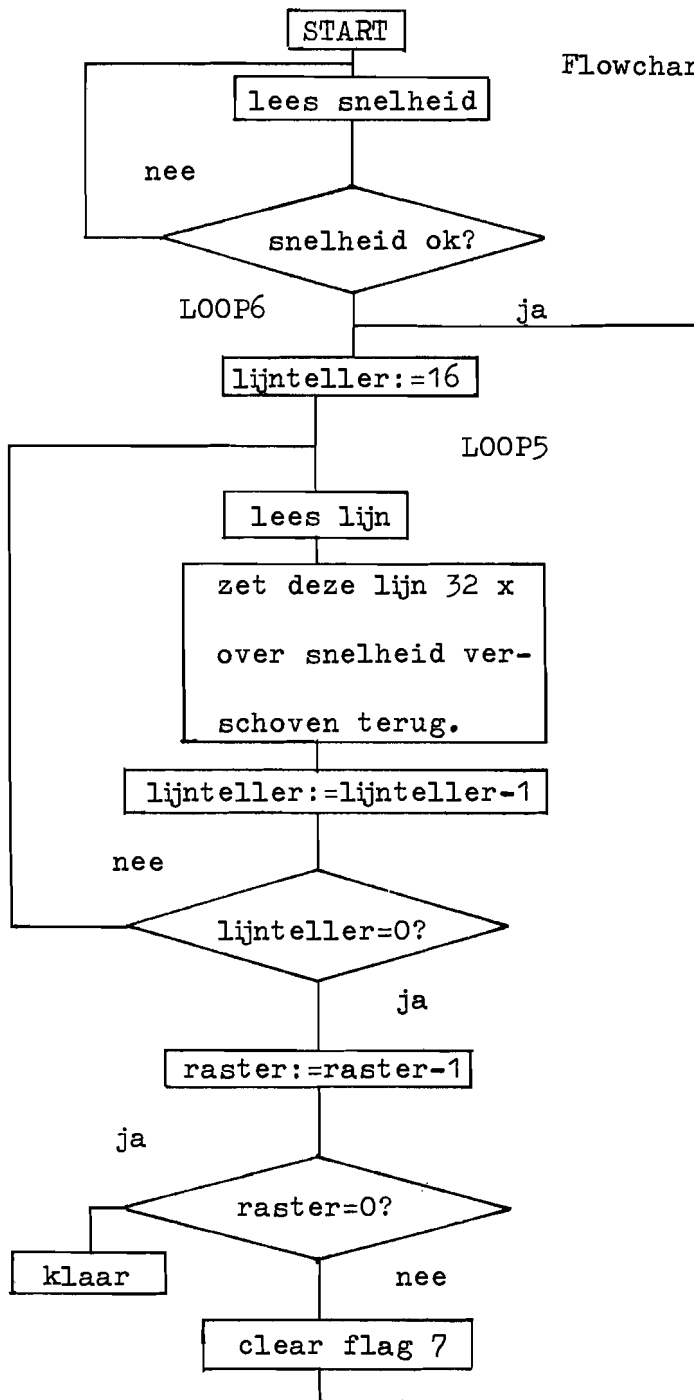
Flowchart HALV



Flowchart AFLLOOP



Flowchart BEWEEG



B10360,00

END PASS 1

1			.TITLE	SUBR, ' INLEZEN NIVEAUS '
2			.EXTD	
3			.LOCAL	
4			.GLOBL	BFNIV, BEFBZ, TABUIT, STUURS, DSHR
5			.GLOBL	LEESDR, BEPMSE, OUTPUT, BEPMI, HALV
6			.GLOBL	TIDEC, TODEC, BINASC, VEEG, TABIN, TABOUT
7	0000		.ASECT	
8	0002	NOISE:	.=,+2	
9	0004	SIGNAL:	.=,+2	
10	0018	DREMP:	.=,+20	
11	0020	NIVEAU:	.=,+20	
12	0040	MI:	.=,+20	
13	0041	DELTAN:	.=,+1	
14	0042	INDEX:	.=,+1	
15	0043	TERM:	.=,+1	
16	0044	XN:	.=,+1	
17	0045	SCHAT:	.=,+1	
18	0046	AANTAL:	.=,+1	
19	0047	SOORT:	.=,+1	
20	0048	INFUT:	.=,+1	
21	0070	TAB:	.=,+40	
22	0000		.TSECT	
23	0000	0D0A A	\$TE1: .WORD	0D0A
24	0001	4E49 A	.ASCII	'NIVEAUS - '
	0002	5645 A		
	0003	4155 A		
	0004	5320 A		
	0005	2D20 A		
25	0006	0000 A	.WORD	0
26	0007	0D0A A	\$TE2: .WORD	0D0A
27	0008	4741 A	.ASCII	'GAMMA - '
	0009	4D4D A		
	000A	4120 A		
	000B	2D20 A		
	000C	2020 A		
28	000D	0000 A	.WORD	0
29	000E	0D0A A	\$TE3: .WORD	0D0A
30	000F	4D49 A	.ASCII	'MI WORDT '
	0010	2057 A		
	0011	4F52 A		
	0012	4454 A		
	0013	2020 A		
31	0014	0000 A	.WORD	0
32	0015	0D0A A	\$TE4: .WORD	0D0A
33	0016	4452 A	.ASCII	'DREMPELS - '
	0017	454D A		
	0018	5045 A		
	0019	4C53 A		
	001A	202D A		
	001B	2020 A		
34	001C	0000 A	.WORD	0
35	001D	4C00 A	VEEG: LI	0,0
36	001E	4F28 A	LI	3,40
37	001F	A347 A	\$VEEG: ST	0, TAB-1(3)

38	0020	4BFF	A		AISZ	3,-1
39	0021	21FD	A		JMP	\$VEEG
40	0022	0200	A		RTS	
41		0024		\$GAMM:	.,.+1	
42	0024	293B	A	\$BEPMI:	JSR	\$RED
43	0025	2D47	A		JSR	@\$MMSG
44	0026	0007	T		.WORD	\$TE2
45	0027	2802	X		JSR	TIDEC
46	0028	A1FA	A		ST	0,\$GAMM
47	0029	294C	A		JSR	HALV
48	002A	A14A	A		ST	0,\$TEL
49	002B	3281	A		RCPY	0,2
50	002C	0750	A		SKBIT	0
51	002D	2101	A		JMP	\$MI5
52	002E	4801	A		AISZ	0,1
53	002F	5CFF	A	\$MI5:	SHR	0,1
54	0030	A157	A		ST	0,\$TEL1
55	0031	3381	A		RCPY	0,3
56	0032	85F0	A		LD	1,\$GAMM
57	0033	A72C	A		ST	1,MI(3)
58	0034	0480	A	\$MI1:	MPY	\$GAMM
	0035	0023	T			
59	0036	0490	A		DIV	\$100
	0037	0094	T			
60	0038	E15C	A		SKG	0,\$50
61	0039	2101	A		JMP	\$MI2
62	003A	4901	A		AISZ	1,1
63	003B	A72D	A	\$MI2:	ST	1,MI+1(3)
64	003C	4B01	A		AISZ	3,1
65	003D	FD37	A		SKNE	3,\$TEL
66	003E	2101	A		JMP	\$MI4
67	003F	21F4	A		JMP	\$MI1
68	0040	4F00	A	\$MI4:	LI	3,0
69	0041	2918	A	\$MI3:	JSR	\$BEP
70	0042	4B01	A		AISZ	3,1
71	0043	4AFF	A		AISZ	2,-1
72	0044	A930	A		ST	2,\$TEL
73	0045	3C81	A		RCPY	3,0
74	0046	D12E	A		SUB	0,\$TEL
75	0047	F13E	A		SKNE	0,\$NUL
76	0048	2106	A		JMP	\$MI7
77	0049	F13D	A		SKNE	0,\$EEN
78	004A	2101	A		JMP	\$MI6
79	004B	21F5	A		JMP	\$MI3
80	004C	8547	A	\$MI6:	LD	1,\$100
81	004D	A72B	A		ST	1,MI-1(3)
82	004E	2101	A		JMP	\$END3
83	004F	290A	A	\$MI7:	JSR	\$BEP
84	0050	2D1C	A	\$END3:	JSR	@\$MMSG
85	0051	000E	T		.WORD	\$TE3
86	0052	2923	A		JSR	HALV
87	0053	A102	A		ST	0,\$GET1
88	0054	2805	X		JSR	TABOUT
89	0055	002C	A		.WORD	MI
90		0057		\$GET1:	.,.+1	

