

CR 10.2

Cursus constructie X1,2.

B.J.Loopstra en C.S.Scholten.

1957.

Bij de hier gebruikte kernen bevat een rij 64 woorden. De 64 gevlochten lijnen zouden alle woorden van een kast kunnen doorlopen. Dat geeft echter zeer lange draden, met alle bezwaren daarvan. Daarom wordt een kast in twee helften verdeeld en het aantal onderste versterkers voor verticale selectie is verdubbeld. De onderscheiding tussen de 8 minst significante en de 8 meest significante versterkers wordt verkregen door de versterkers DDVTo' en DDVT1', die weer worden onderscheiden door de signalen DSG11 en DSG11'. Deze beide signalen worden gemaakt door omkering van signaal SG11', respectievelijk van signaal DSG11.

Een horizontale drijflijn loopt door een rij kernen van de eerste helft en daarna door de overeenkomstige rij kernen van de tweede helft. Voor de 64 rijen zijn dus 32 lijnen nodig. Wanneer een woord uit het dood geheugen moet worden geselecteerd, wordt stroom gestuurd door de horizontale drijflijn van de betreffende rij kernen. Geen van deze kernen zal nu nog kunnen omgaan, daar het veld, dat wordt opgewekt slechts $\frac{1}{2}H_{\max}$ is. De stroom door de gevlochten drijflijn veroorzaakt eveneens $\frac{1}{2}H_{\max}$. De kernen, die beide stromen omvatten, zullen een puls op de leeslijn veroorzaken. Maar ook de kernen, die niet omgaan, geven op de betreffende leeslijn een klein pulsje ten gevolge van het halve veld. De gevlochten drijflijn loopt ook door de overige 31 rijen kernen en al deze kernen (voor zover de draad erdoor is gevlochten) zullen op de bijbehorende leeslijn een stoor-pulsje geven. De gevlochten drijflijn echter loopt afwisselend van links naar rechts en van rechts naar links door de rijen kernen. Zodoende is de stoorpuls van een kern tegengesteld aan die van de overeenkomstige kern op de rij daaronder. Beide stoorpulsjes heffen dan de invloed van elkaar op. Maar het is niet zeker, dat de drijflijn ook inderdaad door het overeenkomstige kerntje gaat. Daarom worden bij de 28 kernen van een woord 28 zg. dummy-kernen gevoegd; gaat de draad niet

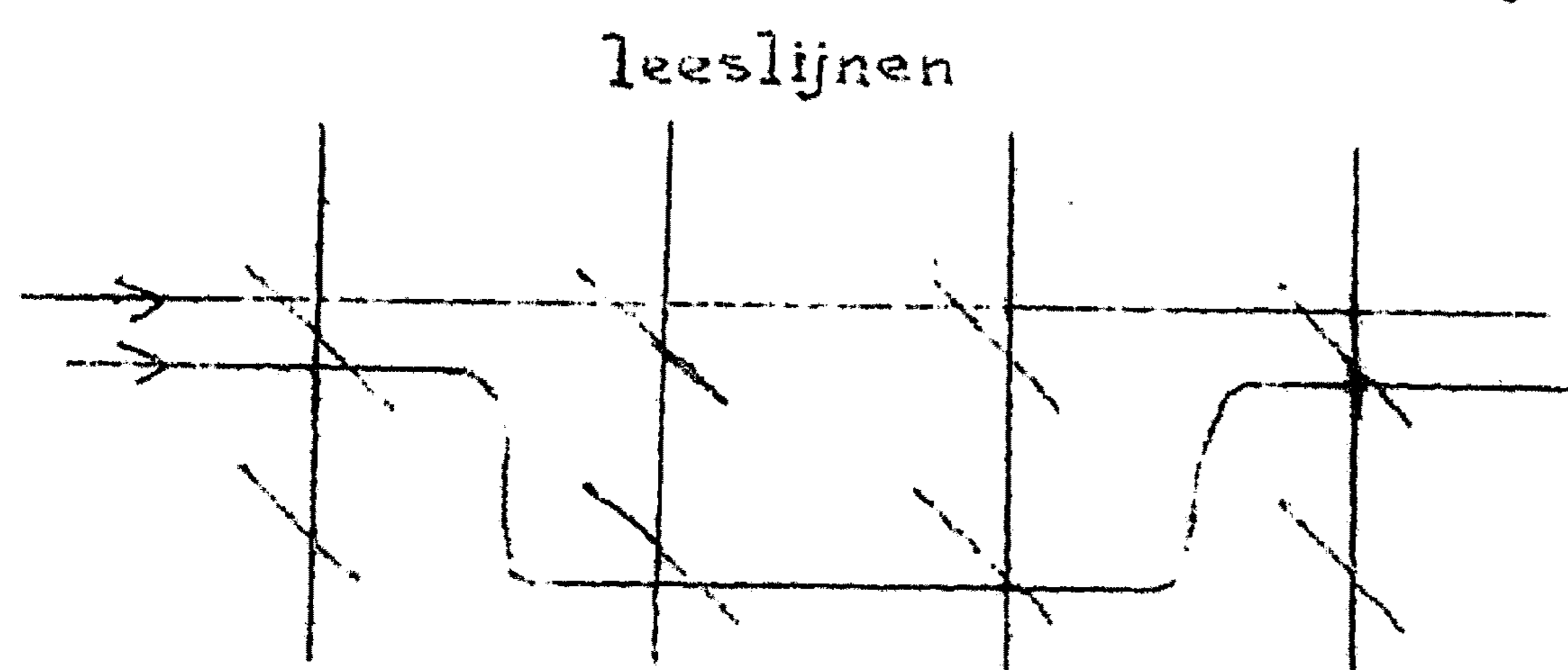


fig. 29

door de eigenlijke kern, dan wordt deze door de dummy-kern gevoerd (fig. 29), terwijl de leeslijn door beide kernen gaat. Ten opzichte van de leeslijn geeft nu dus iedere rij kernen een stoorpulsje, hetzij van

de cijfer-kern, hetzij van de dummy-kern. Nu is het wel zeker, dat alle stoorpulsjes elkaar opheffen.

Er zijn 32 rijen kernen en daarvan is er één geselecteerd. Van die rij levert een bepaalde kern of een signaalpuls tengevolge van de horizontale en verticale drijfstroom, of een stoorpulsje tengevolge van de horizontale drijfstroom, maar dan geeft de dummy-kern een even groot stoorpulsje in dezelfde richting. In het laatste geval zullen op die betreffende leeslijn 30 stoorpulsjes elkaar nivelleren. Resteren de kernen van de geselecteerde rij en van één niet geselecteerde rij. De niet geselecteerde kern geeft één stoorpulsje tegengesteld gericht aan de pulsjes van de geselecteerde kern met de bijbehorende dummy-kern, zodat op de leeslijn altijd één stoorpulsje resteert. Het zou teveel kosten ook dit op te heffen. In het eerste geval resteert eveneens één stoorpulsje, nl. van de niet geselecteerde kern, dat echter geen invloed heeft ten opzichte van de grote puls, die ontstaat door het omgaan van de geselecteerde kern. Bovendien worden de uitgangssignalen nog weer gepoort en gedrempeld, zodat de resterende stoorpuls binnen aanvaardbare grenzen blijft. De stroom door de gevlochten drijflijn wordt gestuurd met behulp van signaal 'DDL14' en de stroom door de horizontale drijflijn met behulp van signaal 'DDL10'. De laatste doet de kernen dus werkelijk omgaan. Een dummy-kern gaat nooit om en behoeft dus ook nooit te worden omgezet door de "terugzetlijnen". De niet geselecteerde helft van de kast, bestaande uit 32 rijen kernen wordt ook doorlopen door dezelfde gevlochten drijflijn. Hier ontstaat echter een even aantal stoorpulsjes, die elkaar dus weer nivelleren.

Een horizontale drijflijn doorloopt $2 \times 28 = 56$ kernen, een verticale (gevlochten) drijflijn $32 \times 28 = 896$ kernen en de terugzetlijnen $32 \times 2 \times 28 = 1792$ kernen. Het gehele dode geheugen is ondergebracht in kast 0. Deze kast heeft slechts één serie leesversterkers. De leeslijnen moeten dus zowel door de kernen van het dood geheugen als door die van dat deel van het levend geheugen lopen, dat in deze kast is ondergebracht. Een leeslijn doorloopt dus 4096 kernen van het levend geheugen (zie blz. 91) en $32 \times 2 \times 2 \times 4 = 1024$ kernen in het dood

geheugen, de dummy-kernen daaronder begrepen. Dat geeft dus een totaal van 5120 kernen per leeslijn.

Het dood geheugen en de cijferschakelaars

Bij de behandeling van de handbediening (blz. 45 - 54) werd reeds gesproken over de schakelaars, die bepaalde adressen in het dood geheugen beïnvloeden. Adres 0 is daarbij vrij, adres 1 werkt samen met de beginadressschakelaars (BS), adres 2 met de stopadressschakelaars (SS) en adres 3 met de getalschakelaars (GS). In fig. 30 is de schakelwijze van de kernen getekend. Afhankelijk van de stand van de schakelaars zal de stroom door of langs de kern lopen. Met de schakelaars is het

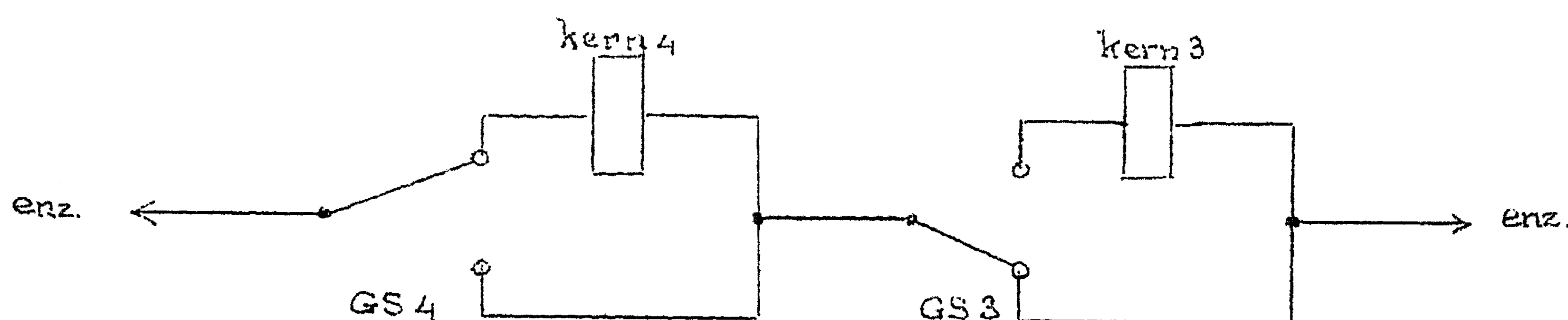
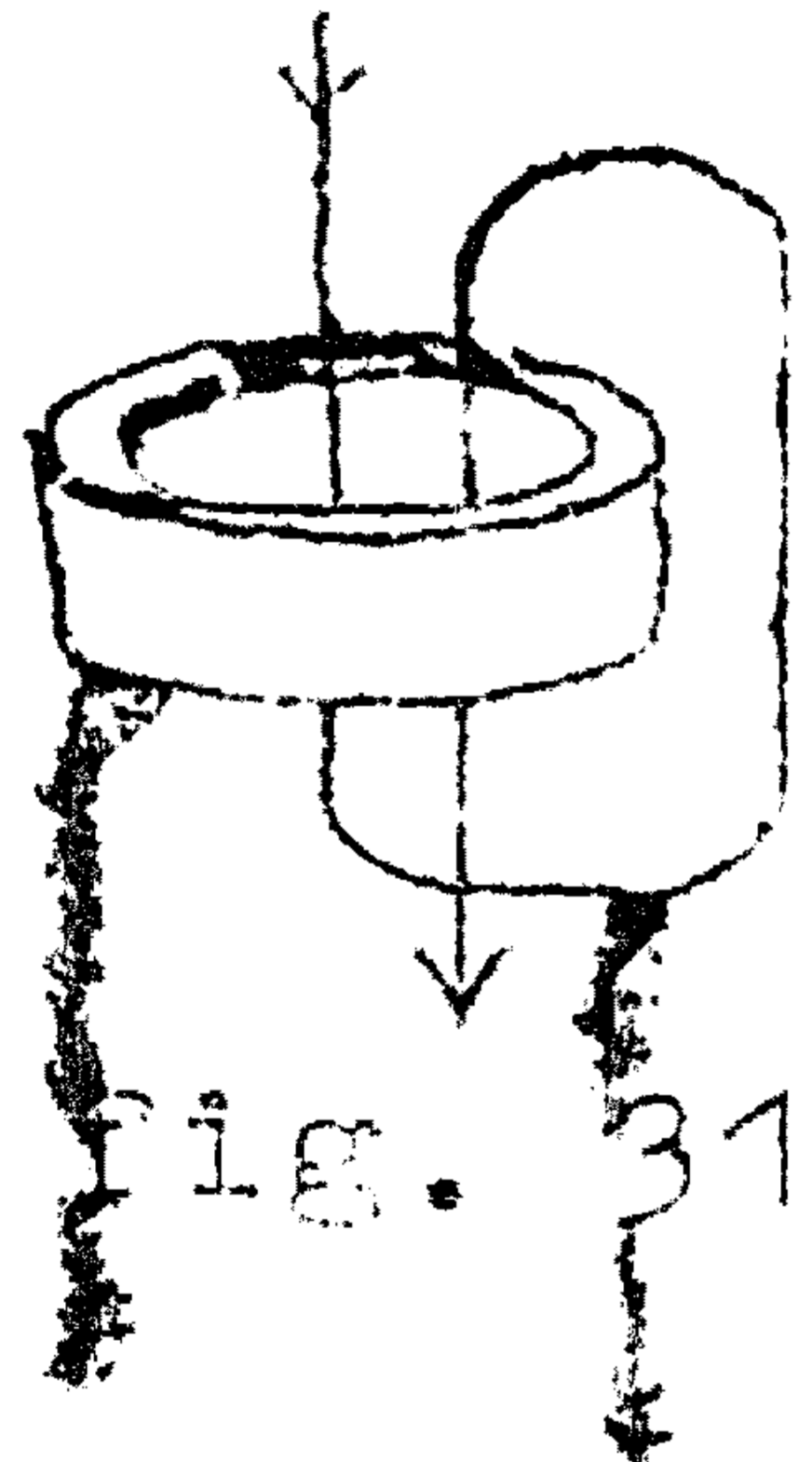


fig. 30

dus mogelijk de "draad" door een kern te "vlechten" of niet en op deze wijze een woord te vormen. Daartoe is een rij van 27 kernen nodig. De getalschakelaars kunnen alle 27 kernen beïnvloeden, terwijl de beginadres- en stopadressschakelaars de 15 minst significante kernen kunnen beïnvloeden. In de beide laatste gevallen leveren de 12 meest significante kernen een nul.

Indien geen bijzondere voorziening was getroffen, zouden vier van de gevlochten drijflijnen, via de schakelaars moeten lopen. De schakelaars bevinden zich echter in de basismachine, zodat zo'n drijflijn vele malen tussen de geheugenkast en de basismachine door de verbindingkabel heen en weer moet lopen. Daarom wordt voor de eerste vier adressen een aparte rij kernen gebezigd met speciale drijflijnen en speciale versterkers.

Die vier drijflijnen, die via de schakelaars hadden moeten gaan, zijn nu direct naar de betreffende versterker (oDDVBo') gevoerd en de eerste rij kernen omvat dus slechts 60 adressen. De drijflijn door de begin-, zowel als die door de stopadres-schakelaars is in twee stukken verdeeld en wel zodanig, dat een gedeelte door de 7 minst significante kernen en schakelaars loopt en een gedeelte door de 8 meest significante. Zo'n gedeelte van een drijflijn is verbonden aan de collector van een eindtransistor, terwijl het andere einde via een weerstand met de -16V is verbonden. De maximale stroom door een drijflijn, die ca 200mA is, in verband met de gebruikte versterker en de waarde van de weerstand, is dus niet in staat een kern te doen omgaan. Aangezien er door deze rij kernen geen drijflijn voor horizontale selectie loopt, moet de kern worden omgezet met deze "verticale" drijflijn. Deze laatste wordt daarom tweemaal



door de kern gevoerd (zie fig. 31), zodat toch i_{max} wordt omvat.

De twee versterkers voor de twee (losse) gedeelten van één drijflijn moeten tegelijkertijd stroom voeren, als die betreffende drijflijn is geselecteerd. De ingangsniveaus van de beide verster-

kers worden met elkaar doorverbonden, zoals in fig. 32 is aan-

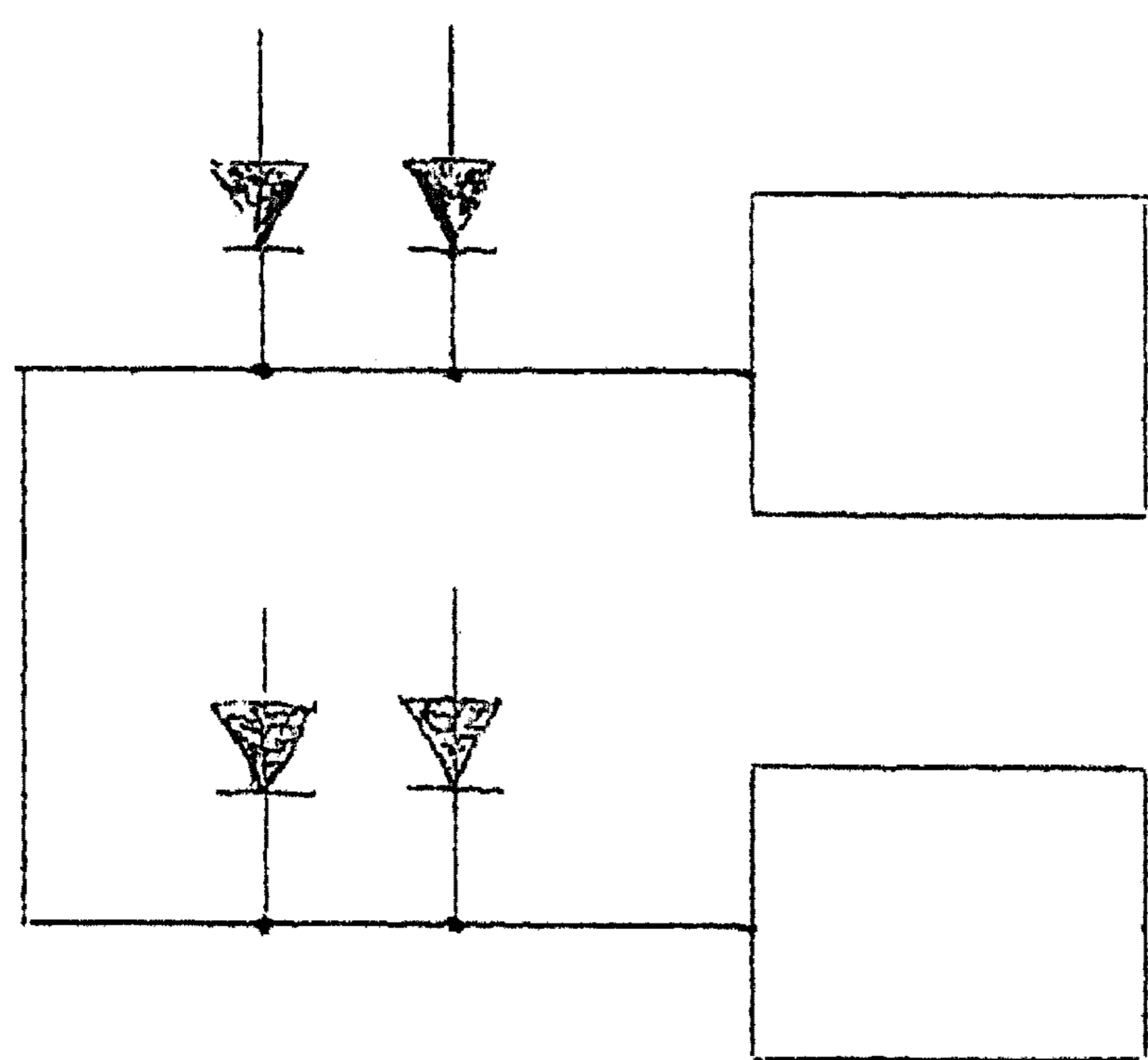


fig. 32

gegeven. De vier ingangen beïnvloeden nu elkaar en het verdere gedeelte van de beide versterkers (V_{o2}).

De eindtransistoren van beide versterkers gaan geleiden, wanneer de signalen aangesloten op de vier dioden laag zijn. Elke versterker drijft daarbij dus stroom door een gedeelte van "dezelfde" drijflijn.

Bij de getalschakelaars is de drijflijn in drie stukken verdeeld, elk gedreven door een versterker (V_{o2} en $2 \times V_{o1}$), omdat hierbij 27 schakelaars en 27 kernen behoren. Elk stukje van deze drijflijn loopt door 9 schakelaars en 9 kernen.

Voor elk adres zijn vier ingangen beschikbaar. Op de eerste ingang wordt signaal DGS' aangesloten, op de tweede het tijdsignaal DDL10', terwijl op de laatste twee de signalen SGo en SG1 zijn aangesloten, met de polariteit, zoals die is vereist om dat adres te selecteren. Signaal DGS' is laag, als de signalen SG2 t/m SG14 laag zijn (de cijfers SG2 t/m SG14 zijn nul) door aansluiting van genoemde signalen op het niveau van plaatje DGS'.

Bij de woorden gevormd door de schakelaars wordt geen gelijkheidscontrolecijfer gevoegd. Om nu te voorkomen, dat de machine zou stoppen tengevolge van deze controle, wordt niveau 3 van plaatje FOUT laag gemaakt met signaal DGS', zodat de machine niet kan stoppen tengevolge van signaal FOUT bij selectie van de schakelaars.

Twee versterkers, die elk een stroom $\frac{1}{2}i_{\max}$ door een "terugzetlijn" sturen, worden gedreven met behulp van signaal DS10, dat wordt omgekeerd en daarna signaal DDS10' heet. Beide terugzetlijnen lopen door alle kernen van een dood-geheugenkast (ook die van de schakelaars), zodat dan alle kernen weer een "een bevatten". Hieruit blijkt, dat bij het dood geheugen het geheugenelement niet in de kernen zit. De kernen bevatten geen informatie. De informatie blijkt uit de wijze van bedraden.

Vermenigvuldiging (algemeen)

Wanneer getallen uit het tientallig stelsel vermenigvuldigd moeten worden handelt men als volgt:

Men vermenigvuldigt het vermenigvuldigtal met het minst significante cijfer van de vermenigvuldiger en schrijft het verkregen product op. Daarna vermenigvuldigt men het vermenigvuldigtal met het tweede cijfer van de vermenigvuldiger. Vervolgens schrijft men het nu verkregen product op onder het voorgaande en wel aan de minst significante zijde aangevuld met een nul, want het cijfer, waarmee is vermenigvuldigd, vertegenwoordigde de tientallen. De (aanvullings)nul van het laatste product komt onder het minst significante cijfer van het voorgaande product.

Het daaropvolgende product wordt aangevuld met twee nullen en zo voort tot de vermenigvuldiging is uitgevoerd met alle cijfers van de vermenigvuldiger. Tenslotte worden de aldus neergeschreven producten bij elkaar opgeteld.

$$\begin{array}{r}
 14 \\
 341 \\
 \hline
 14 \\
 560 \\
 4200 \\
 \hline
 4774
 \end{array}$$

Eigenlijk worden de achtereenvolgens verkregen producten steeds één "plaats" naar links geschoven.

Een uitgebreidere methode (op papier) is: na berekening van het tweede product dit neer te schrijven één plaats naar links verschoven en aangevuld met een nul onder het eerste product en daarna deze beide getallen op te tellen. Het 3e product wordt weer één plaats naar links geschoven en aangevuld met twee nullen onder de som van de eerste twee geschreven en vervolgens weer opgeteld enz. enz. Men kan dit natuurlijk ook zo beschouwen alsof de som van de voorgaande producten een plaats naar rechts geschoven boven het laatst berekende product wordt geschreven.

$$\begin{array}{r}
 341 \times 14 \\
 \begin{array}{r}
 1 \times 14 \\
 \hline
 \text{schuif} \quad 14 \\
 4 \times 560 \\
 \hline
 \text{schuif} \quad 574 \\
 3 \times 4200 \\
 \hline
 4774
 \end{array}
 \end{array}$$

We merken hierbij op, dat na de eerste bewerking het minst significante cijfer gedurende de verdere bewerking gegarandeerd niet meer verandert, want er worden alleen nullen bij opgeteld. We zouden dat dus ook vast kunnen neerschrijven (opbergen) op een bepaalde plaats en daarna de "rest" van het getal recht boven het daaropvolgende product kunnen schrijven. In wezen komen de beschreven bewerkingen op hetzelfde neer en geven in elk geval hetzelfde resultaat. De laatste methode echter wordt gebruikt in de X-1.

