

MATHEMATISCH CENTRUM

2e BOERHAAVESTRAAT 49

AMSTERDAM

REKENAFDELING

HANDBOEK VOOR DE PROGRAMMEUR

FERTA

DEEL II

door

E.W.Dijkstra.

MR 20

1 9 5 5

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM



I N H O U D:

	Blz.
INLEIDING	1
HET INVOERPROGRAMMA VAN DE FERTA	2
HET HANDREGISTER	2
DE BAND (ALGEMEEN)	4
(SYSTEMATISCH)	9
COPIEER BAND	12
VOORPONSING VIA G	12
INVOERPROGRAMMA ALS SUBROUTINE	12
HET IN - EN UITVOERPROGRAMMA IN HET GEHEUGEN	13
HERSTEL VRIJE KANALEN	14
HET TYPEN	15
TYP-CONTROLE; n GETALLEN PER REGEL	18
DE OPDRACHTEN VAN HET IN-EN UITVOERPROGRAMMA	21
HERSTEL VRIJE KANALEN (BAND).	35
BLOKSCHEMA'S	37
BESCHRIJVING: SERVICEPROGRAMMA:PONS BIBAND	43
UTILITY BREUK LEZEN 8 CIJFERS	48
TIJDSDUUR VAN HET INVOERPROGRAMMA	49

Inleiding.

Dit tweede deel is gewijd aan het communicatieprogramma van FERTA; dit omvat het invoerprogramma en de standaard typroutines (waaronder de typcontrole-regelindeling). Evenals het eerste deel is dit rapport uitgebreider dan strict noodzakelijk. De uitvoerige beschrijving is gegeven, opdat allen, die in de in- en uitvoer meer geïnteresseerd zijn - met name de operateurs - zonder onevenredig veel moeite er wegwijs in kunnen worden.

Voor de programmeur zijn de ponsconventies het belangrijkste, omdat zij vastleggen in welke notatie hij zijn programma op dient te schrijven. Wij raden hem aan, zich een keer de moeite te getroosten zich in de typroutines en de typcontrole te verdiepen, speciaal in de wijze, waarop zij in een programma geïncorporeerd worden.

Tot slot zijn twee niet-standaard communicatieprogrammaatjes als voorbeeld gegeven, te weten een service-programma, dat een stuk geheugen volgens de biband-conventies uitponst, en een utility leesprogramma voor het lezen van breuken met exact 8 cijfers achter de komma.

Het invoerprogramma van de FERTA.

Wanneer informatie van de buitenwereld de machine ingevoerd wordt, geschiedt dit met behulp van een standaardprogramma, dat permanent in de machine aanwezig is.

Deze informatie wordt of - met de hand - via toetsenpaneel en getalschakelaars, of - met de geponste telexband - via de bandlezer toegevoerd.

Het handregister.

Het handregister bestaat uit 14 toetsen, te weten 10 cijfertoetsen voor de cijfers van 0 t/m 9 en 4 tekentoetsen voorzien van de etiketten +, -, +. en -. . Als men op een van deze toetsen drukt (bij stilstaande machine) start de machine automatisch op adres 0 0 a, in een gedeelte van het invoerprogramma. Hier wordt onmiddellijk een symbool ingelezen (door de opdracht 27 0 1), afhankelijk van welke toets van het handregister ingedrukt is. Het programma onderscheidt tussen tekentoets en cijfertoets. Als een tekentoets is ingedrukt, wordt de decimale opbouw voorbereid, tevens wordt in het geheugen vastgelegd, welke tekentoets is ingedrukt. Als een cijfertoets wordt ingedrukt, komt de decimale opbouw in werking. Na afloop, d.w.z. als de laatste cijfertoets is ingedrukt, moet een maal op de knop "BEGIN VOLGENDE OPDRACHT" gedrukt worden: de machine brengt naar behoren in rekening, welke tekentoets is ingedrukt, het oorspronkelijke decimaal gegeven getal in tweetallige representatie in S achterlatend.

Ten aanzien van het gebruik van het handregister gelden de volgende conventies:

- 1^o Het inbrengen van een getal wordt altijd ingeleid door het indrukken van een tekentoets. Hiermede begint het programma, "met een schone lei", d.w.z. als een fout gemaakt is, begint men opnieuw met het indrukken van de tekentoets.
- 2^o Vervolgens worden de successievelijke decimale cijfers, normaal "van links naar rechts" aangeslagen.
Bij gehele getallen (+ en -) mag men nullen "aan het begin", dus aan de hoge kant, weglaten: men mag facultatief aanslaan (hoeft niet).
Bij breuken (+. en -.) moet men het getal, zo nodig, tot 8 cijfers achter de komma aanvullen. Eventuele rijtjes nullen, onmiddellijk achter de komma, hoeven niet aangeslagen te worden. (Men slaat het 10⁸-voud facultatief aan).
Er moet altijd minstens een cijfertoets worden aangeslagen!
- 3^o Men drukt eenmaal op de toets "BEGIN VOLGENDE OPDRACHT". De tekentoets wordt hiermede effectief. In geval van +. en -. .

worden hierbij de constanten $5 \cdot 10^7 = \begin{bmatrix} 31 & 13 \end{bmatrix}$ en $10^8 = \begin{bmatrix} 31 & 20 \end{bmatrix}$ uit het vrije kanaal gebruikt.

Seriehandinzet. (Start op 2 16)

Een tweede functie van het invoerprogramma komt bij de seriehandinzet tot zijn recht. Hier wordt niet een woord informatie ingebracht, maar verscheidene woorden, die dus niet in een register, maar in adressen worden afgeleverd. Bij de seriehandinzet wordt een reeks adressen ingevuld, bepaald door het eerste adres en de z.g. spatiering, d.w.z. het verschil tussen twee adressen, waar opeenvolgend ingebrachte moleculen worden opgeborgen. Zo kan men b.v. van een matrix zowel rij als kolom met de seriehandinzet inbrengen. De minimum spatiering is 1: dan bergt men op opeenvolgende plaatsen. De twee parameters, begin (v r) en spatiering ($\Delta v \Delta r$) worden in de getalschakelaars meegegeven: $\langle G \rangle = 0 \Delta v \Delta r 0 v r$, dus eerste adres in de a-helft, spatiering in de b-helft.

Dan start men de machine op 2 16, na de nodige voorbereidingen (G wordt hierbij gelezen) stopt de machine. Van "BEGIN GEKOZEN OPDRACHT" wordt dan geen gebruik meer gemaakt.

Men drukt dan per molecuul informatie - in te vullen adres - steeds twee maal op de knop "BEGIN VOLGENDE OPDRACHT". Na de tweede maal indrukken wordt (S) ter plaatse in het geheugen opgeborgen, en het adres van deze wegbergopdracht met $\Delta v \Delta r$ opgehoogd.

Wil men een molecuul informatie met het handregister inbrengen (getal), dan wordt door het de eerste maal indrukken van "BEGIN VOLGENDE OPDRACHT" het vormen van het juiste getal in (S) afgemaakt (zie Handregister).

Is echter niet een van de toetsen van het handregister ingedrukt, dan wordt door de eerste keer "BEGIN VOLGENDE OPDRACHT" de inhoud van de getalschakelaars in S gelezen. Zo kan men een opdrachtenkoppel inbrengen. Met het oog op aankrengen van wijzigingen in programma's - wat vaak een rijtje opdrachten betreft - en het met de hand inbrengen van heel kleine programmatjes - testen van een bepaalde opdracht b.v. - is deze faciliteit ingelast.

Na het ophogen van de wegbergopdracht - dus na de tweede maal doorstarten - stopt de machine op adres 2 25 a. Als, terwijl een molecuul uit G ingevoerd wordt, abusievelijk een toets van het handregister is ingedrukt, moet "de eerste keer" doorstarten dus vervangen worden door starten op 2 25 b.

De band.

De band is echter het middel, waarmee grote hoeveelheden informatie worden ingevoerd. Invoer met behulp van de seriehandinavet heeft n.l. twee grote nadelen: ten eerste kost het veel machinetijd, ten tweede is de kans op vergissingen groot. De band wordt snel gelezen, en was een band gisteren goed, zo zal hij morgen zijn. Programma's en numerieke gegevens worden daarom als regel via de geponste telexband ingevoerd, standaardprogramma's worden op dit medium bewaard.

In tegenstelling tot de seriehandinavet, die met een willekeurige spatiering "door het geheugen stapt", worden bij het bandlezen successievelijke moleculen in eerste instantie op opeenvolgende adressen in het geheugen geborgen: een dergelijk rijtje wordt op de band voorafgegaan door een groepje pentades, die aangeven, ten eerste, dat er een nieuw rijtje komt, en ten tweede, waar dit volgende rijtje begint, d.w.z. het adres van het eerste molecuul. Een dergelijk groepje pentades noemt men een controlecombinatie: de er in besloten informatie wordt niet als zodanig in het geheugen geborgen, maar doet zich gelden door de wijze, waarop de andere informatie, de moleculen, geborgen worden. Het invoerprogramma kent verscheidene controlecombinaties.

De toetsen van de ponsmachines zijn genummerd van 0 t/m 31; deze getallen zijn gelijk aan de bandaequivalenten, dus aan het getal, dat bij lezing van betrofven pentade in L verschijnt. 6 toetsen zijn echter dubbel benoemd, te weten

0	+
1	-
2	..
3	..
30	F
31	L

De "echte" moleculen, die ingebracht worden, zijn drieërlei:

- 1° 6 pentades voor binaire representatie (BIBAND-molecuul)
- 2° 8 pentades voor opdrachtenkoppel
- 3° $2 + m + 1$ pentades voor getal van m cijfers.

Het BIBAND-molecuul laten we voorlopig buiten beschouwing, ons tot de laatste twee typen beperkend. Laat ons de cyclus beginnen op het moment, dat juist een molecuul geborgen is: het adres van de wegbergopdracht wordt met 1 verhoogd, en de machine staat klaar om een nieuw molecuul te assembleren. Opdrachtenkoppels worden met de functieletter van de a-opdracht voorop geponst, deze pentade is dus maximaal gelijk aan 29. Is de eerste pentade niet groter dan 29, dan

wordt deze met het volgende 7-tal tot opdrachtenkoppel geassembleerd. Resteren twee mogelijkheden: 30 of $31 = X$. Als X als eerste pentade gevonden wordt, wordt deze pentade overgeslagen en de volgende pentade, die weer een X kan zijn wordt aan het onderzoek van de eerste onderworpen. Dit wordt uitgedrukt door "Aan het begin van moleculen worden X -en geskipt". De X (vijf gaatjes) fungeert als "Erase" en maakt het mogelijk om tijdens het ponsen gedetecteerde fouten - mislagen - te corrigeren.

(Na bepaalde combinaties worden voor het nieuwe molecuul pentades 0 - "blank tape" - geskipt. Dit maakt het beschrijven van de band mogelijk. Begint echter het nieuwe molecuul met een 0 -opdracht, dan moet men het door een X vooraf laten gaan: zodra de X gelezen is - een $\neq 0$ - is het skippen van blank tape voorbij, alsnog kan X geskipt worden, zoals kan aan het begin van elk molecuul).

Anders wanneer de eerste pentade $F=30$ is. Dan komt er geen opdrachtenkoppel: wat er wel komt, hangt af van de volgende pentade; is deze niet groter dan 3 , dus $+$, $-$, $+$ of $-$, dan komt er een getal in alle andere gevallen hebben we te maken met een controlecombinatie. Bij getallen worden deze twee pentades, F en teken, gevolgd door de decimale cijfers, en tenslotte een X . Omdat het decimale gedeelte uit een niet van te voren vastgelegd aantal cijfers bestaat, moet n.l. een markering op de band aanwezig zijn, dat het laatste decimale cijfer geweest is. Vandaar de sluitletter. (In feite stopt de decimale opbouw en wordt het einde van het getal geacht bereikt te zijn, zodra een pentade, groter dan 15 gelezen wordt. Om der wille van de ordelijkheid wordt steeds de X gekozen, maar 16 t/m 30 zou hetzelfde effect hebben).

De a - en de b -opdracht worden volgens dezelfde conventies geponst; de a -opdracht voorop, de b -opdracht onmiddellijk hierop volgend. Wie zich in de b -opdracht met ponsen vergist - zo, dat er een gaatje teveel geponst is - moet ook de a -opdracht in X -en veranderen, omdat aan het begin van het molecuul, en niet aan het begin van elke opdracht X geskipt wordt.

