

# MSX EPROM-mer

elektuur maart , april 1987

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

In twee delen beschrijven we een geavanceerde EPROM-programmer die speciaal werd ontwikkeld voor gebruik in combinatie met MSX-computers. Met deze EPROM-mer kunnen EPROM's van praktisch alle denkbare fabrikaten en met een capaciteit van 2 Kbyte (2716) tot en met 64 Kbyte (27512) gelezen, geprogrammeerd en gekopieerd worden.

MSX is een computerfamilie die in korte tijd aardig voet aan de grond heeft weten te krijgen. Je hoort nogal eens van gebruikers van andere typen computers dat MSX een typische "spelletjes-computer" zou zijn. Een dergelijke opmerking is te kwalificeren als een slag in de lucht, want het is niet de computer maar veeleer de gebruiker die de toepassing van zijn machine bepaalt. Dat die gebruiker nogal eens zwicht voor een van de, vaak kwalitatief uitstekende, spelletjes die voor MSX op de markt zijn, is hem nauwelijks te verwijten; in het vrije westen rust op plezier geen embargo. Dat een MSX-computer daarentegen ook voor heel serieuze dingen gebruikt kan worden, bewijst deze EPROM-programmer. En dan nog, hoe "serieus" is het programmeren van een EPROM? Ook spelprogramma's kunnen immers in EPROM gezet worden.

Ter zake nu: wat kan en wat doet de MSX-EPROM-mer en wat is er voor nodig? Over, de benodigheden kunnen we kort zijn. Allereerst de programmer zelf, waarvan de bouw in dit artikel beschreven wordt. Om hem aan te sluiten op een MSX-computer wordt gebruik gemaakt van de I/O-cartridge die in het januarinumnummer werd beschreven. Die moet u dus ook bouwen, maar die is voor meer toepassingen geschikt dan alleen het aansluiten van de EPROM-mer. Tot slot is ook een programma nodig dat in deel 2 volgende maand aan de orde komt. En zoals dat bij computers zo vaak is, is het het programma dat deze EPROM-mer zo krachtig maakt.

Fabrikant	-Type	Geheugen organisatie	Vpp	Programmeer-methode	Opmerking
AMD	AM9716	2Kx8	25 V	N	
	AM2716	2Kx8	25 V	N	
	AM2732	4Kx8	25 V	N	
	AM2732A	4Kx8	21 V	N	
	AM2764	8Kx8	21 V	N; I	
	AM2764A	8Kx8	12.5 V	I	
	AM27128	16Kx8	21 V	N; I	
	AM27128A	16Kx8	12.5 V	I	
	AM27256	32Kx8	12.5 V	I	
	AM27512	64Kx8	12.5 V	I	
	Fujitsu	MBM2716	2Kx8	25 V	N
MBM8516		2Kx8	25 V	N	
MBM2732A		4Kx8	21 V	N	
MBM27C32A		4Kx8	21 V	N	CMOS

	MBM2764	8Kx8	21 V	N; I	
	MBM27C64	8Kx8	21 V	N; I	CMOS
	MBM27128	16Kx8	21 V	N; I	
	MBM27256	32Kx8	12.5 V	I	
	MBM27C256	32Kx8	12.5 V	N; I	CMOS
	MBM27C512	64Kx8	12.5 V	I	CMOS
Hitachi	HN462716	2Kx8	25 V	N	
	HN462532	4Kx8	25 V	N	
	HN462732	4Kx8	25 V	N	
	HN462732A	4Kx8	21 V	N	
	HN482764	8Kx8	21 V	N; 1	
	HN27C64	8Kx8	21 V	N; 1	CMOS
	HN482764P	8Kx8	21 V	N; 1	OTP
	HN4827128	16Kx8	21 V	N; 1	
	HN27128P	16Kx8	21 V	N; 1	OTP
	HN27256	32Kx8	12.5 V	I	
	HN27512	64Kx8	12.5 V	I	
Intel	2716	2Kx8	25 V	N	
	2732A	4Kx8	21 V	N	
	P2732A	4Kx8	21 V	N	OTP
	2764,	8Kx8	21 V	N; 1	
	P2764	8Kx8	21 V	I	OTP
	2764A	8Kx8	12.5 V	I	
	27C64	8Kx8	12.5 V	I	CMOS
	P2764A	8Kx8	12.5 V	I	OTP
	27128	16Kx8	21 V	N; 1	
	27 128 A	16Kx8	12.5 V	I	
	P27218A	16Kx8	12.5 V	I	OTP
	27256	32K x 8	12.5 V	I	
	27C256	32Kx8	12.5 V,	I	CMOS
	87C256	32Kx8	12.5 V	I	CMOS
	27512	64Kx8	12.5 V	I	
	27513	4x16Kx8	12.5 V	I	Paged
Mitsubishi	M5L2716	2Kx8	25 V	N	
	M5L2732	4Kx8	25 V	N	
	M5L2764	8Kx8	21 V	N	
	M5L27128	16Kx8	21 V	N; 1	
	M5L27256	32Kx8	12.5 V	L	
Mostek	MK2716	2Kx8	25 V .-	N	
Motorola	MCM2716	2Kx8	25 V	N	
	MCM27L16	2Kx8	25 V	N	LP
	MCM2532	4Kx8	25 V	N	
	MCM25L32	4Kx8	25 V	N	LP
	MCM68764	8Kx8	25 V	M	
	MCM68766	8Kx8	25 V	M	
	MCM68769	8Kx8	25 V	M	
National	NMC2716	2Kx8	25 V	N	
Semiconductor	NMC27C16	2Kx8	25 V	N	CMOS
	IMMC27C16H	2Kx8	25 V	F2	CMOS
	NMC27C16B	2Kx8	12.5 V	I	CMOS
	NMC27C32	4Kx8	25, V	N	CMOS
	NMC27C32H	4Kx8	25 V	F2	CMOS
	NMC27C32B	4Kx8	12.5 V	I;F2	CMOS

	NMC27C64	8Kx8	12.5 V	I;F2	CMOS
	NMC27CP128	16Kx8	12.5 V	I;F2	CMOS
	NMC27C256	32Kx8	12.5 V	I;F2	CMOS
	NMC27C512	64Kx8	12.5 V	I;F2	CMOS
NEC	uPD2716	2Kx8	25 V	N	
	uPD2732	4Kx8	25 V	N	
	uPD2732C	4Kx8	25 V	N	OTP
	uPD2732A	4Kx8	21 V	N	
	uPD2764	8Kx8	21 V	N; 1	
	uPPD27C64	8Kx8	21 V	N; 1	CMOS
	uPD2764C	8Kx8	21 V	N; 1	OTP
	uPD27C64C	8Kx8	21 V	N; 1	CMOS
	uPD27128	16Kx8	21 V	N; 1	OTP
	uPD27218C	16Kx8	21 V	N: 1	OTP
	uPD27256	32Kx8	21 V	I	
	uPD27C256	32Kx8	21 V	I	CMOS
Rockwell	R87C32	4Kx8	21 V	N	CMOS
	R87C64	8Kx8	21 V	N	CMOS
	R27C64P	8Kx8	21 V	N	CMOS
SEEQ	2764	8Kx8	21 V	N; 1	
	5133	8Kx8	21 V	N; 1	
	27128	16Kx8	21 V	N; 1	
	5143	16Kx8	21 V	N; 1	
	27C256	32Kx8	12.5 V	L	CMOS
SGS/ATES	M2716	2Kx8	25 V	N	
	M2732A	4Kx8	21 V	N	
	M2764	8Kx8	21 V	N; 1	
Texas	TMS2516	2Kx8	25 V	N; F1	
Instruments	TMS2532	4Kx8	25 V	N; F1	
	TMS25L32	4Kx8	25 V	N	LP
	TMS2732	4Kx8	25 V	N	
	TMS2732A	4Kx8	21 V	N	
	TMS2564	8Kx8	25 V	N; F	
	TMS2764	8Kx8	21 V	N; 1	
	TMS27128	16Kx8	21 V	N; 1	
Thomson-	ET2716	2Kx8	25 V	N	
CSF	ETC2716	2Kx8	25 V	N	CMOS
	ETC2732	4Kx8	25 V	N	CMOS
	ET2764	8Kx8	21 V	N	
Toshiba	TMM323	2Kx8	25 V	N	
	TMM2732	4Kx8	25 V	N	
	TMM2764	8Kx8	21 V	N;I	
	TMM2764DI	8Kx8	21 V	N;I	
	TMM27128	16Kx8	21 V	N	
	TMM27256	32Kx8	21 V	I	
	TC57256	32Kx8	21 V	I	

*Tabel 1 Deze lijst bevat bijna alle op dit moment gangbare EPROM's met hun belangrijkste technische gegevens.*

De typenummers uit deze tabel kunnen eventueel gevolgd worden door een accesstime-indikatie.

Opname van een EPROM-type in deze tabel garandeert niet het programmeerbaar zijn met de MSX-EPROM-programmer.

Verklaring van de gebruikte afkortingen:

I = interactief programmeerbaar

N = normaal programmeerbaar (50 ms puls)

F1 = snel programmeerbaar (20 ms puls)

F2 = snel programmeerbaar (10 ms puls)

M = programmeerbaar volgens Motorola-protokol  
(niet mogelijk met deze programmer)

LP = low power type.

