

MINI- ASSEMBLER VOOR DE 6502 M. DOHMEN

Ergert u zich telkens weer aan het uittellen van sprongen bij sprong-instructies? Wij niet meer, want de hier beschreven „Mini-assembler” vergemakkelijkt in zeer veel opzichten het programmeren van de KIM. Het programma zelf vergt slechts 254 bytes RAM/ROM en zal voor veel 6502-KIM-gebruikers dan ook een welkome hulp zijn bij het schrijven van andere programma's.

AD 10	DA 20	PC 30	+ 40
C	D	E	F
8	9	A	B
4	5	6	7
0	1	2	3

Afb. 1
Indeling van het toetsenbord voor de Mini-assembler.

R. KOEKOEK

Werking

Wat doet deze Mini-assembler? Zonder verlies aan programmeerruimte kunnen op eenvoudige wijze labels worden geplaatst. Eenmaal geplaatst kunnen zij later gemakkelijk worden opgezocht. Het adres en de bijbehorende label verschijnen dan op het display. Er kunnen labels worden gebruikt bij alle spronginstructies en alle 3-bytes instructies (dus ook bij bijvoorbeeld LDA ABS).

Dat gaat als volgt in zijn werk. Als u het programma heeft ingetypt, zet u de NMI-vector op 0200 (beginadres van de Mini-assembler). Om vervolgens een label te plaatsen, moet u ervoor zorgen, dat het adres, waar de label moet komen, op het display staat. De naam van een label is een cijfercombinatie. Het toetsenbord van

in ons geval de KIM wordt daarom in het programma anders gebruikt, zoals is aangegeven in afb. 1. Er kunnen daardoor in totaal 80 dec labels worden gebruikt, voorgesteld door de getallen 00 tot en met 4F hex. De manier, waarop de labels in het geheugen worden opgeslagen, is te zien in afb. 2.

Om nu de labels 00 tot en met 0F hex te programmeren, dient u de „ST”-

toets in te drukken en vervolgens een toets tussen 0 en F hex, die hoort bij de label, die u wilt plaatsen (label 02 wordt: „ST”, „2”). Als u die toets los laat, zit u weer in het monitorprogramma en kunt u doorgaan met het intypen. De labels zijn verdeeld in groepen:

00 tot en met 0F is labelgroep 0, 10 tot en met 1F is labelgroep 1 enz.

Het laatst genoemde voorbeeld betreft dus labelgroep 0. Om andere labelgroepen te programmeren dient u als volgt te werk te gaan.

Labelgroep 1: „ST”, „AD” (kort), „0” (tot en met F).

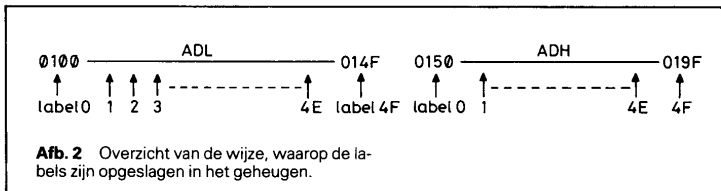
Labelgroep 2: „ST”, „DA” (kort), „0” (tot en met F).

