Zero S.C.

HANDLEIDING

EPROM PROGRAMMER

Leeg tussenvel

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

INHOUDSOPGAVE:

Hoofdstuk 1 - Introductie.....pag.1

,, 2 - Principiele werking.....pag.2

,, 3 - De Hardware.....pag.3
3.1. Eprom lezen.....pag.4
3.2. Eprom programmeren.....pag.4
3.3. De schakeling.....pag.6
3.4. CE en OE (Chip- en Output-Enable)...pag.7
3.5. Rom-sockets.....pag.9
3.6. Selectie Eprom-type.....pag.10
3.7. Low Extraction Force IC-socket.....pag.10

,, 4 - De Software.....pag.12
4.1. Het aanpassen van de software.....pag.12
4.2. Moselijkheden van de software.....pag.14

,, 5 - Het starten van de programmer.....pag.18
5.1. Eprom erin.....pag.18
5.2. De koude start.....pag.18
5.3. Kopieren naar RAM.....pag.18
5.4. Sprong naar RESET-vector.....pag.19
5.5. Aanpassen software.....pag.19
5.6. Het programma wegzetten.....pag.19
5.7. Het inzetten van de programmer.....pag.19
5.8. De warme start van de software.....pag.20
5.9. Startklare versie.....pag.20
5.10 Alleen een hexadecimale dump.....pag.20

,, 6 - Losse opmerkingen en tips.....pag.21

,, 7 - Hexadecimale dump.....pag.22

Leeg tussenvel

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

Hoofdstuk 1

››› Introductie ‹‹‹

De Zero Eprom programmer is een zeer goedkope vervanging (maar niet mindere) voor de dure stand-alone Eprom programmers. Deze stand-alone Eprom programmers hebben over het algemeen een eigen microprocessor, geheugen en invoer- en uitvoerlijnen. De Zero Eprom programmer heeft dit juist niet, omdat deze gebruik maakt van de reeds aanwezige micoprocessor en geheugens in uw systeem. Verder zijn er ook geen invoer- en uitvoerlijnen nodig.
De programmer kan eenvoudigweg in een vrije of vrij te maken Rom-/Eprom-socket worden gestoken.
De Eprom-sockets waar de programmer zonder meer op kan worden aangesloten, kunnen van het type 2708/2758/2716/2516/2732/2532 zijn. Over de Rom-sockets kunt u meer lezen in Hoofdstuk 3.

De bijgeleverde software voor de programmer geeft u alle mogelijkheden die u nodig heeft bij het programmeren van Eproms. Er is van alles aan gedaan om de software gebruikersvriendelijk te maken, o.a. door het geven van foutmeldingen bij het ingeven van verkeerde instructies of onmogelijke getallen.
De software is dialoog gericht, d.w.z. dat er via het beeldscherm vragen en instructies verschijnen waarop de gebruiker via het toetsenbord kan reageren.

Hoofdstuk 2

))) Principiele werking (((

Het unieke van de Zero Eprom programmer is gelegen, naast de zeer uitgebreide software, in het feit dat er geen speciale invoer- en uitvoerlijnen (input en output) nodig zijn. U vraagt zich nu misschien af hoe er dan bij de Zero Eprom programmer communicatie tussen de programmer en de computer mogelijk is. Wel die communicatie is er als u de programmer in een Rom-/Eprom-socket steekt. Deze socket staat altijd in verbinding met de microprocessor in uw systeem, en dat is voor de programmer voldoende. Op dit moment merkt u waarschijnlijk op dat de communicatie tussen de microprocessor en een Rom-/Eprom-socket maar naar 1 richting mogelijk is. Deze richting houdt in, alleen data lezen door de microprocessor, want data schrijven naar een Rom-/Eprom-socket is niet mogelijk. Maar juist dit schrijven van data is noodzakelijk als men data naar een Eprom wil programmeren.
Dit probleem is in de Zero Eprom programmer opgelost door de informatie van en naar de programmer te laten verlopen via de adreslijnen. De commando's en de data voor de programmer worden via deze adreslijnen overgebracht en op de programmer opgeslagen in zgn. flipflops (=geheugens). Deze informatie wordt pas in de programmer opgeslagen als de betreffende Rom-/Eprom-socket, waar de programmer zich in bevindt, door de software is geselecteerd.

De te lezen of te programmeren Eprom kan op de programmer worden gestoken in de Low Extraction Force socket.

Voor het programmeren of inbranden van de Eproms is een hogere spanning nodig dan de in uw computer aanwezige +5V. Op de programmer wordt deze hoge programmeerspanning (+25V) verkregen uit de in uw computer aanwezige +5V. Er is dus geen aparte voeding noodzakelijk.

De pulsen die voor het programmeren nodig zijn moeten aan bepaalde eisen voldoen wat betreft de lengte van de puls. De referentie voor de lengte van deze puls geeft de programmer zelf af. Hierdoor is de software onafhankelijk van de snelheid van uw microprocessor.

Met de programmer kunnen verschillende types Eprom worden geprogrammeerd, de 2758/2716/2732/2532/2516. De keuze tussen deze, in principe drie, verschillende typen kan worden gemaakt door de IC-voet (14-polige) een andere stand te geven, zie hiervoor ook Hoofdstuk 3.6.

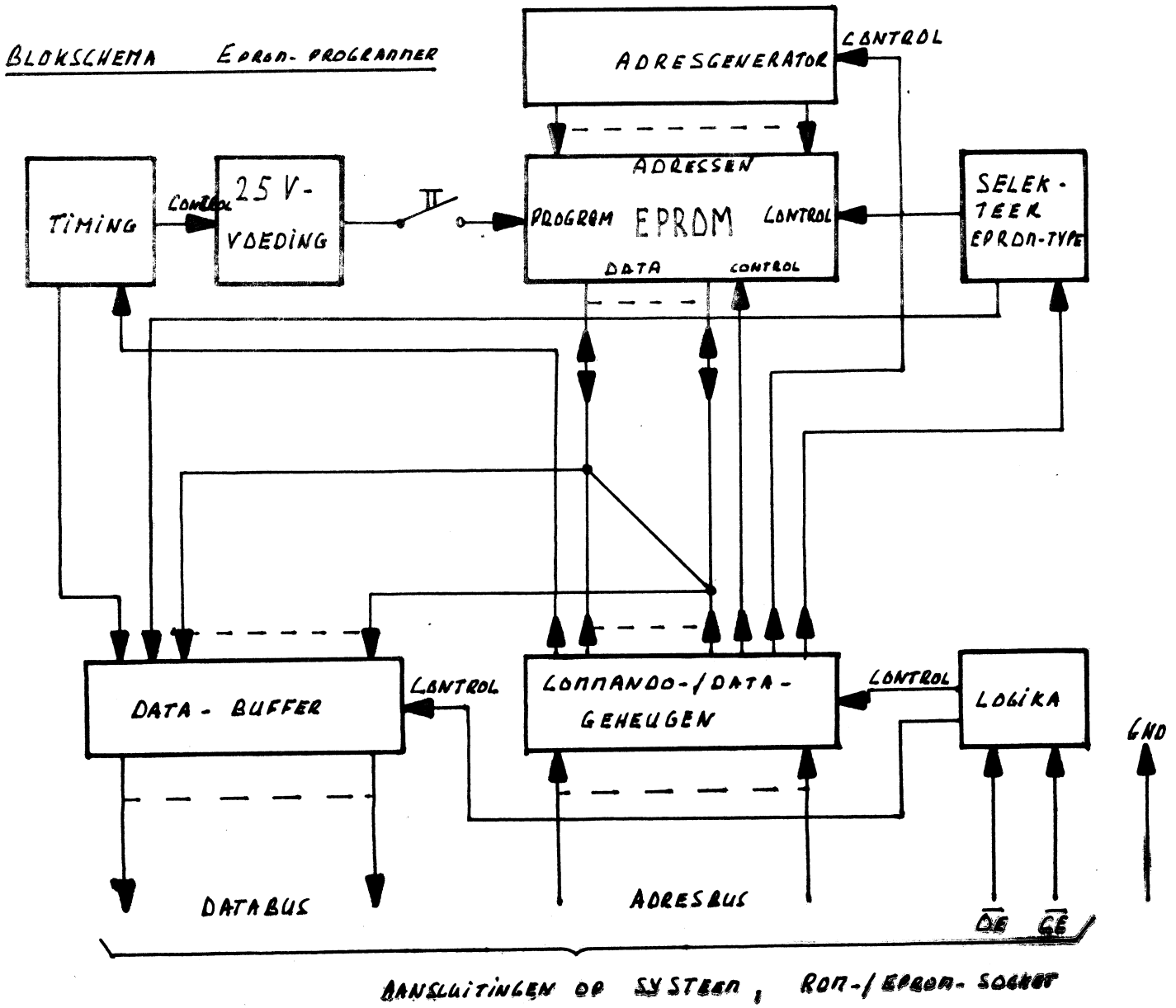
Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

Hoofdstuk 3

))) De Hardware (((

We zullen de werking van de hardware met behulp van een blokschema bespreken en daarna aangeven hoe de verschillende onderdelen uit dit blokschema in de uiteindelijke hardware zijn ingevuld.



