

32-BITS I/O-CARTRIDGE

voor MSX-computers

elektuur januari 1987

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

Hier is 'ie dan. Zoals beloofd. Een I/O-interface voor MSX-computers. Eindelijk kunnen nu ook MSX-bezitters de computerskoop op hun machine aansluiten. Maar, en dat is het leuke ervan, niet alleen de computerskoop. Met deze I/O-print kan de cartridge-aansluiting van elke MSX-machine omgetoverd worden tot een bijdetijdse I/O-poort. MSX-computers zijn zonder meer fijne machines, maar een I/O-poort, nee, die zit er niet op. Probeer dan bijvoorbeeld maar eens het apparaat als besturingscomputer in te zetten. Gelukkig zitten op de cartridge-aansluiting van een MSX-machine alle signalen die noodzakelijk zijn om (met behulp van wat extra elektronica) I/O-verkeer mogelijk te maken.

Het is dan ook niet meer dan "Elektuurlijk" dat wij van dit gegeven gebruik maken. In eerste instantie hebben we deze schakeling ontworpen als interface voor de EPROM-programmer die we begin '87 zullen publiceren.

Specificaties

Belangrijk om te weten is natuurlijk welke mogelijkheden de print biedt. We zetten alles eens op een rijtje: 4x8 I/O-lijnen 4 programmeerbare timers 4 adresseringsgebieden binnen het I/O-bereik (eenvoudig te kiezen m.b.v. een tweetal jumpers).

Volledig daisy-chain-gekoppeld, zodat eventueel interrupt-programmering met prioriteiten mogelijk is.

Blokschema

Het blokschema (figuur 1) laat zien dat de schakeling bestaat uit twee adresdekoders, twee I/O-poorten en een timer-blok. Bij MSX-computers is het I/O-gebied "I/O-mapped", zodat dit geen geheugenplaatsen in beslag neemt. Voor de adressering worden de 8 minst signifi-kante adreslijnen (A0... A7) gebruikt. Om nu verschil te maken tussen het I/O- en het geheugenbereik, heeft de processor in de MSX-computers (de Z80) een speciale uitgang die laag wordt wanneer het I/O-bereik geselecteerd moet worden.

