

MSX geheugen-structuren

De geheugen-geheim

De komst van MSX2 met memory-mapper heeft heel wat verwarring gezaaid. Zowel bij de hobbyisten als bij de professionele software-ontwikkelaars wordt er heel wat op het hoofd gekrabbd, als het over het geheugen in een MSX-computer gaat.

Tot voor kort ook bij ons, overigens. Heel wat vragen omtrent die geheugen-structuren konden we eerlijk gezegd geen zinnig antwoord op geven. Dus hebben we het allemaal maar eens tot in de puntjes uitgezocht.



Onze konklusies - met de nodige uitleg erbij - treft u in dit lijvige artikel aan. Verplichte kost voor iedere MSX-bezitter, zowel voor de beginner als voor de gevorderde.

Standaard

In de MSX standaard ligt een heleboel vast. We weten nu zo langzamerhand wel dat we de Z80a microprocessor tot onze beschikking hebben. Ook de TMS-9918A video-processor is een oude bekende en zijn opwaarts compatibele opvolger van Yamaha, de V9939 heeft al veel van z'n geheimen moeten prijsgeven aan de videokunstenaars onder de MSX2 gebruikers.

Minder duidelijk is het echter hoe nu het processor-geheugen in de computer is ingebouwd. Het processor-geheugen wordt van het video-geheugen onderscheiden omdat deze laatste geheel bestuurd wordt door de video-processor en niet door de Z80a.

De Z80a kan - zoals alle versies van de Z80 - 'slechts' 64 KBytes aan geheugen besturen. De reden hiervoor is dat hij een 16-bits adresbus bezit. Door middel van deze bus kunnen we elk geheugen-plaatsje, dat zich zowel in RAM als in ROM kan bevinden, aansturen.

Dat '16-bits bus' betekent dat er parallel - tegelijkertijd - 16 signalen verstuurd kunnen worden, die ieder uit een logische 1 of 0 bestaan. Elk signaal kan dus twee waarden aannemen.

Stel dat we een 1-bits bus hebben, dan zouden er maar twee geheugenplaatsen 'geadresseerd' kunnen worden. Ne-

men we er echter twee, dan komen we al aan vier geheugenplaatsen, namelijk die met de adressen 00, 01, 10 en 11. Elke keer dat er een bit aan de adresbus wordt toegevoegd zal de capaciteit om geheugenplaatsen te kunnen adresseren met twee vermenigvuldigd worden.

Geheugen-grootte

Als we het totaal willen uitrekenen zonder al deze tussentapjes kan ook 2 tot het aantal bits van de adresbus verheven worden. Bij de Z80a leidt dat dus tot 2 tot de macht 16, wat gelijk is aan 65536.

Het 'natuurkundige' voorvoegsel K betekent duizend, net zoals in kilometer (km) en kilogram (kg). In het computerjargon staat de K echter voor de hoeveelheid geheugen die een 10-bits adresbus zou kunnen adresseren. Dit aantal is gelijk aan 1024. Willen we het totale geheugen uitdrukken in het aantal K's dan delen we 65536 door 1024 en zo krijgen we die beroemde 64K. Daar op elk geheugen-plaatsje een byte - een 8-bits getal - staat, spreken we van 64 Kbyte.

Het lijkt er op dat een Z80a - en dus ook een MSX - maximaal slechts 64K aan geheugenruimte kan adresseren. Echter, bijna alle MSXen bevatten meer dan die 64K, als al het processor-geheugen bij elkaar opgeteld wordt.

Pagina's

De sleutel tot de oplossing van dit probleem is dat het totale geheugen verdeeld wordt in 4 blokken van 16K. Dit geldt zowel voor RAM - het 'werkge-

heugen', dat zowel gelezen als beschreven kan worden - als ROM - het 'vaste geheugen', waarin bijvoorbeeld de Basic-interpret is ondergebracht - en is van toepassing bij zowel de reeds bij MSX1 gebruikte slot-select methode als bij de MSX2 memory mapper.