OTP = one time programmable

CMOS = complementary metal oxide type

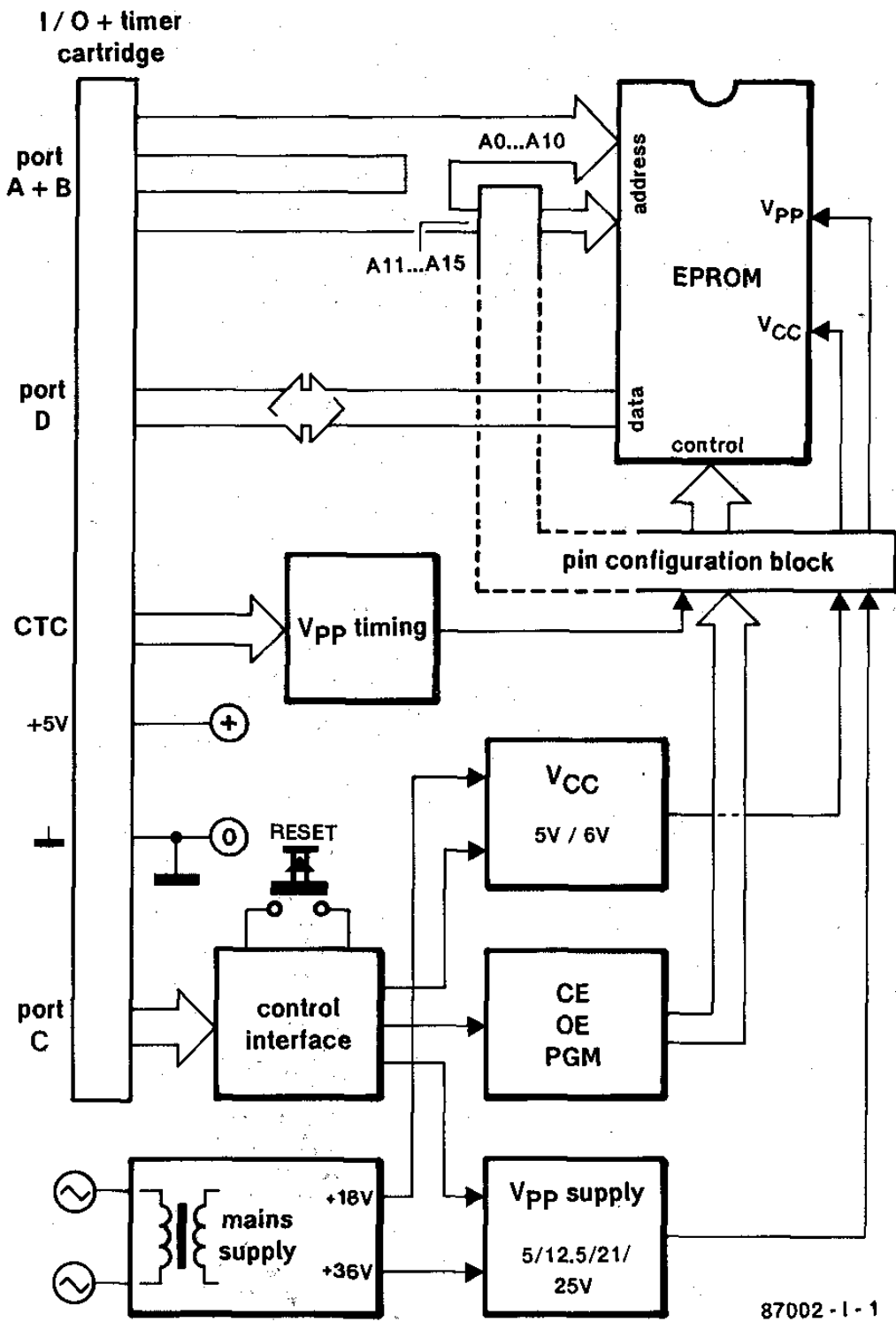
## **EPROM's**

Geheugen-IC's vormen voor fabrikanten de graadmeter van hun technologische expertise. De lay-out van dergelijke IC's is dankzij de regelmatige structuur relatief eenvoudig, zodat men zich kan concentreren op de eigenlijke IC-technologie.

Zo stookt men elkaar behoorlijk op in de schijnbaar oneindige reeks snellere geheugen-IC's met telkens weer een grotere capaciteit. Hoe interessant die vooruitgang in de technologie ook mag zijn, voor de gebruiker tellen de meer praktische kanten:

1. betaalbaarheid
2. een enkele 5-V-voedings-spanning
3. een penbezetting die aansluit op die van de voorgangers

Een breed scala aan EPROM's is tegenwoordig verkrijgbaar. In tabel 1 doen we een poging een bij benadering compleet overzicht te geven van de meest gangbare types op deze woelige markt. Behalve het verschil in geheugenkapaciteit is het voornamelijk de programmeerspanning waarmee verschillende typen EPROM's zich onderscheiden. Ook is te zien dat CMOS-technologie met de ermee samenhangende reductie in dissipatie algemeen ingang begint te krijgen. Mede genoodzaakt door de steeds grotere geheugenkapaciteit gaan steeds meer fabrikanten over tot interactieve programmeeralgoritmes. Het oude, niet interactieve algoritme werkt met programmeerpulsen met een vaste lengte van 50 ms. Het programmeren van grotere geheugens duurt met een dergelijk algoritme natuurlijk evenredig langer. Bij in-teractief programmeren, mogelijk bij de types waar een I bij staat in de tabel, wordt herhaald met 1-ms-pulsen geprogrammeerd. Na elke puls wordt gekeken of de data inderdaad korrekt geprogrammeerd is. Is dat het geval, dan vindt zekerheidshalve nog een "overprogrammering" plaats waarna het volgende byte aan de beurt is. Dankzij dit "intelligente" algoritme vindt het programmeren sneller plaats. De software voor de MSX-EPROMmer voorziet uiteraard ook in deze wijze van programmeren. In deel 2 wordt daar dieper op ingegaan.



87002 - 1 - 1

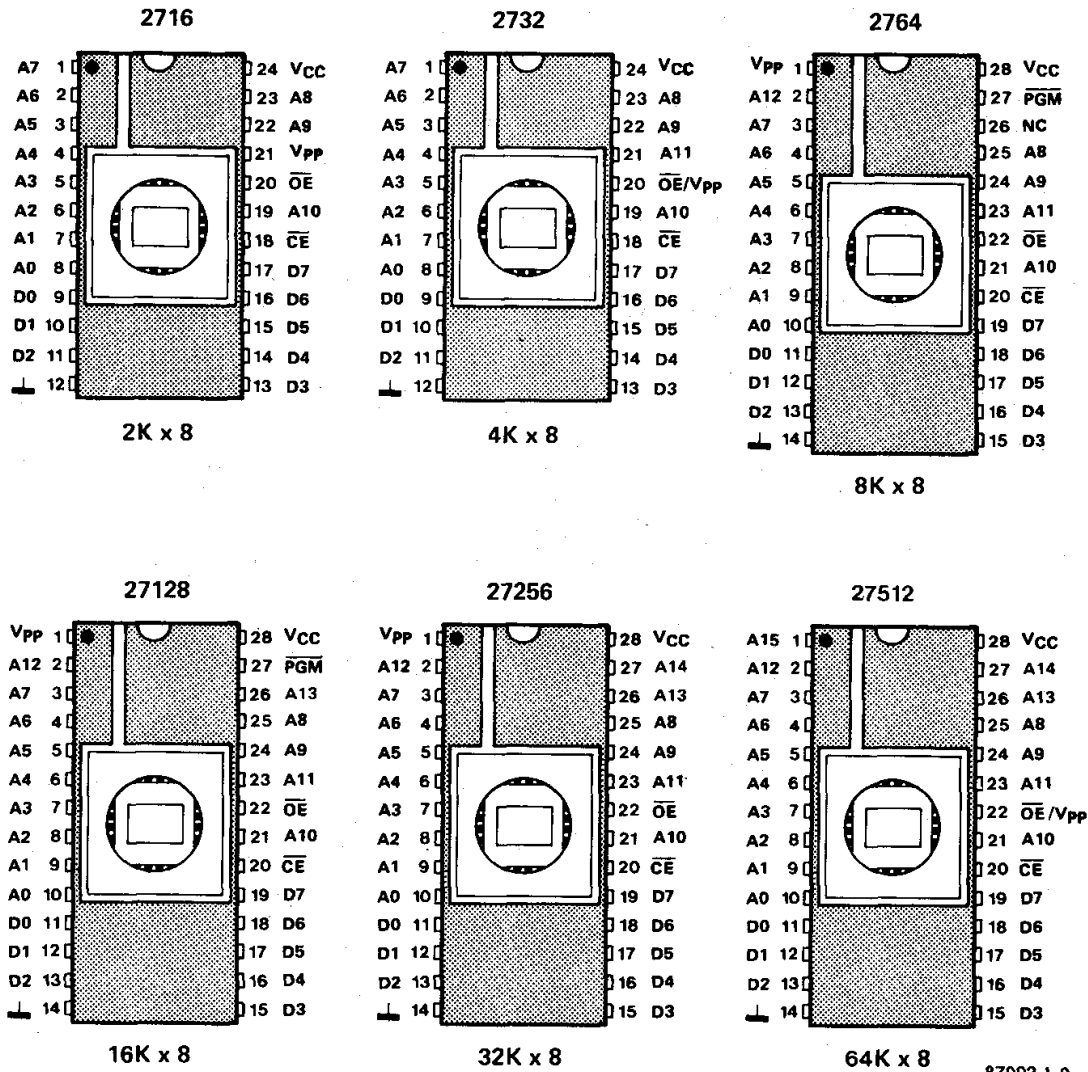
Figuur 1. Blok-schema van de MSX-EPROM-programmer.

## Blokschema

De functionele opzet van de EPROM-programmer is het eenvoudigst te verduidelijken met behulp van het blokschema (figuur 1). Twee outputpoorten, A en B, van de I/O-cartridge leveren de adressen aan de te programmeren EPROM. Een derde poort (D) wordt gebruikt om data te lezen of te schrijven. Poort C tenslotte zorgt voor de besturing van de programmer. Met de juiste bitcombinatie kan via poort C de EPROM-voedingsspanning ingesteld worden op 5 V of 6 V en de programmeerspanning op 5, 12,5, 21 of 25 V. Ook worden met poort C de OE(output enable)- en CE(chip enable)-aansluitingen van de EPROM bediend in afhankelijkheid van de READ- VERIFY- of PROGRAM-mode van de programmer.

De CTC (counter & timer controller) van de I/O-cartridge bestuurt een set/reset-flipflop waarmee de lengte van de programmeerpulsen bepaald wordt.

Figuur 2 geeft een overzicht van de aansluitingen van EPROM's uit de 27XXX-reeks. Die aansluitingen zijn voor het grootste deel hetzelfde als we pen 1 van de 24-pens-types samen laten vallen met pen 3 van de 28-pens-types. De 25XX EPROM's van Texas Instruments wijken echter weer wat af. Een tweerijige contactstrip met kortsluitstekers (jumpers) wordt gebruikt om alle signalen op de juiste pootjes van elk type EPROM te krijgen. Het logika-gedeelte van de schakeling wordt gevoed vanuit de 5-V-voeding van de computer. De programmeerspanning  $V_{pp}$  en de voedingsspanning  $V_{cc}$  voor de te programmeren EPROM zijn afkomstig van de voeding in de EPROM-mer.



