

MASTER

Een full-duplex data-transmissie systeem met foutencorrectie d.m.v. herhaling

Meys, J.A.

Award date:
1975

[Link to publication](#)

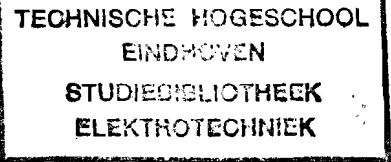
Disclaimer

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



Afstudeerverslag ECB

EEN FULL-DUPLEX DATA-TRANSMISSIE
SYSTEEM MET FOUTENCORRECTIE
D.M.V. HERHALING.

J.A. Meys

Verslag van het afstudeeronderzoek
in de groep Telecommunicatie B.
Uitgevoerd bij:
ir. C.P.J. Schnabel
In de periode van:
februari tot december 1974.

	blz.
Inhoud:	2
0. Trefwoordenlijst	3
1. Inleiding	4
2. Samenvatting	5
3. Afstudeeropdracht	5
4. Bereikte resultaten	8
5. Theoretisch onderzoek	.
5.1. Inleiding	9
5.2. Foutencorrectie-methoden	9
5.3. Optimale bloklengthe	10
5.4. Het code probleem	11
5.4.1. Cyclische codes	12
5.4.2. Detecterende eigenschappen	15
5.4.3. Het generatorpolynoom	19
5.5. Het datatransmissie-systeem	
5.5.1. Algemene principes van de automatische foutencorrectie d.m.v. herhaling	22
5.5.2. Systeem-organisatie	23
5.5.2.1. De normale zendtoestand	23
5.5.2.2. De normale herhalingscyclus	24
5.5.2.3. Bloksynchronisatie en het inschakelen van de datatransmissie	28
5.5.2.4. Het herstellen van de bloksynchro- nisatie	30
5.6. Bitsynchronisatie	32
5.6.1. Pseudo-ternaire codering	32
5.6.2. Digitale modulatie	34
5.6.3. De voor- en nadelen van pseudo-ternaire codering resp. digitale modulatie	36
6. Realisatie van het systeem	38
7. Nabeschuwing	47
8. Literatuur	48
- Flow-diagrammen	49
- Realisatie-schema's	52

0. Trefwoordenlijst:

Datatransmissie.

T.O.R. (=Teletype On Radio).

A.R.Q. (=Automatic Repeat request).

Codes, cyclische -, pseudo-ternaire -.

Optimale bloklengte.

Synchronisatie, blok-, bit-.

Foutendetectie

Digitale modulatie

1. Inleiding:

Wanneer men kijkt naar de systeemstructuur van conventionele rekenmachines, dan valt op, dat door een of meerdere C.P.U.'s (Central Processor Unit) processen worden uitgevoerd. Het koppelen van die processen gebeurt door een ingewikkeld supervisorsysteem: een C.P.U. moet alle verbindingen verzorgen, onder inachtnaam van prioriteiten, waarbij wachttijden optreden bij meerdere aanvragen, daar slechts één device tegelijk een kanaal krijgt toegewezen en het aantal kanalen beperkt is. Ook valt op dat de interne transmissie veelal parallel geschiedt.

De bedoeling is nu: Een nieuw systeem te ontwerpen waarin de processen zoveel mogelijk vanuit de verschillende devices geregeld worden (bv. door gebruik te maken van micro-processoren). Door bij elk device een stuk intelligentie te plaatsen, wordt het in staat gesteld zijn eigen dataverkeer autonoom te regelen. De verbinding met de C.P.U. komt dan grotendeels te vervallen.

Worden alle devices verbonden met een schakelcentrale, waarmee een verbinding kan worden gelegd tussen elk willekeurig tweetal aangesloten devices (welke niet 'in gesprek' zijn), dan zal ook het wachttijdprobleem afnemen, daar nu meerdere verbindingen tegelijk kunnen bestaan.

Door bovendien over te stappen naar serietransmissie zal de benodigde hoeveelheid kabel gereduceerd worden. Dit gaat dan ten koste van de hoeveelheid gebruikte electronica, maar gezien de industriële ontwikkeling is dit minder bezwaarlijk.

Uitgaande van deze filosofie zal een systeem worden beschreven, waarmee datatransmissie tussen de onderlinge devices gerealiseerd kan worden.

2. Samenvatting:

Foutencorrectie door middel van herhaling werd in het begin der jaren vijftig door Dr. van Duuren uitgevonden met als doel de betrouwbaarheid van radiotelegrafie-circuits te verhogen. (1) Deze methode van foutencorrectie wordt naar aanleiding van deze eerste toepassing vaak het T.O.R.-principe genoemd. (T.O.R. = Teletype On Radio). In met name de engelse literatuur worden systemen die van dit principe gebruik maken genoemd: ARQ-systemen (Automatic Repeat request).

Dit verslag behandelt een toepassing van het T.O.R.-principe op datatransmissie tussen twee randapparaten binnen een rekencentrum. Voor zo een toepassing wordt onder inachtnaam van een aantal gestelde eisen een systeem ontworpen. In een volgende fase ontstond een realisatie-ontwerp van het aldus opgezette systeem.

3. Afstudeeropdracht:

Het ontwerpen van een transmissiesysteem waarmee de randapparaten bij een rekenmachine in staat zijn hun datatransmissie autonoom te verzorgen. Daartoe dient een zend-ontvang-eenheid te worden ontworpen die bij elk device geplaatst wordt. Zo een eenheid moet voor het bijbehorende apparaat alle verkeer met een willekeurig ander apparaat zelfstandig kunnen verzorgen: Eerst wordt een duplexverbinding opgezet via een schakelnetwerk en vervolgens wordt in twee richtingen gezonden; foutencorrectie geschiedt door toepassing van het T.O.R.-principe.

Eisen waaraan het systeem verder dient te voldoen:

- 1) Het zenden moet geschieden over een tweemaal 2-draads verbinding; dit impliceert dat klok- en data-informatie in één signaal moeten worden ondergebracht.

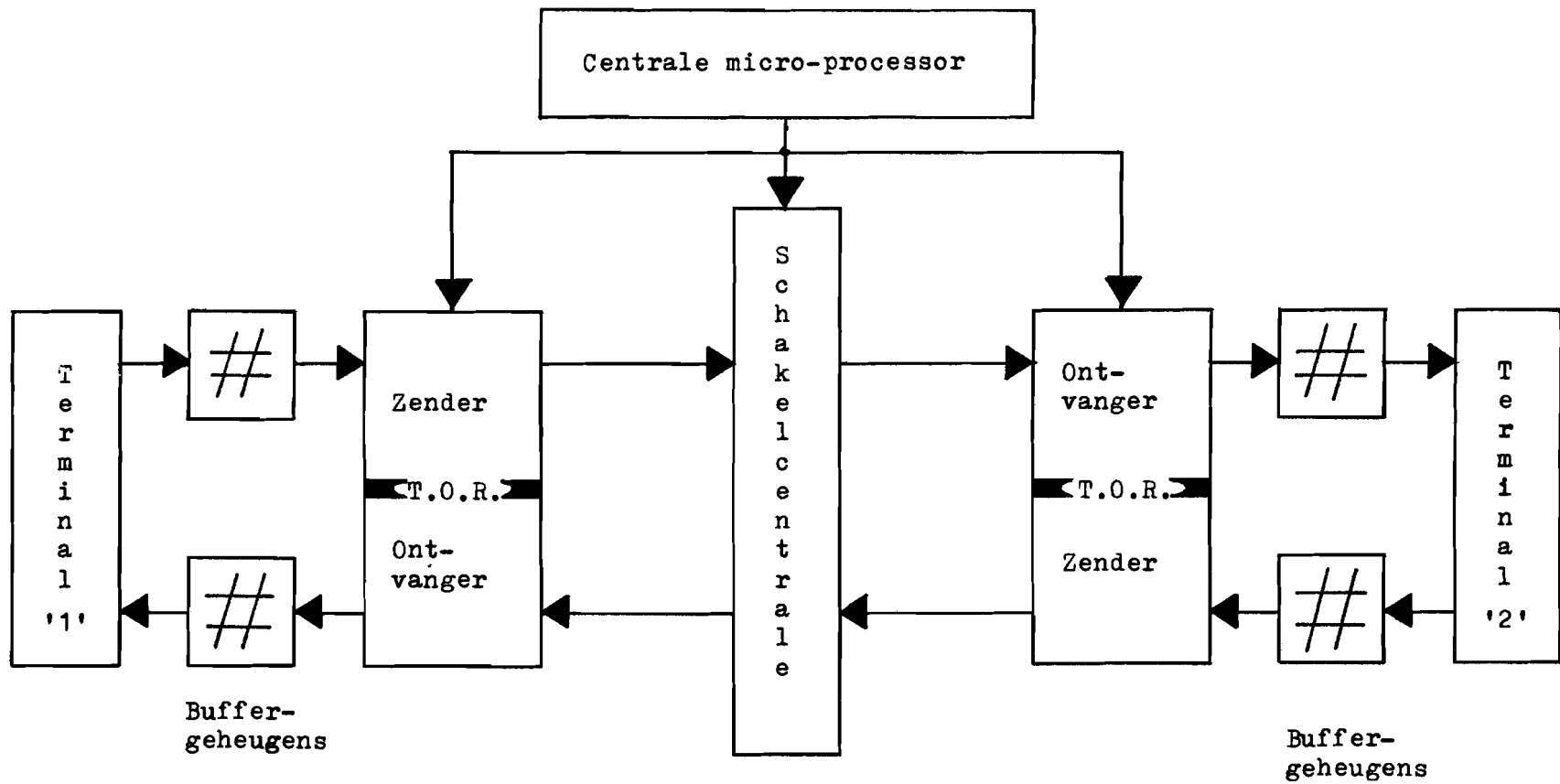


Fig.1: Schematisch overzicht van het systeem.

- 2) Er moet geschakeld kunnen worden met een 'offset'-spanning, hetgeen betekent, dat het over te zenden signaal geen gelijkspanningscomponent mag bevatten.
- 3) Om een goede galvanische scheiding te realiseren dient gebruik te worden gemaakt van optische koppelingen.

Deze drie eisen zijn in hoge mate bepalend voor de keuze van de verschillende bewerkingen die klok- en data-signaal ondergaan alvorens te kunnen worden overgezonden.

- 4) De systeemsnelheid dient zodanig te zijn, dat minimaal een bit-rate van 2 Mbit/sec haalbaar is.
- 5) Ook dient te worden bepaald of het nuttig is om naast foutencorrectie door middel van herhaling nog gebruik te maken van foutencorrigerende codes.

In figuur 1 is een schematisch overzicht gegeven van het totale systeem. De controle bij het schakelnetwerk gebeurt met een centrale micro-processor. Deze ontvangt een oproep en bepaalt of de gewenste verbinding gelegd kan worden. Het verkeer met de centrale kan eveneens via T.O.R.-apparatuur geschieden.

4. Bereikte resultaten:

Bij beëindiging van het afstudeerwerk is een systeemontwerp gereed waarvan het voor dit ontwerp meest specifieke deel in detailschema's is uitgewerkt. Het is de bedoeling dat een proefmodel gebouwd en getest wordt. Tevens zal in een volgend afstudeeronderzoek onderzocht kunnen worden of het mogelijk is dit circuit te integreren.

5. Theoretisch onderzoek:

5.1. Inleiding:

Het is duidelijk dat wanneer digitale datatransmissie geschiedt met behulp van apparatuur waarvan de gemiddelde bitfoutenkans groter is dan de gewenste foutenkans op de systeem-uitgang, een of andere vorm van controle dient te worden toegepast. De bedoeling is dus om, nadat het systeem zoveel mogelijk is geoptimaliseerd, een extra controle uit te oefenen, opdat een voldoende groot percentage van de nog optredende fouten gedetecteerd en gecorrigeerd wordt. (Wat voldoende groot is wordt door de gebruiker bepaald).

5.2. Foutencorrectie-methoden:

De technieken waarbij de benodigde extra controle automatisch wordt uitgevoerd berusten alle op het toevoegen van redundante informatie, aan de hand waarvan in de ontvanger kan worden beslist of het bericht correct dan wel verminkt werd ontvangen. Afhankelijk van de hoeveelheid toegevoegde redundantie is het mogelijk een of meer fouten te corrigeren, fouten te detecteren of een combinatie van detectie en correctie toe te passen. In het geval dat foutendetectie plaatsvindt, zal alleen gecorrigeerd kunnen worden door een vraagom-herhaling naar de zender te sturen en de betreffende informatie te herhalen.

Voor optimale controle bestaat geen vaste regel: Voor elk systeem dient te worden bepaald welke methode(n) moet(en) worden toegepast. Wel is in het verleden bewezen dat redundantie voor foutencorrectie economischer is dan het leggen van een retourverbinding voor herhalingsaanvragen (2),(5). Voor het geval echter dat een retourkanaal toch al aanwezig is, geldt deze stelling niet meer.

Daar in het te ontwerpen systeem duplex transmissie wordt beoogd, hebben wij dus te maken met dit laatste geval.

Voor een beslissing kan worden genomen ten aanzien van de gewenste manier van foutencontrole moeten de statistische ruiseigenschappen van het betreffende kanaal worden geanalyseerd. Daar in de sector van draadverbindingen de foutenkans, die in de circuits verwacht kan worden, relatief laag is, mogen wij aannemen dat slechts een bescheiden hoeveelheid controle noodzakelijk zal zijn. Echter, de controlemethode zou dynamisch moeten zijn, daar de foutenverdeling niet statisch is. De methode zou zich moeten aanpassen aan de schommelende foutenkans en het systeem niet onnodig belasten in perioden dat weinig storingen optreden. Om deze reden zal een vorm van herhalen zeer wenselijk zijn om fouten te corrigeren.

Gezien de bovenstaande beschouwing zal het toevoegen van redundantie voor foutencorrectie niet zo gewenst zijn daar dit een voortdurende verlaging van de overdrachtsefficiency inhoudt. (Onder efficiency wordt verstaan: De tijd nodig voor het verzenden van informatiebits gedeeld door de totale tijd nodig voor het verzenden van informatiebits en redundantiebits plus eventueel de wachttijd voor het goed-fout signaal en de gemiddelde extra tijd voor herhalingen).

