

RA

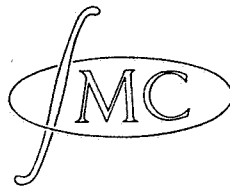
STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM
REKENAFDELING

R 917 a

Addendum bij rapport R 917, X8-simulator, simulatieprogramma
voor het uitvoeren van X8-programma's met behulp van de X1

door

F.J.M. Barning

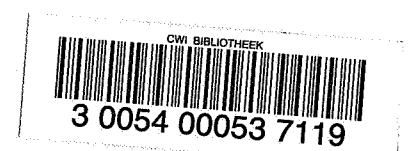


december 1964

RA

F1. 1,75

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM



Inhoud:

0. Inleiding
1. Uitbreiding en herziening testfaciliteiten
2. Aanpassing simulator aan recente gegevens over de X8
3. Opheffing van enkele beperkingen
4. Localisatie aanvullende programma's
5. Slotopmerkingen

0. Inleiding

Dit rapport dient ter aanvulling en ter correctie van hetgeen over de ontwikkelde X8-simulator is medegedeeld in het voorlopige rapport R 917. Het bevat o.m. een uitbreiding en herziening van de testfaciliteiten voor het testen van X8-programma's (par. 1), een aanpassing van het simulatieprogramma aan recente gegevens over de X8 (par. 2), alsmede enkele veranderingen, die dienen ter opheffing van zekere beperkingen in het gebruik van de simulator (par. 3).

Bij elk van de beschreven programma's is aangegeven op welke bladzijde van R 917 de wijziging betrekking heeft.

De correctiebanden kunnen eenvoudig worden toegevoegd aan de biband van de simulator (programma in R 917), mits voorafgegaan door diens voorponsing. Deze voorponsing, zoals die geldt voor de X1-6 van het M.C. (zie ook R 917, blz. 12) wordt, voor zover van belang, hieronder weergegeven.

De paragraaf ZE, in de correctieprogramma's gereserveerd voor de localisatie van de aanvullende programma's, dient vooraf gedefinieerd te worden.

De localisatie van de aanvullende programma's, zoals die het geval is bij de X1-6 van het M.C., is in par. 4 aangegeven.

In par. 5 treft men enkele nadere bijzonderheden aan over het gedrag van een paar speciale X8-opdrachten, en een aanvullende mededeling over de tijden nodig voor de uitvoering van X8-opdrachten in het algemeen, d.m.v. de simulator.

Het rapport geeft tenslotte nog een aantal correcties, m.b.t. de tekst van R 917.

Voorponsing:

DPZU	10240X0
DPZN	11008X0
DPZZ	11264X0
DPZR	2ZZ4
DPZL	30ZR1

DPZF	11ZL2
DPZT	23ZF5
DPZH	1ZT1
DPZS	1ZH3
DPZK	26ZS2
DPZW	16ZK6
DPZY	23ZN0
DPEE	14ZY1
DPEK	29EE2
DPER	19EK2

1. Uitbreiding en herziening testfaciliteiten

1.1. De nog vrij zijnde meervoudige autostart G9 (zie R 917, blz. 4, opm. 2), kan gebruikt worden om de mogelijkheid te realiseren van het uitvoeren van een X8-opdracht, die zich in binaire vorm bevindt in de getalschakelaars van de X1.

Autostart G9 heeft tot effect, dat de in de getalschakelaars gespecificeerde opdracht eenmalig wordt uitgevoerd, en dat in geval deze opdracht niet een niet te skippen sprongopdracht is, de opdrachtteller niet verandert.

Na uitvoering van de opdracht stopt de X1 op de stop 7Y8C0, (zoals ook het geval is bij BVA, SGA enz. (zie R 917, blz. 2,3 onder b), c) en d)). In S vindt men na uitvoering de uitgevoerde (evt. gemodificeerde) opdracht, en in B de opdrachtteller (uitgezonderd in het algemeen na uitvoering van pseudo=typopdrachten, zie R 917, blz. 9 onder pt 16, waarbij geëindigd wordt met herstel van de status quo). De conditie, het laatste teken, en de overflow-indicatie, zoals die luiden na uitvoering van de opdracht, vindt men gerepresenteerd in de corresponderende kleine registers van de X1.