Door nu blokken te gaan 'verwisselen' kan toch het hele geheugen gebruikt worden, maar niet op hetzelfde ogenblik. Pagina's van ieder 16K kunnen worden in- of uitgeschakeld naar behoeven. Er zitten echter wel enige restricties aan dit principe.

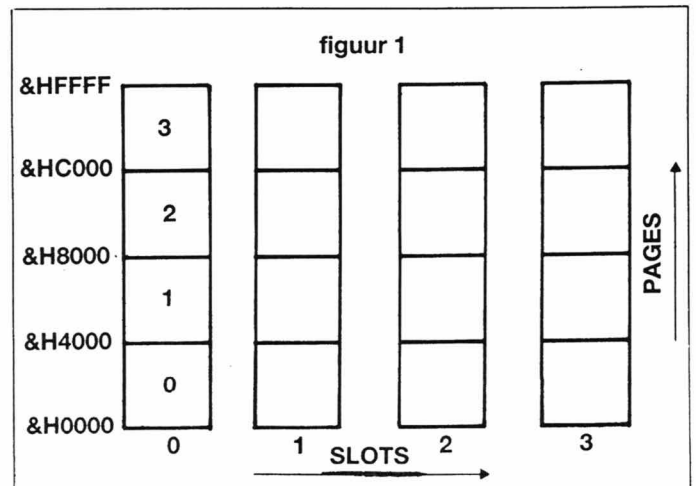
Laten we daarom eens gaan kijken hoe deze oplossing in zijn werk gaat op een 'standaard' MSX1 machine. Onder die standaard MSX1 machine verstaan we voor het gemak even: 64 Kbyte processor-RAM, 32 Kbyte ROM en twee uitbreidings-poorten, welke zowel uit een cartridgepoort als uit een expansionpoort kunnen bestaan. Hoe een en ander bij andere MSX-computers, zoals de ietwat speciale

X'press van SpectraVideo en de MSX2-machines, in zijn werk gaat wordt later verduidelijkt.

Bij die standaard machine kunnen we 16 geheugenblokken kwijt, welke onderverdeeld zijn in vier groepen van ieder vier blokken van 16K. Die groepen worden 'slots' genoemd, en de term die normaal voor die 16K blokken gebruikt wordt is 'page'. Er valt wel wat voor te zeggen om ze te vertalen in iets als poorten en pagina's, maar in de meeste literatuur - en in MSX Computer Magazine - over dit onderwerp worden de Engelse termen aangehouden. Allemaal verschillende uitdrukkingen voor het zelfde werkt tenslotte alleen maar verwarring in de hand.

Indeling

Die vier slots hebben een nummer gekregen, te weten 0 tot en met 3. Zoals bij computers gebruikelijk telt ook hier de nul mee. Ook de vier pages worden van elkaar onderschei-



en van MSX1 en MSX2 ontsluit!



den door middel van de cijfers 0 tot en met 3. Hier komt nu een van de eerste restricties om de hoek kijken. Een bepaalde page kan maar op een plek van het geheugen-bereik staan. Page 0 staat in het geheugen gebied vanaf adres &H0000 tot en met adres &H3FFF. Page 1 staat dan in het gebied tussen &H4000 en &H7FFF, page 2 tussen &H8000 en &HBFFF en ten slotte page 3 tussen &HC000 en &HFFFF. Dan is dan meteen de gehele 64K die de Z80a kan adresseren, zie figuur 1. In bijvoorbeeld het geheugen-gebied vanaf &H4000 tot en met &H7FFF - page 1 dus - kunnen echter wel vier verschillende geheugenblokken worden geplaatst. Dat zijn achtereenvolgens page 1 uit slot 0, page 1 uit slot 1, page 1 uit slot 2 of page 1 uit slot 3. Ditzelfde principe gaat ook op voor elke andere page. Voor dat wisselen van pages wordt normaal de uitdrukking 'slotselecting' gebruikt. Door de pages namelijk te wisselen wordt in feite steeds weer een nieuw slot samengesteld.