91	0057	0008	A		.WORD	8
92	0058	290C	A		JSR	\$REST
93	0059	0200	A		RTS	
94	005A	4C00	A	\$BEP:	LI	0,0
95	005B	8514	A		LD	1,\$TDZ
96	005C	0690	A		DIV	MI(2)
	005D	002C	A			
97	005E	A72B	A		ST	1,MI-1(3)
98	005F	0200	A		RTS	
99	0060	A110	A	\$RED:	ST	0,\$SAVE
100	0061	A510	A		ST	1,\$SAVE+1
101	0062	A910	A		ST	2,\$SAVE+2
102	0063	AD10	A		ST	3,\$SAVE+3
103	0064	0200	A		RTS	
104	0065	810B	A	\$REST:	LD	0,\$SAVE
105	0066	850B	A		LD	1,\$SAVE+1
106	0067	890B	A		LD	2,\$SAVE+2
107	0068	8D0B	A		LD	3,\$SAVE+3
108	0069	0200	A		RTS	
109	006A	8103	A	\$NLCR:	LD	0,\$ODOA
110	006B	2D03	A		JSR	@\$2PUT
111	006C	0200	A		RTS	
112	006D	7EC3	A	\$MSG:	.WORD	X'7EC3
113	006E	0D0A	A	\$ODOA:	.WORD	X'0D0A
114	006F	7ED3	A	\$2PUT:	.WORD	X'7ED3
115	0070	2710	A	\$TDZ:	.WORD	10000
116		0075		\$SAVE:	.,.+4	
117		0076		\$TEL:	.,.+1	
118	0076	8045	A	HALV:	LD	0,AANTAL
119	0077	0750	A		SKBIT	0
120	0078	2101	A		JMP	\$A
121	0079	4801	A		AISZ	0,1
122	007A	5CFF	A	\$A:	SHR	0,1
123	007B	0200	A		RTS	
124	007C	4000	A	BPNIIV:	PUSH	0
125	007D	29F8	A		JSR	HALV
126	007E	A105	A		ST	0,\$GET2
127	007F	2DED	A		JSR	@\$MSG
128	0080	0000	T		.WORD	\$TE1
129	0081	4400	A		PULL	0
130	0082	2804	X		JSR	TABIN
131	0083	0018	A		.WORD	NIVEAU
132		0085		\$GET2:	.,.+1	
133	0085	0200	A		RTS	
134	0086	0000	A	\$NUL:	.WORD	0
135	0087	0001	A	\$EEN:	.WORD	1
136		0089		\$TEL1:	.,.+1	
137	0089	4000	A	LEESDR:	PUSH	0
138	008A	2DE2	A		JSR	@\$MSG
139	008B	0015	T		.WORD	\$TE4
140	008C	29E9	A		JSR	HALV
141	008D	48FF	A		AISZ	0,-1
142	008E	A103	A		ST	0,\$GET3
143	008F	4400	A		PULL	0
144	0090	2804	X		JSR	TABIN

145	0091	0004	A		.WORD	DREMP
146		0093		\$GET3:	.,.+1	
147	0093	0200	A		RTS	
148	0094	0064	A	\$100:	.WORD	100
149	0095	0032	A	\$50:	.WORD	50
150	0096	29C9	A	BEPMSE:	JSR	\$RED
151	0097	8447	A		LD	1,INPUT
152	0098	0480	A		MPY	INPUT
	0099	0047	A			
153	009A	5DFF	A		SHR	1,1
154	009B	04A0	A		DADD	SIGNAL
	009C	0002	A			
155	009D	A002	A		ST	0,SIGNAL
156	009E	A403	A		ST	1,SIGNAL+1
157	009F	290E	A		JSR	\$BPER
158	00A0	850C	A		LD	1,\$FOUT
159	00A1	E5E4	A		SKG	1,\$NUL
160	00A2	5101	A		CAI	1,1
161	00A3	A536	A		ST	1,\$HULP
162	00A4	0480	A		MPY	\$HULP
	00A5	00DA	T			
163	00A6	5DFF	A		SHR	1,1
164	00A7	04A0	A		DADD	NOISE
	00A8	0000	A			
165	00A9	A000	A		ST	0,NOISE
166	00AA	A401	A		ST	1,NOISE+1
167	00AB	29B9	A		JSR	\$REST
168	00AC	0200	A		RTS	
169		00AE		\$FOUT:	.,.+1	
170	00AE	4000	A	\$BPER:	PUSH	0
171	00AF	2905	A		JSR	\$SCH1
172	00B0	0047	A		SUB	0,INPUT
173	00B1	5001	A		CAI	0,1
174	00B2	A1FA	A		ST	0,\$FOUT
175	00B3	4400	A		PULL	0
176	00B4	0200	A		RTS	
177	00B5	4C00	A	\$SCH1:	LI	0,0
178	00B6	4100	A		PUSH	1
179	00B7	8444	A		LD	1,SCHAT
180	00B8	0490	A	\$DEEL:	DIV	\$100
	00B9	0094	T			
181	00BA	E1DA	A		SKG	0,\$50
182	00BB	2101	A		JMP	\$S1
183	00BC	4901	A		AISZ	1,1
184	00BD	3481	A	\$S1:	RCPY	1,0
185	00BE	4500	A		PULL	1
186	00BF	0200	A		RTS	
187	00C0	4C00	A	\$TERM:	LI	0,0
188	00C1	4100	A		PUSH	1
189	00C2	8442	A		LD	1,TERM
190	00C3	21F4	A		JMP	\$DEEL
191	00C4	A5AC	A	OUTPUT:	ST	1,\$SAVE
192	00C5	8446	A		LD	1,SOORT
193	00C6	F5BF	A		SKNE	1,\$NUL
194	00C7	81E5	A		LD	0,\$FOUT