87002-1-2

Figuur 2. Portret van een nog altijd groeiende familie EPROM's: de pen aansluitingen van 2716 tot en met 27512

## De schakeling

Zoals te zien is in figuur 3, is de schakeling van de programmer niet overdreven complex. De adressen van de te programmeren lokaties worden geleverd door de poorten A en B van IC1 op de cartridge-print. Poort A levert het minst significante adresbyte (A0..A7) en poort B het meest significante byte (A8...A15). Voor de "kleinste" te programmeren EPROM (2716 of TI 2516) worden alleen de lijnen A0..A10 gebruikt. Omdat deze aansluitingen voor elk type EPROM hetzelfde zijn, gaan ze rechtstreeks naar de EPROM-voet. De overige adreslijnen zijn aangesloten op het jumperblok K2, alwaar voor het programmeren van grotere EPROM's de nodige doorverbindingen met kortsluitstekertjes gemaakt kunnen worden. Zowel in de adres-als de datalijnen zijn stroombegrenzende serieweerstanden opgenomen om te voorkomen dat defekte EPROM's de PIO's beschadigen.

De besturing van de programmer gaat via poort C (IC2, poort A) van de cartridge-print. Afhankelijk van het type te programmeren EPROM wordt een bepaald bitpatroon door poort C geleverd. Bit 0 en 1 dienen om een van de vier mogelijke programmeerspanningen in te stellen: 5 V, 12,5 V, 21 V of 25 V. Een twee naar vier dekoder (IC4) vertaalt deze twee bits naar een actief laag-nivo op een van de vier Y-uitgangen. Die vier uitgangen zijn verbonden met open-kollektor-buffers waarmee een van de vier mogelijke spanningsdelers geactiveerd wordt. Die spanningsdeler, gevormd door enerzijds R8 en anderzijds de weerstanden R24 en R34, R25 en R35, R26 en R36 of R22 en R37, is aangesloten op de referentie-ingang van IC7. Met IC7 wordt de aldus programmeerbare programmeerspanning opgewekt. Via vier LED's (D6..D9) wordt aangegeven welke van de vier programmeerspanningen ingesteld is. Met een extra stuurlijntje (A2, poort C) kan de programmeerspanning aan of uit gezet worden. Ook deze status wordt met een LED (D5) voor de gebruiker geïndiceerd. Omdat bij sommige van de nieuwere typen EPROM's gedurende het interaktief programmeren een verhoging van de voedingsspanning van 5 V naar 6 V wordt voorgeschreven, is voor die spanning een gelijksoortig omschakelmechanisme aanwezig. Bij een logisch nul-nivo op A3 van poort C worden, via N5 en N6, R31 en R32 parallel geschakeld aan R33 en R11. Hierdoor zal de uitgangsspanning van IC6 verhoogd worden van 5 V naar 6 V. Ook hier weer een LED (D3 of D4) die de ingestelde voedingsspanning aangeeft en een extra besturingslijn (A4, poort C) om de voedingsspanning uit te kunnen schakelen. Zo kan de EPROM-socket spanningsloos gemaakt worden voor het veilig plaatsen en verwijderen van EPROM's. A5 van poort C bepaalt het logische nivo op de OE-ingang van de EPROM. Deze ingang moet laag gemaakt worden om data te kunnen lezen. Twee LED's (D12, D13) geven de richting van het datatransport aan, van computer naar EPROM of vice versa.

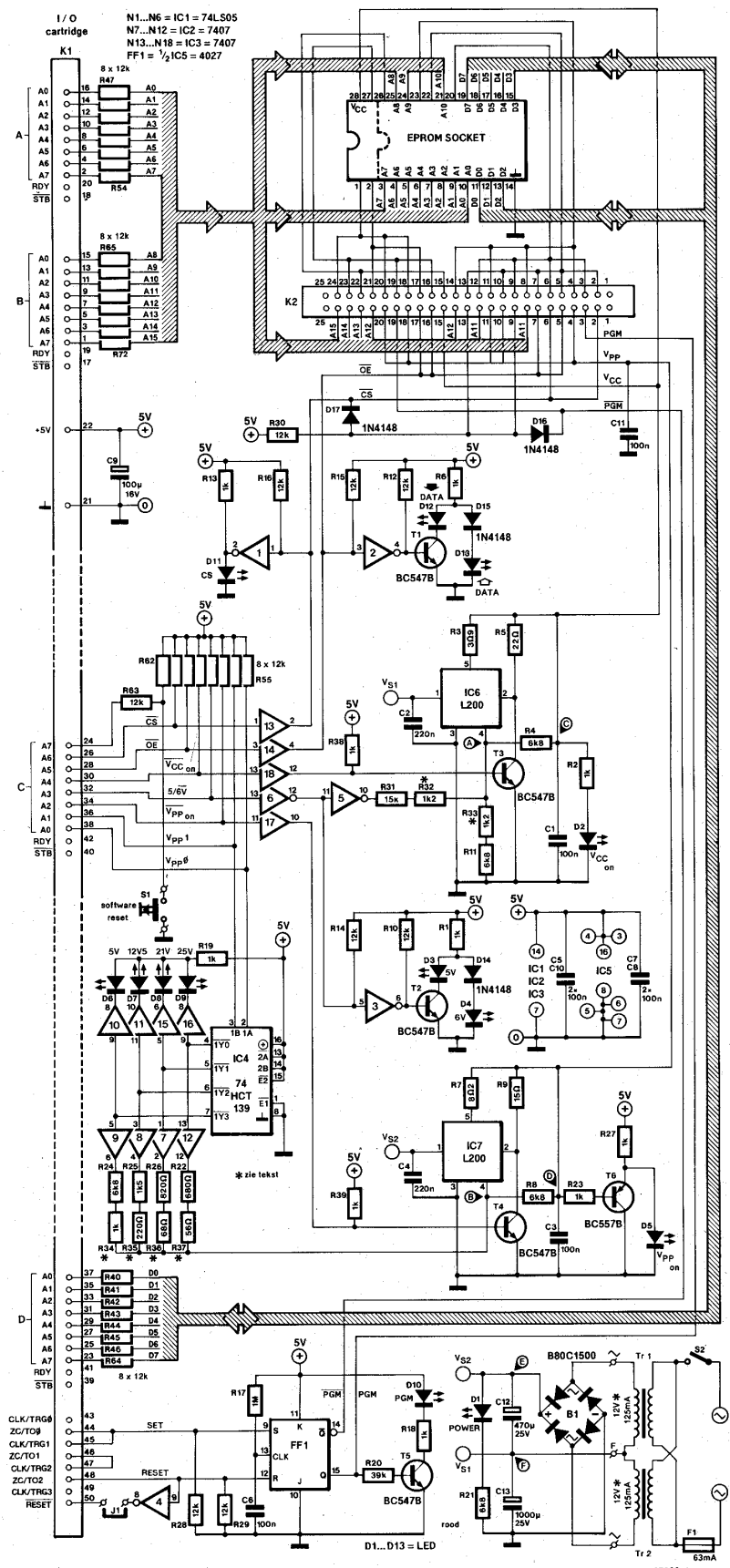
A6 van poort C heeft de supervisie over de CS-lijn (chip select) van de EPROM. De dioden D16 en D17 en pull-up-weerstand R30 vormen een AND-poort waarmee gezorgd wordt voor een juiste aansturing van de gekombineerde CE/PGM-aansluiting die de EPROM-types 2532, 2564, 2732, 27256 en 27512 hebben. Samen met OE, Vpp, A11..A15 en PGM/PGM is de CS-lijn aangesloten op de tweerijige kon-taktenstrip K2. Op K2 kan er door middel van kortsluitstekertjes voor gezorgd worden dat de afwijkende



EPR0M-aansluitingen de juiste signalen toegevoerd krijgen.

A7 van poort C is de enige besturingslijn die, vanuit de computer gezien, als input is geschakeld. Vanuit het programma in de MSX-computer wordt het logische nivo op deze aansluiting in de gaten gehouden. Als dit "0" is (schakelaar S1 moet daarvoor ingedrukt worden) gaat de computer terug naar het hoofdmenu van het EPROM-programma. Wordt daarna op CTRL STOP gedrukt, dan komt men terug in de MSX-BASIC-interpreter. Alle besturingslijnen van de EPROM-mer (poort C) zijn voorzien van pull-up-weerstanden om na het inschakelen verzekerd te zijn van een juiste en gedefinieerde bitconfiguratie. Zowel de programmeerspanning als de voedingsspanning worden opgewekt door middel van de bekende L200 spanningsregelaar. Het uitschakelen van de door deze IC's geleverde spanningen geschiedt door via de lijnen A2 en A4 de current-sensing-ingangen met respectievelijk T3 en T4 naar massa te schakelen. Daarmee worden de interne uitgangstrappen geblokkeerd. Het voordeel van deze wijze van besturen is dat absoluut geen overshoot ontstaat op de Vcc en Vpp-aansluitingen. In figuur 4 is te zien hoe de programmeerspanning (uitgangsspanning IC7) zich gedraagt bij het beurtelings inschakelen op 5 V, 12,5 V, 21 V en 25 V. De VPP- en Vcc-voedingen zijn beide kortsluitvast, waarbij de stroom-begrenzing op respectievelijk 50 mA en 100 mA is vastgelegd, bepaald door R7 (IC7, VPP) en R3 (IC6, Vcc). Dankzij de Vcc- en VPP-aan/uit-indicatie kunnen defekte EPROM's eenvoudig gesignaleerd worden. De 5-V-voeding voor de programmer wordt aan de MSX-computer zelf onttrokken via pen 21 en 22 van KI op de cartridge-print. Dat betekent dat de computer zowel de cartridge als de EPROM-mer van stroom voorziet. Zoals ook al in het artikel van de cartridge werd vermeld, dient men zich wel bewust te zijn van de maximale reserve van de ingebouwde voeding om overbelasting te voorkomen. Een standaard MSX-computer heeft per insteeksleuf een stroomreserve van 300 mA. De programmer en de cartridge samen hebben een verbruik dat zal liggen tussen 100 en 250 mA. Het kan echter geen kwaad dit verbruik met behulp van een aparte 5-V-voeding te controleren, alvorens het geheel op de computer aan te sluiten. De duur van de programmeer-pulsen wordt gestuurd met een set-reset-flipflop (FF1). Met timer 0 van de CTC op de cartridge-print wordt een 4 us vertraging gekreeerd, waardoor men ervan verzekerd is dat data en adres voor de te programmeren EPROM stabiel zijn alvorens de PGM/PGM-lijn geactiveerd wordt. De programmeerpuls start door met het zero count signaal van T0 de flipflop te setten. Ook wordt de zero-count-uitgang van T0 gebruikt om de tweede timer in de CTC te activeren. Deze timer, die de systeemklok deelt door 7 x 256, genereert uitgangspulsen met een periodetijd van 0,5 ms. Een derde timer in de CTC (er zijn er overigens vier beschikbaar) wordt in de counter-mode gebruikt en telt het aantal 0,5-ms-pulsen. Op kommando van zijn count zero wordt de flipflop weer gereset, aldus de lengte van de programmeerpuls bepalend. Gedurende het aanwezig zijn van de programmeerpuls licht D10 ter controle op. In de softwarebeschrijving volgende maand wordt dieper ingegaan op het gebruik van de CTC. Ook de benodigde besturingwoorden voor verschillende typen EPROM's worden in dat artikel gegeven. De jumperplaatsing voor die verschillende EPROM's is gegeven in tabel 2 en op de frontplaat (figuur 6).