5.3. Optimale blok lengte:

Het toevoegen van redundante informatie aan een bericht kan op verschillende manieren gebeuren: Het is mogelijk om elk bit apart te testen en als andere uiterste is er één test over de complete boodschap. Tussen deze twee extremen liggen allerlei andere mogelijkheden. Het is duidelijk dat deze beide uitersten in de meeste gevallen geen optimale efficiency zullen inhouden. Bij het bepalen van een optimale blok lengte dient rekening te worden gehouden met een aantal aspecten. Uit de literatuur (3), (4), (11) is bekend dat de relatie

blok-grootte-efficiency afhangt van een drietal parameters, te weten:

- transmissiesnelheid
- transmissieafstand en
- bitfoutenkans.

De ligging van het optimum voor de efficiency wordt scherper naarmate een of meerdere van deze parameters toenemen.

Voor een systeem zoals eerder besproken:

- bitrate groter dan 2 Mbit/sec
- datatransmissie binnen een rekencentrum (dus een transmissieafstand in de orde-grootte van enkele tientallen meters)
- een lage bitfoutenkans (kleiner dan 10^{-4}),

blijkt dat het aldus gevonden optimum voor de blok-lengte niet zonder meer toegepast kan worden. Het is noodzakelijk om het aantal criteria uit te breiden. De keuze van de blok-grootte zal ook bepaald dienen te worden aan de hand van criteria als: de grootte van zend- en ontvang-buffers, verwerking in de T.O.R.-apparatuur, de vulfactor der blokken (hieronder wordt verstaan de gemiddelde hoeveelheid informatie in een blok. Wordt slechts voor een half blok data aangeboden, dan moet de andere helft met nutteloze bits worden aangevuld), de gemiddelde boodschaplengte enz.

5.4. Het code probleem (6):

Het is bekend, dat willen codes effectief zijn, dan behoren zij lang te zijn. Dan namelijk worden de ruis-effecten over een groot aantal symbolen uitgesmeerd. Van de andere kant echter zijn er eisen die in strijd zijn met lange codes: Eenvoudige verwerking, economische aspecten enz.

De procedure om tot een keuze te komen luidt als volgt:

- 1) Zoek codes die de gewenste detecterende resp. corrigerende eigenschappen bezitten.
- 2) Is er een eenvoudige methode om de informatie te coderen?
- 3) Is het mogelijk om in de ontvanger gemakkelijk een beslissing te nemen, d.w.z. fouten te detecteren resp. te corrigeren?

In de verzameling van alle codes worden twee fundamenteel verschillende typen onderscheiden:

- blokcodes en
- boomcodes.

De encoder voor een blokcode kapt de reeks informatiebits in stukken, blokken, van een vaste lengte. Vervolgens worden deze blokken ieder apart voorzien van een aantal redundante bits en ontstaat een codewoord.

Bij de boomcodes daarentegen wordt de informatie verwerkt zonder deze in stukken te splitsen. De totale reeks wordt in een continu proces omgezet in een code-reeks met meer bits.

Gezien het feit dat bij toepassing van het T.O.R.-principe blokcodes gewenst zijn om de herhaalcyclus correct te kunnen uitvoeren, vallen de boomcodes als eerste af en dient te worden geselecteerd uit de blokcodes.

Beschouwt men de verschillende soorten blokcodes, dan blijken de cyclische codes hierin de belangrijkste plaats in te nemen met betrekking tot datatransmissie. De relatief eenvoudige toepasbaarheid en de eigenschap dat de cyclische codes een zeer duidelijke, wiskundige structuur bezitten, maken deze codes zeer aantrekkelijk. Ook bieden zij zeer goede mogelijkheden wat betreft foutendetectie resp. foutencorrectie. Maar wij zullen alleen kijken naar de detecterende eigenschappen van de cyclische codes, daar wij ons zouden beperken tot foutendetectie met daaropvolgende herhaling.

5.4.1. Cyclische codes:

Cyclische codes zijn erg geschikt voor foutendetectie, want zij kunnen zo ontworpen worden dat veel combinaties van waarschijnlijke fouten worden gedetecteerd. Bovendien zijn encoder en decoder eenvoudig te realiseren. Voor foutendetectie is het aantal checkbits meestal kleiner dan het aantal informatiebits. De encoder kan in dat geval het beste gerealiseerd worden met een schuifregister ter lengte van het aantal checkbits. Eenzelfde ontwerp kan in de ontvanger gebruikt worden

voor de detectie van pariteitsfouten. Wat moet gebeuren is: De ontvangen pariteitsbits aftrekken van de in de ontvanger gecalculeerde reeks. Wanneer de uitkomst nul is, dan is het ontvangen blok een codewoord, anders niet.

Om een en ander toe te lichten zal nu in het kort de wiskundige opbouw van de cyclische codes worden besproken. Een codewoord van n bits bestaat uit k informatiebits en $(n-k)$ checkbits. Om de code wiskundig te doorgronden is het praktisch de bits te bezien als de coëfficiënten van een polynoom in de hulpgrootte X :

Voorbeeld:

$$\begin{aligned} 100111011 &:= 1 \cdot X^0 + 0 \cdot X^1 + 0 \cdot X^2 + 1 \cdot X^3 + 1 \cdot X^4 + 1 \cdot X^5 + 0 \cdot X^6 + 1 \cdot X^7 + 1 \cdot X^8 \\ &= 1 + X^3 + X^4 + X^5 + X^7 + X^8 \end{aligned}$$

Bij het werken met deze polynomen gelden de gewone rekenregels met dien verstande dat de optelling modulo-2 gebeurt. Verder blijven alle eigenschappen gehandhaafd en zal een polynoom slechts op één manier te ontbinden zijn in primitieve en irreducibele factoren.

Een cyclische code wordt gedefiniëerd met behulp van een generatorpolynoom $G(X)$ van de graad $(n-k)$. Een polynoom van een graad kleiner dan n is een code-polynoom, d.w.z. komt voor transmissie in aanmerking, dan en slechts dan wanneer het deelbaar is door $G(X)$. Tengevolge van deze definitie geldt dat de som van twee code-polynomen een nieuw code-polynoom is: De som van twee polynomen van graad kleiner dan n is weer een polynoom van graad kleiner dan n .

Wanneer $G(X) = X \cdot G'(X)$ dan zullen alle code-polynomen de factor X^1 bevatten en de nul-coëfficiënt (X^0) zal geheel overbodig zijn. Er zal dan ook een generatorpolynoom gekozen worden dat niet deelbaar is door X .

Om een boodschap $M(X)$ te coderen wordt $G(X)$ gedeeld op $X^{(n-k)} \cdot M(X)$ en de restterm $R(X)$ bepaald:

$$\frac{X^{(n-k)} \cdot M(X)}{G(X)} = Q(X) + \frac{R(X)}{G(X)}$$

($Q(X)$ is het quotiënt en $R(X)$ is de restterm).

Hiervoor kan geschreven worden:

$$V(X) = X^{(n-k)} \cdot M(X) + R(X) = Q(X) \cdot G(X).$$

$V(X)$ is een veelvoud van $G(X)$ en dus een code-polynoom. Voorts zal $R(X)$ van een graad kleiner dan $(n-k)$ zijn en heeft $X^{(n-k)} \cdot M(X)$ nul als coëfficiënten voor de $(n-k)$ laagste-orde termen. De k hoogste-orde coëfficiënten zullen dus gelijk zijn aan de coëfficiënten van $M(X)$ en de $(n-k)$ laagste-orde coëfficiënten zijn de coëfficiënten van $R(X)$ en dat zijn de checkbits.

Voorbeeld:

Stel een code heeft: $n=11$, $k=7$, $G(X)=1+X+X^4$.

Het te coderen 7-bits bericht is: 1011010.

Analoog aan het bovenstaande vinden wij:

$$V(X) = (X+X^2+X^3) + X^4(1+X^2+X^3+X^5) \\ R(X) + X^{(n-k)} \cdot M(X)$$

Hiermee komt overeen de bitreeks: $\underbrace{0111}_{\text{checkbits}}, \underbrace{1011010}_{\text{informatiebits}}$

Een gecodeerde boodschap die fouten bevat kan als volgt worden voorgesteld:

$$H(X) = V(X) + E(X),$$

waarin $V(X)$ de correcte boodschap is en $E(X)$ een foutenpolynoom dat coëfficiënten één heeft voor alle termen die fout zijn. Wanneer $H(X)$ niet deelbaar is door $G(X)$, dan is klaarblijkelijk een fout opgetreden. Is $H(X)$ wél deelbaar door $G(X)$, dan moet het codewoord geaccepteerd worden. Daar $V(X)$ zó was gemaakt dat het deelbaar is door $G(X)$, zal $H(X)$ slechts dan deelbaar zijn door $G(X)$ wanneer $E(X)$ dat is. Een foutenpatroon zal dan ook slechts te detecteren zijn wanneer het bijbehorende foutenpolynoom $E(X)$ niet deelbaar is door $G(X)$. Om een effectieve controle te kunnen uitoefenen zal $G(X)$ zodanig gekozen moeten worden dat de foutenpatronen die wij willen detecteren niet deelbaar zijn door $G(X)$.

Om fouten te detecteren delen wij het ontvangen woord $H(X)$ door $G(X)$ en kijken naar de restterm. Is deze ongelijk aan nul, dan is een fout gedetecteerd. Is de restterm nul, dan is er geen fout of een niet gedetecteerde fout.

Voorbeeld:

$$V(X) = X+X^2+X^3+X^4+X^6+X^7+X^9 = 01111011010$$

$$E(X) = X^3+X^4+X^8 = 00011000100$$

$$H(X) = X+X^2+X^6+X^7+X^8+X^9 = 01100011110$$

Dit is de $V(X)$ uit het vorige voorbeeld. $H(X)$ gedeeld door $G(X)$ levert als rest: $X+X^2+X^3$. Dit is niet nul, en dus is er een fout gedetecteerd. Daar $V(X)$ deelbaar is door $G(X)$ zal, wanneer $E(X)$ gedeeld wordt door $G(X)$, dezelfde rest gevonden moeten worden.

5.4.2. Detecterende eigenschappen:

Een cyclische code van een generatorpolynoom $G(X)$ met meer dan een term detecteert alle enkele fouten.

Bewijs: Een enkele fout komt overeen met $E(X)=X^i$. Om te zorgen dat dit soort fouten steeds te detecteren is, moet $G(X)$ niet deelbaar zijn door X^i . Hieraan is steeds voldaan als $G(X)$ uit meer dan een term bestaat.

Elk polynoom deelbaar door $(1+X)$ heeft een even aantal termen.

Bewijs: Stel $V(X)=X^a+X^b+X^c+\dots=(1+X)\cdot Q(X)$.

Substitutie van $X=1$ levert:

$$V(1)=1+1+1+\dots=(1+1)\cdot Q(1)=0.$$

Voor elke term in $V(X)$ staat een één en de som is nul, dus moet het aantal termen even zijn. Conclusie is nu dat elke $G(X)$ die een factor $(1+X)$ en meer algemeen een factor $(1+X^c)=(1+X)(X^{c-1}+X^{c-2}+\dots+1)$ bevat, een code genereert die alle oneven foutenaantallen detecteert.

Men zegt in dit verband dat een polynoom $G(X)$ tot een exponent e behoort als e de kleinste positieve integer is, zó dat $G(X)$ factor is van X^e+1 .

Een code gegenereerd door het polynoom $G(X)$ detecteert alle enkele en dubbele fouten, als de lengte van de code niet groter is dan de exponent e waartoe $G(X)$ behoort.

Bewijs: Detectie van alle dubbele fouten houdt in dat $G(X)$ geen factor mag zijn van (X^i+X^j) voor willekeurige $i, j < n$. (X^i+X^j) kan als volgt worden ontbonden (stel i kleiner dan j): $X^i(1+X^{j-i})$. Het is nu voldoende te eisen dat $G(X)$ geen factor is van $(1+X^{j-i})$, want er is reeds gesteld dat $G(X)$ niet deelbaar zou zijn door X .

Daar $G(X)$ behoort tot de exponent e en verder geldt dat $(j-i) < n \leq e$, kan $G(X)$ geen factor zijn van $(1+X^{j-i})$. De code zal dus alle enkele en dubbele fouten detecteren.

Aangetoond kan worden dat voor elke m tenminste één polynoom $G(X)$ van graad m bestaat, dat behoort bij de exponent $e=2^m-1$. Dit is de grootst mogelijke waarde voor e . Polynomen met deze eigenschap zijn steeds irreducibel. Dus voor elke m is er een dubbele-fouten detecterende code ter lengte $n=2^m-1$, gegenereerd door een polynoom van de graad m , die daartoe m checkbits en $(n-m)=2^m-1-m$ informatiebits heeft. Deze codes zijn volledig equivalent aan de enkele fouten-corrigerende Hamming-codes.

Een code gegenereerd door $G(X)=(1+X).G_1(X)$ detecteert alle enkele, dubbele en drievoudige fouten, wanneer de lengte van de code niet groter is dan de exponent e van $G_1(X)$.

Dit is een samenstelling van twee eigenschappen.

Maximum-lengte codes ontstaan als $G_1(X)$ een priempolynoom is en deze codes zijn equivalent aan de enkele-fouten corrigerend, dubbele-fouten detecterende Hamming-codes.

Burst-detecterende eigenschappen:

Onder een foutenburst ter lengte b wordt verstaan: Elk foutenpatroon waarvoor het aantal symbolen vanaf de eerste tot

en met de laatste fout b is.

Voorbeeld: $E(X) = X^2 + X^6 + X^9 = 00 \mid 10001001 \mid 000$

Elke cyclische code gegenereerd door een polynoom van de graad $(n-k)$ detecteert elke burst met een lengte $\leq (n-k)$.

Bewijs: Elk burstpolynoom kan worden ontbonden in de vorm:

$$E(X) = X^i \cdot E_1(X),$$

waarbij $E_1(X)$ graad $(b-1)$ bezit. Deze burst is detecteerbaar wanneer $G(X)$ geen factor is van $E(X)$. Daar $G(X)$ geen factor X zal bevatten, komt dit neer op: $G(X)$ is geen factor van $E_1(X)$. Maar als $b \leq (n-k)$, dan is de graad van $G(X)$ groter dan de graad van $E_1(X)$ en dus nooit factor van $E_1(X)$.