De kern van het blokschema is de Eprom, vanuit dit blok zullen we de twee basisfuncties van de programmer bespreken. De twee basisfuncties zijn:

1. Eprom lezen
2. Eprom programmeren

3.1. Eprom lezen

Stel dat de eerste lokatie van de Eprom gelezen moet worden. Daartoe moet eerst de adresgenerator op nul worden gezet. De software zet dan een adres op de adresbus die binnen het gebied valt van de Eprom-socket waar de programmer zich in bevindt. Als dit bepaalde adres wordt aangeboden op de adresbus al de elektronika in de computer er dus zelf voor zorgen dat de selecteer-signalen voor de Rom-/Eprom-socket aanwezig zijn. Het blok 'logika' zal er dan voor zorgen dat het aangeboden adres in het commando-geheugen komt te staan en daar blijft staan. Het aanbieden van zo'n adres kan eenvoudigweg gebeuren door het uitvoeren van een leesinstructie in het betreffende adresbereik.
De uitgangen van het commando-geheugen hebben alle hun specifieke taak. In dit geval moet dus aan de adresgenerator het commando 'zet adressen voor de Eprom op nul' (=RESET adresgenerator) worden gegeven.
Als het juiste adres op de Eprom staat kan nu de data worden gelezen. Dit gaat door weer met dezelfde leesinstructie de Eprom dusdanige signalen aan te bieden zodat deze de inhoud van de gevraagde lokatie op zijn datalijnen zet.
Het uiteindelijke inlezen van de data door de microprocessor gebeurt door de logika het blok 'data-buffer' 'open' te laten zetten, zodat de data beschikbaar is op de databus van de microprocessor.
Willen we nu het volgende adres lezen dan moeten we een zodanig adres aan het commando-geheugen aanbieden, dat deze de adresgenerator het commando geeft om het aangeboden adres aan de Eprom met 1 te verhogen. Dit commando kan dan ook weer gegeven worden door een bepaald adres in het Rom-/Eprom-socket gebied te lezen.

3.2. Eprom programmeren

Voor het programmeren van een Eprom-lokatie zijn de volgende acties nodig:

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

a. Aanbieden van een adres aan de Eprom
b. Zet de 25V programmeerspanning op de Eprom
c. Zet de data die moet worden geprogrammeerd op de Eprom-datalijnen
d. Geef puls van bepaalde lengte via de Eprom-controllijnen

Stel dat we weer met de eerste Eprom-lokatie beginnen om deze te programmeren.

3.2.a. Het aanbieden van een adres gaat weer op dezelfde manier als is beschreven in Hoofdstuk 3.1.

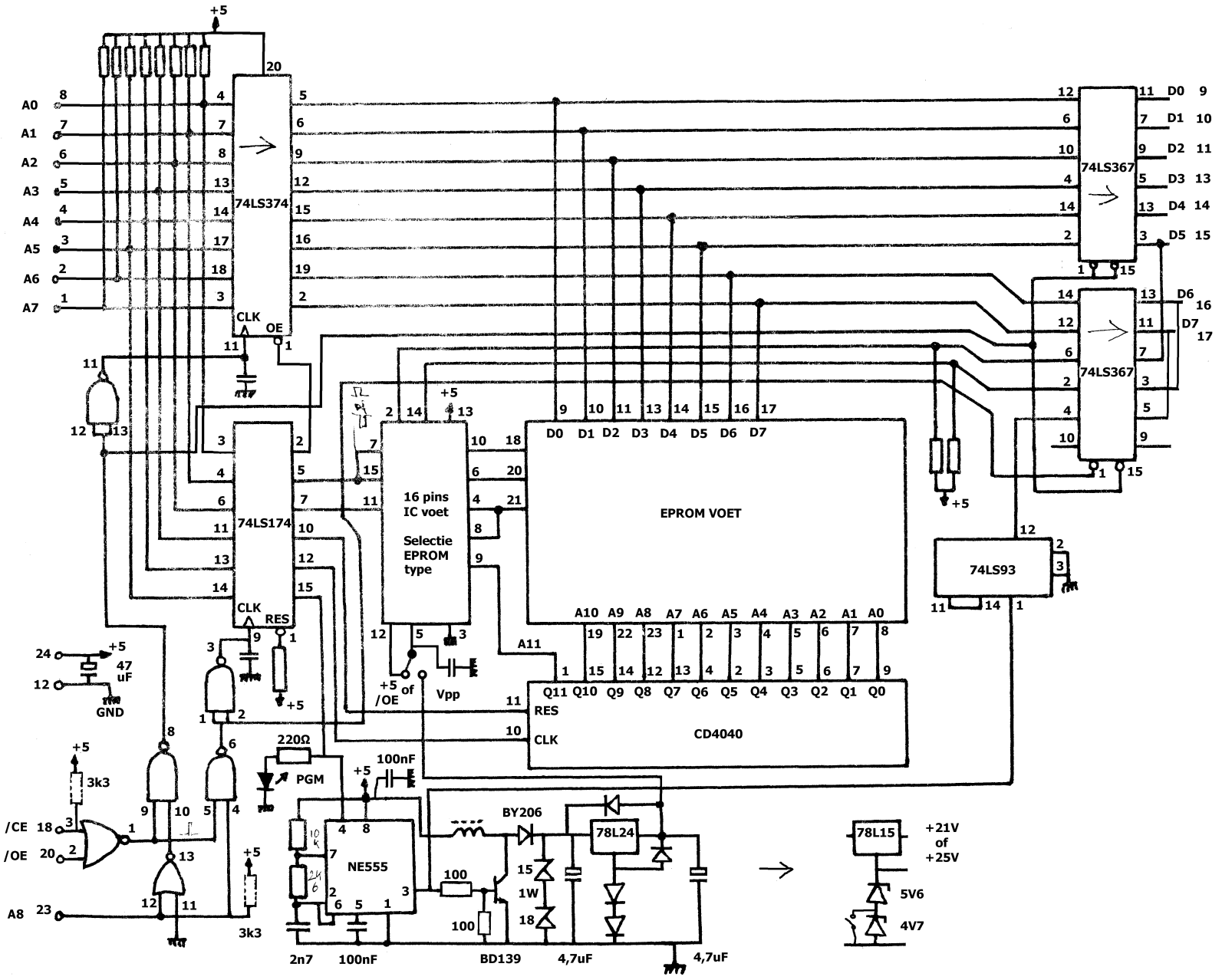
3.2.b. Door weer een bepaald adres in het socket gebied te lezen, kan in het commando-geheugen de instructie 'zet 25V aan worden gezet. Dit gebeurt door de oscillator (=een schakeling die met vaste regelmaat van spanningsnivo verandert) aan te zetten. Dit heeft tot gevolg dat er via een listige schakeling m.b.v. deze oscillator een 25V programmeerspanning kan worden opgebouwd.
Deze aldus opgewekte 25V is via een schakelaar met de programmeerpin van de Eprom verbonden.

3.2.c. Een bepaald adres aanbieden aan het data-geheugen correspondeert met de in te programmeren data. De uitgangen van het data-geheugen zijn verbonden met de datalijnen van de Eprom.

3.2.d. In het commando-geheugen wordt nu het signaal gezet dat het begin van de programmeerpuls geeft. De uitgang van het commando-geheugen is verbonden met de controllijnen van de Eprom, zodat het signaal ook door de Eprom wordt gedetekteerd.
Ogenblikkelijk hierna gaat de microprocessor de informatie van de databuffer lezen. Dit lezen is noodzakelijk om de pulsen te tellen van de oscillator welke is verbonden met de ingang van de data-buffer. Een bepaald aantal pulsen op deze ingang komt dan overeen met de benodigde programmeerpulstijd. Op deze manier is de software niet afhankelijk van de klokfrequentie van de microprocessor maar van de referentiebron op de programmer. Als er nu voldoende pulsen zijn geteld, kan de programmeerpuls worden afgezet door weer een bepaald adres in het commando-geheugen te zetten. Deze cyclus kan nu voor een andere lokatie in de Eprom worden herhaald.