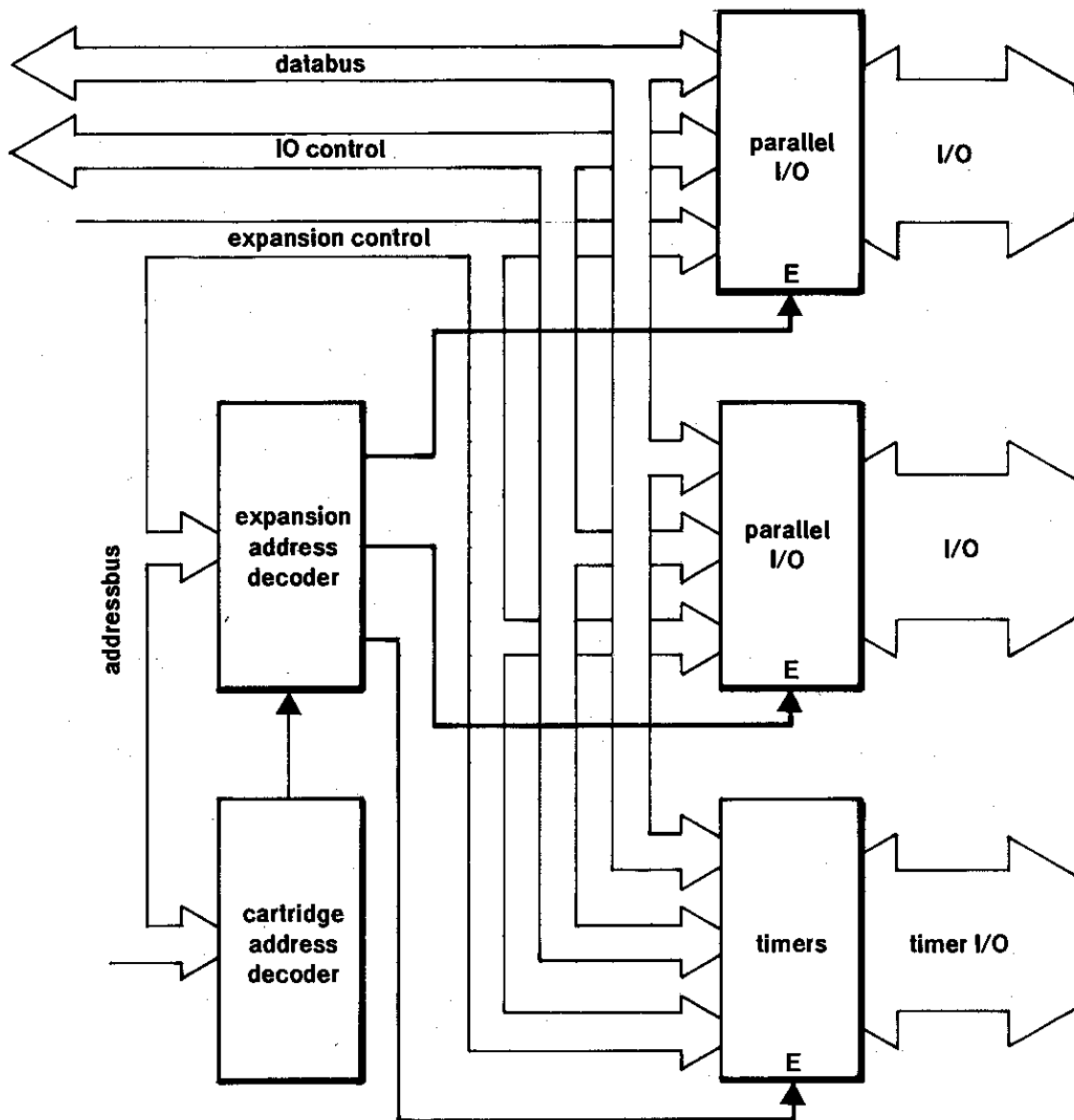
Omdat de systeem-software van de MSX-computers gebruik maakt van de I/O-adressen 64... 255 (&H40...&HFF), kunnen we alleen maar adressen gebruiken die lager liggen. Dit houdt in dat A6 en A7 altijd nul moeten zijn. Het adresbereik van 0 tot 64 (&H00 tot &H3F) blijft dus vrij voor "persoonlijk" gebruik.

De adresdekodering bestaat uit twee delen. Het eerste deel dient om de I/O-cartridge te selekteren. Zodra dat gebeurt, verschijnt op de uitgang een enable-sig-naal dat aan de tweede dekoder wordt doorgegeven. Deze selekteert één van de twee parallele I/O-poorten of de timer. Per I/O-blok kan men over 2x8 I/O-lijnen en 2 x 2 hand-shaking-lijnen beschikken. In totaal dus 32 I/O-lijnen en 4 control-lijnen, wat voor de meeste