Aangezien in het tweetalig stelsel alleen de cijfers 0 en 1 worden gebruikt, heeft men bij een vermenigvuldiging alleen na te gaan aan de hand van de vermenigvuldiger of het vermenigvuldigtal moet worden opgeteld bij het daarvoor verkregen resultaat (maal 1) of niet (maal 0). In het laatste geval wordt bij het voorgaande (tussentijdse) product het getal nul opgeteld en het verkregen resultaat naar rechts geschoven. Immers nul maal het vermenigvuldigtal is weer nul. Vervolgens wordt het minst significante cijfer van het "resultaat" opgeborgen. Bij deze bewerking wordt het vermenigvuldigtal in het M-register gezet en de vermenigvuldiger staat in het S-register. Het resultaat komt in de registers A en S te staan, waarbij register A het meest significante gedeelte daarvan bevat en register S het minst significante gedeelte. De tekencijfers van beide registers moeten dan natuurlijk hetzelfde zijn. Elk register heeft een capaciteit van 26 cijfers + een tekencijfer, zodat het eindresultaat 52 cijfers omvat + een (eigenlijk 2 dezelfde) tekencijfer. Deze capaciteit kan niet worden overschreden. In het volgende voorbeeld zullen we elk register een capaciteit van drie cijfers toekennen, hetgeen in principe geen verschil maakt, maar beter is te overzien (fig. 33)

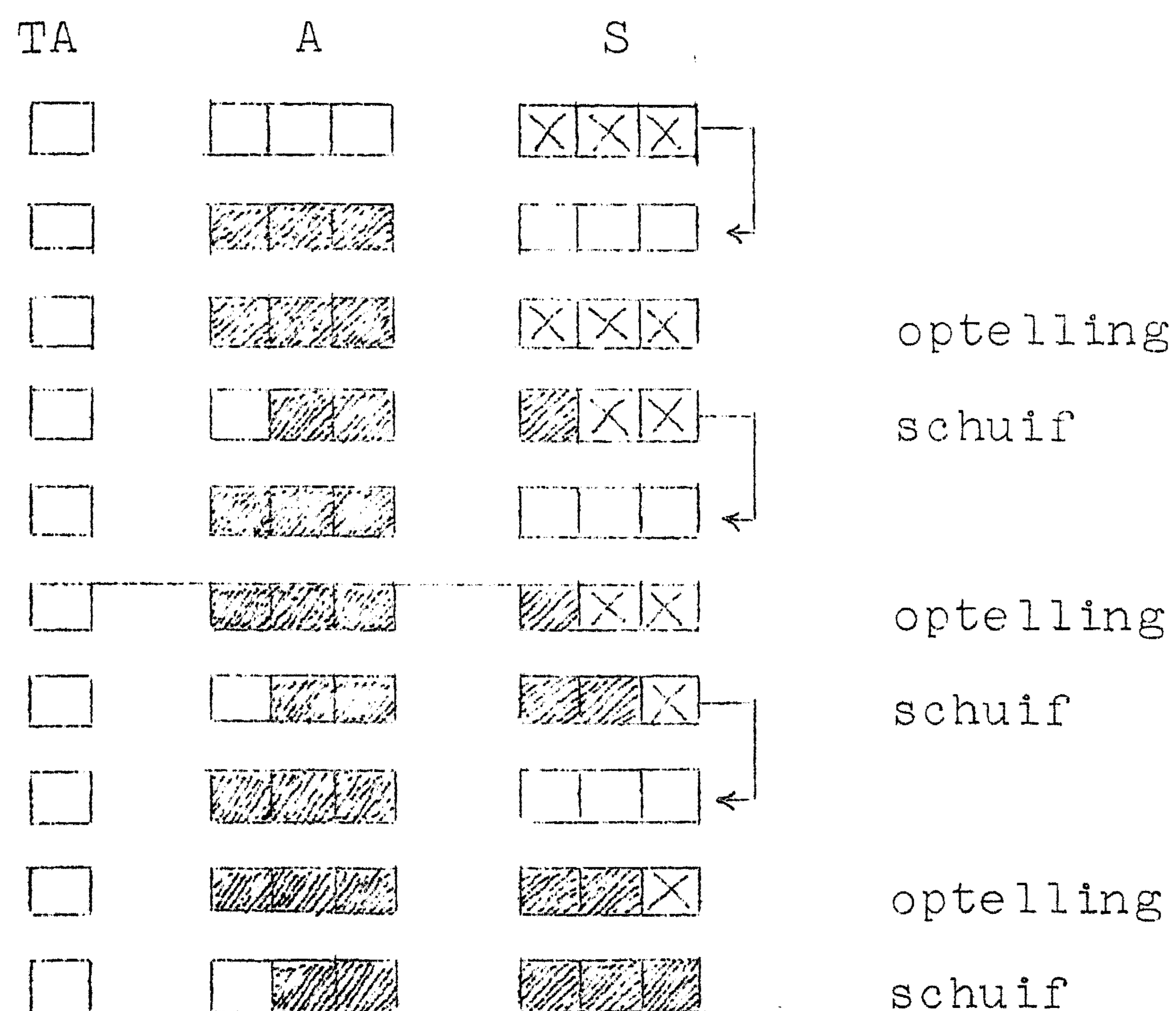


fig. 33

We veronderstellen, dat het A-register schoon is. Is het minst significante cijfer van de vermenigvuldiger een één, dan moet er een optelling plaatsvinden. Daar het A-register echter nog schoon is, neemt dit register de inhoud van het M-register over via het arithmetisch orgaan en wel één plaats naar rechts verschoven. Zo'n optelling tezamen met de "schuif" wordt een slag genoemd. Het resultaat van de optelling verschijnt daarbij in de somvormer en vervolgens neemt het U-register dit woord verschoven over. Daarna pas neemt het A-register op zijn beurt het woord uit het U-register over. Aan het eind van de eerste slag wordt het minst significante cijfer van het verkregen resultaat op de meest significante plaats in het S-register opgeborgen, waarbij we de plaats van het tekencijfer overslaan. De tekencijfers worden apart beschouwd. De rest van het resultaat komt in het A-register te staan. De meest significante plaats in dit register bevat dan een nul (afgezien van de plaats van het teken en gesteld, dat het vermenigvuldigtal positief is), want er kan nog geen overdracht plaatsgevonden hebben. In fig.33 zijn de registers getekend en de vakjes die een cijfer bevatten tengevolge van deze bewerking zijn gearceerd. De vermenigvuldiger in S is aangegeven door "kruisjes", terwijl het vermenigvuldigtal onder het S-register aangevuld gedacht kan worden met tekencijfers. In het besproken voorbeeld zijn dat dus nullen. (blank gelaten) Stel dat het volgende cijfer van de vermenigvuldiger weer een één is, dan zal bij de volgende slag het vermenigvuldigtal worden geteld bij de inhoud van het A-register, zoals die op dat moment is, d.w.z. de meest significante plaats daarvan bevat een tekencijfer, terwijl het minst significante cijfer van het voorgaande resultaat reeds definitief is opgeborgen en dus niet meer kan worden veranderd. Nu is het mogelijk, dat er een overdracht ontstaat, die in het tekencijfer zou lopen (bij het woord in de somvormer!). Door de schuif echter wordt er plaats gemaakt in het A-register voor dat cijfer. Het teken van het A-register wordt daarbij echter niet gewijzigd, aangezien dat speciaal wordt gemaakt.

Bij beschouwing van fig. 33 blijkt dat van het S-register steeds het minst significante cijfer "naar buiten wordt geschoven" en daardoor verloren gaat. Dit register bevat in het begin de vermenigvuldiger en de cijfers daarvan geven alleen aan of een optelling moet worden verricht met het vermenigvuldigtal of met "nul". Heeft een cijfer eenmaal zo'n "opdracht" gegeven, dan is daarmee zijn functie verricht en wordt vernietigd. Het cijfer van de vermenigvuldiger, dat de aard van de optelling bepaalt, staat dus bij iedere slag op de minst significante plaats in het S-register.

We zullen nu enkele uitgewerkte voorbeelden geven van vermenigvuldigen, waarbij de vermenigvuldiger bij het positieve vermenigvuldigtal positief wordt opgeteld:

- a. De schone vermenigvuldiging. Daarbij bevat het A-register bij het begin van de operatie het getal nul. Het eindresultaat moet dus positief zijn.
- b. De additieve vermenigvuldiging. Hierbij stellen we, dat het getal in het A-register aanvankelijk zowel als uiteindelijk positief is.
- c. Dezelfde als bij b. Hierbij stellen we, dat het getal in het A-register zowel aanvankelijk als uiteindelijk negatief is.
- d. Dezelfde als bij b., waarbij we stellen, dat het getal in het A-register aanvankelijk negatief is en tijdens de bewerking positief wordt.

Duidelijkheidshalve geven we de vermenigvuldiger in het S-register aan door kruisjes. Bij een vermenigvuldiging waarbij steeds een positief getal bij het A-register wordt geteld wordt een positieve nul opgeteld, wanneer het betreffende cijfer van de vermenigvuldiger een nul is. Hierbij konden we volstaan met alleen het voorgaande product te schuiven. Het geeft echter organisatorische moeilijkheden bij elke slag de mogelijkheid te openen voor een optelling, zowel als voor een schuif alleen afhankelijk van het bij die slag behorende cijfer van de vermenigvuldiger.

$$\begin{aligned} \text{a) } (+3)(+5) &= +15 \\ 11 \times 101 &= 1111 \end{aligned}$$

$$\begin{aligned} \text{b) } (+3)(+4) + 15 &= 27 \\ 11 \times 100 + 1111 &= 11011 \end{aligned}$$

	T	A	S		T	A	S	
	0	0000	xxxx	0	0	1111	xxxx	15
1 x	0	0101		5	0	0100		4
	0	0101	xxxx	5	(1)	0011	xxxx	19
schuif	0	0010	1xxx		0	1001	1xxx	
1 x	0	0101			0	0100		
	0	0111	1xxx	15	0	1101	1xxx	27
schuif	0	0011	11xx		0	0110	11xx	
0 x	0	0000			0	0000		
	0	0011	11xx	15	0	0110	11xx	27
schuif	0	0001	111x		0	0011	011x	
0 x	0	0000			0	0000		
	0	0001	111x	15	0	0011	011x	27
schuif	0	0000	1111		0	0001	1011	

$$\begin{aligned} \text{c) } (+3)(+4) - 15 &= -3 \\ (0011)(0100) + (10000) &= 1100 \end{aligned}$$

$$\begin{aligned} \text{d) } (+3)(+7) - 15 &= +6 \\ (0011)(0111) + (10000) &= 0110 \end{aligned}$$

	T	A	S		T	A	S	
	1	0000	xxxx	-15	1	0000	xxxx	-15
1 x	0	0100		4	0	0111		+ 7
	1	0100	xxxx	-11	1	0111	xxxx	- 8
schuif	1	1010	0xxx		1	1011	1xxx	
1 x	0	0100			0	0111		
	1	1110	0xxx	- 3	(1)	0010	1xxx	+ 5
schuif	1	1111	00xx		0	0001	01xx	
0 x	0	0000			0	0000		
	1	1111	00xx	- 3	0	0001	01xx	+ 5
schuif	1	1111	100x		0	0000	101x	
0 x	0	0000			0	0000		
	1	1111	100x	- 3	0	0000	101x	+ 5
schuif	1	1111	1100		0	0000	0101	

De eerste drie voorbeelden geven een correcte uitkomst. Bij de 1e slag van voorbeeld b, wijzigt schijnbaar het teken van het A-register. Zoals reeds werd opgemerkt, gebeurt dat alleen bij het woord in de somvormer, terwijl het bij de schuif weer in orde komt. In het vierde voorbeeld is de uitkomst in absolute waarde een eenheid te laag. De algemene regel zegt: Wanneer er na optelling van de tekencijfers een overdracht daaruit ontstaat, moet deze bij de minst significante cijfers worden geteld. (zie ook blz. 77). In voorbeeld d is gesteld, dat positieve

getallen worden opgeteld bij het negatieve getal uit het A-register. Het komt daarbij slechts éénmaal voor, dat het teken van het A-register wisselt en wel van - naar +. Daarbij ontstaat tevens een overdracht uit de tekencijfers. Bij welke slag dat zal gebeuren, is van tevoren niet te zeggen, daar in het algemeen alle getallen een willekeurige waarde kunnen hebben. Het stuit op grote organisatorische moeilijkheden, die overdracht gedurende de betreffende slag bij het minst significante cijfer van het (tussentijdse) product op te tellen, aangezien dat cijfer op elke plaats in het S-register kan staan alleen afhankelijk van de momentele slag. Zoals gezegd, gebeurt dit slechts één keer. De "overlopende" overdracht wordt dan bewaard in een klein register tot na de laatste slag. Het minst significante cijfer van het product staat dan op de minst significante plaats in het S-register. Het kleine register (VC) neemt bij het begin van de vermenigvuldiging het tekencijfer van het M-register over (althans zoals het wordt geïnterpreteerd). Ontstaat er nu tijdens één van de slagen een overdracht uit de tekencijfers, dan wijzigt de flip-flop VC zijn inhoud. In voorbeeld d was $(VC) = 0$ en wordt bij de tweede slag $(VC) = 1$. Na de laatste slag van de vermenigvuldiging wordt de inhoud van het S-register nogmaals (niet verschoven) door het arithmetisch orgaan getransporteerd (correctieslag). Daarbij wordt dan (VC) aan de minst significante zijde erbij geteld. De gecorrigeerde uitkomst van voorbeeld d wordt nu:

S	0101	+5
VC	1	+1
S	0110	+6

Dit is nu dus de correcte uitkomst.

We kunnen ons voorstellen, dat voor de correctieslag het S-register allemaal enen bevatte. Door de optelling van $(VC) = 1$ bevat het S-register dan allemaal nullen, terwijl er uit het meest significante cijfer een overdracht ontstaat, die moet doorlopen tot in het A-register. Daarom volgt nog een correctieslag, waarbij de inhoud van het A-register door het arithmetisch orgaan wordt getransporteerd (niet verschoven) en tegelijkertijd de inhoud van register VC erbij wordt geteld. De inhoud van VC

kan alleen één zijn geweest, indien er een overdracht uit het S-register ontstaat bij deze correctieslag en kan dan ook één blijven bij de correctieslag voor het A-register. Maar bij deze laatste slag moet $(VC) = 0$ worden, indien er geen overdracht ontstond uit het S-register. Dit is het enige geval, waarin de inhoud van VC voor de tweede maal wijzigt.

Hieronder volgen de voorbeelden van vermenigvuldigingen, waarbij een negatief getal wordt opgeteld:

a. De additieve vermenigvuldiging. We stellen hierbij, dat het getal in het A-register zowel aanvankelijk als uiteindelijk negatief is.

b. Dezelfde vermenigvuldiging als bij a. Hierbij stellen we, dat het getal in het A-register zowel aanvankelijk als uiteindelijk positief is.

c. dezelfde als bij a. Nu stellen we, dat het getal in het A-register aanvankelijk positief is en tijdens de bewerking negatief wordt.

d. De schone vermenigvuldiging. Het A-register bevat bij het begin van de operatie het getal +0. Het eindresultaat is dan natuurlijk negatief.

Het vermenigvuldigtal is nu steeds negatief. Is het getal in het A-register eveneens negatief, dan ontstaat er zeker een overdracht uit de tekencijfers. Staat het minst significante cijfer van het tussentijdse product reeds in het S-register, dan is het moeilijk de één daarbij te tellen. Het vermenigvuldigtal moet onder het S-register aangevuld worden met tekencijfers, dat zijn nu dus enen. Wordt daarbij één opgeteld, dan worden deze cijfers nullen en moet een overdracht doorlopen in het A-register. Dit alles echter gebeurt in gedachten, terwijl de overdracht, die in het A-register doorloopt in dit geval dus zeker ontstaat. Deze overdracht wordt dan ook gemaakt en bij het minst significante cijfer van het A-register geteld. Hij wordt dan "extra overdracht" genoemd. In de voorbeelden a en b zal $(VC) = 1$ zijn, daar het teken van het vermenigvuldigtal één is en het teken van het A-register niet verandert ten gevolge van het gestelde. Het is reeds bekend, dat bij de correctieslag het cijfer (VC) wordt opgeteld bij het minst significante cijfer van het eindresultaat. Dat heeft echter hetzelfde gevolg als de extra

overdracht bij de eerste slag. Een van beide zal dus moeten worden geblokkeerd. De keuze is daarbij gevallen op de extra overdracht bij de eerste slag, aangezien in het algemeen (VC) kan veranderen gedurende de bewerking. Geval c. is hiervan het voorbeeld. De inhoud van register VC was weer één, omdat het teken van het vermenigvuldigtal negatief is. Gedurende deze bewerking kan het teken van het A-register slechts eenmaal veranderen en daarbij ontstaat geen overdracht uit de tekencijfers. Het register VC reageert daarop door zijn inhoud te wijzigen en deze wordt nu: (VC) = 0. Ook nu is bij de eerste slag de extra overdracht onderdrukt, terwijl er gedurende de slag, waarbij er geen overdracht ontstond er wel één is opgeteld. Dit compenseert elkaar, zodat nu (VC) = 0 kan worden. Voorbeeld d. is een bijzonder geval van het gestelde onder c., aangezien hier wordt begonnen met een positieve nul in het A-register. Wanneer het bij een bepaalde slag behorende cijfer van de vermenigvuldiger nul is, wordt bij de negatieve vermenigvuldiging een "negatieve nul" opgeteld.

	a) (+3)(-4) - 15 = -27				b) (+3)(-4) + 15 = +3			
	(0011)(1011)+(10000)=100100				(0011)(1011)+01111=011			
	T	A	S		T	A	S	
	1	0000	xxxx	-15	0	1111	xxxx	+15
1 x	1	1011			1	1011		
extra overdr.		-				-		
	(0)	1011	xxxx		0	1010	xxxx	
schuif	1	0101	1xxx	-20	0	0101	0xxx	+10
1 x	1	1011			1	1011		
extra overdr.		1				1		
	1	0001	1xxx		0	0001	0xxx	
schuif	1	1000	11xx	-28	0	0000	10xx	+ 2
0 x	1	1111			1	1111		
extra overdr.		1				1		
	1	1000	11xx		0	0000	10xx	
schuif	1	1100	011x	-28	0	0000	010x	+ 2
0 x	1	1111			1	1111		
extra overdr.		1				1		
	1	1100	011x		0	0000	010x	
schuif	1	1110	0011	-28	0	0000	0010	+ 2
VC			1				1	
	1	1110	0100	-27	0	0000	0011	+ 3

$$c) \begin{matrix} (+3) & (-7) & + & 15 & = & -6 \\ (0011) & (1000) & + & 01111 & = & 1001 \end{matrix}$$

$$d) \begin{matrix} (+3) & (-5) & = & -15 \\ (0011) & (1010) & = & 10000 \end{matrix}$$

	T	A	S		T	A	S	
	0	1111	xxxx	+15	0	0000	xxxx	+ 0
1 x	1	1000		- 7	1	1010		
extra overdr.		-				-		
	0	0111	xxxx	+ 7	1	1010	xxxx	- 5
schuif	0	0011	1xxx		1	1101	0xxx	
1 x	1	1000			1	1010		
extra overdr.		1				1		
	1	1100	1xxx		1	1000	0xxx	-15
schuif	1	1110	01xx	- 6	1	1100	00xx	-15
0 x	1	1111			1	1111		
extra overdr.		1				1		
	1	1110	01xx		1	1100	00xx	-15
schuif	1	1111	001x	- 6	1	1110	000x	-15
0 x	1	1111			1	1111		
extra overdr.		1				1		
	1	1111	001x		1	1110	000x	-15
schuif	1	1111	1001	- 6	1	1111	0000	
VC			0				0	
	1	1111	1001	- 6	1	1111	0000	-15

We zijn hierbij dus tot twee afspraken gekomen, aangevuld met een derde, nl.:

1. Bij optelling van positieve getallen wordt nooit een "extra overdracht" gemaakt.
2. Bij optelling van negatieve getallen wordt altijd een "extra overdracht" gemaakt en opgeteld, behalve bij de eerste slag.
3. Bij de correctieslag wordt het getal (VC) bij het eindresultaat geteld. Onder toepassing van de afspraken geven de voorbeelden een correcte uitkomst.

Wanneer de vermenigvuldiger (S-register) negatief is, wordt het vermenigvuldigtal (M-register) geïnverteerd bij de inhoud van het A-register opgeteld, indien het betreffende cijfer van de vermenigvuldiger een nul is. Eigenlijk worden dus de tekens van het vermenigvuldigtal en de vermenigvuldiger verwisseld (d.w.z. beide getallen worden geheel geïnverteerd beschouwd). Dat heeft geen rekenkundige bezwaren en geeft een correcte uitkomst.

Vermenigvuldiging (toepassing)

In het voorgaande hebben we gesproken over slagen. Het aantal slagen dat wordt gemaakt is bepaald door het aantal cijfers, waaruit de vermenigvuldiger bestaat. Bij de X-1 zijn dat 26 cijfers, zodat 26 slagen moeten worden gemaakt. Ook is er gesproken over de twee laatste slagen, waarbij het S- en het A-register alleen worden getransporteerd. Totaal moeten er dus 28 slagen worden gemaakt, waarbij we de uitgangstoestand geen slag noemen.

Om deze slagen te tellen wordt gebruik gemaakt van een slagenteller, die uit 9 flip-flops bestaat. Vijf flip-flops zijn er nodig voor de teller, omdat het getal 28 wordt geschreven met vijf cijfers uit het tweetalig stelsel (aangegeven met een enkele index). De overige vier flip-flops dienen als hulpteller (aangegeven met een dubbele index). De slagenteller levert signaalcombinaties volgens de code, die wordt gevormd door de achtereenvolgende cijfers van het tweetalig stelsel.

Ook deze teller wordt gedreven door tijdsignalen, namelijk de signalen $DSTo'$ en $DST1'$. Beide signalen worden laag onder voorwaarde dat: $(SDO) = 1$. Het eerste niveau van plaatje SDO' wordt

SDO'	$DSTo'$	$DST1'$
1 DRO' $R2'$	1 SDC' a $4ct'$	1 SDO' a' $4ct'$

laag bij een dubbelregister-opdracht: $(DRO) = 1$ en als de arithmetische controlering in stand 2 staat ($R2'$ is laag). Signaal $DSTo'$ wordt laag op cijfertijd $4a$ en signaal $DST1'$ op cijfertijd $4a'$, zodat de slagenteller om de 8 cijfertijden een stap verder gaat.

Uitgaande van de toestand, waarbij de slagenteller schoon is, zullen we de werking verklaren. De flip-flop STo wordt in stand 1

STo'/STo	gezet door signaal $DST1$ (onaccent),
1 Houd STo' $DSTo$ CST	dat is aangesloten op het houdniveau van plaatje STo en in stand 0 door
1 Houd STo $DST1$	signaal $DSTo$, dat is aangesloten op het

houdniveau van plaatje STo' . Als dus geldt $(DST1) = 1$ wordt eveneens $(STo) = 1$ en als $(DSTo) = 1$ wordt $(STo) = 0$. Flip-flop STo wisselt dus van inhoud bij iedere stap van de slagenteller. De overige flip-flops van de teller behoeven alleen om te gaan bij het vormen van even slagen, omdat ze slechts even waarden

vertegenwoordigen en kunnen dus worden omgezet door signaal $DSTo'$.

ST11'/ST11				ST1'/ST1			
1 Houd	ST11'	CST		1 Houd	ST1'	CST	
2 Lees	ST1	DST1'		2 Lees	ST11'	DSTo'	
-----				-----			
1 Houd	ST11			1 Houd	ST1		
2 Lees	ST1'	DST1'		2 Lees	ST11	DSTo'	

De hulpflip-flop ST11 wisselt van inhoud iedere keer als signaal $DST1'$ laag is, d.w.z. hij blijft dus steeds gedurende twee slagen in dezelfde stand. De inhoud wordt 1, als $(ST1) = 0$ en 0 als $(ST1) = 1$. De flip-flop ST1 gaat om onder controle van signaal $DSTo'$ en wel steeds een slag later dan flip-flop ST11. De inhoud van flip-flop ST1 wordt dan ook 1 als $(ST11) = 1$ en 0 als $(ST11) = 0$.

Van de hulpteller wordt flip-flop ST21 omgezet door signaal $DST1'$

ST21'/ST21				ST2'/ST2			
1 Houd	ST21'	CST		1 Houd	ST2'	CST	
2 Lees	ST2	ST1'	DST1'	2 Lees	ST21'	DSTo'	
-----				-----			
1 Houd	ST21			1 Houd	ST2		
2 Lees	ST2'	ST1'	DST1'	2 Lees	ST21	DSTo'	

namelijk in stand 1 als $(ST1) = 1$ en $(ST2) = 0$ en in stand 0 als $(ST1) = (ST2) = 1$. Ook flip-flop ST2 wordt omgezet onder controle van signaal $DSTo'$ en neemt dan de inhoud van flip-flop ST21 over. Flip-flop ST2 is dus ook steeds een slag achter ten opzichte van flip-flop ST21.

ST31'/ST31					ST3'/ST3		
1 Houd	ST31'	CST			1 Houd	ST3'	CST
2 Lees	ST3	ST2'	ST1'	DST1'	2 Lees	ST31'	DSTo'
-----					-----		
1 Houd	ST31				1 Houd	ST3	
2 Lees	ST3'	ST2'	ST1'	DST1'	2 Lees	ST31	DSTo'

Flip-flop ST31 gaat om in stand 1, als $(ST3) = 0$ en $(ST1) = (ST2) = 1$ ook weer door signaal $DST1'$ en in stand 0 als $(ST1) = (ST2) = (ST3) = 1$. Deze flip-flop wordt één slag later gevolgd door flip-flop 3, die de inhoud van flip-flop 31 overneemt met behulp van signaal $DSTo'$.

ST41'/ST41						ST4'/ST4		
1 Houd	ST41	CST				1 Houd	ST4'	CST
2 Lees	ST4	ST3'	ST2'	ST1'	DST1'	2 Lees	ST41'	DSTo'
1 Houd	ST41					1 Houd	ST4	
2 Lees	ST4'	ST3'	ST2'	ST1'	DST1'	2 Lees	ST41	DSTo'

Tenslotte wordt flip-flop ST41 omgezet in stand 1 door signaal DST1' als (ST4) = 0 en (ST1) = (ST2) = (ST3) = 1 en in stand 0 als (ST1) = (ST2) = (ST3) = (ST4) = 1. De flip-flop ST4 volgt flip-flop ST41 weer een slag later. Aan het laatste gegeven zien we, dat de slagenteller door blijft tellen tot stand 31, daarna overgaat naar stand 0 en weer gaat tellen, als er dan tenminste nog DST-signalen beschikbaar zijn.

DST	ST										slag
10	41	31	21	11	4	3	2	1	0		
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	0	1	0	0	0	0	1	1	2
1	0	0	1	0	0	0	0	0	1	1	3
1	0	0	1	0	0	0	1	0	0	0	4
1	0	1	0	0	0	0	1	1	1	1	7
1	0	1	0	0	0	1	0	0	0	0	8
1	1	0	0	0	0	1	1	1	1	1	15
1	1	0	0	0	1	0	0	0	0	0	16
1	1	1	1	0	1	1	1	0	0	0	28
1	0	0	0	0	1	1	1	1	1	1	31
1	0	0	0	0	0	0	0	0	0	0	0 (32)

Er zijn echter slechts 28 slagen nodig, zodat de teller na de 28e slag moet worden teruggesteld. Daartoe is signaal CST aangesloten

	CST'		op de houdniveau's van de negen accent-
1 R3'	CTR1'	CCTR	plaatjes van de teller. Signaal CST wordt
			hoog als signaal CTR1' laag wordt, mits

de arithmetische controlering in stand 3 staat (R3' is laag) en signaal CCTR (onaccent!) laag is. Zoals we in het voorgaande hebben gezien (blz. 72) komt de arithmetische controlering tengevolge

van signaal CCTR' (en Co') in stand 3. Tevens laat het de cijfertijd-
rijding weer bij nul beginnen te tellen. Signaal CST wordt dus
hoog, de eerstvolgende keer na het terugstellen van de cijfertijd-
rijding, dat signaal CTR1' laag wordt. Daardoor gebeurt dat steeds
op de eerstvolgende cijfertijd 2a' na het laag worden van CCTR'.