Elke opdracht wordt in vier pentades geponst, in de volgorde (van links naar rechts) $f v r s$ (functie, vel, regel, sluitletter). De eerste pentade bepaalt dus, wat voor opdracht het is, de volgende twee bepalen het - aanvankelijke - adres. Aanvankelijk adres, omdat bij dit adres een addendum opgeteld wordt, dat door de sluitletter S bepaald wordt. De sluitletter heeft hier dus een andere functie dan bij getallen: niet om het einde te markeren - opdrachten worden immers in een vast aantal pentades geponst! - maar om (in eerste instantie) het adres te vermeerderen. Ingelezen wordt n.l. $f v r + [30 s]$ voor $s=0$ t/m 27 of $s=31$. Nu is $30 \ 31 = 0$, dus de sluitletter X laat de opdracht ongewijzigd. Dankzij de andere sluitletters is het moge-

lijk, om in programma's z.g. relatief te nummeren. Ieder programma valt n.l. natuurlijkerwijze uiteen in een aantal "losse" onderdelen, zoals b.v. hoofdprogramma, wat subroutines, een rij constanten, een rij parameters en een rij werkruimten. Het is gewoonte om zich tijdens het maken van het programma er niet om te bekommeren, hoe deze brokken ten opzichte van elkaar in het geheugen zullen staan. Alle adressen, die naar plaatsen in deze brokken verwijzen, nummert men ten opzichte van het eerste adres, en karakteriseert dit eerste adres - en daarmee de hele brok - door een of andere sluitletter (b.v. $s=15$). Als het gehele programma klaar is, en men de lengte van diverse onderdelen overzien kan, gaat men het geheugen indelen: men spreekt af waar de eerste opdrachten der brokken komen te staan. Als nu het beginadres van het stuk met $s=15$ in (30 15) is ingevuld, voordat de banden worden ingelezen - deze invulling van de correctie-adressen in kanaal 30 geschiedt door de z.g. voorponsing - wordt elk adres met sluitletter $s=15$ met het juiste bedrag vermeerderd. Voor de vermeerdering van het adres van de b-opdracht hoeft men niet het beginadres maal 2^{15} in kanaal 30 in te vullen: a- en b-opdracht worden door dezelfde sluitletter in hun eigen positie met hetzelfde bedrag vermeerderd. Bij de controlecombinatie, die het begin van een nieuw rijtje aangeeft, wordt eveneens van deze sluitletter gebruik gemaakt. Dankzij de faciliteit van de sluitletter in zijn functie van correctieletter is het mogelijk, om voor standaard subroutines standaardbandjes te maken, die te gebruiken zijn, ongeacht waar in het geheugen men de subroutine plaatsen wil. Subroutines bevatten immers altijd adressen, die afhankelijk zijn van de plaats, waar de subroutine in het geheugen staat. Bij voorbeeld de subroutine, die de goniometrische functie met behulp van een machtreeks uitrekent, zal bestaan uit een rijtje opdrachten (hoofdzakelijk vermenigvuldigingen en optellingen) gevolgd door een open plaats, waar de sprongopdracht terug naar het hoofdprogramma geplaatst wordt (de koppelopdracht) gevolgd door enige werkruimtes (voor 't argument en het kwadraat ervan) en de coëfficiënten van de machtreeks-ontwikkeling. Wordt deze subroutine de ene keer b.v. 160 plaatsen verder in het geheugen geplaatst dan een andere keer, dan zullen de adressen van de opdrachten, die de coëfficiënten aanhalen, omdat ook deze 160 plaatsen verder in het geheugen staan, 160 groter zijn. Het effect van de plaats in het geheugen is zuiver additief: de sluitletter voorziet hier juist in de behoefte. Het (variabele) beginadres wordt door de voorponsing ingevuld op adres 30 27 (het is conventie voor de standaardsubroutines de laatste sluitletters te gebruiken), de rest is onveranderlijke standaardband. Deze begint dus met de controlecombinatie, dat een rijtje opdrachten geborgen moet worden, te beginnen bij het adres ... dat door 30 27 is aangegeven: $v r s = 0 0 27$. Dit is de reden, dat het adresgedeelte van deze controle-

combinatie door een (correctie) sluitletter wordt afgesloten.

Door deze boven in grote trekken geschilderde functies bewijst het invoerprogramma door zijn versatiliteit de programmeur grote diensten. De conversies naar tweetallig stelsel worden hem bespaard, door de sluitletters kan hij standaardbanden voor de subroutines gebruiken en kan hij door zijn eigen programma te verdelen en verschillende sluitletters voor de dan niet te lange stukken te gebruiken, het v r gedeelte der opdrachten klein houden. De praktijk heeft uitgewezen, dat dit onschatbare diensten bewijst bij de samenstelling van langere programma's. Dit echter wordt betaald met de inleestijd van de banden. De optellingen van de correcties, de vorming en plaatsing van de opdrachten, die deze optellingen verrichten, en de tests aan het begin van ieder molecuul kosten tijd. Zodra een probleem vaker terugkomt, zal men hiervoor hetzelfde programma gaan gebruiken. Dit wordt ontworpen, in het gebruik getest, er wordt aan gevijld en na verloop van tijd is een mooi standaardprogramma ontwikkeld. Iedere keer, dat dit probleem nu terugkomt, worden dezelfde banden ingelezen, wordt dus dezelfde informatie op dezelfde plaats en in het geheugen ingevuld. Het is duidelijk inefficiënt om de machine iedere keer weer het ingewikkelde assemblage proces uit te laten voeren: als dit eenmaal gebeurd is, hoeft dit niet ten tweede male: aangezien de machine zelf de bandponser kan bedienen, kunnen we het programma, als het eenmaal ingelezen is, door de machine uit laten ponsen op een zodanige manier, dat van de dan geponste band de informatie zo snel mogelijk kan worden ingelezen. Dit geschiedt met de BIBAND (binaire band) waar de dertig binaire cijfers van rechts naar links over de 30 plaatsen van 6 opeenvolgende pentades verdeeld worden: de eerste pentade bevat de laagste vijf cijfers ($r_a = d_4 \text{ t/m } d_0$), de laatste pentade de hoogste vijf ($f_b = d_{29} \text{ t/m } d_{25}$). Een dergelijke band kan zeer snel gelezen worden. Er doet zich echter een complicatie voor: elke pentade is als openingspentade zinvol, m.a.w. we hebben geen F over om controlecombinaties aan te kondigen. Het einde van de BIBAND kan niet ra afloop op de band aangegeven worden. Daarom wordt bij de controlecombinatie, die een BIBAND annonceert, de lengte van de BIBAND aangegeven, d.w.z. het aantal adressen, dat door de BIBAND ingevuld wordt. Deze controlecombinatie is te lezen als de aankondiging, dat van nu af aan de pentades op de band anders geassembleerd moeten worden, benevens de indicatie, hoe lang deze wijziging in interpretatie van kracht is. Uit snelheidsoverwegingen is aan de BIBAND de beperking opgelegd, dat hij een even aantal adressen invult (dan hoeft slechts om de twee opdrachten getest worden, of het einde van de BIBAND bereikt is).

Men realiseer zich, dat het BIBAND-lezen, het gewijzigd interpreteren van de pentadegroepen, als uitstapje beschouwd moet worden: het BIBAND-lezen wordt vanuit het normale leesproces gelanceerd, na afloop komt de besturing weer in het normale leesprogramma. Dit laatste brengt

met zich mee, dat een normaal molecuul, dat volgt op de BIBAND, normaal gelezen wordt, en wordt geborgen op het adres, dat onmiddellijk volgt op het laatste adres, dat door de BIBAND is ingevuld. De controlecombinatie, die de plaats, waar de BIBAND moet gaan invullen, meeg geeft, is de normale controlecombinatie, die een nieuw rijtje aankondigt; deze en de BIBAND mogen ook best door een paar normale moleculen, die eerst geborgen moeten worden, gescheiden worden, de BIBAND begint dan zoveel plaatsen verder te bergen.

Als een band gelezen worde, wordt deze met het stuk blank tape voor- aan de band onder de bandlezer gelegd en men moet het invoerprogramma starten. Dit kan echter op twee manieren: start men de machine op opdracht 1 0 a, dan leest het invoerprogramma de band en bergt de moleculen ter plaatse. Start men evenwel de machine op 1 1 a, dan worden de opgebouwde moleculen niet in het geheugen geschreven, maar vergeleken met wat er al ter plaatse staat: zijn deze woorden gelijk, dan gaat de machine door, is er een verschil, dan stopt de machine. Op deze manier worden incidentele fouten tijdens het bandlezen meestal tijdig gedetecteerd. Men zegt: het invoerprogramma SCHRIJFT, resp. CONTROLEERT, het kent twee "standen".

Nog een groep controlecombinaties moet iets uitvoeriger onder de loupe genomen worden, n.l. de invulling van de **correctie-adressen**. De correctie-adressen 30 11 t/m 30 27 kunnen, mits een adres ingevuld moet worden, met een speciale controlecombinatie ingevuld worden. Deze invulling is eveneens een soort uitstapje: na de wijziging van het correctie adres gaat het wegbergen (of controleren) door, waar het invoerprogramma gebleven was.

Tevens was het mogelijk om deze adressen niet rechtstreeks in de controle te betrekken: als men een lange band inleest, waar verscheidene subroutines op gecopieerd staan, dan zal (30 27) verschillende waarden aannemen. Leest men de band een tweede keer, maar nu controlerend, dan wordt als de eerste waarde van (30 27) gelezen is, niet vergeleken - de machine zou dan zeker een discrepantie vinden - maar ingevuld, opdat de volgende moleculen ter vergelijking correct opgebouwd kunnen worden. De correctie-adressen fungeren als werkruimtes voor het invoerprogramma, slechts de werkelijke moleculen zijn aan de directe controle onderworpen. Wil men via de correctie-adressen andere parameters invoeren, die niet als adres te schrijven zijn, dan kan men deze controlecombinaties niet gebruiken.

Thans de systematische bespreking.

Start 1 0 a SCHRIJFT. Skipt blank tape.

Start 1 1 a CONTROLEERT. Skipt blank tape.

"Skipt blank tape" betekent, dat daarna eventuele X-pentades eveneens worden geskipt.

F +	"decimaal gedeelte"	X	positief geheel getal	Skipt X
F -	"decimaal gedeelte"	X	negatief geheel getal	Skipt X

Het decimale gedeelte wordt geponst van links naar rechts; nullen aan de hoge kant mogen weggelaten worden. Voor het inbrengen van ± 0 kan het decimaal gedeelte geheel weggelaten worden. Voor het inbrengen van ± 0 kan het decimaal gedeelte geheel weggelaten worden.

F +.	"decimaal gedeelte"	X	positieve breuk	Skipt X
F -.	"decimaal gedeelte"	X	negatieve breuk	Skipt X

De breuk wordt beschouwd 3 cijfers achter de komma te hebben en moet dus eventueel aan de lage kant met nullen aangevuld worden. Voor het aldus gevormde decimale gedeelte gelden dezelfde ponsconventies als bij de gehele getallen.

F 4 v r s plaatsaanduiding Skipt blank tape

Het volgende molecuul heeft betrekking op adres $v r + [30 s]$. Als dit een opdrachtenkoppel is, dat met een 0 begint, moet deze 0 van de eventuele blank tape, die op de combinatie F 4 v r s mag volgen, door een (of meer) extra X gescheiden worden.

F 6 f v r s Doe

Na F 6 wordt een complete opdracht geponst. Deze opdracht wordt door de besturing uitgevoerd. De bekendste toepassingen zijn onderstaande.

F 6 23 00 X Stop

De machine stopt, omdat de besturing deze opdracht met de conditie negatief ontmoet. Dit is de slotcombinatie op elke band.

F 6 7(15) v r s Exit

Op de band wordt het beginadres geponst: de machine gaat "onmiddellijk" werken, te beginnen bij de a(b)-opdracht op adres $v r + [30+s]$.

Deze opdracht komt te staan vlak voor de opdracht, die het opgebouwde molecuul wegbergt, dus op de plaats van de laatste opdracht, die het molecuul in **s** voltooit. Dit geeft twee toepassingen:

F 6 26 8 2 X Copieer uit G, Skip X

Op de plaats, die aan de beurt is, wordt de inhoud van de getalschakelaars geborgen.

F 6 10 v r s Copieer uit geheugen Skip X

Copieer op het adres, dat aan de beurt is, de inhoud van $v r + [30+s]$. (Men moet er van overtuigd zijn, dat betrokken adres al juist is ingevuld!)

F 9 m Sla m plaatsen over. Skip X
($m=1t/m31$)

Het adres van de variabele wegberg(aftrek-)opdracht - gelijk aan het adres van de plaats, die aan bod was om ingevuld te worden - wordt met m verhoogd. m plaatsen worden oningevuld gelaten, daarna gaan we gewoon door. Als tussen een rij opdrachten of constanten, een variabele plaats staat, ponsen we daarvoor F 9 1. Het voordeel boven nul op die plaats invullen, is dat als het programma een tijdje gedraaid heeft, deze band nog bruikbaar is voor controle: de veranderlijke, door het programma ingevulde inhoud, wordt bij de controle overgeslagen. F 9 0 is een ongevaarlijk edoch zinloos tussenvoegsel.

F 10 v r BIBAND. Skipt blank tape na de BIBAND.

Aankondiging van een BIBAND ter lengte van v vel en r regels; r moet even zijn. De $6(32 v+r)$ onmiddellijk op deze combinatie volgende pentades worden bij zessen gecombineerd tot 30-cijferwoorden; in elk zestal wordt steeds de volgende pentade aan de hoge kant naast de voorafgaande geplaatst. Deze woorden worden op successievelijke adressen opgeborgen, te beginnen bij het adres, dat aan bod was, om gevuld te worden.