195	00C8	F5BE	A	SKNE	1,\$EEN
196	00C9	291B	A	JSR	\$IND
197	00CA	F50B	A	SKNE	1,\$TWEE
198	00CB	29F4	A	JSR	\$TERM
199	00CC	F50A	A	SKNE	1,\$DRIE
200	00CD	290D	A	JSR	\$XN
201	00CE	F509	A	SKNE	1,\$VIER
202	00CF	29E5	A	JSR	\$SCH1
203	00D0	F508	A	SKNE	1,\$UYF
204	00D1	290F	A	JSR	\$DELT
205	00D2	2801	X	JSR	STUURS
206	00D3	859D	A	LD	1,\$SAVE
207	00D4	0200	A	RTS	
208	00D5	00FF	A	\$XFF: .WORD	X'FF
209	00D6	0002	A	\$TWEE: .WORD	2
210	00D7	0003	A	\$DRIE: .WORD	3
211	00D8	0004	A	\$VIER: .WORD	4
212	00D9	0005	A	\$UYF: .WORD	5
213		00DB		\$HULP: .,+,1	
214	00DB	4C00	A	\$XN: LI	0,0
215	00DC	4100	A	PUSH	1
216	00DD	8443	A	LD	1,XN
217	00DE	E5A7	A	SKG	1,\$NUL
218	00DF	5101	A	CAI	1,1
219	00E0	21D7	A	JMP	\$DEEL
220	00E1	4C00	A	\$DELT: LI	0,0
221	00E2	4100	A	PUSH	1
222	00E3	8440	A	LD	1,DELTAN
223	00E4	21D3	A	JMP	\$DEEL
224	00E5	4100	A	\$IND: PUSH	1
225	00E6	8045	A	LD	0,AANTAL
226	00E7	4D00	A	LI	1,0
227	00E8	0750	A	SKBIT	0
228	00E9	2101	A	JMP	\$IND1
229	00EA	4901	A	AISZ	1,1
230	00EB	A5EE	A	\$IND1: ST	1,\$HULP
231	00EC	8443	A	LD	1,XN
232	00ED	8041	A	LD	0,INDEX
233	00EE	C135	A	ADD	0,\$MIN1
234	00EF	E534	A	SKG	1,\$MIN1
235	00F0	2101	A	JMP	\$IND2
236	00F1	2102	A	JMP	\$IND3
237	00F2	5000	A	\$IND2: CAI	0,0
238	00F3	C1E6	A	ADD	0,\$HULP
239	00F4	61E0	A	\$IND3: AND	0,\$XFF
240	00F5	5C04	A	SHL	0,4
241	00F6	4500	A	FULL	1
242	00F7	0200	A	RTS	
243	00F8	0060	T	.WORD	\$RED
244	00F9	2DFE	A	TABUIT: JSR	@,-1
245	00FA	8045	A	LD	0,AANTAL
246	00FB	0750	A	SKBIT	0
247	00FC	2101	A	JMP	\$TAB3
248	00FD	4801	A	AISZ	0,1
249	00FE	A1DB	A	\$TAB3: ST	0,\$HULP

SUBR INLEZEN NIVEAUS

```

250 00FF 2D21 A      JSR    @ $NEWL
251 0100 4F00 A      LI     3,0
252 0101 4E04 A      $TAB2: LI     2,4
253 0102 8348 A      $TAB1: LD     0,TAB(3)
254 0103 8749 A      LD     1,TAB+1(3)
255 0104 2803 X      JSR    BINASC
256 0105 811C A      LD     0,$20
257 0106 2D1C A      JSR    @ $PUTC
258 0107 4B02 A      AISZ  3,2
259 0108 FDD1 A      SKNE  3,$HULP
260 0109 2104 A      JMP    $TABE
261 010A 4AFF A      AISZ  2,-1
262 010B 21F6 A      JMP    $TAB1
263 010C 2D14 A      JSR    @ $NEWL
264 010D 21F3 A      JMP    $TAB2
265 010E 2D01 A      $TABE: JSR    @,+2
266 010F 0200 A      RTS
267 0110 0065 T      .WORD $REST
268 0111 0060 T      .WORD $RED
269 0112 2DFE A      BEPBZ: JSR    @,-1
270 0113 8C41 A      LD     3,INDEX
271 0114 CD0F A      ADD   3,$MIN1
272 0115 5F01 A      SHL   3,1
273 0116 8348 A      LD     0,TAB(3)
274 0117 8749 A      LD     1,TAB+1(3)
275 0118 04A0 A      DADD  $DEEN
      0119 011F T
276 011A A348 A      ST     0,TAB(3)
277 011B A749 A      ST     1,TAB+1(3)
278 011C 2D01 A      JSR    @,+2
279 011D 0200 A      RTS
280 011E 0065 T      .WORD $REST
281 011F 0000 A      $DEEN: .WORD 0
282 0120 0001 A      .WORD 1
283 0121 006A T      $NEWL: .WORD $NLCR
284 0122 0020 A      $20:   .WORD X'20
285 0123 7E59 A      $PUTC: .WORD X'7E59
286 0124 FFFF A      $MIN1: .WORD -1
287 0125 0750 A      DSHR:  SKBIT 0
288 0126 2106 A      JMP    $N
289 0127 3180 A      RXCH  0,1
290 0128 5CFF A      SHR   0,1
291 0129 072F A      SETBIT 15
292 012A 3180 A      RXCH  0,1
293 012B 5CFF A      SHR   0,1
294 012C 0200 A      RTS
295 012D 5CFF A      $N:   SHR   0,1
296 012E 5DFF A      SHR   1,1
297 012F 0200 A      RTS
298      0000      .END

```

```

AANTAL 0045 A      BEPBZ 0112 GT*      BEPMI 0024 GT*
BEPMSE 0096 GT*   BINASC 0003 X      BFNIV 007C GT*
DELTAN 0040 A      DREMP 0004 A      DSHR 0125 GT*
HALV   0076 GT    INDEX 0041 A      INPUT 0047 A
LEESDR 0089 GT*   MI     002C A      NIVEAU 0018 A

```

NOISE	0000	A	OUTPUT	00C4	GT*	SCHAT	0044	A
SIGNAL	0002	A	SODRT	0046	A	STUURS	0001	X
TAR	0048	A	TABIN	0004	X	TABOUT	0005	X
TABOUT	00F9	GT*	TERM	0042	A	TIDEC	0002	X
TIDEC	0000	X*	VEEG	001D	GT*	XN	0043	A
\$000A	006E	T	\$100	0094	T	\$20	0122	T
\$2PUT	006F	T	\$50	0095	T	\$A	007A	T
\$BEP	005A	T	\$BPER	00AE	T	\$DEEL	00B8	T
\$DEEN	011F	T	\$DELT	00E1	T	\$DRIE	00D7	T
\$EEN	0087	T	\$END3	0050	T	\$FOUT	00AD	T
\$GAMM	0023	T	\$GET1	0056	T	\$GET2	0084	T
\$GET3	0092	T	\$HULP	00DA	T	\$IND	00E5	T
\$IND1	00EB	T	\$IND2	00F2	T	\$IND3	00F4	T
\$MSG	006D	T	\$MI1	0034	T	\$MI2	003B	T
\$MI3	0041	T	\$MI4	0040	T	\$MI5	002F	T
\$MI6	004C	T	\$MI7	004F	T	\$MIN1	0124	T
\$N	012D	T	\$NEWL	0121	T	\$NLCR	006A	T
\$NUL	0086	T	\$PUTC	0123	T	\$RED	0060	T
\$REST	0065	T	\$S1	00BD	T	\$SAVE	0071	T
\$SCH1	00B5	T	\$TAB1	0102	T	\$TAB2	0101	T
\$TAB3	00FE	T	\$TABE	010E	T	\$TDZ	0070	T
\$TE1	0000	T	\$TE2	0007	T	\$TE3	000E	T
\$TE4	0015	T	\$TEL	0075	T	\$TEL1	0088	T
\$TERM	00C0	T	\$TWEE	00D6	T	\$VEEG	001F	T
\$VIER	00DB	T	\$VYF	00D9	T	\$XFF	00D5	T
\$XN	00DB	T						