Slot opbouwen

Laten we dat eens verduidelijken met een voorbeeld, zie figuur 2. We kiezen page 0 in slot 0, page 1 plaatsen we in slot 2, page 2 in slot 3 en page 3 in slot 1. We hebben nu ons gebruikslot samengesteld, geselecteerd uit de diverse mogelijkheden. De processor kan nu het gehele geheugenbereik overzien, wat kris-kras door de slots heen loopt. Leest de processor bijvoorbeeld de geheugen-plaats &H832A uit, dan kijkt hij dus in page 2, welke in dit voorbeeld in slot 3 geschakeld staat. Adres &H0010, page 0, vinden we in slot 0, terwijl &HFED2 - page 3 - in slot 1 geschakeld is.

Toch blijkt in de praktijk dat we niet zomaar voor een bepaalde page in een bepaald slot terecht kunnen. Neem nu onze 'standaard' MSX1 computer. Bij elke MSX - ook bij MSX2 - zit in slot 0, page 0 de BIOS-ROM.

Dit BIOS-ROM bevat het machinetaal-programma dat de gehele computer bestuurt. Hij zorgt ervoor dat de gegevens die op het beeldscherm moeten komen naar de video-processor worden doorgestuurd, kijkt of er een toets wordt ingedrukt op het toetsenbord, stuurt de geluidsgegevens naar de soundchip, leest de joystick uit etcetera.

Wordt page 0 zomaar in bijvoorbeeld slot 3 geplaatst, dan raakt de processor zijn besturings-programma kwijt waarna de computer niet meer 'weet' wat te doen. Resultaat: de machine slaat vast of reset zichzelf.

Noodzakelijk

Normaal gesproken laten we page 0 dan ook in slot 0 staan, behalve als een bepaald programma die besturingstaak - gedeeltelijk - kan overnemen. Dit gebeurt bijvoorbeeld als er onder MSX-DOS gewerkt wordt. Overigens heeft MSX-DOS de BIOS - de naam is een afkorting van Basic Input/Output System - wel nodig en spreekt hem aan via een zogenaamde 'inter-slot call', maar dat is materie die pas later aan de orde zal komen.

In de tot nu toe uitgekomen MSX-machines zit in page 1, slot 0 de MSX1 BASIC interpreter. Dit programma heeft de taak alle MSX1 Basic-kommando's te vertalen en uit te voeren. Dat betekent dat er ook met page 1 moeten worden uitgekeken, want als we onder Basic werken en we schakelen page 1 in een ander slot dan 0, dan krijgen we dezelfde vervelen-

de effecten als bij het omschakelen van de vorige page, page 0.

Machinetaal

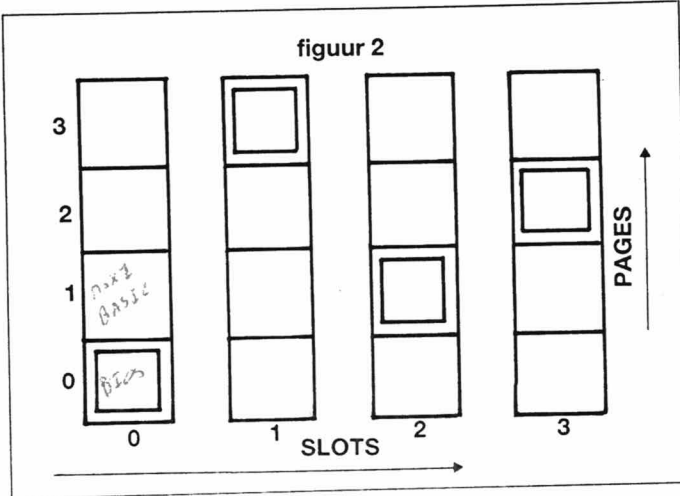
Echter, in machinetaal hebben we de Basic-interpretatie niet nodig. Sterker nog, hij kan beter in machinetaal helemaal niet gebruikt worden. De MSX1-ROM is niet bij elke machine hetzelfde en er plompverloren - zonder de BIOS aangesproken te hebben - met een machinetaal opdracht 'inspringen' kan allerlei verschillende en meestal funeste gevolgen hebben bij de verschillende MSX-computers.