Bij terugkomst in het normale leesproces staat het invoerprogramma klaar om het eerste adres na de binair ingevulde adressen te vullen. De besturing komt binnen in het punt "Skipt blank tape" (en dus daarna eventuele X-en). Als het eerst volgende molecuul met een 0 begint, moet dus een extra X ingelast worden.

F 11 v r s Invullen van de correcties. Skipt blank tape.
t/m

F 27 v r s Op 30 11, resp. 30 27 wordt ingevuld $v r + [30+3]$.

Voor het invullen van een correctie-adres mag men dus gebruik maken van andere correctie-adressen (mits eerder al ingevuld) zelfs van het betrokken correctie-adres zelve. Het volgende molecuul wordt geborgen op het adres, dat aan de beurt was, toen deze controle-

combinatie plotseling er tussen kwam. T.g.v. "skipt blank tape" gelden voor het volgende molecuul de gebruikelijke voorzorgen. Dit alles geldt, ongeacht de "stand" van het invoerprogramma.

F X

Wissel

Skipt blank tape.

Als het invoerprogramma SCHRIJFT, gaat het nu CONTROLEREN; als het CONTROLEERT, gaat het nu SCHRIJVEN. Deze wisseling wordt pas effectief na de eerstvolgende combinatie F 4 v r s. De controle-combinatie F X maakt het mogelijk om met de duplicator z.g. zelf-controlerende banden te maken. Men ponst achter een stukje F X, en dupliceert dit tweemaal op dezelfde band. Start men SCHRIJVEND, dan wordt dit stukje geschreven, gecontroleerd, en het invoerprogramma wordt weer schrijvend achtergelaten. Vanwege "Skipt blank tape" de gebruikelijke voorzorgen voor het volgende molecuul.

Het opdrachtenkoppel wordt geponst:

f_a	v_a	r_a	s_a	a-opdracht	
f_b	v_b	r_b	s_b	b-opdracht	Skip X

Beide regels worden van links naar rechts geponst, de b-opdracht onmiddellijk volgend op de a-opdracht.

Het gevormde molecuul luidt:

f_a	v_a	$r_a + [30+s_a]$
f_b	v_b	$r_b + [30+s_b]$

Opm: Als $[30+s_a]$ niet een adres is, maar een zo grote constante, dat de b-helft $\neq 0$ is, wordt wat in de b-helft staat, bij de b-opdracht ("eronder") opgeteld.

Als dit bij $30+s_b$ het geval is, wordt wat in de b-helft staat bij de a-opdracht ("erboven") van het opdrachtenkoppel opgeteld.

Copieer band.

Start men de machine op 2 29 a dan gaat de machine een pentade lezen, deze ponsen, en zo voort. De machine kan dus als band-copieerinrichting gebruikt worden. Als men de machine niet met de hand stopt, loopt tenslotte de band uit de bandlezer.

Voorponsing via G.

Start men de machine op 2 30 b, dan wordt G gecopieerd op 30 27, en de besturing wordt verwezen naar het begin van het (schrijvende) invoerprogramma. Door een bandje de sluitletter 27 te geven, kan men dus de plaats van een of ander (klein) service programma via de getalschakelaars bepalen. Het bandje moet zelfcontrolerend zijn uitgevoerd.

Invoerprogramma als subroutine.

Met de aanroep	26	0	4
	15	0	31

is het invoerprogramma als subroutine aan te roepen. De koppelopdracht wordt geplaatst op de plaats van de normale wegbergopdracht. Een molecuul wordt van de band gelezen, en met dit molecuul in (S) komt de bestemming terug in het hoofdprogramma. Hierbij mag niet vergeten worden

- 1^o dat het invoerprogramma door deze aanroep komt in de cyclus "skip blank tape", en dat dus de moleculen zonodig door extra X voorafgegaan moeten worden.
- 2^o controlecombinaties, die (31 31) veranderen, vermeden moeten worden: de koppelopdracht wordt dan bedorven.

Als de combinatie F 4 v r s wordt aangetroffen, wordt het invoerprogramma beschouwd in de schrijvende stand te staan.

Het in- en uitvoerprogramma in het geheugen.

Het in- en uitvoerprogramma beslaat in het geheugen

6 opdrachtkanalen, n.l. 0 t/m 5

1 getallenkanaal , n.l. 30

1 gemengd kanaal , n.l. 31

In het normale gebruik zijn de eerste zes kanalen gescheiden geschakeld: het invoerprogramma staat geblokkeerd, omdat de getal selectie van de betroffenen kanalen aan zes andere koppen is verbonden. Deze eerste zes kanalen zijn dus voor berging van numerieke gegevens altijd bruikbaar.

In kanaal 30 staan de correctieadressen der sluitletters: voor de "variabele" sluitletters zijn dit dus werkruimtes van het invoerprogramma, voor de sluitletter X betekent dit een constante, n.l. $+0 = [30\ 31]$. Plaats 28, 29 en 30 uit kanaal 30 worden voor de typcontrole gebruikt, en moeten bij werkend programma als "bezet" beschouwd worden.

In kanaal 31 staan alle overige constanten, die het invoerprogramma nodig heeft, benevens de parameters, die de typ-routines specificeren. Ten leste bevinden zich hier de adressen, die de variabele opdrachten van het werkende invoerprogramma bevatten. Om die reden is 31 als gemengd kanaal geschakeld. Tevens is kanaal 30 als gemengd kanaal geschakeld, en wel om de volgende reden: om der wille van de ordelijkheid is het gewenst, dat "special purpose" in- en uitvoerprogrammatjes (utility leesprogramma voor breuken, pons Biband, en b.v. testprogrammatjes) in een vast kanaal staan. Kanaal 30 is hiervoor aangewezen, omdat het toch bezwaarlijk is, hier een onderdeel van een normaal programma te plaatsen: 30 bevat de tijdens het inlezen onontbeerlijke werkruimtes der correctieadressen. Als echter het programma ingelezen is, kan men de plaatsen van 30 0 t/m 30 27 rustig gebruiken voor het inlezen van een absoluut geponst speciaal programma, b.v. "pons Biband". De ponsmachine is een belangrijk uitvoerorgaan van de machine; toch is het ponsprogramma niet in het standaard in- en uitvoerprogramma opgenomen, omdat de verlangde faciliteiten en conventies zo kunnen variëren, dat het niet gewenst leek, dit door een algemeen programma te laten doen. Een tweede overweging is, dat het probleem van binair-decimale conversie zich hier meestal niet voordoet, waardoor de betroffenen programma's eenvoudiger worden en waarschijnlijker beter ad hoc gemaakt kunnen worden, aangepast aan de speciale situatie. Wel zal in dit deel een voorbeeld van een dergelijk ponsprogramma worden opgenomen.

"Utility leesprogramma's voor breuken" zijn programma's, die banden met breuken sneller lezen dan het standaard invoerprogramma.

Deze snelheid wordt gewonnen ten koste van de versatiliteit: een dergelijk programma kan b.v. enkel breuken met een vast aantal (geponste) cijfers lezen; deze rij breuken mag niet door andere moleculen onderbroken worden. Voor het inlezen van b.v. een matrix loont dit echter de moeite.

Herstel vrije kanalen.

6 kanalen van het invoerprogramma zijn geblokkeerd: deze informatie staat veilig op de trommel. Wat zich echter op kanalen 30 en 31 bevindt, is door de getalselectie bereikbaar en kan dus bij deraillement van de machine of onjuiste programmering vernield worden. Als deze constanten niet goed in het geheugen staan, kan het invoerprogramma geen normale banden lezen. Om het met de hand inzetten van deze constanten te vermijden is een speciaal bandje gemaakt, genaamd "HERSTEL VRIJE KANALEN". Om dit bandje in te lezen start men de machine op adres 2 0a, waar het programma begint, met Skip blank tape: men hoeft dit bandje dus niet met een speciale pentade onder de bandlezer te leggen. De constanten worden door de eerste helft van dit bandje ingevuld, door de tweede helft gecontroleerd, deze controle wordt afgesloten door de wissel FX, zodat het invoerprogramma "schrijvend" in de cyclus skip blank tape wordt achtergelaten. Ieder programma dient voor op de band het bandje "HERSTEL VRIJE KANALEN" gecopieerd te hebben.

In de z.g. service-uitvoering wordt dit bandje door de stop combinatie F 6 23 00 X afgesloten. Dit bandje wordt voor de veiligheid ingevoerd b.v. als, nadat de machine twee uur gewerkt heeft, een nieuwe rij constanten wordt ingelezen. Met het oog hierop is het bandje "HERSTEL VRIJE KANALEN" zo uitgevoerd, dat de typconstanten, die ook in kanaal 31 staan, ongewijzigd blijven.

De volledige tekst van het bandje "Herstel vrije kanalen" staat in dit rapport, onmiddellijk volgend op het uitgeschreven in- en uitvoerprogramma. Elke regel is van links naar rechts geponst: de opdrachten, die door het binaire leesprogramma worden gelezen, staan dus "achterstevoren".

Rechts staat ter verduidelijking de lopende inhoud van de drie laatste adressen van het geheugen: hieronder vallen de wegbergopdrachten van het normale en het binaire bandleesprogramma. Inhouden, die als zodanig niet gehoorzaamd worden, staan tussen haakjes.

Het typen.

De typroutines typen de absolute waarde van (S'). Door de aanroep kan men regelen, of de absolute waarde al dan niet door het teken wordt voorafgegaan, tevens of [S'] - dus als geheel getal - of {S'} - dus als breuk - wordt uitgetypt. Men lette er op, dat de functie-omschrijvingen "Typ G (=Geheel getal)" en "Typ B (=Breuk)" slaan op de interpretatie van (S') als [S'] , resp. {S'} , en niet op het beeld op de pagina: men kan n.l. bij "Typ G" een punt inlassen, dus "de komma zoveel plaatsen naar links schuiven", bij "Typ B" kan men de punt enige plaatsen naar rechts schuiven, ja zelfs geheel onderdrukken. Deze en andere nadere specificaties, zoals aantallen uittetypen cijfers, plaats van (c.q. geen) punt, imperatief en facultatief typen, worden door twee groepen constanten bepaald, die op vaste plaatsen in vel 31 worden ingevuld. De ene groep specificeert de operatie "Typ G", en bestaat uit 1 constante, en 2 z.g. sprongenparen; de andere bepaalt "Typ B", en bestaat uit 4 constanten en 2 sprongenparen. Zonder tussentijdse verandering van deze constanten kan men dus een type [S'] en een type {S'} uittypen, terwijl men het teken, door de verschillende aanroepen, dan wel, dan niet kan onderdrukken.

De typroutine kent vier aanroepen:

"Typ G, met teken" 26 0 4 X
 7 4 0 X =)
 =)

"Typ G, zonder teken" 26 0 4 X
 15 4 1 X =)
 =)

"Typ B, met teken" 26 0 4 X
 7 4 8 X =)
 =)

"Typ B, zonder teken" 26 0 4 X
 15 4 9 X =)
 =)

Onder "Typ G, met n cijfers" wordt verstaan, dat van |[S']| de n succesievelijke decimale cijfers (van de (n-1)^{ste} 10-macht t/m het eenheden-cijfer) worden getypt. Dit impliceert, dat [S'] in absolute waarde kleiner moet zijn dan 10ⁿ.

Onder "Typ B, met m cijfers" wordt verstaan, dat van |[S']| (exact op m cijfers afgerond) de m cijfers (van de -1^{ste} t/m de -m^{de} 10-macht) worden uitgetypt. Door de afronding mag |[S']| niet gelijk aan 1 worden!

Opm: Onder "A cijfers imperatief" verstaan we alle A cijfers; onder "A cijfers facultatief", de A cijfers, maar eventuele nullen aan de hoge kant door spaties vervangen, met dien verstande dat het laatste cijfer altijd getypt wordt (d.w.z: bij facultatief typen van het getal 0 blijft het papier niet schoon, maar wordt de laatste nul wel getypt).

De sprongenparen, die afhankelijk zijn van de plaats (c.q. het onderdrukken) van de punt en het totale aantal cijfers, kunnen samengesteld worden met behulp van de volgende tabel.