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=6C9A

OBJECT CHECKSUM=8552

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 03B0

FINAL INPUT SECTOR HEX - 03B9

FIRST OBJECT SECTOR HEX - 03BA

FINAL OBJECT SECTOR HEX - 03BC

NEXT ASSEMBLY

*.ASM

END PASS 1

```

1          .TITLE  SUBR, ' ALLES INLEZEN '
2          .EXTD
3          .LOCAL
4          .GLOBL  BFNIV,BEPBZ,TABUIT,STUURS,DSHR
5          .GLOBL  LEESDR,BEPMSE,OUTPUT,BEFMI,HALV
6          .GLOBL  BINASC,VEEG,TABIN
7          0000          .ASECT
8          0002          NOISE:  .=.+2
9          0004          SIGNAL: .=.+2
10         0018          DREMP:  .=.+20
11         002C          NIVEAU: .=.+20
12         0040          MI:      .=.+20
13         0041          DELTAN: .=.+1
14         0042          INDEX:  .=.+1
15         0043          TERM:   .=.+1
16         0044          XN:     .=.+1
17         0045          SCHAT:  .=.+1
18         0046          AANTAL: .=.+1
19         0047          SOORT:  .=.+1
20         0048          INPUT:  .=.+1
21         0070          TAB:    .=.+40
22         0000          .TSECT
23 0000 0D0A A  $TE1:  .WORD  0D0A
24 0001 4E49 A  .ASCII  'NIVEAUS - '
      0002 5645 A
      0003 4155 A
      0004 5320 A
      0005 2D20 A
25 0006 0000 A  .WORD  0
26 0007 0D0A A  $TE2:  .WORD  0D0A
27 0008 4D49 A  .ASCII  'MI - '
      0009 202D A
      000A 2020 A
28 000E 0000 A  .WORD  0
29 000C 0D0A A  $TE4:  .WORD  0D0A
30 000D 4452 A  .ASCII  'DREMPELS - '
      000E 454D A
      000F 5045 A
      0010 4C53 A
      0011 202D A
      0012 2020 A
31 0013 0000 A  .WORD  0
32 0014 4C00 A  VEEG:  LI  0,0
33 0015 4F28 A  LI  3,40
34 0016 A347 A  $VEEG: ST  0,TAB-1(3)
35 0017 4BFF A  AISZ  3,-1
36 0018 21FD A  JMP  $VEEG
37 0019 0200 A  RTS
38 001A 4000 A  BEPMI: PUSH  0
39 001B 291C A  JSR  HALV
40 001C A105 A  ST  0,$GET1
41 001D 2D13 A  JSR  @MMSG
42 001E 0007 T  .WORD  $TE2
43 001F 4400 A  FULL  0

```

```

44 0020 2803 X      JSR      TABIN
45 0021 002C A      .WORD    MI
46      0023      $GET1:  .,+.1
47 0023 0200 A      RTS
48 0024 A10F A      $RED:   ST      0,$SAVE
49 0025 A50F A      ST      1,$SAVE+1
50 0026 A90F A      ST      2,$SAVE+2
51 0027 AD0F A      ST      3,$SAVE+3
52 0028 0200 A      RTS
53 0029 810A A      $REST: LD      0,$SAVE
54 002A 850A A      LD      1,$SAVE+1
55 002B 890A A      LD      2,$SAVE+2
56 002C 8D0A A      LD      3,$SAVE+3
57 002D 0200 A      RTS
58 002E 8103 A      $NLCR: LD      0,$ODOA
59 002F 2D03 A      JSR      @2PUT
60 0030 0200 A      RTS
61 0031 7EC3 A      $MSG:   .WORD    X'7EC3
62 0032 0D0A A      $ODOA:  .WORD    X'0D0A
63 0033 7ED3 A      $2PUT:  .WORD    X'7ED3
64      0038      $SAVE:  .,+.4
65 0038 8045 A      HALV:   LD      0,AANTAL
66 0039 0750 A      SKBIT   0
67 003A 2101 A      JMP     $A
68 003B 4801 A      AISZ   0,1
69 003C 5CFF A      $A:    SHR   0,1
70 003D 0200 A      RTS
71 003E 4000 A      BPNIV: PUSH   0
72 003F 29F8 A      JSR    HALV
73 0040 A105 A      ST     0,$GET2
74 0041 2DEF A      JSR    @MSG
75 0042 0000 T      .WORD  $TE1
76 0043 4400 A      FULL   0
77 0044 2803 X      JSR    TABIN
78 0045 0018 A      .WORD  NIVEAU
79      0047      $GET2: .,+.1
80 0047 0200 A      RTS
81 0048 0000 A      $NUL:  .WORD    0
82 0049 0001 A      $EEN:  .WORD    1
83 004A 4000 A      LEESDR: PUSH   0
84 004B 2DE5 A      JSR    @MSG
85 004C 000C T      .WORD  $TE4
86 004D 29EA A      JSR    HALV
87 004E 48FF A      AISZ   0,-1
88 004F A103 A      ST     0,$GET3
89 0050 4400 A      FULL   0
90 0051 2803 X      JSR    TABIN
91 0052 0004 A      .WORD  DREMP
92      0054      $GET3: .,+.1
93 0054 0200 A      RTS
94 0055 0064 A      $100:  .WORD    100
95 0056 0032 A      $50:   .WORD    50
96 0057 29CC A      BEFMSE: JSR    $RED
97 0058 8447 A      LD     1,INPUT
98 0059 0480 A      MPY    INPUT