Bovendien zal een groot aantal van de langere bursts gedetecteerd worden: De fractie bursts met een lengte $b > (n-k)$, die niet gedetecteerd worden, is:

$$2^{-(n-k)} \text{ als } b > n-k+1 \text{ en}$$

$$2^{-(n-k-1)} \text{ als } b = n-k+1.$$

Het foutenpatroon is:

$$E(X) = X^i \cdot E_1(X),$$

waarbij $E_1(X)$ graad $(b-1)$ bezit. Omdat $E_1(X)$ de termen X^0 en $X^{(b-1)}$ bevat, zijn er $(b-2)$ termen X^j met $0 < j < (b-1)$, die nul of één als coëfficiënt kunnen hebben. Dus zijn er $2^{(b-2)}$ verschillende mogelijkheden voor $E_1(X)$. Een burst is niet detecteerbaar als $E_1(X)$ deelbaar is door $G(X)$:

$$E_1(X) = G(X) \cdot Q(X).$$

Daar $G(X)$ graad $(n-k)$ heeft, moet $Q(X)$ graad $(b-1)-(n-k)$ hebben. Is $(b-1) = (n-k)$, dan is $Q(X) = 1$ en is er slechts één $E_1(X)$ die resulteert in één ongedetecteerde fout, nl.

$$E_1(X) = G(X).$$

De verhouding tussen het totale aantal bursts met $b=(n-k+1)$ en de ongedetecteerde bursts van die lengte is dus:

$$\frac{1}{2^{(n-2)}} = 2^{-(n-k-1)}.$$

Wanneer $(b-1) > (n-k)$ dan bezit $q(X)$ de termen X^0 en $X^{(b-1)-(n-k)}$ en verder $2^{(b-2)-(n-k)}$ mogelijkheden voor de tussenliggende termen. De verhouding wordt in dit geval:

$$\frac{2^{(b-2)-(n-k)}}{2^{(b-2)}} = 2^{-(n-k)}.$$

Samenvattend kan gezegd worden dat een code met $(n-k)$ checkbits en k informatiebits in staat is alle mogelijke fouten te detecteren behalve de fractie:

$$\frac{2^k}{2^n - 1} \approx 2^{-(n-k)}.$$

Een code ter lengte n heeft $2^n - 1$ mogelijke foutenpatronen. Door k informatiebits zijn 2^k correcte codewoorden bepaald en de rest zijn illegale woorden en kunnen als zodanig gedetecteerd worden. Dit laatste betekent, dat onafhankelijk van de codelengte en de ruisverschijnselen op het kanaal, de kans op een ongedetecteerde fout, P_e , evenredig is met $2^{-(n-k)}$. P_e wordt dus primair bepaald door het aantal checkbits. Dientengevolge zal in het geval dat volstaan wordt met detectie, het aantal checkbits niet al te groot hoeven te zijn. Stel bv. $(n-k)=20$: Het aantal niet-detecteerbare fouten is dan globaal een miljoenste deel van het totale aantal fouten. Wanneer een generatorpolynoom gekozen wordt dat alle enkele, dubbele en drievoudige fouten zeker detecteert (voor codelengtes kleiner dan de bijbehorende exponent), dan zit dat miljoenste deel in de verzameling van hoger-orde fouten (meer dan drie bits fout). Uitgaande van een random bitfoutenkans is dan meteen duidelijk dat de kans op deze

fouten zeer klein wordt.

Voorbeeld:

Bitfoutenkans $P_b = 10^{-4}$; de kans dat 4 bits fout zijn is dan:

$$P(4 \text{ bits fout}) = (P_b)^4 = 10^{-16}.$$

5.4.3. Het generatorpolynoom:

Gekozen is voor het volgende 16de-grads generatorpolynoom:

$$\begin{aligned} G(X) &= 1 + X^5 + X^{12} + X^{16} \\ &= (1+X)(1+X+X^2+X^3+X^4+X^{12}+X^{13}+X^{14}+X^{15}) \\ &= (1+X) \cdot G_1(X). \end{aligned}$$

De factor $(1+X)$ zorgt voor detectie van alle oneven foutenaantallen. Bovendien worden alle dubbele fouten gedetecteerd zolang de codelengte n kleiner is dan de exponent van $G_1(X)$. Aangezien $G_1(X)$ irreducibel is (zie tabel in (6)), geldt: De exponent is gelijk aan $(2^{15}-1)$ als dit laatste een priemgetal is (Mersenne priemgetallen). Is het geen priemgetal dan zal de exponent gelijk zijn aan een der factoren waarin $(2^{15}-1)$ kan worden ontbonden, én welke geen factor is van een der getallen (2^s-1) voor $s < 15$:

$$2^{15}-1 = 32\,767 = 7 \times 31 \times 151.$$

$$7 = 2^3-1 \text{ en } 31 = 2^5-1.$$

Maar 151 is geen factor van een der mogelijke getallen 2^s-1 en dus de exponent van $G_1(X)$.

Conclusie: Met dit generatorpolynoom worden alle enkele, dubbele en drievoudige fouten gedetecteerd als de code-lengte kleiner dan 151 is.

De realisatie van encoder zowel als decoder kan nu geschieden met behulp van schuifregisters, zoals in de figuren 2 en 3 is

aangegeven.

Encoder:

De k informatiebits worden door het 16-bits register geklokt. Wat in feite gebeurt, is: $X^{(n-k)} \cdot M(X)$ delen door $G(X)$. Na k bits staat de restterm $R(X)$ in het register. Door zodanig te schakelen dat de eerste k bits geldt:

E_a en E_b zijn hoog en E_c is laag,

en voor de volgende 16 bits:

E_a en E_b laag en E_c hoog,

verschijnt op de uitgang de gecodeerde boodschap:

$$V(X) = X^{16} \cdot M(X) + R(X).$$

Decoder:

In de decoder wordt van de eerste k bits de restterm bepaald (delen door $G(X)$) en vervolgens wordt deze afgetrokken (is gelijk aan optellen, daar modulo-2 gewerkt wordt) van de ontvangen $R(X)$. Is na n bits het register leeg (alles nul), dan wordt een goedmelding verzorgd en anders een foutmelding. De goed-fout-melding geschiedt dus door een 'alles-nul' test aan het einde van elk blok.

Deze handelingen worden gerealiseerd door te zorgen dat gedurende de eerste k bits geldt:

E_a , E_b en E_e zijn hoog en E_d is laag,

en voor de volgende 16 bits:

E_a en E_e zijn hoog, E_b en E_d zijn laag.

E_d is hoog in de fase dat bloksynchronisatie verzorgd moet worden. E_a, E_b en E_e zijn dan laag. Het register kan dan het herkenningspatroon detecteren (zie 5.5.2.3.).

5.5. Het datatransmissie-systeem:

5.5.1. Algemene principes van de automatische foutcorrectie door middel van herhaling:

Het is nuttig om nog even naar de principes van deze correctiemethode te kijken alvorens het systeem in detail te gaan beschouwen. De verbindingen waarmee gewerkt wordt zijn 'gehuurde' verbindingen, d.w.z. een verbinding hoeft alleen te bestaan als er data moet worden overgezonden; de verbinding wordt verbroken als de beide abonnees hun boodschap beëindigd hebben.

Het systeem heeft twee terminals nodig: Een genaamd de 'master' (initiatiefnemer) en de ander de 'slave' (adres). De abonnee-apparatuur bevat een zendend en een ontvangend gedeelte (fig.1). Om een boodschap over te sturen wordt deze door de zender gecodeerd en onder toezicht van een controle-eenheid uitgezonden. Wordt in de ontvanger een blok niet geaccepteerd, dan wordt de aanvoer van nieuwe blokken stopgezet. Ook wordt de ontvanger voor bepaalde tijd gesloten en zodoende de abonnee helemaal losgekoppeld van de T.O.R.-apparatuur. Deze verzorgt nu een foutmelding, gevolgd door de laatstgezonden N blokken. Hoeveel blokken herhaald moeten worden hangt af van de propagatietijd van zender naar ontvanger en terug: Er is dus een geheugen nodig waarin de laatste N blokken steeds in de juiste volgorde aanwezig zijn. Bij ontvangst van een foutmelding in de tweede terminal wordt ook daar alle locale verkeer stopgezet en op dezelfde manier overgegaan op het herhalen van de laatste N blokken. De transmissie van deze herhalingscycli gaat in beide richtingen door totdat een volledige cyclus op de juiste manier wordt ontvangen in de ontvanger die de fout ontving. Dan wordt daar de toevoer hersteld en het verkeer van en naar de abonnee begint weer. De tweede terminal herstart even later, wanneer geen foutmelding meer wordt ontvangen.

De datatransmissie tussen twee abonnees gebeurt synchroon: Hieronder wordt verstaan dat de locale timing in de beide terminals refereert aan een moederklok. De moederklok zal, indien niet anders vermeld, steeds de masterklok zijn. Voor de manier waarop dit gerealiseerd kan worden zij verwezen naar het hoofdstuk 'Bitsynchronisatie'.

5.5.2. Systeem-organisatie:

De belangrijkste logische problemen die bij een full-duplex, synchroon systeem met blokcodering en met correctie door middel van herhaling, optreden zijn (7):

- De normale herhalingscyclus;
- Bloksynchronisatie en het overschakelen naar de datafase;
- Correct handelen wanneer de bloksynchronisatie verloren is gegaan: Hersynchronisatie.

Wij zagen, dat om te weten of herhaald moet worden, een foutmelding via de retourverbinding dient te worden ontvangen. Daar duplex verkeer bedreven wordt, zal deze foutmelding in de blokken moeten worden ingepast. De eenvoudigste manier om dit te doen is een apart bit in elk blok hiervoor te gebruiken. Daar bleek dat het systeem geen onderscheid maakt tussen een fout blok en een blok met een foutmelding, is besloten een foutmelding te creëren door het blok opzettelijk fout te maken. Dit kan door een of meerdere pariteitsbits te inverteren. Beschouwen wij vervolgens de verschillende toestanden waarin het systeem kan verkeren.

5.5.2.1. De normale zendtoestand:

Dit is de toestand waarin de beide zenders continu nieuwe informatie verzenden, die dus steeds zonder fout ontvangen

wordt. Het verkeer in de beide richtingen oefent geen invloed op elkaar uit. Deze toestand kan op twee manieren worden doorbroken, doordat:

- 1) De beide zenders hun boodschap beëindigd hebben. De verbinding wordt dan geheel verbroken.
- 2) Een der ontvangers een fout detecteert. Dan zal worden overgegaan tot een normale herhalingscyclus.

5.5.2.2. De normale herhalingscyclus:

Een herhalingscyclus wordt voorafgegaan door de ontvangst van een foutmelding. Dit wordt doorgegeven naar de zender. De zender plaatst een foutmelding in het blok dat net wordt uitgezonden en aan het eind van dat blok wordt de herhaalcyclus gestart en de toevoer van nieuwe informatie gestopt. De zender gaat nu de laatste N blokken, die in het herhaalgeheugen staan, als volgt herhalen:

- De eerste (N-1) blokken met een goedmelding (dit betekent dat deze blokken zonder meer correct worden verzonden);

- Bij het N-de blok (dit is het blok waarin de foutmelding geplaatst werd) wordt weer beslist aan de hand van de door de ontvanger geleverde informatie: Dit kan leiden tot drie verschillende voortzettingen, nl.:

- 1) De ontvanger vraagt niet meer om herhaling en er wordt teruggeschakeld naar de normale zendtoestand;
- 2) De ontvanger heeft weer een fout gedetecteerd en een volgende herhaalcyclus zal moeten worden verzorgd;
- 3) De ontvanger geeft weer een foutmelding door, maar het aantal opeenvolgende herhaalcycli is zó groot, dat verondersteld moet worden dat de bloksynchronisatie verloren is gegaan. In dit geval wordt automatisch overgeschakeld naar de hersynchronisatieprocedure.

Voorbeeld voor N=3:

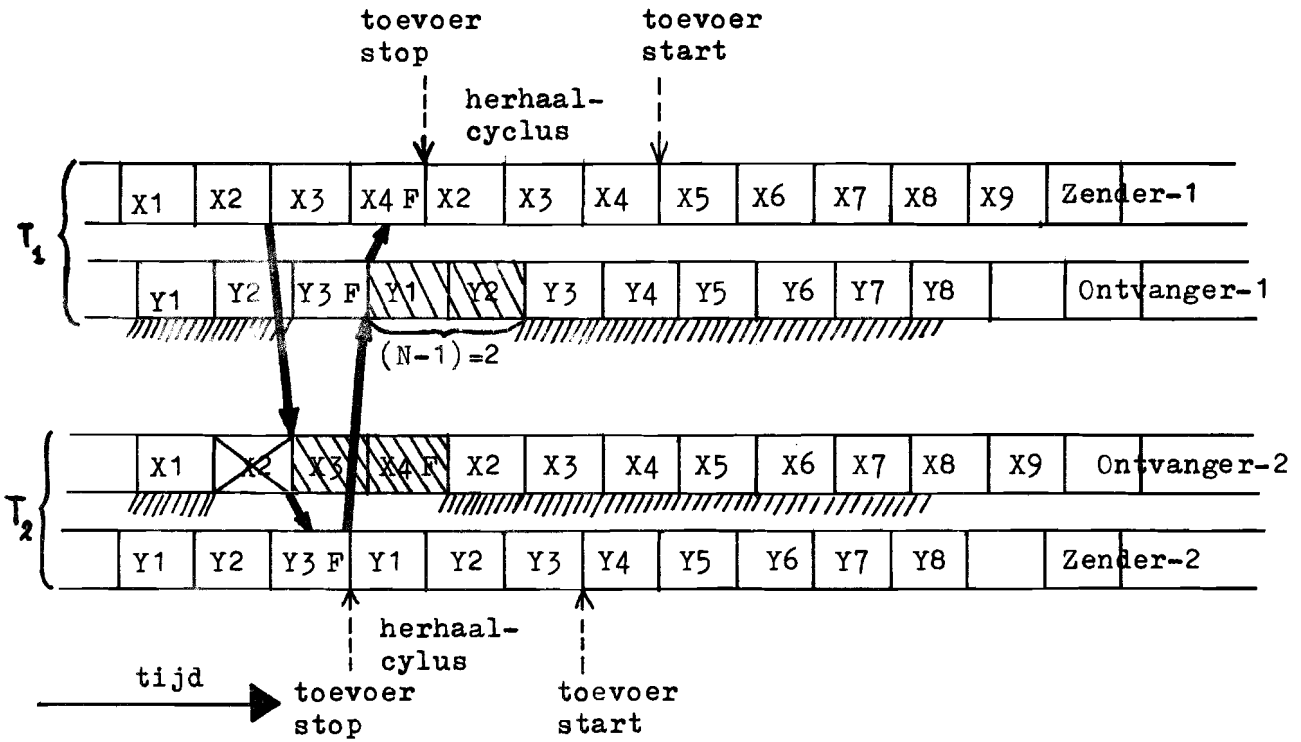


Fig.4: Herhaalcyclus voor N=3.