A=	A cijfers,imperatief	A cijfers,facultatief	A cijfers,facultatief, na extra spatie
1	7 5 30 X		15 5 17 X
2	15 5 28 X	7 5 16 X	15 5 14 X
3	7 5 27 X	7 5 13 X	15 5 11 X
4	15 5 25 X	7 5 10 X	15 5 8 X
5	7 5 24 X	7 5 7 X	15 5 5 X
6	15 5 22 X	7 5 4 X	15 5 2 X
7	7 5 21 X	7 5 1 X	15 4 31 X
8	15 5 19 X	7 4 30 X	15 4 28 X

Voorts zijn sprongen, die in de sprongenparen verwerkt worden:

"punt in G" : 15 4 16 X	"punt in B" : 15 4 26 X
"G klaar" : 7 3 25 X	"B klaar" : 15 3 19 X

Elk sprongenpaar verzorgt de "ingang" en "uitgang" in een decimaal typend stuk programma ("ingang" bepaalt aard en aantal der cijfers; "uitgang" bepaalt, wat er daarna gebeuren gaat). De sprong op de b-plaats fungeert als ingang en wordt het eerste gehoorzaamd! Deze stuurt in het algemeen de besturing naar de reeks typ-opdrachten, na afloop van het stukje "continu cijfers typen" gaat de besturing via de sprongopdracht op de a-plaats van het betrokken sprongenpaar verder. Het 1^{ste} sprongenpaar heeft betrekking op wat voor de punt getypt wordt (inclusief het geval, dat er geen punt getypt wordt); het 2^{de} sprongenpaar, dat pas aangehaald wordt, als de punt getypt wordt, op wat na de punt komt. Als geen punt getypt wordt, wordt het tweede sprongenpaar niet gebruikt.

Enkele voorbeelden mogen de vorming der sprongenparen toelichten.

Geheel getal,	1 ^{ste} sprongenpaar G: a 7 3 25 X "G klaar"	(Uitgang)
4 cijfers fac.	b 15 5 8 X "sp.4 fac."	(Ingang)
na extra spatie	2 ^{de} sprongenpaar G :	wordt niet gebruikt.

Geheel getal,	1 ^{ste}	sprongenpaar G: a	15	4	16	X	"punt in G" (Uitgang).
3 cijfers voor,		b	7	5	13	X	"3 fac" (Ingang)
4 na de punt: geh. gedeelte fac; 2 ^{de}		sprongenpaar G: a	7	3	25	X	"G klaar" (Uitgang)
geen extra spatie.		b	15	5	25	X	"4 imp" (Ingang)
Geheel getal,	1 ^{ste}	sprongenpaar G: a	15	4	16	X	"punt in G"
6 cijfers fac,		b	15	5	2	X	"sp. 6 fac"
na extra spatie: gevolgd door	2 ^{de}	sprongenpaar G: a					"wordt niet gehoorzaamd"
punt		b	7	3	25	X	"G klaar"

Opm. Als na de punt niets meer getypt hoeft te worden, kunnen we via de b-opdracht van het tweede sprongenpaar de besturing meteen door 1 laten gaan naar "G klaar".

Breuk, 2 cijfers voor,	1 ^{ste}	sprongenpaar B: a	15	4	26	X	"punt in B"
5 cijfers na		b	15	5	28	X	"2 imp"
de punt; alles	2 ^{de}	sprongenpaar B: a	15	3	19	X	"B klaar"
imperatief.		b	7	5	24	X	"5 imp"
Breuk, niets	1 ^{ste}	sprongenpaar B: a					"wordt niet gehoorzaamd"
voor de punt,		b	15	4	26	X	"punt in B"
na de punt	2 ^{de}	sprongenpaar B: a	15	3	19	X	"B klaar"
4 cijfers			15	5	25	X	"4 imp"

Opm. Ook als niets voor de punt getypt worde, kunnen we via de b-opdracht - nu van het 1^{ste} sprongenpaar - meteen de besturing door laten gaan.

Plaatsing der constanten.

Constanten voor "Typ G" (n cijfers)

$$\begin{aligned}
 [31, 23] &= 10^n \\
 \langle 31, 19 \rangle &= 1^{\text{ste}} \text{ sprongenpaar G} \\
 \langle 31, 14 \rangle &= 2^{\text{de}} \text{ sprongenpaar G}
 \end{aligned}$$

Constanten voor "Typ B" (m cijfers) (N.B.: $2^{28} = 2684\ 35456$)

$$[31,17] = [10^{-m} \cdot 2^{28}] \quad (\text{d.w.z: gehele gedeelte, dus b.v. voor } m=6 : 268)$$

$$[31,18] = 10^m$$

$$[31,26] = 10^m [10^{-m} \cdot 2^{28}]$$

$$[31,27] = 2^{29} - (10^m + 1) = 5368\ 70911 - 10^m$$

$$\langle 31,21 \rangle = 1^{\text{ste}} \text{ sprongenpaar B}$$

$$\langle 31,15 \rangle = 2^{\text{de}} \text{ sprongenpaar B}$$

Typ-controle; n getallen per regel.

Controle op het typen is mogelijk, doordat de typroutines er naar streven, met exact dezelfde inhoud van S in het hoofdprogramma terug te komen, als waarmede ze werden aangeroepen. Als aan $(S'')=(S')$ voldaan is, wordt aangenomen, dat het typen feilloos is geschied. Door het te typen getal dan van S af te trekken en te onderzoeken of $S = -0$ is, strekt de controle zich ook uit over het aanhalen van het betrokken getal uit het geheugen. In geval van een geconstateerde typfout moet het getal overgetypt worden; opdat dit een regel lager, maar in dezelfde kolom geschiedt, moet een telling van de plaats in de regel bijgehouden worden, opdat de machine uit kan rekenen, hoeveel TAB-signalen na de TWN R gegeven moeten worden, om de wagen van de schrijfmachine weer in dezelfde positie te krijgen. Om die reden is de typ-controle met de regelindeling gekoppeld.

Deze routine wordt ingesteld door twee constanten:

$$[30,28] = 16n - 11, \text{ mits TAB, zo goed,} \quad n \text{ getallen per regel}$$

$$16n - 16, \text{ mits SPATIE, zo goed}$$

$$[30,29] = 16m + 11, \text{ waar m een maat is voor vertraging na TWN R}$$

(n.l: $(n+1)50 \text{ msec}$)

De inloopsaanroep luidt

$$\begin{array}{cccc} 26 & 0 & 4 & X \\ 7 & 3 & 31 & X \end{array} =)$$

=) met de functie

- 1) Telling wordt ingesteld op het begin van de regel
- 2) TWN R-sig naal wordt gegeven, na vertraging gevolgd door een TAB-sig naal.

Deze aanroep dient de besturing altijd te ontmoeten aan het begin van een probleem! Kleine tijdwinst kan bij deze aanroep geboekt worden, door van te voren $(S') = -0$ te maken.

Na elke typroutine-aanroep volgt de controle-aanroep

```

26 0 4 X
7 3 0 X =)
=) ..... met de functie

```

1) zolang steeds $(S') = -0$, geeft de routine of een TAB-sigitaal, of een SPATIE (afhankelijk van $[30,28]$), uitgezonderd elke n^{de} keer, waar een TWRN-sigitaal wordt gegeven, na vertraging door een TAB gevolgd. De besturing komt terug in het hoofdprogramma, met de conditie negatief.

2) als een keer $(S') \neq -0$ - dat wil zeggen, een getal fout getypt is - dan geeft de routine TWRN, en na de vertraging j TAB-signalen, als het j^{de} getal in de regel mislukt is. De wagen staat dan weer in de goede positie, de besturing komt in het hoofdprogramma terug met de conditie positief.

Opm.1 Het eerste getal begint dus niet bij de kantlijn, maar bij de eerste tabulatorstop (circa vijf plaatsen rechts van de kantlijn).

Dit, omdat de wagen bij het teruglopen soms een plaats terugslaat.

Opm.2 Bij een TAB-sigitaal moet de wagen minstens 3 plaatsen doorlopen.

Als men veel getallen op een regel wil typen, kan het zijn, dat men zich niet zoveel ruimte tussen de getallen kan veroorloven. Dan scheidt men de getallen door een SPATIE (door $[30,28] = 16n - 16$ te maken). De tabulatorstoppen dienen dan met zorg gezet te worden, omdat deze ingeval van een fout wel gebruikt worden!

Het volgende voorbeeldje illustreert de uitvoering van de typcontrole: hierbij wordt aangenomen dat op adres V,R het programma staat, dat $[v,r]$ als geheel getal zonder teken uittijpt.

V,R+3,a	-+	V,R	a	10	v	r	X	(v,r) $\neq$ (S)
			b	26	0	4	X	
		V,R+1	a	15	4	1	X	=) "Typ G" zonder teken
	=)		b	9	v	r	X	-0 $\neq$ (S)
		V,R+2	a	26	0	4	X	
			b	7	3	0	X	=) "Typcontrole"
	=)	V,R+3	a	6	V	R	X	-+ als fout
			b					

Als het typen goed gegaan is, vormt de 9-opdracht op V,R+1 b -0 in S, de besturing komt in V,R+3 a met de conditie negatief, en de besturingsverplaatsing wordt genegeerd. Als het het n^{de} getal van de regel was, is TWRN etc. gegeven, anders TAB of SPATIE.

Is echter bij het typen een fout opgetreden, dan zal de 9-opdracht op V,R+1 b geen -0 in S vormen; de besturing arriveert dan met de

conditie positief bij de conditionele sprong op $V, R+3$ a, die nu wel gehoorzaamd wordt en het getal wordt opnieuw getypt. Inmiddels is echter het papier een regel opgevoerd en de wagen is in die positie gebracht, waarin hij stond, toen aan het mislukte getal begonnen werd.

De opdrachten van het In- en Uitvoerprogramma.

Kanaal 0; 1^{ste} helft

0.24b=+	0	0	27	0	1	HANDREGISTER test cijfer of teken
			24	4	10	
	1		29	20	30	
			6	2	27	
	2		25	4	10	
			24	16	10	
	3		7	2	28	
			2	31	9	
	4		0	31	10	
			1	31	25	
	5		15	1	1	
			28	0	16	
	6		22	0	0	
			28	8	16	
31.30a=+	7		22	0	0	WISSEL FX (De som minus de een, geeft de ander).
			3	31	29	
	8		24	17	0	
			0	31	22	
	9		24	17	0	
31.30a -+			29	20	30	BIBAND: controle. Stop, als (A) fout. Stop, als (S) fout ophogen wegbergopdrachten en test voor einde.
	10		6	1	2	
			1	31	16	
	11		5	31	29	
			26	29	5	
2.23a -+	12		24	17	0	Einde "nalaatste" ophoging voltooid Lees twee BIBAND-moleculen.
			26	29	5	
	13		24	17	0	
			26	29	5	
	14		24	17	0	
			26	29	5	
	15		24	17	0	
			26	29	5	

Kanaal 0; 2^{de} helft.

0	16	24	17	0		Voortzetting: Lees twee BIBAND-Moleculen.
		26	29	5		
17		24	17	0		
		26	29	5		
18		24	17	0		
		26	29	5		
19		24	17	0		
		26	29	5		
20		24	17	0		
		26	29	5		
21		24	17	0		
		26	29	5		
22		24	17	0		
		26	29	5		
23	1.19a=+	7	31	29	=+	Naar variabele wegbergopdracht(en). D 10 Splits Wissel en inv. corr.adressen.
		25	4	20		
24		28	20	30	-+	-+ Wissel FX plaats var.opdracht. (nl. welk correctieadres wordt ingevuld). Ga correctie le- zen.
		14	0	3		
25		0	31	7		
		25	5	1		
26		4	31	30		
		2	31	8		
27		15	1	12	=+	(als adres b-opdracht). BANDLEZEN. Decimale cyclus.
		31	0	0		
28	0.30a=+	24	16	10		
		26	21	30		
29	1.30a-+	28	20	5		
		26	20	25		
30		6	0	28	-+	----- S-+A; dan +, -, +. of -. effectief.
		26	28	30		
31		7	31	30	=+	SUBROUTINEAANROEP.
		4	31	31		

Kanaal 1; 1^{ste} helft.