```

005A	0047	A		
99	005B	5DFF	SHR	1,1
100	005C	04A0	DADD	SIGNAL
	005D	0002		
101	005E	A002	ST	0,SIGNAL
102	005F	A403	ST	1,SIGNAL+1
103	0060	290E	JSR	\$BPER
104	0061	850C	LD	1,\$FOUT
105	0062	E5E5	SKG	1,\$NUL
106	0063	5101	CAI	1,1
107	0064	A536	ST	1,\$HULP
108	0065	0480	MPY	\$HULP
	0066	009B	T	
109	0067	5DFF	SHR	1,1
110	0068	04A0	DADD	NOISE
	0069	0000		
111	006A	A000	ST	0,NOISE
112	006B	A401	ST	1,NOISE+1
113	006C	29BC	JSR	\$REST
114	006D	0200	RTS	
115		006F	\$FOUT:	.,=,+1
116	006F	4000	\$BPER:	PUSH 0
117	0070	2905	JSR	\$SCH1
118	0071	D047	SUB	0,INFUT
119	0072	5001	CAI	0,1
120	0073	A1FA	ST	0,\$FOUT
121	0074	4400	PULL	0
122	0075	0200	RTS	
123	0076	4C00	\$SCH1:	LI 0,0
124	0077	4100	PUSH	1
125	0078	8444	LD	1,\$CHAT
126	0079	0490	\$DEEL:	DIV \$100
	007A	0055	T	
127	007B	E1DA	SKG	0,\$50
128	007C	2101	JMP	\$S1
129	007D	4901	AISZ	1,1
130	007E	3481	\$S1:	RCPY 1,0
131	007F	4500	PULL	1
132	0080	0200	RTS	
133	0081	4C00	\$TERM:	LI 0,0
134	0082	4100	PUSH	1
135	0083	8442	LD	1,TERM
136	0084	21F4	JMP	\$DEEL
137	0085	A5AE	OUTPUT:	ST 1,\$SAVE
138	0086	8446	LD	1,\$OORT
139	0087	F5C0	SKNE	1,\$NUL
140	0088	81E5	LD	0,\$FOUT
141	0089	F5BF	SKNE	1,\$EEN
142	008A	291B	JSR	\$IND
143	008B	F50B	SKNE	1,\$TWEE
144	008C	29F4	JSR	\$TERM
145	008D	F50A	SKNE	1,\$DRIE
146	008E	290D	JSR	\$XN
147	008F	F509	SKNE	1,\$VIER
148	0090	29E5	JSR	\$SCH1

```

149 0091 F508 A          SKNE      1,$VYF
150 0092 290F A          JSR      $DELT
151 0093 2801 X          JSR      STUURS
152 0094 859F A          LD       1,$SAVE
153 0095 0200 A          RTS
154 0096 00FF A  $XFF:  .WORD  X'FF
155 0097 0002 A  $TWEE: .WORD  2
156 0098 0003 A  $DRIE:  .WORD  3
157 0099 0004 A  $VIER:  .WORD  4
158 009A 0005 A  $VYF:   .WORD  5
159      009C      $HULP:  .=.+1
160 009C 4C00 A  $XN:    LI      0,0
161 009D 4100 A          PUSH    1
162 009E 8443 A          LD       1,XN
163 009F E5A8 A          SKG     1,$NUL
164 00A0 5101 A          CAI     1,1
165 00A1 21D7 A          JMP     $DEEL
166 00A2 4C00 A  $DELT:  LI      0,0
167 00A3 4100 A          PUSH    1
168 00A4 8440 A          LD       1,DELTAN
169 00A5 21D3 A          JMP     $DEEL
170 00A6 4100 A  $IND:   PUSH    1
171 00A7 8045 A          LD       0,AANTAL
172 00A8 4D00 A          LI      1,0
173 00A9 0750 A          SKBIT   0
174 00AA 2101 A          JMP     $IND1
175 00AB 4901 A          AISZ    1,1
176 00AC A5EE A  $IND1:  ST      1,$HULP
177 00AD 8443 A          LD       1,XN
178 00AE 8041 A          LD       0,INDEX
179 00AF C135 A          ADD     0,$MIN1
180 00B0 E534 A          SKG     1,$MIN1
181 00B1 2101 A          JMP     $IND2
182 00B2 2102 A          JMP     $IND3
183 00B3 5000 A  $IND2:  CAI     0,0
184 00B4 C1E6 A          ADD     0,$HULP
185 00B5 61E0 A  $IND3:  AND     0,$XFF
186 00B6 5C04 A          SHL     0,4
187 00B7 4500 A          PULL    1
188 00B8 0200 A          RTS
189 00B9 0024 T          .WORD   $RED
190 00BA 2DFE A  TABUIT: JSR     @.-1
191 00BB 8045 A          LD       0,AANTAL
192 00BC 0750 A          SKBIT   0
193 00BD 2101 A          JMP     $TAB3
194 00BE 4801 A          AISZ    0,1
195 00BF A1DB A  $TAB3:  ST      0,$HULP
196 00C0 2D21 A          JSR     @$NEWL
197 00C1 4F00 A          LI      3,0
198 00C2 4E04 A  $TAB2:  LI      2,4
199 00C3 8348 A  $TAB1:  LD      0,TAB(3)
200 00C4 8749 A          LD      1,TAB+1(3)
201 00C5 2802 X          JSR     BINASC
202 00C6 811C A          LD      0,$20
203 00C7 2D1C A          JSR     @$PUTC

```

```

204 00C8 4B02 A      AISZ      3,2
205 00C9 FDD1 A      SKNE      3,$HULP
206 00CA 2104 A      JMP       $TABE
207 00CB 4AFF A      AISZ      2,-1
208 00CC 21F6 A      JMP       $TAB1
209 00CD 2D14 A      JSR       @$NEWL
210 00CE 21F3 A      JMP       $TAB2
211 00CF 2D01 A      $TABE:   JSR       @.+2
212 00D0 0200 A      RTS
213 00D1 0029 T      .WORD    $REST
214 00D2 0024 T      .WORD    $RED
215 00D3 2DFE A      BEPBZ:   JSR       @.-1
216 00D4 8C41 A      LD        3,INDEX
217 00D5 CD0F A      ADD       3,$MIN1
218 00D6 5F01 A      SHL       3,1
219 00D7 8348 A      LD        0,TAB(3)
220 00D8 8749 A      LD        1,TAB+1(3)
221 00D9 04A0 A      DADD      $DEEN
      00DA 00E0 T
222 00DB A348 A      ST        0,TAB(3)
223 00DC A749 A      ST        1,TAB+1(3)
224 00DD 2D01 A      JSR       @.+2
225 00DE 0200 A      RTS
226 00DF 0029 T      .WORD    $REST
227 00E0 0000 A      $DEEN:   .WORD    0
228 00E1 0001 A      .WORD    1
229 00E2 002E T      $NEWL:   .WORD    $NLCR
230 00E3 0020 A      $20:     .WORD    X'20
231 00E4 7E59 A      $PUTC:   .WORD    X'7E59
232 00E5 FFFF A      $MIN1:   .WORD    -1
233 00E6 0750 A      DSHR:    SKBIT   0
234 00E7 2106 A      JMP       $N
235 00E8 3180 A      RXCH     0,1
236 00E9 5CFF A      SHR      0,1
237 00EA 072F A      SETBIT   15
238 00EB 3180 A      RXCH     0,1
239 00EC 5CFF A      SHR      0,1
240 00ED 0200 A      RTS
241 00EE 5CFF A      $N:      SHR      0,1
242 00EF 5CFF A      SHR      1,1
243 00F0 0200 A      RTS
244      0000      .END
    
```

AANTAL	0045	A	BEPBZ	00D3	GT*	BEPMI	001A	GT*
BEPMSE	0057	GT*	BINASC	0002	X	BPNIU	003E	GT*
DELTAN	0040	A	DREMP	0004	A	DSHR	00E6	GT*
HALV	0038	GT	INDEX	0041	A	INPUT	0047	A
LEESDR	004A	GT*	MI	002C	A	NIVEAU	0018	A
NOISE	0000	A	OUTPUT	0085	GT*	SCHAT	0044	A
SIGNAL	0002	A	SOORT	0046	A	STUURS	0001	X
TAB	0048	A	TABIN	0003	X	TABUIT	00BA	GT*
TERM	0042	A	VEEG	0014	GT*	XN	0043	A
\$0DOA	0032	T	\$100	0055	T	\$20	00E3	T
\$2PUT	0033	T	\$50	0056	T	\$A	003C	T
\$BPER	006F	T	\$DEEL	0079	T	\$DEEN	00E0	T
\$DELT	00A2	T	\$DRIE	0098	T	\$EEN	0049	T