Compatibiliteit

In onze standaard MSX1 machine zit de 64K RAM in een van de drie overgebleven slots. Het kan zich zowel in slot 1, 2 als 3 bevinden. Het verschijnsel dat binnen de MSX-standaard het niet vast ligt in welk slot - of welke slots, die RAM mag ook over meerdere slots verdeeld zijn - deze 64K RAM geplaatst is, blijkt in 95 procent van de gevallen de oorzaak van de MSX incompatibiliteit-problemen te zijn. Bij MSX2 is het probleem van de RAM nog complexer, maar ook daarover later meer.

Wat is er nu aan de hand. In de tabel in figuur 3 staat een redelijk volledige lijst van de in Nederland en België op de markt zijnde MSX-computers. In die tabel staat aangegeven in welk slot de RAM zit, met eventuele opmerkingen daarbij. Bij de meeste MSX1 machines is dat slot 1, 2 of 3. De Mitsubishi ML-FX1 is de

figuur 2



nige 64K MSX1 die wel eens problemen wil geven, want wat dat aangaat reageert deze computer in feite als een MSX2. Het RAM-geheugen zit in een sub-slot, een bij MSX1 heel ongebruikelijke oplossing. Wat zo'n sub-slot inhoudt komen we straks nog op terug.

Toen MSX net in Europa op de markt verscheen waren er nog niet zoveel verschillende types. De eerste 64K machines die in Europa aankwamen waren de Toshiba HX-10, in Engeland vooral, de Goldstar FC 200 en de Sony HB-75p. In de Benelux was er dan nog de Spectravideo 728.

e eerste fout

Zoals we in de tabel kunnen zien hebben die drie vroege MSX-computers hun RAM in slot 2 zitten. De eerste machinetaal-software die met slots schakelde kwam uit Groot Brittanie, deze programma's maakten vaak gebruik van een routine die plompverloren page 1 in slot 2 schakelde, hetgeen op die eerste MSXen uitstekend funktioneerde. Die page 1 werd daardoor keurig omgeschakeld van de ROM Basic-interpret naar RAM.

Als zo'n programma wordt gedraaid - althans geprobeerd - op een MSX computer met een andere slotindeling, dan gaat het volkomen fout. Dan vindt het programma geen RAM in slot 2.

Dit is bijvoorbeeld het geval bij machines als de Philips V8020 (RAM in slot 3), de SANYO MPC 100 (idem) en de al eerder genoemde Spectravideo 728 (RAM slot 1).

Bekende probleem-programma's zijn onder andere 'Break out' en 'Checkmate' van Toshiba, programma's die duidelijk alleen gemaakt en getest waren op een Toshiba MSX.

Kortom, een 'slordigheidje' van de software-huizen, die zich niet aan de MSX-standaard hielden - die voorschrijft dat een programma zelf naar RAM moet zoeken, maar zeer gemakzuchtig er vanuit gingen dat al die machines wel helemaal gelijk zouden zijn.

De tweede fout

Toen deze nalatigheid tot grote problemen bleek te leiden gingen vele software-huizen - de goede die zich van het begin af aan wel aan de standaard hielden niet te na gesproken - over op andere en inderdaad iets betere routines, waarbij het niet uitmaakte of de 64K RAM in slot 1, 2 of 3 bevond. Deze routines zochten inderdaad zelf het RAM.