CCTR'/CCTR		Op cijfertijd 3a' wordt	
1 Houd	CCTR' Co'	niveau 7 van plaatje CCTR'	
7 ACR 2-3	STS STB 3ct' a' SCO	laag mits de slagenteller	
1 Houd	CCTR	niet schoon is: (STS) = 0,	
		flip-flop STB in stand 0	

staat en deze opdracht geen schuifopdracht is: (SCO) = 0.

Het niveau van plaatje STS' wordt laag, als de slagenteller is
teruggesteld, doordat de onaccentsignalen van de 5 flip-flops
der teller op dit niveau zijn aangesloten.

Verder zijn er nog twee flip-flops (STA en STB), die in hoofdzaak
dienen om signalen uit te coderen bij het begin en aan het eind
van de vermenigvuldiging.

STA'/STA		STB'/STB	
1 Houd	STA'	1 Houd	STB'
2 in verm.	R2' 2ct' ST4 DRO'	3 in verm.	STA' 3ct' a
1 Houd	STA	1 Houd	STB
3 uit verm.	ST4' ST3' ST1' 4ct' a' OR24	2 uit verm.	STA Oct'

Flip-flop STA komt in stand 1 de eerstvolgende 2 cijfertijd, nadat de
arithmetische controlering in stand 2 is gekomen (R2' is laag en
ST4 is nog laag) en (DRO) = 1. Deze voorwaarden zijn uitgecodeerd
op niveau 2 van plaatje STA'. De flip-flop wordt teruggesteld door
het laag gaan van niveau 3 van plaatje STA. Dat gebeurt op de eerst-
volgende cijfertijd 4a', nadat de slagenteller in stand 26 is
gekomen: (ST1) = (ST3) = (ST4) = 1 en als (OR24) = 0. Het laatste
signaal geeft een onderscheiding ten opzichte van de deling.

Op de eerstvolgende cijfertijd 3a na het laag worden van signaal
STA' wordt (STB) = 1 ten gevolge van dit signaal. Evenzo wordt
de flip-flop STB teruggesteld op de eerstvolgende Oct, nadat
flip-flop STA is teruggesteld.

De standen van de slagenteller zijn getekend in het tijdschema op tek. XE 14, en omdat gedurende de slagen 2 t/m 26 alle signalen per slag op dezelfde cijfertijd komen is in dit tijdschema alleen het begin en het eind van de vermenigvuldiging gegeven. Het tijdschema begint op cijfertijd 4a, dat is op het moment dat de getalcyclus begint (ACR in stand 2, zie blz. 71).

CUIR'	CU'	UIR
1 STA' OR24	4 R2' 3ct' DRO'	1 CU
	6 5ct' STA'	2 CUIR

Signaal CUIR is hoog als signaal OR24 laag is (vermenigvuldiging) en zolang signaal STA' laag is. Het tweede niveau van plaatje UIR is daarmee hoog gemaakt. Zodra nu signaal CU eveneens hoog wordt, gaat signaal UIR' laag. Op elke 3 cijfertijd wordt niveau 4 van plaatje CU' laag, mits de arithmetische controlering in stand 2 staat (R2' is laag) en (DRO) = 1. Op elke 5 cijfertijd wordt niveau 6 van ditzelfde plaatje laag, zolang geldt (STA) = 1. De eerste keer in het tijdschema (op cijfertijd 3a') dat signaal CU hoog wordt, heeft weinig zin, aangezien er dan in de somvormer nog geen woord staat, dat voor deze getalcyclus van belang is. Het komt dan wel in het U-register, terwijl voorkomen moet worden, dat het daarna door het A-register wordt overgenomen. Dat mag op dit moment nooit gebeuren, want bij de additieve vermenigvuldiging moet de inhoud van het A-register onbeschadigd blijven en bij de schone vermenigvuldiging moet het A-register schoon zijn.

AI'/AI	iedere 4 cijfertijd niveau 8
3 CCTR' STS Oct' SCO	van plaatje AI' laag, zolang
4 STA' STB 4ct' OR22' OR24	geldt: (STB) = 1. Niveau 3
8 STB' 4ct'	van dit plaatje wordt laag
<hr/>	
1 AI	aan het eind van de laatste
3 STA' STB	correctieslag tengevolge van
	de signalen CCTR' en Oct',

mits de slagenteller nog niet is teruggesteld (STS) = 0 en deze opdracht geen schuifopdracht is : (SCO) = 0. Bij de opdrachten 18 en 19 moet het A-register worden schoongemaakt. Dat gebeurt met het AI-sigitaal bij het begin van de eerste slag.

Op 4 cijfertijd wordt niveau 4 van plaatje AI' laag, mits geldt: (STA) = 1, (STB) = 0, (OR24) = 0 en (OR22) = 1. De eerste drie signalen zijn reeds besproken, het laatste signaal dient ter bepaling van de opdrachten voor schone vermenigvuldiging. (18 en 19). Zoals gewoonlijk wordt het hoge signaal AI omgekeerd op het onaccentplaatje, zodat signaal AI' laag zou zijn. Het plaatje AI (onaccent!) bestaat echter uit meer niveau's. Niveau 3 hiervan is laag, als: (STA) = 1 en (STB) = 0, dat is dus op het juiste moment. Nu is zowel signaal AI als signaal AI' hoog, zodat het A-register zeker schoon is.

Eveneens op 4 cijfertijd wordt het S-register geselecteerd door het coderend circuit OI met behulp van het multiplietsignaal LS'. De niveau's 3 en 4 van plaatje LS' worden laag op 4 cijfertijd, het derde als: (STA) = 1 en het vierde als: (STB) = 1. Niveau 4 is

LS'	nodig om het S-register te selecteren bij de
3 STA' 4ct'	27e slag. Op hetzelfde moment wordt de opteller
4 STB' 4ct'	in stand 6 (transportstand) gezet, waardoor de
	inhoud van het S-register onveranderd in de som-

vormer verschijnt. Op cijfertijd 5 neemt het U-register het woord uit de somvormer één plaats naar rechts verschoven over (CU-UIR).

	SI'	Voor nu het S-register het nieuwe
2 STA' 7ct'	OR24	woord uit het U-register overneemt,
4 STS STB 5ct'	a SCO	moet eerst de opteller in de juiste
		stand worden gezet op grond van het

"oude" cijfer (So). Daarom neemt het S-register pas op 7 cijfertijd de inhoud van het U-register over, doordat niveau 2 van plaatje SI' laag wordt bij de vermenigvuldiging: (STA) = 1 en (OR24) = 0.

Bij de correctieslag kan dit niveau niet laag worden. Daarvoor dient dan niveau 4, dat laag wordt als de slagenteller nog niet is teruggesteld: (STS) = 0, als: (STB) = 0 en wel op cijfertijd 5a, mits: (SCO) = 0. Ondertussen is op 6 cijfertijd het A-register geselecteerd, door het laag worden van niveau 3 op plaatje LA'.

De opteller wordt eveneens op 6 cijfertijd in de juiste stand

LA'	gezet, met behulp van signaal OIV' dat laag wordt
3 STA' 6ct'	door de niveau's van plaatje OIV hoog te maken
	met de signalen STA, OR24' en 6ct. De opteller

komt in de gewenste stand afhankelijk van vier factoren, namelijk: So, TS, OR21 en TM. Het cijfer TS geeft aan hoe So moet worden

OIV beschouwd. Wanneer geldt $(So) = (TS)$, behoeft alleen te worden geschoven, dat wil zeggen er wordt een positieve of een negatieve nul opgeteld en wanneer (So) en (TS) ongelijk zijn, moet het vermenigvuldigtal positief of negatief worden opgeteld. Volgens de opdrachtencode geeft (OR21) aan hoe het vermenigvuldigtal moet worden geïnterpreteerd, terwijl (TM) aangeeft of het vermenigvuldigtal zelf positief of negatief is.

In onderstaande tabel zijn de mogelijke combinaties vermeld met daarachter de stand van de opteller.

	So	TS	OR21	TM	opteller stand
schuif	0	0	0	0	12
	0	0	0	1	13
	0	0	1	0	13
	0	0	1	1	12
aftrekken	0	1	0	0	1
	0	1	0	1	1
optellen	0	1	1	0	0
	0	1	1	1	0
	1	0	0	0	0
	1	0	0	1	0
aftrekken	1	0	1	0	1
	1	0	1	1	1
schuif	1	1	0	0	13
	1	1	0	1	12
	1	1	1	0	12
	1	1	1	1	13

Van de signalen, die in deze tabel worden genoemd, zijn de juiste polariteiten op niveau's van de COI-plaatjes aangesloten, die de opteller in de juiste stand zetten.

Met de opteller in stand 12 wordt een optelling met een positieve nul uitgevoerd en geen extra overdracht gemaakt.

Stand 13 geeft een optelling met een negatieve nul, waarbij wel een extra overdracht gemaakt kan worden.

	COIo'			COI1'		
1 OIV'	So'	TS	OR21	1 OIV'	So	TS' OR21
2 OIV'	So	TS'	OR21'	2 OIV'	So'	TS OR21'

COI12'				COI13'					
1	OIV'	WTM	TS	So	1	OIV'	WTM	TS'	So'
2	OIV'	WTM'	TS'	So'	2	OIV'	WTM'	TS	So

Het signaal WTM, dat op de plaatjes COI12' en COI13' is aangesloten,

WTM'		TM	OR21	WTM	is de uitcodering van het teken van
1	TM'	OR21	0	0	het vermenigvuldigtal (TM), zoals
2	TM	OR21'	0	1	het door de opdracht wordt geïnter-
			1	0	preteerd. Het is duidelijk, dat
			1	1	alleen een niveau van plaatje WTM'
				0	laag behoeft te worden, als de opdracht het teken van het M-regis-

ter niet geïnverteerd beschouwt: (OR21) = 0 en (TM) = 1 en als de opdracht het teken negatief interpreteert: (OR21) = 1 en (TM) = 0. Signaal GTM is de uitcodering van (TM), zoals het door de opteller wordt gezien en wordt gebruikt om het nieuwe teken van het A-register te bepalen en voor de flip-flop VC.

GTM		Opteller	COA	COB	GTM
1	COA	TM	0	0	gelijk TM
2	COB'	TM'	1	1	gelijk TM'
		12	0	1	altijd laag (0)
		13	1	0	altijd hoog (1)

In de tabel zijn de signalen gegeven, die het plaatje GTM (onaccent!) moeten beïnvloeden. Signaal GTM' is dus laag, wanneer tengevolge van de optelling de inhoud van het A-register steeds negatiever wordt.

Bij iedere optelling moet het teken van het A-register worden bepaald. Dat komt dan normaal en dus niet naar rechts verschoven op de meest significante plaats in het U-register te staan en ontstaat op de normale wijze uit de optelling van de tekencijfers en een eventuele opdracht.

Flip-flop U26 is nu immers niet nodig om (SV26) over te nemen, aangezien er één plaats naar rechts wordt geschoven. Nu neemt flip-flop U25 dit cijfer over en blijft flip-flop U26 dus vrij voor het nieuwe teken.

In onderstaande tabel zijn de verschillende mogelijkheden gegeven van de combinatie der signalen, die het nieuwe teken van het A-register vaststellen.

		U26'/U26			GTM	TA	CV26	U26 = Ta nieuw
					oud			
1	U26'	CU						
8	GTM'	CV26	UIR'	CCC	0	0	0	0
9	TA'	CV26	UIR'	CCC	0	0	1	onmogelijk
9	TA'	CV26	UIR'	CCC	0	1	0	1
10	TA'	GTM'	UIR'	CCC	0	1	1	0 (tekenwisseling)
10	TA'	GTM'	UIR'	CCC	1	0	0	1 (tekenwisseling)
1	U26				1	0	1	0
					1	1	0	onmogelijk
					1	1	1	1

Op plaatje U26' zijn op de niveau's 8,9 en 10 de drie gevallen uitgecodeerd, waarin het nieuwe teken: (TA) = 1 moet worden. Aangezien signaal CCC in de transportstand van de opteller niet laag is, bepaalt de combinatie van de signalen UIR' en CCC, dat deze niveau's alleen werken in de andere standen van de opteller bij de vermenigvuldiging. Van de tabel zelf is verder weinig te zeggen, want deze geeft eenvoudig de logische optelling van de drie genoemde cijfers.

Het cijfer, dat uit het A-register loopt tengevolge van de schuif, moet door het S-register worden opgevangen. Daartoe neemt de flip-flop S25 het cijfer (SVo) over onder voorwaarde, dat de signalen UIR', STS, SCO en LA' laag zijn. Het eerste signaal

		S25'/S25					
1	Houd	S25'	SI				bepaalt dat dit alleen moet
3	Lees	SVo'	UIR'	LA'	STS	SCO	gebeuren bij een "schuif naar
3	Lees	SVo'	UIR'	LA'	STS	SCO	rechts". Het tweede signaal
1	Houd	S25					geeft aan, dat de slagenteller
							nog aan het tellen moet zijn

en signaal SCO voorkomt de werking van dit niveau bij de schuifopdracht, terwijl het laatste signaal bepaalt, dat dit alleen moet gebeuren zolang het A-register is geselecteerd.

Aan het eind van de vermenigvuldiging kan het teken van het S-register gelijk worden gemaakt aan dat van het A-register. Flip-flop TS wordt nu dubbelzijdig ingelezen met behulp van niveau 6 op het accentplaatje en niveau 4 op het onaccentplaatje onder voorwaarde: (STB) = 1, (STA) = 0 en (OR24) = 0. Het laatste signaal onderscheidt de vermenigvuldigingopdracht, terwijl de eerste twee signalen het eind van de vermenigvuldiging aangeven, d.w.z.

		TS' / TS			
1	Houd	TS'	SI		
2	Houd	TS'	DRO'		
6	Lees	TA'	STB'	STA	OR24
7	Lees	U26'	SI'	DRO	

1	Houd	TS			
4	Lees	TA	STB'	STA	OR24

na de 26e slag. We zien nu tevens, dat de flip-flop TS gedurende de vermenigvuldiging niet beïnvloed kan worden met behulp van signaal SI, want niveau 7 kon niet laag worden, omdat signaal DRO hoog was en wanneer het 1e houdniveau hoog werd tengevolge van signaal SI, bleef de werking van het 2e houdniveau mogelijk, omdat signaal DRO' laag was. Vandaar dat dubbelzijdig moet worden ingelezen.

Het kleine register VC wordt op 7 cijfertijd gedurende de 1e slag: (STA) = 1 en (STB) = 0, dubbelzijdig ingelezen en neemt daarbij

		VC' / VC			
1	Houd	VC'			
2	Lees	CV26'	GTM	3ct'	
3	Zet	STA'	STB	7ct'	GTM'

1	Houd	VC			
2	Lees	CV26	GTM'	3ct'	
3	Zet	STA'	STB	7ct'	GTM
4	laatste slag	STA	stb	4ct'	CV25

(GTM) over. Dat gebeurt dan met behulp van de 3e niveau's van beide plaatjes. Zoals bekend is, moet deze flip-flop worden omgezet, afhankelijk van het feit of er een overdracht uit de tekencijfers ontstaat of niet. Aan het eind van elke slag op 3 cijfertijd bestaat daartoe de mogelijk-

heid met behulp van de 2e niveau's van beide plaatjes. Tijdens de eerste correctieslag (STA) = (STB) = 0 wordt op 4 cijfertijd dit register teruggesteld, indien daarbij geen overdracht ontstond, die aan het A-register moest worden doorgegeven.

De inhoud van register VC, wordt opdezelfde wijze bij de registers A en S geteld, als de ophoging van de opdrachtteller tot stand kwam (blz. 59), dus door een niveau van plaatje CV26'A laag te maken. Niveau 3 van dit plaatje wordt laag als: (VC) = 1 en niet

		CV26'A				CX'					
3	CX'	VC'	SCO			1	ST4'	ST3'	ST1'	STo'	OR24
4	R2'	DRO'	BEC	GTM'	CX	2	ST4'	ST3'	ST2'		OR24

bij schuifopdrachten: (SCO) = 0. Bovendien moet signaal CX' laag zijn. Dit signaal is een uitcodering van de slagen 27 (niveau 1) en 28 (niveau 2) bij de vermenigvuldiging: (OR24) = 0. Om de extra overdracht te maken moet signaal CV26'A eveneens laag worden.

Dit moet gebeuren wanneer de arithmetische controlering in stand 2 staat (R2' is laag), signaal DRO' laag is en de slagenteller nog niet in de correctiestanden (27 en 28) staat: (CX) = 0. Deze signalen zijn aangesloten op niveau 4 van plaatje CV26'A.

Bovendien moet de extra overdracht alleen worden gemaakt, wanneer een negatief getal bij het A-register wordt geteld: (GTM) = 1, terwijl deze overdracht niet wordt gemaakt bij de eerste slag.

BEC'	Daartoe is de eerste slag uitgecodeerd
1 ST1 ST2 ST3 ST4	op plaatje BEC' en het uitgangssignaal
	van dit plaatje eveneens aangesloten

op niveau 4 van plaatje CV26'A.

Deling (algemeen)

Bij een deling met getallen uit het tientalligstelsel gaat men als volgt te werk: Men schrijft het deeltal op en daaronder de deler, (in gedachten) aangevuld met nullen, zodat de meest significante cijfers van beide getallen onder elkaar komen. Daarna trekt men de laatste van de eerste af, zoveel maal als mogelijk is en zodanig dat er nog een positieve rest overblijft, aannemende dat deeltal, zowel als deler positief zijn. Dat wordt trouwens altijd eerst aangenomen. Pas na de berekening wordt het teken van het quotiënt bepaald uit de tekens van de deler en deeltal. Vervolgens wordt de deler (eventueel vermenigvuldigd met een getal kleiner dan 10) daaronder geschreven, maar nu met één nul minder. Daarna worden de handelingen herhaald en zo voort tot de rest kleiner is dan de deler. Men kan het ook zo beschouwen, alsof de rest steeds één plaats naar links werd geschoven boven de deler. Onderstaand voorbeeld licht een en ander nader toe

$\frac{45867}{147} = 312 \text{ rest } 3$	<p>3 x 147</p> <p>schuif</p> <p>1 x 147</p> <p>schuif</p> <p>2 x 147</p> <p>rest</p>	$\begin{array}{r} 45867 \\ 44100 \\ \hline 1767 \\ 1767 \\ \hline 1470 \\ 297 \\ 297 \\ \hline 294 \\ 3 \end{array}$
---	---	--

Hierbij dient nog te worden opgemerkt, dat wanneer de deler groter is dan de tussentijdse rest, de laatste alleen een plaats naar links wordt geschoven, terwijl een nul in het quotiënt wordt ingevuld.

Bij gebruik van het tweetalig stelsel zijn er steeds maar twee mogelijkheden, namelijk de deler kan wel van de tussentijdse rest worden afgetrokken of niet. Het eerste geval is op zichzelf reeds duidelijk, terwijl het tweede geval een nadere toelichting vereist. De $X-1$ kan niet "van tevoren" bekijken of de aftrekking zal gaan of niet er voert deze dus zonder meer uit. Daarbij ontstaat echter een rest met een negatief teken, als we weer aannemen, dat zowel deeltal als deler positief zijn. Om nu volgens het hierboven besproken systeem verder te gaan, moet eerst de "verkeerd" afgetrokken deler er (niet verschoven) weer bij worden opgeteld. Met de aldus teruggekregen rest kan dan weer verder worden gewerkt. Dat duurt echter tweemaal zo lang, als wanneer het "goed" was gegaan. Daarom werken we verder met de negatieve rest. Deze schuiven we normaal één plaats naar links en tellen er nu de deler bij op. Is de rest daarna nog steeds negatief, dan verschuiven we hem weer een plaats en tellen de deler er weer bij op. Dat gebeurt net zo lang tot het teken van de rest weer positief is en daarna wordt het oude systeem weer herhaald.

Het volgende heeft nu plaats gevonden. Tengevolge van het schuiven wordt de deler steeds door twee gedeeld (tweetalig stelsel). Ten opzichte van de laatste keer, dat de rest positief was vóór dat deze negatief werd, is de deler eenmaal geheel van die rest afgetrokken, vervolgens werd de helft van de deler er bijgeteld en daarna nog een kwart van de deler (de helft van de helft). Daarmee werd bijvoorbeeld de rest weer positief. Bij deze drie slagen is er dus in totaal één kwart van de deler van de eerstgenoemde rest afgetrokken, waardoor de laatste positieve rest ontstond. Populair gezegd werken we steeds naar nul toe. De rest moet steeds tussen nul en de deler liggen, beide in absolute waarde beschouwd. Verder spreken we af, dat de laatste rest hetzelfde teken als het deeltal moet hebben, anders is er meer dan één oplossing mogelijk:

$$\frac{+7}{+3} = +2 \text{ rest } +1 \quad \frac{+7}{+3} = +3 \text{ rest } -2 .$$

Wanneer de deler als negatief getal van het deeltal of bij de tussentijdse rest moet worden opgeteld wordt er, evenals dat bij de vermenigvuldiging het geval was, een extra overdracht bij gemaakt. We denken de deler dan onder het S-register weer aangevuld met tekencijfers, zodat deze extra overdracht doorloopt tot onder het minst significante cijfer van het A-register en daar dus meteen kan worden bijgeteld. Verder is uit de vermenigvuldiging bekend, dat bij tekenwisseling van het tussentijdse resultaat, dit een eenheidje te groot of te klein is. Dit geldt dan natuurlijk ook voor de tussentijdse rest bij de deling. Het teken hiervan kan echter voortdurend wisselen. Nu is het teken van de laatste rest volgens afspraak gelijk aan dat van het deeltal, dat wil zeggen er heeft een even aantal wisselingen plaats gehad. Bij tekenwisseling van positief naar negatief is het resultaat een eenheid te groot, omdat de extra overdracht er ten onrechte bij is geteld, want er ontstond geen overdracht uit de tekencijfers.

Bij tekenwisseling van negatief naar positief is het resultaat een eenheid te klein, omdat er geen extra overdracht is bijgeteld en ten onrechte, want er ontstond wel een overdracht uit de tekencijfers. Was bijvoorbeeld het deeltal positief, dan is de rest na de eerste tekenwisseling een eenheid te groot en na de daaropvolgende tekenwisseling, waarbij het teken van die tussentijdse rest weer gelijk wordt aan het teken van het deeltal, is die betreffende rest dus een eenheid te klein ten opzichte van de voorgaande tussentijdse rest, maar wel correct ten opzichte van het oorspronkelijke deeltal. Zo is de laatste rest tengevolge van het even aantal tekenwisselingen dus ook correct. Hetzelfde geldt voor het geval, dat het deeltal negatief was. Wanneer men een rest nul vindt, heeft ook die nul hetzelfde teken als het oorspronkelijke deeltal.

De quotiëntcijfers worden als volgt bepaald. Wordt een rest gevonden met het teken van het deeltal, dan wordt in het quotiënt een één geschreven in absolute waarde. Zetten we de ontstaansmogelijkheden van de quotiëntcijfers onder elkaar, dan ontstaat de volgende tabel waarbij TA het teken van het oorspronkelijke

deeltal is, TA_n het teken van het n^e tussentijdse resultaat, (A) de rekenkundige waarde daarvan en $|Q|$ de absolute waarde van het quotiëntcijfer.

TA_0	TA_n	$ Q $	rest
0	0	1	(A)
0	1	0	$(A)-1$
1	0	0	$(A)+1$
1	1	1	(A)

We zullen deze tabel uitbreiden door het teken van het deeltal, zoals het door de opdracht wordt geïnterpreteerd (WTM) er bij te voegen en na te gaan, wat het teken van het quotiënt (TQ) moet worden. Het quotiëntcijfer, zoals het moet worden opgeborgen, is weer afhankelijk van (TQ).

TA_0	TA_n	WTM	$ Q $	TQ	Q
0	0	0	1	0	1
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	0
1	1	1	1	0	1

In de X-1 wordt niet de absolute waarde van het quotiëntcijfer genoteerd, maar het quotiëntcijfer, zoals het werkelijk moet zijn. Er wordt dan een één in het quotiënt geschreven, indien het teken van de tussentijdse rest overeenkomt met het teken van de deler, zoals deze door de opdracht wordt geïnterpreteerd (WTM) en een nul in het andere geval. Immers, gesteld dat het deeltal en de deler positief zijn, dan "ging" de aftrekking, wanneer de tussentijdse rest ook positief blijkt te zijn. In het quotiënt moet dan een één worden genoteerd. Blijkt de rest negatief te zijn, dan "ging" de aftrekking niet en wordt een nul genoteerd. Is de deler positief en het deeltal en de rest negatief, dan moet er een geïnverteerde één (dus een nul) in het quotiënt worden geschreven, aangezien de aftrekking "ging" en het quotiënt negatief moet zijn (WTM en TA_n zijn ongelijk). Wordt de rest positief, dan moet een één worden genoteerd, want de aftrekking "ging" niet (WTM en TA_n zijn gelijk).

Hieruit blijkt dus, dat het teken van het oorspronkelijk deeltal als gegeven overbodig is, als het quotiëntcijfer op deze wijze wordt bepaald.

Van het deeltal wordt het meest significante gedeelte in het A-register gezet en het minst significante gedeelte in het S-register. De deler staat in het M-register. De deler wordt nu steeds van het getal, dat in het A-register staat afgetrokken (c.q. erbij opgeteld). Door het naar links schuiven komt de flip-flop S_0 steeds vrij. Deze flip-flop wordt daarom benut voor het quotiëntcijfer. Bij het einde van de deling staat de rest in het A-register en het quotiënt in het S-register, terwijl het deeltal als zodanig verloren is gegaan. In het S-register is ruimte voor 26 cijfers en een tekencijfer. Het tekencijfer van het quotiënt wordt bij het einde van de deling afzonderlijk gemaakt. De 26 cijfers beperken de capaciteit van de deling, want er kunnen slechts 26 slagen worden gemaakt, waarbij een quotiëntcijfer kan worden opgeborgen. Er wordt met een schuif begonnen, waarbij dan een quotiëntcijfer wordt gemaakt, dat weer verloren gaat, omdat het uiteindelijk buiten de capaciteit van het S-register loopt. De deler wordt van het getal, zoals dat na de schuif in het A-register staat afgetrokken. De rest moet dan kleiner zijn dan de deler, want anders zou de deler zonder te schuiven voor de tweede maal van deze rest afgetrokken kunnen worden en dat zou inhouden, dat het quotiënt uit 27 cijfers ging bestaan. De volgende regel moeten we nu stellen: De absolute waarde van het getal in het A-register vóór de schuif moet kleiner zijn dan de absolute waarde van de deler (M). Immers een schuif naar links wil zeggen een vermenigvuldiging met 2. Als nu vóór de schuif geldt: $(A) < (M)$, is na de schuif vanzelfsprekend $(A_{na}) = 2(A_{voor}) < 2(M)$, en zal de rest (onder het A-register) na aftrekking van (M) kleiner zijn dan (M), hetgeen was afgesproken. Ook gedurende het verdere verloop zal de rest (onder het A-register) steeds kleiner zijn dan de deler.