\$FOUT	006E	T	\$GET1	0022	T	\$GET2	0046	T
\$GET3	0053	T	\$HULP	009B	T	\$IND	00A6	T
\$IND1	00AC	T	\$IND2	00B3	T	\$IND3	00B5	T
\$MESG	0031	T	\$MIN1	00E5	T	\$N	00EE	T
\$NEWL	00E2	T	\$NLCR	002E	T	\$NUL	0048	T
\$PUTC	00E4	T	\$RED	0024	T	\$REST	0029	T
\$S1	007E	T	\$SAVE	0034	T	\$SCH1	0076	T
\$TAB1	00C3	T	\$TAB2	00C2	T	\$TAB3	00BF	T
\$TABE	00CF	T	\$TE1	0000	T	\$TE2	0007	T
\$TE4	000C	T	\$TERM	0081	T	\$TWEE	0097	T
\$VEEG	0016	T	\$VIER	0099	T	\$UYF	009A	T
\$XFF	0096	T	\$XN	009C	T			

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=9AB0

OBJECT CHECKSUM=4E1F

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 03C0

FINAL INPUT SECTOR HEX - 03C7

FIRST OBJECT SECTOR HEX - 03C8

FINAL OBJECT SECTOR HEX - 03C9

NEXT ASSEMBLY

*.ASM

D10400,BO

END PASS 1

```

1          .TITLE  SUBR,'TABEL LEZEN EN PRINTEN '
2          .EXTD
3          .LOCAL
4          .GLOBL  TABOUT,TABIN,TIDEC,TODEC
5          0000    .TSECT
6 0000 2914 A   TABOUT: JSR    $SAVE
7 0001 5700 A           XCHRS  3
8 0002 3E81 A           RCFY   3,2
9 0003 5700 A           XCHRS  3
10 0004 2938 A          JSR    $AANT
11 0005 4A01 A          AISZ   2,1
12 0006 8600 A   $A:    LD     1,(2)
13 0007 911D A   $B:    LD     0,@$LOC
14 0008 2802 X          JSR    TODEC
15 0009 2918 A          JSR    $SPAT
16 000A 791A A          ISZ    $LOC
17 000B 8119 A          LD     0,$LOC
18 000C F119 A          SKNE   0,$MAX
19 000D 2104 A          JMP    $END
20 000E 49FF A          AISZ   1,-1
21 000F 21F7 A          JMP    $B
22 0010 290E A          JSR    $NLCR
23 0011 21F4 A          JMP    $A
24 0012 290C A   $END:  JSR    $NLCR
25 0013 2906 A          JSR    $REST
26 0014 0203 A          RTS    3
27 0015 A111 A   $SAVE: ST    0,$RED
28 0016 A511 A          ST    1,$RED+1
29 0017 A911 A          ST    2,$RED+2
30 0018 AD11 A          ST    3,$RED+3
31 0019 0200 A          RTS
32 001A 810C A   $REST: LD    0,$RED
33 001B 850C A          LD    1,$RED+1
34 001C 890C A          LD    2,$RED+2
35 001D 8D0C A          LD    3,$RED+3
36 001E 0200 A          RTS
37 001F 810B A   $NLCR: LD    0,$ODOA
38 0020 2D0B A          JSR    @$PUT2
39 0021 0200 A          RTS
40 0022 4C20 A   $SPAT: LI    0,X'20
41 0023 2D09 A          JSR    @$PUTC
42 0024 0200 A          RTS
43          0026   $LOC:  .,+.1
44          0027   $MAX:  .,+.1
45          0028   $RED:  .,+.4
46 002B 0D0A A   $ODOA:  .WORD  X'0D0A
47 002C 7ED3 A   $PUT2:  .WORD  X'7ED3
48 002D 7E59 A   $PUTC:  .WORD  X'7E59
49 002E 29E6 A   TABIN:  JSR    $SAVE
50 002F 5700 A          XCHRS  3
51 0030 3E81 A          RCFY   3,2
52 0031 5700 A          XCHRS  3
53 0032 290A A          JSR    $AANT
54 0033 2801 X   $C:    JSR    TIDEC

```

```

55 0034 B1F0 A      ST      0,@$LOC
56 0035 29E9 A      JSR     $NLCR
57 0036 79EE A      ISZ     $LOC
58 0037 81ED A      LD      0,$LOC
59 0038 F1ED A      SKNE   0,$MAX
60 0039 2101 A      JMP     $END2
61 003A 21F8 A      JMP     $C
62 003B 29DE A      $END2: JSR     $REST
63 003C 0202 A      RTS     2
64 003D 8200 A      $AANT: LD      0,(2)
65 003E A1E6 A      ST      0,$LOC
66 003F 4A01 A      AISZ   2,1
67 0040 C200 A      ADD     0,(2)
68 0041 A1E4 A      ST      0,$MAX
69 0042 0200 A      RTS
70                                .LOCAL
71                                .END

```

```

TABIN  002E GT*      TABOUT  0000 GT*      TIDEC  0001 X
TIDEC  0002 X        $ODOA  002B T        $A     0006 T
$AANT  003D T        $B     0007 T        $C     0033 T
$END   0012 T        $END2  003B T        $LOC   0025 T
$MAX   0026 T        $NLCR  001F T        $PUT2  002C T
$PUTC  002D T        $RED   0027 T        $REST  001A T
$SAVE  0015 T        $SPAT  0022 T

```

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=0056

OBJECT CHECKSUM=EEB8

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 0400

FINAL INPUT SECTOR HEX - 0402

FIRST OBJECT SECTOR HEX - 0403

FINAL OBJECT SECTOR HEX - 0403

NEXT ASSEMBLY

*.ASM

ND PASS 1

```

1          .TITLE AFLOOP, 'AFLOOP MET LESEC '
2          .GLOBL LESEC, AFLOOP
3          .LOCAL
4          0000 .TSECT
5          .EXTD
6 0000 4300 A LESEC: PUSH 3
7 0001 4F03 A $LEES: LI 3,3
8 0002 4C00 A LI 0,0
9 0003 A128 A ST 0,$VORG
10 0004 2D20 A $VOLG: JSR @$GECO
11 0005 6120 A AND 0,$X7F
12 0006 F122 A SKNE 0,$X20
13 0007 21F9 A JMP $LEES
14 0008 E11F A SKG 0,$X40
15 0009 2101 A JMP $TTAL
16 000A C120 A ADD 0,$9
17 000B 611E A $TTAL: AND 0,$XF
18 000C C11F A ADD 0,$VORG
19 000D 4BFF A AISZ 3,-1
20 000E 2102 A JMP $LUS
21 000F 4700 A PULL 3
22 0010 0200 A RTS
23 0011 5C04 A $LUS: SHL 0,4
24 0012 A119 A ST 0,$VORG
25 0013 21F0 A JMP $VOLG
26 0014 4000 A AFLOOP: PUSH 0
27 0015 2D0E A JSR @$MESG
28 0016 002D T .WORD $T1
29 0017 2D0D A JSR @$GECO
30 0018 610D A AND 0,$X7F
31 0019 F10D A SKNE 0,$X4E
32 001A 2105 A JMP $END
33 001B 2D08 A JSR @$MESG
34 001C 0037 T .WORD $T2
35 001D 29E2 A JSR LESEC
36 001E A104 A ST 0,$BGNF
37 001F 2D02 A JSR @$BOOT
38 0020 4400 A $END: PULL 0
39 0021 0200 A RTS
40 0022 C000 A $BOOT: .WORD X'C000
41 0023 0024 $BGNF: .=.+1
42 0024 7EC3 A $MESG: .WORD X'7EC3
43 0025 7E73 A $GECO: .WORD X'7E73
44 0026 007F A $X7F: .WORD X'7F
45 0027 004E A $X4E: .WORD X'4E
46 0028 0040 A $X40: .WORD X'40
47 0029 0020 A $X20: .WORD X'20
48 002A 000F A $XF: .WORD X'F
49 002B 0009 A $9: .WORD 9
50 002C 002D $VORG: .=.+1
51 002D 0D0A A $T1: .WORD 0D0A
52 002E 4449 A .ASCII 'DISK? (Y OR N) '
    002F 534B A
    0030 3F20 A