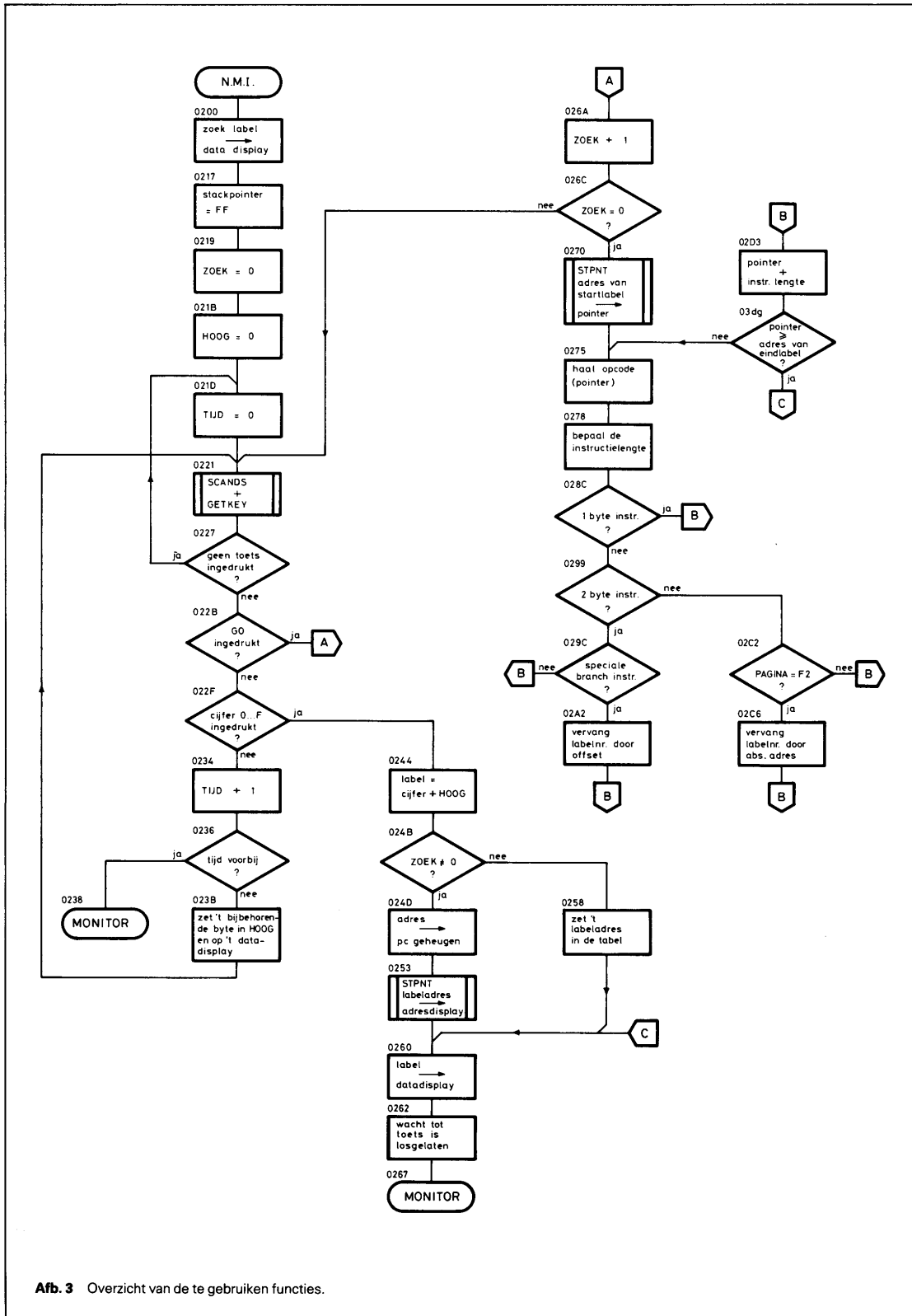
Labelgroep 3: „ST”, „PC” (kort), „0” (tot en met F).

Labelgroep 4: „ST”, „+” (kort), „0” (tot en met F).

De toetsen „AD”, „DA”, „PC” en „+” geven dus de labelgroep aan. Deze toetsen hebben nog een functie, die later zal worden besproken. Als u deze toetsen wilt gebruiken om een labelgroep aan te geven, dienen deze toetsen **kort** te worden ingedrukt. Evenals bij labelgroep 0 komt u na het indrukken van een cijfer toets terug in de monitor en kunt u doorgaan met het intypen van een te ontwikkelen programma. Als u de toets nog ingedrukt houdt verschijnt het labelnummer op het datadisplay ter controle. Een tweede functie: „zoek label”.