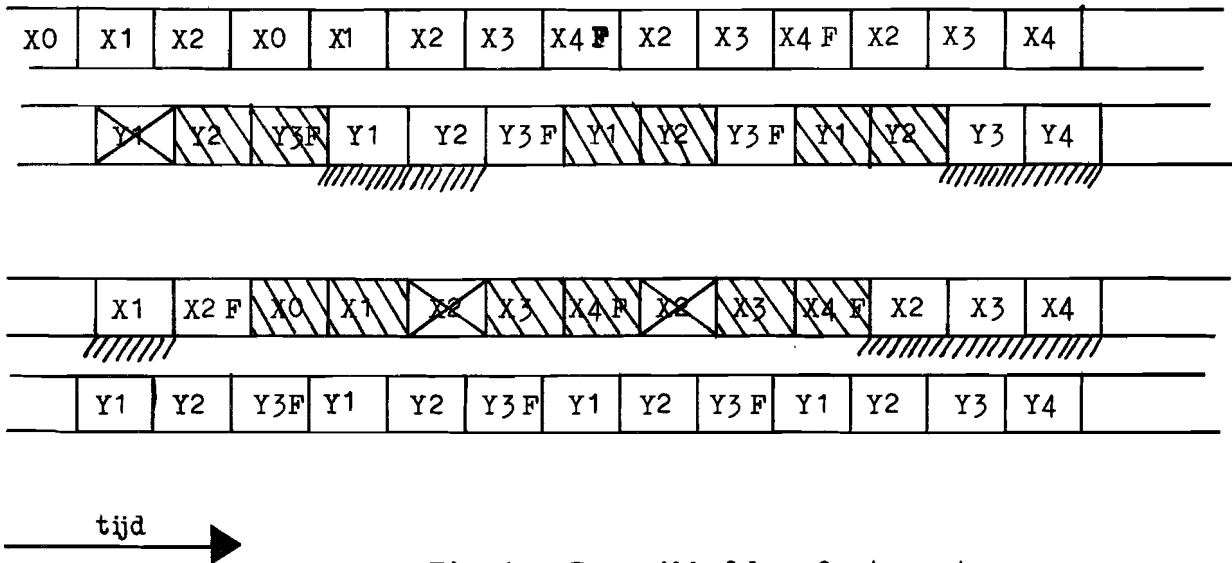


Fig.4a: Ingewikkelder foutenpatroon voor N=3.

Voorbeeld voor $N=3$ (zie fig.4):

T_1 zendt de symbolen X uit en T_2 de symbolen Y. De transmissie in de beide richtingen verloopt normaal, totdat X_2 verminkt wordt ontvangen door ontvanger-2. Ontvanger-2 geeft dit door naar zender-2 en symbool Y_3 wordt verzonden met een foutmelding. Ontvanger-2 beoordeelt de eerstvolgende twee ($=N-1$) blokken niet (gearceerd).

Ontvanger-1 zal wanneer Y_3 met een foutmelding wordt ontvangen, dit doorgeven naar zender-1. Symbool X_4 wordt dan met een foutmelding verzonden. Ook hier beoordeelt de ontvanger de eerstvolgende twee blokken niet.

De beide zenders gaan, wanneer zij de foutmelding in het eerstvolgende blok geplaatst hebben, over tot een herhaalcyclus. Daar T_2 bij het laatste blok van de cyclus, Y_3 , geen foutmelding meer hoeft te plaatsen (X_2 is nu correct ontvangen), wordt na Y_3 de toevoer weer gestart. T_1 zal even later hetzelfde doen, daar ook hier symbool Y_3 nu goed wordt ontvangen.

De blokken die naar de abonnees worden doorgegeven zijn aan de onderkant voorzien van een arcering. Er worden geen blokken dubbel afgeleverd dan wel overgeslagen. Dit kan ook voor meer ingewikkelde fouten worden aangetoond met behulp van eenzelfde diagram (fig.4a).

De horizontale verschuivingen in het diagram duiden de optredende looptijden en blokfase aan.

De pijlen geven een 'gesloten lus' aan: Een fout vanuit T_1 komt in T_2 en wordt weer teruggemeld naar T_1 .

Looptijden en blokfase:

De maximaal toelaatbare looptijd waarover een systeem nog correct zal werken is afhankelijk van het aantal herhaalde blokken N en van de relatieve blokfase tussen de twee zendrichtingen. Stel de blokduur is t_b seconden en de foutmelding heeft plaats t_c seconden voor het einde van het blok. Een lus zal dan gesloten dienen te zijn binnen $((N-1) \cdot t_b - t_c)$ seconden, daar anders blokken verloren kunnen gaan. Zolang dit geldt zal namelijk het blok dat fout was overgekomen nog in het herhaalgeheugen aanwezig zijn (fig.5).

Van deze maximale lustijd gaat een deel verloren aan de blokfase. Wat overblijft is de tijd beschikbaar voor het verzenden in de twee richtingen. De blokfase is optimaal wanneer een ontvangen foutmelding meteen kan worden doorgespeeld naar het blok dat op dat moment verzonden wordt. De tijd beschikbaar voor het verzenden is dan maximaal, nl.

$((N-1) \cdot t_b - 2t_c)$. Een tijdsduur t_c gaat verloren, omdat de foutmelding geplaatst wordt in een blok t_c seconden van het einde, terwijl detectie van de foutmelding pas aan het einde van het blok geschiedt.

Is de blokfase ongunstig, dan zal de foutmelding pas in het volgende blok geplaatst kunnen worden. In het meest ongunstige geval (net te laat voor plaatsing in het blok waar de zender mee bezig is!) zal de tijd beschikbaar voor het verzenden t_b seconden kleiner worden: $((N-2) \cdot t_b - 2t_c)$.

Wanneer het mogelijk is de blokfase te controleren, dan kan in het gunstigste geval worden volstaan met $N=2$. Is deze controle niet mogelijk, dan moet rekening worden gehouden met de slechtste toestand en dient het herhaalgeheugen met één blok te worden uitgebreid.

Daar bij een verbinding via een schakelnetwerk het aanwijzen van een blokfase-regelende terminal moeilijk is, omdat elke abonnee het initiatief kan nemen om een verbinding te leggen, zal moeten worden uitgegaan van random blokfase. Gezien de aard van de te realiseren verbindingen en de te verwachten korte looptijden, wordt gewerkt met een drie-bloks herhaalgeheugen, hetgeen dus het minimum is onder deze voorwaarden.

Het inpassen van de foutmelding:

Wij zagen dat een foutmelding gecreëerd kan worden door het inverteren van een of meer pariteitsbits. Daar bij de cyclische code deze pariteitsbits de laatste 16 bits van een blok zijn, kan dit helemaal op het einde gebeuren. In eerste instantie is besloten te volstaan met het inverteren van één bit. Indien wenselijk kunnen meerdere bits geïnverteerd worden.

Door deze manier van foutmelden wordt de tijdsduur t_c van

Foutmelding tot einde blok zeer klein. Dit ten gunste van de maximaal toelaatbare lusvertraging.

5.5.2.3. Bloksynchronisatie en het inschakelen van de datatransmissie:

Voor gepraat kan worden over bloksynchronisatie zal eerst de bitsynchronisatie verzorgd dienen te worden. Het probleem van de bitsynchronisatie wordt elders in dit verslag behandeld en zal hier verder aanwezig worden verondersteld.

Aan de bloksynchronisatie zijn dan een aantal eisen gesteld:

- Bloksynchronisatie moet goed tot stand komen, ook wanneer tijdens deze fase storingen optreden;
- De synchronisatie-informatie mag niet naar de abonnee worden doorgegeven;
- Alle datablokken moeten door middel van herhaling gecorrigeerd kunnen worden.

Verder is het wenselijk dat geen extra bits in het blok nodig zijn voor volgnummers en ook, dat geen voorwaarden zijn gesteld aan de toegestane bitpatronen.

Beschouwen wij nu de logische procedure die aan deze eisen voldoet en die steeds zal leiden tot een correcte start van de datatransmissie. Daar uitgegaan moet worden van random blokfase, is niet bekend welke terminal als eerste zal synchroniseren. Tijdens de bloksynchronisatie kunnen nog geen normale herhaalcycli verzorgd worden. Er moet dus voor gezorgd worden dat de beide terminals nagenoeg tegelijkertijd (binnen t_b seconden) overschakelen naar de datafase, waarin zij beide in staat zijn een normale herhaalcyclus te verzorgen. Gebeurt dit niet, dan is het mogelijk dat de ene terminal in de datafase verkeert en de andere niet. Eventueel door deze tweede terminal ontvangen foute blokken resulteren dan niet in een herhaalcyclus en dit kan tot het verdwijnen van informatie leiden.

Voor de bloksynchronisatie wordt gebruik gemaakt van twee typen blokken:

- Synchronisatieblok: Dit blok bevat een synchronisatiepatroon op een vaste plaats en kan met een goed- dan wel met een foutmelding verzonden worden;
- Startblok: Dit blok geeft aan dat de volgende blokken data bevatten. Het zenden van het startblok introduceert ook de mogelijkheid tot het plegen van een normale herhalingscyclus. Goede ontvangst van het startblok geeft tevens aan dat vanaf dat moment goede blokken naar de abonnee moeten worden doorgegeven.

Gehandeld wordt nu als volgt: De beide zenders gaan synchronisatieblokken met een foutmelding zenden. Op het moment dat het synchronisatiepatroon in een ontvanger wordt herkend, is in die terminal bloksynchronisatie aanwezig. De bijbehorende zender gaat over op het zenden van synchronisatiecycli, d.w.z. zolang er synchronisatieblokken met een foutmelding worden ontvangen zendt hij: Eén synchronisatieblok met een foutmelding en twee ($=N-1$) synchronisatieblokken met een goedmelding. De tweede ontvanger zal wanneer bloksynchronisatie wordt bereikt door herkenning van het synchronisatiepatroon, eveneens de bijbehorende zender doen overgaan op het zenden van zulke cycli. Gedurende zo'n cyclus wordt niet gelet op een goed- of foutmelding in de ontvangen blokken.

Om nu vanuit deze toestand tot een correcte start te komen, heeft elke ontvanger een teller nodig: Deze wordt telkens met één opgehoogd bij ontvangst van een correct synchronisatieblok; een fout synchronisatieblok zorgt voor een reset van de teller in de stand nul. Heeft een zender een synchronisatiecyclus beëindigd, dan wordt een nieuwe cyclus begonnen als de bijbehorende teller in de nulstand staat; is de tellerstand niet nul, dan gaat de zender verder met het zenden van goede synchronisatieblokken.

Een startblok wordt gezonden wanneer aan twee voorwaarden is voldaan:

- De zender is niet met een synchronisatiecyclus bezig óf heeft deze net beëindigd, én
- De tellerstand is groter of gelijk aan twee ($=N-1$).

Ook wordt op dit moment de mogelijkheid voor een herhalingscyclus ingeschakeld. Beide terminals kunnen een herhaalcyclus verzorgen en bovenstaande moeilijkheid is opgelost. Herkenning van het startblok verleent tevens de toestemming om blokken naar de abonnee door te geven, wanneer deze correct zijn. Dit om te voorkomen dat synchronisatieblokken worden doorgegeven. Na het zenden van het startblok bevindt het systeem zich in de normale zendtoestand (fig.6).

5.5.2.4. Het herstellen van de bloksynchronisatie:

Wanneer een storing tot gevolg heeft dat de bloksynchronisatie verloren gaat, zal het systeem in een toestand komen waarin voortdurend herhaalcycli plaatsvinden. Om hierin verandering te brengen wordt een grens gesteld aan het aantal opeenvolgende herhaalcycli. Bij het bereiken van deze grens wordt automatisch overgegaan naar de hersynchronisatie-procedure. Onverkwikkelijke bijkomstigheid hierbij is, dat nu wel een soort volgnummer van de blokken bekend moet zijn. Dit is nodig om na hersynchronisatie weer correct naar de datafase te kunnen overschakelen. De noodzaak hiervan moge blijken uit figuur 4: Terminal-1 moet het eerste blok uit het herhaalgeheugen als eerste naar de abonnee doorgeven (X_2), maar terminal-2 moet het derde blok uit het herhaalgeheugen als eerste doorgeven (Y_3), daar de eerste twee al goed ontvangen waren vóór de fout optrad. Wanneer in deze situatie moet worden onderbroken voor hersynchronisatie dan zal deze beslissing niet meer glad verlopen. Om dit te ondervangen worden twee tellers gebruikt: Een teller bij de zender, die opgehoogd wordt telkens wanneer een nieuw blok wordt verzonden; en een teller in de ontvanger, die opgehoogd wordt telkens als een blok wordt doorgegeven naar de abonnee. Door nu bij hersynchronisatie de tellerstand van de zender in de eerste synchronisatieblokken mee te zenden, kan door vergelijken in de ontvanger beslist worden welk blok als eerstvolgende weer moet worden doorgegeven.

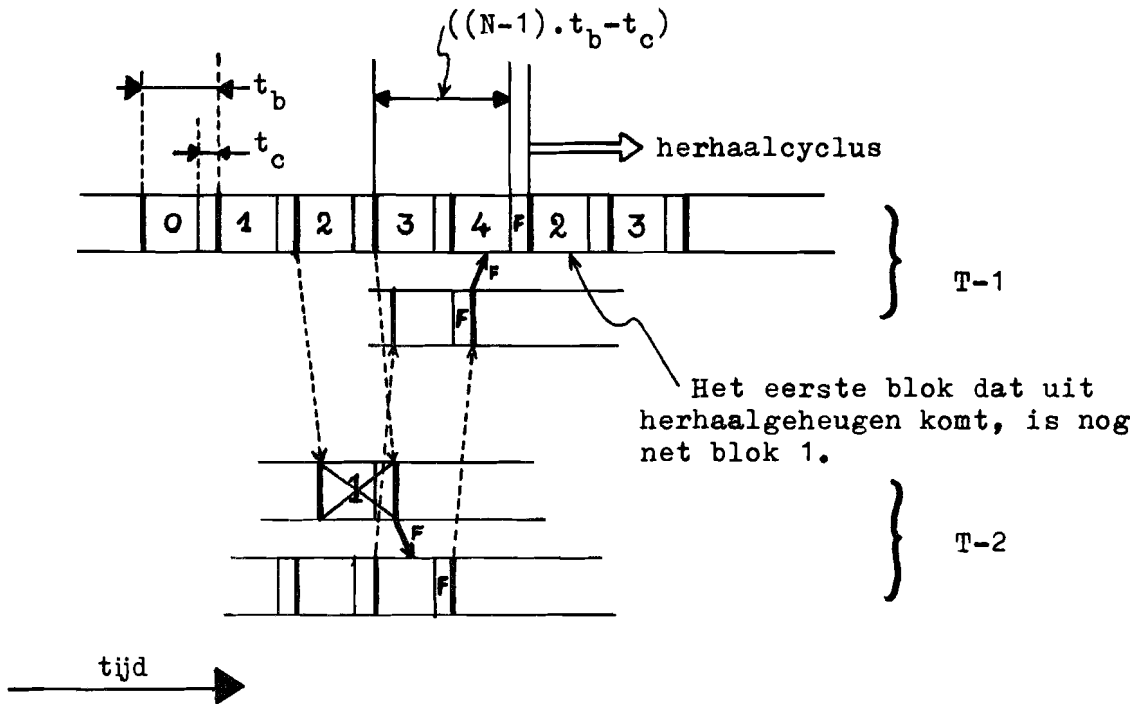


Fig.5: Looptijd en blokfase.