```

```

0031 2859 A
0032 204F A
0033 5220 A
0034 4E29 A
0035 2020 A
53 0036 0000 A      .WORD 0
54 0037 0D0A A      $T2: .WORD 0D0A
55 0038 5345 A      .ASCII 'SECTOR HEX - '
0039 4354 A
003A 4F52 A
003B 2048 A
003C 4558 A
003D 202D A
003E 2020 A
56 003F 0000 A      .WORD 0
57                      .LOCAL
58          0000      .END

```

```

IFLOOP 0014 GT*      LESEC 0000 GT      $9      002B T
$BGNP 0023 T        $BOOT 0022 T      $END    0020 T
$GECO 0025 T        $LEES 0001 T      $LUS    0011 T
$MSG 0024 T         $T1   002D T      $T2     0037 T
$TTAL 000B T        $VOLG 0004 T      $VORG   002C T
$X20 0029 T         $X40 0028 T      $X4E    0027 T
$X7F 0026 T         $XF   002A T

```

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=E618

OBJECT CHECKSUM=A105

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 01F0

FINAL INPUT SECTOR HEX - 01F1

FIRST OBJECT SECTOR HEX - 01F2

FINAL OBJECT SECTOR HEX - 01F2

NEXT ASSEMBLY

CLASH

ND PASS 1

```

1          .TITLE  AFLOOP,'AFLOOP ZONDER LESEC'
2          .GLOBL  LESEC,AFLOOP
3          .LOCAL
4          0000  .TSECT
5          .EXTD
6 0000 4000 A AFLOOP: PUSH 0
7 0001 2D0E A JSR @$MSG
8 0002 0014 T .WORD $T1
9 0003 2D0D A JSR @$GECO
10 0004 610D A AND 0,$X7F
11 0005 F10D A SKNE 0,$X4E
12 0006 2105 A JMP $END
13 0007 2D08 A JSR @$MSG
14 0008 001E T .WORD $T2
15 0009 2801 X JSR LESEC
16 000A A104 A ST 0,$BGNP
17 000B 2D02 A JSR @$BOOT
18 000C 4400 A $END: PULL 0
19 000D 0200 A RTS
20 000E C000 A $BOOT: .WORD X'C000
21 0010 0010 $BGNP: .,+.1
22 0010 7EC3 A $MSG: .WORD X'7EC3
23 0011 7E73 A $GECO: .WORD X'7E73
24 0012 007F A $X7F: .WORD X'7F
25 0013 004E A $X4E: .WORD X'4E
26 0014 0D0A A $T1: .WORD 0D0A
27 0015 4449 A .ASCII 'DISK? (Y OR N) '
0016 534B A
0017 3F20 A
0018 2859 A
0019 204F A
001A 5220 A
001B 4E29 A
001C 2020 A
28 001D 0000 A .WORD 0
29 001E 0D0A A $T2: .WORD 0D0A
30 001F 5345 A .ASCII 'SECTOR HEX - '
0020 4354 A
0021 4F52 A
0022 2048 A
0023 4558 A
0024 202D A
0025 2020 A
31 0026 0000 A .WORD 0
32          .LOCAL
33          0000  .END

```

```

FLOOP 0000 GT* LESEC 0001 X $BGNP 000F T
BOOT 000E T $END 000C T $GECO 0011 T
MSG 0010 T $T1 0014 T $T2 001E T
X4E 0013 T $X7F 0012 T

```

0 ERROR LINES

ND PASS 4

SOURCE CHECKSUM=88CE

OBJECT CHECKSUM=726F

DISC SECTORS USED

DI0269,DD

ND PASS 1

```

1          .TITLE  BINASC, '24/6/77'
2          0000    .TSECT
3          .GLOBL  BINASC
4          .EXTD
5          ; OMZETTING BINAIR DATAWOORD NAAR ASCII
6          ; UITPRINTEN VAN HET RESULTAAT ALS XXXXX,XXXXX
7 0000 4200 A  BINASC: PUSH 2
8 0001 4300 A      PUSH 3
9 0002 4EFF A      LI 2,-1
10 0003 04B0 A  $Q:  DSUB $HD
    0004 003D T
11 0005 C93C A      ADD 2,$1
12 0006 12FC A      BOC 2,$Q
13 0007 04A0 A      DADD $HD
    0008 003D T
14 0009 4000 A      PUSH 0
15 000A 4100 A      PUSH 1
16 000B 3981 A      RCPY 2,1
17 000C 4C00 A      LI 0,0
18 000D 2909 A      JSR $START
19 000E 2915 A      JSR $PRT1
20 000F 2923 A      JSR $KOM
21 0010 4500 A      PULL 1
22 0011 4400 A      PULL 0
23 0012 2904 A      JSR $START
24 0013 290E A      JSR $PRT0
25 0014 4700 A      PULL 3
26 0015 4600 A      PULL 2
27 0016 0200 A      RTS
28 0017 4FOA A  $START: LI 3,10
29 0018 0490 A  $L:  DIV $10
    0019 0040 T
30 001A CD2A A      ADD 3,$ADR
31 001B A300 A      ST 0,(3)
32 001C DD28 A      SUB 3,$ADR
33 001D 4C00 A      LI 0,0
34 001E DD23 A      SUB 3,$1
35 001F E521 A      SKG 1,$0
36 0020 0200 A      RTS
37 0021 21F6 A      JMP $L
38 0022 4C30 A  $PRT0: LI 0,'0'/256
39 0023 2101 A      JMP .+2
40 0024 4C20 A  $PRT1: LI 0,' '/256
41 0025 CD1C A      ADD 3,$1
42 0026 FD18 A      SKNE 3,$6
43 0027 2101 A      JMP .+2
44 0028 290D A      JSR $ZER
45 0029 DD18 A      SUB 3,$1
46 002A CD17 A  $PRT:  ADD 3,$1
47 002B CD19 A      ADD 3,$ADR
48 002C 8300 A      LD 0,(3)
49 002D DD17 A      SUB 3,$ADR
50 002E C114 A      ADD 0,$030
51 002F 2D14 A      JSR @ $PUTC