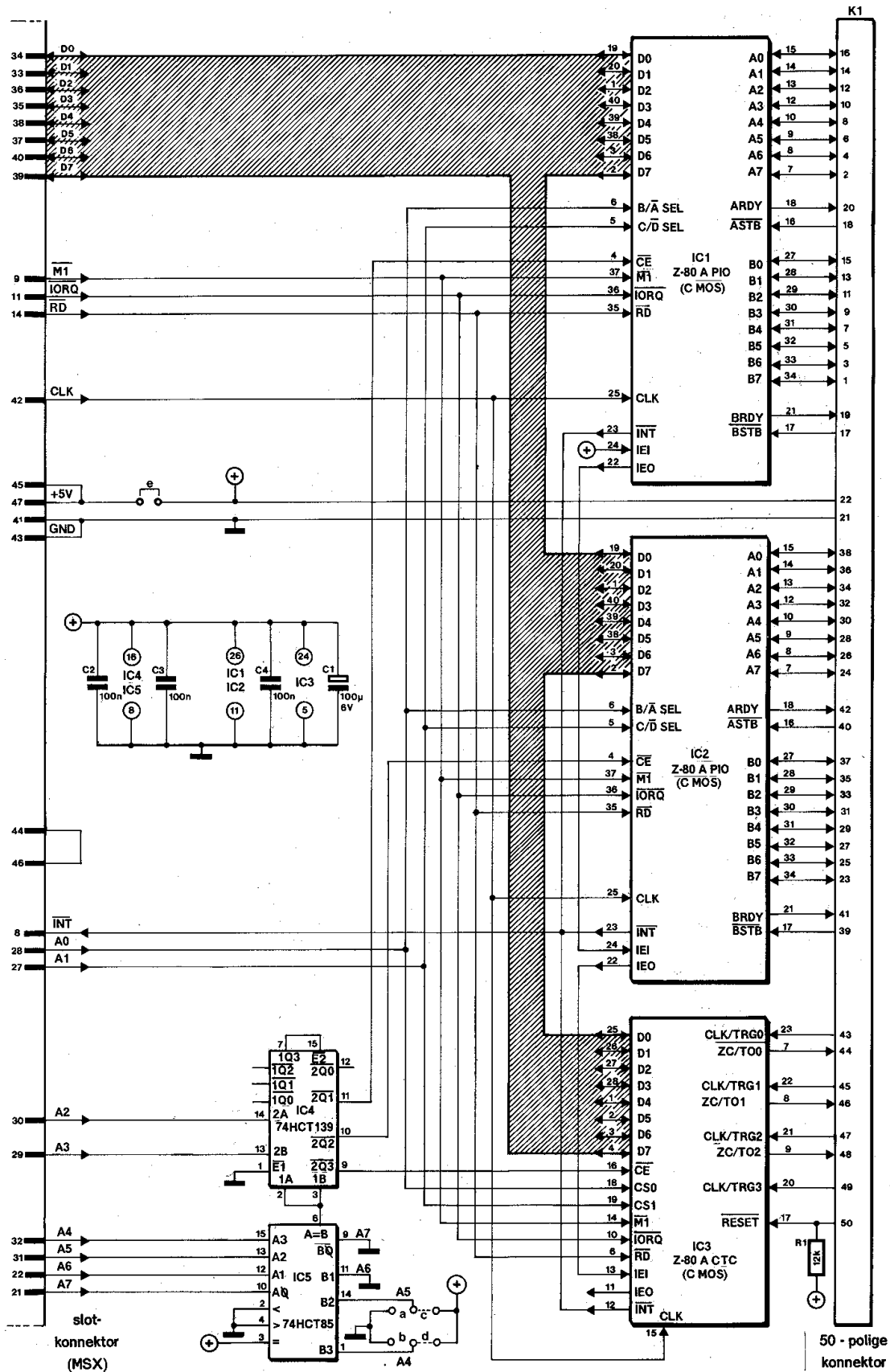
applicaties ruim voldoende zal zijn. De 4 timers kunnen naar keuze als teller of als timer worden ingezet.

jumper	adresgebied
a en b	&H0 (&H00 - &H0F)
a en d	&H1 (&H10 - &H1F)
c en b	&H2 (BH20 - &H2F)
c en d	&H3 (&H30 - &H3F)

Tabel 1. De I/O-adres-blokken die met de jumpers ingesteld kunnen worden.



Figuur 1. Het blokschema van de MSX-I/O-timer-print.



Figuur 2. Het schema.

De hardware nader bezien

Zoals uit figuur 2 blijkt, is er niet zo gek veel verschil tussen het blokschema en het "echte" schema. De grote overeenkomst ontstaat doordat in de Z80-familie complete IC's bestaan voor de gewenste functies.

De adresdekoder bestaat uit IC4 en IC5. Met IC5 (een komparator) wordt de kaart geselecteerd. Het IC vergelijkt de data op de vier adreslijnen A4... A7 met de ingestelde niveaus op de punten BO... B3. Op het moment dat de komparator gelijke data ziet, wordt uitgang A=B (pen 6) hoog. Dit is het teken dat de kaart geselecteerd is.

Zoals al gezegd, mogen de I/O-adressen 64... 255 (&H40... &HFF) niet gebruikt worden. De adreslijnen A6 en A7 worden daarom vergeleken met "0" (pen 9 en pen 11 liggen aan massa). De vergelijkings-data voor A4 en A5 kunnen met behulp van de jumpers a, b, c en d ingesteld worden. Hiermee ligt dan het I/O-gebied vast dat door de kaart in beslag genomen wordt (zie tabel 1).

