

MATHEMATISCH CENTRUM  
2e BOERHAAVESTRAAT 49  
AMSTERDAM  
REKENAFDELING

Korte beschrijving van de opdrachtencode etc. voor ARMAC

door

E.W. Dijkstra

MR 23

1956

## VOORWOORD

Dit rapport is van een compactheid, grenzend aan onvollèdigheid. Men beschouwe het als een voornamelijk voor intern gebruik bestemd uittreksel van een uitvoeriger rapport over ARMAC, dat mettertijd verschijnen zal. De opstelling daarvan vergt echter meer tijd, dan wij ons nu kunnen permitteren. Om die redenen hebben wij ons hier praktisch beperkt tot het noteren van de verschillen tussen ARMAC en ARRA.

Het handregister (onafhankelijk van trommel-buffer-schakelaar).  
Details worden ingebracht door

1. de betrokken tekentoets in te drukken
2. de opeenvolgende decimale cijfers
3. na het laatste cijfer de machine door te starten.

Als dan de machine stopt, staat het bedoelde getal in S.

Opm.: Gehele getallen moeten in absolute waarde kleiner zijn dan  $2^{33} = 85899\ 34592$ . Breuken worden ingevoerd in maximaal 9 cijfers. Bij gehele getallen hoeven nonsignificante nullen aan het begin, bij breuken non-significante nullen aan het einde niet aangeslagen te worden. Het doorstarten na het laatste cijfer is verplicht, ook als men halverwege tot de ontdekking komt, dat het helemaal niet de bedoeling is, om een specifiek getal in S te brengen.

Opm.: Als men een fout gemaakt heeft, begint men (bij het teken) opnieuw.

#### De automatische start.

De toetsen van het handregister worden alleen als cijfer verwerkt, als na de laatste indrukking van een tekentoets nog niet is doorgestart voor afmaking. Anders - normaal - starten zij de machine. Voorlopig:

- 0: start invoerprogramma schrijvend
- 1: start invoerprogramma controlerend
- 9: start binair ponsprogramma.

#### Het binair ponsprogramma.

Het eerste uit te ponsen adres moet in de a-helft van de getal-schakelaar staan; de lengte van de biband in de b-helft. Start men dan met "9", dan wordt de band geponst en afgesloten door een pentade blank (eveneens voor "run out"); start men dan door dan wordt geponst R.C, blank. (voor "run out"); start men ten tweede male door, dan komt na roffel blank EE blank. Een zelf-controlerende biband nu maakt men door:

start 9	band
start door	RC blank
start 9	band
start door	RC blank
start door	roffel blank, EE, blank
run out	roffel blank.

Korte beschrijving der ponsconventies voor ARMAC

1. De 32 toetsen van de telexbandponser zijn voorzien van de volgende etiketten (in kolom 1: staat het eerste symbool = bandwaarde der betrokken pentade, in kolom 2: staat het tweede symbool, indien aanwezig):

<u>1:</u>	<u>2:</u>	<u>1:</u>	<u>2:</u>	<u>1:</u>	<u>2:</u>	<u>1:</u>	<u>2:</u>
0		8		16	A	24	J
1		9		17	B	25	K
2		10	00	18	C	26	L
3		11	000	19	D	27	M
4		12	+	20	E	28	P
5		13	-	21	F	29	Q
6		14	+	22	G	30	R
7		15	-	23	H	31	X

De eerste tien zijn dus enkel-, de laatste twee en twintig dubbelbenaemd.

2. Indien het invoerprogramma leest (d.w.z. successieve pentades assembleert tot woorden van 34 binaire cijfers) moet van te voren het volgende gespecificeerd zijn:
1. de wisselstand, d.w.z. als het woord uit de gelezen pentades is opgebouwd, kan het
    - a) in het geheugen geschreven worden
    - b) met inhoud van het geheugen vergeleken worden.
 a) noemen we de "schrijfstand": hier vindt de werkelijke invoer (van band naar geheugen) plaats. b) noemen we de "controlestand": als bij vergelijking een verschil optreedt, stopt de machine. Zodoende is enige controle op de invoermogelijk.
  2. de plaats, d.w.z. het adres in het geheugen, waarop het van de band opgebouwde woord moet (komen te) staan. Opeenvolgende moleculen hebben, tenzij anders aangegeven, op successieve adressen betrekking; de plaats hoeft dus steeds alleen voor het eerste molecuul van een rijtje expliciet gespecificeerd te worden.
  3. het soort, d.w.z. de regels, die de pentadeassemblage tot een geheel molecuul beheersen. Deze regels zijn nl. voor verschillende soorten moleculen verschillend. Thans kent het invoerprogramma 3 soorten moleculen, nl. getallen, opdrachtenkoppels en binaire moleculen.