De EPROM-programmer heeft een ingebouwde konventionele voeding die twee gelijkgerichte en afgevlakte spanningen levert voor de spanningsregelaars. Omdat deze spanning niet boven het voor de L200 toegestane maximum uit mag komen (40 V), dient u bij het testen de spanning op punt E te controleren.



*Figuur 3. Het schema van de programmer. De schakeling moet aangesloten worden op de I/O- en timer-cartridge die in het januarinumnummer werd beschreven.*

pos.	Signal	2716	2516	2732	2532	2764	2564	27128	27256	27512
1	n.c.									
2	CE		CE							
3	PGM	CE/ PGM	PD/ PGM							
4	VPP	VPP	VPP							
5	ÖE	ÖE								
6	CE					CE		CE		
7	ÖE					ÖE		ÖE	ÖE	
8	A11			A11		A11		A11	A11	A11
9	CE/ PGM			CS/ PGM					CE/ PGM	CS/ PGM
10	VPP			ÖE/ Vpp						ÖE/ Vpp
11	VPP				VPP					
12	A11				A11		A11			
13	CE/ PGM				PD/ PGM		PD/ PGM			
14	A12						A12			
15	Vcc	Vcc	Vcc	Vcc	Vcc		Vcc			
16	ÖE	n.c.	n.c.	n.c.	n.c.		S1			
17	OE	n.c.	n.c.	n.c.	n.c.		S2			
18	VPP	n.c.	n.c.	n.c.	n.c.		VPP			
19	PGM					PGM		PGM		
20	VPP					VPP	^	VPP	VpP	
21	A12					A12		A12	A12	A12
22	A13					n. c.		An	A13	A13
23	A14								A14	A14
24	A15									A15
25	n.c.									

*Tabel 2. Met twee samengestelde jumperblokken, een van drie en een van vier jumpers, kunnen op K2 voor elk type EPROM de juiste doorverbindingen worden gemaakt, die de programmeerspanning levert (IC7); vrij van overshoot en andere onhebbelijkheden.*

## Bouwen maar

De EPROM-programmer kan opgebouwd worden op een zeer compacte print die ook in de EPS opgenomen is. Die compactheid was mede mogelijk door een dubbelzijdig ontwerp te maken. De doorgemetalliseerde gaten maken een snelle en betrouwbare opbouw mogelijk. Om ruimte te sparen worden alle weerstanden, op R31.. .R37 na, vertikaal gemonteerd. De spanningsregelaars, IC6 en IC7, hebben geen extra koellichaam nodig. Enkele componenten moeten aan de sporenzijde gemonteerd worden (whats in a name, een dubbelzijdige print heeft natuurlijk aan beide zijden sporen, maar we bedoelen de kant zonder componentenopdruk). Die componenten zijn gestreept aangegeven; het zijn de LED's, de kontaktenstrip (K2) en de EPROM-voet. Als eerste kunt u echter het beste D15.. .D17 en R30 monteren. Daarna soldeert u de EPROM-voet en de LED's aan de andere kant van de print. Let erop dat, indien u gebruik maakt van het kastje uit de stuklijst, de LED's en de EPROM-voet voldoende ver uitsteken, zodat ze boven de bovenkant van het kastje uitkomen. Een Textool ZIF-voet (zero insertion force) heeft (te) korte en platte aan-sluitpennen. Een dergelijke voet kan daarom het beste op twee enkelrijige 14-pens contactstrips gesoldeerd worden. Voeten van een ander merk kunnen wel normale soldeeraansluitingen hebben, zodat ze boven op een 28-pens wirewrap-voet geplaatst kunnen worden.

Alle rechtstreeks met het licht-net verbonden onderdelen worden niet op de print gemonteerd. Het gaat hier om S2, Tr1 en Tr2 en F1, en ze moeten met inachtneming van de veiligheidsvoorschriften voor netspanning-voerende onderdelen in het kastje gemonteerd worden. Aan een kant van de print blijft in het kastje wat ruimte voor deze onderdelen over. De enige liggende weerstanden, R31.. .R37, kunnen het beste in een 14-pens IC-voetje gemonteerd worden, zodat latere aanpassingen om de uitgangsspanning van de spanningsregelaars bij te schaven eenvoudig uitgevoerd kunnen worden (zie figuur 7). Zoals u wellicht al gedestilleerd hebt uit tabel 2, moeten de jumpers op K2 altijd in twee blokken van drie en vier geplaatst worden. Door twee "jumper-blokken" samen te stellen door respectievelijk drie en vier jumpers aan elkaar te lijmen, wordt het erg gemakkelijk om bij elke type EPROM de juiste jumpers op K2 te plaatsen. De frontplaat kan u daarbij ook van dienst zijn. Die frontplaat, zoals afgebeeld in figuur 6, is trouwens net datgene dat de EPROM-mer een waardig gedistingeerd voorkomen geeft.

De verbinding tussen EPROM-mer en cartridge-print komt tot stand via een 50-polige flatcable. Aan de zijkant van de kast moet een uitsparing gemaakt worden om K1 het zicht op de vrije wereld te gunnen.

## Onderdelenlijst

### Weerstanden:

R1,R2,R6,R13,R18,R19, R23,R27,R38,R39 = 1k

R3 = 3ohm9

R4,R8,R11,R21,R24 -6k8

R5 - 22 ohm

R7 = 8ohm2 .

R9 - 15 ohm

R10,R12,R14...R16, R28...R30,R40...R72 = 12k

R17 - 1M

R20 - 39 k

R22 - 680 ohm

R25 - 1k5

R26 - 820 ohm

R31 = 15 k

R32,R33 =1k2\*

R34 =1k\*

R35 - 220 ohm\*

R36 = 68 ohm\*

R37 = 56 ohm\*

### Kondensatoren:

C1,C3,C5...C8,C10,C11= 100 n

C2,C4 = 220 n

C9 = 100 uF/16 V radiaal

C12 = 470 uF/25 Vaxiaal

C13 = 1000 uF/25 V axiaal

### Halfgeleiders:

B1 = B80C1500

D1...D13 = LED, rood

D14...D17 = 1N4148

IC1 = 74LS05

IC2,IC3 - 7407

IC4 = 74HCT139

IC5 = 4027

IC6,IC7 = L200

T1...T5 = BC547B

T6 = BC 557B

### Diversen:

F1 = zekering, 63 mA traag plus houder

K1 = 50-polige haakse konnektor, male, voor printmontage, met twee rijen kontakten in 0,1" raster

K2 = 50-polige rechte konnektor, male, voor printmontage, met twee rijen kontakten in 0,1" raster

twee kontaktstiftjes voor J1

7 jumpers voor K2 plus een hulp-jumper (J1)

S1 = drukschakelaar, 1 x maakkontakt

S2 = netschakelaar

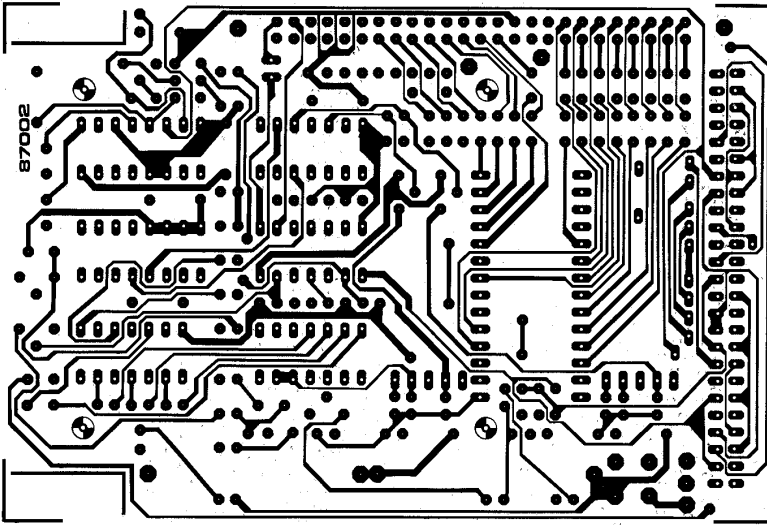
Tr1,Tr2 = 12 V (10 V) 1,2 VA nettrafo  
behuizing, bijvoorbeeld OKW 9409111

28-pens ZIF-IC-voet (bijv. Textool 28)

