PSEUDO-ROM voor MSX

MSX CLUB MAGAZINE 25

Theo van Dooren

Scanned, ocr'ed and converted to PDF by HansO, 2001

Een artikel voor programmeurs die zich bezig houden met het schrijven, debuggen of aanpassen van programmatuur voor een EPROM.

Voordat men een machinetaal-programma zover ontwikkeld heeft, dat het in een EPROM zonder problemen kan draaien, is men nogal wat tijd kwijt aan het ontluizen (debuggen) van het programma.
Een machinetaal-programma wordt vaak geschreven in assembler of een andere programmeertaal (bv Pascal of C). Dit heet het bronprogramma, ofwel de source. Na het assembleren hiervan, wordt het betreffende programma dan in een EPROM "gebakken", waarna men kan overgaan tot het uittesten. Komen in deze fase nog fouten boven water, of wil men nog iets veranderen, dan wordt de source gewijzigd, en wordt het programma, na het assembleren, opnieuw in een EPROM gezet. Deze procedure moet vaak nog enige malen herhaald worden, voordat het programma helemaal naar wens funktioneert.

Deze werkwijze resulteert vaak in een aantal foutief geprogrammeerde EPROM's. Als men niet kan beschikken over een aantal van deze, toch vrij kostbare geheugenchips, is men genoodzaakt om de gebruikte EPROM's tussendoor telkens te wissen, teneinde ze opnieuwte kunnen gebruiken. Dit kost nogal wat tijd en ergernis. Om dit te voorkomen zocht ik naar een alternatief voor deze werkwijze. Dit heb ik gevonden in een zogenaamde pseudo-ROM. Een pseudo-ROM, is een RAM-geheugen dat zich kan gedragen als een (EP)ROM. Dit is bereikt door een schakelaar in de write-lijn te zetten. Als de schakelaar gesloten is, kan men in het geheugen lezen en schrijven. Is de schakelaar open dan kan men het geheugen alleen uitlezen, zoals een (EP)ROM.

Na het assembleren, staat de data (object-code) zoals die in de EPROM moet komen op een bepaalde plaats in het geheugen. Sommige assemblers zetten de object-code ook op de disk. Men kan dit eventueel ook zelf doen met het basic-commando BSAVE"NAAM",beginadres,eindadres. De data kan men dan weer in het geheugen plaatsen met het BLOAD-commando. Wil men de data op een andere plaats in het geheugen zetten, dan kan aan het BLOAD-commando nog een verschuiving worden toegevoegd. Heeft men bijvoorbeeld een data-blok met

BSAVE" DATA",&H5000,&H5800

op disk gezet, dan komt de data met

BLOAD"DATA", + &H4000

op adres 9000H tot 9800H in het geheugen. Dit is noodzakelijk als de data anders op een plaats in het geheugen komt waar al een ander (monitor) -programma staat.

Nu is het zaak de data in het geheugen, te verplaatsen naar de RAM-module, die zich in een cartridge-slot bevindt. Dat kan met een zelfgemaakte machinetaal-routine, het is echter eenvoudiger om een hiervoor geschikt monitor-programma te gebruiken. Hierbij maak ik graag gebruik van het monitor-programma MSXBUG, waarmee data eenvoudig verplaatst kan worden, en waarbij men zelfde slot-indeling van de verschillende pagina's kan bepalen. Deze monitor staat zelf op adres 4000H. Overigens heb ik zelf dit monitor-programma in een EPROM gezet, zodat dit nauwelijks nog enige geheugenruimte in beslag neemt, waardoor het beschikbare vrije geheugen zo groot mogelijk blijft. Dit monitor-programma is dan op elk gewenst moment, vanuit basic, met een CALL aan te roepen. Erg makkelijk dus. Nadat men de data in de RAM-module heeft geplaatst, kan met het testen worden begonnen. Hiertoe zet men de Write-Protect-schakelaar in de geopende stand, en wordt de computer gereset. Funktioneert het programma nog niet naar wens of zitten er nog fouten in, dan kan de data in de RAM-module gewijzigd worden door de WP-schakelaar weer in te schakelen. Eventueel wordt de module weer opnieuw geprogrammeerd.