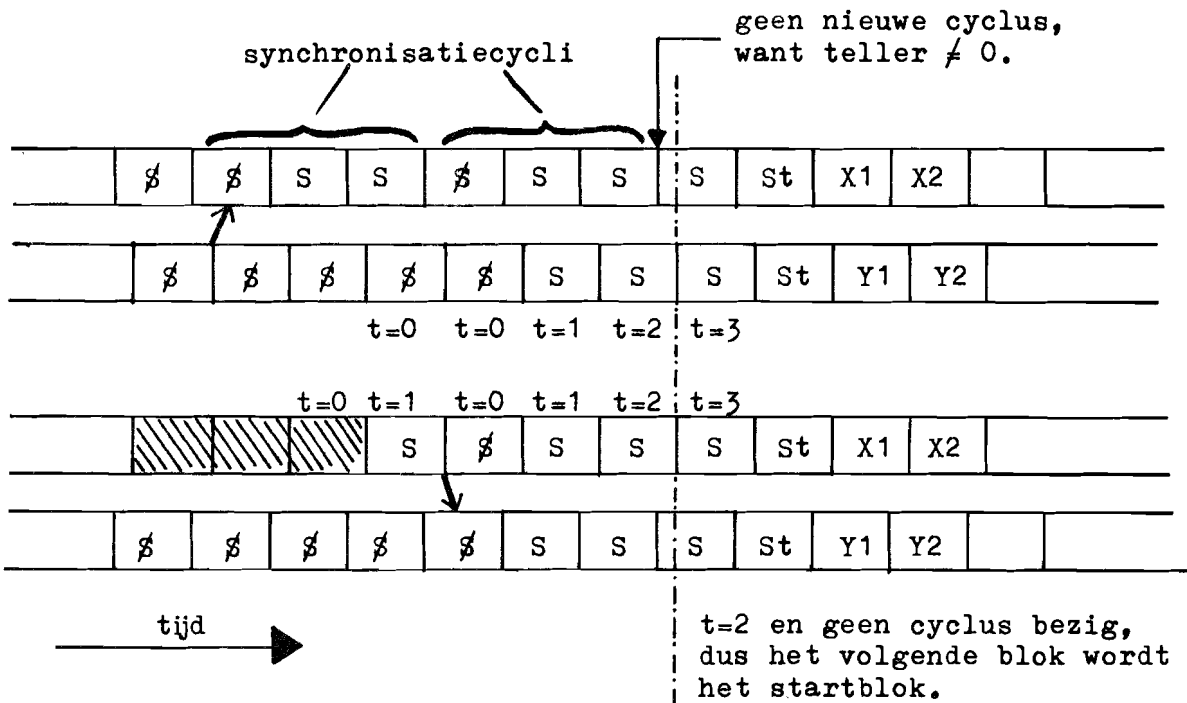


Fig.6: Bloksynchronisatie en start datafase.

5.6. Bitsynchronisatie:

In het voorgaande werd steeds verondersteld dat bitsynchronisatie aanwezig was en dat bovendien de beide terminals met precies dezelfde snelheid zenden (synchrone werking). Dit houdt in dat de locale timing refereert aan eenzelfde moederklok: De klok van de initiatiefnemer. In de master kan dit zonder meer gebeuren. Om in de slave aan deze klok te kunnen refereren moet op de een of andere manier klokinformatie uit het binnenkomend signaal verkregen kunnen worden. Aan de hand van deze informatie kan een kloksignaal geregenereerd worden, of is het mogelijk een locale klok te sturen met behulp van regelnetwerken (fig.7).

Hoe kan nu deze klokinformatie overgezonden worden?

Een tweetal systeemeisen hadden tot gevolg dat klok- en data-informatie in één signaal moeten worden ondergebracht. Dit moet bovendien zodanig gebeuren dat de gelijkspanningscomponent niet meer in het verkregen signaal aanwezig is. Wanneer aan de hand van deze criteria geselecteerd wordt, dan bieden een tweetal methodes zich aan waarmee aan deze eisen kan worden voldaan, nl.:

- Pseudo-ternaire codering,
- Digitale modulatie.

5.6.1. Pseudo-ternaire codering:

Het is mogelijk om bepaalde frequenties in het spectrum van een binair datasignaal te onderdrukken met behulp van bekende technieken (8).

Figuur 8 toont een principeschakeling: Een vertraagde versie $Y(t-\tau)$ wordt van het oorspronkelijke signaal $Y(t)$ afgetrokken. Het verschilsignaal $Z(t)$ kent drie niveaus.

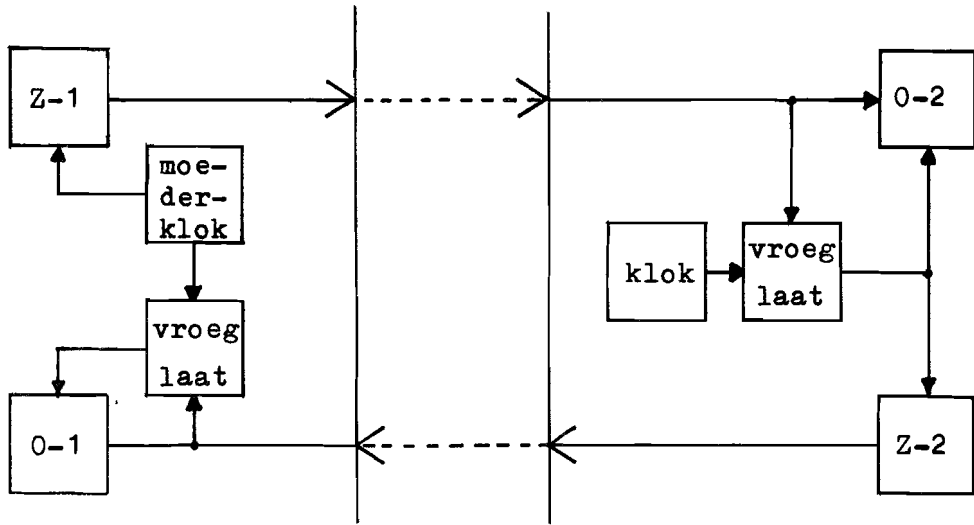


Fig.7: Synchrone werking.

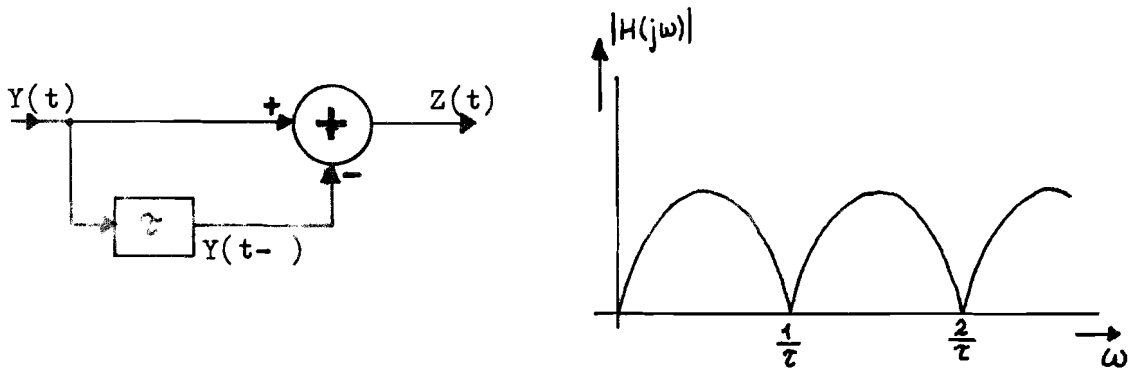


Fig.8: Schakeling voor het onderdrukken van bepaalde frequenties; overdrachtsfunctie.

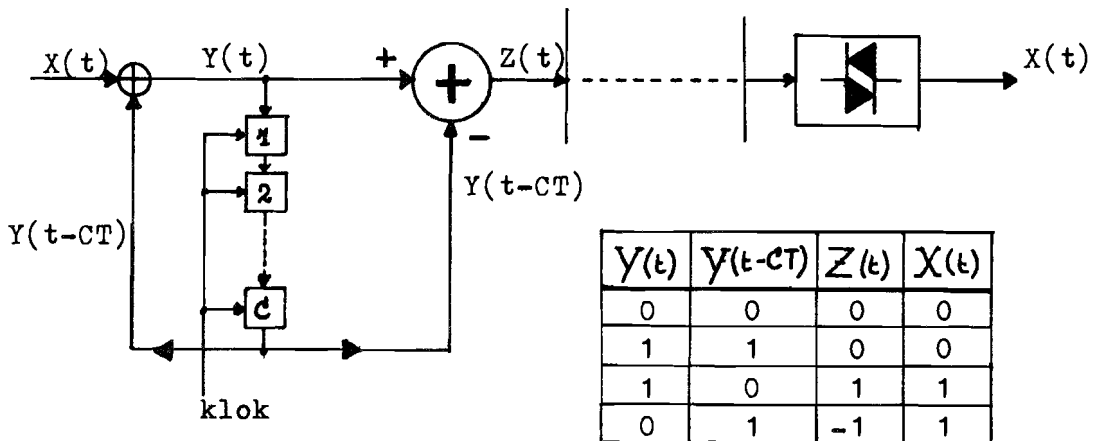


Fig.8a: Principeschakeling voor de pseudo-ternaire encoder; tabel: Input- outputrelatie van de pseudo-ternaire encoder resp. decoder.

De overdrachtsfunctie van dit netwerk is:

$$H(j\omega) = (1 - \exp(-j\omega\tau)).$$

Het amplitudespectrum van $Y(t)$ wordt vermenigvuldigd met:

$$|H(j\omega)| = 2 \left| \sin \frac{\omega\tau}{2} \right|.$$

Dit betekent dat het spectrum van $Z(t)$ spectrale nulpunten heeft bij de frequenties waarvoor geldt:

$$\frac{\omega\tau}{2} = k\pi, \quad \text{waarin } k=0,1,2,3,\dots$$

Dus onder meer ook een nulpunt voor de frequentie nul.

Stel de vertraging τ is gelijk aan een geheel aantal bitvertragingen: $\tau = CT$. Voor $C=2$ worden dan nulpunten gevonden bij elk veelvoud van de halve bitfrequentie. Het vertragen kan nu gebeuren met een schuifregister dat C trappen heeft.

Pseudo-ternaire codering is schematisch weergegeven in figuur 8a. Door de vertraagde versie van $Y(t)$ eerst modulo-2 bij hetingangssignaal $X(t)$ op te tellen wordt bereikt dat het oorspronkelijke signaal $X(t)$ in de ontvanger teruggevonden wordt door 'full-wave rectification' (de negatieve pulsen worden positief).

Het is mogelijk om synchronisatiesignalen samen met het datasignaal over te zenden zonder dat hinderlijke interferentieverschijnselen optreden, zolang de piloottfrequenties overeenkomen met de nulpunten in het dataspectrum.

5.6.2. Digitale modulatie:

Om synchronisatie-informatie in een binair signaal te plaatsen en tevens de gelijkspanningscomponent kwijt te raken, kan ook gebruik worden gemaakt van digitale modulatie (10). Als modulator kan hiervoor een modulo-2-opteller gebruikt worden (fig.9).

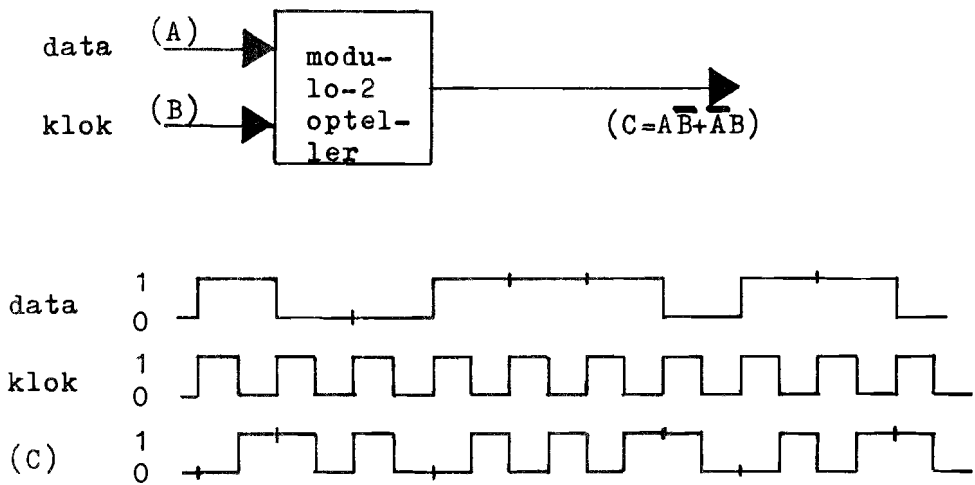


Fig.9: Digitale modulator.