Wanneer u vermoedt dat u een label al heeft gebruikt, of wanneer u niet meer weet waar een bepaald label staat, dan kunt u deze label terug vinden door „ST” in te drukken, vervolgens kort „GO” en dan de naam van de label. Na deze handelingen te hebben verricht zit u weer in het monitorprogramma en staat op het display het adres, dat bij de zojuist opgezochte label hoort. Als u nu wilt verder gaan met intypen vanaf het adres, waar u was gebleven, hoeft u enkel simpelweg de „PC”-toets in te drukken. Wanneer u wel het adres weet waar u een label heeft neergezet, maar u weet niet welke label het is, dan toetst u terwijl u nog in de monitor zit het betreffende adres in en drukt u daarna „ST” in. Op het adresdisplay blijft het adres staan dat u heeft ingetypt. Op het datadisplay komt, als er geen label op dat adres was geplaatst, FF en als er op dat adres wel een label





Afb. 3 Overzicht van de te gebruiken functies.

was geplaatst, de naam van de label te staan. Als u daarna weer wilt doorgaan met het intypen van een programma moet u, om in Data-mode te komen, „DA” net zo lang ingedrukt houden, totdat het labelnummer, dat op het datadisplay staat, wordt vervangen door de werkelijke data. Om in de Adres-mode te komen, dient u al het voorgaande te doen, maar dan in plaats van „DA” de „AD” toets te gebruiken. Hetzelfde geldt voor „PC” en „+”. Hierna begint alles weer van voor af aan. Een overzicht van alle functies vindt u in afb. 3.

Uitvoering van het programma

Als u het te ontwikkelen programma inclusief de labels heeft ingetypt, kunt u deze per labelgroep vervangen. Wanneer u bijvoorbeeld labelgroep 0 heeft gebruikt, zet u label 00 op het beginadres en label 0F op het eindadres („AD”, beginadres, „ST”, 0, „AD”, eindadres, „ST”, F). Nu toetst u na „AD” weer het beginadres in en vervolgens „ST”. Hierna houdt u de „GO”-toets net zolang ingedrukt, totdat het eindadres op het adresdisplay verschijnt en het beginlabel op het datadisplay. Wanneer dat is gebeurd, moeten alle sprongopdrachten en alle 3-bytes instructies naar de juiste adressen wijzen.

Aangeven van labels in een programma

Alle relatieve spronginstructies eindigen op 10000 bin. Wanneer u geen label wilt gebruiken, plaatst u op normale wijze een relatieve sprong, bijvoorbeeld:

20 1F 1F

E6 00

D0 F9

F9 is nu geen label, maar een werkelijke offset. Wilt u refereren naar labels, dan verandert u X0 in X2. Het tweede byte van de (nu niet meer bestaande) instructie geeft het labelnummer aan. In het voorbeeld wordt dit

„AD”, 0300,

„ST”, 1 (label 01)

20 1F 1F

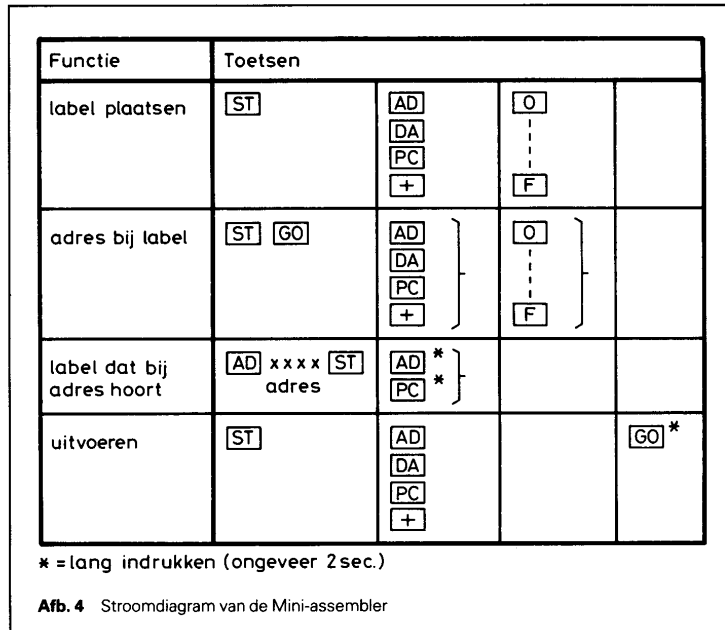
E6 00

D2 01 (label 01)

Ook bij 3-bytes instructies kan met labels worden gewerkt. Het eerste byte blijft hierbij onveranderd. Het tweede byte wordt vervangen door het labelnummer en het derde byte wordt altijd F2, mits pagina F2 niet wordt gebruikt voor RAM/ROM. In ons voorbeeld stond een 3-bytes instructie. Deze gaan we vervangen door een 3-bytes instructie met label. Adres 1F1F hex moet het label 02 krijgen, de rest blijft hetzelfde.