1	0	2	31	9		BANDLEZEN. START
		15	1	1	=+	LEES EN SCHRIJF
	1	2	31	10		START:
0.5a;1.0b		4	31	25		LEES EN CONTROLEER
0.10a;1.3b-+	2	26	31	30		Skip blank tape
		29	8	16		
	3	6	1	7	-+	
		7	1	2	=+	
31.31b=+	4	28	8	16		CONTROLE
		22	0	0		
31.31b-+	5	26	4	1		Increment voor
1.31a -+		0	31	31		Ophoging van wegberg-
	6	4	31	31		opdracht.
1.16a -+		26	31	30		Lees en onderzoek 1 ^{ste}
1.3a -+	7	25	12	29		pentade van nieuw mo-
		28	30	30		lecuel
	8	14	1	14	-+	-+ F of X
		24	30	25		Dus f _a
	9	24	31	25		v _a
		24	31	25		r _a
	10	2	31	11		Const.voor a-corr.
		24	21	30		s _a
1.28b -+	11	4	31	30		plaats opdracht
		24	31	25		f _b
	12	2	31	2		Const.voor b-corr.
1.27a;0.27a-+		24	31	25		v _b
	13	24	31	30		r _b
		24	21	30		s _b
	14	7	1	20	=+	naar:PLAATS en DOE
1.8a =+		25	12	1		Splitsing tussen
	15	24	17	0		1 ^{ste} pentade = F of
		28	30	30		= X

Kanaal 1; 2^{de} helft

1	16	<u>14</u>	<u>1</u>	<u>6</u>	-+	----- -+ Als =X -----
		26	21	30		Lees D (van FD) test of
	17	24	17	0		D > 10
		25	4	10		(= invullen correctie-adres
	18	24	17	0		of Wissel)
		28	20	30		
	19	<u>14</u>	<u>0</u>	<u>23</u>	-+	----- -+ D > 10 -----
		<u>0</u>	<u>31</u>	<u>1</u>		Vorm sprong
1.14a-+	20	4	31	29		PLAATS en
		<u>7</u>	<u>31</u>	<u>29</u>	=+	DOE
	21	<u>31</u>	<u>0</u>	<u>0</u>		
31.29a=+		2	31	3		D=0 ("+")
	22	<u>7</u>	<u>1</u>	<u>29</u>	=+	
" =+		2	31	4		D=1 ("-")
	23	<u>7</u>	<u>1</u>	<u>29</u>	=+	
" =+		2	31	5		D=2 ("+.")
	24	<u>7</u>	<u>1</u>	<u>29</u>	=+	
" =+		2	31	6		D=3 ("-.")
	25	<u>7</u>	<u>1</u>	<u>29</u>	=+	
" =+		2	31	7		D=4 Contr.comb.
	26	4	31	30		"Ga schrijven"
		2	31	0		(Als adres van de b-opdracht
	27	<u>15</u>	<u>1</u>	<u>12</u>	=+	gelezen).
" =+		27	4	1		D=6 DOE
	28	0	31	7		(Als b-opdracht gelezen)
		<u>7</u>	<u>1</u>	<u>11</u>	=+	
1.22t/m25a=+	29	4	31	30		Voortzetting Getal.
		26	12	0		Ga door naar
	30	<u>15</u>	<u>0</u>	<u>28</u>	=+	Decimale cyclus.
31.29a=+		26	21	30		D=9: Lees increment.
	31	<u>15</u>	<u>1</u>	<u>5</u>	=+	Naar ophoging
" =+		<u>15</u>	<u>2</u>	<u>3</u>	=+	BIBAND

Kanaal 2; 1^{ste} helft

START:HERSTEL VRIJE KANALEN

(Skip blank Tape)

2	0	26	21	5	
		28	20	30	
1		6	2	0	++
		26	12	13	-----
					plaats quasiwegberg.
2		24	30	20	opdracht op 31 31
		25	12	2	
3		12	31	31	<S>=000;12 31 30

1.31b-+		10	31	31	Inloop BIBAND.
4		26	28	15	<A>=9(12)VR <S>=0,0,0,7,1,4(5)
		10	31	31	=7,1,4(5).9(12),V,R
5		26	28	10	=V,R,9(12),V,R, =0,0,7,1,4(5),9(12)
		25	12	8	=0,0,7,1,4(5),1(4)
6		24	30	30	OS eff.
		26	28	5	=1(4),V,R,9(12),V,R =0,0,0,7,1,4(5)
7		24	4	1	=1(4),V,R,9(12),VR+1
		24	12	30	=0,0,0,7,2,2(3)
8		24	30	5	=2(3),0,0,0,7,2
		24	12	14	=2(3),0,0,0,7,16
9		4	31	29	Eerste tweetal geborgen
		24	30	24	=0,0,0,15,0,4(6)
10		24	12	1	=0,0,0,15,0,5(7)
		12	31	30	Sprong daaropvolgend geborgen
11		26	31	25	=0,0,0,0,v,0
		24	31	30	=0,0,0,0,v,r
12		12	31	28	Bergt lengte van Biband
		24	30	15	=0,v,r,0,0,0
13		2	31	31	<A>=7.1.4(5);9(12),V,R
		0	31	28	=7.1.4(5);9(12),V+v,R+r
14		4	31	31	Bergt "doorgaande wegbergopdracht
		8	31	28	=0,v,r,0,v,r
15		8	31	29	Lengte + eerste geeft "nalaatste"
		7	2	20	+=

Kanaal 2; 2^{de} helft.

2	16	26	4	0	Inloop SERIEHANDINZET
		26	8	2	
		17	24	28 15	
			0	31 1	
		18	12	31 29	
2.15b	=+ 20	24	4	24	increment in 31 29 geschreven
		19	24	20 15	Eerste wegbergopdracht
			15	2 24	gaat naar 31 30
			12	31 16	Voltooiing Inloop BIBAND
			25	12 2	"nalaatste" is geborgen; nu
31.30b	=+ 21	24	30	15	minus 2; ook in de b-positie
			25	12 2	minus 2
		22	24	30 15	weer goed
			12	31 22	bergt "laatste".
		23	15	0 11	Naar BIBANDCYCLUS
2.19b	-+ 24		2	31 29	SERIEHANDINZET
			0	31 30	ophoging van variabele
			4	31 30	wegbergopdracht
		25	31	0 0	Stop
			26	8 2	Lees uit G in S
0.1b	=+ 27		31	0 0	Stop (ook na "H" in S)
			7	31 30	Naar wegbergopdracht
			4	31 28	HANDREGISTER.Bergteken
			27	12 0	ZET (S) =0
			31	0 0	STOP
0.3a	=+ 28		7	3 14	Naar "Tekens effectief"
		2.30a	26	21 30	Copieer BAND
			28	21 5	
		30	7	2 29	
			26	0 2	Vul [G] in voor s = 27
	31		4	30 27	
			7	1 0	Ga Band lezen.

Kanaal 3; 1^{ste} helft.

3	0	4	31	28		TYPCONTROLE
		28	8	16		test (S) ≠ -0?
	1	14	3	4	++	++ als fout, dan TAB.-----
		26	4	16		als goed
	2	1	30	30		tel of einde van
		5	30	30		regel bereikt is;
	3	6	3	8	++	++ nog niet, dan TAB-----
3.31b	++	2	30	28		herstel
	4	4	30	30		telling-----
3.1a	++	2	30	29		TWNR
	5	28	0	8		
3.6b	++	25	4	16		vertraging
	6	28	20	30		na
		14	3	5	++	TWNR-----
	7	28	8	16		Test weer of fout
		6	3	9	++	++ TWNR was na fout-----
3.3a	++	28	0	8		TAB, als goed
		7	31	28	==+	Cond.-: naar link.
3.7b	==+	2	30	30		Na fout
		1	30	28		zet telling
	10	4	31	29		aantal TAB's-----
3.30b	++	26	4	10		
	11	28	0	8		TAB
		18	31	30		(verm.voor vertraging)
	12	26	4	16		
		0	31	29		Telling
	13	4	31	29		
		15	3	29	==+	nog een verm.
2.28b	==+	3	31	28		HANDREGISTER
		28	20	2		teken effectief
	15	6	3	17	++	++ als geheel-----
		26	28	30		als breuk

Kanaal 3; 2^{de} helft.

3	16	11	31	13		HANDREGISTER; voortzetting deel door 10 ⁸ -----
		20	31	20		
		3	31	28		
3.15a	-+	17	29	20	1	+ ? -----
			6	2	26	
			27	16	1	
31.16a	=+	18	7	2	26	-+ als - en -. ----- -(S) ~~≠~~ (S) als + en +. -----
			8	31	25	
			29	30	30	
3.21a	-+	19	8	31	26	B klaar: TYPROUTINE. test of verhoging door afronding -----
			14	3	22	
			25	4	9	
3.21a	-+	20	9	31	27	-+ als geen verhoging afronding ----- weer ongedaan ----- op teken na
			24	16	10	
			3	31	25	
31.16a	=+	21	24	28	30	Controle-resultaat voltooid door deling door 10 ^m -----
			20	31	18	
			15	3	26	
3.24b	-+	22	9	31	31	=+ verzorg teken ----- G klaar TYPROUTINE -----
			8	31	23	
			24	16	10	
3.13b	=+	23	2	31	24	Controleresultaat op teken na maak ----- teken (") in orde. -----
			24	17	0	
			29	20	30	
3.13b	=+	24	6	31	22	-+ als -, dan naar lmk. wissel (S) van teken. -----
			27	16	1	
			7	31	22	
3.13b	=+	25	6	31	28	Alsnog naar lmk. ----- TYPCONTROLE; -+cond+naar lmk Verm.voor verhoging -----
			18	31	30	
			15	3	10	
3.13b	=+	26	4	31	28	=+ nog een TAB ----- TWNr: INLOOPS -----
			15	3	3	
			15	3	3	

AANROEP

Kanaal 4; 1^{ste} helft.

4	0	4	31	22	
		26	0	4	
	1	15	4	23	=+
		12	31	24	
	2	6	4	3	-+
		11	31	24	
4.2a	-+	3	4	31	22
31.30a	-+		26	28	30
		4	21	31	23
			25	12	1
	5		4	31	31
			2	31	19
4.16a	-+	6	4	31	16
			27	4	0
	7		15	31	16
			31	0	0
	8		4	31	22
			26	0	4
	9		15	4	23
			12	31	24
	10		6	4	11
			11	31	24
4.10a	-+	11	4	31	22
31.30a	-+		12	31	28
		12	24	17	0
			19	31	18
	13		13	31	25
			26	30	28
	14		9	31	17
			9	31	28
	15		24	17	0
			2	31	21

TYP GEHEEL GETAL MET TEKEN

naar subr. "Typ teken"

TYP GEHEEL GETAL ZONDER TEKEN.

vorm

absolute waarde

(2 aanr. komen samen)

arithmetische voorbereiding

berg "REST"

1^{ste} sprongenpaar S

berg 1^{ste} sprongenpaar

en -0~~7~~(A)

via b-sprong naar het typen

TYPBREUK MET TEKEN

naar subr. "Typ teken".

TYPBREUK ZONDER TEKEN.

vorm

absolute waarde

(2 aanr. komen samen)

arithmetische
vorbereiding.

berg "STAART"

rond

exact

af.

1^{ste} sprongenpaar B

Kanaal 4; 2^{de} helft

4	16	7	4	6	=+	Als in "TYP G" verder
31.16a	=+	4	31	28		PUNT IN G
	17	2	31	14		2 ^{de} sprongenpaar G
		15	4	27	=+	
4.25a	=+	18	27	16	1	voortzetting
			26	4	12	interne
4.26a	-+	19	28	0	8	subroutine
			27	4	12	"typteken (S)"
	20		24	0	8	en vorm abs.waarde
			4	31	24	
	21		7	31	30	naar link
4.28a	=+		26	4	14	Typ de punt
	22		28	0	8	en
			2	31	28	herstel laatste cijfer
	23		15	31	16	ga weer typen (b-sprong)
4.9a;4.1a=+			24	4	2	"2 verder" terug
	24		4	31	30	subroutine
			29	30	30	"Typ Teken"
	25		6	4	18	-+ als (S) neg.
			26	4	13	"13" = +
	26		7	4	19	
31.16a	=+		4	31	28	PUNT IN B
	27		2	31	15	2 ^{de} sprongenpaar B
4.17b	-+		4	31	16	naar:
	28		15	4	21	Typ de punt
31.16b	=+		28	0	8	sp;8 fac.FACULTATIEF TYPEN.
	29		27	4	31	
			24	0	8	
31.16b	-+	30	24	16	10	8 fac.
			28	0	16	
	31		6	5	20	-+
31.16b	-+		28	0	8	sp.7 fac.

Kanaal 5; 1^{ste} helft

Voortzetting FAC.TYPEN.

	5	0	27	4	31	
			<u>24</u>	<u>0</u>	<u>8</u>	
31.16b	++	1	24	16	10	7 fac.
			28	0	16	
		2	<u>14</u>	<u>5</u>	<u>21</u>	++
"	++		28	0	8	sp.6 fac.
		3	27	4	31	
			<u>24</u>	<u>0</u>	<u>8</u>	
"	++	4	24	16	10	6 fac.
			28	0	16	
		5	<u>6</u>	<u>5</u>	<u>23</u>	++
"	++		28	0	8	sp.5 fac.
		6	27	4	31	
			<u>24</u>	<u>0</u>	<u>8</u>	
"	++	7	24	16	10	5 fac.
			28	0	16	
		8	<u>14</u>	<u>5</u>	<u>24</u>	++
"	++		28	0	8	sp.4 fac.
		9	27	4	31	
			<u>24</u>	<u>0</u>	<u>8</u>	
"	++	10	24	16	10	4 fac.
			28	0	16	
		11	<u>6</u>	<u>5</u>	<u>26</u>	++
"	++		28	0	8	sp.3 fac.
		12	27	4	31	
			<u>24</u>	<u>0</u>	<u>8</u>	
"	++	13	24	16	10	3 fac.
			28	0	16	
		14	<u>14</u>	<u>5</u>	<u>27</u>	++
"	++		28	0	8	sp.2 fac.
		15	27	4	31	
			<u>24</u>	<u>0</u>	<u>8</u>	

Kanaal 5; 2^{de} helft.