3.3. De schakeling

Hier zal worden volstaan met het invullen van de blokken uit het blokschema. De precieze werking beschrijven zou voor een handleiding te ver gaan.



Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

De ADRESGENERATOR wordt gevormd door een 12-bitteller, CD4040B, die gereset kan worden.

Het SELECTEREN EPROM-TYPE gebeurt door de stand van de 14-pins IC-voet, die d.m.v. bepaalde op de voet aangebrachte doorverbindngen, te wijzigen.

De LOGIKA wordt gevormd door enkele NOR- en NAND-poorten, deze logika geeft de signalen voor de besturing van het commando-geheugen, data-geheugen en de data-buffer.

Het COMMANDO-GEHEUGEN wordt gevormd door een 6-bit register, 74LS174. Het commando-geheugen bestuurt de oscillator (25V), de adresgenerator, het data-geheugen en de Eprom.

Het DATA-GEHEUGEN bestaat uit een 8-bit register, 74LS374 Omdat de Eprom ook als uitgang kan fungeren evenals het data-geheugen, moet het data-geheugen op tri-state kunnen worden gezet. Dit gebeurt onder controle van de 74LS174. De 74LS374 biedt dus de te programmeren data voor de Eprom aan.

De DATA-BUFFER is een 12-bit tri-state buffer, 2*74LS367. Via deze buffer kan de timing worden gelezen, de stand van het 14-pins IC-voetje voor de Eprom-selectie en verder kan hiermee de data uit de Eprom worden gelezen.

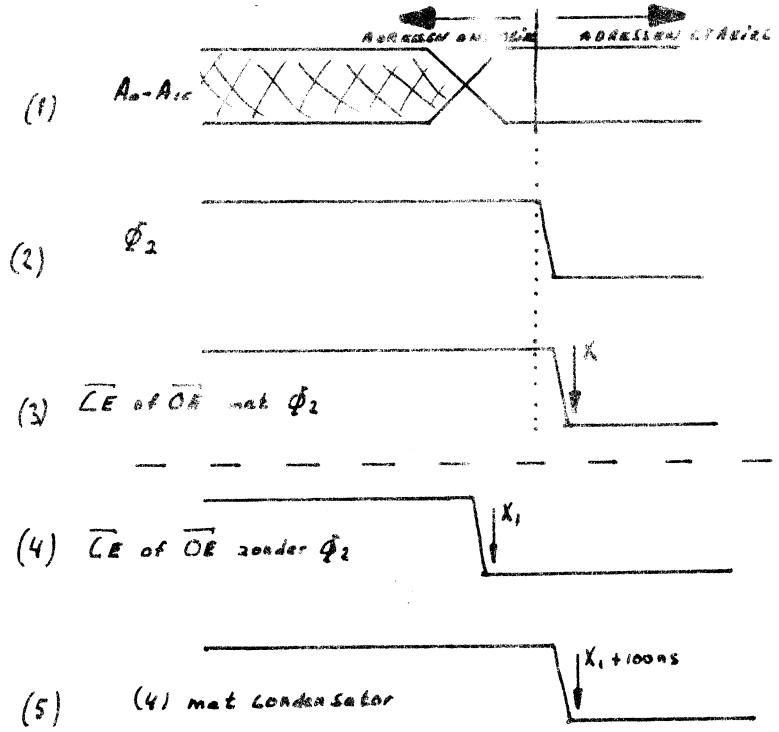
De TIMING voor de programmeerpuls wordt geleverd door de 74LS93, die het aangeboden oscillatorsignaal door zestien deelt.

De VOEDING is opgebouwd rond een oscillator met een 555, timer IC. De oscillator zorgt ervoor dat de transistor met de spoel een hoge spanning kan opbouwen over de condensator aan de ingang van de spanningsregelaar, 78L24. De zenerdiode begrenst de ingangsspanning voor de spanningsregelaar. De diode in serie met de condensator houdt het weglekken van de lading tegen. De diode in de minleiding van de spanningsregelaar zorgt ervoor dat de uitgangsspanning van de schakeling op 25V komt te liggen.

3.4. CE en OE (Chip Enable en Output Enable)

Deze signalen spelen een sleutelrol in de besturing van de programmer. Pas als deze signalen beide 'laag' of 'logisch nul' zijn, worden commando's of data door de programmer geaccepteerd.

Over de klok-ingangen van de IC's 74LS174 en 74LS374 zijn condensatoren naar de min aangebracht. De reden hiervoor is dat ondanks dat de signalen CE en OE aanwezig zijn, het niet altijd hoeft te zijn dat de dan aangeboden adressen al stabiel zijn. Dat stabiel zijn van de adressen is echter een eis voor goed functioneren van de programmer, omdat de programmer de adressen binnenhaalt en er daarna niet meer naar kijkt.
Dit probleem is er niet als er in de socket een Rom of een Eprom zit, want dan kan de Rom of de Eprom toch niet gelijk reageren door de relatief grote toegangstijd van deze IC's.
Dit probleem speelt alleen bij de microprocessoren 6502 en 6800. Bij deze processoren ziet de timing er als volgt uit:



X - het moment waarop de programmer de commando's via de adreslijnen overneemt.
(2)- het signaal Ø2(spreek fie 2) geeft aan dat de adressen op dat moment stabiel zijn. Dit signaal is beschikbaar aan een uitgang van de microprocessor.
(3)- als Ø2 is verwerkt in de opwekking van CE of OE dan gaat het altijd goed.

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

(4)- als Ø2 niet is verwerkt in de opwekking van CE of OE dan kan het fout gaan.
(5)- Door het toepassen van condensatoren over de klok-ingangen van de IC's 74LS174 en 74LS374 wordt het signaal X1 met 100ns vertraagd, zodat een stabiel adres wordt gelezen.

De bij (5) bereikte vertraging is meestal voldoende. Het zou echter mogelijk kunnen zijn dat als b.v. de adreslijnen erg krap zijn gebufferd het extra lang duurt voordat de adressen stabiel zijn. Het niet goed functioneren van het zojuist besprokene uit zich meestal in het niet goed programmeren. Het beste is om dan het signaal Ø2 aan te sluiten op CE of OE van de programmer of Ø2 te verwerken in de opwekking van CE of OE.

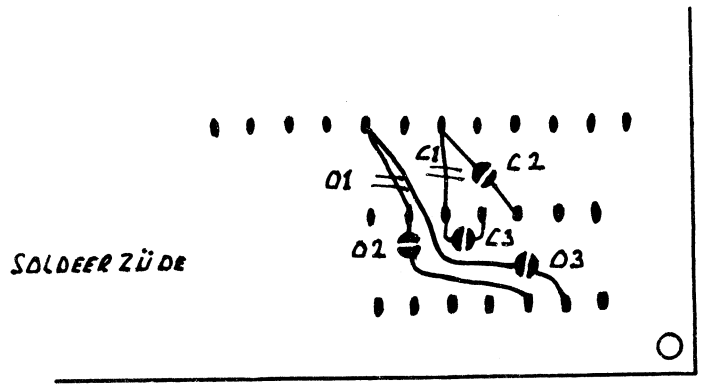
Opmerking !!:
Hopelijk bent u niet bang gemaakt door het voorgaande verhaal. Het hierboven beschreven probleem zal in de praktijk bij de bekende computers niet of nauwelijks voorkomen!!!
Maar er zijn echter zelfbouwers die mogelijkerwijs iets afwijkends hebben ontworpen of iets aan de bestaande computer hebben gewijzigd.

3.5. Rom-sockets

Een Eprom wordt normaal gesproken geselecteerd door het 'laag' of 'logisch nul' worden van de signalen OE en CE (OE=Ø en CE=Ø). Dit geldt ook voor de meeste Roms. Maar er kunnen Roms zijn die bijvoorbeeld een combinatie van OE=1 en CE=Ø of OE=1 en CE=1 nodig hebben. Het is mogelijk om de programmer ook met deze afwijkende sockets te laten werken. Het is namelijk zo dat om een '1' in een 'Ø' te veranderen, er een inverter (=omkeerder) nodig is. Op de programmer zijn nog twee inverters hiervoor gereserveerd, zodat de afwijkende signalen eerst via de inverters kunnen worden omgekeerd .

Aanpassing om OE te inverteren: onderbreek het printspoor bij O1 en soldeerdruppels op O2 en O3.