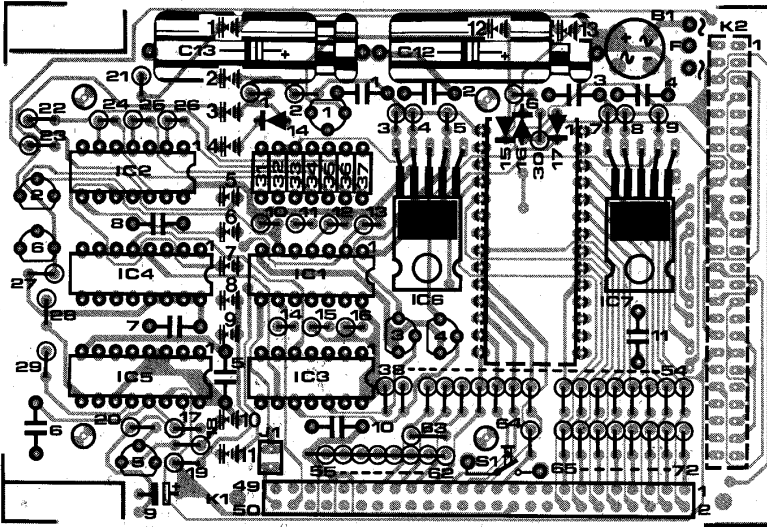
\* zie tekst

Geschatte bouwkosten: circa f 150,-

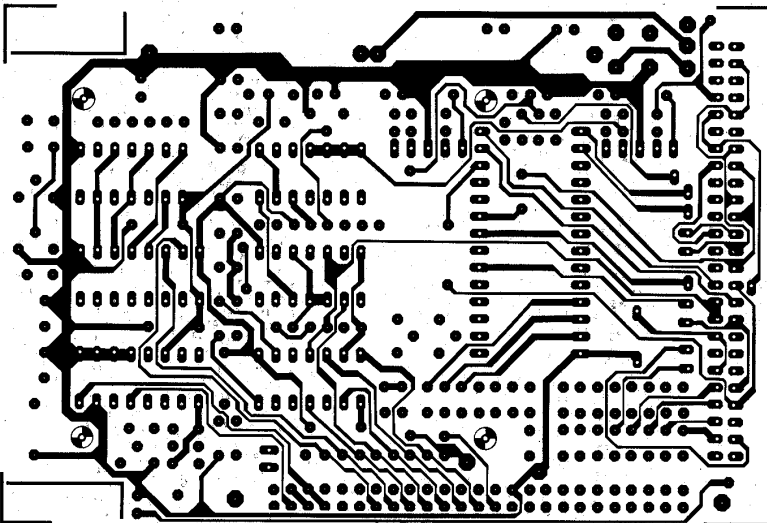
soldeerzijde



komponentenopstelling



komponentenzijde



*Figuur 5. Print-layout en componentenopstelling van de dubbelzijdige print. De gestreept getekende onderdelen worden aan de andere kant van de print gemonteerd.*

```

10 '***** Testprogramma Eprom Programmer *****
20 '
30 CLS:DEFINT A-Z:F=3*16
40 DA=F+4:DB=F+5:DC=F+8:DD=F+9
50 CA=F+6:CB=F+7:CC=F+10:CD=F+11
60 T0=F+12:T1=F+13:T2=F+14:T3=F+15
70 OUT CA,255:OUT CA,0:OUT CA,7:OUT CA,3
80 OUT CB,255:OUT CB,0:OUT CB,7:OUT CB,3
90 OUT CD,255:OUT CD,0:OUT CD,7:OUT CD,3
100 OUT CC,255:OUT CC,128:OUTCC,7:OUT CC,3
110 OUT DA,0:OUT DB,0:OUT DC,255:OUT DD,0
120 OUT T0,3:OUT T1,3:OUT T2,3:OUT T3,3
130 OUT T0,3:OUT T1,3:OUT T2,3:OUT T3,3
140 ON KEY GOSUB 280,340,380,530,570,610,640,680,720,760
150 FOR I=1 TO 10:KEY(I)ON:NEXT
160 ON STOP GOSUB 250:STOP ON
170 A=0:B=0:C=127:D=0:GOSUB 800:GOSUB 810:GOSUB820
180 '
190 'WAITLOOP FUNKTIONKEY *****
200 OUT DA,A:OUT DB,B:OUT DC,C:OUT DD,D
210 IF INP(DC)<128 THEN 250
220 GOTO 200
230 '
240 'HANDLING STOP-BUTTON *****
250 STOP OFF:GOSUB 600:OUT DC,127:END
260 '
270 'ROTATE ADRESSES *****
280 KEY(1)OFF:IF (A=0 AND B=0) THEN A=1:GOTO 310
290 IF A=128 THEN A=0:B=1:GOTO 310 ELSE A=A*2
300 IF B=128 THEN B=0:A=1 ELSE B=B*2
310 GOSUB 800:KEY(1)ON:RETURN
320 '
330 'ROTATE DATALINES *****
340 KEY(2)OFF:IF (D=0 OR D=128) THEN D=1 ELSE D=D*2
350 GOSUB 810:KEY(2)ON:RETURN
360 '
370 '100 mSek. PULS *****
380 KEY(3)OFF:LOCATE 0,9:PRINT"100 mSek. puls"
390 C=C AND 11:C=C OR 160:OUTDC,C:GOSUB 820
400 C=C AND 127:C=C AND 127:C=C OR 84
410 OUT T0,&H5 '***** Prescaler = 16, timermode, continued operation,
start when writing timeconstant (Zie regel 460)
420 OUT T1,&HD '***** Prescaler = 16, timermode, continued operation
after started by output of timer 0
430 OUT T2,&H45 '***** Countermode, triggered by output of timer 1
440 OUT T1,112 '***** Preset counter (outputpulse each 16x112
clockcycles)
450 OUT T2,200 '***** Preset for downcounter (200x0.5 miliseconds)
460 OUT T0,1 '***** Timer 0 starts timer 1 after 16 clockcycles
470 OUT T0,3 '***** Reset starter (timer 1 has been started)
480 IF INP(T2)<100 THEN490ELSE480 '***** Check if countdown half done
490 IF INP(T2)>100 THENOUTT2,3:OUTDC,C:OUTT1,3 ELSE490 '***** Check
time-out

```



```

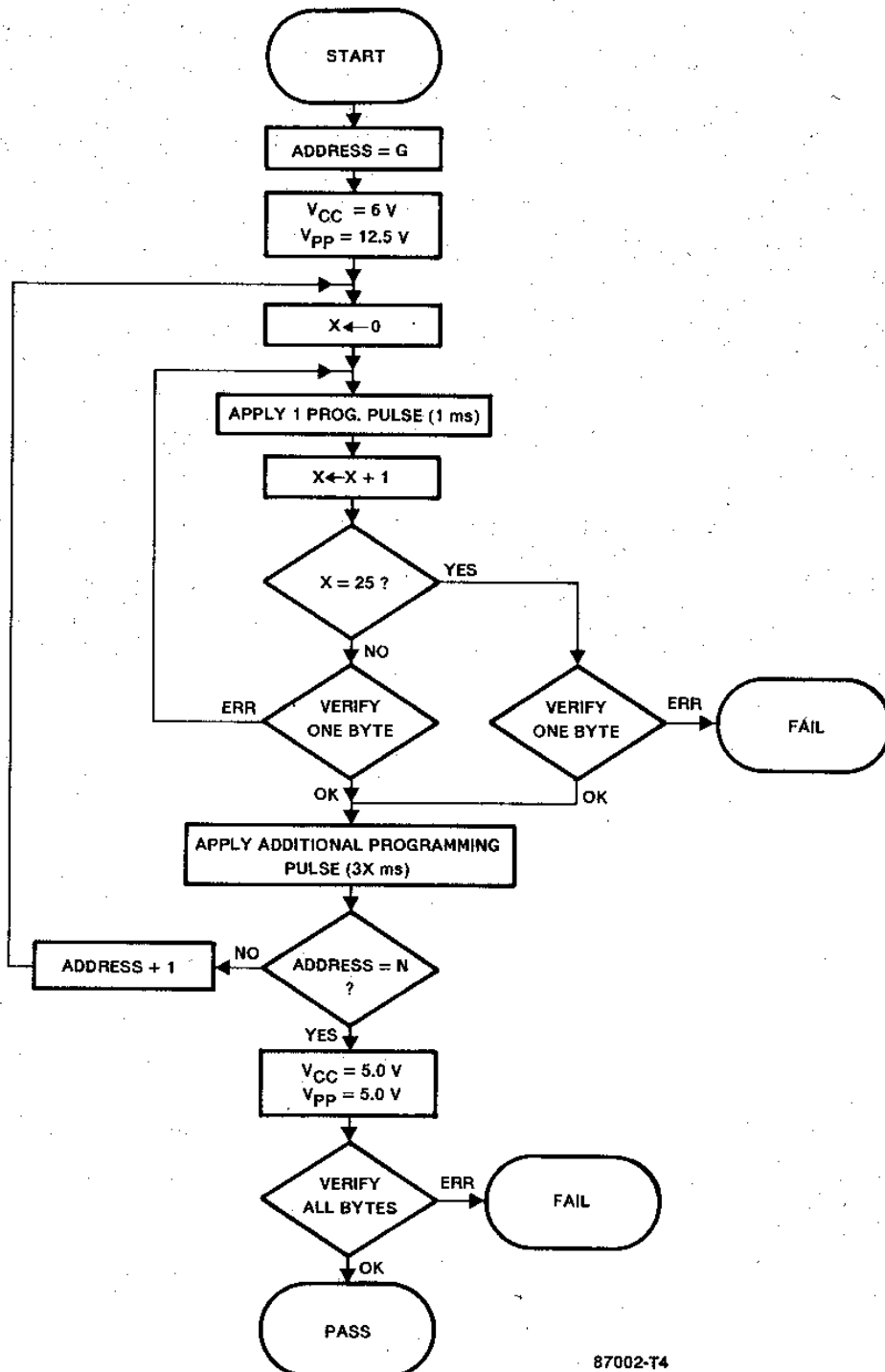
500 FOR I=1 TO 250:NEXT:LOCATE 0,9:PRINT CHR$(27);"K":GOSUB
820:KEY(3)ON:RETURN
510 '
520 'Vcc CHANGE *****
530 KEY(4)OFF:C=C AND 8:C=C+8:C=C AND 8
540 C=(INP(DC) AND 119) OR C:GOSUB 820:KEY(4)ON:RETURN
550 '
560 'Vpp CHANGE *****
570 KEY(5)OFF:C=C AND 3:C=C-1:C=C AND 3
580 C=(INP(DC) AND 124) OR C:GOSUB 820:KEY(5)ON:RETURN
590 '
600 'RESET *****
610 KEY(6)OFF:A=0:B=0:C=127:D=0:GOSUB 800:GOSUB 810:GOSUB
820:KEY(6)ON:RETURN
620 '
630 'CHIP ENABLE *****
640 KEY(7)OFF:C=C AND 64:C=C+64:C=C AND 64
650 C=(INP(DC) AND 63) OR C:GOSUB820:KEY(7)ON:RETURN
660 '
670 'OUTPUT ENABLE *****
680 KEY(8)OFF:C=C AND 32:C=C+32:C=C AND 32
690 C=(INP(DC) AND 95) OR C:GOSUB 820:KEY(8)ON:RETURN
700 '
710 'Vcc ON/OFF *****
720 KEY(9)OFF:C=C AND 16:C=C+16:C=C AND 16
730 C=(INP(DC) AND 111) OR C:GOSUB 820:KEY(9)ON:RETURN
740 '
750 'Vpp ON/OFF *****
760 KEY(10)OFF:C=C AND 4:C=C+4:C=C AND 4
770 C=(INP(DC) AND 123) OR C:GOSUB 820:KEY(10)ON:RETURN
780 '
790 'SUNROUTINES SCREEN *****
800 LOCATE 0,0:PRINT RIGHT$("0000000"+BIN$(B),8);"
";RIGHT$("0000000"+BIN$(A),8);" Adreslines":RETURN
810 LOCATE 0,3:PRINT RIGHT$("0000000"+BIN$(D),8);" Datalines":RETURN
820 LOCATE 0,6:PRINT RIGHT$("0000000"+BIN$(C),8);" Output channel
C":RETURN