Met het hoog worden van de A = B-uitgang wordt IC4 geactiveerd. Dit IC bevat een dubbele 2-bits binaire dekoder/ demultiplexer (2 naar 4). Omdat de enable-ingang (pen 15) van de dekoder actief laag is, wordt één dekoder als inverter gebruikt (wanneer de ingangen IA en IB hoog zijn, wordt uitgang 1Q3 laag). Door de tweede dekoder worden de bits 2A en 3A gedecodeerd naar 3 enable-lijnen voor IC1... IC3. Tenslotte zijn twee expansion-control-lijnen nodig, die aangeven of de data op de databus bedoeld zijn om de PIO's (Parallele I/O-controller) in te stellen of bestemd zijn als data die via de PIO's naar buiten gebracht moeten worden. Bij de CTC (Counter/Timer Controller) bepalen deze twee lijnen welke van de vier timers de data op de databus moet overnemen. Hiervoor hebben we AO en AI gekozen. Bij het Z80-systeem wordt gewerkt met een IORQ-lijn (I/O request) die aangeeft of het adres dat op de adresbus aangeboden wordt, een adres uit het geheugenbereik of uit het I/O-bereik is. Deze lijn wordt verbonden met de daarvoor bestemde ingang van IC1... IC3, zodat deze drie IC's pas actief worden wanneer de IORQ-lijn laag is. Wij hebben al gezegd dat het werken met interrupts mogelijk is. Hiervoor zijn de interrupt-uitgangen (INT) van IC1... IC3 aan elkaar gekoppeld. Genereert nu één van de IC's een interrupt, dan maakt dit IC de INT-lijn laag. De processor zal nu het lopende programma onderbreken om naar een interrupt-programma te springen. Het minst significante byte van het adres waarop de processor het startadres van het interrupt-programma kan vinden, wordt door de processor uit het onderdeel gehaald dat de interrupt heeft aangevraagd. Dit wordt gedaan door de IORQ-lijn en de MI-lijn laag te maken. Het onderdeel dat de interrupt aangevraagd heeft, reageert daarop met dit byte op de databus te zetten. Bij het initialiseren van de kaart moet in elk IC dit byte als een vector geladen worden. De kaart bevat drie delen die een interrupt kunnen aanvragen. Deze drie delen zijn in een zogenaamde daisy-chain geschakeld om verschil in prioriteit te kunnen maken. Hiervoor zijn de IEI- en IEO-lijnen (interrupt enable input en output) van IC1... IC3 met elkaar doorverbonden, zoals te zien is in figuur 2. Wordt bijvoorbeeld door IC1 een interrupt gegenereerd, dan wordt de IEO-uitgang laag. Dit niveau wordt doorgegeven aan de IEI-ingang van IC2 die dit niveau op zijn beurt weer doorgeeft aan IC3. Beide IC's kunnen daardoor geen interrupt aanvragen totdat het interrupt-

programma voor IC1 afgewerkt is. Een RETI-instructie reset het IEO-nivo. We zien dus dat IC 1 de hoogste prioriteit heeft en IC3 de laagste.

De schakeling kan zowel uit de computer als uit een externe voedingsbron worden gevoed. In het laatste geval moet dan wel draadbrug e verwijderd worden en moet de voeding via de uitgangskonnektor van de kaart geschieden. Gebruikt men CMOS-IC's, dan kan de schakeling zonder meer uit de computer zelf worden gevoed. Het voedingsgedeelte in een MSX-computer kan namelijk 300 mA per cartridge-aansluiting leveren en dat is voor CMOS-IC's meer dan voldoende. Gaat het echter om gewone IC's, dan kan de maximale stroom in het ongunstigste geval oplopen tot zo'n 320 mA. Ofschoon bij ons prototype de stroomopname slechts 100 mA bedroeg, is het toch raadzaam om eerst na te gaan hoeveel stroom de schakeling trekt voordat deze op de computer wordt aangesloten.