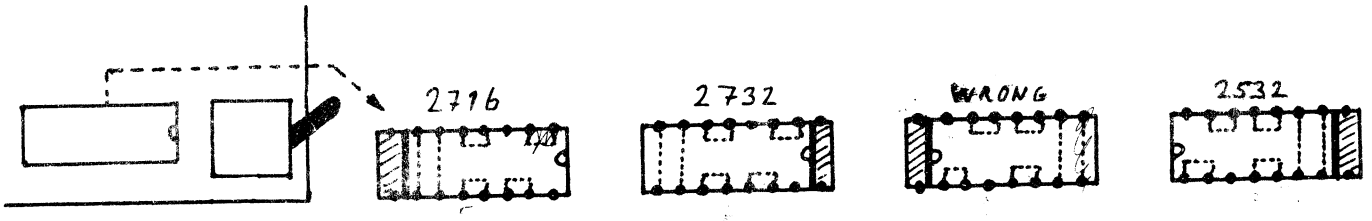
Aanpassing om CE te inverteren: onderbreek het printspoor bij C1 en soldeerdruppels op C2 en C3.



Overigens kan de programmer wel in een 2708-socket worden gestoken.

3.6. Selectie Eprom-type

Stand van 14-polige IC-voet in 16-polige IC-voet voor de selectie van het Eprom-type



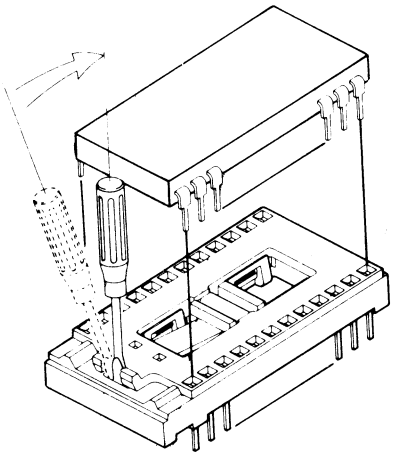
De stand 'wrong' kan destructief zijn voor de Eprom als de schakelaar in de 25V-stand staat!!!!!

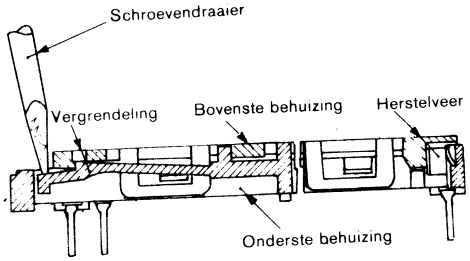
3.7. Low Extraction Force IC-socket

Het voordeel van deze socket is dat de Eprom met weinig kracht eruit gehaald kan worden. Voor het erin zetten van de Eprom moet de socket op dezelfde manier worden behandeld als een gewone IC-voet, dus gewoon stevig duwen tijdens het erin zetten.
Het verwijderen van de Eprom is hieronder schematisch weergegeven:

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S. C.





Hoofdstuk 4

))) De Software (((

De software voor de Zero Eprom programmer is zeer uitgebreid, bijna 2kBytes. Alle bewerkingen die nodig zijn om een stuk geheugen in een Eprom te programmeren zijn aanwezig.
De besturing van de programmer is dialooggericht. Deze dialoog verloopt via toetsenbord en beeldscherm.

4.1. Het aanpassen van de software

De twee belangrijkste aanpassingen die nodig zijn om de sofware te kunnen laten draaien, is het aanpassen van de invoer- en uitvoerroutines. Overigens is aanpassen alleen dan nodig als u geen startklare versie heeft. Deze startklare versie is er voor enkele populaire microcomputers.

- De invoer-routine is een subroutine die wacht tot er een toets op het toetsenbord wordt ingedrukt. Als de toets is ingedrukt moet de processor uit de subroutine terugkomen. Bij het verlaten van deze subroutine moet de ASCII-code van de ingedrukte toets in register A of accumulator A van de microprocessor staan.

- De uitvoer-routine is een subroutine die de ASCII-code in het register A of de accumulator A van de microprocessor afdrukt op het beeldscherm.

Het moge duidelijk zijn dat de plaatsen van deze subroutines van computer tot computer zullen verschillen. Deze subroutines staan over het algemeen in het monitorprogramma (=standaard besturingsprogramma) van uw systeem. Mocht u niet achter de plaats van deze subroutines kunnen komen, dan kan iemand van een gebruikersclub u misschien verder helpen.

Voor de verschillende software-versies volgen hier de plaatsen waar de adressen van de invoer- en uitvoerroutines moeten worden ingevuld:

```
6800     :    0104 : high byte adres invoer-routine
----          0105 : low  byte adres invoer-routine

              0121 : high byte adres uitvoer-routine
```

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S. C.


```
                0122 : low  byte adres uitvoer-routine

6502     :     0808 : low  byte adres invoer-routine
----           0809 : high byte adres invoer-routine

               0820 : low  byte adres uitvoer-routine
               0821 : high byte adres uitvoer-routine

8080/Z80:      4807 : low  byte adres invoer-routine
--------       4808 : high byte adres invoer-routine

               481F : low  byte adres uitvoer-routine
               4820 : high byte adres uitvoer-routine
```

Verder zijn er nog 4 bytes die aangepast kunnen worden:

- Echo , dit heeft betrekking op de invoer-routine. Als de door gevonden invoer-routine de ingedrukte toets gelijk op het scherm zet, dan heet dat 'echo'. Gebeurt dit niet dan heeft u geen 'echo'.
- Backspace , hier komt de ASCII-code te staan die er in uw systeem voor zorgt dat een ingetypt karakter weer kan worden gewist ('pijltje terug' of 'Delete').
- Escape , dit heeft te maken met de programmer-software. Stel dat u zich in de programmer-software bevindt en dat u allerlei adressen heeft ingetypt die de programmer-software moeten aangeven om b.v. te gaan programmeren, maar u heeft daarmee een fout gemaakt, dan kunt u het Escape-karakter intypen en komt u weer terug bij het begin. U vult dus op dit adres uw eigen Escape-code in. (ASCII)
- Return , afsluitkarakter voor een ingetypte zin.

De adressen voor de verschillende processors zijn:

```
6800 :  Echo      , 010E   er staat: 04, geen echo:04, wel echo:15
        Backspace, 0155        ,,   : 08
        Escape    , 010C       ,,   : 1B
        Return    , 0163       ,,   : 0D

6502 :  Echo      , 080C       ,,   : 20, geen echo:20, wel echo:4C
        Backspace, 0860        ,,   : 5F
        Escape    , 086C       ,,   : 1B
```

```
        Return    , 0873        ,,   : 0D

8080 :  Echo      , 480C        ,,   : 11, geen echo:11, wel echo:22
        Backspace, 4849         ,,   : 08
        Escape    , 4858        ,,   : 40
        Return    , 485D        ,,   : 0D
```

4.2. Mogelijkheden van de software

Als de software is opgestart (hoe het opstarten precies verloopt, komen we later nog terug, Hoofdstuk 5), dan meldt deze zich met het geven van een checksumbyte. Een checksum representeert de optelsom van alle data en instructies uit het programma. Als nu iets bij het starten van de software niet goed is gegaan dan komt dat bijna altijd tot uiting door het geven van de verkeerde checksum. Als deze checksum is afgedrukt, wordt het volgende gevraagd:

```
BASE ADR. PROM SOCKET: ....   ;hier kan het hexadecimale
                              ;adres worden ingetypt
                              ;waar de programmer zich in
                              ;bevindt.
```

Als dit is geaccepteerd verschijnt het menu met de standaardmogelijkheden van de software:

```
THE PROGRAMMER IS IN THE XXXX MODE

1 READ PROM INTO MEMORY
2 PROG PROM
3 COMPARE WITH MEMORY
4 CHECK IF EMPTY
5 LIST PROM CONTENTS

CHOOSE NOW: .
```

Voor XXXX kan staan : 2716, 2732 of 2532 dit is afhankelijk van de stand van het 14-polige IC-voetje.

Bij de vraag CHOOSE NOW moet het nummer voor de bijbehorende mogelijkheid worden gekozen.

4.2.a. READ PROM INTO MEMORY/PROG PROM/COMPARE WITH MEMORY

De benodigde gegevens voor het uitvoeren van bovenstaande bewerkingen zien er als volgt uit:

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.


```
(1)  FIRST MEM. BYTE: ....
(2)  LAST  MEM. BYTE: ....
(3)  BYTE COUNT     : ....
(4)  FIRST PROM BYTE: ....
(5)  LAST  PROM BYTE: ....
```

De hexadecimale getallen worden op een intelligente manier verwerkt, d.w.z. dat b.v. 0000 mag worden getypt, maar 0 ook. Niet hexadecimale getallen worden niet geaccepteerd.
Er kunnen dus vijf variabelen worden ingevuld. Eventuele ontbrekende gegevens worden voor zover mogelijk door de software uitgerekend, deze worden dan ingevuld en afgedrukt:

```
THE COMPLETE DATA IS:

FIRST MEM. BYTE: AAAA
LAST  MEM. BYTE: BBBB
BYTE COUNT     : CCCC
FIRST PROM BYTE: DDDD
LAST  PROM BYTE: EEEE

IS THIS OK? .                ;alleen bij intypen 'Y' wordt
                             ;de bewerking uitgevoerd.
```

- READ PROM INTO MEMORY, Eprom naar RAM kopieren

(1) en (2) , hier kan worden aangegeven waar het gedeelte van de Eprom naar toe moet in RAM.
(3) , het aantal in RAM te zetten bytes.
(4) en (5) , het gedeelte van de Eprom wat in RAM moet worden gezet.