31.16b-+ 5	16	24	16	10	2 fac. voortzetting FAC.TYPEN.
		28	0	16	
	17	6	5	29	-+
31.16b-+		28	0	8	
	18	27	4	31	sp.1 imp.
		24	0	8	
	19	7	5	30	=+
31.16b=+		24	16	10	8 imp. IMPERATIEF TYPEN.
4.31a -+	20	29	0	8	
		27	0	8	
31.16b-+	21	24	16	10	7 imp.
5.2a -+		29	0	8	
	22	27	0	8	
31.16b-+		24	16	10	6 imp.
5.5a -+	23	29	0	8	
		27	0	8	
31.16b-+	24	24	16	10	5 imp.
5.8a -+		29	0	8	
	25	27	0	8	
31.16b-+		24	16	10	4 imp.
5.11a -+	26	29	0	8	
		27	0	8	
31.16b-+	27	24	16	10	3 imp.
5.14a -+		29	0	8	
	28	27	0	8	
31.16b-+		24	16	10	2 imp.
5.17a -+	29	29	0	8	
		27	0	8	
5.19a -+	30	24	16	10	1 imp.
		29	0	8	
	31	27	0	8	
		7	31	16	=+

30	28	"16n-11(16)"	Typcontrole	Kanaal 30, laatste 4.
	29	16m+11	Typcontrole	Kanaal 31, 1 ^{ste} helft.
	30	Werkruimte telling	Typcontrole	
	31	0 0 0	Constante + 10 voor	
		0 0 0	sluitletter X	
31	0	8 30 0	Voor correctie en voltooiing bij F ⁴ vrs	
		8 31 25		
	1	15 1 31	Voor strooisprong in F-complex;	
		12 0 0	Voor de serie-handinzet.	
	2	8 30 0	Voor correctie b-opdrachten	
		24 30 15	schuiven in goede positie.	
	3	24 17 0	Variabele opdracht bij "F+	
		24 28 30		
	4	4 31 28	" " " F-	
		11 31 28		
	5	10 31 13	" " " F+ .	
		20 31 20		
	6	10 31 13	" " " F- .	
		21 31 20		
	7	12 31 31	Opdracht voor F ⁴ ; constante	
		7 1 2	voor invullen correctieadressen.	
	8	8 30 0	Opdracht voor invullen	
		24 17 0	van correctieadressen.	
	9	12 0 0	Constante <u>Lees-schrijf</u>	
		7 1 5		
10	9	0 0	Constante <u>Lees-controleer</u>	
		7 1 4		
11	8	30 0	Voor correctie a-opdracht;	
		8 31 12	herstel van f _a uit "f _a -29"	
12	29	0 0	Constante voor het herstel van f _a .	
		0 0 0		
13	28	4 0 =50 000 000		
		1 15 21		
14	2 ^{de}	sprongenpaar G		
15	2 ^{de}	sprongenpaar B		

Kanaal 31; 2^{de} helft.

31	16	"Nalaatste wegbergopdracht bij "lees Biband" (zie 31.22).Koppelopdracht decimale typcyclus (=actueel sprongen- paar)
17	$10^{-m} \cdot 2^{28}$	Constante voor Typ B.
18	10^m	" " Typ B.
19	1 ^{ste} sprongenpaar G	
20	24 8 0 = 100 000 000	
	2 31 11	
21	1 ^{ste} sprongenpaar B	
22		"laatste" wegbergopdracht bij "lees Biband" (zie 31.16). Koppelopdracht van de typrountines
23	10^n	Constante voor Typ G
24		Werkruimte: onthouden van teken tijdens typen"
25		Werkruimte "staart" bij typ B. Indicatie "Schrijven of Controleren" bij bandlezen.
26	$10^m [10^{-m} \cdot 2^{28}]$	Constante voor Typ B
27	$2^{29} - (10^m + 1)$	" " Typ B
28		Werkruimte "digit voor punt" bij typen Koppelopdracht van de typcontrole
29		
30		Koppelopdracht van sub-subroutine; typ teken
31		Werkruimte "Rest" bij Typ G

HERSTEL VRIJE KANALEN.

Roffel Blank	<31 29>	<31 30>	<31 31>
X	(Wegbergopdracht Bi-		(Normale wegbergop-
0 0	band)		dracht)
3 2 15	12 31 31	(0 29 29	(12 31 30
0 0 0	4 31 30	0 0 0)	0 0 0)
31 30 12	↓	15 2 3	12 30 31
5 1 7		0 0 0	7 1 5
0 14			
0 0 0	12 31 0	15 0 7	(12 31 13
0 0 0	4 30 31	0 0 0	7 1 5)
0 30 8	↓		
25 31 8			
31 1 15	12 31 2	"	("
0 0 12	4 31 1		)
0 30 8	↓		
15 30 24			
0 17 24	12 31 4	"	("
30 28 24	4 31 3		)
28 31 4	↓		
28 31 11			
13 31 10	12 31 6	"	("
20 31 20	4 31 5		)
13 31 10	↓		
20 31 21			
31 31 12	12 31 8	"	("
2 1 7	4 31 7		)
0 30 8	↓		
0 17 24			
0 0 12	12 31 10	"	("
5 1 7	4 31 9		)
0 0 9	↓		
4 1 7			
0 30 8	12 31 12	"	("
12 31 8	4 31 11		)
0 0 29			
0 0 0			
F+5000 000X	15 1 21	24 17 0	12 31 13
	12 0 0	24 28 30	7 1 5 ↓
F 9 6	15 1 30	("	(12 31 14 ↓
	12 0 0	)	7 1 5) ↓
F+10000 0000X	15 1 21	24 17 0	12 31 20 ↓
	12 0 0	24 28 30	7 1 5 ↓
F 6	8 30 31 ^x)	12 31 30	(12 31 21
7 1 1 X	24 30 15	7 1 1	7 1 5)

De controle gaat beginnen.

^x) <31 29> = 12 00; 15 1 27 in het begin van F 6

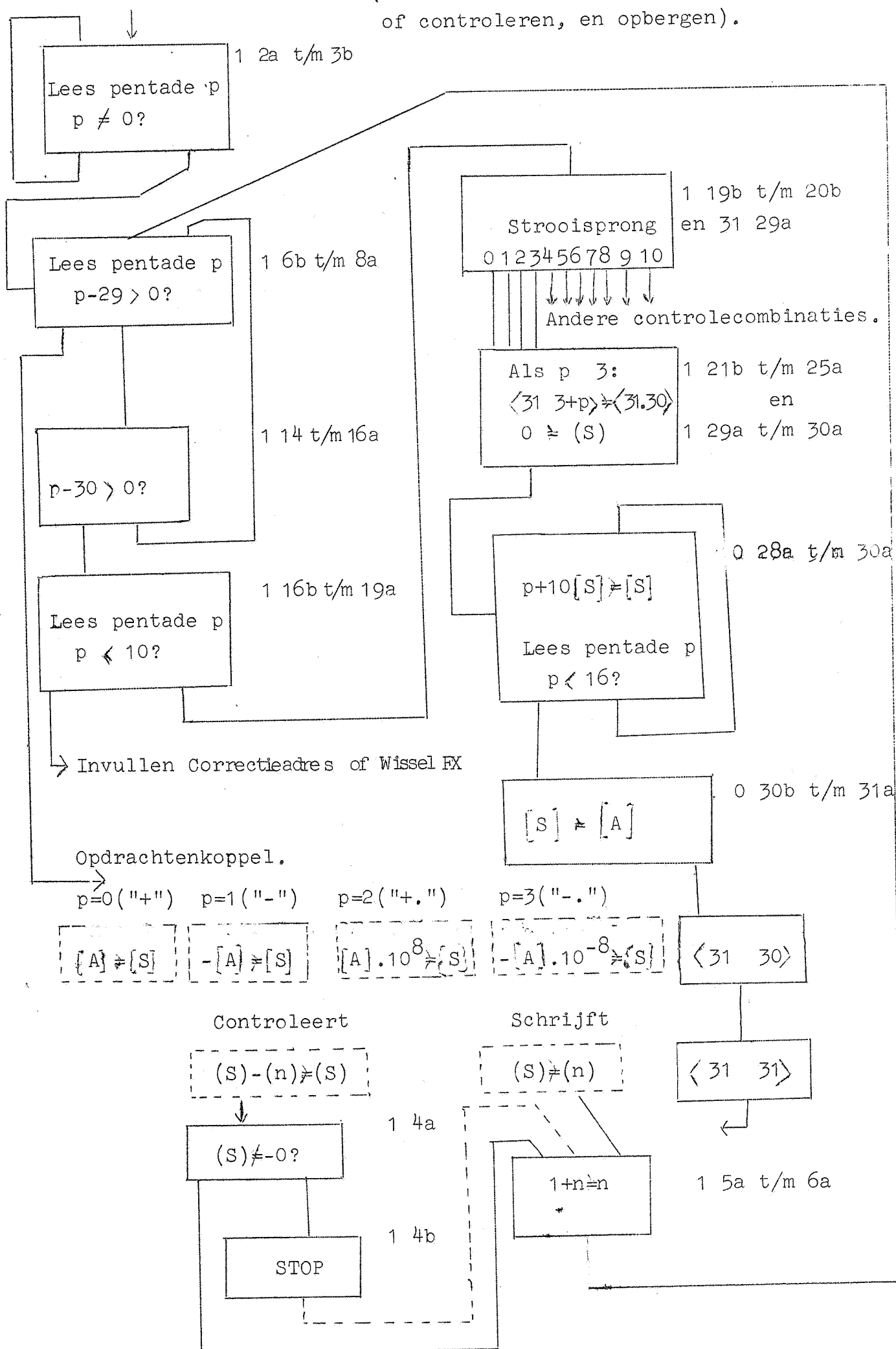
HERSTEL VRIJE KANALEN (VERVOLG).

	< 31 29> xx)	< 31 30>	< 31 31>
F 4 30 31 X			(9 30 31
F 10 0 14			7 1 4)
0 0 0	9 31 0	15 0 5	(9 31 13
0 0 0	1 30 31	0 0 0	7 1 4)
0 30 8	↓		
25 31 8			
31 1 15	9 31 2	"	("
0 0 12	1 31 1		)
0 30 8	↓		
15 30 24			
0 17 24	9 31 4	"	("
30 28 24	1 31 3		)
28 31 4	↓		
28 31 11			
13 31 10	9 31 6	"	("
20 31 20	1 31 5		)
13 31 10	↓		
20 31 21			
31 31 12	9 31 8	"	("
2 1 7	1 31 7		)
0 30 8	↓		
0 17 24			
0 0 12	9 31 10	"	("
5 1 7	1 31 9		)
0 0 9	↓		
4 1 7			
0 30 8	9 31 12	"	("
12 31 8	1 31 11		)
0 0 29			
0 0 0			
F+ 5000 0000 X	15 1 21	24 17 0	9 31 13
	12 0 0	24 28 30	7 1 4
F 9 6	15 1 30	("	(9 31 14
	12 0 0	)	7 1 4)
F+10000 0000 X	15 1 21	24 17 0	9 31 20
	12 0 0	24 28 30	7 1 4
FX			(9 31 21
Röffeltje blank			7 1 4)
(F 6			
23 00 X)			

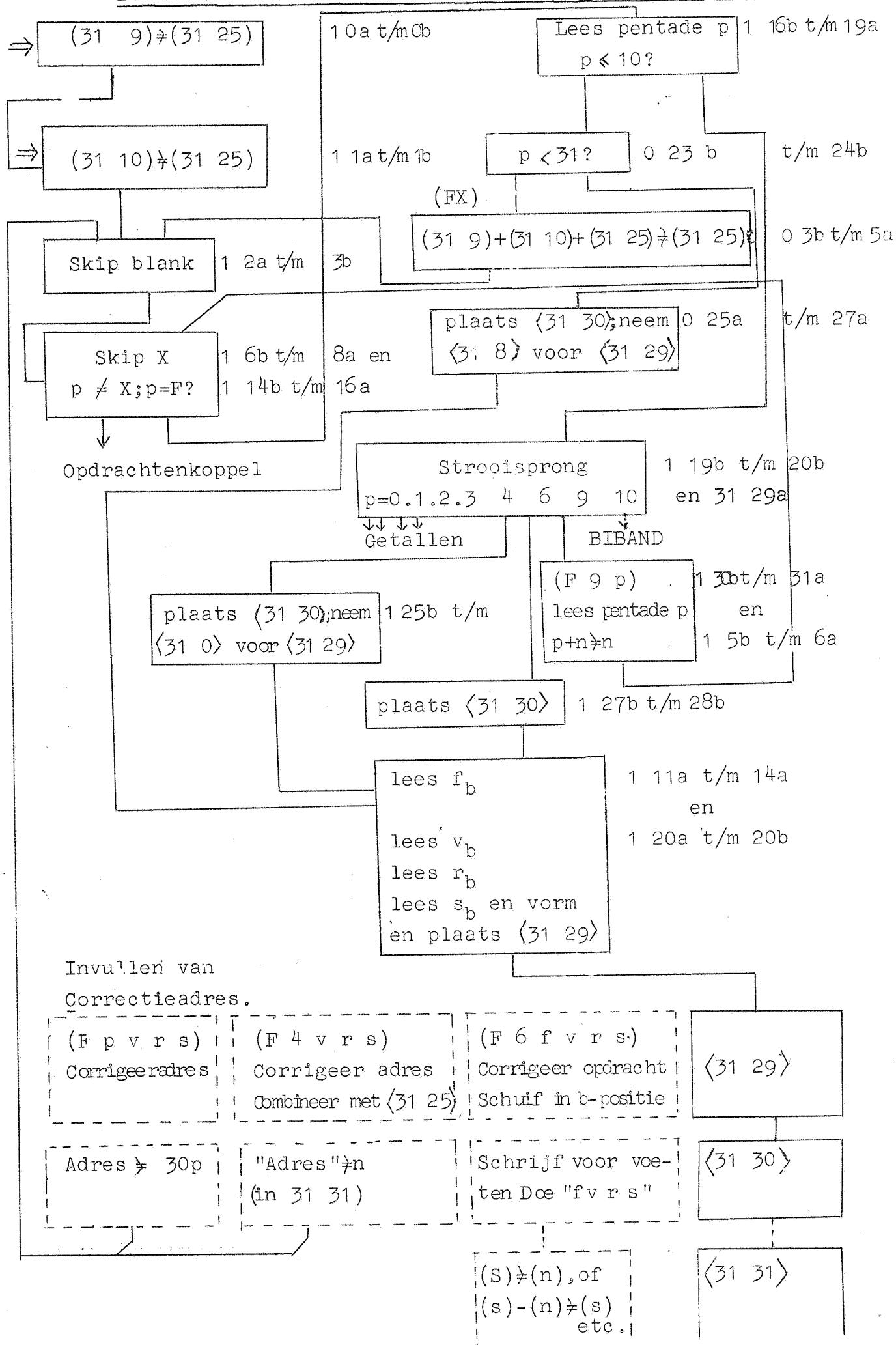
^{xx}) Bij F 4 is $\langle 31.29 \rangle = 12\ 00;15\ 1\ 25$, vervolgens $= 8\ 31\ 25;8\ 30\ 0$
 Bij F 4 is $\langle 31.29 \rangle = 12\ 00;15\ 1\ 31$, wordt dan als onder ^{xx}) inge-
 vuld.