Naast de moleculen komen voor de zg. controlecombinaties: dit zijn o.a. speciale pentade-groepen, die de drie genoemde specificaties omschrijven en wijzigen. Tenslotte vindt men op de

band:

pentades X (vijf gaatjes): deze worden aan het begin van elk (niet binair) molecuul geskipt. Deze faciliteit is ingelast om tijdens het ponsen ontdekte ponsfouten te kunnen corrigeren.  
pentades blank (= 0, geen gaatjes): deze mogen aan het begin van een band en na bepaalde combinaties ingelast worden. Deze faciliteit is ingelast om het inleggen van de band in de bandlezer te vergemakkelijken en om verschillende stukken van de band te kunnen scheiden; het ontbreken van ponsingen maakt hier de band gemakkelijk beschrijfbaar.

3. Omdat het de bedoeling is, binaire moleculen niet zelf te ponsen maar dit met behulp van een ponsprogramma door ARMAC te laten doen, beperken wij ons in de nu volgende beschrijving van de ponsconventies voor de moleculen tot de twee andere soorten.

Om getallen te kunnen lezen, moet het soort gespecificeerd zijn door de controle-combinaties RG (zie later). Het invoerprogramma kan dan gehele getallen en echte breuken lezen; zij bestaan uit een teken: (+ of - voor gehele getallen, +. of -. voor echte breuken), gevolgd door het decimale gedeelte. Voor het decimale gedeelte mag men van twaalf voetsen gebruik maken, nl. 0 t/m 9, 00 (dubbel nul) en 000 (tripel nul); deze twee laatste toetsen zijn er aan toegevoegd om pons tijd te verkorten. Gehele getallen moeten in absolute waarde kleiner zijn dan  $2^{33} = 85899\ 34592$ , de breuken in absolute waarde (uit de aard der zaak) kleiner dan 1. Bij gehele getallen moeten nonsignificante nullen aan het begin weggelaten worden tot en met een nul op de tientallenplaats, m.a.w. er moet minstens 1 cijfertoeets aangeslagen worden. Breuken worden ingevoerd in maximaal 9 decimalen achter de komma, desgewenst in minder: nonsignificante nullen aan het einde hoeven dus niet geponst te worden.

N.B. Breuken worden niet exact ingevoerd! De meeste decimale breuken repeteren in het twee-talig stelsel, en kunnen dus niet exact ingevoerd worden, maar ook een breuk, die door de machine wel exact kan worden voorgesteld, is soms  $2^{-33}$  mis. (B.v. +.75 komt als  $+.75-2^{-33}$  in de machine.)

Om opdrachtenkoppels te kunnen lezen, moet het soort gespecificeerd zijn door de controle-combinatie RD (zie later). Het is onmogelijk één losse opdracht in te lezen: bij elkaar behorende a- en b-opdracht worden altijd samen ingevoerd. Van een opdrachtenpaar wordt eerst de a-opdracht, dan de b-opdracht geponst, beide volgens dezelfde conventies. Zo worden b.v. ook aan het

begin van de b-opdracht - dus midden in een molecuul! - eventuele extra pentades X geskipt. Elke opdracht wordt in tweeën geponst: functiegedeelte gevolgd door adres (of algemener: numeriek gedeelte; omdat het echter voor de ponsconventies geen verschil maakt, zullen wij dit gedeelte van de opdracht met de (kortste) naam "adres" blijven aanduiden." Omdat op zijn beurt het adres in drieën wordt geponst, bestaat elke opdracht totaal (in volgorde) uit de onderdelen:

f = functie

p = plaats

s = sluitletter

k = kanaalcorrectie

} adres

ad f: het functiegedeelte loopt van 0 t/m 29 en wordt altijd in 1 pentade geponst