„AD”, 1F1F, „ST”, 2,

„AD”, 0300, „ST”, 1,



20 02 F2

E6 00

D2 01

Om het besprokene te illustreren wordt het programma „Checksum” uit lijst 1 stap voor stap ingetypt met gebruikmaking van labels. Daarbij wordt labelgroep 1 toegepast.

„AD”, 0300, „ST”, „AD”, 0:

label 10 op beginadres

„DA”, A2 + F2 +

86 + F6 +

86 + F7 +

„ST”, „AD”, 1 :

label 11 op adres 0306

BD + 12 + F2 + :

label 12 op adres 01FF

20 + 13 + F2 + :

label 13 op adres 1F91

CA +

D2 + 11 +

4C + 14 + F2 + :

label 14 op adres 1C22

Nu moet op adres 01FF label 12 komen te staan, op 1F91 label 13 en op adres 1C22 label 14.

„AD”, 01FF, „ST”, „AD”, 2

„AD”, 1F91, „ST”, „AD”, 3

„AD”, 1C22, „ST”, „AD”, 4

Nu zijn alle labels geplaatst en moeten alleen nog het eind- en beginadres worden aangegeven.

„AD”, 0312, „ST”, „AD”, F

„AD”, 0300, „ST”, „AD”, 0

Nu kunnen de labels worden vervangen. We toetsen „ST” in en dan „AD”, omdat labelgroep 1 moet worden omgezet. Vervolgens houden we de „GO”-toets net zolang ingedrukt, totdat het eindadres verschijnt. Als alles goed is gegaan, zijn alle labels vervangen en werkt het programma. Na

uitvoeren van dit programma moet op adres 00F6 81 hex staan en op adres 00F7 F7 hex.

Het programma „Checksum” is een soort optelprogramma dat alle bytes van het programma „Mini-assembler” bij elkaar optelt en de som op de adressen 00F6 en 00F7 hex zet. Het is dus een controle, of de „Mini-assembler” goed is ingetypt. Dit programma maakt gebruik van een routine uit de KIM-monitor, waardoor het niet zonder meer op andere systemen werkt, maar dat geldt ook voor de „Mini-assembler” zelf.

Meerdere labelgroepen

Als in een programma meer labels aanwezig zijn dan een labelgroep kan bevatten, moeten, om alle labelgroepen tegelijk te kunnen omzetten, alle startlabels (het startadres aangevend) en alle eindlabels (het eindadres aangevend) hetzelfde zijn. Wanneer u in een programma de labelgroepen apart houdt voor de diverse programmadelen (bijvoorbeeld bij subroutines), hoeft u slechts één labelgroep te vervangen, zodat de andere groepen onaangetast blijven.

Het is mogelijk meerdere labels op een adres te zetten. Bij het zoeken naar een bepaalde label staat altijd de hoogste label op het datadisplay. Een voordeel van dit programma is dat men niet altijd naar labels hoeft te refereren. De mensen, die het uittellen van sprongen tot hun hobby hebben gemaakt, kunnen deze dan ook blijven uitoefenen. Wanneer u een nieuw programma wilt ontwikkelen, kunt u

het programma uit lijst 2 gebruiken om de labels te wissen.

Relatieve sprongen

In plaats van X0 is X2 gebruikt bij relatieve sprongen. Dit is allereerst gedaan om onderscheid te maken met X0, maar ook om het in de soep draaien van een programma te voorkomen,

als u vergeten bent de labels om te zetten. Immers, de microprocessor kent de instructie XXX10010 niet en zal daardoor stoppen. Alleen een reset kan de microprocessor weer aan de gang brengen. De 3-bytes instructies eindigen op F2. Bij spronginstructies springt de microprocessor dus naar pagina F2. Indien daar geen geheugen

aanwezig is, vindt hij de instructie F2, welke ook aan het bitpatroon XXX10010 voldoet.