Een en ander wordt geëffectueerd door het volgende programma:

programma 1.1:

		DA	0	Z Y 0 DI		correctie R 917, blz. 50
4ZZO	=)	0	3B	0	A P	voorbereiding
		1	7B	8	X 1	zetting NINT(=0)
			DA	6	Z Y 0 DI	correctie R 917, blz. 50
6ZZO	=)	6	2A	3	D 0 P	geen inspectie? afwerking
		7 Y	2S	27	X 0 Z	geen handregister-opdracht?
		8 Y	2T	11	X 0 E	→
		9	2S	27	X 0 Z	geen handregister-opdracht?
		10 N	2T	25	Y 0 A	→
		11	2B	62	U 0	OT
		12 U	2LA	3	A Z	subroutine uitlopen?
		13 Y	2T	1	Y 1 A	→
		14	2LA	1	A Z	niet SVA?
		15 N	2T	21	Y 0 A	→
		16	2A	5	X 1 Z	bij SGA in, overeenkomst met stop-
		17 N	3A	12	X 1 P	stopopdracht? adres?
		18	1B	2	D 0	
		19	6B	5	X 1	
		20	2T	7	Y 1 A	⇒
15	⇒	21	3A	11	X 1 P	subr.sprong of herst.sprong?
		22 Y	2A	17	X 1 P	subr.sprong?
		23 Y	2B	7	X 1	
		24 Y	6B	6	X 1	reserveer terugkeeradres
10;3,7ZY1	→	25	2B	62	U 0	OT
		26	2S	10	X 1	opdracht
		27	2A	11	X 0	link
		28	3LA	32767	A	
		29	6A	11	X 0	
		30	OT	11	X 0 P	⇒
30	⇒	31	7Y	8	C 0	herstel kleine registers
						stop

		DA	0	Z Y 1 DI		correctie R 917, blz. 51
		0	2T	8	Z 0 A	⇒
13ZY0	⇒	1	3A	12	X 1 P	stopopdracht?
		2 Y	6A	12	X 1	indicatie
		3 Y	2T	25	Y 0 A	→
		4	3A	11	X 1 P	subr.sprong of herst.sprong?
		5 Y	3A	17	X 1 P	herst.sprong?
		6 Y	1B	6	X 1 Z	stoppen na terugkeer?
20ZY0	→	7 Y	2T	25	Y 0 A	→
		8	2T	11	X 0 E	⇒

(vervolg progr. 1.1):

		DA	4	E E 2 DI	correctie R 917, blz. 54
	4	OX	0	Z E 0 A	
		DA	15	Z Z 0 DI	correctie R 917, blz. 13
	15	6T	7	E 0 0	=)
	16	OT	0	A	=>
		DA	25	Z F 4 DI	correctie R 917, blz. 27
	25	2T	10	F 0 A	=>
13ZE0 =>	26	2B	62	U 0	OT
	27	1B	1	A Z	met 1 aflagen
	28 Y	5P		B B	
	29	6B	62	U 0	
		DA	16	Z F 1 DI	correctie R 917, blz. 24
	16 Y	2T	11	E 0 A	→
		DA	0	Z E 0 DI	
26EE1 =>	0	2S	3	D 0	opdracht bij toep. G9
	1	2A	23	X 0	link
	2	3LA	32767	A	
	3	OLA	0	D15	verwijder stopbit
	4	6A	11	X 0	
	5	OT	11	X 0 P	=>
5 =>	6	2T	11	Z 0 A	=>
15ZZ0 =)	7 U	2S	27	X 0 Z	geen handregister-opdracht?
	8 Y	2B	1	A	
	9 Y	4B	62	U 0	verhoog OT met 1
	10	2T	8	X 0 E	=>
16ZF1 =>	11 U	2S	27	X 0 Z	geen handregister-opdracht?
	12 N	2T	30	F 4 A	→
	13	2T	26	F 4 A	=>
		DC			
		DO			

- Opm. 1. Met betrekking tot de stopmogelijkheden, genoemd in R 917, blz. 2,3 onder b) en c), wordt opgemerkt, dat voor "stoppen op het volgende adres" in een X8-programma, in het algemeen slechts de mogelijkheid onder b) ten dienste staat, en niet die onder c), indien men het stoppen zou wensen te realiseren door plaatsing van het volgende adres in het stopadres van de X1.
2. De stopmogelijkheid genoemd in R 917, blz. 3 onder d), te gebruiken voor het "uitlopen" van een subroutine, reageert alleen op normale terugkeer op het adres, direct volgend op het adres, vanwaar de subroutine is aangeroepen.