```

*Tabel 3. Met dit testprogramma kan de werking van de EPROM-meet met behulp van de MSX-functie-toetsen stap voor stap getest worden.*

Tabel 4

interaktief programmeren



87002-T4

Tabel 4. Intelligent (of interaktief) programmeren berust in essentie op het teruglezen van data tijdens het programmeren. Wordt een databyte korrekt teruggelezen, dan wordt tzekeerheidshalve een soort overprogrammering uitgevoerd.

## Test en in-bedrijfstelling

Dankzij de LED's die de status van de programmer aangeven, kan het testen van de hardware, programmer plus I/O-cartridge, grotendeels via de software uitgevoerd worden. De eerste test is echter een pure hardware-aangelegenheid. Schakel de EPROM-mer in en meet de afgevlakte gelijkspanning op punt E. Deze moet, in onbelaste toestand, minder dan 40 V bedragen. Meer verdraagt een L200 niet. Bij bepaalde nettrafo's kan het zijn dat de spanning in onbelaste toestand zoveel hoger is dan in belaste toestand, dat u boven die 40 V uitkomt. In dat geval zult u andere trafo's moeten gebruiken. Met twee 10-V-trafo's zit u altijd veilig, maar helaas zijn die wat moeilijker verkrijgbaar. Meet u op punt E een veel lagere spanning, dan heeft u de trafo's per ongeluk in anti-serie geschakeld. Pool er dan een om, primair of secundair, dat maakt niet uit.

Plaats de I/O- en timer-cartridge in de daarvoor bestemde aansluiting van de MSX-computer (het "slot" zoals dat in computerjargon heet, oftewel sleuf of gleuf voor mensen die het in Nederlands willen zeggen) en sluit de EPROM-mer aan. Bij het inschakelen moet de computer met het normale openingsplaatje opstarten. Met het testprogramma (tabel 3) kunt u door gebruik te maken van de tien funktietoetsen en de status van de EPROM-mer af te lezen, aan de LED's de werking van de schakeling controleren. Direct na inschakelen moeten de volgende LED's oplichten:

D1                =power  
D3, Vcc         = 5 V  
D6, VpP         = 5 V  
D13,    DATA IN

Als het testprogramma de besturing krijgt, hebben de funktietoetsen het volgende resultaat:

F1: maakt de adreslijnen een voor een "hoog"  
F2: maakt de datalijnen een voor een "hoog"  
F3: veroorzaakt een 50 ms PGM/PGM-puls  
(plaats tijdelijk jumper J1 aan de uitgang van N4)  
F4: omschakelen tussen Vcc =5 V en Vcc = 6 V  
F5: maakt VPP achtereenvolgens 5 V, 12,5 V, 21 V en 25 V  
F6: terug naar beginsituatie  
F7: activeert CE  
F8: omschakelen richting datatransport(OE)  
F9: in/uitschakelen van Vcc  
F10: in/uitschakelen VPP

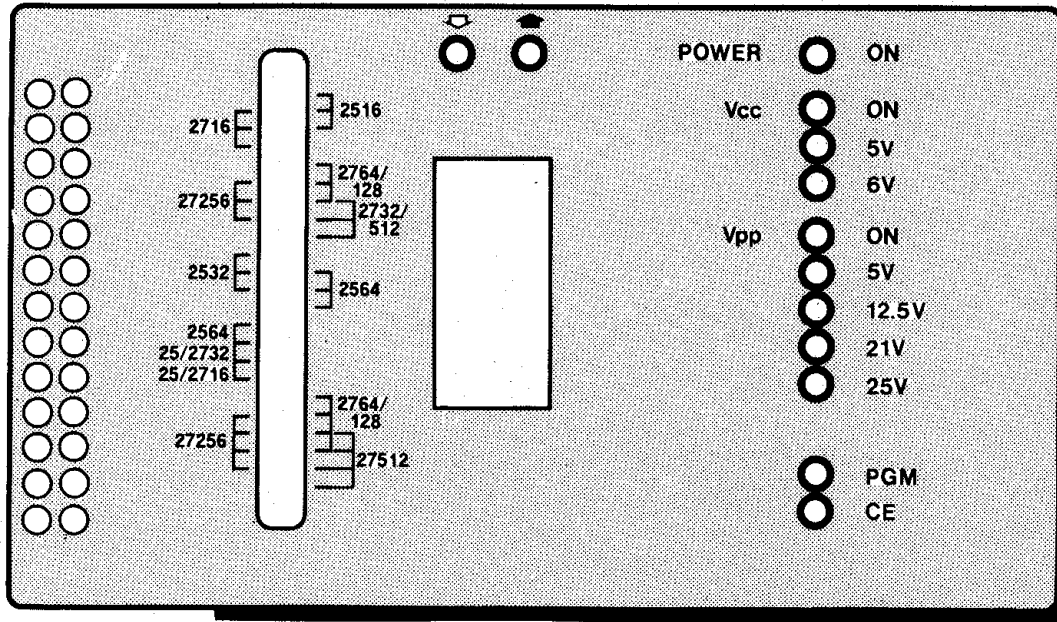
Drukken op de reset-knop van de EPROM-mer maakt dat de computer het testprogramma verlaat en terugkomt in de BASIC-interpreter.

Meet Vcc en VPP gedurende het testen om te zien of een van de weerstanden R32.. R37 eventueel aangepast moet worden om de juiste uitgangsspanningen te verkrijgen.

Pas R33 aan als Vcc niet exakt 5 V is. Druk daarna op F4 en meet of Vcc 6 V is. Zo niet,

dan moet R32 aangepast worden.

Vcc moet met een nauwkeurigheid van 0,1 V vastliggen, voor VPP geldt een tolerantie van 0,5 V.



*Figuur 6. Een mogelijke layout voor de front plaat van de EPROM-mer. Voor elk type EPROM is duidelijk aangegeven waar de jumpers op K2 geplaatst moeten worden.*

## Deel 2: Het programmeer-programma

Om de MSX-EPROM-mer van vorige maand operationeel te maken is een programma onontbeerlijk. Dit programma, geschreven in Z80-machinetaal in een 27128 EPROM, is menugestuurd en beschikt over een aantal help-pagina 's ter verhoging van het gebruikerscomfort. Bovendien is een testroutine voor de EPROM-mer zelf opgenomen en worden foutmeldingen gegeven bij weerbarstige te programmeren EPROM's. Tevens is het mogelijk om BASIC-programma's rechtstreeks in EPROM te zetten,

Bij het in bedrijf stellen van de MSX-EPROM-mer komt het bekende probleem van de hijskraan kijken. Hoe bouw je een hijskraan op? Met een andere kraan is het meest voor de hand liggend naar de meeste kranen bouwen tegenwoordig zichzelf op, eigenlijk een vorm van bootstrapping. Voor het EPROM-mer programma kan hetzelfde gelden. Als u het ervoor over heeft om ruim 8 K programmeerregels in te toetsen kunt u uw EPROM-mer aan het werk zetten en als eerste klus zou u het programmeer-programma in EPROM kunnen zetten. De moeite van het intoetsen, en het daaropvolgend onvermijdelijke foutzoeken, willen wij u besparen. Het programma is verkrijgbaar bij de software-service,

maar dat is dan de laatste EPROM die u moet laten programmeren!

Voor we het programma uitgebreid zullen beschrijven, bespreken we het intelligente programmeeralgoritme. Bij vele meestal "grote" EPROM's kan deze tijdbesparende programmeermethode toegepast worden.

### **Intelligent/ interaktief programmeren**

Met het toenemen van de geheugenkapaciteit van EPROM's is het niet meer dan logisch dat fabrikanten zochten naar snellere methoden om deze betrouwbaar te kunnen programmeren. Zou het normale programmeeralgoritme gebruikt worden dat per byte 50 ms vraagt, dan zou voor een 27256 (32 K x 8) een half uur nodig zijn. Bij de modernere algoritmes wordt afgezien van de vaste lengte van de programmeerpulsen. De nieuwe aanpak wordt verduidelijkt in het stroomdiagram (tabel 4). Allereerst wordt de voedingsspanning opgevoerd van 5 V naar 6 V. Vervolgens wordt de EPROM byte voor byte telkens 1 ms geprogrammeerd en geverifieerd. Deze programmeren verificatie-lus wordt pas verlaten als een byte geprogrammeerd en goed bevonden is of als na 25 pulsen het programmeren niet succesvol is gebleken. Nadat een byte geprogrammeerd en goed bevonden is, vindt nog een "overprogrammering" plaats gedurende drie maal de tijd die nodig was om het byte geprogrammeerd te krijgen. Een voorbeeld: voor een bepaald byte waren 9 1-ms-pulsen nodig om het goed geprogrammeerd te krijgen. De totale programmeercyclus zal nu  $(9 \times 1) + (3 \times 9) = 36$  ms duren. Daarbij verwaarlozen we de tijd die het programma dat het interaktieve algoritme uitvoert nodig heeft, maar die is zeer klein ten opzichte van de lengte van de programmeerpulsen.