Het S-register is bij het begin van stand 2 van de slagenteller een plaats naar links geschoven, omdat bij stand 1 het A-register het meest significante cijfer daarvan reeds had overgenomen. Daardoor is de minst significante plaats in het S-register vrijgekomen, terwijl alleen nog maar het A-register is geschoven en dus nog geen quotiëntcijfer is bepaald. Zoals gezegd nam het A-register bij stand 1 het eerste cijfer uit het S-register over en bij stand 26 dus het 26e cijfer, zodat bij stand 27 geen cijfer van het deeltal meer in het S-register beschikbaar is. Daarom wordt het A-register bij stand 27 niet geschoven, maar daarna (bij het begin van stand 28) het S-register nog wel eenmaal. Het gevolg hiervan is, dat het eerste (onbepaalde) quotiëntcijfer uit het S-register wordt geschoven en verloren gaat. Het quotiëntcijfer, dat ontstaat gedurende de 2e slag, wordt opgeborgen bij het begin van de 3e slag, dat is dus de 2e maal dat het S-register wordt geschoven. Het 26e quotiëntcijfer wordt dus opgeborgen bij het begin van de 28e slag. Nu is er gesteld, dat de rest bij het einde van de deling hetzelfde teken moet hebben als het oorspronkelijke deeltal. De mogelijkheid bestaat, dat bij de 27e slag door de optelling resp. aftrekking van de deler de rest een tegengesteld teken krijgt. Dat wordt dan bij de 28e slag gecorrigeerd door de deler nu af te trekken resp. op te tellen bij de niet verschoven rest. Ook hierna wordt de rest niet verschoven.

Bij de opdrachten 26 en 27 bestaat het gedeelte van het deeltal in het A-register uit een willekeurig woord, terwijl het gedeelte in het S-register alleen bestaat uit tekencijfers (in gedachten). Het deeltal is dan gegeven door $2^{26}(A)$.

Hieronder volgen nog enkele uitgewerkte voorbeelden, waarin een en ander nader wordt toegelicht. Terwille van het overzicht bevatten de registers hierin slechts vier cijfers, hetgeen in principe geen verschil maakt. Ook is het quotiënt om dezelfde reden niet in het S-register geschreven, maar ter plaatse aangevuld met kruisjes. De letters: "e.o." betekenen: extra opdracht.

$$a) \quad \frac{+9}{+3} = +3 \text{ rest } +0$$

$$01001/0011=0011 \text{ rest } 000$$

+9	0 0000	1001		
	0 0001	001x	schuif	
-24	1 1100			
e.o.	1			
-14	1 1110	001x	0 x	
	1 1100	01xx	schuif	
+12	0 0011			
-2	1 1111	01xx	0 x	
	1 1110	1xxx	schuif	
+6	0 0011			
+3	0 0001	1xxx	1 x	
	0 0011	xxxx	schuif	
-3	1 1100			
e.o.	1			
+0	0 0000	xxxx	1 x	

$$b) \quad \frac{-9}{+3} = -3 \text{ rest } -0$$

$$10110/0011 = 1100 \text{ rest } 111$$

-9	1 1111	0110		
	1 1110	110x	schuif	
+24	0 0011			
+14	0 0001	110x	1 x	
	0 0011	10xx	schuif	
-12	1 1100			
e.o.	1			
+2	0 0000	10xx	1 x	
	0 0001	0xxx	schuif	
-6	1 1100			
e.o.	1			
-3	1 1110	0xxx	0 x	
	1 1100	xxxx	schuif	
+3	0 0011			
-0	1 1111	xxxx	0 x	

$$c) \quad \frac{+10}{+5} = +2 \text{ rest } +0$$

$$01010/0101 = 010 \text{ rest } 000$$

+10	0 0000	1010		
	0 0001	010x	schuif	
-40	1 1010			
e.o.	1			
-29	1 1100	010x	0 x	
	1 1000	10xx	schuif	
+20	0 0101			
-9	1 1101	10xx	0 x	
	1 1011	0xxx	schuif	
+10	0 0101			
+0	0 0000	0xxx	1 x	
	0 0000	xxxx	schuif	
-5	1 1010			
e.o.	1			
-4	1 1011	xxxx	0 x	
+5	0 0101		correctie	
+0	0 0000	xxxx		

$$d) \quad \frac{-10}{+5} = -2 \text{ rest } -0$$

$$10101/0101 = 101 \text{ rest } 111$$

-10	1 1111	0101		
	1 1110	101x	schuif	
+40	0 0101			
+29	0 0011	101x	1 x	
	0 0111	01xx	schuif	
-20	1 1010			
e.o.	1			
+9	0 0010	01xx	1 x	
	0 0100	1xxx	schuif	
-10	1 1010			
e.o.	1			
-0	1 1111	1xxx	0 x	
	1 1111	xxxx	schuif	
+5	0 0101			
+4	0 0100	xxxx	1 x	
-5	1 1010			
e.o.	1			
-0	1 1111	xxxx		

$$e) \quad \frac{+0}{+7} = +0 \text{ rest } +0$$

$$000/0111=000 \text{ rest } 000$$

+0	0 0000	0000		
	0 0000	000x	schuif	
-56	1 1000			
e.o.	1			
-55	1 1001	000x	0 x	
	1 0010	00xx	schuif	
+28	0 0111			
-27	1 1001	00xx	0 x	
	1 0010	0xxx	schuif	
+14	0 0111			
-13	1 1001	0xxx	0 x	
	1 0010	xxxx	schuif	
+7	0 0111			
-6	1 1001	xxxx	0 x	
+7	0 0111		correctie	
+0	0 0000	xxxx		

$$f) \quad \frac{-87}{+9} = -9 \text{ rest } -6$$

$$10101000/01001 = 10110 \text{ rest } 1001$$

-87	1 1010	1000		
	1 0101	000x	schuif	
+72	0 1001			
-15	1 1110	000x	0 x	
	1 1100	00xx	schuif	
+36	0 1001			
+20	0 0101	00xx	1 x	
	0 1010	0xxx	schuif	
-18	1 0110			
e.o.	1			
+2	0 0001	0xxx	1 x	
	0 0010	xxxx	schuif	
-9	1 0110			
e.o.	1			
-6	1 1001	xxxx	0 x	

Deling (toepassing)

Het tijdschema van de deling op tek.XE 13 begint op het moment, dat de arithmetische controlering in stand 2 komt (zie blz. 71) dus na de opdrachtcyclus, terwijl ook hiervan alleen het begin en het eind is gegeven op tek. XE 13.

STA'/STA					STB'/STB			
1	Houd	STA'			1	Houd	STB'	
1	Houd	STA			2	Deel	STA'	1ct' OR24'
2	Deel	ST4'	ST3'	ST2'	6ct'	1	Houd	STB

De beide flip-flops STA en STB hebben hier weer dezelfde functies als bij de vermenigvuldiging. De eerste komt onder dezelfde voorwaarden in stand 1 als bij de vermenigvuldiging. Ook het terugstellen van flip-flop STB geschiedt onder dezelfde voorwaarden (zie blz. 117). Het terugstellen van flip-flop STA gebeurt met behulp van niveau 2 op plaatje STA en wel op 6 cijfertijd, mits de slagenteller in stand 28 staat. De flip-flop STB wordt in stand 1 gezet op de eerstvolgende 1 cijfertijd, nadat flip-flop STA in stand 1 kwam en als geldt $(OR24) = 1$. Deze voorwaarden zijn uitgecodeerd op niveau 2 van plaatje STB'.

Signaal CU' wordt op dezelfde tijden en onder dezelfde voorwaarden laag, als bij de vermenigvuldiging (blz.118). De eerste twee keren, dat dit signaal laag wordt hebben weinig zin, aangezien het U-register weer wordt teruggesteld, zonder dat het A- of het S-register deze gegevens hebben overgenomen.

UIL		CUIL'			CUILA						
1	CU	1	STA'	CUILA'	OR24'	1	ST4'	ST3'	ST1'	STo'	CTR2
2	CUIL										

Het niveau van plaatje CUILA is laag, als de slagenteller in stand 27 staat en als signaal CTR2 laag is, dat is dus van 0 tot en met 3 cijfertijd. Signaal CUILA' is dan hoog. Signaal CUIL' is laag bij de deling: $(OR24) = 1$, mits de signalen STA' en CUILA' laag zijn en is dus hoog gedurende de 2e helft van de 27e slag. Signaal UIL' wordt laag, als signaal CU zowel als signaal CUIL hoog zijn. Dan neemt het U-register het in de somvormer ontstane woord één plaats naar links verschoven over.

Het inleessignaal van het A-register komt op geheel overeenkom-

stige wijze als bij de vermenigvuldiging
 3' STB' 7ct' OR24' (blz. 118) Het inleessignaal voor het S-register
 komt wel op hetzelfde tijdstip (7ct') als bij
 de vermenigvuldiging, maar onder andere voorwaarden, nl.: (STB) = 1
 en (OR24) = 1.

Aangezien bij het begin van de deling het A- en S-register elk
 een gedeelte van hetzelfde deeltal bevatten, moeten de tekens van
 beide registers gelijk zijn. Zo ze al niet gelijk waren, worden ze
 gelijk gemaakt met behulp van de signalen STA', STB en OR24' op
 niveau 3 en 2 respectievelijk van de plaatjes TS' en TS. Deze

	TS' / TS							
1 Houd	TS'	SI						flip flop neemt dan (TA)
2 Houd	TS'	DRO'						over en wel dubbelzijdig
3 Begin	STA'	STB	TA'	OR24'				om dezelfde reden als bij
4 Tq	2ct'	R3'	TA'	WTM	DRO'	OR24'		de vermenigvuldiging.
5 Tq	2ct'	R3'	TA	WTM'	DRO'	OR24'		Dit gelijkmaken der tekens
<hr/>								
1 Houd	TS							is speciaal gedaan voor
2 Begin	STA'	STB	TA	OR24'				de opdrachten 26 en 27,
3 Tq=0	1ct'	R3'	DRO'	OR24'				waarbij het A-register

tekencijfer. Daarvoor wordt dan (TS) gebruikt, aangezien (TA)
 gedurende de bewerking steeds kan wijzigen.

Aan het eind van de deling moet het tekencijfer van het quotiënt
 in flip-flop TS worden gezet. Daartoe wordt deze flip-flop terug-
 gesteld op 1 cijfertijd als de arithmetische controlering in
 stand 3 staat (R3' is laag) en onder voorwaarde dat (DRO) = (OR24)=1
 Op de volgende cijfertijd wordt onder dezelfde voorwaarden (TS) = 1
 gemaakt met behulp van de niveau's 4 en 5 op het accentplaatje als
 (TA) en (WTM) ongelijk zijn. Uit de rekenkunde is bekend, dat het
 teken van het quotiënt negatief is, als de tekens van deeltal en
 deler ongelijk zijn. Het cijfer (TA) heeft nu dezelfde waarde als
 het oorspronkelijke deeltal, terwijl het cijfer (WTM) het teken
 van de deler is, zoals de opdracht het beschouwt.

Tengevolge van een optelling ontstaat een nieuw tekencijfer normaal
 in SV26. Nu zou door de schuif dit cijfer verloren gaan. Wanneer
 het S-register wordt geschoven is dat geen bezwaar, aangezien zelfs
 het cijfer (SV25) verloren gaat, doordat het wel wordt overgenomen

door flip-flop U26, maar daarna niet door de flip-flop TS (S26) wordt overgenomen. Deze flip-flop blijft immers gedurende de gehele deling onveranderd. Evenwel bij de optelling en schuif van het A-register is (SV26) wel van belang, daar dit het "nieuwe" cijfer (TA) is. Daarom is het U-register uitgebreid met flip-flop U27, die dan (SV26) overneemt.

U27'/U27				
1 Houd	U27'	6ct	2ct	
2 Houd	U27'	SCO	2ct	
3 Lees	SV26'	LA'	CU'	SCO
<hr/>				
1 Houd	U27			

Dat gebeurt op niveau 3 van plaatje U27' onder voorwaarde dat de signalen LA', CU' en SCO laag zijn. Immers het moet alleen plaats vinden in de periode, dat het woord uit het A-register wordt verwerkt: (LA) = 1 en op het moment dat het U-register wordt ingelezen: (CU) = 1. Verder moet dit niet gebeuren bij de schuifopdrachten: (SCO) = 0. Op 2 cijfer-tijd, dus vlak voor het inlezen, wordt deze flip-flop teruggesteld met behulp van signaal 2ct, dat is aangesloten op beide houdniveau's van plaatje U27'. Waarom hier twee houdniveau's zijn wordt later besproken.

Bij de opdrachten 24 en 25 moet bij elke slag het A-register het cijfer (S25) overnemen. Dat cijfer wordt dan in flip-flop U₀ gezet,

U ₀ '/U ₀					
1 Houd	U ₀ '	CU			
5	U27	WTM	UIL'	LS'	DRO'
6	U27'	WTM'	UIL'	LS'	DRO'
7	S25'	OR22	UIL'	LA'	
8	TS'	OR22'	UIL'	LA'	DRO'
<hr/>					
1 Houd	U ₀				

waaruit het A-register het overneemt. De flip-flop U₀ leest het alleen in (met behulp van niveau 7 van het accentplaatje), wanneer het A-register wordt geselecteerd: (LA)=1, er een schuif naar links plaatsvindt: (UIL) = 1 en bij de opdrachten 24 en 25: (OR22) = 0. Bij de andere twee opdrachten (26 en 27) behoeft het A-register slechts (TS) over te nemen, aangezien het S-register alleen tekencijfers bevat. Niveau 8 van plaatje U₀' wordt laag als: (TS) = 1, wanneer het A-register is geselecteerd: (LA) = 1, er een schuif naar links plaatsvindt: (UIL) = 1, bij de opdrachten 26 en 27: (OR22) = 1 en als (DRO)=1.

Gedurende de periode, dat het S-register wordt geschoven, leest de flip-flop U₀ het quotiëntcijfer in, zodat deze op de minst significante plaats in het S-register komt te staan. Dat gebeurt dus als: (LS) = 1 en verder bij de schuif naar links: (UIL) = 1 en als geldt: (DRO) = 1. Het is reeds bekend, dat (U₀) = 1 moet worden, indien (U₂₇) = (WTM) is. Deze voorwaarden zijn uitgecodeerd op de niveau's 5 en 6 van plaatje U₀'.

Het coderend circuit OI selecteert de registers A en S op geheel overeenkomstige wijze als bij de vermenigvuldiging, dus met behulp van dezelfde niveau's van de plaatjes LA' en LS'.

Gedurende de eerste slag en gedurende de tijd, dat het S-register is geselecteerd bij elke volgende slag moet de opteller in de transportstand (stand 6) staan, omdat er alleen moet worden geschoven. Dat gebeurt met behulp van hetzelfde niveau als bij de vermenigvuldiging. Echter gedurende de tijd, dat het A-register is geselecteerd, moet de opteller de deler "normaal" bij het getal uit dit register optellen (stand 0), indien de tekens van de tussentijdse rest (U₂₇) en van de deler (TM) verschillen.

COIo'						
3	deel	U ₂₇	TM'	6ct'	STB'	OR ₂₄ '
4	deel	U ₂₇ '	TM	6ct'	STB'	OR ₂₄ '

In dat geval wordt niveau 3 of 4 van plaatje COIo' laag op 6 cijfertijs, mits geldt: (STB) = (OR₂₄) = 1. Zijn de tekens (U₂₇) en (TM) gelijk, dan moet de opteller de deler geïnverteerd

COI1'						
3	deel	U ₂₇	TM	6ct'	STB'	OR ₂₄ '
4	deel	U ₂₇ '	TM'	6ct'	STB'	OR ₂₄ '

bij de tussentijdse rest optellen (stand 1). Dit is uitgecodeerd op de niveau's 3 en 4 van plaatje COI1'. Deze niveau's worden op dezelfde cijfertijs en onder dezelfde verdere voorwaarden laag, als die van plaatje COIo'.

Bij de correctieslag wordt de opteller normaal op 6 cijfertijs in stand 0 of stand 1 gezet. Zijn de tekens van het oorspronkelijk deeltal (TS) en van de laatste rest (TA) gelijk, dan behoeft bij

COI6'

3 Correctie TA TS 7ct' STA STS OR24'
 4 Correctie TA' TS' 7ct' STA STS OR24'

die rest de deler niet meer te worden opgeteld en kan de opteller in de transportstand worden gezet. Op 7 cijfertijd wordt dan niveau 3 of 4 van plaatje COI6' laag, als dus (TA) = (TS) is en bij de laatste slag: (STA) = (STS) = 0 van de deling: (OR24) = 1. Kan één van deze niveau's niet laag worden, omdat de genoemde tekens ongelijk zijn, dan blijft de opteller dus in stand 0 resp. in stand 1 staan.

Tenslotte wordt de arithmetische controlering op dezelfde wijze als bij de vermenigvuldiging in stand 3 gezet met behulp van de signalen CCTR' en Co'. Ook deze laatste keer wordt het A-register eveneens op dezelfde wijze ingelezen als bij de vermenigvuldiging.

Schuifopdrachten (tijdschema)

De opdrachten 48 t/m 63 heten communicatie-opdrachten. Bij deze opdrachten wordt nooit een getal uit het geheugen gehaald, zodat het adresgedeelte van de opdracht daarvoor ook niet gebruikt hoeft te worden. De cijfers, die normaal voor de adressaanduiding worden gebruikt, kunnen nu dienen om het functiegedeelte der opdracht nader te specificeren.

Van deze communicatieopdrachten zullen nu eerst de schuifopdrachten worden besproken. Onderstaande tabel geeft de specificatie van de schuifopdrachten.

functie adres		links 54	rechts 55	links 62	rechts 63
0	n	I rond	I rond	II rond	II rond
1	n	I schoon	I schoon	II schoon	II schoon
2	n	III rond	III rond	IV rond	IV rond
3	n	III schoon	III schoon	IV schoon	IV schoon

De n in deze tabel kan de waarde 0 t/m 31 hebben en stelt het aantal plaatsen voor, dat men wil schuiven. Daarvoor zijn vijf binaire cijfers nodig. Om het getal n uit te drukken, worden de cijfer OR_0 t/m OR_4 gebruikt. De cijfers OR_5 en OR_6 dienen om de getallen in de eerste kolom uit te drukken. De overige cijfers van het adresgedeelte moeten bij de schuifopdrachten nul zijn. Hoewel de normeeropdrachten hierna pas worden behandeld willen we nu opmerken, dat deze opdrachten, die veel overeenkomst vertonen met de schuifopdrachten, van de laatstgenoemde worden onderscheiden door het cijfer (OR_7).

De Romeinse cijfers in de tabel slaan op de circuits waarin wordt geschoven. Circuit I omvat alleen het A-register en circuit II het S-register. Circuit III omvat de registers A en S, waarbij het A-register op de meest significante plaats staat en het S-register op de minst significante plaats. Van circuit IV, dat ook de beide registers A en S omvat, kan men zeggen, dat het S-register op de meest significante plaats staat en het A-register daarachter.

Het teken van het circuit is het teken van het register, dat op de meest significante plaats staat. Bij de circuits I en II is dat het teken van het register zelf.

Bij rondschiiven wordt de meest significante plaats aangevuld met het minst significante cijfer of omgekeerd respectievelijk bij rechts of links schuiven.

Bij een schone schuif blijft het teken van het circuit staan.

De vrijkomende plaatsen worden aangevuld met deze tekencijfers, terwijl de cijfers, die uit het circuit worden geschoven, verloren gaan. Het schuiven ontstaat door het woord, dat in het A-register (resp. S-register) staat via het arithmetisch orgaan in de somvormer te laten verschijnen en daarna één plaats naar links of rechts verschoven door het U-register over te laten nemen. Deze handeling wordt dan zoveel maal herhaald als er plaatsen geschoven moeten worden.

Voor we nu het schuiven in de registers behandelen, zullen we de tijdschema's op tek. XE23 verklaren. De opdrachtcyclus van de geheugencontroleling met alle bijbehorende signalen komt geheel overeen met hetgeen is behandeld op blz. 55-61 voor de ongemodificeerde opdrachten of met hetgeen is behandeld op blz. 84-85 voor de gemodificeerde opdrachten. Deze ring wordt daarna evenwel niet in stand 4 gezet, aangezien de getalcyclus hiervan vervalt en de teller daarom moet worden teruggesteld. Om nu te voorkomen, dat op 7gct de teller in stand 4 komt,

	GCR1'/GCR1					zoals reeds beschreven is op
1 Houd	GCR1'					blz. 60, wordt op 6gct, dus
<hr/>						1 cijfertijd eerder, niveau 2
1 Houd	GCR1					van plaatje GCR1 laag gemaakt,
2 3-0	GCRo	GCR2	6gct'	GG		omdat (GCRo) = (GCR2) = 0 en

daar geldt (GG) = ●. Het laatste signaal beduidt, dat er geen getalcyclus is van de geheugencontroleling en het niveau van plaatje GG' wordt hoog gemaakt met signaal CO, dat bij de communicatieopdrachten hoog is (blz. 63).

Het tijdsignaal 6gct' valt echter niet altijd samen met dezelfde "gewone" cijfertijd (2a). Bij de gemodificeerde opdrachten namelijk duurt de opdrachtcyclus vier cijfertijden langer dan die bij de ongemodificeerde opdrachten, zodat het signaal 6gct' nu samenvalt met cijfertijd 6b. Op 7gct neemt de opdrachtteller de inhoud van het U-register over en de inhoud van dit register mag dus niet eerder worden gewijzigd. Bij een gemodificeerde opdracht mag de schuifoperatie dus pas beginnen vier cijfertijden later dan bij de ongemodificeerde opdrachten. Een plaats schuiven duurt (in alle circuits) ook vier cijfertijden, zodat het onderscheid tussen de laatste gedeelten van de betreffende tijdschema's niet groot is.

Het onderscheid wordt aangegeven met de flip-flop BSO. Op cijfertijd 3a wordt signaal BAO' laag (blz. 68). Op het houdniveau van plaatje BSO (onaccent!) is signaal BAO (onaccent!) aangesloten,

BSO' / BSO			
1	Houd	BSO'	1ct
1	Houd	BSO	BAO
2	terug	5ct'	MB

waarmee dus de flip-flop in stand 1 wordt gezet op cijfertijd 3a. Bij de ongemodificeerde opdrachten: $(MB) = 0$ wordt deze flip-flop weer teruggesteld op 5 cijfertijd met behulp van niveau 2 op plaatje BSO.

Als $(MB) = 1$, dat is dus bij de gemodificeerde opdrachten, kan dit niveau niet laag worden en blijft de flip-flop in stand 1 staan tot op 1 cijfertijd het houdniveau van het accentplaatje hoog wordt gemaakt met signaal 1ct. Dat zijn dus juist de vier cijfertijden, die de opdrachtcyclus van de gemodificeerde opdrachten langer duurt.

COI6'		SCO'					
7	BAO' SCO'	1	CO'	OR8	OR9	CR10	NACO

Signaal BAO' maakt tezamen met signaal SCC' niveau 7 van plaatje COI6' laag. Daardoor wordt de opteller in de transportstand gezet, waarbij het interne register niet geïnverteerd wordt getransporteerd. Het signaal SCO' is laag, omdat nu: $(CC) = 1$ is, als de signalen OR8 t/m CR10 laag zijn en als signaal NACO laag is. Op het niveau van plaatje NACO' is o.a. signaal OR11' aangesloten. Als dus $(OR11) = 1$ is, zal signaal NACO laag zijn. De cijfers (OR8) t/m (OR11) zijn nul zowel bij de schuif- als bij de normeeropdrachten. Signaal SCO, dat is aangesloten op het niveau van plaatje COI7' en nu hoog is, voorkomt dat de opteller in stand 7 kan worden gezet (blz. 80).

Ook niveau 2 van plaatje LA' wordt laag gemaakt met signaal BAO', (blz. 68) tenminste wanneer er geschoven moet worden in de circuits I of III: $(OR24) = 0$. Daardoor wordt op de bekende wijze het A-register geselecteerd. Indien er in de circuits II of IV moet worden geschoven, maakt signaal BAO' niveau 2 van plaatje LS' laag (blz. 69), zodat het S-register wordt gekozen als: $(OR24) = 1$. Deze niveau's werken dus slechts eenmaal en wel aan het begin van de schuifoperatie. Bij schuiven in de circuits I en II is dit voldoende, daar het eenmaal gekozen register gedurende de verdere operatie geselecteerd moet blijven.

Echter bij schuiven in de circuits III en IV moeten het A-register en het S-register beurtelings worden geselecteerd.

LA'/LA					LS'/LS				
1	Houd	LS	LB	LOT	1	Houd	LA	LB	LOT
5	Schuif	SC'	37ct'	SAR'	5	Schuif	SC'	15ct'	SSR'
<hr/>					<hr/>				
1	LA				1	LS			

Daartoe wordt niveau 5 van plaatje LA' laag gemaakt op 3 en 7 cijfertijd met signaal 37ct', als de signalen SC' en SAR' laag zijn en niveau 5 van plaatje LS' op 1 en 5 cijfertijd met signaal 15ct' als de signalen SC' en SSR' laag zijn.