De schakeling heb ik gebouwd op een MSX-experimenteer-print, Deze print heeft een 50-polige connector welke zo in een cartridge-slot van een MSX-computer past. Ideaal voor deze toepassing dus. Op de print komt volgens bijgaand schema een adresselektor in de vorm van een ie type 74LS138, dit-is een 3-naar-8 decoder.
De SLTSL-lijn aktiveert het ie. Dit ic selekteert uit de 3 hoogste adreslijnen A13, A14 en A15 een bepaald geheugen-gebied via de uitgangen Y0 t/m Y7, die de shipselect-lijn van de RAM-ic's aktiveren. Het geheugen-gebied wordt als het ware opgedeeld in blokken van 8 Kbyte. De begin-adressen van de blokken staan in het schema bij de uitgangen aangegeven. Op deze manier kan men een aantal RAM-ic's naar eigen behoefte toepassen.

Wil men het geheel eenvoudig houden, en heeft men voldoende aan 8 Kbyte, dan kan met 1 RAM-ic van 8 Kbyte op adres 4000H volstaan worden. Het is echter ook mogelijk om een geheugen van 64 Kb samen te stellen. Ook kan men andere RAM-ic's toepassen, bijvoorbeeld 1 van het type 32Kb op adres 4000H. In mijn geval heb ik 4 RAM-ic's van 8 Kb toegepast, welke aangesloten zijn op de uitgangen Y2 t/m Y5. In totaal dus 32 Kb op adres 4000H t/m BFFFH. Alle RAM-ic's worden parallel aan elkaar geschakeld, uitgezonderd pen 20, de chip-select. Men kan de RAM-ic's daartoe eventueel op elkaar stapelen, zodat de bedrading op de print eenvoudiger kan worden uitgevoerd.
Verder zien we links in het schema de aansluitingen van de RAM-ic's op de data- en adresbus via de MSX-cartridgecon-nector.

Rechts in het schema vinden we nog de diverse controllijnen en het voedingsgedeelte. De READ-lijn wordt doorverbonden met de OUTPUT-ENABLE-ingang en de WRITE-lijn via de eerder besproken schakelaar met de WRITE-ENABLE-in-gang. Als de computer ingeschakeld is, wordt de module via een diode vanuit de computer gevoed. Is de

computer uitgeschakeld, of de module los genomen, dan vindt de voeding plaats vauit de back-up batterij. De weerstand van 4,7 K en de condensator van 100 nF zorgen voor een "zachte" overgang van de ene naar de andere voeding.

De weerstand van 10 K zorgt ervoor dat de WE-ingang van de RAM "hoog" blijft, zodat de data in de RAM niet ongewild verminkt wordt in de write-protect-stand van de schakelaar, bij het overschakelen van de voeding. Indien als voedingsbron een NiCad-accu wordt gebruikt, wordt deze via weerstand RL bij een ingeschakelde computer opgeladen. De laadstroom en dus de waarde van deze weerstand hangt af van de capaciteit van de NiCad-accu. De laadstroom dient ongeveer 1/10 van de capaciteit in mA te zijn. In mijn geval heb ik een weerstand van 100 ohm toegepast, waardoor de gemeten laadstroom 7 a 8 mA bedraagt. Als men voor de voedingsbron gewone batterijen kiest, dient de weerstand RL uiteraard niet toegepast te worden.
Zoals u ziet kan deze flexibele schakeling op verschillende manieren ingevuld dan wel toegepast worden. Ik spreek uit ervaring, als ik zeg dat ik van deze uitbreiding al erg veel plezier heb gehad, en wens eenieder die de uitbreiding gaat gebruiken, hetzelfde toe.