3. Toepassing van bovenstaand programma heeft tot effect, dat bij stoppen (op stop 7Y8C0) volgens een der stopmogelijkheden b), c) of d), niet alleen de conditie (zie R 917, blz. 2 en 3), doch ook het laatste teken en de overflow in de betreffende registers van de X1 overgenomen worden.
 4. Gebruikt men autostart G9 om ergens in een X8-programma opdrachten voor uitvoering in te lassen, dan kan na uitvoering het programma normaal worden doorgestart met H, BVA. Wenst men door te starten op een adres n, ongelijk aan OT, dan bedenke men dat toepassing van autostart 0 met n in het startadres van de X1, tot gevolg heeft, dat de inhoud van de kleine registers in het algemeen verandert. Dit kan worden voorkomen door autostart G9 toe te passen met in de getalschakelaars de opdracht 2TnA en na uitvoering hiervan door te starten met H, BVA.
 5. Zie 2.5.
- 1.2. Om het testen van X8-programma's verder te vergemakkelijken, is het simulatieprogramma uitgebreid met een aantal stoppen. De thans ingebouwde stoppen 7Y3C0-7Y8C0 geven de volgende indicaties:
- stop 3: 7Y3C0 : ~~***~~ bij aangeboden niet te skippen stapelende subroutinesprongen (subroutinesprongen "nieuwe stijl"): de stapelwijzer B voldoet niet aan $0 \leq B < 2 + 15$
- stop 4: 7Y4C0 : bij aangeboden niet te skippen sprongopdrachten: de nieuwe OT voldoet niet aan $0 \leq OT < 2 + 15$.
- stop 5: 7Y5C0 : bij aangeboden niet te skippen opdrachten die indirect zijn geadresseerd:
 $L+p$ ($L=(63X0)$) voldoet niet aan $0 \leq L+p < 2 + 15$.
- stop 6: 7Y6C0 : bij aangeboden niet te skippen opdrachten, die indirect (incl.B-gecorrigeerd) zijn geadresseerd:
het te vormen adres r van de geheugenoperand voldoet niet aan:
 $0 \leq r < 2 + 15$ bij niet=absolute opdrachten of aan
 $0 \leq r < 2 + 18$ bij absolute opdrachten.

- stop 7: 7Y7C0 : (gedeeltelijke wijziging R 917, blz. 9, pt 17), een niet te skippen opdracht wordt ter uitvoering aangeboden met functiegedeelte 4 of 5 Y of Z of van het type 6 of 7 Y of Z nC0 met $n \geq 32$ (uitgezonderd INT en de ingevoerde pseudo-opdrachten (zie onder 2.6 en R 917, blz. 9)).
- stop 8: 7Y8C0 : gewenste opdracht(en) uitgevoerd: stop bij BVA, SGA, enz. en bij uitvoering opdracht in consolewoord (zie onder 1.1 en R 917, blz. 2,3).

De registers A, S en B van de X1 bevatten nadat gestopt is op een der bovenvermelde stoppen de volgende informatie:

	A	S	B
stop 3:	B(X8)	uit te voeren subr.sprong	OT(+1)
stop 4:	gevormde "OT"	uit te voeren sprong opdr.	oude OT(+1)
stop 5:	L+p	uit te voeren opdracht	OT(+1)
stop 6:	r	idem	idem
stop 7:	ongedefinieerd	idem	idem
stop 8:	idem	uitgevoerde opdracht	nieuwe OT

Na stoppen op een der stoppen 3 t/m 7 heeft doorstarten met H, BVA tot effect, dat het ter uitvoering aangeboden wordt geskipt. Voor stop 8 zie 1.1.