Programma

Wanneer de „ST”-toets wordt ingedrukt, wordt er een NMI gegenereerd. Hierdoor springt de microprocessor vanuit de monitor naar 0200 hex, waar

Lijst 1 Programma om de labels te wissen en het programmavoorbeeld „Checksum”.

MICRO-WARE ASSEMBLER 6500-1.0 PAGE 01			
0010:		1090:	0204 C9 19
0020:		1100:	0206 F0 0C
0030:	MINI-ASSEMBLER	1110:	0208 29 00
0040:		1120:	0209 C9 08
0050:	M. DOHREN	1130:	020C F0 45
0060:	R. KOEKOEK	1140:	020E 29 0C
0070:		1150:	0209 C9 0C
0080:	MICROPROCESSOR 6502	1160:	0202 D0 01
0090:		1170:	0204 00
0100:		1180:	0205 E8
0110:	0200	1190:	0206 98
0120:	ORG #0200	1200:	0207 00 01
0130:	ZERO-PAGE ADRESSEN	1210:	0209 CA
0140:	VAN DE KIM	1220:	0209 D0 20
0150:		1230:	020C 29 1F
0160:	0200	1240:	0206 49 12
0170:	PCL * #00EF	1250:	0208 D0 16
0180:	PCH * #00F0	1260:	0202 B1 FA
0190:	INL * #00F8	1270:	0204 00
0200:	INH * #00F9	1280:	0205 BC 00 01
0210:	POINTL * #00FF	1290:	0208 88
0220:	POINTH * #00FB	1300:	0209 98
0230:	HOOG * #00FD	1310:	0209 10
0240:	ZOEK * #00FE	1320:	020E E5 FA
0250:		1330:	020D 00 01
0260:	TIJDELIJKE ADRESSEN	1340:	020F 91 FA
0270:	TABR * #0100	1350:	0208 68
0280:	ENDAD * #010F	1360:	0202 B1 FA
0290:	TAB8 * #0150	1370:	0204 29 F0
0300:	TAB0 * #015F	1380:	0206 91 FA
0310:	TAB0 * #02E9	1390:	0208 A2 01
0320:		1400:	020A D0 17
0330:	SUBROUTINES UIT DE	1410:	0203 61 FA
0340:	KIM-MONITOR	1420:	020E 00
0350:		1430:	020F C8
0360:	TTVKB * #1C77	1440:	020C B1 FA
0370:	MONA * #1C83	1450:	020C C9 F2
0380:	INCPY * #1F63	1460:	0204 D0 06
0390:	GETXY * #1F6A	1470:	020C 80 50 01
0400:	SCANDS * #1F1F	1480:	020C 91 FA
0410:		1490:	020C 88
0420:		1500:	020C 80 00 01
0430:	HOOFDPROGRAMMA	1510:	020F 91 FA
0440:		1520:	0201 02 02
0450:		1530:	0203 20 63 1F
0460:	0200 A2 4F	1540:	0206 CA
0470:	0202 05 FA	1550:	0207 10 FA
0480:	0204 D0 00 01	1560:	0209 A6 F0
0490:	0207 D0 07	1570:	020B 05 FB
0500:	0209 05 FB	1580:	020D D0 5F 01
0510:	020E D0 50 01	1590:	0203 91 FA
0520:	020E F0 03	1600:	02E2 D0 07
0530:	0210 CA	1610:	02E4 A5 FA
0540:	0211 10 EF	1620:	02E0 D0 0F 01
0550:	0213 06 F9	1630:	02E9 90 88
0560:	0215 A2 FF	1640:	02EB 4C 60 02
0570:	0217 9A	1650:	02EE D0 00 01
0580:	0218 E8	1660:	02F1 85 FA
0590:	0219 86 FE	1670:	02F3 D0 50 01
0600:	021B 86 FD	1680:	02F6 85 FB