37ct			SC'		SAR		SSR				
1	3ct	7ct	1	SC0'	R2'	1	OR24'	OR6	1	OR24	OR6

Het signaal 37ct is de uitcodering van de cijfertijden 3 en 7. Gedurende cijfertijd 3 wordt het niveau van plaatje 37ct hoog gemaakt met signaal 3ct en gedurende cijfertijd 7 met signaal 7ct, Signaal 37ct' is dus laag gedurende deze beide cijfertijden en wel 2 omkeringen na het laagworden van signaal C1', evenals de signalen 3 en 7ct' (zie blz.27-29). Signaal 15ct' wordt op analoge wijze afgeleid. Signaal SC' is laag bij de schuifopdrachten: (SC0)=1 en als signaal R2' laag is. Dit laatste signaal eist dat de arithmetische controle-ring in stand 2 staat. Deze komt normaal in stand 2, zoals is beschreven op blz. 71. Signaal SAR' is hoog, wanneer de signalen OR24' en OR6 laag zijn. Deze beide signalen zijn de uitcodering van circuit II (zie tabel op blz. 135). Signaal LA' wordt dus niet laag, wanneer er wordt geschoven in circuit II. De signalen OR24 en OR6 zijn de uitcodering van circuit I en maken signaal SSR' hoog, zodat signaal LS' niet laag wordt bij schuiven in circuit I. Niveau 5 van plaatje LA' wordt ook regelmatig laag bij schuiven in circuit I, terwijl het multiplet dan reeds in de juiste stand is gezet met niveau 2 van dit plaatje. Dit verandert niets aan de stand van het multiplet en behoeft dus niet te worden voorkomen. Een overeenkomstige redenering geldt voor niveau 5 van plaatje LS' bij schuiven in circuit II.

CU'					
10	schuif	SC0'	BSO'	4ct'	MB
11	schuif	SC'	BSO'	0ct'	
8	schuif	SC'	SAR'	4ct'	BSO CCTR
9	schuif	SC'	SSR'	26ct'	BSO CCTR

Op 4 cijfertijd wordt signaal CU' laag, indien (MB) = 0 en (BSO) = (SCO) = 1 of op 0 cijfertijd, indien (BSO) = (SC) = 1 bij de gemodificeerde opdrachten. Het eerste geval is uitgecoördineerd op niveau 10 van plaatje CU' en het tweede op niveau 11. Kan niveau 10 niet werken, omdat (MB) = 1, dan kan niveau 11 wel laag worden, want die werkt enige cijfertijden later, zodat daarop niet gesteld behoeft te worden: (MB) = 1. Wel moet daarbij stand 2 van de arithmetische controlering in rekening worden gebracht (signaal SC'). De signalen CUIL en CUIR zijn laag,

CUIL'	CUIR'	CUI'
2 schuif SC' BSO OR21	2 schuif SC' BSO OR21	1 CUIL CUIR

omdat (BSO) = 1, zodat nu signaal CUI hoog is. Signaal UI' wordt laag, daar de signalen CU en CUI beide hoog zijn (blz. 58).

Wanneer nu nul plaatsen geschoven moet worden, zou de "schuif"-cyclus hiermede beëindigd moeten worden. Hierbij kan men natuurlijk niet spreken van een "schuif" in de ware zin des woords, maar in verband met de conditiesetting, die later zal worden besproken, kan deze "schuif" toch nuttig zijn. In dit geval kan direct hierna de arithmetische controlering in stand 3 worden gezet en wel met behulp van de signalen CCTR' en Co', dus op de bekende wijze.

Moeten er meer plaatsen geschoven, dan kan signaal CUIL' of signaal CUIR' laag worden, omdat flip-flop BS0 inmiddels is teruggezet: (BSO) = 0 en signaal SC' laag is. De keuze tussen links en rechts schuiven wordt bepaald met het cijfer (OR21). De signalen UIL' en UIR' ontstaan op dezelfde wijze, als bij de deling en de vermenigvuldiging (blz. 131 en 118). Signaal CU' wordt laag gemaakt met behulp van de niveau's 8 en 9 van dit plaatje. Op beide niveau's zijn de volgende signalen aangesloten: signaal SC', hetgeen ringstand 2 impliceert, signaal CCTR, dat in combinatie met deze ringstand 2 alleen deze niveau's laat werken vóór de overgang naar ringstand 3 en signaal BS0, dat deze niveau's hoog houdt, wanneer de niveau's 10 of 11 laag moeten worden. Verder zijn aangesloten op niveau 8 signaal O4ct',

dat een uitcodering is van de cijfertijden 0 en 4 en signaal SAR', dat dit niveau hoog houdt bij schuiven in circuit II. Op niveau 9 is signaal SSR' aangesloten, dat dit niveau niet laat werken, wanneer er geschoven moet worden in circuit I en verder nog signaal 26ct', dat laag is gedurende de cijfertijden 2 en 6. De beide genoemde combinatiecijfertijdsignalen ontstaan door analoge uitcodering, als bij het besproken signaal 37ct'. Zoals gezegd het schuiven eindigt door de arithmetische controlering met signaal CCTR' in stand 3 te zetten. De voorwaarden

		CCTR'/CCTR					
1	Houd	CCTR'	Co'				daarvoor zijn uitgecodeerd op de niveau's
9	Schuif	SC'	ES'	CES'	5ct'	OR7 GCR1	9 en 10 van plaatje
10	Schuif	SC'	ES'	CES	1ct'	OR7	CCTR'. Signaal SC' stelt o.a. de eis, dat deze niveau's
1	Houd	CCTR					alleen mogen werken als de arithmetische controlering in stand 2 staat (zie hierboven) en signaal OR7 houdt deze niveau's hoog bij de normeeropdrachten. Om één plaats te schuiven zijn vier cijfertijden beschikbaar (schuifslag). Gedurende één stand van de slagenteller kan het woord dus twee plaatsen worden geschoven, zodat de stand van de slagenteller aangeeft het aantal malen, dat het woord twee plaatsen is geschoven. De slagenteller staat b.v. in stand 2, wanneer er <u>vier</u> plaatsen zijn geschoven, maar ook wanneer het er <u>vijf</u> zijn geweest. (Nu wordt de slagenteller niet omgezet precies bij het einde van een schuifslag, maar op het moment dat het U-register wordt ingelezen voor een <u>even</u> schuifslag.) Wanneer nu een <u>oneven</u> aantal plaatsen moet worden geschoven, wordt dat met de signalen CES' of CES bepaald. Zoals gezegd, wordt het aantal te schuiven plaatsen vastgelegd in de cijfers (ORo) t/m (OR4) van de opdracht. Daarvan geeft het cijfer (ORo) aan of er een even of oneven aantal plaatsen moet worden geschoven en daarom is signaal ORo aangesloten op niveau 1 van plaatje CES'. Dit niveau moet kunnen werken bij de

alleen mogen werken als de arithmetische controlering in stand 2 staat (zie hierboven) en signaal OR7 houdt deze niveau's hoog bij de normeeropdrachten. Om één plaats te schuiven zijn vier cijfertijden beschikbaar (schuifslag). Gedurende één stand van de slagenteller kan het woord dus twee plaatsen worden geschoven, zodat de stand van de slagenteller aangeeft het aantal malen, dat het woord twee plaatsen is geschoven. De slagenteller staat b.v. in stand 2, wanneer er vier plaatsen zijn geschoven, maar ook wanneer het er vijf zijn geweest. (Nu wordt de slagenteller niet omgezet precies bij het einde van een schuifslag, maar op het moment dat het U-register wordt ingelezen voor een even schuifslag.) Wanneer nu een oneven aantal plaatsen moet worden geschoven, wordt dat met de signalen CES' of CES bepaald. Zoals gezegd, wordt het aantal te schuiven plaatsen vastgelegd in de cijfers (ORo) t/m (OR4) van de opdracht. Daarvan geeft het cijfer (ORo) aan of er een even of oneven aantal plaatsen moet worden geschoven en daarom is signaal ORo aangesloten op niveau 1 van plaatje CES'. Dit niveau moet kunnen werken bij de

1	ORo	MB	ongemodificeerde opdrachten: (MB) = 0 en wordt laag als (ORo) = 0. In dit geval kan niveau 9 van plaatje CCTR' laag worden op 5 cijfertijd en komt de arithmetische controlering in stand 3 op
2	ORo'	MB'	

6 cijfertijd, mits ook is voldaan aan de voorwaarde $(ES) = 1$. Was $(ORo) = 1$, dan wordt niveau 1 van plaatje CES' niet laag en blijft niveau 9 van plaatje CCTR' hoog, maar nu kan niveau 10 van dit plaatje laag worden op 1 cijfertijd onder overigens dezelfde voorwaarden. Dat is dus vier cijfertijden later en in die tijd kan nog één plaats worden geschoven, zodat nu het aantal geschoven plaatsen oneven is geworden.

Zoals reeds werd opgemerkt gebeurt alles bij de gemodificeerde opdrachten vier cijfertijden later dan bij de ongemodificeerde. Daarom wordt nu niveau 10 van plaatje CCTR' laag gemaakt op 1 cijfertijd, indien $(ORo) = 0$ is, dus bij een even aantal plaatsen schuiven of niveau 9 op 5 cijfertijd, indien $(ORo) = 1$ is, dus als het woord een oneven aantal plaatsen geschoven moet worden. Het kiezen tussen de niveau's 9 en 10 gebeurt weer met behulp van de signalen CES' of CES. Daartoe wordt nu niveau 2 van plaatje CES' laag gemaakt met signaal ORo' , mits geldt $(MB) = 1$. Het signaal $GCR1$, dat is aangesloten op niveau 9 van plaatje CCTR', dient om laag gaan van dit niveau te voorkomen in het gemodificeerde geval, gedurende de tijd, dat de geheugen controlering nog in stand 3 staat en de arithmetische controlering tegelijkertijd al in stand 2 staat.

De slagenteller wordt in de volgende stand gezet met de signalen $DSTo'$ en $DST1'$ en wel op 4 cijfertijd met behulp van dezelfde niveau's als die, welke bij de vermenigvuldiging en deling worden gebruikt (blz. 114). Het signaal SDO' , dat hierbij een rol speelt, wordt nu echter laag gemaakt met niveau 2 van plaatje SDO' .

SDO'	DSTo'	DST1'
2 SC' MB	2 SC' Oct' b	2 SC' Oct' b'

Dit niveau mag alleen werken bij de ongemodificeerde opdrachten: $(MB) = 0$ en als signaal SC' laag is. In het geval van een gemodificeerde opdracht moeten ook de signalen $DSTo'$ en $DST1'$ vier cijfertijden later komen. Daarom worden genoemde signalen nu laag gemaakt op 0 cijfertijd met behulp van de niveau's 2 op deze plaatjes. Signaal SC' moet daarbij laag zijn, terwijl de signalen b' en b dezelfde functie hebben als de signalen a' en a op de 1e niveau's.

ES			Signaal ES' wordt laag, wanneer de acht niveau's van plaatje ES alle hoog zijn. Op dit plaatje wordt de stand van de slagenteller vergeleken met de cijfers (OR1) t/m (OR4) en alle acht niveau's zijn hoog, indien beide getallen gelijk zijn.
1	OR1'	STo	Zoals is opgemerkt, geeft de slagenteller aan, het aantal malen, dat het woord <u>twee</u> plaatsen is verschoven, terwijl het getal, dat wordt gevormd door de cijfers (OR1) t/m (OR4) aangeeft het aantal malen, dat er twee plaatsen <u>moet worden</u> geschoven.
2	OR1	STo'	
3	OR2'	ST1	
4	OR2	ST1'	
5	OR3'	ST2	
6	OR3	ST2'	
7	OR4'	ST3	
8	OR4	ST3'	

Beschouwen we bijvoorbeeld de niveau's 1 en 2, dan blijkt dat niveau 1 hoog is tengevolge van signaal OR1', als $(OR1) = 0$ is en niveau 2 tengevolge van signaal STo', als $(STo) = 0$ is. Een gelijke redenering geldt voor de overige niveau's. Door combinatie van de signalen ES en CES wordt de arithmetische controlering dus na het juiste aantal geschoven plaatsen in stand 3 gezet en daarmee de schuif beëindigd. De tijdsduur van deze opdrachten is dus afhankelijk van het aantal te schuiven plaatsen.

Signaal CST' ontstaat op dezelfde wijze als bij de vermenigvuldiging en de deling (blz. 116) en stelt de slagenteller terug. Bij schuiven in de circuits I, III en IV moet signaal AI' regelmatig laag worden, zodat het A-register het woord uit het U-register overneemt en wel op 1 en 5 cijfertijd. Daarvoor wordt op de besproken wijze signaal 15ct' gemaakt met behulp van de signalen 1ct en 5ct.

AI'				Dit signaal is aangesloten op niveau 6 van plaatje AI', evenals signaal STS, dat laag is als de slagenteller <u>niet</u> in stand 0 staat. Verder zijn op dit niveau de signalen SAR' en SC' aangesloten, die hiervoor reeds zijn besproken. Niveau 6 van plaatje AI' kan dus alleen werken bij deze opdrachten, nadat de slagenteller in stand 1 is gekomen en totdat de arithmetische controlering in stand 3 komt. Maar vóór de slagenteller in stand 1 komt moet signaal AI' ook al een keer laag worden en wel op 1 cijfertijd bij de ongemodificeerde opdrachten: $(MB) = 0$,
6	SC'	15ct'	SAR' STS	
5	SC'	1ct'	SAR' MB	
9	SC'	5ct'	a' SAR' MB'	

als de signalen SC' en SAR' laag zijn. Dit is uitgecodeerd op niveau 5 van plaatje AI'. Dit niveau kan dus ook laag worden tegelijkertijd met niveau 6 en wel iedere keer op 1 cijfertijd mits aan de overige voorwaarden is voldaan. Dit behoeft echter niet te worden voorkomen. Ook bij de gemodificeerde opdrachten: (MB) = 1 moet signaal AI' laag worden vóór de slagenteller in stand 1 komt, namelijk op cijfertijd 5a' en weer mits de signalen SC' en SAR' laag zijn. Deze signalen zijn aangesloten op niveau 9 van plaatje AI'. Dit niveau kan ook weer laag worden tegelijkertijd met niveau 6, mits aan de overige voorwaarden is voldaan. Ook nu behoeft het niet te worden voorkomen.

SI'	Signaal SI' moet laag worden bij
5 SC' 37ct' SSR' BSO	schuiven in de circuits II, III en
	IV op de cijfertijden 3 en 7.

Daartoe wordt signaal 37ct' gemaakt met de signalen 3ct en 7ct, welk signaal is aangesloten op niveau 5 van plaatje SI' en dit niveau laag maakt op de genoemde cijfertijden. De overige voorwaarden waaronder dit niveau laag moet worden, zijn uitgecodeerd met de volgende signalen: signaal SC', dat reeds is besproken, signaal SSR', dat de werking van dit niveau uitsluit bij schuiven in circuit I en signaal BSO, dat de werking van dit niveau pas laat beginnen, nadat de flip-flop BSO is teruggesteld.

In verband met de conditie-zetting (wordt later behandeld) is het noodzakelijk, dat bij het eind van de schuif het woord uit het meest significante register in het U-register staat. Bij schuiven in de circuits I en II is dat eenvoudig, aangezien daarbij slechts sprake is van één register. Maar bij schuiven in circuit III staat het A-register op de meest significante plaats en bij circuit III is het woord uit het A-register b.v. op cijfertijd 4a (verschoven) in het U-register gezet, waarna op cijfertijd 5a het A-register dit woord weer overneemt. Het woord uit het meest significante register blijft dus in het U-register staan, totdat het wordt schoongemaakt. Echter bij schuiven in circuit IV staat het woord, dat in het A-register hoort eveneens het laatst in het U-register. Nu moet daarin het woord uit het S-register komen, aangezien dat het meest significante is. Het S-register is reeds

geselecteerd. Wordt nu signaal CU hoog en signaal UI' laag, dan neemt daardoor het U-register het woord uit het S-register (niet verschoven) over. Nadat de arithmetische controlering in stand 3 is gekomen, wordt altijd al niveau 3 van plaatje CU' laag (zie blz. 72). Dat gebeurt dus ook bij de opdracht tot schuiven in circuit IV. Bij schuiven in circuit III echter moet het worden voorkomen, omdat het juiste woord reeds in het U-register staat.

AS'	Daartoe is signaal AS op genoemd niveau aange-
1 SCO' OR24	sloten. Signaal AS is hoog, zolang de signalen
	SCO' en OR24 laag zijn, dat is dus bij de op-
	drachten tot schuiven in de circuits I en III.

Het schuiven in de registers

In het voorgaande is reeds opgemerkt, dat bij rondschiiven naar links op de minst significante plaats het meest significante cijfer wordt gezet, terwijl bij rondschiiven naar rechts op de meest significante plaats het minst significante cijfer wordt gezet.

Bij schoon schuiven, zowel links als rechts blijft het teken van het circuit onveranderd (het circuit wordt als het ware tussen het tekencijfer en het daarnaast staande cijfer doorgeknipt) en de vrijkomende plaatsen worden aangevuld met deze tekencijfers. De cijfers, die uit het circuit worden geschoven, gaan verloren. Een en ander is nader toegelicht in fig. 34.(blz.146) Het schuiven in circuit II is niet getekend, aangezien dat geheel overeenkomt met het geheuren in circuit I, alleen de daarbij betrokken signalen heten anders.

Zoals eveneens reeds is opgemerkt ontstaat het schuiven door het woord uit het A-register (resp. S-register) via het arithmetisch orgaan in de somvormer te laten verschijnen, waarna het woord één plaats verschoven naar links of rechts door het U-register over wordt genomen. Dan geldt dus gedurende het schuiven van het A-register (in het vervolg te noemen A-schuif): $(A_0) = (S_{v0})$, $(A_1) = (S_{v1}) \dots (A_n) = (S_{v26})$.

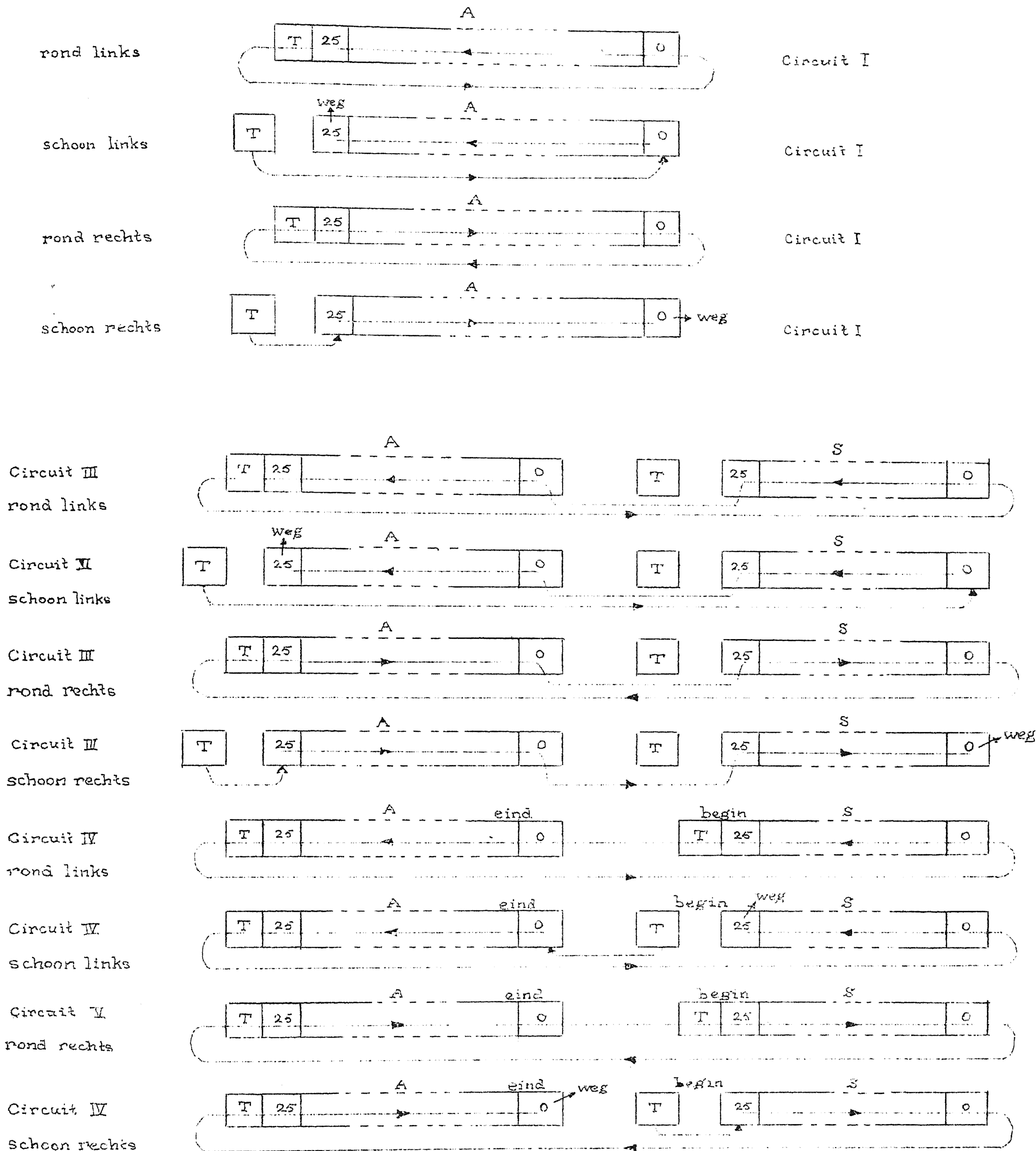
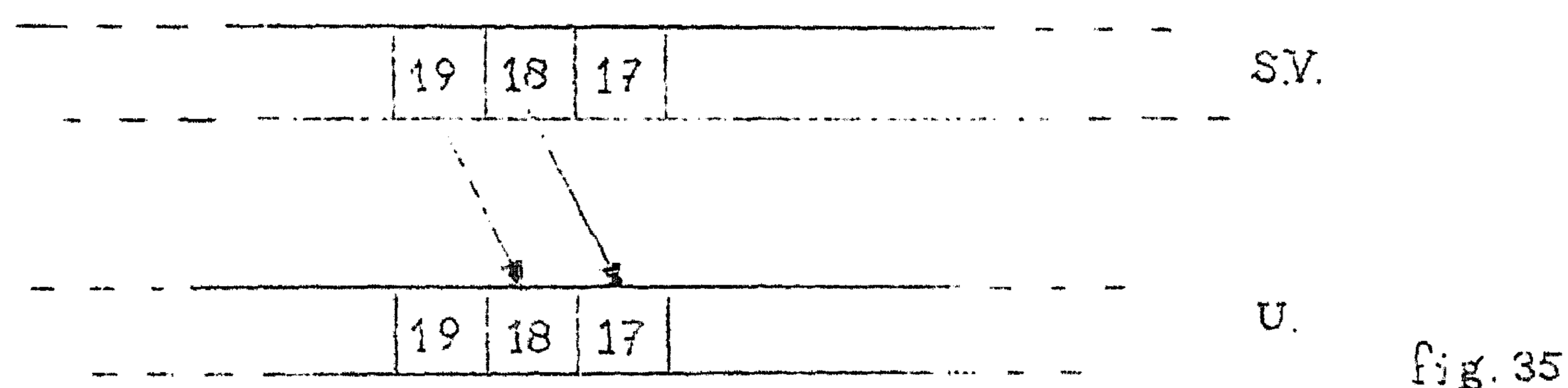


fig. 34.

In het middengedeelte van het U-register neemt een flip-flop niet het overeenkomende cijfer uit de somvormer over, maar het cijfer, dat daarnaast staat. Bij schuiven naar rechts neemt b.v. flip-flop U18 het cijfer (SV19) over en flip-flop U17, het cijfer (SV18) enz. (zie fig.35). Bij schuiven naar links neemt flip-flop U18



het cijfer (SV17) over enz. Dit is hetzelfde als bij de vermenigvuldiging en deling en vereist geen bijzondere voorzieningen. Een voorbeeld van de uitcodering op de niveau's is gegeven op blz. 44. Voor de flip-flops aan de uiteinden van de registers moeten wel bijzondere maatregelen worden getroffen. Aangezien de over te nemen cijfers eenvoudig zijn af te leiden uit fig. 34, zullen we nu de verschillende gevallen in tabelvorm geven en daarmee de werking der niveau's verklaren. Het zal daarbij blijken dat in sommige gevallen meer dan één niveau van een plaatje werkt. Zoals bekend is, levert dat geen bezwaren op.

In het hierna volgende gedeelte dient men te bedenken, dat in de circuits III en IV steeds eerst het S-register: $(LS) = 1$ en daarna het A-register: $(LA) = 1$ wordt geschoven. Nu kan het U-register gedurende de S-schuif het overlopende cijfer uit het A-register direct daaruit overnemen, maar gedurende de A-schuif zou het cijfer, dat uit het S-register moet worden overgenomen, reeds verloren zijn. Daarom wordt dit cijfer bewaard in de extra flip-flop U27. Deze flip-flop wordt dus ingelezen onder voorwaarde dat $(LS) = 1$ is. Ook moet dan gelden: $(SCO) = 1$. Of flip-flop U27 ook wordt ingelezen bij schuiven in de circuits I en II is onbelangrijk. Deze flip-flop heeft twee houdniveau's op het accentplaatje, waarvan het tweede constant hoog wordt gehouden met signaal SCO, zodat alleen het eerste houdniveau zijn functie uit kan oefenen. Dit houdniveau wordt hooggemaakt op 2 en 6 cijfertijd,

		U27'/U27					
1	Houd	U27'	6ct	2ct			dat zijn de cijfertijden
2	Houd	U27'	SCO	2ct			volgende op die, waarop
4		SV26'	UIL'	LS'	SCO'	OR24'	het A-register de inhoud
5		SV25'	UIL'	LS'	SCO'	OR24	van het U-register over-
6		SVo'	UIR'	LS'	SCO'		neemt (1 en 5 cijfertijd),
<hr/>							zodat dan ook (U27) wordt
1	Houd	U27					overgenomen. Het inlezen
							van flip-flop U27

gebeurt met behulp van het signaal UIL' of UIR' en deze signalen worden laag drie omkeringen nadat de signalen 2ct of 6ct hoog werden. De flip-flop is dus zeker teruggesteld voor het opnieuw inlezen.

In onderstaande tabel zijn de cijfers gegeven, die in flip-flop U27 moeten worden gezet. Achter elk geval staan de niveau's, die daarop betrekking hebben.

		rond			schoon		
links in III	SV25'	LS'	(5)	SV25'	LS'	(5)	
in IV	SV26'	LS'	(4)	SV26'	LS'	(4)	
rechts in III	SVo'	LS'	(6)	onbelangrijk			
in IV	SVo'	LS'	(6)	SVo'	LS'	(6)	

De niveau's 4 en 5 voor de circuits IV en III worden onderscheiden door het cijfer (OR24).