Programmeren van de PIO's

De Z80A PIO van Zilog bevat twee 8-bits poorten die afhankelijk van de initialisatie voor byte-output (mode 0), byte-input (mode 1), byte-input/out-put (mode 2) of bit-input/output (mode 3) ingesteld kunnen worden. De modi 0, 1 en 2 kunnen werken met interrupts als de processor in interrupt-mode 2 staan. Om dit te kunnen instellen, hebben we een machinetaalprogramma nodig dat de processor-status verandert. Bij het werken met interrupts is het wel vereist dat de interrupts van de VDP (video-display-processor) uitgeschakeld worden. Dit kan met de BASIC-instructie "VDP(1) = VDP(1) AND 223". Deze instructie moet voorafgaan aan het programma dat gebruik maakt van interrupt-mode 2. Na het beëindigen van het programma moet de interrupt-mode weer teruggezet worden in de oude stand. Dit doen we met de BASIC-instructie "VDP(1) = VDP(1) OR 32". Vergeten we de interrupt-mode te herstellen, dan kan niet meer met het beeldscherm of het toetsenbord gewerkt worden. Om de PIO's in de gewenste modus te schakelen, moet het bij de modus behorende kom-mando-byte in de kommando-registers van de PIO's geschre-ven worden. Hiervoor moet de C/D-lijn hoog zijn, terwijl het nivo op de B/A-lijn bepaalt of we het kommando-register van de A-kant of de B-kant programmeren.

Gezien de complexiteit van de verschillende modi, zullen we ons alleen verdiepen in mode 3, die op een eenvoudige wijze vanuit BASIC te programmeren is. Zij die geïnteresseerd zijn in de andere modi, verwijzen we naar het Zilog-databoek over de Z80. Het initialiseren van mode 3 wordt gedaan door in de juiste volgorde de volgende kommando-bytes in de PIO's te laden:

Mode-control-byte = &HFF (definieert mode 3); I/O-register-control-byte = &Hxx (zie voorbeeld); Interrupt-control-byte = &H07 (interrupt disabled); interrupt-disable-byte = &H03 (is niet vereist).

De data die in het I/O-register geschreven worden, bepalen welke lijnen als ingang of als uitgang geschakeld zijn. Staat er een nul in één van de bits van het register, dan is de bij dit bit behorende lijn als ingang geprogrammeerd. Is het bit één, dan is de bijbehorende lijn een uitgang. Voorbeeld: Wordt in het register het getal &HFO geschreven, dan zijn de lijnen AO.. A3 als ingang geprogrammeerd en de lijnen A4i, A7 als uitgang. Van beide PIO's moeten de twee poorten op de voorgaande manier geïnitieerd worden. Op welke I/O-adressen de onderdelen van de PIO's en CTC zich bevinden, is te zien in tabel