0610:	021D 09 00	1690:	02F8 68
0620:	021F 85 FE	1700:	02F9 10
0630:	0221 20 1F 1F	1710:	02FA 20
0640:	0224 20 6A 1F	1720:	02FB 40
0650:	0227 C9 15	1730:	02FC 50
0660:	0229 F0 F2	1740:	02FD 30
0670:	022B C9 13		
0680:	022D F8 38		
0690:	022F C9 10		
0700:	0231 90 11		
0710:	0233 08		
0720:	0234 E6 FE		
0730:	0236 00 03		
0740:	0238 4C 83 1C		
0750:	023B BE E9 02		
0760:	023E 86 F9		
0770:	0240 86 F0		
0780:	0242 D0 DD		
0790:	0244 00 FD		
0800:	0246 00		
0810:	0247 05 FA		
0820:	0249 04 FE		
0830:	024B F0 0E		
0840:	024D 85 EF		
0850:	024F 05 FB		
0860:	0251 05 FB		
0870:	0253 00 EE 02		
0880:	0256 90 08		
0890:	0258 90 00 01		
0900:	025B 05 FE		
0910:	025D 90 50 01		
0920:	0260 06 F9		
0930:	0262 20 1F 1F		
0940:	0265 D0 FE		
0950:	0267 4C 77 1C		
0960:	026A E6 FE		
0970:	026C D0 83		
0980:	026E 06 FD		
0990:	0270 20 EE 02		
1000:	0273 00 00		
1010:	0275 01 FA		
1020:	0277 06		
1030:	0278 02 00		
1040:	0279 C9 20		
1050:	027C F0 1E		
1060:	027E 29 9F		
1070:	0280 F0 51		
1080:	0282 29 1D		
1090:	0284 C9 19		
1100:	0286 F0 0C		
1110:	0288 29 00		
1120:	0289 C9 08		
1130:	028C F0 45		
1140:	028E 29 0C		
1150:	0289 C9 0C		
1160:	0282 D0 01		
1170:	0284 00		
1180:	0285 E8		
1190:	0286 98		
1200:	0287 00 01		
1210:	0289 CA		
1220:	0289 D0 20		
1230:	028C 29 1F		
1240:	0286 49 12		
1250:	0288 D0 16		
1260:	0282 B1 FA		
1270:	0284 00		
1280:	0285 BC 00 01		
1290:	0288 88		
1300:	0289 98		
1310:	0289 10		
1320:	028E E5 FA		
1330:	028D 00 01		
1340:	020F 91 FA		
1350:	0208 68		
1360:	0202 B1 FA		
1370:	0204 29 F0		
1380:	0206 91 FA		
1390:	0208 A2 01		
1400:	020A D0 17		
1410:	0203 61 FA		
1420:	020E 00		
1430:	020F C8		
1440:	020C B1 FA		
1450:	020C C9 F2		
1460:	0204 D0 06		
1470:	020C 80 50 01		
1480:	020C 91 FA		
1490:	020C 88		
1500:	020C 80 00 01		
1510:	020F 91 FA		
1520:	0201 02 02		
1530:	0203 20 63 1F		
1540:	0206 CA		
1550:	0207 10 FA		
1560:	0209 A6 F0		
1570:	020B 05 FB		
1580:	020D D0 5F 01		
1590:	0203 91 FA		
1600:	02E2 D0 07		
1610:	02E4 A5 FA		
1620:	02E0 D0 0F 01		
1630:	02E9 90 88		
1640:	02EB 4C 60 02		
1650:	02EE D0 00 01		
1660:	02F1 85 FA		
1670:	02F3 D0 50 01		
1680:	02F6 85 FB		
1690:	02F8 68		
1700:	02F9 10		
1710:	02FA 20		
1720:	02FB 40		
1730:	02FC 50		
1740:	02FD 30		