Een programmeercyclus volgens dit algoritme hoeft niet noodzakelijkerwijze sneller te zijn dan het oude algoritme dat met vaste 50 ms, 20 ms (fast-1) of 10 ms (fast-2) pulsen werkt. Worst case kan een programmeercyclus zelfs  $25 + (3 \times 25) = 100$  ms duren. In de praktijk blijkt echter dat een EPROM, zo vers van de plank, in het algemeen slechts de minimale programmeerpuls van 4 ms  $(1 + 3 \times 1)$  nodig heeft om een byte te bestemmer plaats te krijgen. Het volbakken van een 27128 zal daarom slechts twee minuten duren.

Het hierboven beschreven intelligente algoritme is in wezen van Fujitsu (Quick Pro). Intel heeft voor de kleinere types EPROM's (tot 27128) een algoritme (Intelligent (sic) programming) dat afwijkt wat betreft het aantal pogingen voordat een EPROM als niet programmeerbaar beschouwd wordt (15) en de overprogrammeringfaktor (4 x), maar ook het Fujitsu-algoritme kan bij deze Intel-EPROM's toegepast worden. National Semiconductor heeft ook een eigen algoritme dat weer iets meer afwijkt. Deze fabrikant past een verhoogde programmeerspanning toe (13 V in plaats van 12,5 V) en werkt met 0,5-ms-pulsen. De MSX-EPROM-mer is niet geschikt voor dit algoritme maar bezwaarlijk is dat nauwelijks want ook National-chips onderwerpen zich gewillig aan het Fujitsu-algoritme.

De timing van de programmeerpulsen wordt logischerwijze uitgevoerd door de CTC (counter and timer controller) op de MSX-cartridge-print. Timer 0 wordt gebruikt om 4 us vertraging te creëren zodat de adres- en datalijnen zich kunnen stabiliseren. Op het countzero-sigitaal van T0 wordt de flipflop geset waarmee de programmeerpuls gekreeerd wordt. Ook wordt op dat sigitaal timer 1 in de CTC gestart die zorgt voor de

opwekking van 0,5-ms-pulsen. Timer 2, die hier in de counter-mode gebruikt wordt, telt het aantal 0,5 ms pulsen en reset de flipflop met zijn count-zero-sigitaal. De preset-waarde van T2 bepaalt de lengte van de programmeerpuls.

## **Programma beschrijving**

Een MSX-computer heeft maximaal vier primaire "slots" (sleuven), genummerd van 0 tot en met 3. Elk slot heeft een geheugenkapaciteit van 64 Kbyte, onderverdeeld in vier pagina's van elk 16 K. Het is mogelijk elk slot uit te breiden, waarbij het wordt onderverdeeld in vier "sub-slots": X-0, X-1, X-2 en X-3. Zo kunnen in theorie maximaal 16 sub-slots, van 0-0 tot 3-3, gerealiseerd worden. Omdat een Z80-processor nu eenmaal slechts 16 adreslijnen heeft, waarmee niet meer dan 64 K direct geadresseerd kan worden (vier pagina's dus) kunnen niet alle (sub-)slots gelijktijdig door de processor "gezien" worden. Elk van die vier pagina's kan in een willekeurig slot of sub-slot zitten. De absolute adressen zijn voor pagina 0 0000-3FFF, voor pagina 1 4000-7FFF, voor pagina 2 8000-BFFF en voor pagina 3 C000-FFFF. Met systeemkommando's is het mogelijk om te schakelen naar pagina's in andere (sub-)slots. Pagina 0 is gereserveerd voor het MSX-BIOS (Basic Input/Output System). Pagina 3 wordt gebruikt voor de stack, systeemvariabelen, toetsenbord-buffer en andere voor het operating system van belang zijnde aangelegenheden. Direct na het inschakelen kijkt de MSX-computer naar pagina 1 en 2 in alle slots of er ergens een programma in (EP)ROM aanwezig is. Dat programma wordt automatisch gestart als in de eerste 16 bytes een bepaalde identificatiecode aanwezig is. Wordt die code niet gevonden, dan zal de computer zijn eigen Basic-ROM op pagina 1 initialiseren en komt hij automatisch in de Basic-interpret. Deze EPROM kan op een printje geplaatst worden, bijvoorbeeld de MSX-EPROM-cartridge-print uit Elektoor april '86, waarna het geheel in een vrij slot gestoken kan worden. Het programmer-programma start automatisch na het inschakelen van de computer en staat op pagina 1. Na de nodige initialisaties zoekt het programma uit welke slots op pagina 1 en 2 RAM hebben zitten dat gebruikt kan worden als EPROM-data-veld (adres 4000.. BFFF, maximaal 32 K voor een 27256, een 27512 moet in twee slagen geprogrammeerd worden). Hierna geeft het programma weer de vrije hand aan de normale MSX-opstartprocedure. Dat wil zeggen dat de computer in de meeste gevallen opkomt met het openingsplaatje van de BASIC-interpret. Het EPROM-mer-programma kan opgeroepen worden met CALL EPROMx. "x" staat hier voor het cartridge-adresgebied (zie het artikel over de 32-bits-I/O-cartridge, januari '86). Het programma zal als het aangeroepen wordt automatisch het/de juiste slot(s) selecteren voor de RAM-buffer. Bij de eerste aanroep kopieert het programma een gedeelte van zichzelf in het hoogste nog vrije RAM-bereik in pagina 3. Het begin van dat bereik kan afgelezen worden in een systeemvariabele die ook gevonden kan worden op pagina 3. Vektor-adressering en gebruik van het BIOS waar het data-transport van het toetsenbord en naar het beeldscherm betreft, garandeert dat het programma loopt op elk type MSX-computer met tenminste 64 K RAM. Als u een data-blok, bijvoorbeeld een machinetaalprogramma, plaatst op pagina 3, moet u zich realiseren dat de nog vrije geheugenruimte door aanwezigheid van het EPROM-mer-programma beperkt is. Om te voorkomen dat u gedeeltelijk de stack overlapt (of beter gezegd: de ruimte die daarvoor gereserveerd moet worden) die ook op pagina 3

gelokaliseerd is, kunt u intoetsen: PRINT HEX (FRE(O)+&H8000) Het adres dat u hierop krijgt, moet hoger zijn dan het hoogste adres waar u zelf nog iets kwijt wilt, rekening houdend met nog een aantal te reserveren bytes voor systeem-variabelen en de stack (figuur 9). MSX-gebruikers die een apparaat met een diskdrive hebben, kunnen het DISKBASIC-werkgeheugen beperken door de CONTROL-toets ingedrukt te houden tijdens het inschakelen van de computer. Het systeem wordt daarmee te kennen gegeven dat niet meer dan een drive aangesloten is. Door tijdens het inschakelen de SHIFT-toets in te drukken, gaat het systeem uit van geen enkele aangesloten drive.

### **Te gebruiken kommando's**

Hoewel het programma extreem eenvoudig is in het gebruik, geven we hier toch een samenvatting van de kommando's die u ter beschikking staan. Daarmee krijgt u meteen een idee van wat het programma in zijn mars heeft. Na het intoetsen van CALL EPROMx verschijnt het openingsplaatje. Om "door te bladeren" naar de help-pagina's met alle EPROM-gegevens en informatie voor het programmeren drukt u op een willekeurige toets. Met behulp van de cursor-toetsen (up/down) kunt u door de help-pagina's bladeren. Het menu-scherm roept u op door op de spatiebalk te drukken. Dat menu-scherm (figuur 10) geeft u de toegang tot de hoofdmoden van het programma. Het feitelijke werken met de EPROM-mer wordt vanaf dit scherm gedirigeerd. De volgende toetsen zijn daarbij van dienst:

- Cursor up en down om naar de regel toe te gaan waar u een verandering aan wilt brengen. Met de spatiebalk kan op een regel de gewenste instelling gemaakt worden.
- Toets H om terug te gaan naar de help-pagina's.
- Toets P om een afdruk op de printer te maken van het plaatje op het beeldscherm. Indien er geen printer aangesloten is, antwoordt het programma met "NO PRINTER".
- Toets I maakt het mogelijk om een BASIC-programma in EPROM te zetten. De begin- en eind-adressen worden door het programma bepaald en daarna op het scherm getoond. Tevens worden alle link-adressen aangepast.
- Met toets T wordt het testprogramma geexecuteerd dat alle EPROM-mer functies achtereenvolgens vrijgeeft. U ziet dat aan de knipperende PGM-LED op de EPROM-mer en de beurtelings oplichtende andere LED's. Tijdens het testprogramma mag draadbrug of kortsluit-steker J1 op de programmerprint niet aanwezig zijn. Draai het testprogramma alleen als er geen EPROM in de ZIF-voet aanwezig is, deze kan anders ook geexecuteerd worden. Executie door elektrocutie wel te verstaan, want alle programmeerspanningen zullen hem achtereenvolgens aangeboden worden en een 12,5-V-EPROM bijvoorbeeld zal dat niet in dank afnemen.
- Met toets S start het programma met het uitvoeren van wat op dat moment op het menu-scherm wordt aangegeven. Dit zijn de parameters die op het menuscherm ingesteld kunnen worden:

### **EPROM-type en programmeerspanning.**

Raadpleeg tabel I of een van de help-pagina's om een bepaalde EPROM met de daarbij horende programmeerspanning te selekteren. Merk op dat EPROM BEGIN & END vanzelf meeveranderen met de geheugenkapaciteit van het ge-selekteerde EPROM-type. Ook is het mogelijk een bepaald deel van een EPROM te programmeren door een bepaald adresgebied in te toetsen (alleen hexadecimaal), dat kleiner is dan de geheugenkapaciteit van de te programmeren EPROM. Als u een adresgebied intoetst groter dan de geheugenkapaciteit van de te programmeren EPROM, of als het adresgebied van het geheugen (vastgelegd door MEMORY BEGIN & END) waar u data naar toe gaat schrijven (copy) of data vandaan haalt (program) in grootte niet overeenstemt met EPROM BEGIN & END, dan volgt een foutmelding. Een voorbeeld: u wilt de tweede helft van een 2764 (8 K) EPROM programmeren:

```
EPROM BEGIN      = 1000;  
EPROM END        = 1FFF;  
MEMORY BEGIN     = 4000;  
MEMORY END       = 4FFF.
```

BLANK CHECK is een routine die voor zichzelf spreekt. Ze controleert of een bepaald met EPROM BEGIN & END gedefinieerd adresgebied alleen maar FF-bytes bevat. Dat gebied hoeft niet de hele EPROM te beslaan. Het is mogelijk op slechts een gedeelte van de EPROM een blank check los te laten.