ad p: de plaats wordt geponst in minstens 1 pentade. Als  $p \leq 31$  (dit is regel), wordt p in 1 pentade geponst; als  $p \geq 32$ , p onst men de successieve decimale cijfers, waarbij ook dubbel- en tripelnul - mits niet aan het begin - gebruikt mogen worden. (Als  $p \leq 31$ , mag de plaats ook wel in twee pentades geponst worden: als  $p \geq 32$ , maar het getal, gevormd door de twee kopcijfers is  $\leq 31$ , dan mogen die twee ook wel in 1 pentade geponst worden. Het is beter, hier geen gewoonte van te maken).

ad s: de sluitletter is een van de symbolen A,B,C,D,E,F,G,H,J, K,L,T,P,S,R of X en wordt dus altijd in 1 pentade geponst

ad k: de kanaalcorrectie loopt van 0 t/m 31, en wordt altijd in 1 pentade geponst.

De 16 sluitletters markeren beginpunten (adressen), relatief ten opzichte waarvan men de adressen nummert. Het effect van een sluitletter is, dat (het adres van) het bijbehorende beginpunt opgeteld wordt bij het door p en k in eerste instantie omschreven adres =  $p + 32k$ . De kanaalcorrectie geeft dus aan het aantal 32-vouden, dat bij p moet worden opgeteld: dit is ook de reden, waarom doorgaans  $p \leq 31$  is. De sluitletter X markeert altijd adres 0, dus het begin van het geheugen. X laat dus het adres  $p + 32k$  onveranderd. De 15 andere additieve parameters mag de programmeur zelf kiezen: zij worden aan de machine meegedeeld door speciale controle-combinaties (de vulindicaties, zie later) op de zg. voorponsing, die ingelezen moet worden voor de band, waarop de sluitletters gebruikt worden.

4. Voordat ARMAC in het invoerprogramma gestart wordt, moet de band met het stuk blank onder de bandlezer worden gelegd. De wijze van

starten bepaalt de wisselstand: men start het invoerprogramma met de toets 0 van het handregister in de schrijfstand, met de toets 1 in de controlestand. Wat betreft de soort-specificatie is het invoerprogramma nog in het ongewisse: deze is daarom (door de start) ingesteld op "skip blank, X tot R", d.w.z. de machine gaat band lezen, skipt blank (indien aanwezig) tot aan X of R; als X gelezen mocht zijn, skipt de machine alsnog deze en eventuele verdere pentades X tot R. Komt er nu een pentade  $\neq$  R, dan stopt de machine: bij doorstarten wordt deze pentade evenwel als R geïnterpreteerd. R is nl. de aankondigingspentade voor controle-combinaties. Voordat er moleculen op de band verschijnen, moet immers nog plaats en soort gespecificeerd worden. Omdat deze specificeringen door controlecombinaties worden beschreven, "eist" het invoerprogramma de pentade R. Dit is een van de voorbeelden, waar het invoerprogramma de band "nakijkt". Zo stopt ARMAC bv. eveneens, als de plaats niet gespecificeerd wordt.

5. Tot nog toe zijn de volgende controle-combinaties toegestaan. (Voor elke R wordt een willekeurig aantal pentades X geskipt.)

#### RA (Adresindicatie)

De plaats van het eerstvolgende echte molecuul wordt gespecificeerd door onmiddellijk achter RA het adres (in drieën, als bij de opdracht) te ponsen, waar het eerstvolgende echte molecuul moet (komen te) staan. Afhankelijk van de wisselstand zal er worden geschreven of vergeleken. Evenals de wisselstand, blijft ook de soort-specificatie hierbij onveranderd, d.w.z. was de soort nog ongespecificeerd, dan komt de besturing na afloop terug in "Skip blank, X tot R"; was de soort al gespecificeerd als "getallen" dan is het programma klaar om nu de getallen te lezen, evenzo met opdrachten. Voor binaire moleculen geldt dit niet (zie onder)

#### RB (Biband)

De controle-combinatie RB stelt de soort-specificatie in op binaire moleculen; de uitzonderingspositie van RB bestaat hierin, dat bij het lezen van bibanden ARMAC "blind" is voor alle controlecombinaties op één na: de indicatie dat het einde van de biband bereikt is. (Omdat een binair molecuul nooit met een pentade = blank begint, kan dit een pentade blank na de laatste pentade van het laatste binaire woord zijn.) Deze blindheid impliceert, dat voor het speciale geval van de biband de soort (RB) pas gespecificeerd mag worden, als de plaats vast ligt. Na

afloop komt de besturing terug bij "Skip blank, X tot R", wel nieuwe soort-specificatie eisend, maar niet nieuwe plaats; tenzij anders op de band aangegeven is, heeft het volgende echte molecuul normaal betrekking op het volgende adres.