Een en ander wordt geëffectueerd door het volgende programma:

programma 1.2:

	DA 10 Z F 4 DI	correctie R 917, blz. 27.
10	2T 19 E 0 A	=> voor evt. stoppen op stop 3
	DA 21 Z F 5 DI	correctie R 917, blz. 28.
21	3LA 32767 A Z	mog.bruikbare OT? (anders stoppen
22	2T 12 E 0 A	=> op stop 4)
	DA 31 Z R 0 DI	correctie R 917, blz. 18
31	2T 0 E 0 A	=> voor evt. stoppen op stop 5
	DA 19 Z R 1 DI	correctie R 917, blz. 19 (zie ook
19	U 3LA 12 N 0 Z	3.3)
20	2T 4 E 0 A	=> voor evt. stoppen op stop 6
6,8ZE0 =>	21 3LS 32767 A	
	22 Y 0S 0 X 0	
	23 3LS 13 N 0	
	24 Y 6S 10 X 1	gemod. opdr.
	25 Y 6A 11 X 1	r
	DA 16 Z L 0 DI	correctie R 917, blz. 20
16	2T 29 E 0 A	=> voor evt. stoppen op stop 7
	DA 14 E E 0 DI	correctie R 917, blz. 52
14	Y 2T 24 Z E 0 A	-> naar stop 7
	DA 0 Z E 0 DI	
31ZR0 =>	0 U 3LA 32767 A Z	mog.bruikbare L+p?
	1 N 2T 14 E 0 A	-> naar stop 5
	2 U 3LA 12 Y 1 Z	beschermd adres?
	3 2T 0 R 1 A	=>
20ZR1 =>	4 N 2T 9 E 0 A	-> naar stop 6
	5 U 2A 14 X 1 P	niet absoluut?
	6 N 2T 21 R 1 A	->
	7 U 3LA 32767 A Z	r < 2 + 15?
	8 Y 2T 21 R 1 A	->
4 ->	9 2B 62 U 0	OT
	10 7Y 6 C 0	
	11 2T 23 F 1 A	=> skip
22ZF5 =>	12 Y 2T 18 F 1 A	->
	13 2S 0 A Z	maak cond. Y
1 ->	14 2B 62 U 0	OT
	15 2S 10 X 1	opdracht
	16 Y 7Y 4 C 0	onbruikbare OT
	17 N 7Y 5 C 0	onbruikbare L+p
	18 2T 23 F 1 A	=> skip
10ZF4 =>	19 2A 61 U 0 Z	B
	20 Y 2A 0 A	
	21 U 3LA 32767 A Z	mog.bruikbare B?
	22 Y 4P A B	
	23 Y 2T 11 F 4 A	->

(vervolg progr. 1.2):

		DA	24	Z E O DI	
14EE0,0ZE1	→24	2B	62	U O	OT
	25	2S	10	X 1	opdracht
	26 N	7Y	3	C O	onbruikbare B
	27 Y	7Y	7	C O	onbruikbare opdracht
	28	2T	23	F 1 A	⇒ skip
16ZLO	⇒	29 U	2LS 10	N O Z	4,5Y,Z-opdracht?
	30 N	6T	3	R 1 O	=)
	31 N	2T	17	L O A	→
	0	2T	24	E O A	⇒ naar stop 7
		DC			
		DO			

2. Aanpassing simulator aan recente gegevens over de X8

2.1. In tegenstelling tot het vermeldde in R 917, blz. 6 pt 7, dient OF(Overflow-indicatie) bij floating-opdrachten alleen gezet te worden als het een in-opdracht betreft, waarvan de tekens van de mantisse van de geheugenoperand verschillen.

Er wordt niet gereageerd op een eventuele tekeninconsistentie van de mantisse van F vóór de uitvoering van de floating-optelling,-vermenigvuldiging en -deling, of bij transport uit-opdrachten. Ter correctie:

Programma 2.1:

		DA	26	Z H 2 DI	correctie R 917, blz. 33
	26	2T	6	E O A	⇒
		DA	9	Z H 0 DI	
	9	2T	0	E O A	⇒
		DA	0	Z E 0 DI	
9ZH0	⇒	0	6T 0	T O 1	=) haal (n) gepakt in (A,S)
	1	OP	12	AA P	niet
	2 U	5P		SS E	tekenconsistent?
	3 Y	6T	7	EO 0	=)
	4	1P	12	AA	
	5	2T	10	HO A	⇒
26ZH2	⇒	6	4P	BB P	
3 =)	7	6S	4	X 1	
	8 N	2T	8	X 0 E	→
	9	2T	27	H 2 A	⇒
		DC			
		DO			

2.2. NINT wordt in tegenstelling tot het gedrag van simulatieprogramma van R 917 (zie blz. 6, pt. 8) niet gezet bij geheugentransport uit F als de exponent van $F \neq 0$ is (wel bij G uit-opdrachten als de kop van $F \neq 0$ is). Ter correctie:

programma 2.2.:

```

          DA 28 Z H 1 DI      correctie R 917, blz. 32
28 N 2A 57   U 0   Z      kop = 0?
29   2T 18   H 0 A      ==>
          DC
          DO