Blokschema: lezen van getallen.

(inclusief Skip blanc, Skip X; schrijven of controleren, en opbergen).

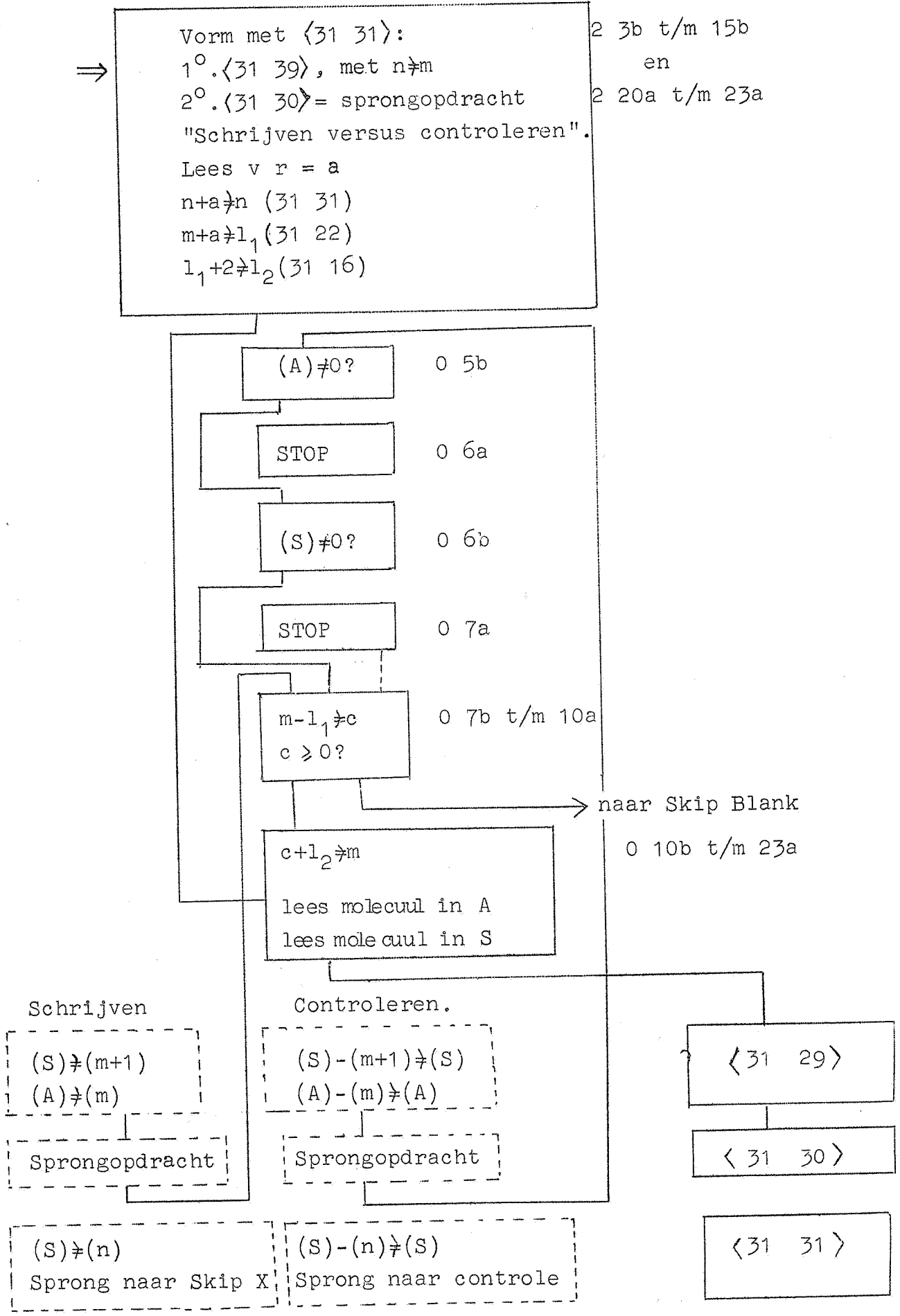


Blokschēma Invullen correctieadressen en Controlecombinaties.



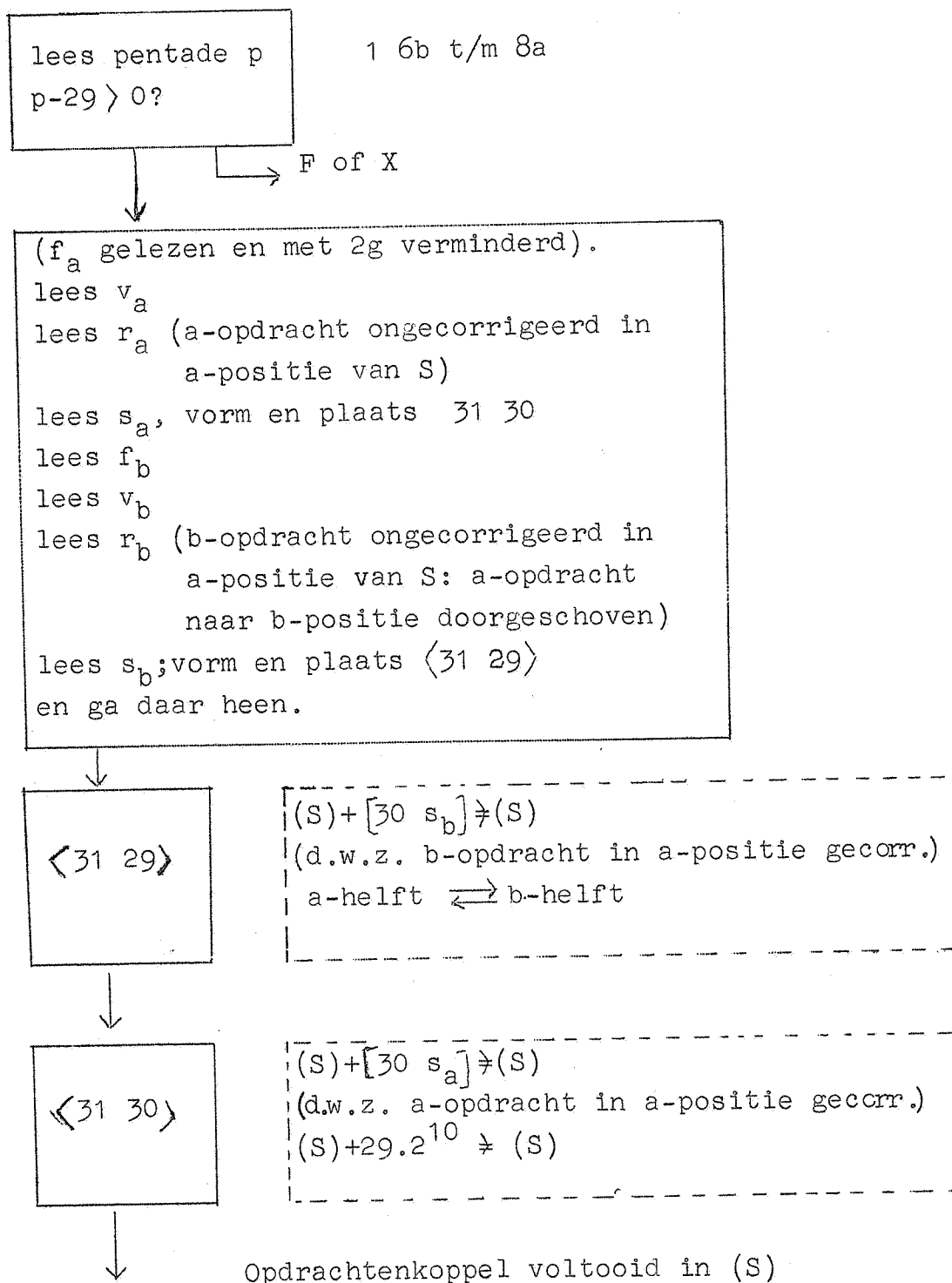
Blokschema "lees Biband" (na de strooisprong).

van strooisprong



Blokschema "lezen van opdrachtenkoppel".

(van Skip X tot variabele opdracht in 31 31)

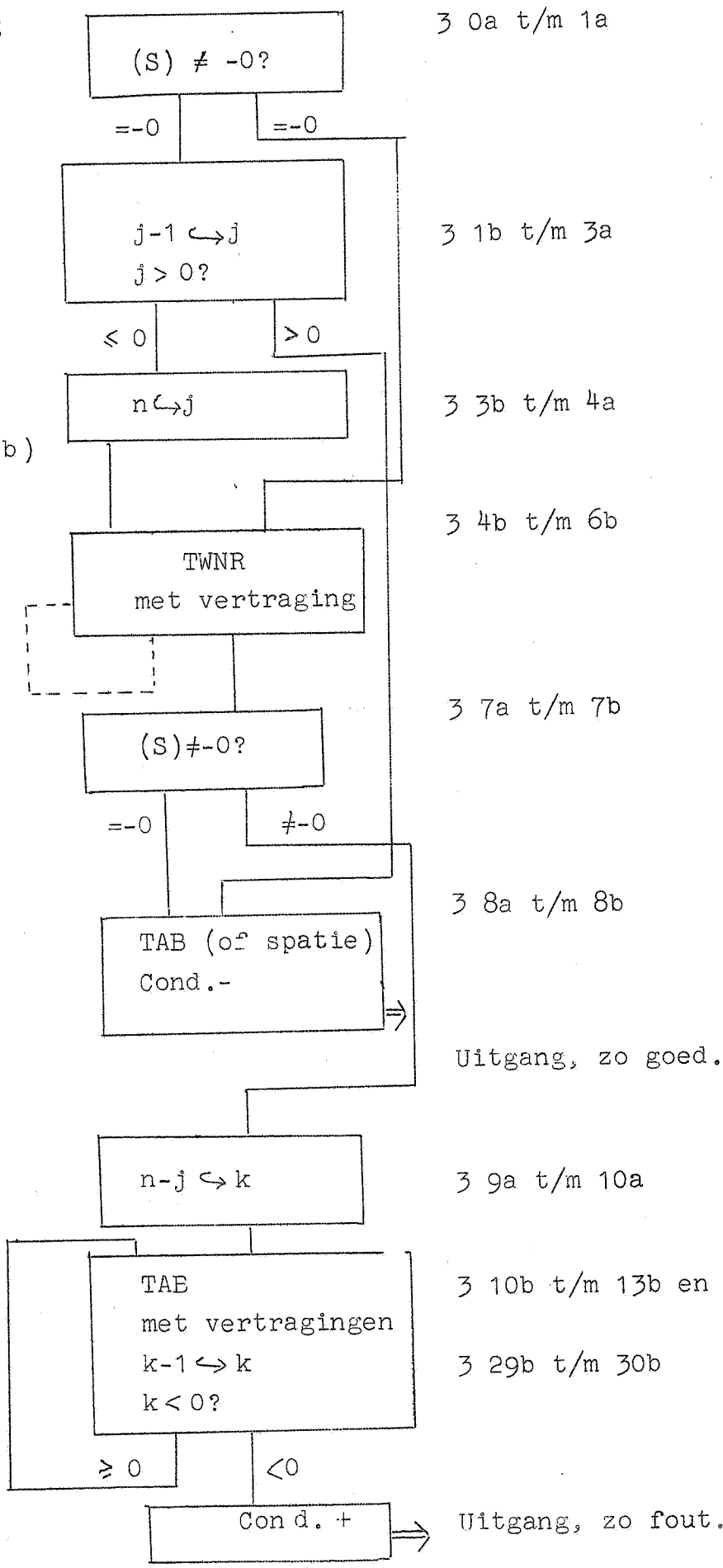


Blokschema van de typcontrole.