#### RC (Controlewissel)

Door de controlecombinatie RC wordt de stand van de controlewissel omgezet; deze omzetting wordt pas effectief bij de eerstvolgende RA-combinatie. Na RC komt de besturing terug in "Skip blank, X tot R", zowel plaats; als soortspecificatie eisend.

#### RD (lees opdrachten)

De controle-combinatie RD stelt de soort-specificatie in op opdrachten. Het invoerprogramma leest daarna geen blank tape, getallen of binaire moleculen, slechts opdrachten en controle-combinaties. RD laat wisselstand en plaatsbepaling onbeïnvloed.

Aan het begin van een band met opdrachten mag RD dus evengoed voor als na de RA-combinatie komen; als na een rij getallen daarbij aansluitend een rij opdrachten moet worden ingelezen, hoeven wij voor de eerste opdracht slechts RD in te lassen.

#### RE (Einde band)

Met behoud van de wisselstand gaat ARMAC over in "Skip blank, X tot R", beide andere specificaties eisend. Deze controlecombinatie is meer de aankondiging van het stuk blank, dat volgt. Als men de band af wil scheuren, ponst men achter het laatste stukje blank EE, de stoppende versie van RE; maakt men met behulp van de reproducer van vele kleine een lange band, dan laat men deze EE-combinatie midden in steeds weg. De EE-combinatie dient tevens om begin en eind van de band te onderscheiden.

#### RF (Vulindicatie, adres)

Om voor de sluitletters met variable betekenis (dus: A,B,C,D,E,F,G, H,J,K,L,T,P,S en R) de bijbehorende additieve parameters gelijk te maken aan een of ander adres, gebruikt men de zg. vulindicatie RF; hierna ponst men de letter, waaraan de nieuwe waarde toegekend wordt, daarna wordt de nieuwe parameter als adres (dus plaats, sluitletter, kanaalcorrectie) geponst. (De betekenis van de in dit adres gebruikte sluitletter, indien  $\neq$  X, moet door een eerdere voorponsing zijn ingevoerd!) De vulindicatie verandert niets aan de wisselstand, de plaats van wegbergen of de soortspecificatie.

Om aan een sluitletter-parameter een waarde, anders dan een adres - dus b.v. een opdrachtenkoppel, wat voor speciale doeleinden



soms gewenst is - toe te kennen, gebruikt men de controle-combinatie RH (zie onder).

#### RG (Lees getallen)

De controle-combinatie RG stelt de soort-specificatie in op getallen. Het invoerprogramma leest daarna geen blank tape, opdrachten of binaire moleculen, slechts getallen (vier types!) en controle-combinaties. RG laat wisselstand en plaatsbepaling onbeïnvloed. Aan het begin van een band met getallen mag RG dus evengoed voor als na de RA-combinatie komen; als na een rij opdrachten daarbij aansluitend een of meer getallen moeten worden ingelezen, hoeven wij voor het eerste getal slechts RG in te lassen.

#### RH (Vulindicatie, molecuul)

Om voor de sluitletters met variabele betekenis de bijbehorende additieve parameters gelijk te maken aan iets anders dan een adres - dus b.v. een getal of een opdrachtenkoppel - gebruikt men de zg. vulindicatie RH; hierna ponst men de letter, waaraan de nieuwe waarde toegekend wordt, daarna (indien nodig) een soortspecificatie, gevolgd door het molecuul. De vulindicatie RH verandert niets aan de wisselstand, de plaats van opbergen of de soortspecificatie, zoals deze was, toen RH gelezen werd. De tussen RH en het volgend molecuul geponste soortspecificatie is dus alleen op dit volgende molecuul van toepassing. Ontbreekt deze soortspecificatie, dan wordt een molecuul volgens het "op de hoofdband" gespecificeerde soort gelezen. N.B. Getallen moeten in de RH-combinatie door een pentade X worden afgesloten!