Hierbij gingen ze er echter gemakshalve van uit dat de 64K RAM altijd 'recht boven elkaar' in een slot staat, alweer een foute veronderstelling die sommigen nog steeds niet afgeleerd hebben. Het is namelijk eenvoudig om na te gaan in welk slot de RAM in de pages 2 en 3 staat. Dat komt omdat als de computer opgestart is in Basic - dus bijvoorbeeld niet met een cartridge programma - page 0 in slot 0 staat, page 1 ook in slot 0 en de pages 2 en 3 in het slot waar de RAM staat.

Wanneer de computer namelijk opgestart wordt, zoekt het systeem - de BIOS dus - zelf uit in welk slot hij in de pages 2 en

3 RAM kan vinden en schakelt deze dan ook meteen in. Wilde men page 1 ook in RAM plaatsen, dan zette men page 1 domweg in het zelfde slot als page 2 en 3.

Deze oplossing werkt in de meeste gevallen wel, maar er zijn in de praktijk situaties te bedenken waarin het toch allemaal spaak kan lopen. Bijvoorbeeld bij een met een extra RAM-module uitgebreide 16- of 32K RAM MSX.

MSX1 met minder dan 64K RAM

We weten nu wat er met twee van de vier slots gedaan wordt. De beide overige slots - dat zijn de slots waar geen RAM of ROM in staat - doen dienst als de uitbreidings-poorten van de computer. Deze dragen de naam cartridge- of expansion-poort. In enkele gevallen is zo'n intern niet gebruikt slot niet naar buiten gevoerd - bijvoorbeeld bij Yashica YC 64 - en is dan ook niet zomaar te gebruiken. Alleen een handige electronica-knutselaar zou er dan nog wat mee kunnen doen.

Stel, we hebben een 32K RAM machine, met een slotindeling als in figuur 4. Omdat er maar 32K RAM aanwezig is plaatsen de fabrikanten deze gewoon in slot 0, boven de ROM. Bij de Sony HB55p overigens, welke 16K RAM bezit, zit de ingebouwde database in die lege page 2 en in page 3 de RAM. Slot 1 doet dienst als cartridge poort.

Maar verder met onze uitgebreide 32K MSX. We hebben in de computershop een 64K RAM uitbreidings-module ge-

kocht en steken hem in de cartridge gleuf en denken 'ziezo, RAM voldoende, nietwaar? Maar liefst 96K!'

Het programma, dat eerst niet wilde lopen omdat het gemaakt was voor 64K RAM machines, wordt geladen. Laden klaar... Tilt of Reset! Hoe kan dat?

Het antwoord is vrij eenvoudig te geven: de routine die werd toegepast ging er van uit dat de RAM recht onder elkaar stond. 'Ja maar, er staat nu toch 64K RAM onder elkaar?'

Nieuw probleem

Helemaal waar, maar er is nog een probleem bijgekomen dat we nog niet eerder tegen gekomen waren. In deze situatie staan er voor de pages 2 en 3 namelijk twee verschillende slots met RAM klaar. Welke moet het systeem nu met opstarten kiezen?

Het MSX-systeem is zo ontworpen dat het van links naar rechts gaat zoeken naar pages die volledig gevuld zijn met RAM. Vindt de BIOS die 'volle' pages niet, dan neemt hij een onvolledig gevulde. Is ook die niet te vinden, dan schakelt hij in slot 0. Dit geldt zowel voor page 2 als 3, alleen met de aantekening dat als er geen RAM in page 3 zit, het systeem niet opstart.

De reden daarvoor is dat er minimaal 8K RAM bovenin page 3 moet zitten, anders kan de computer de systeem-variabelen - dat zijn geheugenplaatsen waar bijvoorbeeld de regelbreedte wordt bijgehouden - niet kwijt. Overigens is die 8K ook deel van de officiële minimum standaard van MSX1.