- PROG PROM, programmeren van een Eprom

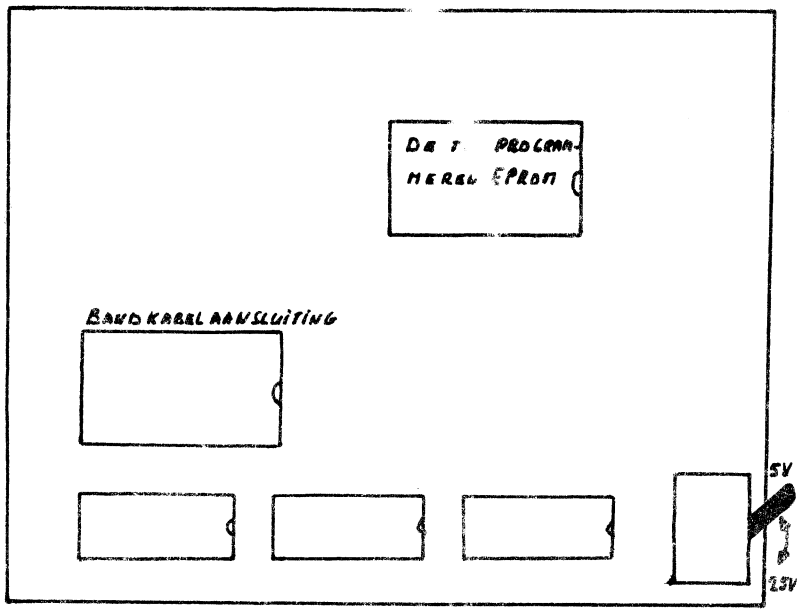
(1) en (2) , gebied in RAM wat naar Eprom moet worden geprogrammeerd.
(3) , aantal te programmeren bytes.
(4) en (5) , gebied in Eprom waar naar toe moet worden geprogrammeerd.

Na de vraag 'IS THIS OK?', verschijnt de volgende vraag:

```
--- SURE ABOUT PROGRAMMING? --- .
```

Alles behalve 'Y' is 'N'.
Deze vraag is bedoeld als extra voorzorgsmaatregel. Nu zijn we aan het programmeren toe en wordt de opdracht gegeven:

```
SWITCH VPP TO 25 VOLT        ;de software wacht hier op het
                             ;indrukken van een willekeu-
                             ;rige toets. Druk deze
                             ; toets pas in als u de scha-
                             ;kelaar op 25 Volt heeft ge-
                             ;zet !!!!!!
```



Nu gaat de programmer programmeren !!!!!

Ter indicatie van de programmeertijd voor de Eproms:
2716 - ongeveer 100 seconden
2732 - ongeveer 200 seconden

Als het programmeren is afgelopen wordt de volgende opdracht gegeven:

```
SWITCH VPP TO 5 VOLT         ;Nadat de schakelaar
                             ;is teruggezet naar 5 Volt,kan
                             ;een willekeurige toets worden
                             ;ingedrukt.
```

Nu het programmeren dus klaar is kan de software gaan controleren of de Eprom goed is geprogrammeerd. Klopt het

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

niet dan worden de afwijkende lokaties afgedrukt.

- COMPARE WITH MEMORY, vergelijken van de Eprom met RAM

(1) en (2) , het stuk RAM waarmee een gedeelte van de Eprom kan worden vergeleken.
(3) , het aantal te vergelijken bytes.
(4) en (5) , het gedeelte uit de Eprom dat met Eprom moet worden vergeleken.

4.2.b. CHECK IF EMPTY/LIST PROM CONTENTS

De benodigde gegevens om bovenstaande twee opdrachten te kunnen uitvoeren zijn:

(1) BYTE COUNT : ....
(2) FIRST PROM BYTE: ....
(3) LAST PROM BYTE: ....

De ontbrekende gegevens worden uitgerekend maar niet afgedrukt.

- CHECK IF EMPTY, controleren of de Eprom leeg is

(1) , het aantal bytes in de Eprom waarbij moet worden gekeken of deze lokaties 'leeg' zijn.
(2) en (3), het gedeelte in de Eprom waarvan moet worden gecontroleerd of deze leeg is.

Overigens als een Eprom-lokatie gevuld is met 'enen' ofwel hexadecimaal FF, dan kan deze nog geprogrammeerd worden. Dus door te programmeren kan van een '1' een '0' worden gemaakt en niet andersom.

-LIST PROM CONTENTS, het afdrukken van de Eprom inhoud

(1) , het aantal bytes uit de Eprom dat moet worden afgedrukt.
(2) en (3), het gedeelte uit de Eprom wat moet worden afgedrukt.

Hoofdstuk 5

))) Het starten van de programmer (((

We nemen aan bij de bespreking van het opstarten van de programmer dat u voorgaande hoofdstukken heeft gelezen. Er zal regelmatig worden terugverwezen.

Er zijn drie mogelijkheden:

1. Er is een Eprom met de universele software erin (5.1.)
2. Er is een Eprom met startklare software (5.9.)
3. Er is helemaal geen Eprom met software (5.10.)

5.1. Eprom erin

U steekt de Eprom in de vrijgemaakte socket. Deze socket is geschikt voor Eproms of de socket is daarvoor geschikt gemaakt. (zie ook Hoofdstuk 3.5.)

5.2. De koude start

Stel dat de socket het gebied beslaat van 8000-87FF (dit is hexadecimaal, dus 2048 bytes). U springt nu met uw monitorprogramma of wat voor programma dan ook naar in dit geval: 8000H + COLDM.
De waarde van COLDM is voor iedere processor weer anders en het adres in uw geval kunt u vinden bij de hexadecimale dump in Hoofdstuk 7.

5.3. Kopieren naar RAM

Dus u heeft de sprong uitgerekend en uitgevoerd. Nu gaat de software in de Eprom uitzoeken waar hij zich bevindt, want dat kan hij niet weten. Dit wordt berekend via de stackpointer. Nu gaat hij zichzelf naar RAM kopieren omdat het programma uiteindelijk in RAM moet gaan draaien. De plaats in RAM is aangegeven op de hexadecimale dump in Hoofdstuk 7. Mocht u een andere versie hebben, dan is dit aangegeven op de Eprom-sticker van de bijgeleverde Eprom, d.m.v. ORG XXXX, waarbij XXXX het eerste RAM-adres is waar de software naar toe gaat.

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S. C.

5.4. Sprong naar RESET-vector

Als het in RAM zetten is gebeurd, springt de software naar de RESET-vector van de microprocessor, waar dan ook meestal het monitorprogramma van de computer staat.

5.5. Aanpassen software

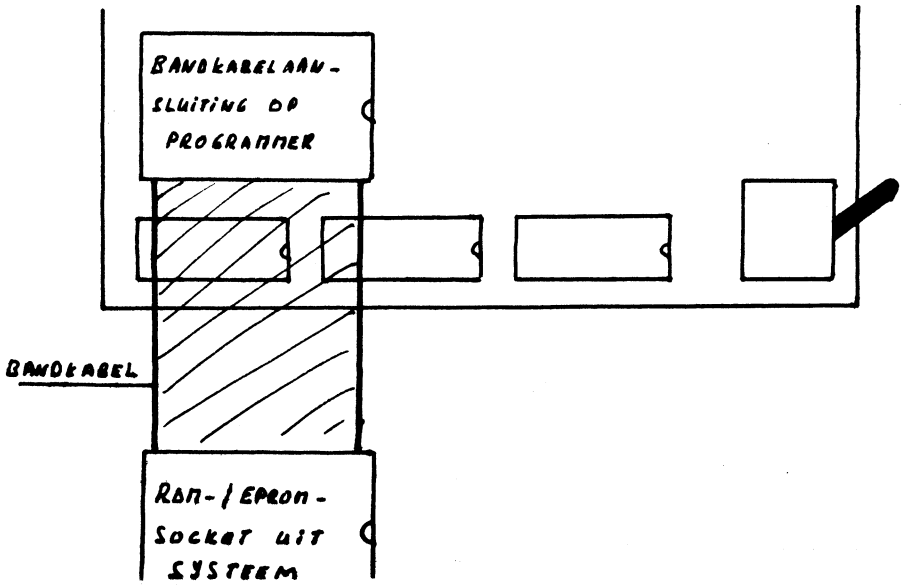
Het programma staat nog steeds in RAM. Nu kunnen de aanpassingen gemaakt worden die nodig zijn om het programma op uw eigen computer te kunnen laten draaien, zie ook Hoofdstuk 4.1. Houdt rekening met een eventuele andere ORG-versie, de adressen waar de invoer- en uitvoer-routines komen te staan veranderen dan evenredig.