#### RJ (Jump)

Na RJ wordt (ongeacht de gespecificeerde soort) één opdracht gelezen; deze wordt vervolgens uitgevoerd. Aan het feit, dat deze opdracht altijd een sprong-opdracht zal zijn, ontleent deze controle-combinatie zijn naam. Wisselstand soort en plaats blijven onbeïnvloed.

#### RX (Skip)

De controle-combinatie RX wordt altijd gevolgd door 1 pentade, die gelijk is aan het aantal adressen, dat wordt overgeslagen bij het wegbergen of vergelijken: als in een programma een adres opengelaten wordt - omdat dit door het programma zelf ingevuld wordt - ponst men die op die plaats dus RX1. (Deze faciliteit is overbodig; men zou met RA, gevolgd door een adres hetzelfde

hebben kunnen bereiken) RX wijzigt de plaats, verandert niets aan wisselstand en soort. RX mag niet gegeven worden, als de plaats nog met RA gespecificeerd moet worden; doet men dit toch, dan stopt de machine.

6. Men kan zelfcontrolerende banden als volgt maken: de band begint met een roffel blank, plaats- en soortspecificatie van het eerste molecuul, dan de moleculen ( en eventuele controle-combinaties, als moleculen van verschillend soort geassembleerd moeten worden, etc.); na het laatste molecuul ponsst men RC, gevolgd door een roffeltje blank; dan herhaalt men de band tot zo ver (d.w.z. van het roffeltje blank aan het begin tot en met de roffel blank na RC); na deze pentades blank ponsst men EE, daarna weer roffel blank, waarin de band afgescheurd kan worden. Assembleert men de zelfcontrolerende banden tot een geheel, dan wordt de EE-combinatie, behalve aan het einde, steeds weggelaten.
7. Met de kanaalcorrectie bestrijkt men 32 kanalen, dus slechts een kwart van het geheugen: om een adres in het laatste driekwart van het geheugen aan te geven, moeten we dus iets van de tafel van 32 kennen: dit wordt beperkt tot de voorponsing! Het lijkt verstandig, zich hier te beperken tot de tafel van 1024: als b.v. kanaal B0 kanaal 73 moet worden, ponsse men in de voorponsing: RFB 2048 X9. We zien dan omgekeerd onmiddellijk, dat met  $B0 \ 2 \times 32 + 9 = 73$  bedoeld is.

#### Korte beschrijving van de opdrachtencode voor ARMAC

De opdrachten worden, waar het numeriek gedeelte een echt adres is, door f/n genoteerd. Waar een uitdrukking met een vraagteken staat, wordt de conditie positief gezet, als het antwoord op de vraag bevestigend is, negatief als het antwoord ontkennend is. Als hier een geheugenplaats of register in is genoemd, betreft het de nieuwe inhoud.

(Lees voor het symbool #: "vervangt")

- 0/n (A) + (n) ≠ (A)
- 1/n (A) - (n) ≠ (A)
- 2/n + (n) ≠ (A)
- 3/n - (n) ≠ (A)
- 4/n + (A) ≠ (n); (n) ≥ + 0? of (A) ≥ + 0?
- 5/n - (A) ≠ (n); (n) ≥ + 0? of (A) ≤ - 0?
- 6/n spring naar n, a-opdracht
- 7/n spring naar n, b-opdracht
- 8/n (S) + (n) ≠ (S)
- 9/n (S) - (n) ≠ (S)
- 10/n + (n) ≠ (S)
- 11/n - (n) ≠ (S)
- 12/n + (S) ≠ (n); (n) ≥ + 0? of (S) ≥ + 0?
- 13/n - (S) ≠ (n); (n) ≥ + 0? of (S) ≤ - 0?
- 14/n als conditie positief, spring naar n, a-opdracht
- 15/n als conditie negatief, spring naar n, b-opdracht
- 16/n (A) + (n).(S) ≠ (AS)
- 17/n (A) - (n).(S) ≠ (AS)
- 18/n + (n).(S) ≠ (AS)
- 19/n - (n).(S) ≠ (AS)
- 20/"n" Tracktransport Trommel → Matrix
- 21/"n" Tracktransport Matrix → Trommel
- 22/n link ≠ (A), spring naar n, a-opdracht
- 23/n link ≠ (A), spring naar n, b-opdracht
- 24/n }  
⋮ } Schuif- en communicatieopdrachten.  
29/n }