		Uo'/Uo					
1	Houd	Uo'	CU				Gedurende de A-schuif:
3		SV26'	UIL'	SCO'	OR6		(LA) = 1 in circuit III
4		SV1'	UIR'				moet flip-flop Uo het
9		TA'	UIL'	SCO'	OR6'	LS'	cijfer (S25) overnemen,
10		U27'	UIL'	SCO'	OR6'	LA'	dat van te voren in de
<hr/>							flip-flop U27 is gezet
1	Houd	Uo					en bij schuiven in circuit
							IV het teken van het S-
							register, dat dan eveneens reeds in de flip-flop U27 staat.

	rond	schoon
rechts in I,II,III,IV	SV1' (4)	SV1' (4)
links in I,II	SV26' (3)	SV26' (3)
in III,IV	TA' LS' (9)	TA' LS' (9)
	U27' LA' (10)	U27' LA' (10)

Het cijfer (OR6) geeft het onderscheid tussen de circuits I en II en de circuits III en IV.

De flip-flop U25 moet de volgende cijfers overnemen:

	rond	schoon
links in I,II,III,IV	SV24' (3)	SV24' (3)
rechts in I	SV26' (5)	SV26' (5)
II	SV26' (4)	SV26' (4)
III	Ao' LS' (6)	Ao' LS' (6)
	SV26' LA' (5)	SV26' LA' (5)
IV	SV26' (LS')(4)	SV26' (LS')(4)
	SV26' (LA')(4+5)	SV26' (LA')(4+5)

		U25'/U25			
1	Houd	U25'	CU		
3		SV24'	UIL'		
4		SV26'	UIR'	SCO'	OR24'
5		SV26'	UIR'	LA'	
6		Ao'	UIR'	LS'	OR24 SCO'
1	Houd	U25			

Het cijfer (OR24) dient weer ter onderscheiding tussen de circuits III en IV. De signalen LA' en LS' in de tabel bij IV zijn tussen haakjes geplaatst, aangezien deze hierbij onbelangrijk

zijn. Ze zijn alleen vermeld om de werking van de niveau's aan te geven.

De flip-flop U26 heeft vrij veel niveau's in verband met het feit, dat verschillende gevallen zeer speciaal moeten worden uitgecodeerd. De functies van de cijfers (OR6) en OR24) zijn reeds bekend. De niveau's 11,12 en 13 moeten laag kunnen worden, zowel bij het links- als bij het rechtsschuiven. Daarom zijn daarop de signalen CU' en SC' aangesloten.

		rond		schoon	
links	in I	SV25'	(5)	SV26'	(13)
	II	SV25'	(4)	SV26'	(11+12)
	III	SV26' LS'	(11)	SV26' LS'	(11+12+13)
		SV25' LA'	(5)	SV26' LA'	(13)
	IV	SV25' LS'	(4)	SV26' LS'	(12)
		SV25' LA'	(3+4+5)	SV25' LA'	(3)
rechts	in I	SVo'	(6)	SV26'	(13)
	II	SVo'	(6)	SV26'	(11+12)
	III	SV26' LS'	(11)	SV26' LS'	(11+12+13)
		U27' LA'	(16)	SV26' LA'	(13)
	IV	Ao' LS'	(14)	SV26' LS'	(12)
		U27' LA'	(15+16)	U27' LA'	(15)

U26' / U26

1	Houd	U26'	CU				
3		SV25'	UIL'	SCO'	LA'	OR24'	
4		SV25'	UIL'	SCO'	OR5	OR24'	
5		SV25'	UIL'	SCO'	LA'	OR5	
11		SV26'	CU'	SC'	LS'	OR24	
12		SV26'	CU'	SC'	LS'	OR5'	
13		SV26'	CU'	SC'	OR5'	OR24	
14		Ao'	UIR'	SCO'	LS'	OR5	OR6' OR24'
15		U27'	UIR'	SCO'	LA'	OR24'	
16		U27'	UIR'	SCO'	LA'	OR5	OR6'
6		SVo'	UIR'	SCO'	OR5	OR6	

11	Houd	U26					
----	------	-----	--	--	--	--	--

Het cijfer (OR5) geeft het onderscheid tussen schoon- en rond-schuiven.

Normmeeropdrachten

Wanneer b.v. het register A genormeerd wordt, wil dat zeggen, dat de inhoud van het register zo lang naar links wordt geschoven, tot het meest significante cijfer verschilt van het teken van dit register. Normeren is alleen mogelijk in de (schuif-) circuits I, II en III. Wanneer het getal, dat als "lang" woord in de registers A en S staat (circuit III), wordt genormeerd, is het teken van het circuit bepalend. Dat geldt natuurlijk ook voor de circuits I en II, maar daar is het teken van het circuit tevens het teken van het register. Zoals bij de schuifopdrachten reeds is gezegd, worden deze van de normmeeropdrachten onderscheiden door het cijfer (OR7). Aangezien normeren alleen mogelijk is met schoon links schuiven, zijn ook alleen die schuifopdrachten, uitgebreid met het cijfer (OR7), mogelijk.

Adres		functie	
		54	62
5	0	Normeer (A) $n \Rightarrow B$	Normeer (S) $n \Rightarrow B$
7	0	Normeer (AS) $n \Rightarrow B$	

Het aantal geschoven plaatsen, dat nodig is om aan het gestelde te voldoen (n), wordt genoteerd in het B-register. Aan het eind van de normering wordt daartoe de stand van de slagenteller in de flip-flops B1 t/m B5 gezet. Daarmee staat het aantal malen, dat er twee plaatsen is geschoven in het B-register. Vervolgens wordt bij iedere oneven schuif de flip-flop B0 in stand 1 gezet, zodat het getal in het B-register nu het juiste aantal geschoven plaatsen geeft.

B_0'/B_0

- 1 Houd B_0' BI
- 2 Lees U_0' BI'
- 3 Norm. SC' OR7' 7ct' MB
- 4 Norm. SC' OR7' 3ct' MB'

1 Houd B_0

is behandeld (blz. 137). Deze niveau's mogen alleen werken als de arithmetische controlering in stand 2 staat bij de normeeropdrachten: $(SC) = (OR7) = 1$.

Op 5 cijfertijd als: $(MB) = 0$ of op 1 cijfertijd als: $(MB) = 1$, dus vlak voor het inlezen van flip-flop B_0 , wordt niveau 2, resp. 3 van plaatje BI' laag gemaakt, weer onder de voorwaarde:

 BI'/BI

- 2 Norm. SC' OR7' 5ct' MB
- 3 Norm. SC' OR7' 1ct' MB'

1 BI
2 SC0'

Dat gebeurt op 7 cijfertijd bij de ongemodificeerde opdrachten: $(MB) = 0$ en op 3 cijfertijd bij de gemodificeerde opdrachten: $(MB) = 1$. Het verschil is weer vier cijfertijden, zoals bij de schuifopdrachten reeds uitvoerig

$(SC) = (OR7) = 1$. Daardoor wordt signaal BI hoog, zodat het gehele B-register wordt schoongemaakt. Tevens is niveau 2 van plaatje BI laag door signaal SC0', zodat het B-register niet, zoals

gewoonlijk, het woord uit het U-register over kan nemen.

Gedurende stand 2 van de arithmetische controlering kan dus flip-flop B_0 iedere keer worden ingelezen (zie ook tek. XE 23), terwijl de flip-flops B_1 t/m B_5 worden ingelezen gedurende de stand 3 van dezelfde ring met behulp van signaal BIN. Dit signaal wordt laag bij de normeeropdrachten: $(SC0) = (OR7) = 1$,

BIN'

als signaal R3' laag is.

- 1 R3' SC0' OR7' Tevens is het signaal 1ct' op de inleesniveau's aangesloten, zodat op de eerst-

volgende 1 cijfertijd nadat de arithmetische controlering in stand 3 is gekomen, het juiste aantal geschoven plaatsen in het B-register wordt gezet.

B1'/B1

1 Houd B1' BI
 2 Lees U1' BI'
 3 Norm. BIN' 1ct' ST0'

1 Houd B1

B3'/B3

1 Houd B3' BI
 2 Lees U3' BI'
 3 Norm. BIN' 1ct' ST2'

1 Houd B3

B5'/B5

1 Houd B5' BI
 2 Lees U5' BI'
 3 Norm. BIN' 1ct' ST4'

1 Houd B5

CCTR'/CCTR

1 Houd CCTR' Co'
 11 Norm. U26 U25' 15ct' SC' OR7' BSO
 12 Norm. U26' U25 15ct' SC' OR7' BSO
 14 Norm. ST0' ST2' ST3' 15ct' SC' OR7' OR6
 13 Norm. ST1' ST3' ST4' 15ct' SC'

1 Houd CCTR

B2'/B2

1 Houd B2' BI
 2 Lees U2' BI'
 3 Norm. BIN' 1ct' ST1'

1 Houd B2

B4'/B4

1 Houd B4' BI
 2 Lees U4' BI'
 3 Norm. BIN' 1ct' ST3'

1 Houd B4

Het einde van de schuif en dus ook van de normering wordt bewerkstelligd door signaal CCTR'. De niveau's 9 en 10 van plaatje CCTR' kunnen nu niet laag worden, omdat daarop signaal OR7 is aangesloten (zie blz. 141). De

niveau's 11 en 12 kunnen echter laag worden op de cijfertijden 1 en 5: signaal 15ct' is laag, bij de normeer-

opdrachten: (SC) = (OR7) = 1, mits de flip-flop BSO is teruggesteld: (BSO) = 0 en verder op voorwaarde, dat het tekencijfer (U26) ongelijk is aan het meest significante getalcijfer (U25). Nu is het ook mogelijk, dat het getal nul (+0 of -0) wordt genormeerd. Dan zullen de niveau's 11 en 12 nooit laag kunnen worden, eenvoudig omdat geen enkel getalcijfer verschilt van het tekencijfer. Bij normering in de circuits I en II wordt het schuiven gestopt op 1 of 5 cijfertijd nadat de slagenteller in stand 13 is gekomen, d.w.z. nadat het woord 26 plaatsen is verschoven. Bij normeren in circuit III wordt het schuiven gestopt

eveneens op 1 of 5 cijfertijd, nadat de slagenteller in stand 26 is gekomen, m.a.w. nadat het woord 52 plaatsen is verschoven. Deze standen van de slagenteller zijn uitgecodeerd op niveau 14 resp. 13. Verder zijn op niveau 14 aangesloten signaal SC', reden bekend, de signalen OR7' en OR6 om dit niveau alleen te laten werken bij de normeeropdrachten wel voor de korte circuits. Op niveau 13 zijn aangesloten signaal SC' en signaal OR6' voor het lange circuit. Hierbij is het niet nodig ook signaal OR7' aan te sluiten, aangezien bij een "gewone" schuif niet meer dan 31 plaatsen geschoven kan worden in verband met de beschikbare cijfers (OR0) t/m (OR4). Dat deze niveau's zowel op 1 als op 5 cijfertijd laag kunnen worden hangt weer samen met de gemodificeerde en ongemodificeerde opdrachten. Signaal CCTR' tezamen met signaal Co' zetten de arithmetische controlering in stand 3 op de bekende wijze (zie blz. 72).

Snelle vermenigvuldiging met tien (Maal tien)

Ten behoeve van een snelle omzetting van 10-tallig- naar 2-tallig stelsel en omgekeerd zijn er twee speciale opdrachten. Door van deze opdrachten gebruik te maken kan het getal, dat in het S-register staat met tien worden vermenigvuldigd. Deze bewerking duurt 64 μ sec, hetgeen aanmerkelijk korter is dan wanneer van de normale vermenigvuldigopdracht gebruik wordt gemaakt (500 μ sec). Ook de maal-tienopdracht is weer een gespecificeerde communicatieopdracht (opdracht 62), terwijl het "adresgedeelte" weer in twee delen is gesplitst en de opdracht nader specificeert. De tien meest significante cijfers daarvan vormen het getal 32: (OR10)=1 en de vijf minst significante cijfers het getal 0 of 1. Deze opdracht wordt als volgt genoteerd: 62/32/0 of 62/32/1. Is het laatste getal 0, dan wordt het tienvoud van de inhoud van het S-register als "lang woord" in de registers A en S gezet. Is het getal: 1, dan komt het tienvoud van het getal in het S-register ook weer in dit register zelf te staan. Cijfers, die eventueel aan de meest significante zijde uit dit register lopen, gaan verloren, terwijl de inhoud van het A-register onaangetast blijft.

De vermenigvuldiging vindt plaats in vier stappen . De inhoud van het S-register wordt bij de 1e stap een plaats naar links rondgeschoven en met behulp van signaal CM' in het M-register gezet. In dit register staat nu dus $2 \times (S)$. Bij de tweede stap moet de opteller de inhoud van register M transporteren (stand 2) en weer een plaats naar links rondschiiven. Nu staat in het M-register $4 \times (S)$. Bij de derde stap wordt de inhoud van het M-register opgeteld bij de inhoud van het S-register (opteller in stand 0), zodat nu $5 \times (S)$ in de somvormer verschijnt. Echter voor deze optelling plaatsvond, zijn de twee minst significante cijfers van het S-register gelijk gemaakt aan (TS), zodat de twee minst significante cijfers van het M-register onveranderd in de somvormer verschijnen. Deze beide cijfers zijn nu tengevolge van het rondschiiven de twee meest significante cijfers van het oorspronkelijke getal. Tenslotte wordt het laatste woord uit de somvormer een plaats naar links verschoven door het U-register overgenomen en daarvandaan in het S-register gezet. Indien het resultaat in de registers A en S moet komen, neemt het S-register weer het woord uit het U-register over, terwijl het A-register de betreffende cijfers direct uit de somvormer overneemt.

Evenals bij de "gewone" schuif geven de flip-flops in het middengedeelte van het U-register geen moeilijkheden, terwijl voor de flip-flops aan de uiteinden van dit register bijzondere voorzieningen moeten worden getroffen. De twee minst en de twee meest significante cijfers van de somvormer worden direct in de vier minst significante flip-flops van het A-register gezet. Slechts vier, omdat vermenigvuldiging in het tweetallig stelsel met tien (1010) een getal uitbreidt met vier binaire cijfers. De overige cijfers van de somvormer worden een plaats naar links geschoven door het U-register overgenomen met uitzondering van de flip-flops U_0 t/m U_2 . Voor deze flip-flops moet een speciale voorziening worden getroffen, hetgeen we zullen aantonen met behulp van fig. 36.

$$\begin{array}{r}
 \text{---} \quad \boxed{3} \boxed{2} \boxed{1} \boxed{0} \quad S \\
 \text{---} \\
 \hline
 1 \ 0 \ 1 \ 0 \times 10 \\
 \\
 \begin{array}{r}
 3 \ 2 \ 1 \ 0 \ T \ T \ T \\
 \hline
 3 \ 2 \ 1 \ 0 \ T \ + \\
 1 \ 0 \ T
 \end{array}
 \end{array}$$

FIG. 36

Normale vermenigvuldiging met tien in het tweetallig stelsel (op papier) ontstaat door: het vermenigvuldigtal drie plaatsen naar links te schuiven en aan te vullen met tekencijfers; daarna het vermenigvuldigtal een plaats naar links verschoven en aangevuld met een tekencijfer daaronder te schrijven en tenslotte de beide verkregen getallen op te tellen (zie fig. 36). Het blijkt, dat de drie minst significante cijfers van tevoren reeds eenvoudig zijn te bepalen. De flip-flop U_6 moet dus (TS) overnemen en de flip-flops U_1 en U_2 de cijfers (S₀) resp. (S₁). Deze cijfers zijn in het S-register echter niet meer beschikbaar, maar nog wel in de flip-flops M₂ resp. M₃. Daarom neemt flip-flop U_1 het cijfer (M₂) over en flip-flop U_2 het cijfer (M₃). Flip-flop U_{26} neemt (TS) regelrecht over. Het gehele verloop van de snelle vermenigvuldiging met tien is nader toegelicht in fig. 37.



Tijdschema van de snelle vermenigvuldiging met tien

De opdrachtcyclus van de gemodificeerde opdracht duurt, zoals in het voorgaande reeds is behandeld, vier cijfertijden langer dan die van de ongemodificeerde opdracht. Het laatste gedeelte van beide versies is gelijk, aangezien na de opdrachtcyclus van de ongemodificeerde opdracht vier cijfertijden wordt gewacht (zie tek. X1-E31).

De eerste keer dat de inhoud van het S-register een plaats

CU'	verschoven door het U-register wordt over-
5 Oct' b MTA'	genomen, is op cijfertijd 5b. Daartoe wordt
7 Oct' MTA'	niveau 5 van plaatje CU' laag gemaakt met
	de signalen Oct' en b, als signaal MTA' laag

is. Dit laatste signaal is de uitcodering van stand 2 der arithmetische controlering: (R2) = 1 bij deze opdrachten: (CO) = (OR10) = 1.

MTA'	tische controlering: (R2) = 1 bij deze
1 R2' CO' OR10'	opdrachten: (CO) = (OR10) = 1.

Verder wordt niveau 7 van plaatje CU' laag als geldt: (MTA) = 1 op 3 cijfertijd en tenslotte niveau 3 (zie blz. 72) op 1 cijfertijd wanneer de arithmetische controlering reeds in stand 3 staat: (R3) = 1. Ook signaal CUIL' moet laag worden om het links schuiven

CUIL'	tot stand te brengen en wel in de beide
3 ACR1' CO' OR10'	standen 2 en 3 van de arithmetische controlering: (ACR) = 1 bij deze opdrachten (CO) = (OR10) = 1. Op de cijfertijd 1b en 4a' moet het M-register het woord uit het U-register overnemen. Dan moet signaal CM' laag worden en wel alleen bij deze opdracht in ringstand 2: (MTA) = 1.

Tengevolge van de aard der opdracht (niet additieve uit-opdracht) is de opteller in stand 6 geplaatst op cijfertijd 3a (zie blz.79). Op cijfertijd 1b moet de opteller in stand 2 worden gezet, zodat het M-register wordt getransporteerd: $(CO12) = 1$.

COI2'	COI0'
3 1ct' b MTA' CCTR	7 4ct' a' MTA'

Op cijfertijd 4a' wordt de opteller in de optelstand gezet: $(COI0) = 1$ voor de 3e stap. Echter voor deze stap wordt gedaan, moeten de beide minst significante flip-flops van het S-register het tekencijfer (TS) overnemen. Dit wordt gedaan met behulp van de signalen CS0 en CS1. Signaal CS0 wordt hoog doordat het niveau van plaatje CS0' laag wordt op 2 cijfertijd als $(TS) = 0$ is, mits signaal MTA' laag is. Dit signaal is aangesloten op de houdniveau's der accentplaatjes van de twee genoemde flip-flops.

CS0'	CS1'
1 TS 2ct' MTA'	1 TS' 2ct' MTA'

Signaal CS1 wordt hoog door het laag worden van het niveau van plaatje CS1' op 2 cijfertijd als $(TS) = 1$ is en weer mits signaal MTA' laag is.

De arithmetische controlering wordt normaal in stand 3 gezet met behulp van signaal CCTR'. Daartoe wordt niveau 16 van plaatje CCTR' laag gemaakt op cijfertijd 1b', mits signaal MTA' laag is. Doordat de genoemde ring in stand 3 komt wordt signaal MTA'

weer hoog. Echter het signaal MTB' wordt daardoor juist laag bij deze opdrachten: $(CO) = (OR10) = 1$.

MTB'	De flip-flops van het U-register worden
1 R3' CO' OR10'	normaal ingelezen met behulp van de signalen
	CU en UIL'. Wanneer de arithmetische

controlering in stand 3 staat moeten de flip-flops U26 en U0 het tekencijfer (TS) overnemen. Daartoe wordt niveau 17 van plaatje U26' resp. niveau 12 van plaatje U0' laag gemaakt met de signalen UIL' en MTB', mits $(TS) = 1$. Flip-flop U1 neemt dan onder dezelfde voorwaarden het cijfer (M2) over, terwijl flip-flop U2 eveneens onder dezelfde voorwaarden het cijfer (M3) overneemt.

$$U_0' / U_0$$

1 Houd U_0' CU
12 x 10 TS' UIL' MTB'

1 Houd U_0

$$U_2' / U_2$$

1 Houd U_2' CU
5 x 10 M3' UIL' MTB'

1 Houd U_2

$$U_1' / U_1$$

1 Houd U_1' CU
5 x 10 M2' UIL' MTB'

1 Houd U_1

$$U_{26}' / U_{26}$$

1 Houd U_{26}' CU
17 x 10 TS' UIL' MTB'

1 Houd U_{26}

Indien het resultaat in de registers A en S moet worden gezet: $(OR_0) = 0$, zal het A-register schoongemaakt moeten worden.

Signaal AI' wordt hoog gehouden door niveau 3 van plaatje AI laag te maken met signaal MTA' , zodat de inhoud van het U-register niet kan worden overgenomen. Signaal AI wordt eveneens

$$AI' / AI$$

10 x 10 MTA' 2ct' OR_0

1 AI
3 x 10 MTA'

hoog gemaakt, met behulp van niveau 10 op het accentplaatje, gedurende ringstand 2: $(MTA) = 1$ op 2 cijfer-tijd, mits $(OR_0) = 0$ is. Nadat de arithmetische controlering in stand 3 is gekomen, worden de vier minst

significante flip-flops van het A-register ingelezen. Op 1 cijfer-tijd neemt flip-flop A_0 het cijfer (SV25) over, flip-flop A_1 -(SV26), flip-flop A_2 -(SV0) en flip-flop A_3 -(SV1), mits geldt: $(MTB) = 1$ en $(OR_0) = 0$.

$$A_0' / A_0$$

1 Houd A_0' AI
3 x 10 SV25' 1ct' OR_0 MTB'

1 Houd A_0

$$A_2' / A_2$$

1 Houd A_2' AI
3 x 10 SV0' 1ct' OR_0 MTB'

1 Houd A_2

$$A_1' / A_1$$

1 Houd A_1' AI
3 x 10 SV26' 1ct' OR_0 MTB'

1 Houd A_1

$$A_3' / A_3$$

1 Houd A_3' AI
3 x 10 SV1' 1ct' OR_0 MTB'

1 Houd A_3

Wanneer het resultaat positief is moeten de overige flip-flops van het A-register een nul (tekencijfer) bevatten, hetgeen reeds het geval is tengevolge van signaal AI. Is het resultaat negatief, dan moeten de overige flip-flops een één bevatten. Daartoe worden de houdniveau's van de onaccentplaatjes dezer flip-flops hoog gemaakt met signaal CA. Op 3 cijfertijd, dus nadat het A-register is schoongemaakt met signaal AI, wordt het niveau van plaatje CA' laag, wanneer $(TS) = 1$ is en als geldt $(MTA) = 1$ en $(ORo) = 0$.

CA' Op alle inleesniveau's van het A-
 1 TS' 3ct' MTA' ORo register, die betrekking hebben op deze opdrachten is signaal ORo aangesloten, evenals op het niveau van plaatje CA'. Wanneer men nu zeker weet, dat het resultaat nog binnen de capaciteit van het S-register blijft of als men de uitloop onbelangrijk acht, kan men ook de andere opdracht: $(ORo) = 1$ gebruiken. Dan blijft het A-register onaangetast, omdat signaal ORo hoog is.

Er is slechts één additieve stap, waarbij dus een rondlopende overdracht kan optreden. Indien $(TS) = 0$, zal er geen rondlopende overdracht ontstaan. Indien $(TS) = 1$, zal de rondlopende overdracht, die uit de tekencijfers ontstaat "vanzelf" bij de minst significante cijfers worden geteld, want de tekencijfers staan nu in flip-flop M1 resp. flip-flop S1. De eerste is daar gekomen door het rondschiiven en de tweede is aan (TS) gelijk gemaakt. Deze overdracht is eigenlijk niet de "echte" rondlopende overdracht (CV26A). De echte rondlopende overdracht ontstaat uit het 25e cijfer van het woord (M24), dat nu op de meest significante plaats van het M-register staat en het tekencijfer van het S-register. Als uit deze cijfers een overdracht ontstaat verloopt alles zoals in het voorgaande reeds uitvoerig is behandeld.

Tenslotte moet op 2 cijfertijd nadat de arithmetische controle-ring in stand 3 is gekomen het resultaat in het S-register

SI' worden gezet. Dat gaat normaal met behulp
 7 x 10 MTB' 2ct' van signaal SI'. Daartoe wordt niveau 7 van plaatje SI' laag gemaakt met de signalen 2ct' en MTB'.

Type-opdrachten

De type-opdrachten vallen ook weer onder de communicatie-opdrachten, waarbij het adresgedeelte weer dient voor nadere specificatie van het functiegedeelte. We zullen hierbij de volgende notatie bezigen: b.v. 54/16/2. Het eerste getal (54) geeft het functiegedeelte van de opdracht, het tweede getal (16) geeft de waarde van het getal gevormd door de tien meest significante cijfers van het adresgedeelte en de vijf minst significante cijfers daarvan komen overeen met het derde getal (2). Van het adresgedeelte zijn bij dit voorbeeld de volgende cijfers: (OR9) = (OR1) = 1, de overige cijfers zijn nul.

Met de (electrische) schrijfmachine kunnen 84 symbolen worden getypt, waaronder begrepen de symbolen van de "hoofdletter-serie". Voor deze symbolen zijn 42 toetsen (electromagneetjes) beschikbaar, die genummerd zijn. Verder zijn er nog de toetsen "tabelleer" (Tab), "terug wagen, nieuwe regel" (TWNR), "hoofdletter" (HL) en "kleine letter" (KL). Totaal zijn er 46 cijfercombinaties nodig om de magneten uit te coderen. Met zes binaire cijfers zijn de getallen 0 t/m 63 beschikbaar. De eerste 46 zijn benut voor de genoemde magneten, terwijl de getallen 56 t/m 63 een "spatie" veroorzaken.

Moet een symbool worden getypt, dan zorgt de programmeur dat de bedoelde cijfercombinatie (of het inverse daarvan) in het A- of het S-register staat aan de minst significante zijde.

Er zijn vier type-opdrachten, nl. 54, 55, 62 of 63/16/2.

Tengevolge van een van deze opdrachten worden de 6 cijfers door het TP-register overgenomen en wel als volgt:

54/16/2	+(A) → (TP)
55/16/2	-(A) → (TP)
62/16/2	+(S) → (TP)
63/16/2	-(S) → (TP)

We zullen nu veronderstellen dat niveau 5 van plaatje CCTR' laag wordt. Tengevolge van de signalen CCTR' en Co' wordt niveau 3

	ACR1'/ACR1	
1 Houd ACR1'		van plaatje ACR1' laag, mits
3 1-2 ACRo' GCRO CCTR' Co'		de signalen ACRO' (arithme-
<hr/>		tische controlering in stand
1 Houd ACR1		1) en GCRO (geheugencontrole-
		ring in stand 0) laag zijn.