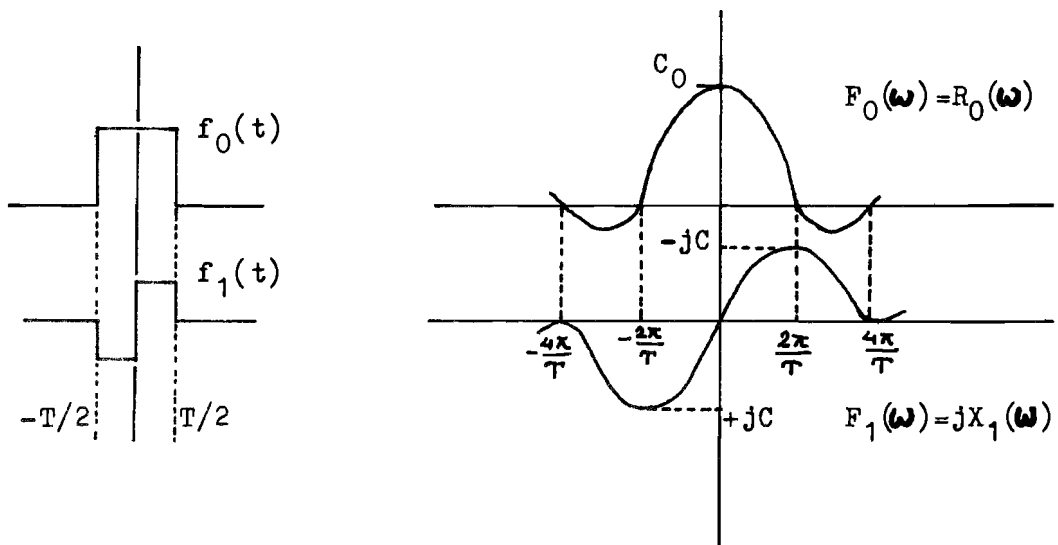


Fig.9a: Spectrumbeschouwing voor en na digitale modulatie.

Op deze manier wordt elk bit omgezet in een dubbelpuls, waarvan de fase afhankelijk is van de polariteit van het gezonden element. Wanneer wij de bijbehorende amplitude-spectra bekijken, zien wij dat de gelijkspanningscomponent nul geworden is (fig.9a).

Aan de ontvangkant kan door middel van flankdetectie klokinformatie teruggewonnen worden.

5.6.3. De voor- en nadelen van pseudo-ternaire codering resp. digitale modulatie:

Om te komen tot een beslissing t.a.v. de toe te passen methode zijn verschillende aspecten van de beide methoden nagegaan:

- Beide methoden reduceren de gelijkspanningscomponent. Bij pseudo-ternaire codering zijn de in het spectrum gecreëerde gaten in werkelijkheid niet echt. Dat geldt alleen als het aangeboden signaal zuiver random is en geen lange series gelijke bits optreden. Bij digitale modulatie is dit niet het geval, daar dan elke puls in een dubbelpuls wordt omgezet.
- De modulatorschakelingen zijn in beide gevallen eenvoudig van opzet en betrouwbaar te realiseren. Bij gebruik van pseudo-ternaire codering is de signaalbewerking tweeledig: Eerst nulpunten creëren en vervolgens synchronisatie-informatie invoegen. Bij digitale modulatie is het signaal voor transmissie gereed, nadat data- en kloksignaal modulo-2 zijn opgeteld.
- Wat betreft het gebruik van de beschikbare bandbreedte is pseudo-ternaire codering optimaal, daar geen extra bandbreedte nodig is voor het synchronisatiesignaal. Digitale modulatie daarentegen heeft voor eenzelfde bit-rate de dubbele bandbreedte nodig.

- Aan de ontvangkant moet voor pseudo-ternaire demodulatie eerst het kloksignaal uitgefilterd worden. Het datasignaal kan vervolgens hersteld worden door een 'full-wave' hersteller. Bij digitale demodulatie wordt gebruik gemaakt van flankdetectie, waarmee zowel klok- als datasignaal kunnen worden teruggevonden.
- Bij pseudo-ternaire codering vormt interferentie tussen data- en synchronisatiesignaal een niet te verwaarlozen probleem, dat zijn eisen stelt aan de nauwkeurigheid waarmee het lijnsignaal gemaakt wordt. Een grote verhouding tussen de amplitudeniveaus van data- en pilotsignaal is gewenst, maar stelt wel weer scherpere eisen aan de detectie-apparatuur. Bij digitale modulatie speelt dit niet zo'n grote rol: Data en klok hebben dezelfde amplitude en kunnen vrij eenvoudig verwerkt worden.
- Jitterproblemen zullen in beide gevallen de nodige aandacht vragen en zijn dan ook nauwelijks als keuzecriterium te gebruiken.
- Pseudo-ternaire codering heeft een extra verzwakking van ongeveer 3dB tengevolge van de omzetting van binaire naar pseudo-ternaire signalen, bij een ontvangen signaalvermogen, dat gemiddeld in de tijd constant is.
- Daar dit onderzoek tevens gericht is op het gebruik van optische koppelingen, is het eenvoudiger om slechts twee niveaus te moeten kunnen onderscheiden. Bij pseudo-ternaire codering zijn dat er drie en moet daarenboven nog een synchronisatiesignaal gedetecteerd worden, dat een amplitudeniveau bezit veel lager dan het datasignaal.

Samenvattend kan gezegd worden dat pseudo-ternaire codering alleen dan de voorkeur zal verdienen wanneer de beschikbare bandbreedte beperkt is en optimaal dient te worden benut. Daar dit in ons geval niet geldt, is gekozen voor digitale modulatie.

6. Realisatie van het systeem:

Alvorens over te gaan tot het bespreken van de ontworpen schema's, zal aan de hand van het blokschema van fig.10 een overzicht gegeven worden van de functies die verricht moeten worden.

Data-terminal apparatuur:

- verbinding aanvragen (→)
- verbinding is gelegd (→)
- data stop-start-schakelaar (←)
- data-ingang (←)
- data-uitgang (→)
- indicatie vrij-bezet
- einde boodschap (→)
- synchronisatie-informatie voor het te zenden signaal (→)
- synchronisatie-informatie voor het ontvangen signaal (←)

Opmerking: Indien nodig moet een parallel-serie-omzetter gebruikt worden om de bits in serie te kunnen aanbieden; dit kan bv. gebeuren door het buffergeheugen de informatie parallel aan te bieden, terwijl de T.O.R.-apparatuur in serie uitleest.

Bloksynchronisatie:

- start de synchronisatie (←)
- synchronisatie-informatie van de data-terminal (←)
- synchronisatieblokken creëren (→)
- schakelen naar het zenden van synchronisatiecycli en het startblok wanneer aan de voorwaarden daartoe is voldaan.
- Na het startblok de mogelijkheid tot herhalen inschakelen en overgaan op datatransmissie
- hersynchronisatie-oproep (←)

Dit blok heeft tot taak het verzorgen van de bloksynchronisatie. Synchronisatie- en startblokken moeten gemaakt worden

met een goed- dan wel foutmelding. De mogelijkheid om correct te kunnen herhalen wordt stapsgewijs ingeschakeld. Wanneer hersynchronisatie gewenst wordt, dient tijdens de synchronisatieprocedure de rangteller te worden ingevoegd.

Blok-encoder:

- de aangeboden informatie coderen
- stop-start mogelijkheid in verband met herhaalcycli
- uitgang waarop de gecodeerde blokken verschijnen en worden doorgegeven naar de modulator.
- klokinformatie (van de moederklok of van de locale klok).

De encoder heeft tot taak de aangeboden informatie te verwerken tot correcte codewoorden. vervolgens worden deze blokken doorgegeven naar de modulator en tevens in het herhaalgeheugen geschoven. Moet een herhaalcyclus verzorgd worden, dan krijgt de encoder geen nieuwe informatie aangeboden en worden de blokken die in het herhaalgeheugen staan herhaald.

Modulator:

- signaalingang
- klokingang
- uitgang naar de zender

Bij toepassing van digitale modulatie heeft deze schakeling een betrekkelijk eenvoudig karakter. Op de uitgang verschijnt het gemoduleerde signaal waaruit in de ontvanger data en klokinformatie kunnen worden teruggevonden.

Demodulator:

- signaalingang
- klokinformatie uitgang
- data-informatie uitgang

De taak van de demodulator is het scheiden van data- en kloksignaal. Zoals wij zagen gebeurt dit bij digitale modulatie door middel van flankdetectie.

Decoder:

- ingang datasignaal
- ingang kloksignaal
- uitgang voor correct ontvangen data.
- herkenning synchronisatiepatroon melden (→)
- herkenning van het startblok (→)
- starten van de blokteller
- foutmelding doorgeven (→)

Met behulp van de decoder wordt de bloksynchronisatie verzorgd: op het moment dat het herkenningspatroon gedetecteerd wordt, wordt de blokteller gestart en is bloksynchronisatie aanwezig. Ook wordt dit gemeld naar de zender die dan synchronisatiecycli gaat zenden; de decoder zelf wordt geschakeld naar de toestand waarin cyclisch gedecodeerd kan worden, zoals in het voorgaande werd beschreven. Wordt een fout gedetecteerd, dan levert de 'alles nul' test een negatief resultaat op en dit heeft tot gevolg dat een herhalingscyclus verzorgd zal worden. Correct ontvangen informatie mag naar het ontvangbuffer worden doorgegeven. Is het buffer vol, dan mag geen nieuwe informatie meer worden doorgegeven. Dit betekent dat dan continu een foutmelding moet worden doorgegeven naar de zender, opdat geen informatie verloren zal gaan.

Herhalingsblok:

- inschakelen eerste fase van het herhaalprocédé.
- inschakelen tweede fase van het herhaalprocédé.
- starten van een cyclus in geval een foutmelding wordt aangeboden door de decoder: Data toevoer stop, herhaalgeheugen start etc.

De taak van dit blok is: Het verzorgen van de herhalingscyclus, d.w.z. de verschillende bitstromen regelen zodat geen blokken verloren gaan, dan wel dubbel ontvangen worden. Daartoe dienen aan het begin van zo'n cyclus verschillende stromen te worden stopgezet. Voorts moet administratie worden gepleegd (tellers), opdat aan het einde van de cyclus de gestopte stromen weer tijdig gestart worden, tenzij nog een keer herhaald moet worden of hersynchronisatie gewenst is. De eerste fase van het herhaalprocédé is het verzorgen van de synchronisatiecycli lettend op de twee voorwaarden voor het startblok.

Is aan deze twee voorwaarden voldaan, dan wordt de tweede fase ingeschakeld en tegelijkertijd het startblok gezonden. In deze tweede fase is het systeem dus in staat normale herhaalcycli te verzorgen.

Opmerking: De pijltjes achter de verschillende signaal-aanduidingen geven aan of het betreffende signaal het blok binnenkomt, dan wel door het blok gemaakt dient te worden.

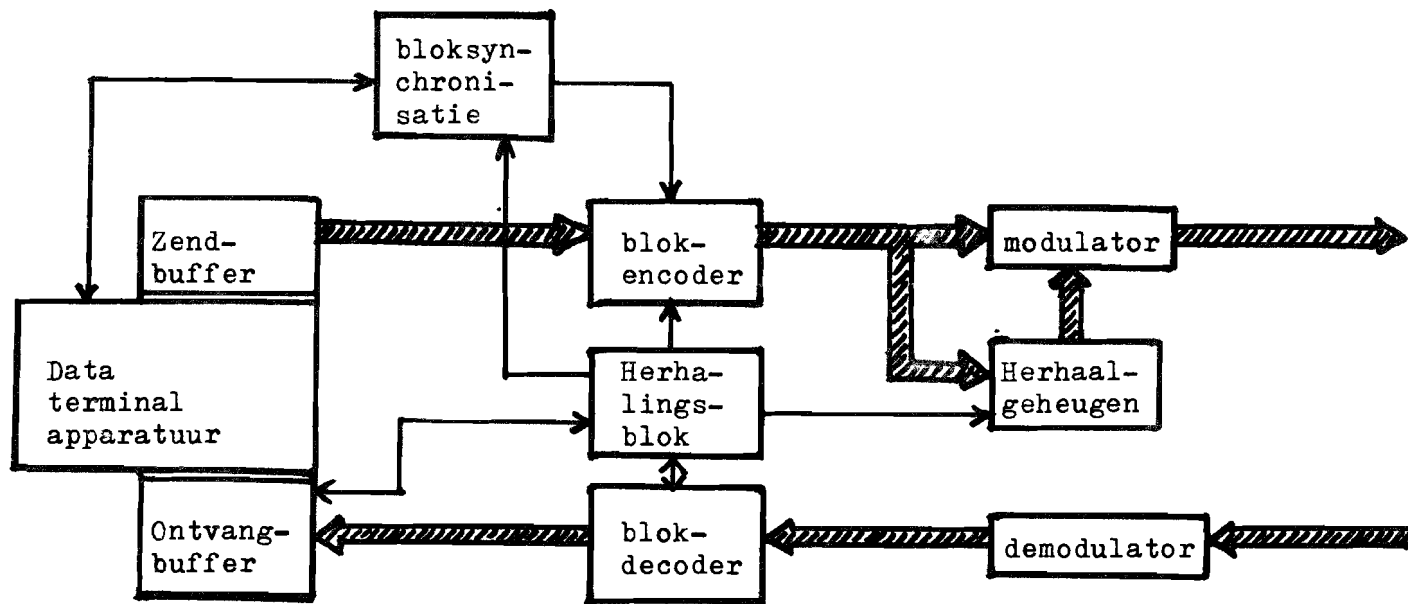


Fig.10: Blokschema van het transmissiesysteem.
 De gearceerde pijlen geven het datapad aan; de andere
 pijlen duiden aan de interactie tussen de verschillende blokken.

Tot slot zal nu een en ander gezegd worden over het realisatie-ontwerp zoals dat op dit moment bestaat. Voor een duidelijk inzicht in de verschillende procedures zijn deze in de flowdiagrammen 1, 2 en 3 overzichtelijk weergegeven voor resp. bloksynchronisatie en start datafase, de normale herhalingscyclus en voor de hersynchronisatieprocedure.

Schema 1:

Hierin zijn weergegeven de encoder en de blokteller. De blokteller kan d.m.v. instelschakelaars voor verschillende bloklengtes ingesteld worden. Aan deze teller worden schakelsignalen voor de encoder ontleend. Eveneens wordt m.b.v. het minst significante tellerbit (LSB) de synchronisatie informatie gecreëerd; de synchronisatieblokken worden a.v. gemaakt: de eerste $n-20$ bits zijn nul; de volgende 4 bits (0101) worden door het register geklokt en genereren daar het herkenningpatroon: 010100001010010-1. Het laatste bit maakt geen deel uit van het herkenningpatroon, daar dit bit gebruikt wordt voor de foutmelding en het herkenningpatroon zowel bij een goedmelding als bij een foutmelding aanwezig dient te zijn.

Opmerking: Wanneer de bloklengte veranderd wordt, dient wel rekening te worden gehouden met het feit dat het herkenningpatroon gegenereerd wordt door (0101), de tellerstand moet dus even zijn bij het eerste van deze 4 bits.

Schema 2:

Dit schema bevat de blokteller voor het ontvangregister en tevens twee schakelingen: 'Synchronisatiecyclusteller' en de 'Goedteller' welke na herkenning van het synchronisatiepatroon de goed-foutmelding verzorgen in de te zenden synchronisatieblokken ($F_{outpuls}^Z(1+2)$). Bovendien kan aan de hand van deze schakelingen worden nagegaan of aan de twee voorwaarden voor het zenden van het startblok is voldaan. Is dit het geval dan wordt de foutmelding verder verzorgd door ($F_{outpuls}^Z(3)$) (schema 3).