ad "20" en "21": de opdrachten voor de tracktransport gebruiken de registers A en S niet, edoch steeds een kanaal (matrix) uit het snelle geheugen en een kanaal (spoor) op de trommel. De inhoud van het ene kanaal wordt in het andere gecopieerd, het functiegedeelte bepaalt de richting van het transport, het numeriek gedeelte als volgt, op welke kanalem het transport betrekking heeft: de hoogste 7 cijfers van het "adres" bepalen het spoor op de trommel (het getal, door deze gevormd, moet dus minstens 16 zijn) de laagste 4 bepalen het nummer van het snelle kanaal. Hiertussen staat een ongebruikt cijfer.

ad "22" en "23": dit zijn de zg. "subroutineaanroepen": de opdrachtsteller wordt uitgelezen: in de a-helft van het A-register komt de 6- of 7-sprong naar de onmiddellijk volgende opdracht, in de b-helft worden 17 nullen geplaatst. Vervolgens

wordt de sprong naar de a- resp. b-opdracht van adres n uitgevoerd.

ad "24" t/m "29": bij de schuif- en communicatie-opdrachten is het numeriek gedeelte nooit een adres, maar omlijnt het anderszins de functie. Wij noteren hier de opdrachten in overeenstemming met de standaardponsconventies.

Nultest op ongetekende nul

28	0	XO: $A \neq 0?$	28	0	X8: $S \neq 0?$
29	0	XO: $A = 0?$	29	0	X8: $S = 0?$

Handregister H (komt alleen in het invoerprogramma voor)

24	1	XO: $(A) + (H) \neq (A)$	24	1	X8: $(S) + (H) \neq (S)$
25	1	XO: $(A) - (H) \neq (A)$	25	1	X8: $(S) - (H) \neq (S)$
26	1	XO: $+ (H) \neq (A)$	26	1	X8: $+ (H) \neq (S)$
27	1	XO: $- (H) \neq (A)$	27	1	X8: $- (H) \neq (S)$

Getalschakelaar G

24	2	XO: $(A) + (G) \neq (A)$	24	2	X8: $(S) + (G) \neq (S)$
25	2	XO: $(A) - (G) \neq (A)$	25	2	X8: $(S) - (G) \neq (S)$
26	2	XO: $+ (G) \neq (A)$	26	2	X8: $+ (G) \neq (S)$
27	2	XO: $- (G) \neq (A)$	27	2	X8: $- (G) \neq (S)$

Bandlees en -ponsopdracht

24	4	XO: $(A) + (B) \neq (A)$	24	4	X8: $(S) + (B) \neq (S)$
25	4	XO: $(A) - (B) \neq (A)$	25	4	X8: $(S) - (B) \neq (S)$
26	4	XO: $+ (B) \neq (A)$	26	4	X8: $+ (B) \neq (S)$
27	4	XO: $- (B) \neq (A)$	27	4	X8: $- (B) \neq (S)$
28	4	XO: $+ (A) \neq (U) \neq (B);$ $(A) \geq + 0?$	28	4	X8: $+ (S) \neq (U) \neq (B);$ $(S) \geq + 0?$
29	4	XO: $- (A) \neq (U) \neq (B);$ $(A) \leq - 0?$	29	4	X8: $- (S) = (U) = (B);$ $(S) \leq - 0?$

Opm.: De bovenste vier regels zijn bandleesopdrachten: het "symbolische" register B bevat aan de hoge kant 29 nullen, de laagste vijf cijfers worden van de telexband gelezen, die daarna een pentade opschuift. De onderste twee zijn band-pons-opdrachten: al of niet met tekenwisseling (0-1 inversie) worden de laagste vijf cijfers van A of S geponst. Dit gaat via een tussenregister U, dat de laagste zes cijfers van A of S overneemt.

Deze opdrachten zetten tevens de conditie.