fig.4

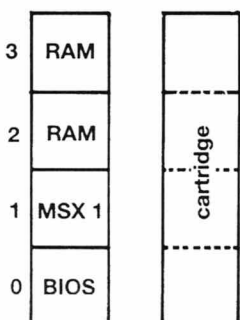
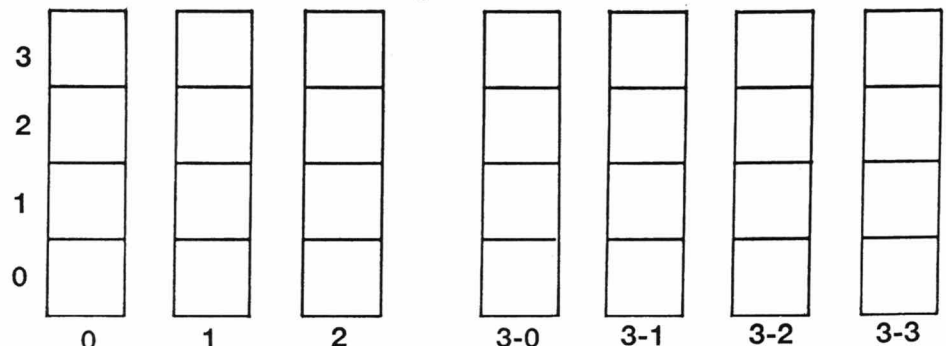


fig.5



Dat zoeken naar RAM doet de BIOS eerst voor page 3, en dan voor page 2. Het systeem zoekt dus van links naar rechts, van slot 0 naar slot 3. Op deze wijze wordt weer gewoon de standaard RAM ingeschakeld en kijkt de slotselect-routine voor page 1 in het zelfde slot als page 2 en 3, die dus gewoon weer in slot 0 staan. Daardoor wordt page 1 ook weer in slot 0 geschakeld en omdat daar de Basic staat zijn we dus even ver als we al waren, ondanks die 64K geheugenuitbreiding. In feite is die uitbreiding door de opstart-routine meteen buiten spel gezet. In een 32K machine met het RAM in dezelfde slot als de ROM zal de machine altijd in precies dezelfde slot-configuratie opstarten, of er nu wel of niet een extra RAM-uitbreiding is.

Dat betekent dat die 'oude' software, die alle RAM in een en dezelfde slot veronderstelt, het dus nog steeds niet zal doen. De computer bevat op dat moment weliswaar 96K RAM, maar de gekozen configuratie houdt in dat er op de pages 0 en 1 van het slot waar van de RAM gekozen is - slot 0 - geen RAM maar ROM bevindt.

Overigens betekent dit ook dat als we de 'extra belegen' software willen gebruiken - die er vanuit gaat dat alle RAM zich in slot 2 bevindt - op een machine met 64K RAM in een van de twee andere slots (1 of 3) we ook voor problemen kunnen komen te staan. Simpelweg een 64K RAM module steken in een cartridge poort die in slot 2 geplaatst is zal niet altijd de oplossing brengen. Die truuik werkt al-

leen als de oorspronkelijke RAM in slot 3 staat en niet als hij in slot 1 staat. Want in dat laatste geval staat de cartridge rechts van de oorspronkelijke RAM, zodat het ingebouwde RAM als eerste gevonden en ingeschakeld wordt. Met andere woorden, de cartridge in slot 2 wordt alweer niet geïnitieerd, de BIOS kiest het RAM in slot 1.

Als het ingebouwde RAM zich in slot 3 bevindt zal het echter wel goed gaan, dan wordt slot 2 - die 64K cartridge - gekozen.

16K machines

Bij een 16K machine als de Sony HB55p ligt het er maar aan of de cartridge-truuik succes oplevert. Page 3 zal gewoon in slot 0 blijven staan, de meest linker, maar page 2 zal wel in de cartridge-RAM staan, want in slot 0 zit geen RAM in page 2. Nu ligt het er maar aan of de routine van de 'oude' software kijkt in welk slot page 3 staat of in welk slot page 2 staat. Plaatst hij page 1 in dezelfde als page 3 dan gaat het weer mis.