Zodra echter signaal ACR1' laag is, wordt niveau 2 van plaatje ACRO (blz. 72) laag gemaakt, met dezelfde signalen CCTR' en Co' (binnen dezelfde cijfertijd), zodat de arithmetische controle-ring van stand 1 via stand 2 direct naar stand 3 gaat.

Op 1 cijfertijd daarna wordt signaal CU' laag, zodat daarmee het "resultaat" in het U-register staat (blz. 72). De volgende cijfertijd (2ct') wordt het niveau van plaatje TPI' laag bij de

	TPI'	
1 R3' CO' OR9' OR23' NACO		type- (en pons-) opdrachten,
		want (CO) = (OR9) = (OR23) =
		1, mits signaal NACO laag

is. Dit signaal wordt later behandeld. Met behulp van de signalen TPI en TPI' neemt register TP de zes minst significante cijfers

	TPo'/TPo	
1 Houd TPo' TPI CTP		van het U-register over. Als voorbeeld is
2 Lees Uo' TPI'		flip-flop TPo gegeven.
<hr/>		Staat het woord eenmaal in register TP,
1 Houd TPo		dan wordt met behulp van de uitgangssignalen
		hiervan de relais van een zg. relaisboom

ingesteld (zie tek. 2.00.3 P1), met dien verstande, dat de relais worden bediend met electronenbuizen, terwijl deze buizen weer worden gestuurd door de signalen van register TP. De werking van een relaisboom wordt bekend verondersteld. We willen slechts wijzen op de overeenkomst tussen het uitcoderen van signalen op niveau's en het besturen van relais met signalen. Staan alle relais in de juiste stand, dan kan aan het geselecteerde electro-magneetje een stroompuls worden toegevoerd, zodat het bepaalde slagarmpje de gekozen letter tegen de rol van de schrijfmachine slaat en de letter (symbool) op het papier verschijnt.

De stroompuls ontstaat door het in geleiding brengen van een transistor, die op analoge wijze is geschakeld als een transistor uit een T- eenheid. Op de ingang van deze schakeling komt het

signaal TYPE', dat laag wordt tengevolge van signalen van de type-ponsring. Deze ring (zie tek x4) dient om de tijdsduur te bepalen, die minstens moet verlopen tussen twee type-opdrachten. (10 symbolen per seconde)

De type-ponsring wordt gestuurd door de signalen Do en D1, die weer worden afgeleid van de netspanning (50Hz). Elke periode van de netspanning worden zowel signaal Do' als signaal D1' eenmaal laag (zie fig. 38). Deze tijdsignalen lopen altijd door, zolang de machine aanstaat. Het tijdsverschil tussen het laag

Do Io Do Do



worden van signaal Do' en van signaal D1' is 10 m.sec. De typeponsring wordt gestart met hetzelfde signaal TPI, waarmee ook het TP-register wordt teruggesteld. Daartoe is dit signaal aangesloten op het houd niveau van plaatje TPRo, zodat deze flip-flop in stand 1 wordt gezet. Zodra dus een gegeven in dit register

wordt gezet gaat tevens de ring lopen en elke volgende overgang wordt veroorzaakt door een van de signalen Do' of D1'.

De overgang van stand 5 naar stand 6 is alleen mogelijk op 5 cijfertijd, mits signaal BP laag is. De reden hiervan wordt later behandeld.

TPRo'/TPRo

1 Houd TPRo'
2 7-8 TPR2' TPR3' Do' TC2'

1 Houd TPRo TPI
0-1
2 3-4 TPR2' TPR3 Do'
3 10-11 TPR2 TPR3' D1

TPR1'/TPR1

1 Houd TPR1'
2 1-2 TPRo' TPR3 Do' BPV

1 Houd TPR1
2 8-9 TPRo' TPR3' D1'

TPR2' / TPR2			TPR3' / TPR3		
1	Houd	TPR2'	1	Houd	TPR3'
2	2-3	TPR0' TPR1' D1'	2	5-6	TPR0 TPR1' TPR2 5ct' Oc'BP TCI'
3	6-7	TPR1' TPR3' D1'	<hr/>		
<hr/>			1	Houd TPR3	
1	Houd TPR2		2	11-0 2ct' IA'-SP	
2	4-5	TPR0 TPR3 D1'			
3	9-10	TPR1 Do'			

Is de ring in stand 11 gekomen, dan wordt deze op 2 cijfertijd teruggesteld met behulp van signaal IA'-SP, Dit signaal is afkomstig van een flip-flop, die in de 1-stand wordt gezet op 3 cijfertijd als de type-ponsring in stand 11 komt. Deze flip-flop

IA'-SP/IA-SP			dient om te onthouden of er een type- (of pons-) opdracht aan de gang is (stand 0), of niet (stand 1).		
1	Houd	IA'-SP TPR0 CIA-BSP			
2	type	3ct' TPR3' TPR1 TPR0			
<hr/>					
1	Houd IA-SP				

Daarom wordt de flip-flop in stand 0 gezet, zodra de type-ponsring gaat lopen: (TPR0) = 1, door aansluiting van signaal TPR0 op het houd niveau van het accentplaatje. Er zijn nog meer van dit soort flip-flops, die alle later zullen worden behandeld. Daarom zullen we hier volstaan met deze korte opmerking.

TYPE' Het reeds eerder genoemde signaal TYPE' is laag gedurende de standen 8, 9 en 10 van de type-ponsring door uitcodering van deze standen op het niveau van plaatje TYPE'.

TPRS' Signaal TPRS is laag zolang de type-pons-ring loopt en wordt hoog wanneer de ring in stand 0 komt. Dit signaal is aangesloten op het eerste niveau van plaatje WACHT'. Verder is op dit niveau een uitcodering van de type- (en pons-) opdrachten aangesloten en signaal NACO. Wanneer dus een type- of een ponsopdracht wordt

WACHT'			uitgevoerd en voordat deze klaar is komt er een nieuwe opdracht, dan gaat het 1e niveau van plaatje WACHT' laag en daarmee signaal WACHT hoog.		
1	TPRS	CO' OR9' OR23' NACO			

Signaal WACHT voorkomt de overgang van stand 1 (via stand 2) naar stand 3 van de arithmetische controlering. Op normale wijze gaat deze ring op 4 cijfertijd naar stand 2 en blijft daarin staan, totdat signaal WACHT laag wordt, omdat signaal TPRS hoog wordt. Op de eerstvolgende cijfertijd 3a gaat het genoemde niveau van plaatje CCTR' laag en daarmee de arithmetische controlering naar stand 3. Zolang signaal WACHT hoog is, kunnen dua geen andere opdrachten worden uitgevoerd.

Zoals reeds werd opgemerkt zijn er zes binaire cijfers nodig om de relais in de juiste stand te zetten. Nu bedient het signaal van een flip-flop soms een groep relais tegelijk. Van elke groep relais is er een contact gebruikt om na te gaan of de relais ook inderdaad in de goede stand staan. Indien de relais niet zijn bekrachtigd, geven die contacten verbinding met aardpotentiaal (hoog) en indien de relais zijn bekrachtigd, zijn de signalen, die van deze contacten worden afgenomen, laag. Deze signalen zijn TRo' t/m TR5' genoemd en dienen om de stand van de relais weer in het M-register te zetten ter controle. Dat wordt gedaan met opdracht 50/16/2 (deze voorstelling van een opdracht wordt nu bekend verondersteld) als het controlewoord uiteindelijk in het A-register moet komen en met opdracht 58/16/2 als het in het S-register moet worden gezet. Verder zijn er nog de negatieve versies: 51/16/2 resp. 59/16/2. Deze vier terugleesopdrachten worden ook weer uitgevoerd volgens het middelste tijdschema op tekening XE 24.

CMIT'	Het niveau van plaatje CMIT' is laag,
1 ACRO' a OR1' BTL'	als de arithmetische controlering in stand 1 staat: (ACRO) = 1, als geldt (OR1) = 1 en als signaal a laag is. Bovendien moet (BTL) = 1 zijn. Het niveau van plaatje BTL' is laag, als: (CO) = (OR9) = 1 en (OR23) = (NACO) = 0.
BTL'	
1 CO' OR9' OR23 NACO	Op cijfertijd 3a wordt niveau 4 van plaatje CM' laag, mits geldt:
CM'	(ACRO) = (BTL) = 1 en:
4 ACRO' 3ct' a FO SKIP BTL' MB.	(FO) = (SKIP) = (MB) = 0.

Door combinatie van de signalen CM en CMIT op plaatje MIT wordt signaal MIT' laag gemaakt, waarmee het woord in de zes minst

MIT		Mo'/Mo				significante flip-flops van het
1	CM	1	Houd	Mo'	CM	M-register gelezen wordt. Als
2	CMIT		lees			voorbeeld zijn de betreffende
		5	terug	TRo'	MIT'	niveau's van flip-flop Mo gegeven.
		<hr/>				
		1	Houd	Mo		Op niveau 4 van plaatje CM' is

inhoud van het M-register niet wordt aangetast, indien de opdracht fout is gelezen. Immers de arithmetische controlering wordt pas teruggesteld op cijfertijd 6b (blz. 68) en dan zou de opdracht in het M-register reeds zijn vernietigd, terwijl deze (foute) opdracht juist bewaard moet blijven om na te kunnen gaan hoe de fout is ontstaan. Met behulp van signaal BAO' wordt de opteller in stand 2 gezet (niveau 2 van plaatje CO12', blz. 69) of in stand 3 (niveau 1 van plaatje CO13', blz. 70), waardoor het woord uit het M-register positief of negatief in het U-register komt en vandaar afhankelijk van de opdracht in het A- of het S-register wordt gezet. Daarvoor moet niveau 1 van plaatje AI' of niveau 1 van plaatje SI' laag worden (blz. 73). Er zijn ook nog enkele opdrachten waarmee het controlewoord uit het M-register kan worden opgeteld bij of afgetrokken van het woord uit het A- of het S-register, waarna het resultaat weer in het zelfde register wordt gezet.

De additieve terugleesopdrachten zijn: 48, 49, 56 of 57/16/2. De eerste twee hebben betrekking op het A-register en de andere op het S-register. De gegevens van de relaiscontacten worden op overeenkomstige wijze in het M-register gezet als bij de niet additieve opdrachten. Het tijdschema voor de additieve opdrachten staat bovenaan op tek. XE24. De opteller wordt in stand 0 of stand 1 gezet door het laag worden van niveau 6 van plaatje COIo' of niveau 5 van plaatje COI1' (zie blz. 69). Omdat nu een optelling c.q. aftrekking moet worden verricht, die langer duurt dan een transport, wordt de arithmetische controlering later in stand 3 gezet. Daartoe moet een niveau van plaatje CCTR laag worden.

CCTR'/CCTR

1 Houd CCTR' Co'

6 2-3 ACRo' 7ct' b ABS' DRO WACHT SCO MB

1 Houd CCTR

Niveau 2 van dit plaatje kan nu niet laag worden, want signaal CO is hoog. Niveau 6 echter wordt nu laag op cijfertijd 7b, als $(ACro) = (ABS) = 1$ en $(DRO) = (SCO) = (MB) = 0$. Ook signaal WACHT moet laag zijn om dezelfde reden als hiervoor is beschreven. Het inlezen van de interne registers geschiedt overeenkomstig het inlezen bij de niet additieve opdrachten.

Aangezien een terugleesopdracht direct na een typeopdracht kan worden gegeven moet deze opdracht enige tijd wachten tot de relais voldoende gelegenheid hebben gehad om zich in te stellen, anders zou een nog niet correct controlewoord terug gelezen worden. Daarom worden de niveau's 3 en 4 van plaatje WACHT' laag

WACHT'	gemaakt bij deze opdrachten:	$(BTL) = 1$
3 TPRo' TPR3 BTL' ORo	en $(ORo) = 0$,	zolang de type-pons-
4 TPR1' TPR3 BTL' ORo	ring in de standen 1 t/m 3 (niveau 3)	en 2 t/m 5 (niveau 4) staat. Het

signaal WACHT is dus hoog tot en met stand 5 van genoemde ring, de relaisboom zal dan zijn ingesteld. Het niveau van plaatje CMIT' echter wordt laag iedere keer als signaal a laag wordt en niveau 4 van plaatje CM' is nu op iedere cijfertijd 3a laag. Het M-register neemt dus iedere cijfertijd 3a, zowel in stand 1 als stand 2 van de arithmetische controlering: $(ACRo) = 1$, het controlewoord over, correct of niet. Zodra echter signaal WACHT (onaccent!) laag wordt, kan niveau 5 van plaatje CCTR' laag worden op cijfertijd 3a. Op dezelfde cijfertijd is dan het controlewoord, dat nu correct wordt verondersteld, weer in het M-register gezet. De arithmetische controlering gaat nu naar stand 3, zodat het woord verder kan worden verwerkt.

Van al deze type- en terugleesopdrachten is zowel de B-correctie als de B-modificatie mogelijk. Alleen dient hierbij in het oog te worden gehouden, dat daarmee wel het adresgedeelte van de opdracht wordt gewijzigd, maar dat dat adresgedeelte geen adres in het geheugen aangeeft. Door een B-correctie of -modificatie wordt de aard van de opdracht gewijzigd. Van een type-opdracht wordt daardoor b.v. een ponsopdracht gemaakt. Wordt toch van B-modificatie gebruik gemaakt, dan verloopt deze opdracht volgens het onderste tijdschema op tek. XE 24. Alles verloopt daarbij overeenkomstig het hiervoor behandelde op twee signalen na.

Zoals bekend is duurt een gemodificeerde opdracht vier cijfer-
 tijden langer dan een ongemodificeerde. Daarom wordt op cijfer-
 tijd 7b niveau 5 van plaatje CM' laag, als: $(R2) = (BTL) = (MB) = 1$
 CM' en $(FO) = 0$. Ook de arithmetische
 5 R2' 7ct' b BTL' MB' controlering moet later in stand
 3 worden gezet. Daarvoor dient
 niveau 15 van plaatje CCTR'. Dit niveau wordt laag op cijfertijd
 3a', als: $(R2) = (CO) = 1$ en $(SCO) = (OR10) = (WACHT) = 0$.

CCTR'/CCTR

1 Houd CCTR' Co'
 15 R2' 3ct' a' CO' SCO WACHT OR10

1 Houd CCTR

Ponsopdrachten

De bandponser dient om gegevens uit de X-1 in een (papieren)
 band te ponsen. Door de opdrachten 54, 55, 62 of 63/16/1 neemt
 het TP-register een te ponsen woord uit de registers A of S over.
 Dat gebeurt volgens hetzelfde tijdschema en met behulp van het-
 zelfde signaal TPI', zoals bij de type-opdrachten. Signaal TPI'

BP'/BP

1 Houd BP' TPI
 2 Lees ORo' TPI'

1 Houd BP

start ook weer de type-ponsring. Tevens
 neemt flip-flop Bp het cijfer (ORo) over
 met behulp van signaal TPI. Deze flip-flop
 komt dus in stand 1 bij een ponsopdracht
 $(ORo) = 1$ en wordt pas teruggesteld de

eerstvolgende keer dat signaal TPI hoog wordt. Het signaal BP
 is, zoals reeds werd opgemerkt bij de type-opdrachten, aange-
 sloten op niveau 2 van plaatje TPR3', waardoor de overgang van
 stand 5 naar stand 6 van deze ring wordt voorkomen (blz. 165).

IA'-SP/IA-SP

1 Houd IA'-SP TPRo CIA-BSP
 3 Pons 3ct' TPR2 TPR1' TPRo BP'

1 Houd IA-SP

Op 3 cijfertijd wordt de
 flip-flop IA'-SP in
 stand 1 gezet met behulp
 van signaal BP' als de
 type-ponsring in stand

5 staat. Op 2 cijfertijd daarna wordt de flip-flop TPR1 terug-
 gesteld door niveau 3 van plaatje TPR1 laag te maken met

TPR1'/TPR1

1 Houd TPR1'

1 Houd TPR1'

3 5-0 TPR0 TPR2 TPR3 2ct' IA'-BP

en daarna terug, dat duurt dus 50 m.sec. De bandponser kan 20 ponsingen per sec volbrengen.

In de ponsband worden zg, pentaden geponst. Een pentade is een woord van vijf binaire cijfers. Aangezien er alleen direct binair geponst wordt, behoeft er niet eerst een relaisboom te worden ingesteld, maar kunnen de uitgangssignalen van het TP-register direct de electromagneetjes (via electronenbuizen) bedienen. Staan de magneetjes in de juiste stand, dan kan de electromagneet, die de ponsing veroorzaakt worden bekrachtigd. Dat doet signaal PONS' (weer via een electronenbuis). In de standen 3 en 4 van de type-ponsring, wordt het niveau van plaatje PONS' laag gemaakt met de signalen TPR2', TPR3 en BP'. Ook hier kan men uit een eenvoudig rekensommetje de conclusie trekken, dat er ruim 700 opdrachten (van ongeveer 60 à 70 mm.sec elk) kunnen worden uitgevoerd tussen twee ponsopdrachten.

Voor ponsopdrachten met B-correctie of B-modificatie geldt dezelfde opmerking als bij de type-opdrachten is gemaakt.

De band is door de bandponser tevens voorzien van geleidegaatjes,

zodat de band, indien deze in de bandlezer wordt gelegd, ook referentiepunten heeft om op grond daarvan te worden getransporteerd. De twee meest significante cijfers van de

pentade worden aan de ene zijde van de geleidegaatjes geponst en de drie minst significante aan de andere zijde.

Bandleesopdrachten

De bandlezer dient om een geponste band te lezen, zodat de X-1 over de gegevens uit de band kan beschikken. De band wordt in de bandlezer door een opstepmechanisme, dat afhankelijk van de X-1 werkt en alleen door de X-1 wordt gestart, tussen een lampje en vijf fotocelletjes gevoerd en wel zodanig, dat na iedere step een rij (eventueel aanwezige) gaatjes precies boven de rij met die gaatjes corresponderende fotocellen ligt. Indien er gaatjes in de band zitten, ontvangen de fotocelletjes licht en geven een spanning af. Deze signaalspanningen worden via electronenbuizen toegevoerd aan het M-register.

De band wordt niet "vloeiend" door de bandlezer gevoerd, maar met schokjes (stappen) en wel zodanig, dat een pentade steeds korte tijd boven de fotocellen stilligt en daarna verder wordt getransporteerd. De geleidegaatjes dienen om een fotocel te belichten en daarmee te bepalen of de pentade zuiver voor de overige fotocellen ligt. Met de geleide-fotocel wordt het signaal TBL' gemaakt. Dit signaal wordt toegevoerd aan de bandcontrole-ring. In de eerste twee standen van deze ring ligt de band stil en gedurende de laatste twee standen wordt de band getransporteerd. Normaal staat de ring in stand 1 in rust.

De bandlezer kan ca 150 pentaden per sec lezen, zodat tussen twee bandleesopdrachten ongeveer 100 andere opdrachten kunnen worden uitgevoerd. De bandleesopdrachten, waarbij het woord uit de bandlezer via het M-register geïnverteerd of niet naar het A- of S-register wordt getransporteerd, zijn: 50, 51, 58, 59/16/1. Deze opdrachten verlopen volgens het middelste tijdschema op tek. XE 24. Signaal BTL' is ook bij deze opdrachten laag (blz. 166).

Om de gelezen pentade in het M-register te zetten, wordt het niveau van plaatje CMIB' laag gemaakt in de standen 1 en 2 van de

	CMIB'		MIB	
1	ACRo'	a BTL'	ORo'	1 CM
				2 CMIB

arithmetische controlering: (ACRo) = 1 als signaal a laag is, bij een bandleesopdracht: (BTL) = (ORo) = 1. Van plaatje CM' werkt nu hetzelfde niveau als bij de terugleesopdrachten. Met de signalen CM en CMIB wordt signaal MIB' gemaakt. Overigens ontstaan alle signalen bij dit tijdschema als in het voorgaande is besproken.

BCRo' / BCRo					BCR1' / BCR1				
1	Houd	BCRo'			1	Houd	BCR1'		
2	0-1	BCR1	BLV	2ct'	2	1-2	R3'	BTL'	ORo'
<hr/>					<hr/>				
1	Houd	BCRo			1	Houd	BCR1		
2	2-3	BCR1'	TBL		2	3-0	BCRo	TBL'	1ct'

Wanneer nu de arithmetische controlering in stand 3 is gekomen ($R3'$ is laag), gaat de bandcontrolering van stand 1 naar stand 2, mits geldt: $(BTL) = (ORo) = 1$ en als de fotocel geen licht meer ontvangt door de geleidegaatjes: $(TBL) = 0$, gaat de band-controlering door naar stand 3. Zolang deze ring in stand 2 of 3 staat, wordt de band door het opstepmechanisme getransporteerd. Komt er dan een geleidegaatje voor de betreffende fotocel, dan wordt signaal TBL' laag en gaat de band-controlering op 1 cijfertijd naar stand 0. Daardoor wordt het opstepmechanisme gestopt en zo dus ook de band. Men dient hierbij in het oog te houden, dat de band-controlering zéér langzaam loopt ten opzichte van een ring, die door cijfertijden wordt gestuurd b.v. de arithmetische controlering.

Stel een pentade ligt goed voor de fotocellen. Dan gaat de band-controlering naar stand 2 (zie boven) en het opstepmechanisme wordt gestart, daardoor gaat de band verder. Op het moment, dat het schuiven begint, valt er nog licht door het geleidegaatje op de fotocel, waardoor signaal TBL dus hoog is. Pas na enkele milliseconden zal het geleidegaatje zover weg zijn, dat de fotocel door de band is afgesloten, signaal TBL dus laag wordt en de bandcontrolering naar stand 3 kan gaan. Dan duurt het weer enkele milliseconden voor het eerste stukje van het volgende geleidegaatje licht doorlaat en signaal TBL' laag gaat worden. Eerst als signaal TBL' "goed" laag is, kan de bandcontrolering naar stand 0 en de band worden gestopt. Dan moet de band enige milliseconden stilliggen om de fotocellen gelegenheid te geven de gegevens juist over te nemen en door te geven.

Zodra de bandcontrolering in stand 0 komt wordt de "langzame" one-shot-multivibrator BLV gestart met signaal BCR1. Het uitgangssignaal BLV van deze multivibrator wordt daardoor gedurende enkele milliseconden hoog en daarna weer laag. Zodra nu signaal BLV weer laag is, wordt niveau 2 van plaatje BCRo' laag op 2 cijfertijd, de bandcontrolering gaat daardoor naar stand 1 en blijft daarin staan tot de volgende bandleesopdracht. Het signaal BLV wordt de overgang van stand 0 naar stand 1 van deze ring dus enkele milliseconden vertraagd. Het is dus zo, dat alleen over het woord uit de bandlezer kan worden beschikt, als de band stilligt. Omdat te garanderen, worden twee niveau's van plaatje WACHT' laag gemaakt bij deze opdrachten: (BTL) = (ORo) = 1 gedurende de standen 2, 3 en 0 van de bandcontrolering. De standen 2 en 3: (BCR1) = 1 zijn uitgecodeerd op niveau 2 van plaatje WACHT' en de stand 0: (BCRo) = 0 op niveau 5. De arithmetische controlering kan nu niet naar stand 3, omdat het betreffende niveau van plaatje CCTR' niet laag kan worden (blz. 167) Wel wordt iedere cijfertijd 3a het (dan) gelezen woord in het M-register gezet, maar kan niet verder worden getransporteerd. Pas als de bandcontrolering in stand 1 komt, wordt signaal WACHT laag, doordat dan de niveau's 2 en 5 van plaatje WACHT' beide hoog zijn. De arithmetische controlering gaat nu naar stand 3, waardoor het woord, dat het M-register de laatste keer uit de bandlezer heeft overgenomen, verder kan worden verwerkt. Tevens gaat nu de bandcontrolering naar stand 2 en vervolgens naar 3-0-1, zodat de volgende pentade voor komt te liggen en bij de eerstvolgende bandleesopdracht gelezen kan worden, mits de bandcontrolering reeds in stand 1 staat op het moment dat deze opdracht wordt gegeven.

De additieve bandleesopdrachten zijn: 48,49,56 en 57/16/1. Deze opdrachten verlopen volgens het bovenste tijdschema op tek. XE24. Dit tijdschema behoeft nu geen nader betoog meer (zie terugleesopdrachten), terwijl de bandlezer zelf functioneert als hierboven is beschreven.

Ook bij de bandlcesopdrachten is B-correctie en B-modificatie mogelijk, hoewel daaraan dezelfde bezwaren kleven als bij de type-opdrachten zijn behandeld.

Register-copieer-opdrachten

Teneinde de inhoud van het ene register in het andere te kunnen kopiëren zijn de opdrachten 54 en 55/8+p/q toegevoegd. Hierin stelt p het register van uitgang voor en q het register van bestemming. De letters p en q kunnen elk daarbij drie waarden aannemen, waardoor het interne register wordt bepaald:

0 (00)	A-register
1 (01)	S-register
2 (10)	B-register

De 8 in het tweede deel van de opdracht geeft aan dat het cijfer (OR8) = 1 is. Overigens is de voorstelling van deze opdracht zoals in het voorgaande is behandeld. De inhoud van het ene register kan de inhoud van het andere geïnverteerd overnemen (55) of niet geïnverteerd (54).

RCO' Bij deze opdrachten: (CO) = (OR8) = 1 en
 1 CO' OR8' NACO (NACO) = 0 wordt het niveau van plaatje
 RCO' laag. Afhankelijk van de waarde van p

wordt een van de interne registers geselecteerd. Dat is uitgecodeerd op de betreffende multipletniveau's, die worden laag

LA'	gemaakt met signaal BAO'. Het A-
1 Houd LS LB LOT	register wordt geselecteerd als
6 BAO' RCO' OR5 OR6	signaal RCO' laag is en als p = 00,
LS'	LB'

1 Houd LA LB LOT	1 Houd LA LS LOT
6 BAO' RCO' OR5'	4 BAO' RCO' OR6'

m.a.w. als (OR5) = (OR6) = 0. Het S-register wordt geselecteerd, als p = 01, dus: (OR5) = 1, mits geldt (RCO) = 1 en het niveau van plaatje LB' wordt laag als p = 10: (OR6) = RCO) = 1.

Deze opdrachten verlopen volgens het tijdschema op tek. XE24. Het "resultaat" wordt op grond van q in één van de interne registers geborgen op 2 cijfertijd, nadat de arithmetische controlering in stand 3 is gekomen.

AI'	Natuurlijk moet voor alle
2 R3' 2ct' RCO' ORo OR1 NRW	drie de inleessignalen gelden:
	(RCO) = 1 en (NRW) = 0.