5.6. Het programma wegzetten

Op dit moment kunt u het beste het stuk RAM waar de programmer-software staat, dus met de aangepaste adressen, op disk/cassette/ponsband enz. wegzetten. Hierdoor spaart u zich de volgende keren tijd.

5.7. Het inzetten van de programmer

Let erop dat de schakelaar op de programmer in de 5Volt-stand staat, zie tekening in 4.2.a.
De Eprom is op dit moment niet meer nodig en kan verwijderd worden.
Nu kan de programmer erin:



Het is beter om de computer tijdens deze handeling uit te zetten omdat tijdens het insteken van de programmer de computer op hol kan slaan door mogelijke storing op de processorbussen.

5.8. De warme start van de software

Nu de programmer erin zit, kan de software weer van disk/cassette/ponsband enz. gehaald worden, als deze is weggezet zoals beschreven bij 5.6.
Nu kan de software worden gestart met het RAM-startadres, dit heet WARM. Het adres van WARM is afhankelijk van de processorversie en dit staat in Hoofdstuk 7. Bij een andere ORG-versie (ORG XXX op de Eprom-sticker), is het WARM startadres gelijk aan XXXX.
Het adres van WARM is dus altijd de eerste RAM-lokatie.
De programmer zal zich melden met het geven van een checksum-byte, zie ook 4.2. Dit moet overeenstemmen met het getal dat op de Eprom-sticker staat. Als het niet klopt dan zou dat kunnen komen doordat u de 'backspace'- en 'escape'-karakters anders heeft gekozen dan in de originele versie. De checksum verandert daarna namelijk evenredig.

5.9. Startklare versie

Voor sommige bekende microcomputers is een startklare versie beschikbaar. Er zijn twee mogelijkheden:

a. Via de Eprom het programma starten. Volg dan de punten 5.1, 5.2, bij 5.2 echter moet u niet starten met COLDM maar met COLDG, de software wordt dan gelijk gestart. Verder gaan met 5.6, 5.7 en 5.8
b. De Eprom met een monitorprogramma naar RAM kopieren en verder gaan met punt 5.6, 5.7 en 5.8

5.10. Alleen een hexadecimale dump

U staat nu voor de ondankbare taak de hexadecimale listing uit Hoofdstuk 7 in te typen, daarna verder met punt 5.5, 5.6, 5.7 en 5.8

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero S.C.

Hoofdstuk 6

))) Losse opmerkingen en tips (((

- De dialoog is in principe niet geschikt voor het werken met een hexadecimaal toetsenbord en display. De besturingssoftware is hiervan niet afhankelijk. U zult in dit geval een terminal moeten aansluiten of de dialoogsoftware zelf moeten aanpassen.

- Houdt er rekening mee dat de software in RAM staat, dus naar dat RAM gedeelte kunnen geen data of programma's.

- Bij de 6800 en 6502 microprocessors kan de zero-page niet geprogrammeerd worden omdat de inhoud van de zero-page tijdens het programmeren gedeeltelijk verandert.

- De inhoud van het videogeheugen is ook niet te programmeren omdat deze steeds verandert tijdens het programmeren.

- Geef tijdens het programmeren geen RESET omdat anders de programmeersspanning op de Eprom kan blijven staan.

- Volgens de fabrikanten van de Eproms mag een Eprom niet onder spanning (5 Volt) verwijderd worden. Eerst zou de spanning moeten worden uitgezet.
  Onze ervaringen hebben uitgewezen dat u zich daarover geen zorgen hoeft te maken en gewoon de Eprom kunt verwijderen terwijl de +5 Volt aanwezig blijft.

- Tijdens de eerste start van de programmer kunt u ook gewoon programmeren zonder dat er een Eprom inzit of zonder dat de schakelaar in de 25 Volt-stand staat. Dit kan nuttig zijn om te kijken of de programmeersoftware goed draait.

- De programmer kan niet als Epromkaart worden gebruikt. De data uit de Eprom kunnen alleen via speciale programma's worden gelezen.

- Er kunnen alleen Eproms geprogrammeerd worden die single +5 Volt zijn. Dus b.v. geen 2716's van het fabrikaat Texas Instruments, want deze hebben verschillende voedingsspanningen.

Leeg tussenvel

Prinsenstraat 49
2983 CH RIDDERKERK
Tel.: 01804 - 13230
Bank: Rabobank Ridderkerk
Rek. nr.: 35.54.95.562

Zero Software Consultants

ZERO S. C.
BERGWEG - NOORD 38-2
2661 CR BERGSCHENHOEK
THE NETHERLANDS
TEL. 01892 - 5333

Het gebruik van 2716 Eprom's in een computer voor 2708 heeft enkele voordelen: 1) de 2716 is beter verkrijgbaar
2) de 2716 is goedkoper
3) de 2716 is eenvoudig te programmeren
4) lager energieverbruik :

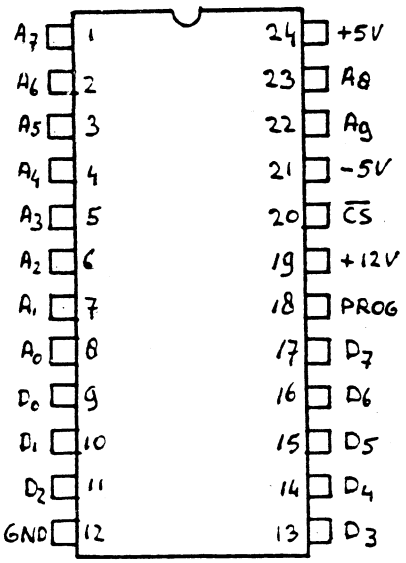
2708	800 mW
2716	132 mW (standby)
	525 mW (active)

Nu blijkt het mogelijk een adapter te maken zodat een 2716 als 2708 gebruikt kan worden.
Zo'n adapter zou uit twee IC voeten kunnen bestaan:

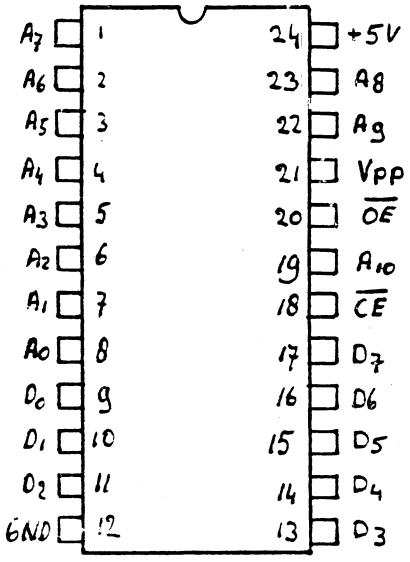
IC voet 1:	(24 pins)	afknippen pinnen 18,19 en 21
IC voet 2:	(24 pins)	verbinding pin 18 en 20
		verbinding pin 21 en 24
		verbinding pin 19 en 12

Steek nu voet 2 in voet 1 en klaar is de adapter.