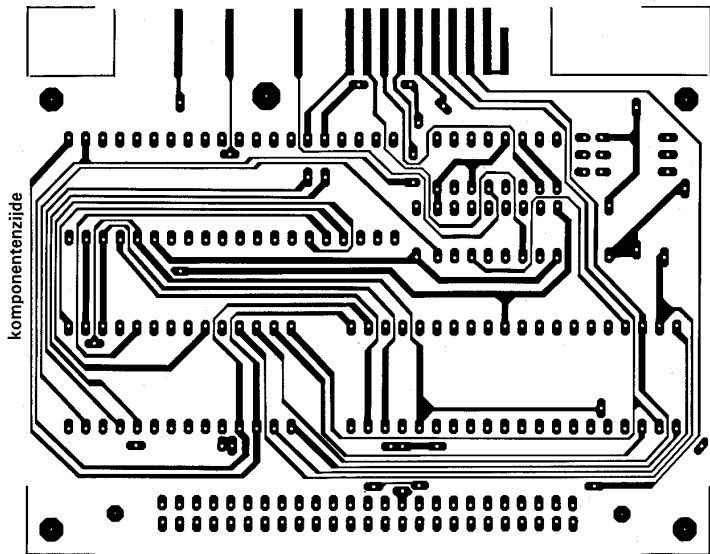
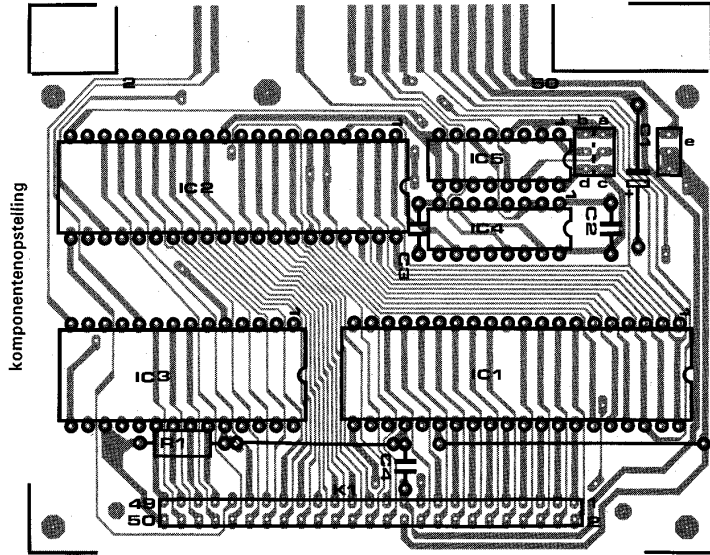
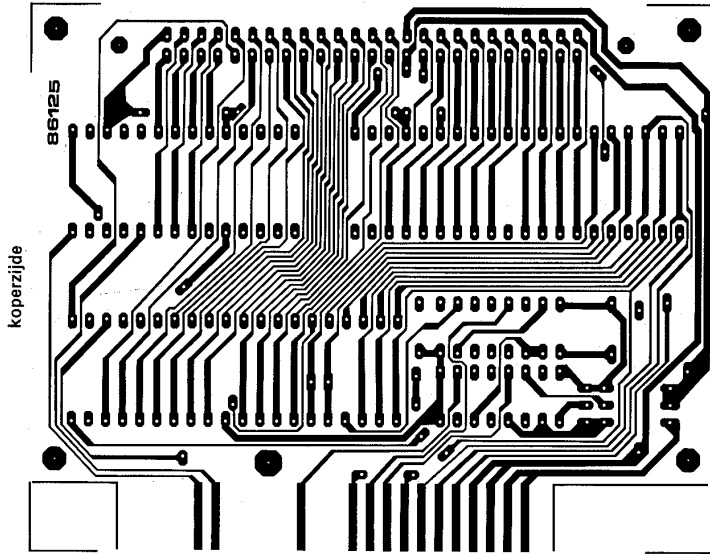
2. Tussen haakjes staat de afkorting van de naam zoals wij die in onze programma's hanteren. Na het initialiseren van de kaart kunnen data via de poorten naar binnen gehaald worden of naar buiten gestuurd worden door respectievelijk het lezen van het I/O-adres of het schrijven naar dit adres. Met BASIC kan het ook. Hierbij gebruiken we het kommando "INP (X)" om data van poort X te lezen of "OUT X,n" om een getal n naar poort X te schrijven.

Adres	Functie				
&HX4	IC1,	data,	Kant	A	(DA)
&HX5	IC1,	data,	kant	B	(DB)
&HX6	IC1,	control,	kant	A	(CA)
&HX7	IC1,	control,	kant	B	(CB)
&HX8	IC2,	data,	Kant	A	(DO)
&HX9	IC2,	data,	Kant	B	(DD)
&HXA	IC2,	control,	Kant	A	(CC)
&HXB	IC2,	control,	Kant	B	(CD)
&HXC	IC3,	timer 0			(T0)
&HXD	IC3,	timer 1			(T1)
&HXE	IC3,	timer 2			(T2)
&HXF	IC3,	timer 3			(T3)

Tabel 2. De I/O-adressen van de diverse registers. De afkortingen die tussen haakjes staan, zijn hiervoor gebruikt in het programma.

	Functie		
CTC-bit	laag (0)	hoog (1)	Opmerking
D0	Vektor	control-byte	
D1	_	software-reset	
D ₂	er volgt geen tijdconstante	er volgt een tijdconstante	
D3	triggert door tijdconstante te laden	klok/trigger-puls start timer	alleen timermode
D4	neergaande flank	opgaande flank	klok- of triggerpuls
D5	klok / 16	klok / 256	alleen timermode
D6	timer-mode	counter-mode	
D7	enable INT	disable INT	

Tabel 3. De functies van de bits in het control-register van de Z80-CTC.