PROGRAM mag ook voor zichzelf sprekend verondersteld, worden. Hier maakt men gebruik van zowel EPROM BEGIN & END als MEMORY BEGIN & END.

VERIFY vergelijkt de EPROM-inhoud met die van de RAM-buffer. Ook hier wordt weer van EPROM BEGIN & END en MEMORY BEGIN & END gebruik gemaakt om de te vergelijken adresgebieden te onderscheiden.

READ AND RUN CHECKSUM laadt data van de EPROM in RAM-buffer en telt alle data-bytes op om een 16-bit check-sum te verkrijgen. DISPLAY MEMORY geeft de mogelijkheid om de inhoud van een EPROM op het beeldscherm weer te geven. Die weergave vindt plaats met opeenvolgende adressen en zowel in hexadecimaal als in ASCII-formaat. Het is niet mogelijk data te veranderen, DISPLAY MEMORY is louter een

### **Leesfaciliteit.**

Met PROGRAM MODE stelt men normaal, fast-1, fast-2 of intelligent programmeren in. De toegepaste programmeermethode hangt uiteraard af van het type EPROM. Raadpleeg daarvoor tabel I of de desbetreffende help-pagina.

ADRESS COUNTER, bijna onderaan het scherm, geeft de momentele waarde van het EPROM-adres waar geschreven of gelezen wordt.

Onder aan het scherm is een regel gereserveerd waar de volgende meldingen zichtbaar



kunnen worden:

ADDRESS ERROR is een boodschap die aangeeft dat EPROM BEGIN & END en MEMORY BEGIN & END met elkaar in tegenspraak zijn. Voer deze opnieuw in voor u opnieuw op S(tart) drukt.

BLANK vergewist u ervan dat het opgegeven adresgebied alleen maar FF-bytes bevat. Er wordt in de blank-check-routine niets van EPROM naar RAM-buffer gekopieerd.

NOT BLANK geeft aan dat een of meer adreslokaties in het gespecificeerde adresgebied een andere waarde dan FF bevat(ten). De adresteller staat op het adres waar het eerste niet-FF-byte gelezen werd.

READING COMPLETED wil zeggen dat een EPROM gelezen is en de inhoud beschikbaar is om met DISPLAY MEMORY te bestuderen. VERIFIED volgt op het voltooiën van de VERIFY-routine als er geen afwijkingen zijn gevonden.

VERIFY ERROR rapporteert een verschil tussen EPROM- en buffer-data. De adresteller staat op het adres van het eerste niet overeenstemmende byte.

REPROGRAMMABLE geeft aan , dat er een verify error is gevonden, maar dat het desbetreffende byte opnieuw geprogrammeerd kan worden zonder de EPROM eerst te wissen. Gezien op logika-nivo betekent dat, dat alleen bits die "1" zijn "0" gemaakt moeten worden om EPROM-data overeen te laten komen met buffer-data.

NOT PROGRAMMABLE is een indicatie dat een byte op het adres dat door de adresteller wordt weergegeven zich na 25 programmeerpulsen nog niet heeft laten programmeren. Deze melding treedt alleen op bij het interactief programmeren.

EXECUTION STOPPED wordt weergegeven nadat op de reset-knop van de programmer gedrukt is.

DEVICE I/O-ERROR geeft aan dat de computer geen interrupt-signalen ontvangt van de I/O-cartridge. Mogelijk wordt dit veroorzaakt door een verkeerd I/O-adres.

NO PRINTER spreekt voor zich.

ILLEGAL COMMAND ORDER is een melding die u noopt de YES/NO-instelling van het menu-scherm opnieuw uit te voeren. Merk op dat als aangegeven wordt dat BLANK CHECK, PROGRAM en VERIFY alle uitgevoerd moeten worden het programma ervoor zorgt dat ze alle drie in de juiste volgorde aangeroepen worden. Ze hoeven niet afzonderlijk gestart te worden.

```

===== ELEKTOR MSX EPROMMER =====

EPROM TYPE:                27128

PROGRAMMING VOLTAGE:      12.5V

EPROM BEGIN:              &H0000
EPROM END:                 &H3FFF

MEMORY BEGIN:             &H4000
MEMORY END:               &H7FFF

BLANK CHECK:              YES
PROGRAM:                  YES
VERIFY:                   YES
READ AND RUN CHECK SUM:  NO
DISPLAY MEMORY:          NO

PROGRAM MODE:             INTELL

CHECK SUM:                 &H0000
ADDRESS COUNTER           &H3FFF

RESULT: PROGRAMMED

```

*Figuur 10. In dit menu kunnen alle gewenste parameters ingesteld worden; EPROM a la carte.*

Tabel 5 EPROM-mer besturingswoorden (poort C)

EPROM	READ	VERIFY	WRITE
2716	0B	08 + V <sub>pp</sub>	28 + V <sub>pp</sub>
2732	0F	0C + V <sub>pp</sub>	68 + V <sub>pp</sub>
2764	0B	08 + V <sub>pp</sub>	28 + V <sub>pp</sub>
27128	0B	08 + V <sub>pp</sub>	28 + V <sub>pp</sub>
27256	0B	48 + V <sub>pp</sub>	68 + V <sub>pp</sub>
27512	0F	0C + V <sub>pp</sub>	68 + V <sub>pp</sub>
2516	0B	0B	68 + V <sub>pp</sub>
2532	0B	0B	68 + V <sub>pp</sub>
2564	0B	08	48 + V <sub>pp</sub>

alle datawoorden zijn hexadecimaal.

V<sub>pp</sub> = 5 V : 3. V<sub>pp</sub> = 12.5 : 2. V<sub>pp</sub> = 21 V : 1. V<sub>pp</sub> = 25 V : 0.

Bit 7 van poort C is als input geprogrammeerd.

*Tabel 5. De verschillende besturings-woorden voor de EPROM-programmer die aan poort C aangeboden moeten worden.*

Het zal duidelijk zijn dat het menu-scherm grondig gecontroleerd moet worden op het juist ingesteld zijn van alle parameters alvorens met "S" de zaak in gang gezet wordt. Krijgt u een foutmelding terug dan is er geen reden tot paniek. Zolang er maar geen te hoge programmeerspanning ingesteld is, kan er eigenlijk niets stuk. Zoek op het menu-scherm de eventuele foute instelling en probeer het gewoon opnieuw. Na een paar minuten werken met dit programma zult u ervaren hoe gebruikersvriendelijk het is. Ter herinnering: met "H" kunt u de help-pagina's oproepen. Mocht u een onbekende EPROM hebben waarvan niet bekend is hoe deze geprogrammeerd moet worden, probeer dat dan gewoon uit, beginnend met de laagste (12,5 V) programmeerspanning. Kies voor normaal programmeren (bij intelligent programmeren wordt de voedingsspanning tot 6 V opgevoerd hetgeen niet altijd gewenst is).

Nog een paar tips.

- Het is mogelijk om bij een geprogrammeerde EPROM voor de documentatie een screendump te maken. Alle instellingen van het menu-scherm heeft u zo netjes op een rij en ook de check-sum van de geprogrammeerde EPROM heeft u dan op papier.
- Een 27512-EPROM (64 K) moet in twee slagen geprogrammeerd worden omdat de RAM-buffer niet groter is dan 32 K.
- Met CTRL STOP verlaat u het EPROM-mer-programma en komt u terug in de MSX-BASIC.
- Ook met andere MSX-software kunt u de RAM-buffer vullen met data die in EPROM geprogrammeerd moeten worden, maar pas op dat met deze software het EPROM-mer-programma zelf of de stack niet overschreven wordt. Let op:, na het draaien van een BASIC-programma dat gebruik maakt van het PLAY-kommando, moet de computer gereset worden. Dat is nodig omdat dit anders de jump table en variabelen in de voice queue area aantast. Voor het EPROM-mer-programma maakt dat een nieuwe initialisatie nodig. Deze kan uitgevoerd worden door de computer opnieuw te starten met de EPROM-cartridge voorzien van het programma in een slot.

In tabel 5 zijn de diverse besturingswoorden af te lezen voor diverse types EPROM's. Lezers die zelf hun besturingsprogramma willen schrijven voor de EPROM-mer kunnen daarmee in de slag.

### **Opstarten**

Begin met het installeren van jumpers B, D, E en I op de EPROM-cartridge-print. Plaats vervolgens de EPROM met het programma in het 28-polige voetje. Plaats de cartridge in een vrije konnektor-gleuf van de MSX-computer en plaats de 32-bits-I/O-cartridge in een andere vrije gleuf. Is deze niet meer aanwezig, gebruik dan de konnektor op de EPROM-cartridge. De EPROM-mer zelf wordt met 50-polige flat-cable op de I/O-cartridge aangesloten (figuur 11). Het is niet mogelijk om samen met de I/O-cartridge de MSX-busprint (Elektuur mei '86) te gebruiken. Plaats nog niet meteen een EPROM in de ZIF-voet. Schakel eerst de computer in en roep het EPROM-mer-programma aan nadat het zijn initialisatie voltooid heeft. Ga nu naar het menu-scherm en laat het test-programma lopen door op "T" te drukken. Controleer of alle LED's van de programmer op hun beurt willen oplichten. Als dat het geval is, kan in alle redelijkheid aangenomen worden dat hard- en software naar behoren functioneren. Een EPROM in de voet gestoken en het feest kan beginnen.