Verder wordt het A-register ingelezen, als $q = 0$, dus:
(ORo) = (OR1) = 0.

SI'	BI'
6 R3' 2ct' RCO' ORo' NRW	4 R3' 2ct' RCO' OR1' NRW

Het S-register neemt het woord over als $q = 01$, dus (ORo) = 1 en het B-register als $q = 10$, dus (OR1) = 1. Dit alles is uitgedoceerd op de niveau's van de betreffende inleessignalen. Ook bij deze opdrachten is B-correctie en B-modificatie mogelijk, hoewel hieraan de reeds bekende bezwaren kleven.

Absoluut

Van de opdrachten 0 t/m 43 kunnen de in-opdrachten ook absoluut worden uitgevoerd, dat wil zeggen het adresgedeelte van de opdracht wordt als absoluut getal beschouwd en verwerkt. Daarbij moeten de cijfers (OR19) = 1 en (OR20) = 0 zijn. Aangezien deze cijfers ook dienen om aan te geven, dat de opdracht normaal, met B-correctie of met B-modificatie moet worden uitgevoerd zijn deze varianten hierbij onmogelijk.

De additieve absolute opdrachten verlopen volgens het bovenste tijdschema op tek. XE24 en de niet-additieve absolute opdrachten volgens het middelste tijdschema op dezelfde tekening.

De opdrachtcycli op beide tijdschema's verlopen op normale wijze dus zowel het functie- als het adresgedeelte van de opdrachten worden door het opdrachtregister overgenomen uit het M-register. Het woord in het M-register wordt daardoor niet veranderd.

ABS'	CMF'
1 OR19' OR20 TSS SKIP	1 ACRo' 3ct' a ABS' FO NACO

Tengevolge van de aard van de opdracht wordt het eerste niveau van plaatje ABS' laag: $(OR19) = 1$ en $(OR20) = (TSS) = 0$, mits signaal SKIP laag is. Van het woord in het M-register, wordt het functiegedeelte nul gemaakt door aansluiting van signaal CMF op de houdniveau's der betreffende accentplaatjes, zodat alleen het adresgedeelte in het M-register blijft staan. Dit getal is altijd positief, aangezien het hele functiegedeelte met inbegrip van cijfer 26 (het tekencijfer) nul wordt gemaakt met signaal CMF. Het niveau van plaatje CMF' wordt laag op cijfertijd 3a als de arithmetische controlering in de standen 1 of 2 staat, mits geldt: $(ABS) = 1$ en $(FO) = (NACO) = 0$. Signaal FO is hier weer aangesloten om de opdracht in het M-register niet te verminken, indien de opdracht fout uit het geheugen is gelezen. De geheugencontrolering wordt teruggesteld op 6 gct, omdat signaal GG nu laag is (blz. 86). Het niveau van plaatje GG' wordt hooggemaakt met signaal ABS. Overigens zijn beide tijdschema's in het voorgaande voldoende besproken.

Signaal WACHT, dat is aangesloten op de niveau's 5 en 6 van plaatje CCTR' is bij deze opdrachten altijd laag, want: $(CO) = 0$.

Conditie zetten en - gehoorzamen.

De cijfers (OR17) en (OR18) dienen om de "conditie te zetten". Wanneer beide cijfers nul zijn, wordt de opdracht uitgevoerd, waarbij dan geen conditiezetting plaatsvindt (normaal). Zijn beide of een van beide cijfers één, dan wordt de conditie gezet.

CZO'	Dat gebeurt met behulp van signaal CZO',
1 R3' S0 OR17'	dat laag is, als geldt: $(R3) = 1$ en $(S0) = 0$,
2 R3' S0 OR18'	mits of $(OR17) = 1$ is of $(OR18) = 1$.

In de flip-flop COND kan een antwoord worden opgeborgen op een door de opdracht gestelde vraag (conditie zetten).

(COND) = 0 betekent: ja

(COND) = 1 betekent: neen.

Indien de cijfers $(OR17) = 1$ en $(OR18) = 0$ zijn, neemt de flip-flop COND het teken COND'/COND

1 Houd COND'					over van het getal,
2 Lees U26'	3ct: CZO'	OR18			dat bij deze opdracht
3 Lees NUL	3ct: CZO'	OR17			het laatst in het U-
4 Lees U26 LTR'	3ct' CZO'	OR17'	OR18'		register is gezet (re-
5 Lees U26' LTR	3ct' CZO'	OR17'	OR18'		sultaat), waarmee dus

1 Houd COND					een antwoord is ver-
2 terug 2ct' CZO'					kregen op de vraag:

Is het resultaat positief? Dit is uitgecodeerd op niveau 2 van plaatje COND' en wel op 3 cijfertijd, nadat de arithmetische controlering in stand 3 is gekomen: $(CZO) = 1$, terwijl op 2 cijfertijd daarvoor de flip-flop COND is teruggesteld met behulp van signaal CZO' op niveau 2 van plaatje COND.

Dit laatste gebeurt dus bij elke conditie-zettende opdracht.

Indien de cijfers $(OR17) = 0$ en $(OR18) = 1$ zijn, wordt $(COND)=0$ gelaten, mits de laatste inhoud van het U-register gelijk aan nul is en de inhoud van flip-flop COND één wordt gemaakt op 3 cijfertijd als het resultaat (U-register) niet gelijk aan nul is.

Daartoe zijn alle onaccent uitgangssignalen van het U-register (laag bij +0 in U) aangesloten op niveau 2 van plaatje NUL' en alle accentuitgangssignalen (laag bij -0 in U) zijn aangesloten op niveau 1 van plaatje NUL'. Signaal NUL is dus hoog, indien het getal in het U-register gelijk is aan nul. Deze voorwaarde is uitgecodeerd op niveau 3 van plaatje COND'. Hier wordt dus de vraag gesteld: Is het resultaat nul?

Het is ook mogelijk de conditie te zetten op grond van gelijkheid of ongelijkheid der tekens van het resultaat der voorgaande conditie-zettende opdracht en van het resultaat bij deze conditie-zettende opdracht. Dat gebeurt op 3 cijfertijd met behulp van de niveau's 4 en 5 van plaatje COND in het geval dat de cijfers $(CR17)$ en $(OR18)$ beide een zijn. Daartoe wordt het teken van het resultaat der voorgaande conditie-zettende opdracht onthouden in de flip-flop LTR. Op 3 ct kan de inhoud van de flip-flop worden

LTR'/LTR

1 Houd LTR'

2 Lees U26' 4ct' CZO'

1 Houd LTR

2 Lees U26' 4ct' CZO'

gebruikt voor genoemde conditiezetting, terwijl op 4 cijfertijd flip-flop LTR het teken van het nu in het U-register staande resultaat overneemt. De hier gestelde vraag luidt: Heeft er tekenwisseling plaats gehad?

De conditie kan dus op verschillende wijzen worden gezet. Het is mogelijk de X-1 te laten handelen afhankelijk van het antwoord op een gestelde vraag (conditie gehoorzamen). Het gehoorzamen van de conditie gebeurt met behulp van de cijfers (OR15) en (OR16). Indien beide cijfers nul zijn, wordt de opdracht uitgevoerd ongeacht de conditie (normaal).

Als de cijfers (OR15) = 0 en (OR16) = 1 zijn, moet de opdracht worden uitgevoerd, indien (COND) = 0 is en als (COND) = 1 is moet de opdracht worden overgeslagen (geskipt). Onder deze voorwaarden wordt daartoe niveau 2 van plaatje SKIP' laag gemaakt. Hoe met behulp van signaal SKIP de opdracht wordt "overgeslagen" wordt hierna behandeld.

Als de cijfers (OR15) = 1 en (OR16) = 1 zijn, moet de opdracht alleen worden uitgevoerd, als (COND) = 1 is. Indien hierbij (COND) = 0 is, wordt de opdracht overgeslagen. Deze voorwaarde is uitgecodeerd op niveau 1 van plaatje SKIP'.

Stel nu, dat de cijfers (OR15) = 1 en (OR16) = 0 zijn. In dit geval wordt de inhoud van het interne register niet gewijzigd, hoewel dit volgens de aard der opdracht (in-opdracht) wel moest plaatsvinden. Daartoe is deze voorwaarde uitgecodeerd op het niveau van plaatje NRW', dat dus laag wordt, als geldt: (OR15) = 1 en (OR16) = 0, mits (SO) = 0. Het signaal NRW is aangesloten op alle betreffende niveau's van de plaatjes, waarmee de inleessignalen worden gemaakt, zodat daardoor wel de opdracht geheel wordt uitgevoerd, maar het resultaat wordt daarbij niet door het interne register uit het U-register overgenomen, aangezien dan signaal NRW hoog is. Deze variant heeft niets te maken met conditie-gehoorzamen, maar is nuttig bij conditie-zetten.

SKIP' worden uitgevoerd, indien (COND) = 0 is

1 COND OR15' OR16' en als (COND) = 1 is moet de opdracht

2 COND' OR15 OR16' worden overgeslagen (geskipt). Onder deze

voorwaarden wordt daartoe niveau 2 van

plaatje SKIP' laag gemaakt. Hoe met behulp van signaal SKIP de

opdracht wordt "overgeslagen" wordt hierna behandeld.

Als de cijfers (OR15) = 1 en (OR16) = 1 zijn, moet de opdracht

alleen worden uitgevoerd, als (COND) = 1 is. Indien hierbij

(COND) = 0 is, wordt de opdracht overgeslagen. Deze voorwaarde

is uitgecodeerd op niveau 1 van plaatje SKIP'.

Stel nu, dat de cijfers (OR15) = 1 en (OR16) = 0 zijn. In dit

geval wordt de inhoud van het interne register niet gewijzigd,

hoewel dit volgens de aard der opdracht (in-opdracht) wel moest

plaatsvinden. Daartoe is deze voorwaarde uitgecodeerd op het

niveau van plaatje NRW', dat dus laag wordt, als geldt: (OR15) = 1

NRW' en (OR16) = 0, mits (SO) = 0. Het signaal NRW

1 SO OR15' OR16 is aangesloten op alle betreffende niveau's

van de plaatjes, waarmee de inleessignalen

worden gemaakt, zodat daardoor wel de opdracht geheel wordt uitge-

voerd, maar het resultaat wordt daarbij niet door het interne

register uit het U-register overgenomen, aangezien dan signaal NRW

hoog is. Deze variant heeft niets te maken met conditie-gehoorzamen,

maar is nuttig bij conditie-zetten.

In het voorgaande gold steeds $(S0) = 0$. Maar als $(S0) = 1$, hebben de cijfers (OR15) t/m (OR18) een andere betekenis. Conditie-zetten is nu onmogelijk. Voor de opdrachten 44 en 45 (zie blz. 83 en 84) geldt, dat de sprongopdracht normaal wordt uitgevoerd, indien de cijfers (OR17) en (OR18) beide nul zijn. Zoals bekend is, wordt bij deze opdrachten (τ) met 1 verminderd. Als de cijfers (OR17) = 1 en (OR18) = 0 zijn, wordt de sprong alleen uitgevoerd, indien van (τ) het teken positief blijft: $(U26) = 0$. Is dit het geval, dan neemt de opdrachtteller het

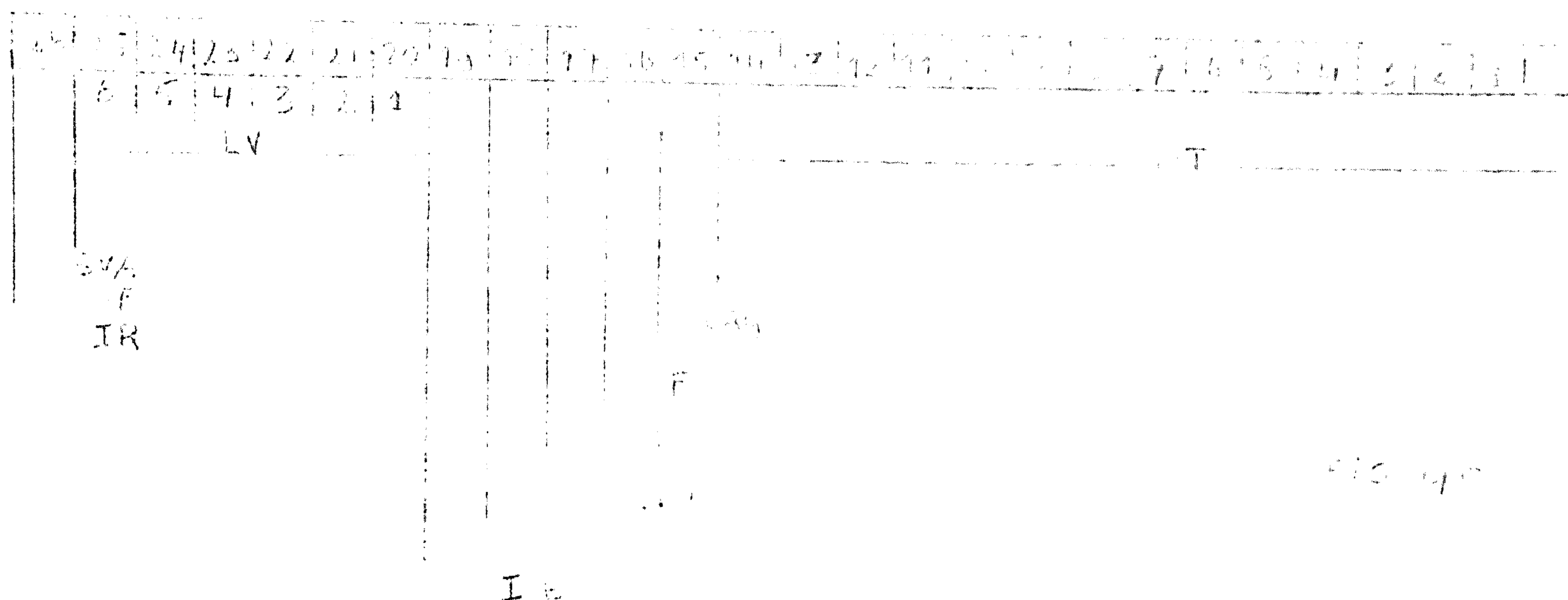
COT'	adresgedeelte uit het opdracht-
5 R3' 3ct' U26 OR17' TSS'	register over. Daartoe wordt
6 R3' 3ct' NUL' OR18' TSS'	dan niveau 5 van plaatje COT'
	laag gemaakt.

Wanneer de cijfers (OR17) = 0 en (OR18) = 1, wordt de sprong alleen uitgevoerd, als na de vermindering geldt $(\tau) = +0$ of $(\tau) = -0$. Ook nu komt (τ) weer in het U-register en zoals hiervoor reeds is besproken, wordt daarvan signaal NUL' afgeleid. Als signaal NUL' laag is, wordt niveau 6 van plaatje COT' laag, zodat dan de sprong weer wordt uitgevoerd.

Zijn de cijfers (OR17) = (OR18) = 1, dan kan of niveau 5 of niveau 6 van plaatje COT' laag worden afhankelijk van (τ). Deze voorwaarde is dus een combinatie van beide voorgaande gevallen. Wordt signaal COT' laag, dan wordt de sprong weer uitgevoerd.

Bij de opdrachten 46 en 47 (blz. 81 en 82) heeft alleen het cijfer (OR18) betekenis en dient om het aantal λ 's uit te breiden. Indien (OR18) = 1 is, worden de adressen $\lambda_m + 8$ bedoeld door uitcodering op de SG-plaatjes.

Wanneer een van deze beide opdrachten wordt gegeven, worden behalve de inhoud van de opdrachtteller nog een aantal gegevens, afkomstig van flip-flops in een van de adressen λ opgeborgen. De gegevens en de plaatsen waar ze gezet zijn, zijn aangegeven in fig. 40.



Deze gegevens komen met behulp van signaal LR' in het coderend circuit OI. De inhoud van de opdrachtteller komt daar met behulp van het multipletsignaal LOT'. Dit signaal is ook aangesloten op het niveau van plaatje LR'. Verder zijn daarop aangesloten de signalen TSS' en OR22', zodat dit niveau alleen laag kan worden bij de opdrachten 46 en 47 en wel zolang de arithmetische controlering in de standen 2 en 3 staat: (ACR1) = 1. Dit laatste om te voorkomen, dat bij het op-
 hogen van de opdrachtteller deze gegevens reeds in OI komen
 CI18' (zie tek. XE10-schoon-uit). Als
 4 lees reddend COND' LR' voorbeeld is niveau 4 van plaatje
 OI18' gegeven. De betreffende niveau's
 van de overige OI-plaatjes zijn op overeenkomstige wijze uitge-
 codeerd.

Bij de opdrachten 40 t/m 43 heeft alleen het cijfer (OR17) betekenis. Als dit cijfer gelijk aan nul is, verloopt alles zoals is beschreven op blz. 74 en 75. Is het cijfer (OR17) = 1, dan worden behalve de opdrachtteller, bovendien de flip-flops LV, COND, LTR en OF ingelezen (hersteld) op grond van (n). Het adres (n) kan elk willekeurig adres in het geheugen zijn en dus ook λ -adressen. Wanneer de inhoud van de genoemde flip-flops hersteld moet worden, wordt dat gedaan op grond van (λ).

	HI'				Het herstellen wordt verricht met	
1	R3'	b	SO'	TSS	OR17'	de signalen HI' en HI. Het niveau
						van plaatje HI' wordt laag als de
						arithmetische controlering in stand 3 staat (R3' is laag) en
						signaal b laag is bij deze opdrachten: (SO) = (OR17) = 1 en
			COND'/COND			(TSS) = 0. Als voorbeeld worden de
1	Houd		COND'	HI		betreffende niveau's van flip-flop COND
6	Herstel	M18'	HI'			gegeven met behulp waarvan de inhoud
						wordt hersteld.
1	Houd		COND			

Een sprongopdracht kan geen conditie zetten, maar wel gehoorzamen, zoals we nu zullen zien. De conditie moet dan door een andere (vroegere) opdracht zijn gezet. Wanneer het cijfer (OR16) = 1 is, geldt hetzelfde in het voorgaande bij die beide gevallen is besproken.

Indien de cijfers (OR15) = 1 en (OR16) = 0 zijn, wordt de sprongopdracht uitgevoerd, als (OF) = 1 is en

	SKIP'				opdracht uitgevoerd, als (OF) = 1 is en
3	OF	OR15'	OR16	SO'	wordt deze overgeslagen (geskipt) als
					(OF) = 0 is. De flip-flop OF wordt

ingelezen aan de hand van het laatst verkregen resultaat uit een van de volgende opdrachten:

		0, 1, 8 en 9:	(OR26) = (OR23) = 0, (AO) = 1
en:		4, 5, 12 en 13:	(OR26) = 0, (OR23) = (AO) = 1.

Dat gebeurt dan op 1 cijfertijd nadat signaal R3' laag werd. De flip-flop OF wordt in stand 1 gezet, indien de capaciteit van de registers wordt overschreden, dat wil zeggen bij positieve getallen, dat er een overdracht in de tekencijfers loopt en bij negatieve getallen dat er juist geen overdracht in de tekencijfers loopt. Daardoor vindt de tekenwisseling plaats en dus is het resultaat niet meer correct. Om te kunnen nagaan of er capaciteitsoverschrijding (overflow) is opgetreden of niet, wordt dit onthouden in de flip-flop OF. Hoe de getallen in de registers staan (als positief of als negatief getal) is hierbij

OF'/OF

1	Houd	OF'						
2	R3'	1ct'	CV25'	OI26	WTM	AO'	OR23	OR26
3	R3'	1ct'	CV25	OI26'	WTM'	AO'	OR23	OR26
4	R3'	1ct'	CV25'	TM	WTO	AO'	OR23'	OR26
5	R3'	1ct'	CV25	TM'	WTO'	AO'	OR23'	OR26

1 Houd OF
 2 terug R3' OR15' OR16 SO'

niet belangrijk. Van belang is alleen hoe deze getallen in het arithmetisch orgaan worden geïnterpreteerd. Vandaar dat nu

	WT0'	OI26	OR21	WTO	gebruik wordt gemaakt
		0	0	0	van signaal WTM' (blz. 121)
1	OR21'	OI26	0	1	en van signaal WTO'. Dit
		1	0	1	laatste signaal geeft het
2	OR21	OI26'	1	1	teken van het getal zoals

het in het coderend circuit OI wordt beschouwd en is laag als de cijfers (OI26) en (OR21) ongelijk zijn. Wanneer gold: (OF) = 1, werd de sprongopdracht uitgevoerd. De flip-flop OF wordt dan teruggesteld, zodra de arithmetische controlering in stand 3 komt (R3' is laag) bij deze opdrachten: (SO) = (OR15) = 1 en (OR16) = 0.

Het overslaan van een opdracht (skip)

Wanneer een opdracht moet worden overgeslagen, zal eerst de opdracht bekend moeten zijn, opdat op grond van die opdracht een niveau van plaatje SKIP' laag kan worden gemaakt. De opdracht wordt uit het geheugen gehaald, waarna op cijfertijd 2a' het functiegedeelte daarvan in het opdrachtregister wordt gezet. (signaal ORFI zie blz. 56 en het onderste tijdschema op tek.X-1 E31).

Lijst van afkortingen met verklaring

I

a	Teller
A	A-register
ABS	Absoluut
ACR	Arithmetische controle ring
AI	A-register in
AO	Additieve opdracht
AS	A-schuif
AUO	Additieve uit opdracht
b	Teller
B	B-register
BAO	Begin arithmetische operatie
BC	B-correctie
BCR	Band controle ring
BG	Blokkade GCR
BGA	Begin gekozen adres
BI	B-register in
BIN	B-register in normering
BP	Band pons opdracht
BRO	B-register opdracht
BS	Blokkade stop
BSO	Begin schuif operatie
BTL	Band- en type leesopdracht
C	Kloksignaal
CA	Clear A
CCA	Controle carry A
CCB	Controle carry B
CCC	Controle carry C
CCTR	Clear cijfer tijdring
CES	Controle einde schuif
CHB	Clear handbediening
CLG	Controle lees geheugen
CLR	Controle lees reproducer
CM	Clear M
CMF	Clear M functiegedeelte
CMI	Controle MI
CMIB	Controle M in band
CMIT	Controle M in type

CO	Communicatie opdracht
COA	Controle opteller A
COB	Controle opteller B
COC	Controle opteller C
COD	Controle opteller D
COI	Controle opteller in
COND	Conditie
COT	Clear opdrachtteller
CS	Clear S
CST	Clear slagenteller
ct	Cijfertijd
CTR	Cijfertijdring
CU	Clear U
CUI	Controle UI
CUIL	Controle UIL
CUILA	Controle UIL-A
CUIR	Controle UIR
CV	Carryvormer
CX	Correctie vermenigvuldiging
CZO	Conditiezettende opdracht
D	Tijdsignaal t.b.v. TPR
DI	Decimaal in
DKAC	Decimaal knopje aan-contact
DKG	Decimaal knopje gedrukt
DKS	Dood kastsignaal
DL	Drijfsignaal leesgeheugen
DRO	Dubbel register opdracht
DS	Drijfsignaal
DST	Drijfsignaal slagenteller
DU	Decimaal uit
ES	Einde schuif
FG	Fout getal
FO	Fout opdracht
FOUT	Fout
GCR	Geheugen controlering
gct	Geheugen cijfertijd
GCTR	Geheugen cijfertijdring

GG	Getal geheugen
GKoA	Geheugenkast o plug A
GKoB	geheugenkast o plug B (enz.)
GK1A	geheugenkast 1 plug A (enz.)
GTM	Gecorrigeerde teken M
HB	Handbediening
HBI	Handbediening in
HBS	Handbediening schoon
HBU	Handbediening uit
HS	Horizontale selectie
IO	In-opdracht
IS	Indicatie selectie
KC	Kortsluiting carry
KO	Knopjes op
KS	Kastsignaal
LA	Lees A
LB	Lees B
LC	Latente conditie
LO	Logische opteller
LOT	Lees opdrachtsteller
LPC	Lees parity digit
LR	Lees reproducer
LS	Lees S
M	M-register
MB	Gemodificeerd B
MI	M-in
MIB	M in band
MIT	M in type
MTA	Maal tien A
MTB	Maal tien B
NACO	Niet arithmetische communicatie opdracht
NAIO	Niet additieve in-opdracht
NAUO	Niet additieve uitopdracht
NBO	Normaal en B ongemodificeerd
NRW	Niet registerwijziging
NUL	U-nul
OF	Overflow
OI	Opteller in

OIV	Opteller in vermenigvuldiging
OKAC	Operationeel knopje aan-contact
OKG	Operationeel knopje gedrukt
OR	Opdrachtregister
ORAI	OR adresgedeelte in
ORFI	OR functiegedeelte in
OT	Opdrachtteller
OTIA	OT in
OTIB	OT in bij tellende en subroutinesprong
OTIC	OT in bij handbediening
PC	Parity check
PONS	Ponssignaal
PS	Plaatsignaal
R	Ringstand
RLGU	Reproducer lees gang uit
RLO	Reproducer lees opdracht
RLR	Reproducer lees ring
RP	Reproducer ponsregister
RPGU	Reproducer ponsgang uit
RPI	Reproducer pons in
RSR	Reproducer start ring
RTS	Reproducer tijdsignaal
S	S-register
SAR	Schuif A-register
SAS	Stop adresschakelaars
SC	Schuif
SCA	Stop combinatie adres
SCO	Schuifopdracht
SDO	Schuif- en dubbelregisteropdracht
SG	Selectie geheugen
SGA	Stop gekozen adres
SI	S-register in
SKIP	Skip
SO	Sprongopdracht
SOR	Selecteer opdrachtregister
SOT	Selecteer opdrachtteller
SSR	Schuifregister
ST	Slagenteller

STA	Slagenteller A
STB	Slagenteller B
STOP	Stop
STS	Slagenteller schoon
SV	Somvormer
SVA	Stop volgend adres
TA	Teken A
TB	Teken B
TBS	Teken B-selectie
TBL	Tijdsignaal bandlezer
TM	Teken M
TP	Type ponsregister
TPI	Type ponsregister in
TPR	Type ponsring
TPRS	Type ponsring thuis
TS	Teken S
TSS	Tellende en subroutine sprong
TYPE	Typesignaal
U	Uitregister
UI	U-register in
UIL	U-register in links
UIR	U-register in rechts
UO	Uit-opdracht
V	Versterker
VC	Vermenigvuldiging correctie
VO	Verboden opdracht
VS	Verticale selectie boven
VSO	Verticale selectie onder
WACHT	Wacht
WTM	Ware teken M
WTO	Ware teken van OI