Figuur 3. Koper-layout en componenten-opstelling van de cartridge-print.

Onderdelenlijst

Weerstand:

R1 = 12 k

Kondensatoren:

C1 = 100 uF6 V

C2...C4 = 100 n

Halfgeleiders:

IC1,IC2 = Z80A-PIO*

IC3 = Z80A-CTC*

IC4 = 74HCT139

IC5 = 74HCT85

* CMOS-IC's zijn wenselijk; zie tekst

Diversen:

9 kontaktpennen voor jumpers

3 jumpers

K1 = 50-polige konnektor

1 compact-cassettedoosje (zie tekst)

Geschatte bouwkosten: circa f 75, —

Programmeren van de CTC

De Z80-CTC bevat 4 timers/ counters die ieder apart ingesteld kunnen worden. Voor de initialisatie moet in elke timer/counter een control-woord geschreven worden, eventueel gevolgd door een tijdconstante-woord. De functies van de bits in het control-woord zijn te vinden in tabel 3. Van het control-woord moet bit D0 altijd "1" zijn. Is dit niet het geval, dan ziet de CTC dit woord als een interrupt-vektor. Deze vektor wordt tijdens de interrupt-afwerking naar de processor overgebracht en fungeert als LSB van het adres waar het be-ginadres van het interrupt-programma is opgeborgen. Als MSB wordt de inhoud van het I-register gebruikt. Van de 8-bits-vektor die in de CTC geladen is, zijn de bits D7... D3 vrij te programmeren. De bits D2 en D1 worden door de CTC geprogrammeerd en geven aan welke timer/counter een interrupt heeft aangevraagd. De interrupt-vektor wordt in timer 0 geladen en is dan voor alle timers geprogrammeerd. Wanneer de CTC als timer gebruikt wordt, is een timer-konstante vereist. Deze konstante programmeren we direkt na het control-woord. In het control-woord moet bit D2 dan "1" zijn. Is eenmaal een timer-konstante geprogrammeerd, dan is het niet strikt noodzakelijk om bij het veranderen van het control-woord een nieuwe konstante in te voeren. In dit laatste geval moet bit D2 "0" zijn. Aan de hand van de timer-konstante bepaalt de CTC na hoeveel klokpulsen de ZC/TO-uitgang hoog wordt. Elke timer/counter blijft doorwerken totdat de CTC een software- of een hardware-reset ontvangen heeft. De

software-reset wordt gegeven door bit D1 van het control-woord hoog te maken. Een hardware-reset wordt gegeven door pen 17 van de CTC laag te maken.

De bouw

De I/O-timer-module is bedoeld als insteek-unit voor de cartridge-aansluiting van de MSX-computer. Hierdoor is het noodzakelijk dat de print de juiste afmetingen en steek voor de printkonektor heeft., Het is daarom aanbevelenswaardig om gebruik te maken van de Elektuurprint die in de EPS is opgenomen (zie figuur 3). Deze heeft namelijk de juiste afmetingen en past zonder problemen in de konektor van de computer.

Als behuizing voor de I/O-timer-kaart gebruiken we een doosje van een compact-cassette. Dit doosje past namelijk precies in het gat van de cartridge-aansluiting. Voordat we de onderdelen op de print zetten, beginnen we met het prepareren van het doosje.

Verwijder de nokken uit het deksel en maak de uitsparingen in de onderkant, zoals aangegeven is in figuur 4 (zie ook de foto). De print moet nu dusdanig in de behuizing gemonteerd kunnen worden, dat de printkonektor exakt in het midden van het gat komt te zitten. U zult zien dat dit alleen kan wanneer u de IC's direkt op de print soldeert. De hoeveelheid onderdelen op de print is gering, maar bij het solderen moeten we er wel op letten geen kortsluiting te maken tussen de verschillende sporen. De sporen zitten nogal dicht opeen, zodat nauwkeurig solderen vereist is. Vergeet niet de twee draadbrugjes op de print aan te brengen, zoals is aangegeven op de componentenopstelling.