```

```

52 0030 ED15 A          SKG      3,$9
53 0031 21F8 A          JMP      $PRT
54 0032 0200 A          RTS
55 0033 4C2C A    $KOM:   LI      0,'',/256
56 0034 2D0F A          JSR     @$PUTC
57 0035 0200 A          RTS
58 0036 3E81 A    $ZER:   RCPY   3,2
59 0037 D907 A          SUB     2,$6
60 0038 A90E A          ST      2,$T
61 0039 2D0A A          JSR     @$PUTC
62 003A 7D0C A          DSZ     $T
63 003B 21FD A          JMP     .-2
64 003C 0200 A          RTS
65 003D 0001 A    $HD:   .WORD  01
66 003E 86A0 A          .WORD  086A0
67 003F 0006 A    $6:   .WORD   6
68 0040 000A A    $10:  .WORD  10
69 0041 0000 A    $0:   .WORD   0
70 0042 0001 A    $1:   .WORD   1
71 0043 0030 A    $030: .WORD  030
72 0044 7E59 A    $PUTC: .WORD  07E59
73 0045 0048 T    $ADR:  .WORD  $TAB
74 0046 0009 A    $9:   .WORD   9
75      0048      $T:   .=.+1
76      0053      $TAB:  .=.+11
77      0000      .END
    
```

```

BINASC 0000 GTX      $0      0041 T      $030  0043 T
$1      0042 T      $10     0040 T      $6     003F T
$9      0046 T      $ADR    0045 T      $HD    003D T
$KOM    0033 T      $L      0018 T      $PRT   002A T
$PRT0   0022 T      $PRT1   0024 T      $PUTC  0044 T
$Q      0003 T      $STAR   0017 T      $T     0047 T
$TAB    0048 T      $ZER    0036 T
    
```

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=1CE2

OBJECT CHECKSUM=EE12

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 0269

FINAL INPUT SECTOR HEX - 026B

FIRST OBJECT SECTOR HEX - 026C

FINAL OBJECT SECTOR HEX - 026C

NEXT ASSEMBLY

C.ASM

END PASS 1

1				.TITLE	BEWEEG
2	0000			.BSECT	
3				.EXTD	
4				.GLOBL	NEEMS,STUURS,AFLOOP
5	0000	0000	A	SNELH:	.WORD 0
6	0001	0080	A	G128:	.WORD 128
7	0002	0000	A	NUL:	.WORD 0
8	0003	0000	A	XTEL:	.WORD 0
9	0004	0000	A	YTEL:	.WORD 0
10	0005	8000	A	G8000:	.WORD X'8000
11	0006	C000	A	GC000:	.WORD X'C000
12	0007	0032	A	X32:	.WORD X'32
13	0008	0034	A	X34:	.WORD X'34
14	0009	007F	A	X7F:	.WORD X'7F
15	000A	0030	A	X30:	.WORD X'30
16	000B	7EC3	A	LMESG:	.WORD X'7EC3
17	000C	7E73	A	LGECO:	.WORD X'7E73
18		000E		HALVV:	.=.+1
19	000E	0000	A	BEELDF:	.WORD 0
20		0000		.TSECT	
21	0000	0D0A	A	TEKST1:	.WORD 0D0A
22	0001	4745	A	.ASCII	'GEWENSTE SNELHEID: '
	0002	5745	A		
	0003	4E53	A		
	0004	5445	A		
	0005	2053	A		
	0006	4E45	A		
	0007	4C48	A		
	0008	4549	A		
	0009	443A	A		
	000A	2020	A		
23	000B	0000	A	.WORD	0
24	000C	2C0B	B	START:	JSR @LMESG
25	000D	0000	T	.WORD	TEKST1
26	000E	2C0C	B	JSR	@LGECO
27	000F	6009	B	AND	0,X7F
28	0010	F008	B	SKNE	0,X34
29	0011	2105	A	JMP	VERVA
30	0012	F007	B	SKNE	0,X32
31	0013	2103	A	JMP	VERVA
32	0014	F00A	B	SKNE	0,X30
33	0015	2101	A	JMP	VERVA
34	0016	21F5	A	JMP	START
35	0017	D00A	B	VERVA:	SUB 0,X30
36	0018	A000	B	ST	0,SNELH
37	0019	8005	B	LD	0,G8000
38	001A	2801	X	JSR	NEEMS
39	001B	2802	X	JSR	STUURS
40	001C	8000	B	LD	0,SNELH
41	001D	5CFF	A	SHR	0,1
42	001E	A00D	B	ST	0,HALVV
43	001F	0707	A	SETST	7
44	0020	4F10	A	LOOP6:	LI 3,16
45	0021	AC04	B	ST	3,YTEL

```

46 0022 8C01 B LOOP5: LD      3,G128
47 0023 2801 X LOOP2: JSR    NEEMS
48 0024 A311 B      ST      0,BEELDP+3(3)
49 0025 4E1F A      LI      2,31
50 0026 2801 X LOOP1: JSR    NEEMS
51 0027 4AFF A      AISZ   2,-1
52 0028 21FD A      JMP    LOOP1
53 0029 4BFF A      AISZ   3,-1
54 002A 21F8 A      JMP    LOOP2
55 002B 8801 B      LD      2,G128
56 002C A803 B      ST      2,XTEL
57 002D 8C03 B LOOP4: LD      3,XTEL
58 002E 0747 A      SKSTF  7
59 002F DC0D B      SUB    3,HALVV
60 0030 EC02 B      SKG    3,NUL
61 0031 CC01 B      ADD    3,G128
62 0032 4E20 A      LI      2,32
63 0033 8311 B LOOP3: LD      0,BEELDP+3(3)
64 0034 2802 X      JSR    STUURS
65 0035 DC00 B      SUB    3,SNELH
66 0036 EC02 B      SKG    3,NUL
67 0037 CC01 B      ADD    3,G128
68 0038 4AFF A      AISZ   2,-1
69 0039 21F9 A      JMP    LOOP3
70 003A 7C03 B      DSZ    XTEL
71 003B 21F1 A      JMP    LOOP4
72 003C 7C04 B      DSZ    YTEL
73 003D 21E4 A      JMP    LOOP5
74 003E 0747 A      SKSTF  7
75 003F 2102 A      JMP    KLAAR
76 0040 0717 A      CLRST  7
77 0041 21DE A      JMP    LOOP6
78 0042 8006 B KLAAR: LD      0,GC000
79 0043 2802 X      JSR    STUURS
80 0044 2803 X      JSR    AFLOOP
81 0045 21C6 A      JMP    START
82      000C      .END   START

AFL00P  0003  X      BEELDP  000E  B      G128    0001  B
G8000   0005  B      GC000   0006  B      HALVV   000D  B
KLAAR   0042  T      LGECO   000C  B      LMSG    000B  B
LOOP1   0026  T      LOOP2   0023  T      LOOP3   0033  T
LOOP4   002D  T      LOOP5   0022  T      LOOP6   0020  T
NEEMS   0001  X      NUL     0002  B      SNELH   0000  B
START   000C  T      STUURS  0002  X      TEKST1  0000  T
VERVA   0017  T      X30    000A  B      X32     0007  B
X34     0008  B      X7F    0009  B      XTEL    0003  B
YTEL    0004  B

```

NO ERROR LINES

END PASS 4

SOURCE CHECKSUM=564B

OBJECT CHECKSUM=46C9

DISC SECTORS USED

FIRST INPUT SECTOR HEX - 023C

FINAL INPUT SECTOR HEX - 023E

FIRST OBJECT SECTOR HEX - 023F