```

2.3. Indien men de floating-deling van het NINT-effect wenst uit te sluiten dient als volgt gecorrigeerd te worden:

programma 2.3:

```

          DA 17 Z H 0 DI      correctie R 917, blz. 31
17   2T 0   E 0 A      ==>

          DA 0 Z E 0 DI
17ZHO ==> 0 2S 10 X 1      opdracht
          1 0P 6   SS
          2 2LS 63   A      functiecijfers
          3 1S 57   A Z      floating-deling?
          4 N 3P 15 AA Z      exponent = 0?
          5   2T 18 HO A      ==>
          DC
          DO

```

2.4. Aan de set van X8-opdrachten is toegevoegd de U-variant van de normeeropdrachten van A,S of AS. Deze U-variant heeft tot effect, dat het normeren zelf achterwege blijft, doch dat wel in het B-register het aantal verschuivingen wordt genoteerd, dat nodig is voor de normering.

Ter correctie:

programma 2.4:

```

          DA 15 Z L 1 DI      correctie R 917, blz. 21
15 N 2T 0   E 0 A      ->
          DA 2 Z L 2 DI      correctie R 917, blz. 22
          2 2T 4   E 0 A      ==>
6ZE0 ==> 3 Y 6S 60   U 0      berg S

```

(vervolg progr. 2.4):

```

          DA 0 Z E O DI
15ZL1 => 0 2A 12 X 1 Z non U?
          1 N OLS 0 N 0 maak opdracht non U
          2 Y 2A 1 A Z maak cond. N
          3 2T 21 L 1 A =>
2ZL2 => 4 U 2A 12 X 1 Z non U?
          5 Y 6A 59 U 0 berg A
          6 2T 3 L 2 A =>
          DC
          DO

```

2.5. Ook bij de uitvoering van de execute-opdracht, wordt nadat deze in het opdrachtenregister is gebracht, de opdrachtteller met 1 opgehoogd. Dit betekent een correctie in het simulatieprogramma, die reeds is verwerkt in programma 1.1. (Eventueel wijzige men nog de opdracht op adres 21ZF4(R 917, blz. 27) in Y 2 T 10 F 0 A).

2.6. De code voor INT is thans vastgesteld, en komt in X1-code overeen met 6 Y 10240 C 0. Hierdoor komt de door ons voor INT ingevoerde pseudo-opdracht 6 Y 522 C 0 (zie R 917, blz. 6) te vervallen.

Verder is besloten 3 pseudo-opdrachten in te voeren voor lezen en ponsen, t.w. in X1-code:

```

6 Y 0 C P  functie: lees symbool van bandlezer en plaats in A
6 Y 1 C P  functie: " " " " " " " " S
6 Y 2 C P  functie: pons één symbool uit A (7-gaats, langzame ponser).

```

De simulator heeft de beperking, dat deze pseudo-opdrachten altijd worden uitgevoerd in de niet-conditiezettende versie en non-undisturbed. Ze kunnen wel altijd geconditioneerd worden uitgevoerd.

Een en ander wordt geëffectueerd door het volgende programma:

programma 2.6:

```

          DA 10 E E O DI      correctie R 917, blz. 52
10 2T 0 Z E O A =>

```

(vervolg progr. 2.6):

```

          DA 0  Z E O DI
10EE0 => 0 U 1S 512      A Z      lees in A?
          1 Y 2T 11      E O A      →
          2 U 1S 513      A Z      lees in S?
          3 Y 2T 10      E O A      →
          4 U 1S 514      A Z      pons uit A?
          5 Y 2A 59      U O
          6 Y 6Y 1      * H      pons langzaam, 7-gaats
          7 Y 2T 15      E E 1 A      →
          8 U 1S 10240    A Z      INT?
          9  2T 11      E O A      ==>
3 => 10  2A 1          A Z      maak conditie N
1 → 11 Y 2Y 1          X P      lees in A
      12 N 2Z 1          X P      lees in S
      13 Y 6A 59      Z U O      )berg
      14 N 6S 60      U O
      15  2T 15      E E 1 A      ==>
          DC
          DO

