



Mini-assembler

Klein, maar dapper, kort en krachtig

M. Dohmen

R. Koekoek

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

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

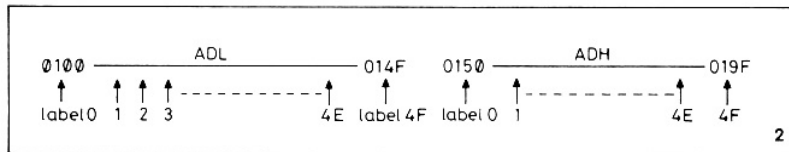
AD 10	DA 20	PC 30	+	40
C	D	E	F	
8	9	A	B	
4	5	6	7	
0	1	2	3	

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 3bytes 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 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.



Afb. 2 Overzicht van de wijze, waarop de labels zijn opgeslagen in het geheugen.

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 en 10 tot en met 1F is labelgroep 1 enz. Het laatst genoemde voorbeeld betrof 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 cijfertoets 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 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.

Functie	Toetsen			
label plaatsen	[ST]	[AD] [DA] [PC] [+]	[0] : : : [F]	
adres bij label	[ST] [GO]	[AD] [DA] [PC] [+]	[0] : : : [F]	
label dat bij adres hoort	[AD] xxxx [ST] adres	[AD] * [PC] *		
uitvoeren	[ST]	[AD] [DA] [PC] [+]		[GO] *

* = lang indrukken (ongeveer 2 sec)

3

Afb. 3 Overzicht van de te gebruiken functies

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 OF 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

DO F9

F9 is nu geen label, maar een werkelijke offset. Wilt u refereren naar labels, dan verandert u XO 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 3bytes 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 l0 op beginadres

„DA”, A2 + FE +

86 + F6 +

86 + F7 +

„ST” „AD” , 1:

label l1 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 l3 en op adres 1C22 label l4.

„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 programmaatje maakt gebruik van een routine uit de KIM-monitor, waardoor het niet

zonder meer op andere systemen werkt, maar dat geldt ook voor de „Miniassembler” 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 start-labels (het startadres aangevend) en alle eind-labels (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 1 (zie deel 1) 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 bit-patroon 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 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

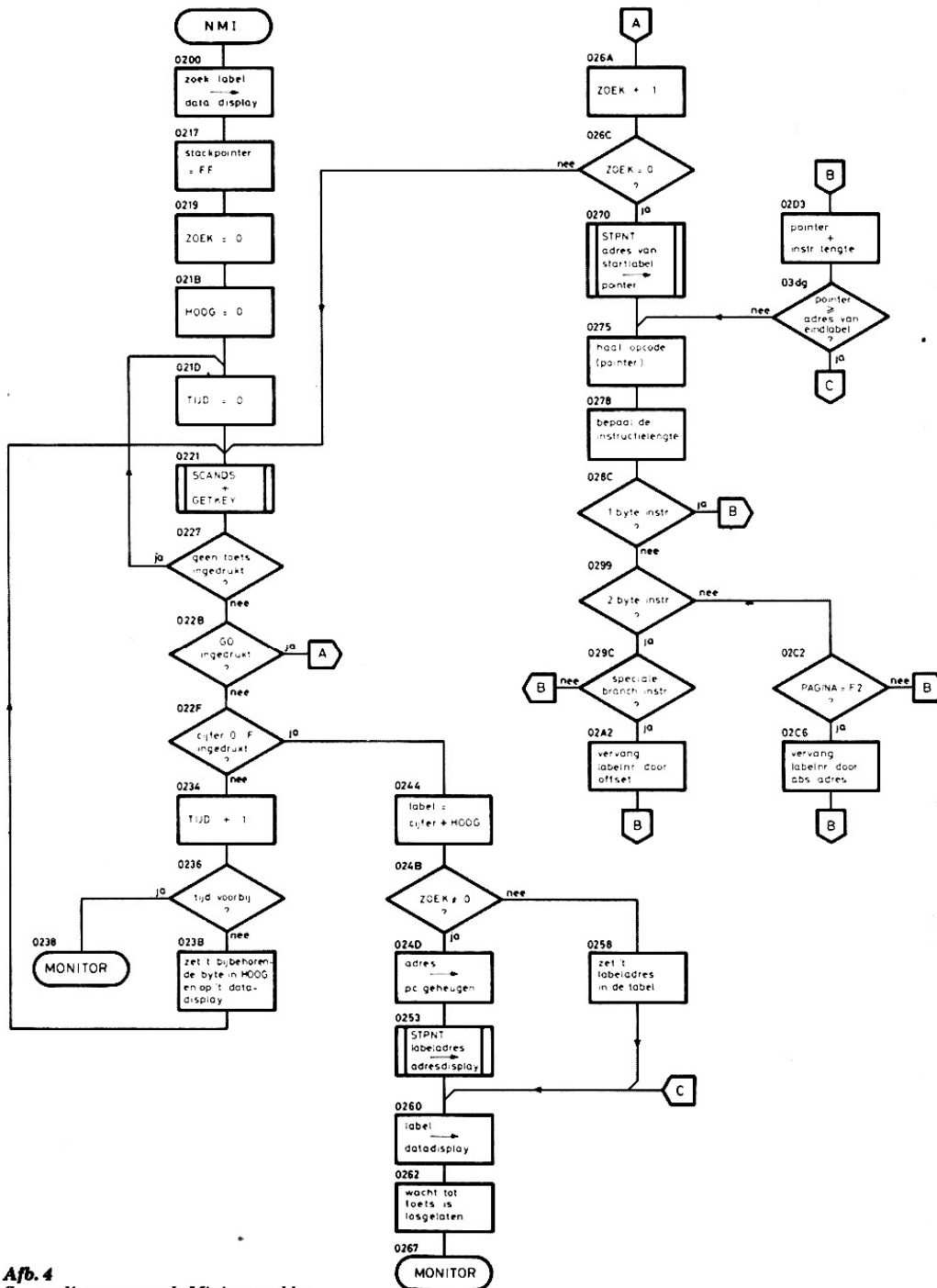
gezet. Het adres, dat bij de zojuist ingetypte label hoort, wordt op het display gezet. Zodra de laatste toets is losgelaten, springt het programma terug naar de monitor. Indien het programma zich niet in de zoekmode bevindt, wordt het adres, dat bij de label hoort, in de tabel gezet. Wanneer de „GO”-toets 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)e instructie opgehaald. Vervolgens wordt de instructielengte bepaald. Eén-byte instructies slaat hij over. Bij 2-bytes instructies kijkt het naar het volgende bit-patroon: 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 3bytes 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 in de KIM-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.



Mini-assembler



Afb. 4
Stroomdiagram van de Mini-assembler

Lijst 2 Programma van de Mini-assembler.

MICRO-WARE ASSEMBLER 65XX-1.0 PAGE 01

[illegible]