Lijst 2 Programma van de mini-assembler

1150:	0300	ORG	#0300
1200:			
1210:	0300	CHKHI *	#00F6
1220:	0300	CHKSUM *	#00F7
1230:	0300	CHECK *	#01FF
1240:	0300	MONIT *	#1C22
1250:	0300	CHK *	#1F91
1260:			
1270:		PROGRAMMA OM LABELS	TE WISSEN
1280:			
1290:	0300 A9 FF	WIS	LDXIM #FF
1300:	0302 A2 00	LOOPV	LDXIM #00
1310:	0304 40		
1320:	0305 CA	DEX	
1330:	0306 D0 FC	TABE	LOOPV
1340:	0308 4C 22 1C	JMP	MONIT
1350:			
1360:			
1370:			
1380:			
1390:	0300	ORG	#0300
1400:			
1410:		VOORBEELDPROGRAMMA OM DE	CHECKSUM TE BEREKENEN VAN
1420:		DE MINI-ASSEMBLER	
1430:			
1440:			
1450:	0300 A2 FE	CSUM	LDXIM #FE
1460:	0302 86 F6	STX	CHKHI
1470:	0304 86 F7	STX	CHKSUM
1480:	0306 D0 FF 01	LOOPV	LDXIM #CHECK
1490:	0309 20 91 1F	JSR	CHK
1500:	030C CA	DEX	
1510:	0300 D0 F7	BNE	LOOPV
1520:	030F 4C 22 1C	JMP	MONIT

de stackpointer wordt goed gezet en er naar de label wordt gezocht, die bij het op het adresdisplay zichtbare adres hoort (zie afb. 4). De tijdteller voor de „GO“-toets wordt op 0 gezet, evenals de geheugenplaats „Hoog“, die ervoor dient de labelgroep te onthouden. Zolang er geen toets wordt ingedrukt blijft „Tijd“ op nul. Wanneer „AD“, „DA“, „PC“ of „+“ wordt ingedrukt, begint „Tijd“ te lopen en wordt de bijbehorende labelgroep in „Hoog“ gezet. Als de toets lang wordt vastgehouden („Tijd“ is weer nul), springt het programma terug naar de monitor. De bijbehorende code wordt nu als monitorfunctie geïnterpreteerd en uitgevoerd. Als een cijfer wordt ingedrukt, wordt de waarde opgeteld bij „Hoog“, dat zo het uiteindelijke label vormt. Bevindt het programma zich in de zoekmode, dan wordt het adres, dat op het display staat, in het PC-geheugen (00EF, 00F0 hex) van de monitor gezet. Het adres, dat bij de zojuist ingetypte label hoort, wordt op het display gezet. Zodra de laatste toets is losgelaten, springt het programma terug naar de monitor. Indien

het programma zich niet¹ in de zoekmode bevindt, wordt het adres, dat bij de label hoort, in de tabel gezet. Wanneer de „GO“-toets wordt ingedrukt, wordt „Zoek“ opgehoogd en springt het programma, zolang „Zoek“ niet weer 00 is, terug naar de keyboard-routine. Als de „GO“-toets slechts kort wordt ingedrukt, zal „Zoek“ ongelijk zijn aan nul en is het programma in de zoekmode. Wordt de „GO“-toets lang ingedrukt dan zal op een gegeven moment (na ongeveer 1 s) „Zoek“ gelijk worden aan nul en begint het programma met het vervangen van de labels. Het adres van de startlabel wordt in de pointer gezet. Hierna wordt de opcode van de eerst-(volgend) instructie opgehaald. Volgens wordt de instructielengte bepaald. Eén-byte instructies slaat hij over. Bij 2-bytes instructies kijkt het naar het volgende bitpatroon: XXX10010. Indien de instructie hieraan voldoet, worden de laatste vier bitjes 0 gemaakt en het tweede byte van de instructie vervangen door de offset. Bij 3-bytes instructies kijkt hij eerst naar het laatste byte. Wanneer

dit gelijk is aan F2, vervangt hij de laatste twee bytes door het absolute adres. Daarna wordt de pointer op de volgende instructie gezet. Indien de pointer het eindadres overschrijdt, springt het programma weer terug naar de monitor, zodra de „GO“-toets wordt losgelaten. Tot slot een paar tips.

Nu u met labels kunt werken, zult u ook wel instructies willen kunnen weghalen of tussenvoegen. Dit gaat heel gemakkelijk als alle relatieve spronginstructies en alle 3-bytes instructies door labels zijn vervangen. U moet dan een „normaal“ schuifprogramma gebruiken en op het moment, dat er wordt geschoven, even kijken of het betreffende adres een label voorstelt (pagina 01 RAM). Als het een label bevat moet het adres van de label worden vervangen door het nieuwe adres.

Het is vaak handig om labels op cassette te zetten. Dit kan door het geheugenblok van adres 0100 tot en met 019F hex op te slaan. ■