Typ-, en terugleesopdrachten

24	8	X0: (A) + (U) ≠ (A)	24	8	X8: (S) + (U) ≠ (S)
25	8	X0: (A) - (U) ≠ (A)	25	8	X8: (S) - (U) ≠ (S)
26	8	X0: + (U) ≠ (A)	26	8	X8: + (U) ≠ (S)
27	8	X0: - (U) ≠ (A)	27	8	X8: - (U) ≠ (S)
28	8	X0: + (A) ≠ (U); (A) ≥ + 0?	28	8	X8: + (S) ≠ (U); (S) ≥ + 0?
29	8	X0: - (A) = (U) (A) ≤ - 0?	29	8	X8: - (S) = (U) (S) ≤ - 0?

Opm.: De onderste twee regels zijn de typopdrachten: er wordt een symbool getypt, gespecificeerd door de laagste 6 cijfers van  $\pm$  (A) of  $\pm$  (S). Zij zijn weer conditiezettend. De eerste vier regels beschrijven de mogelijkheid om het laatst getypte (of geponste) symbool voor controledoeleinden terug te lezen.

Conditionele stopopdrachten

- 26 16 X0 (en 26 16 X8): stop, als de conditie positief is.  
 27 16 X0 (en 27 16 X8): stop, als de conditie negatief is.

Bufferschrijfopdrachten op plaats n in de buffer (0 ≤ n ≤ 31)

28	n	X2: + (A) ≠ buffer; (A) ≥ +0?	28	n	X10: + (S) ≠ buffer; (S) ≥ +0?
29	n	X2: - (A) ≠ buffer; (A) ≤ -0?	29	n	X10: - (S) ≠ buffer; (S) ≤ -0?

De adresloze optelling (0 ≤ n ≤ 127)

24	n	X4 (A) + n ≠ (A)	24	n	X12 (S) + n ≠ (S)
25	n	X4 (A) - n ≠ (A)	25	n	X12 (S) - n ≠ (S)
26	n	X4 + n ≠ (A)	26	n	X12 + n ≠ (S)
27	n	X4 - n ≠ (A)	27	n	X12 - n ≠ (S)

De schuifopdrachten

De cijfers in een bepaald "schuifcircuit" (zie onder) worden een bepaald aantal plaatsen naar rechts geschoven; dit kan op vier wijzen, bepaald door het functiecijfer:

f = 24, de additieve schuif: de cijfers, die aan de lage kant het circuit verlaten, komen aan de hoge kant in het tekencijfer weer binnen;

f = 26, de schone schuif: de cijfers, die aan de lage kant het circuit verlaten, gaan verloren; aan de hoge kant wordt met het tekencijfer aangevuld.

f = 28, de pos. uit-schuif: arithmetisch als bij f = 24, maar tevens wordt de conditie gezet op het nieuwe tekencijfer

van het circuit.

$f = 29$ , de neg. uit-schuif: arithmetisch als bij  $f = 24$ , maar tevens wordt de conditie gezet op de inverse van het nieuwe teken van het circuit.

Het aantal cijferposities, waarover geschoven wordt, geeft men aan in het plaatsgedeelte van de opdracht; het aantal is maximaal 35. Het einde van de opdracht, (de sluitletter X en) de kanaal-correctie, bepaalt het circuit; er zijn vier mogelijkheden:

- X20: A→A: het circuit begint bij het tekencijfer van A en eindigt aan de lage kant van A.
- X22: S→A: het circuit begint bij het tekencijfer van S; aan de lage kant van S wordt het voortgezet met het hoogste (= teken-) cijfer van A en het eindigt aan de lage kant van A. ("Schuif van S naar A door het teken van A heen".) Het tekencijfer van S fungeert als tekencijfer van het circuit.
- X28: A→S: het circuit begint bij het tekencijfer van A, aan de lage kant van A wordt het voortgezet met het op een na hoogste cijfer van S en het eindigt aan de lage kant van S. ("Schuif van A naar S, onder het teken van S door") Het tekencijfer van A fungeert als tekencijfer van het circuit. Het tekencijfer van S blijft ongewijzigd.
- X30: S→S: het circuit begint bij het tekencijfer van S en eindigt aan de lage kant van S.