pinout 2708



pinout 2716



Leeg tussenvel

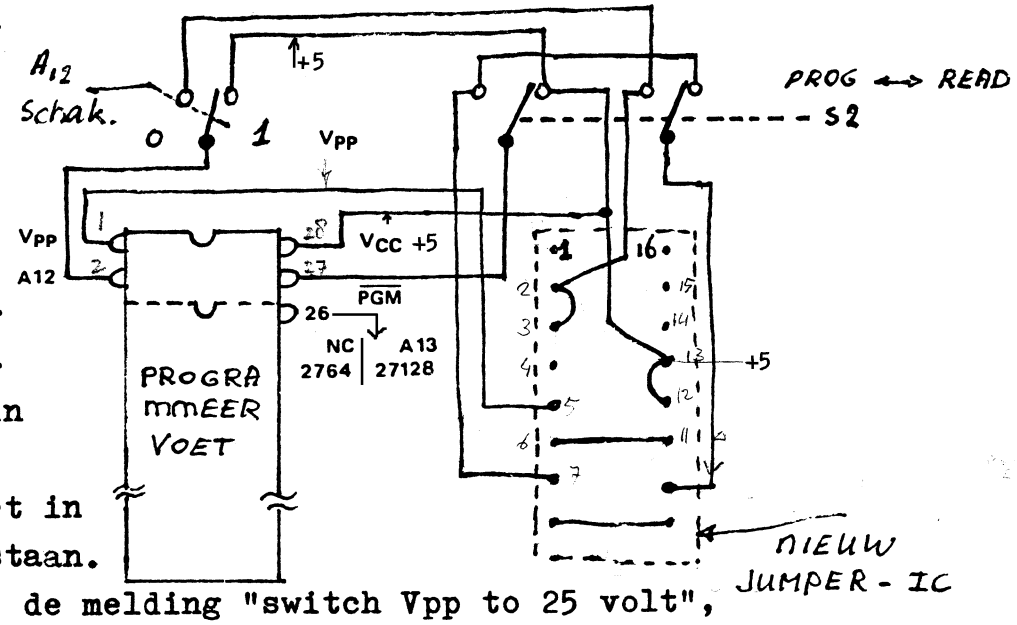
Aanpassing van de ZERO EPROM programmer,zodanig dat er

2764 EPROMs mee geprogrammeerd kunnen worden.

Benodigdheden: 1 28 pins socket
1 schakelaar dubbel om
1 schakelaar enkel om
1 zener 4v7 400mW
1 zener 5v6 400mW
1 spannings regelaar 78115

indien Uw programmer reeds met een 78115 is uitgerust,dan zijn de laatste 3 componenten niet nodig.
U vindt naast het IC NE555 eerst 4 weerstanden en dan 3 diodes De eerste diode vervangt U door een zener van 4V7 (streepje richting programmeersocket) en de tweede diode vervangt U door een zener van 5v6 (richting idem).Vervolgens vervangt U het component direct naast de schakelaar (78124) door een 78 l 15.Het soldeer-eilandje aan de onderkant van de print ,onder de eerste zener van 4v7 kan nu doorverbonden woden voor 2732A en 2764 EPROMs,of opengelaten worden voor 2716, 2732,en 2532 EPROMs.

Verwijder nu het jumper IC en vervang dit door nevenstaande schakeling,incl. nieuwe 28 pin programmeervoet. De A12 schakelaar is om de te programmeren helft in te stellen.
Schakelaar S2 moet in de stand "read" staan.



Programmeren: bij de melding "switch Vpp to 25 volt", eerst S2 in de stand "prog",daarna de altijd gebruikte schakelaar in de stand "25 volt" zetten en dan pas de spatiebalk indrukken.
Bij terug schakelen: eerst 25volt afschakelen,daarna S2 omzetten en pas daarna spatie geven. Het is normaal dat de programmer in de 2732 mode staat.

Leeg tussenvel

De bij de Zero SC Prom Programmer geleverde software voor gebruik bij de Exidy Sorcerer wordt eerst verplaatst naar het geheugenblok 4800H-4FFFH, en daarna gestart. Het programma zelf is niet relocatable en werkt dus alleen op genoemde adressen. Dit betekent onder meer dat voor gebruik van de Z.SC programmer minimaal een 32K Sorcerer vereist is.
Om deze beperking van de overigens zeer goed bruikbare programmer te omzeilen heb ik een aangepaste versie gemaakt die werkt in de EPROM waarin het is geplaatst. Alleen de flags, pointers en databytes, welke door het programma worden gebruikt en gewijzigd, zijn in RAM geplaatst en wel in het blok van 0190H tot 01AFH. In dit blok mag geen file worden geplaatst welke bestemd is om in EPROM te worden geprogrammeerd.
De RUB toets (zonder shift) kan worden gebruikt om het laatst ingetypte karakter te wissen terwijl ESC bestemd is om op ieder willekeurig moment terug te keren naar het menu. CTRL/C dient om het programma te verlaten naar de Exidy Monitor. Terugkeer naar de programmer gebeurt met 'PP'.
In deze versie wordt niet naar het Base Adr. Prom socket gevraagd. Het programma vult dit adres zelf in.
Voorts is er een zesde routine toegevoegd waarmede het mogelijk is te testen of bij het programmeren van een EPROM (gedeelte) een bit van 0 naar 1 moet worden geset. Deze aanvulling is gemaakt om te kunnen testen of, zonder eerst de EPROM te moeten wissen, wijzigingen daarin kunnen worden aangebracht. Immers, slijtage van de EPROM vindt hoofdzakelijk plaats bij het wissen. De werkwijze van deze routine is gelijksoortig aan die van de overige programmaonderdelen en behoeft dan ook geen nadere uitleg. Er is niet voorzien in een listingsmogelijkheid van die locaties die test 6 niet doorstaan.

Tenslotte enige aanwijzingen bij het vervangen van de jumper door een 3 standen 6mc schakelaar.
De nummers in de volgende tabel verwijzen achtereenvolgens naar de aansluitnummers van de jumpervoet aan de 6 schakeldekken.

moedercontact		stand 1 2716	stand 2 2732	stand 3 2532
dek 1	10	7	7	9
dek 2	4	5	9	5
dek 3	6	11	5	7
dek 4	12	13	11	13
dek 5	2	nc	3	7/15
dek 6	14	7	13	3

7 ≙ 15
4 ≙ 8

Deze hardware aanpassing maakt inbouw in een kastje zeer simpel omdat de print niet bereikbaar behoeft te blijven. Selecteren van een EPROM type kan on line geschieden door de drie standenschakelaar op de gewenste stand te zetten en de ESC. toets in te drukken.

Bij bovengenoemde SW aanpassing is er vanuit gegaan dat de connector wordt aangesloten op (EP)ROMpack locatie 3 (D000H) en de programma EPROM op locatie 4 (D800H).

De bij de programmer geleverde EPROM bevat een andere programmafile dan die waarvan de hex.listing is afgedrukt in de handleiding. Een juiste hex.listing van het door mij aangepaste programma is hierbij gevoegd.

Het aangepaste programma, de beschrijving daarvan evenals de hex.listing mag vrij worden gepubliceerd dan wel verstrekt.