```

3. Opheffing van enkele beperkingen

3.1. Het simulatieprogramma beperkt zich bij het halen en bergen bij F-opdrachten, tot die operanden ($m, m+1$), waarbij $m \neq n = (12 \text{ Z Y } 1) =$ eindadres van het beschermde traject $0 \text{ X } 0 - n \text{ X } 0$ (zie R 917, blz. 1). Ditzelfde geldt voor het typen van een floating getal van genoemd dubbeladres (zie 3.2 en R 917, blz. 4,9). Deze beperking wordt opgeheven door het volgende programma:

programma 3.1:

```

          DA 19  Z T O DI      correctie R 917, blz. 29
19  2T 0      E O A      ==>
          DA 27  Z H 1 DI      correctie R 917, blz. 32
27 Y 2T 6      E O A      →
          DA 1    E K O DI      correctie R 917, blz. 55
1  2A      O X O B Z      kop = 0?
2  2T 3      Z E O A      ==>
5ZE0 ==> 3 Y 2T 2      E R O A      → typ integer*
          4 U 0LA 7      K 2  Z      misschien + inf?
          5 N 2T 8      K O A      →
          6 U 0LS 0      D15  Z      + inf?
          7 Y 2T 23     K 1  A      →
5  → 8  6A 0      X 1      red kop

```

(vervolg progr. 3.1):

		DA	13	E K 0 DI	correctie R 917, blz. 55
13		2A	0	X 1	herstel kop
		DA	25	E K 1 DI	correctie R 917, blz. 56
25		4P		SS P	+?
		DA	0	Z E 0 DI	
19ZT0	=> 0	6T	9	E 0 0	=) onderzoek adres
	1	2S	1	X 0 B	haal staart geh. operand
	2	2T	9	X 0 E	=>
2EKO	=> 3	6T	9	E 0 0	=) onderzoek adres
	4	2S	1	X 0 B	haal staart geh. operand
	5	2T	3	E K 0 A	=>
27ZH1	=> 6	6T	9	Z E 0 0	=) onderzoek adres
	7	6S	1	X 0 B	berg + F-staart
	8	2T	20	H 0 A	=>
0,3,6	=)	9	1B	0 U 0 A	-v
		10 U	1B	12 Y 1 Z	adres = n?
		11 N	OB	0 U 0 A	+v
		12	2T	8 X 0 E	=>
		DC			
		DO			

3.2. Behoort tot een traject $[a, b]$, $a < b$, van adressen, waarvan men de inhoud en het adresdeel van de inhoud wenst uit te typen (zie R 917, blz. 4,9) zowel n als $n+1$ ($\neq b$), dan splitst men dit traject bij n of $n+1$ in twee deeltrajecten en behandel deze deeltrajecten afzonderlijk. Het traject $[a, b]$ wordt dan gesplitst in de deeltrajecten $[a, n]$ en $[n+1, b]$, behalve als n en $n+1$ het adrespaar vormen van een uit te typen floating-getal. In dit laatste geval wordt het traject $[a, b]$ gesplitst in de trajecten $[a, n+1]$ en $[n+2, b]$ en het simulatieprogramma gecorrigeerd volgens progr. 3.1.

3.3. Hoewel bij indirect geadresseerde opdrachten adressering mogelijk is tot $2 + 18$, is, in verband met het kleinere afbeeldingsgeheugen van de X1, de simulator ingesteld op een absoluut adres $< 2 + 15$. Dit ongeacht of de opdracht wel of niet een absolute opdracht is. Wenst men bij absolute opdrachten, die indirect zijn geadresseerd, het getallenbereik uit te breiden tot $2 + 18$, dan wijzige men het simulatieprogram-

ma volgens programma 3.3 (zie onder).

Opgemerkt wordt, dat bij gebruikmaking van de testfaciliteiten b) of c) van R 917, blz. 2,3, de laatst uitgevoerde opdracht, als deze een absoluut opdracht is, die indirect is geadresseerd, bij invoering van deze correctie door het S-register van de X1 wordt aangegeven met ongewijzigd adresgedeelte, ook al is het berekende adres $< 2 + 15$. Hetzelfde is het geval bij toepassing van autostart G9 (zie 1.1).

programma 3.3:

	DA	20	Z	R	1	DI		correctie R 917, blz. 19 (zie ook progr. 1.2 waar deze correctie in is verwerkt)
20	U	2A	14	X	1	P		geen A-variant?
21	Y	OS	0	X	0			
22		3LS	13	N	0			maak opdracht ongecorrigeerd
23	Y	6S	10	X	1			berg gemodificeerd (in het niet-)
24	Y	6A	11	X	1			berg gevormde adres (absolute geval)
25		2T	8	X	0	E	=>	
		DC						
		DO						

3.4. Het simulatieprogramma van R 917 voert onjuist uit de niet-absolute additieve sprongen, die voor de vorming van OT het speciale adres $T = 62 X 0$ ook als "haaladres" gebruiken. Correctie:

programma 3.4:

	DA	12	Z	F	5	DI		correctie R 917, blz. 28
12	N	2T	0	E	0	A	→	
12ZF5 =>	0	U	1B	62	U	0	A	Z
	1	N	2A	0	X	1		adres = T?
	2		2T	13	F	5	A	=>
		DC						
		DO						

3.5. Zoals vermeld in R 917, blz. 10 pt 21, wordt door de simulator de floating-vermenigvuldiging en deling uitgevoerd volgens de methoden, toegepast bij de routines P 16 en P 51 van het P 9-complex.

Het is zeker niet uitgesloten, dat de gevolgde wijze van uitvoering in bepaalde gevallen een resultaat oplevert, dat verschilt van het door de X8 te leveren resultaat. Een eventuele herziening van de simulator op dit punt zullen we achterwege laten, wegens het ontbreken van gedocumenteerde gegevens over de X8 floating-vermenigvuldiging en -deling. Wel wijzen we er op, dat met betrekking tot de floating-deling het huidige deelprogramma (evenals P51) niet altijd het quotient goed afrondt. Deelt men bijv. $31 * 2 \uparrow -2047$ door 2, dan wordt $15 * 2 \uparrow -2047$ afgeleverd en niet $2 \uparrow -2043$. Correcte afronding vereist in bepaalde critieke gevallen (waarvan boven een voorbeeld) een zorgvuldiger bewerking. Hierin voorziet het volgende correctieprogramma:

programma 3.5:

	DA	27	Z W 1 DI	correctie R 917, blz. 46
	27 U OB	2048	A P	$B \geq -2047?$
	28 Y 2T	1	W 2 A	$\rightarrow$
	29 OB	2047	A	$-2087 \leq B \leq -2048$
	30 5P	BB		
	31 6T	18	S 2 0	=) schuif uit tot exp = -2047
	0 3B	2047	A	
28W1 $\rightarrow$	1 4P	SS	P	afronding
	2 Y OS	4096	A	
	3 1S	2048	A E	geen overloop naar de kop?
	4 Y 2T	12	W 2 A	$\rightarrow$
	5 OLS	0	D15 P	maak tekenbit in orde
	6 Y OA	2	A	verwerk overloop in kop
	7 1A	1	A E	
	8 N 1P	1	AA	
	9 N OB	1	A	
	10 U 1B	2048	A Z	
	11 Y 2T	1	K 5 A	$\rightarrow \frac{+}{-} \infty$
	DA	16	Z W 3 DI	correctie R 917, blz. 48
	16 2T	0	E 0 A	$\Rightarrow$
6ZEO $\Rightarrow$	17 Y 2T	23	W 3 A	$\rightarrow$
	18 OLS	0	D15 P	
	19 Y 1A	1	A	] corrigeer met $2 \uparrow 26$.
	20 Y OS	1	A	
	21 N OA	1	A	
	22 N 1S	1	A	

(vervolg progr. 3.5):

```

          DA 0   Z E O DI
16ZW3 => 0   0A 22   X 1   E
          1   2D 23   X 1
          2   2A 21   X 1
          3 Y 5P      SS
          4 N 4P      SS   Z
          5 Y 5P      SS
          6   2T 17   W 3 A   =>
          DC
          DO

```

3.6. Het simulatieprogramma van R 917 onderstelt, dat van een uit te typen floating getal, de mantisse tekenconsistent is (zie R 917, blz. 4, voetnoot).