Controleingang

3 0a t/m 1a

Inloopsingang
(3 31a t/m 31b)



3 1b t/m 3a

3 3b t/m 4a

3 4b t/m 6b

3 7a t/m 7b

3 8a t/m 8b

3 9a t/m 10a

3 10b t/m 13b en

3 29b t/m 30b

Uitgang, zo fout.

Beschrijving: Serviceprogramma: pons Biband.

Het serviceprogramma: "pons Biband" beslaat in kanaal 30 de plaatsen 0 t/m 27, en is in staat, een binaire band te ponsen van de inhoud van een aaneensluitend, even aantal geheugenplaatsen. (bv. een met voorponssingen e.d. ingelezen programma, dat vele malen vaker ingelezen moet worden).

Begin en lengte worden in de getalschakelaars meegegeven:

nl: $\langle G \rangle = 0, v, r, 0, V, R.$

als V R het beginadres is, waar de betroffen inhoud staat (het "begin van de binaire band")

als v r het aantal vellen en regels is, dat de band lang is (r even).

Men start de machine op adres 30 0a; dan wordt geponst:

Roffel blank (69 stuks)

F 4 V R X

F 10 v r

(32v+r) binaire moleculen.

F X

F 4 V R X

F 10 v r

(32v+r) binaire moleculen.

F X

1 blank

F 6

23 0 0 X

5 blank

De band wordt dus tweemaal geponst, iedere keer gevolgd door de wissel FX. Aan het einde van de band wordt de stopcombinatie geponst. Als men de band wil incorporeren in een grotere band wil men deze stopcombinatie onderdrukken. Daartoe stopt men de machine op adres 30 22a, de opdracht, die de extra blank pons. Door op de toets "Repeat" van de automatische ponsmachine te drukken, wordt de laatste pentade herhaald: de band loopt er met een stuk blank uit.

Opm. 31 29 en 31 30 worden als werkruimtes gebruikt.

Het uitgeschreven programma, dat volgt op deze beschrijving, is genoteerd, als of het via het normale leesprogramma is ingebracht. Dit is gedaan, om de leesbaarheid van het programma te vergroten. In de praktijk wordt dit programma met een binaire band ter lengte van 28 plaatsen ingebracht.

Thans volgt ter instructie een puntsgewijze behandeling van het service-programma "pons Biband":

- 30 0a In A wordt 3 x 32 geplaatst; dit eindigt binaal op vijf
t/m nullen; de twee volgende ponsopdrachten ponsen dus blank;
- 30 2a de 3^{de} ponsopdracht ponst eveneens blank, maar verdubbelt
tevens de inhoud van A en test het teken: aanvankelijk
is dit steeds positief, en de besturing springt terug naar
de eerste ponsopdracht.
- 30 2b In 31 30 wordt de negatieve inhoud van A neergeschreven,
t/m n.l.: $d_{29} = d_{28} = 1$; d_{27} t/m $d_0 = 0$. (Aan (31 30) wordt
30 11b onthouden, of het stuk geheugen voor de eerste of de
tweede keer wordt uitgeponst). Vervolgens wordt 18 (bi-
naal 10010) in A geplaatst, en een 2 slagen schuivende
ponsopdracht ponst $F=11110$ (nl. de 0, de 1 en verder met
het laatste cijfer aangevuld); in A blijft aan de hoge
kant 4 (binaal 100) over, en de 4 wordt geponst. Inmiddels
is $(S)=(G)=0$, v, r, 0, V, R uit de getalschakelaars ge-
lezen (30 4a); door de schuifopdracht in 30 5a komt
0, V, R, in de hoge helft van A; in de lage helft van A
staat, wat voor deze schuifopdracht in de hoge helft
stond: maar dit zijn nullen, want door de ponsopdracht,
die 4 ponste, is juist weer 18 in A achtergelaten; dus
 $(A)=0$, V, R, 0, 0, 0, en $(S)=0$, 0, 0, 0, v, r. Door de
volgende schuifopdracht wordt $(A)=R$, 0, 0, 0, 0, V, waar-
na de V geponst kan worden, door de 25 plaatsen schuivende
ponsopdracht, die R op het laagste vijftal achterlaat.
Als R geponst is, staat 0, V, R weer in de b-positie, zo-
dat door een negatieve ponsopdracht de hoogste vijf nul-
len een $X = 31$ ponsen, waarbij 0, V, R weer op de a-plaats
staat. Na optelling van een constante (uit 30 27) wordt
door de opdracht op 30 8a in 30 12 het eerste koppel
neergeschreven, nl. $30.12 = 28, 21, 5, 2, V, R$. Analooq
aan F 4 wordt nu F 10 uit A geponst. De opdracht op
30 10a draait in S (waar nog steeds 0,0,0,0,v,r stond),
de v op de vijf laagste plaatsen, de v wordt geponst, en
de r wordt op de vijf laagste plaatsen gezet en geponst.
De opdracht in 30 11_b bergt $32v+r$ in 31 29, de werk-
ruimte voor de telling.
- 30 12a De opdracht 2 V R haalt het eerste te ponsen molecuul in
t/m A; dit wordt geponst door de 6 volgende b-opdrachten, en
30 18b wel van rechts naar links: elke ponsopdracht ponst de
laagste vijf cijfers, en draait het volgende vijftal "voor".

Onderdehand wordt in S de administratie uitgevoerd (dit uit tijdsoverwegingen: de ponsopdrachten zijn van een wacht (blokkade) voorzien). De a-opdrachten op 30 13 t/m 30 15 verhogen 30 12 met 1; de a-opdracht wordt dus nu 2 V R+1, en straks wordt het volgende woord aangehaald. De volgende drie a-opdrachten verrichten in S de telling, die de (aanvankelijke) v r met 1 verminderen, waarna, door het wegschrijven van het resultaat getest wordt, of de nul al bereikt is. Zo worden precies $32v+r$ woorden uitgeponst, de besturing verlaat de cyclus met $(S) = -0$.

- 30 19a De volgende drie opdrachten ponsen FX; dan wordt de test
t/m uitgevoerd, om uit te maken, of dit de eerste keer of de
30 21b tweede keer is geweest: (31 30) komt verdubbeld in A te
staan, de sprong wordt gehoorzaamd, als het nieuwe teken-
cijfer = 1 is; dit is het geval, want nu is $d_{29}=d_0=1$, en
 d_{28} t/m $d_1=0$. Na de sprong wordt door de opdracht in 30 2b
deze nieuwe inhoud in 31 30 opgeborgen. De Biband wordt
weer geponst (we zijn, na de roffel blank tape in het pro-
gramma gesprongen: deze roffel verschijnt dus uitsluitend
aan het begin van de tape). Nu echter vormt de verdubbe-
ling van (31 30) (A)=3, en de besturing gehoorzaamt de
sprong niet. Weer is nog steeds $(S) = -0$.
- 30 22a Door de opdracht op 30 22a wordt de aangekondigde extra
t/m blank geponst (hiermede gebruik makend van $(S) = -0$).
- 30 26b (A) =3 wordt benut, om in (31 29), de werkruimte van de
telling iets voldoende groot positiefs te zetten: we wil-
len nl. dat de sprongopdracht in 30 18b nog twee maal
gehoorzaamd moet worden: 3 is dus juist groot genoeg.
Dan wordt in S de koppelopdracht gelezen, d.w.z. de sprong
naar 30 24a; (30 26) wordt in A opgenomen ter ponsing
van F 6 23 0 0 X, de F wordt nog door dit stuk programma
geponst, voorbij het aanhalen van (30 12) springt de be-
sturing weer in de pons cyclus, en het ponsen van de slot-
combinatie wordt voltooid. In 30 12 wordt de in S meege-
geven koppelopdracht met 1 verhoogd neergeschreven. De
telling vormt (31 29)=2 > 0, dus de conditionele sprong in
30 18b wordt gehoorzaamd, en de besturing ontmoet in 30 12a
de sprong naar 30 25a; (A)=+0 wordt meegegeven, er worden
vijf pentades blank geponst: weer wordt de sprongopdracht
in 30 12a met 1 verhoogd, de telling vormt $2-1=1 > 0$, de
sprong in 30 18 b wordt weer gehoorzaamd en de besturing

komt in 30 26a; hier gebeurt niets (cond.+), op de
opdracht 31 0 0 stopt de machine.

Serviceprogramma: pons · Biband.

[illegible]

Utility breuk lezen. 8 cijfers.

		F4	30	0	X									
	30	0	2	31	31	X		30	16	24	16	10	X	
			0	30	25	X				26	21	30	X	
		1	4	31	31	X			17	24	16	10	X	
13a.3'a	--		26	21	30	X				26	21	30	X	
		2	29	0	16	X			18	24	16	10	X	
			14	30	12	X	--			26	21	30	X	
		3	15	30	1	X	==		19	24	16	10	X	
5'a	==		26	21	30	X				26	21	30	X	
		4	24	17	0	X			20	24	16	10	X	
			29	20	5	X				26	21	30	X	
		5	14	30	3	X	--		21	24	16	10	X	
			29	0	16	X				26	21	30	X	
		6	14	30	13	X	--		22	24	16	10	X	
			2	31	31	X				24	28	30	X	
		7	1	30	25	X			23	24	17	0	X	
			4	31	31	X				10	31	13	X	
		8	7	1	2	X	==klaar		24	15	31	30	X	==
31.31b	==		28	8	16	X				31	0	0	X	
		9	22	0	0	X			25	0	0	0	X	
31.31b	--		2	31	31	X				8	29	4	X	
		10	24	17	0	X			26	F	9	1		
			24	4	1	X								
		11	4	31	31	X			27	24	17	0	X	
			26	21	30	X				18	31	20	X	
		12	24	17	0	X								
2'b	--		29	20	5	X								
		13	14	30	3	X	--							
			0	30	27	X								
		14	24	17	0	X								
			26	31	30	X								
		15	4	31	30	X								
			26	21	30	X								

In kanaal 31 verschijnt:

31 30

30 24a ==

31 31

(24 17 0 X)

(20/21 31 20 X)

(9/12 V' R')

15 30 8/9) ==

Opm. Om der wille van de leesbaarheid is wederom dit programma in (absolute) normale ponsing weergegeven. In de praktijk wordt dit programma met behulp van een binair bandje ingebracht.

De inloopscombinatie luidt F 6 7 30 0 X tot aan de eerste afwijkende pentade skipt dit leesprogrammaatje blank tape. Breuken worden geponst: +. (of -.) gevolgd door de 8 cijfers. Aan het begin van elke breuk wordt X geskript (Erase). X 0 is de uitloopcombinatie; de wegbergopdracht van het hoofd-invoerprogramma staat dan ingesteld op het volgende adres, met behoud van de controle-, dan wel schrijfstand. In beide standen is dit programma bruikbaar. Na afloop komt de besturing bij Skip blank tape in het hoofd- invoerprogramma terug. Omdat de X, die anders het einde van het decimaal gedeelte aangeeft, weggelaten mag worden, moeten exact 8 cijfers geponst worden. Dit programma is voor breuken ruim 2x zo snel als in het hoofd-invoerprogramma.

De tijdsduur van het invoerprogramma.

	Schrijven	Controleren
Skip pentade blank tape	40 msec.	40 msec.
Skip pentade erase	60 msec.	60 msec.
Opdrachtenkoppel	390 msec. (=154 koppels per minuut =12.5 sec.per kanaal)	410 msec. (=146 koppels per minuut =13.1 sec.per kanaal)
Geheel getal (k cijfers)	(365+60k)msec.	(385+60k) msec.
Breuk (k cijfers)	(465+60k)msec.	(485+60k) msec.
F 9 ("sla over")	180 msec.	180 msec.
Biband:Aanloop en afloop	460 msec.	460 msec.
Biband:	205 msec. (=585 woorden per minuut =3.28 sec.per kanaal)	225 msec. (=533 woorden per minuut = 3.6 sec. per kanaal).
Invullen correctieadres	380 msec.	380 msec.
Wissel F X	260 msec.	260 msec.
Breuk, Utility, 8 cijfers	425 msec. (=141 breuken per minuut = 13.6 sec.per kanaal)	425msec. (=141 breuken per minuut = 13.6 sec. per kanaal)

ERRATUM:

In MR 17: HANDBOEK VOOR DE PROGRAMMEUR, FERTA, DEEL I
De 8^{ste} t/m 10^{de} regel van onderen op pg.33 moeten luiden:
de omwenteling 20 msec.
de getaltijd 2.5 msec.

De trommel hoort te draaien met een snelheid van 50 omwentelingen