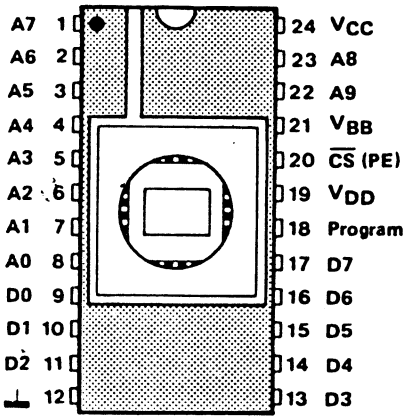
Gouda, 22 juli 1981

Floor Vogelaar, Karnemelksloot 197 2806 BE GOUDA

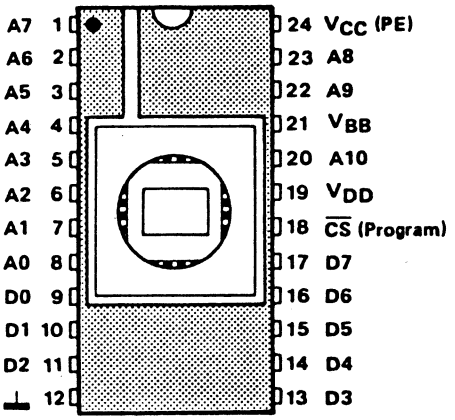
Leeg tussenvel

elektuur infokaart 76 | μ-computer-IC's 4 | EPROM's aansluitingen

2708

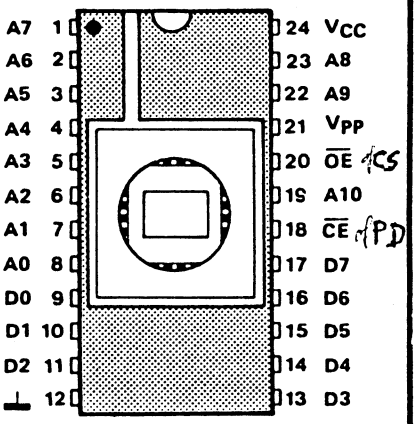


2716
Texas Instruments



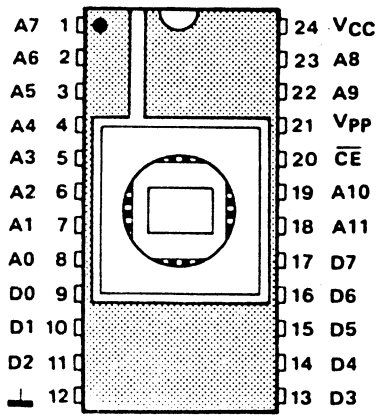
2516
Texas Instruments

2716
andere fabrikaten

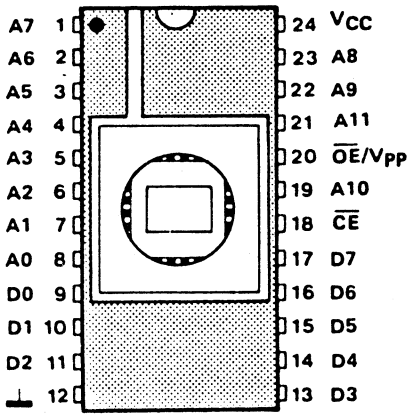


elektuur infokaart 76 | μ-computer-IC's 4 | EPROM's aansluitingen

2532



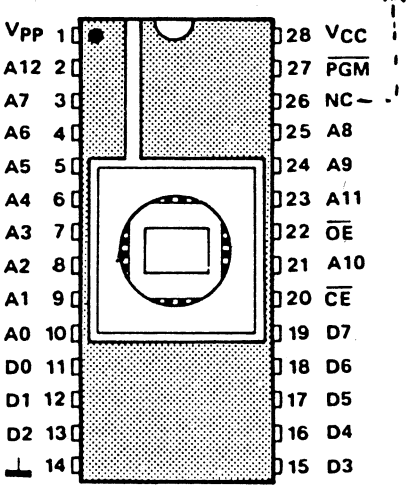
2732



27128 - - - ?

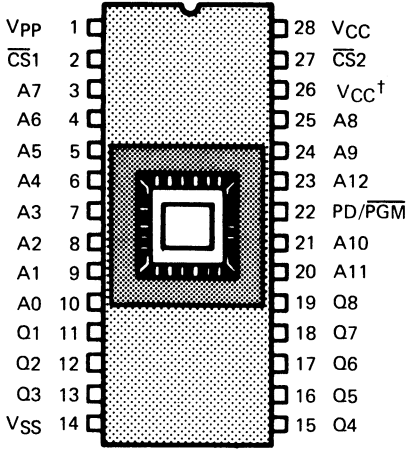
2764

A13



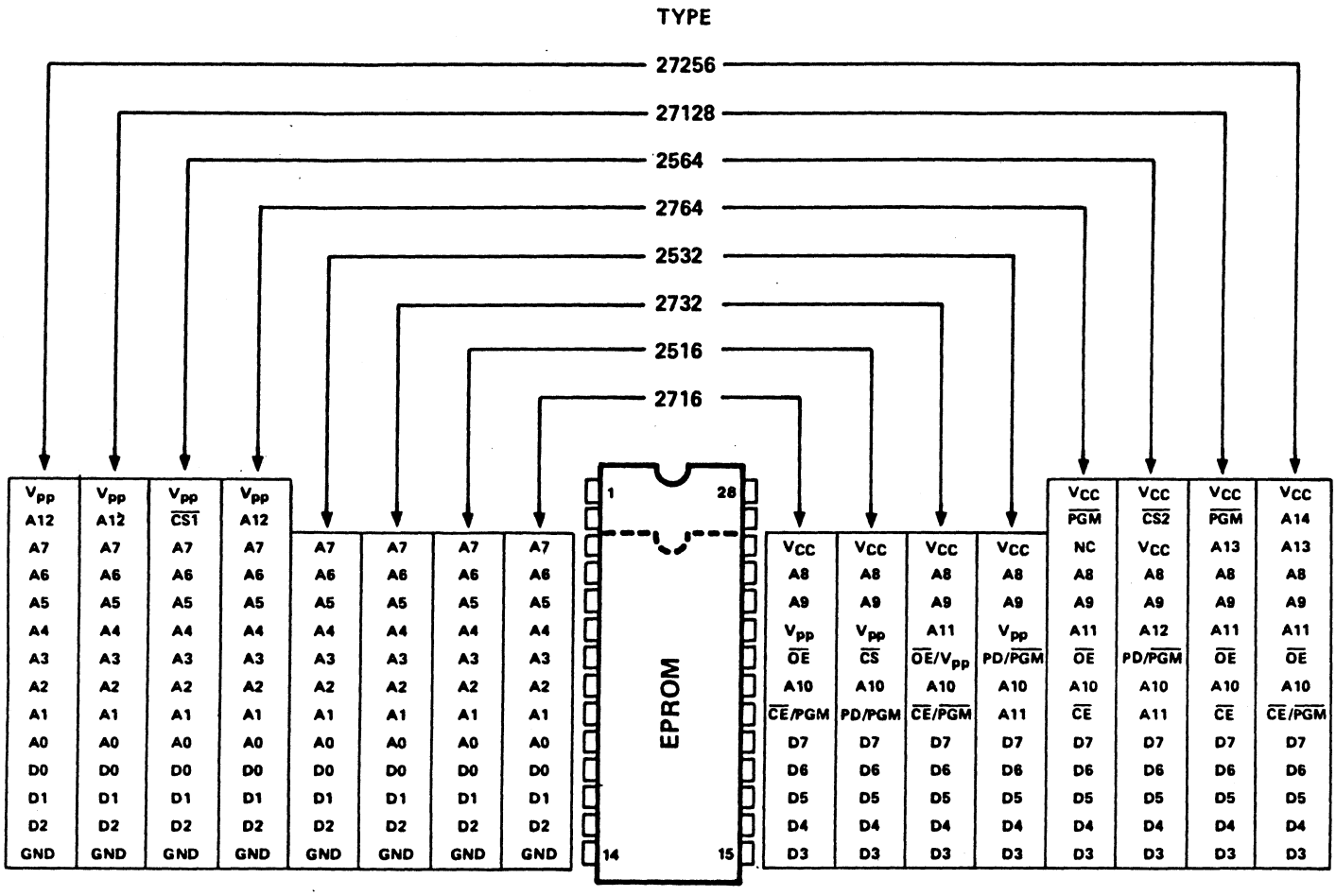
TMS 2564-45 JL

TMS 2564
28-PIN CERAMIC
DUAL-IN-LINE PACKAGE
(TOP VIEW)



†V_{CC} may be connected to pin 26 for 24-pin ROM compatibility.

PIN NOMENCLATURE	
A(N)	Address inputs
$\overline{CS}$(N)	Chip Selects
PD/$\overline{PGM}$	Power Down/Program
Q(N)	Input/Output
V_{CC}	+5 V Power Supply
V_{PP}	+25 V Power Supply
V_{SS}	0 V Ground



Tabel 2. De veranderlijke pen-aansluitingen met de benodigde signalen.

24p. voet ↓ | 28p. voet ↓

	2716		2732		2732A		2764		27128		2516		2532		2564	
	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM	RD	PGM
2 (X)	NC		NC		NC		A12		A12		NC		NC		$\overline{CS1}$	
	*	*	*	*	*	*	#	#	#	#	*	*	*	*	"0"	"0"
20 (18)	$\overline{CE}$/PGM		$\overline{CE}$/$\overline{PGM}$		$\overline{CE}$/$\overline{PGM}$		$\overline{CE}$		$\overline{CE}$		PD/PGM		A11		A11	
	"0"	⎍	"0"	⎍	"0"	⎍	"0"	"0"	"0"	"0"	"0"	⎍	#	#	#	#
22 (20)	$\overline{OE}$		$\overline{OE}$/V_{pp}		$\overline{OE}$/V_{pp}		$\overline{OE}$		$\overline{OE}$		$\overline{CS}$		PD/$\overline{PGM}$		PD/$\overline{PGM}$	
	"0"	"1"	"0"	+25	"0"	+21	"0"	"1"	"0"	"1"	"0"	"1"	"0"	⎍	"0"	⎍
23 (21)	V_{pp}		A11		A11		A11		A11		V_{pp}		V_{pp}		A12	
	"1"	+25	#	#	#	#	#	#	#	#	"1"	+25	"1"	+25	#	#
26 (24)	V_{CC}		V_{CC}		V_{CC}		N.C.		A13		V_{CC}		V_{CC}		V_{CC}*	
	+5	+5	+5	+5	+5	+5	*	*	#	#	+5	+5	+5	+5	+5	+5
27 (X)	NC		NC		NC		$\overline{PGM}$		$\overline{PGM}$		NC		NC		$\overline{CS2}$	
	*	*	*	*	*	*	"1"	⎍	"1"	⎍	*	*	*	*	"0"	"0"
V_{pp}	25 V		25 V		21 V		21 V		21 V		25 V		25 V		25 V	

* = don't care
# = dynamic input