Wenst men ook intekenconsistente floating getallen ter typing aan te bieden (waarbij het teken van de kop van de mantisse bij evaluatie van het getal prevaleert), dan corrigeren men het programma als volgt:

programma 3.6:

```

          DA 3   E K O DI      correctie R 917, blz. 55, na ver-
3       2T 0   Z E O A   => werking van correctie 3.1.

          DA 0   Z E O DI
3EKO  => 0   3LS 0   D15
          1 U 2LA 16384   A Z   +?
          2 N OLS 0   D15
          3   4P   AA   Z
          4 Y 2T 2   E R O A   -> typ integer
          5   2T 4   K O A   =>
          DC
          DO

```

4. Localisatie aanvullende programma's

De programmaonderdelen van vorengenoemde correctieprogramma's, die het simulatieprogramma van R 917 niet overschrijven, zijn genoteerd in termen van ZE.

Bij de X1-6 van het MC bevinden deze aanvullende delen zich aaneengesloten in het geheugentraject 10.20.0-10.23.18 (op het adres 10.24.0 begint het simulatieprogramma van R 917 (zie R 917, blz. 12)).

De indeling is als volgt:

beginadres:	aantal plaatsen:	zie programma:
10.20.0	13	3.1
10.20.13	14	1.1
10.20.27	7	3.5
10.21.2	10	2.1
10.21.12	6	2.3
10.21.18	7	2.4
10.21.25	16	2.6
10.22.9	3	3.4
10.22.12	6	3.6
10.22.18	33	1.2

Door deze toevoeging is het simulatieprogramma uitgebreid tot ongeveer 44 pagina's. Het beslaat bij de X1-6 van het MC nagenoeg geheel het heugentraject 10.20.0.-11.31.31.

De overige correcties hebben tot gevolg een overschrijving van een gedeelte van het simulatieprogramma van R 917. Het resulteert in een inhoudswijziging van de volgende adressen:

adres	Rapport R 917, blz:	zie programma:
15 ZZ0	13	1.1
31 ZR0	18	1.2
19-25 ZR1	19	1.2(waarin 3.3)
16 ZL0	20	1.2
15 ZL1	21	2.4
2- 3 ZL2	22	2.4
16 ZF1	24	1.1
10,25-29 ZF4	27	1.2, 1.1
12,21-22 ZF5	28	3.4, 1.2
19 ZT0	29	3.1
17 ZH0	31	2.3
27-29 ZH1	32	3.1, 2.2
26 ZH2	33	2.1
27-31 ZW1	46	3.5
0-11 ZW2	47	3.5
16-22 ZW3	48	3.5
0-1,6-31 ZY0	50	1.1
0-8 ZY1	51	1.1
10,14 EE0	52	2.6, 1.2
4 EE2	54	1.1
1-8, 13 EK0	55	3.6, 3.1
25 EK1	56	3.1

5. Slotopmerkingen

5.1. De simulator verwerkt de E-versie van de normale en additieve sprongen (0-2T) op dezelfde wijze als de P-versie (dus herstellend), en de Z-versie op dezelfde wijze als het normale geval (dus niet-herstellend). Dit met de X1 overeenstemmende gedrag, geldt ook voor de 3T-opdrachten.

5.2. Een 3T-opdracht verwerkt de simulator als een normale 2T-sprong-opdracht met A-variant. Het adresdeel van de opdracht wordt opgevat als te zijn indirect geadresseerd, ongeacht de waarden van de modificatie bits d19 en d20 van de opdracht.

5.3. De tijdsduur van 4-8 msec, zoals in R 917, blz. 2 vermeld voor het uitvoeren van een X8-opdracht door de simulator, geldt o.m. niet voor de floating-optelling. De tijd, die voor de uitvoering van de floating-optelling nodig is, bedraagt meer en wel 12-22 msec. De tijdsduur van de floating-vermenigvuldiging en -deling bedraagt bij de toegepaste methode (R 917, blz. 10, pt 21): 12-18 msec.

Een skip-opdracht vergt minder dan is opgegeven en wel ongeveer 2 msec. De meeste niet-floating opdrachten worden binnen 6 msec uitgevoerd.

Correcties in de beschrijving van R 917:

- blz. 3 7^e regel v.b. : A-vervalt
- blz. 9 14^e regel v.o. : i.p.v. getal, leze men: getal resp.
- blz. 11 achter no. 23 : i.p.v. (bij, leze men: (hoofdzakelijk bij
- blz. 23 bij opdracht 20 wordt niet ingesprongen
- blz. 39 achter de opdrachten op plaats 18 en 28: i.p.v. $ea < 0$ leze men $ea \leq 0$
- blz. 55 achter de opdracht op plaats 14: i.p.v. tekenconsistent?
leze men: mantisse en exponent zelfde teken?