De blokteller van het ontvangregister zal in de stand n (einde blok) blijven wachten totdat het herkenningsspatroon gedetecteerd wordt. Gebeurt dit, dan wordt de teller ge-'load' door de herkenningsspuls (1). Daarna wordt steeds ge-'load' m.b.v. de door de herkenningsspuls geschakelde flipflop (2). Het signaal 'Load-blokteller' wordt ook gebruikt om het ontvangend register aan het eind van een blok te clearen.

Schema 3:

De decoder wordt in eerste instantie zo geschakeld, dat de bits zonder meer worden doorgeschoven. Op deze manier kan het herkenningsspatroon gedetecteerd worden. Na deze detectie wordt het register geschakeld voor cyclische detectie. Door middel van een alles-nul test worden fouten gedetecteerd en zonodig een foutpuls gemaakt, die naar de zender wordt doorgegeven en ervoor zorgt dat daar het laatste bit van het eerstvolgende blok wordt geïnverteerd. Bovendien wordt met behulp van de 'Herhalingscyclus-teller' het register afgesloten voor de eerstvolgende twee blokken. Ook is in dit schema nog weergegeven hoe het doorgeven van de correcte informatie kan geschieden: De in dit geval 90 informatiebits worden in een register geschoven en mogen in geval het blok goed is ontvangen naar het ontvangbuffer worden doorgegeven. Dit doorgeven kan parallel dan wel in serie geschieden. Serie verwerking lijkt eenvoudiger daar dan niet nog eens een 'byte-klok' hoeft te worden gemaakt, maar dit kan van geval tot geval verschillen.

Opgemerkt moet nog worden dat het signaal herkenning-startblok specifiek is voor het hier beschouwde geval van blokken van 106 bits. Het startblok bestaat nl. uit 90 bits(0101....01) die een checkbitpatroon genereren (0101011011000100). Voor andere bloklengtes zal dit patroon anders zijn, waarmee dus rekening dient te worden gehouden. Het is misschien beter om gebruik te maken van het signaal 'Starblok' om aan te geven dat vanaf een bepaald ogenblik goede blokken naar het ontvangbuffer moeten worden doorgegeven.

Schema 4:

Hierin wordt weergegeven de organisatie rond het herhaalgeheugen: Het herhaalgeheugen moet gestart worden na een blok waarin een foutmelding werd geplaatst. Drie blokken later moet beoordeeld worden of teruggeschakeld moet worden naar de normale zendtoestand, dan wel nogmaals herhaald dient te worden. Alleen in het geval dat hersynchronisatie gewenst is moet het herhaalgeheugen daadwerkelijk stopgezet worden. Tydens de dan volgende synchronisatie moeten nl. de in het geheugen staande blokken behouden blijven. Dit betekent ook, dat het schakelen steeds zodanig moet geschieden, dat de blokken correct in het geheugen staan. Hersynchronisatie is gewenst wanneer het aantal opeenvolgende herhaalcycli de 15 dreigt te overschrijden; zij wordt dan verzorgd vanuit de 'Teller aantal herhaalcycli'.

In schema 4 is ook opgenomen de inputregeling voor het zendregister. In eerste instantie wordt bloksynchronisatie-informatie aan het register doorgegeven. Na het startblok wordt de datatoevoer ge-enabled. Tijdens een herhaalcyclus wordt de datatoevoer stopgezet. Er wordt dan wel iets door het register geschoven, maar de zendregisteruitgang is niet met de modulator en de ingang van het herhaalgeheugen doorverbonden.

Het signaal 'Transmissie-voorwaarde' wordt aan het begin van een verbinding gegeven om ervoor te zorgen dat alle deel-schakelingen in de goede uitgangspositie verkeren.

Schema 5:

Hierin wordt weergegeven hoe de rangtellers werken en hoe tijdens hersynchronisatie de tellerstand van de zender kan worden meegezonden en in de ontvanger kan worden opgevangen. Gezien de uitbreiding die deze hersynchronisatie-procedure met zich meebrengt, lijkt het nuttig om na te gaan of het niet toch voordeliger is om aan de blokken een tweetal rangtellerbits toe te voegen. Dit betekent dan wel een verlaging van de efficiency, maar zal wel de hersynchronisatie aanmerkelijk vereenvoudigen. Belangrijk bij deze beslissing zal zijn de bloklengte waarmee gewerkt wordt. Bij een bloklengte van 106 bits zal de efficiency-verlaging niet zo groot zijn.

Deze zal echter toenemen naarmate de bloklengte kleiner wordt. In de schakeling zoals die nu gerealiseerd is, wordt de ontvangen tellerstand vergeleken met de tellerstand in de ontvanger. Overgezonden wordt het rangnummer van het eerste blok dat uit het herhaalgeheugen zal komen wanneer dit gestart wordt. De rangteller in de ontvanger geeft aan het rangnummer van het laatste goed ontvangen blok:

$$r_z + \bar{r}_0 = r_z - 1 - r_0.$$

Wanneer $(r_z - 1) = r_0$, dan moet het volgende blok (met rang $r_0 + 1$) weer doorgegeven worden.

Hoe na hersynchronisatie teruggekeerd wordt naar de normale zendtoestand is in flow-diagram 3 weergegeven.

7. Nabeschouwing:

Aan het einde van dit afstudeeronderzoek is een schema gemaakt voor een realisatie waarin in eerste instantie gewerkt wordt met blokken van 106 bits: 90 informatiebits en 16 checkbits. Er zijn geen extra bits aangebracht voor het toekennen van een volgnummer aan de blokken. Opgemerkt werd reeds, dat wanneer de bloklengte niet al te klein is het aan te bevelen is om i.p.v. het werken met rangtellers in zender en ontvanger, toch een tweetal rangbits aan een blok toe te voegen. Ook valt op dat hoewel het mogelijk is de bloklengte te variëren, dit toch een aantal kleinere problemen met zich meebrengt.

De grote moeilijkheid bij het ontwerpen van het systeem lag in het feit dat een aantal eigenschappen van het systeem vrij vaag zijn. Dit leidde tot een streven naar een zo groot mogelijke plooibaarheid van het systeem. Een plooibaarheid, die niet steeds gehaald is. Om een beter inzicht te krijgen in de systeemgedragingen, is het waarschijnlijk nuttig een proefmodel te bouwen, opdat het systeemgedrag getest kan worden. Aan de hand van daaruit voortvloeiende resultaten moet het mogelijk zijn het systeem beter af te stemmen op het gestelde doel.

Het is duidelijk dat het door mij verrichte werk niet het einde kan betekenen van dit onderzoek; er moeten nog een aantal aspecten onderzocht worden en het is reeds aangeduid dat resultaten van dit onderzoek alsnog kunnen leiden tot veranderingen in dit systeem,

J.A. Meys.

8. Literatuur:

- (1) H.C.A. van Duuren, 'Typendruktelegrafie over radioverbindingen (T.O.R.)', Tijdschrift van het Nederlands radiogenootschap XVI, pp 53-67, 1951.
- (2) W.B. Bishop, B.L. Buchanan, 'Message redundancy versus feedback for reducing message uncertainty', National convention record, Institute of radio engineers, pt 2, Vol 5, pp 33-39, 1957.
- (3) F.B. Wood, 'Optimum blocklength for datatransmission with error checking', Associate member AIEE International business machines corp. San José, California.
- (4) Thomas G. Kuhn, 'Retransmission error control', IEEE Trans. on comm. systems, june 1963, pp 186-201.
- (5) H.O. Burton, D.D. Sullivan, 'Errors and error control', Proceedings of the IEEE, nov.1972, pp 1293-1301.
- (6) W.W. Peterson, E.J. Weldon, 'Error correcting codes', sec. edition.
- (7) R.O. Carter, A.D. Marr, 'System organisation for error correction by retransmission in duplex networks', Internat. Zürich seminar on digital comm. 1974, pp F4(1)-F4(5).
- (8) P.J. van Gerwen, 'On the generation and application of pseudo-ternary codes in pulsetransmission', Phil. Res. Rep., 1965, Vol 20, pp 469-484.
- (9) S.W. Golomb, 'Shift register sequences', Holden Day Inc. San Francisco, 1967.
- (10) A.P. Verlijdsdonk, 'Digitale modulatie- en filtertechnieken bij synchrone datatransmissie', De Ingenieur, nr 9, febr. 1970.
- (11) W.W. Chu, 'Optimal fixed message block size for computer communications', Proc. of the IFIP, Ljubljana, Yugoslavia, august 1971.

ONTVANGKANT

ZENDKANT

Foutmelding in de ontvanger.

Het ontvangend register kijkt niet naar de eerstvolgende twee binnenkomende blokken.

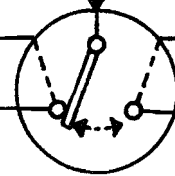
Foutmelding inpassen in het eerstvolgende te zenden blok.

Aan het einde van dit blok:
1) toevoer stop,
2) herhaalgeheugen start.
De eerste twee blokken uit het herhaalgeheugen krijgen een goedmelding.

Het derde blok wordt weer beoordeeld: Is het goed of fout?

GOED

FOUT



Het derde blok wordt met een goedmelding verzonden; aan het einde van dit blok wordt de toevoer weer gestart.

Het derde blok uit het herhaalgeheugen wordt verzonden met een foutmelding.

Zou de nu volgende herhaalcyclus de 16de op rij worden?

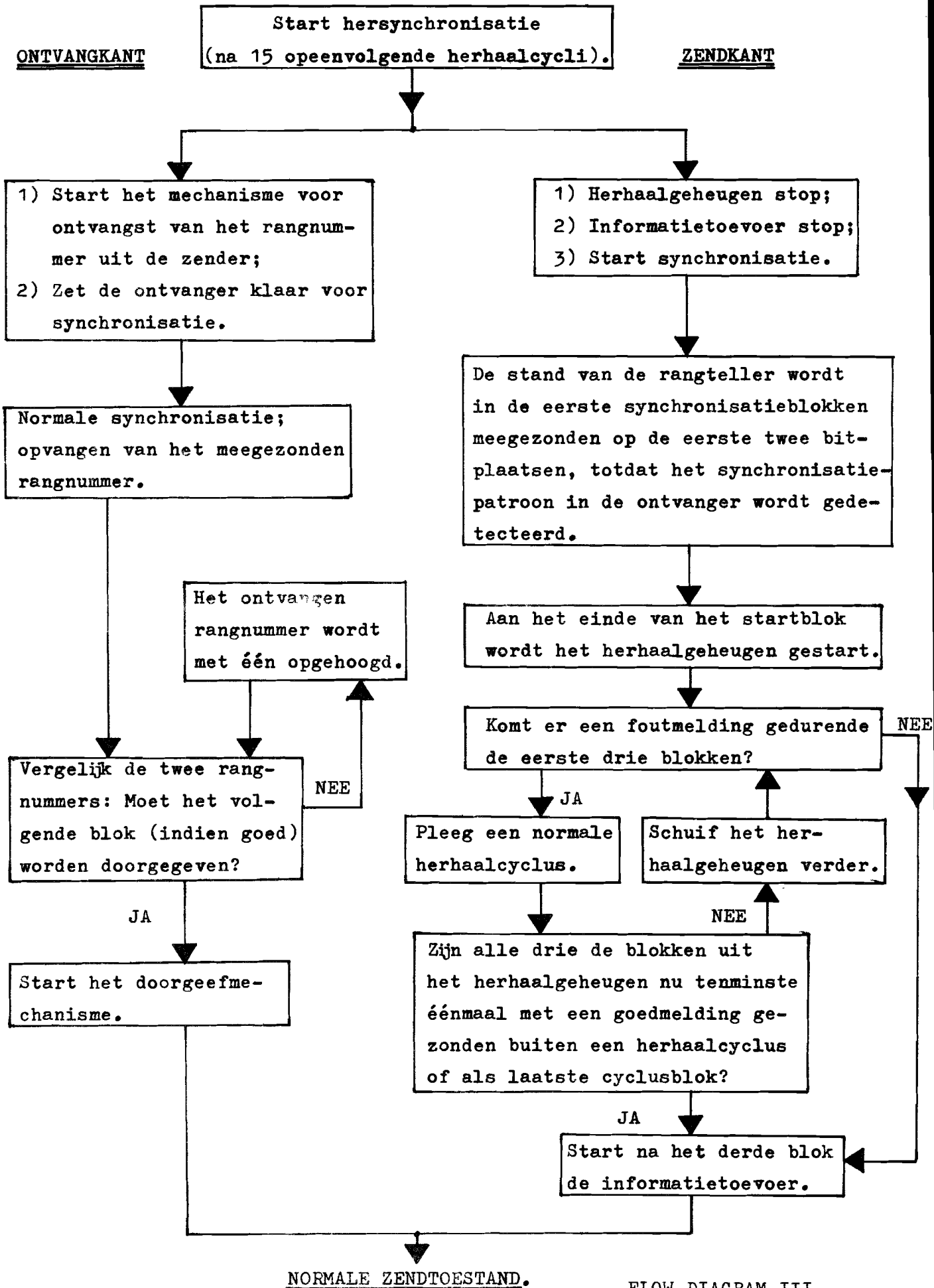
JA

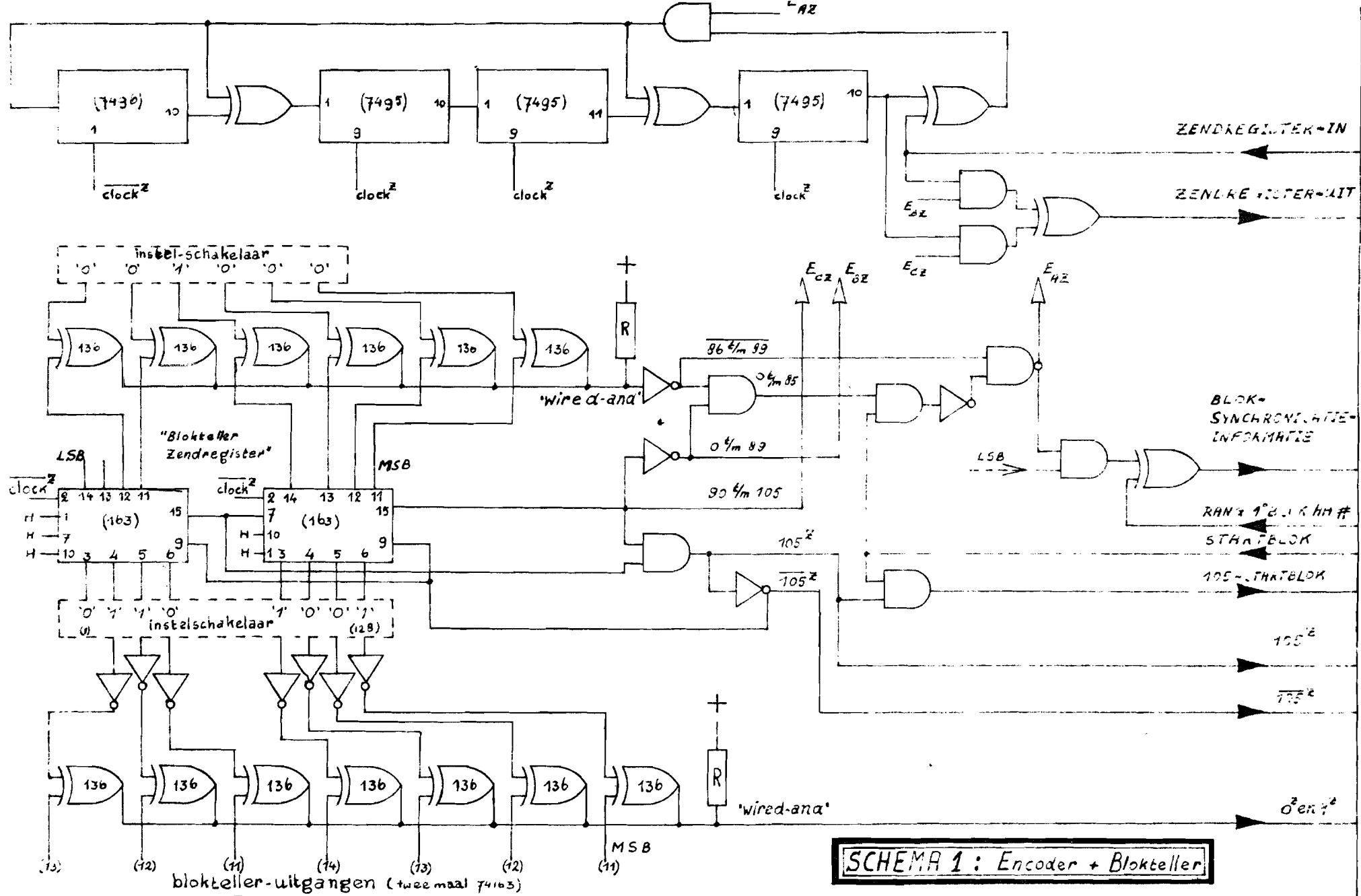
NEE

NORMALE ZENDTOESTAND.

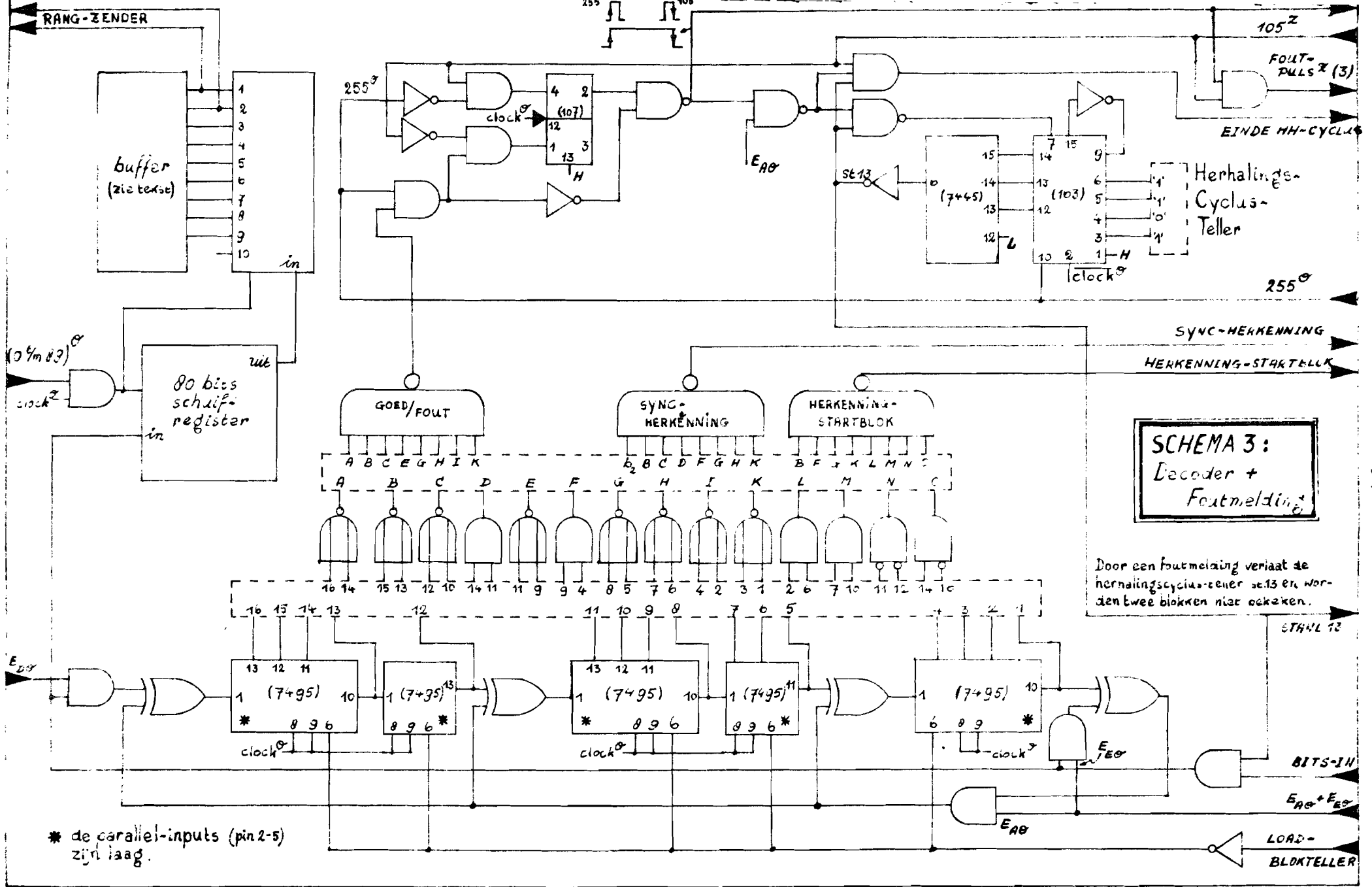
HERSYNCHRONISATIE-PROCEDURE.

FLOW-DIAGRAM II





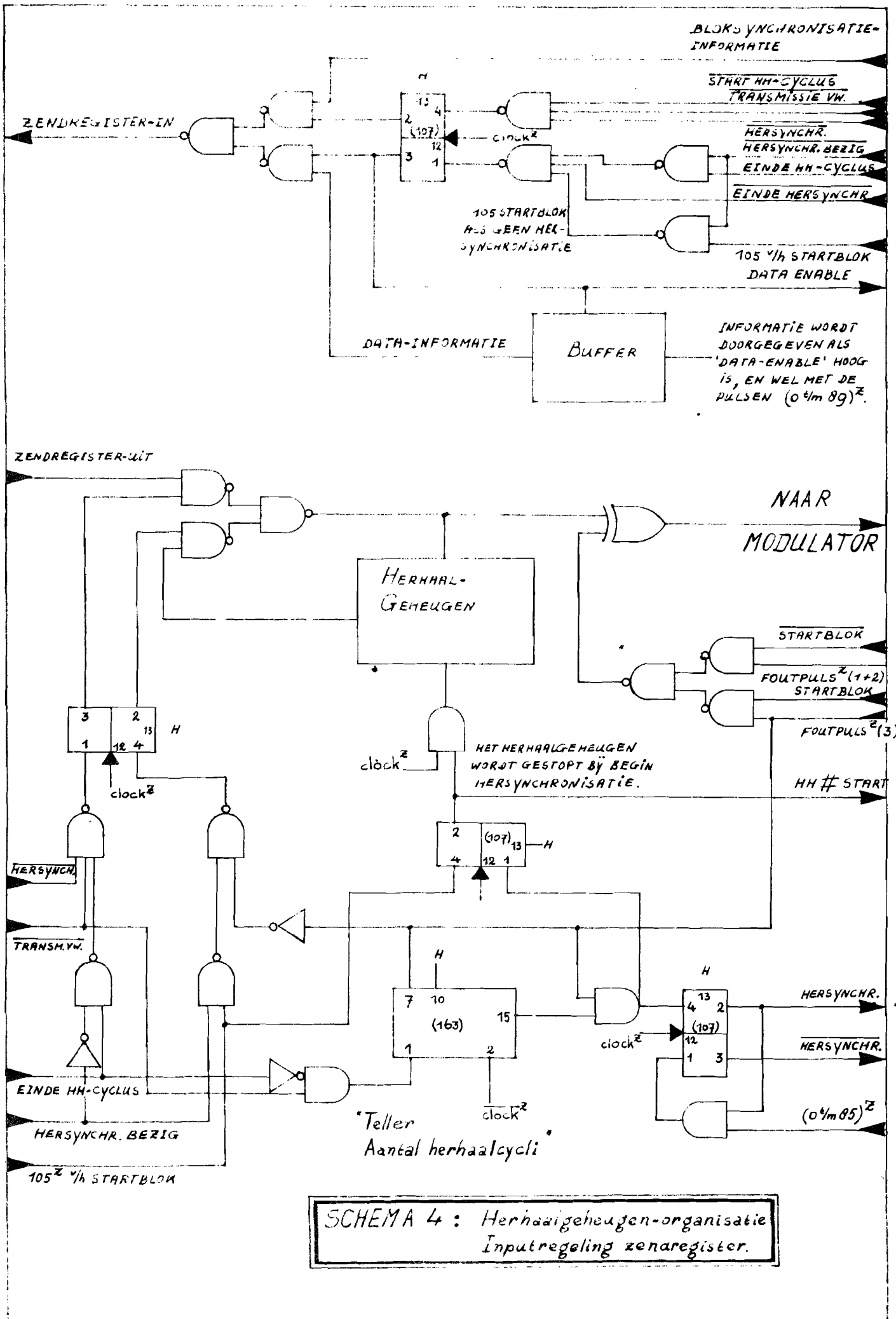
SCHEMA 1: Encoder + Blokteller

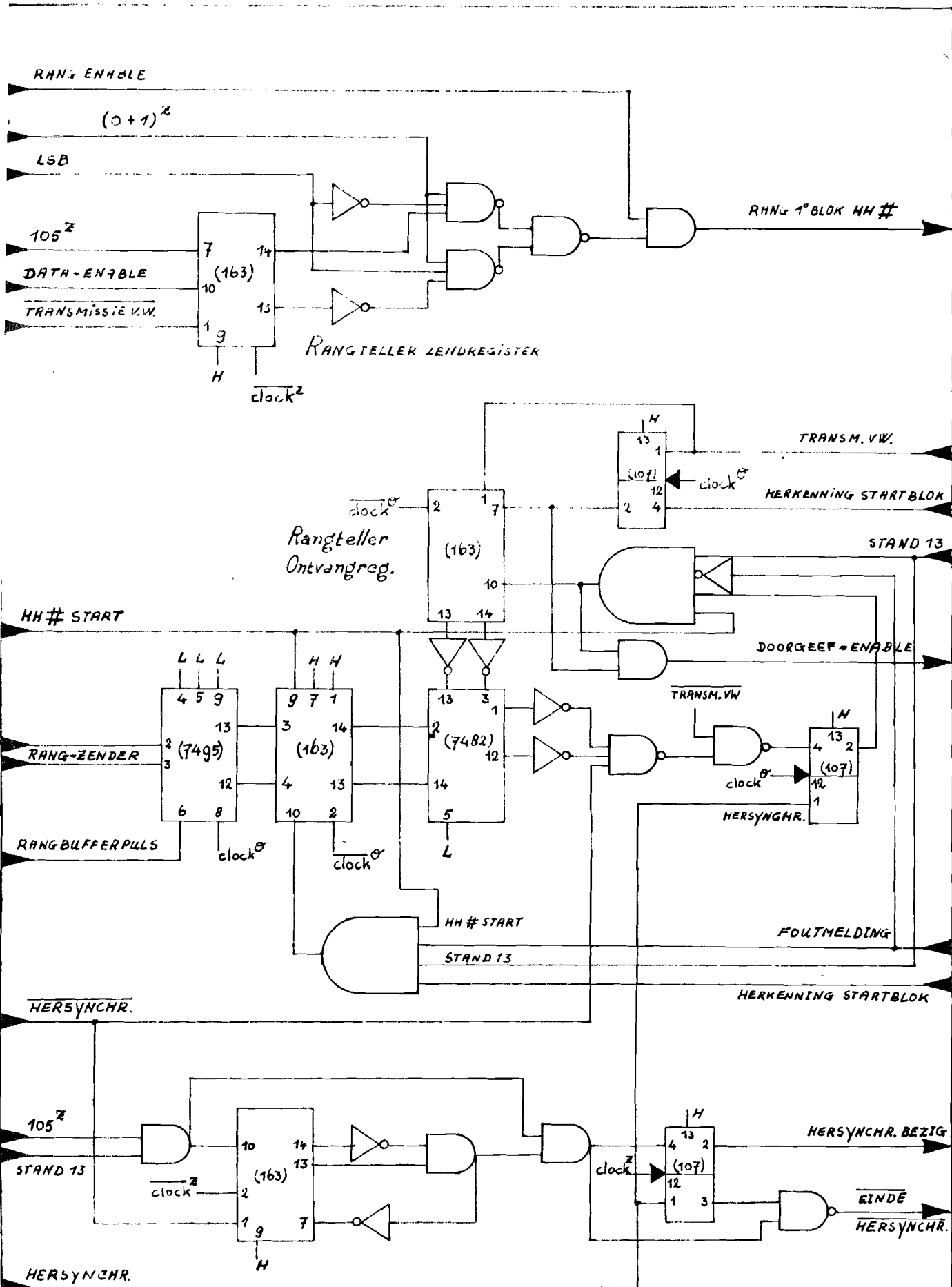


SCHEMA 3:
Decoder +
Faultmelding

Door een foutmelding veriaat de herhalingscyclus-eener se.13 en worden twee blokken naar oekzen.

* de parallel-inputs (pin 2-5) zijn laag.





SCHEMA 5: Rangtellers