Neemt hij daarentegen het slot van page 2 over, dan bestaat er een grote kans dat het programma dan eindelijk wel wil opstarten.

Vele software-fabrikanten hebben voor dit probleem een eenvoudige oplossing bedacht. De pages 2 en 3 blijven gewoon in het slot staan waar ze al stonden en voor page 1 wordt van links naar rechts, van slot 1 tot en met 3 in dit geval, 'gekeken', net zo lang tot dat er RAM wordt gevonden. Is er, na slot 3 gecheckt te hebben,

nog geen succes, dan geeft het programma een boodschap als 'Too little memory', of 'Geen 64 Kb RAM aanwezig'.

Hiermee is echter nog niet alles opgelost, hoewel dergelijke programma's op de meeste machines wel zullen kunnen draaien. Zodra er echter 'expanded slots' om de hoek komen kijken, zoals in bijna alle MSX2 computers en bijvoorbeeld de Mitsubishi ML-FX1, gaat het alweer mis!

Expanded slots

Als er gekeken wordt naar een willekeurige MSX2 machine, dan zien we dat er veel meer geheugen in zit dan in de standaard MSX1 machine. Nemen we een 64K RAM MSX2 met diskdrive als voorbeeld, dan zit daar niet alleen 64K RAM in, maar ook nog eens 64K ROM.

Deze computer heeft daarnaast ook nog eens twee cartridge poorten.

De Slot-indeling van MSX-Computers in Nederland

MSX 1 MERK

AVT Daewoo DPC-200
Canon V20
Goldstar FC 200
JVC HC-7-gb
Mitsubishi MFL-FX1
Mitsubishi MFL-48
Mitsubishi MFL-80
Panasonic CF 2700
Philips VG8020
Philips VG8010
Sanyo MPC-100
Sony HB 201p
Sony HB 75p
Sony HB 55p
Sony HB 10p
Sony HB 501p
Spectravideo 738

Spectravideo 728
Toshiba HX-10
Yamaha CX5M
Yashica YC-64

MSX 2 MERK

AVT Daewoo CPC-300

Sony HB-F500P
Sony HB-F700P

Sony HB-F900P

Sony HB-F9P

Philips VG8220

Philips VG8230
Philips VG8235

Philips VG8250

Philips VG8280

RAM- OPMERKINGEN SLOT

1	
3	
2	
2	
3-2	Slot 3 geëxpandeerd, 64Kb RAM
0*	32Kb RAM
1	
1	
3	
0	32Kb RAM, slot 2 onbruikbaar
3	
3	16Kb ROM firmware in slot 0
2	16Kb ROM firmware in slot 0
0	16Kb RAM, 16Kb ROM firmware in slot 0
3	
3	
1	Slot 3 geëxpandeerd, RS232/ Diskrom
1	
2	
0*	32Kb RAM
3	Slot 1 onbruikbaar

RAM- OPMERKINGEN SLOT

0-2	Slot 0 geëxpandeerd. 128Kb Memory mapper
0-0-2	Slot 0 geëxpandeerd
3-3	Slot 3 geëxpandeerd. 256Kb Memory mapper
0-0-2	Slot 0 geëxpandeerd. Video-digitizer
3-2	Slot 3 geëxpandeerd. 128Kb Memory mapper
3-2*	16Kb ROM firmware Slot 3 geëxpandeerd. 16Kb ROM firmware
3-2	Slot 3 geëxpandeerd
3-2	Slot 3 geëxpandeerd. 128Kb Memory mapper
3-2*	Slot 3 geëxpandeerd. 128Kb Memory mapper
3-2*	Slot 3 geëxpandeerd. 128Kb Memory mapper Video-digitizer

* betekent dat RAM-slot niet met zekerheid vastgesteld is.



De 64K ROM bestaat uit de BIOS (page 0), de MSX1-ROM (page 1), de MSX2-ROM (page 0) en de Disk-ROM (page 1). Dus als we nu het aantal slots gaan berekenen dat minimaal in deze machine aanwezig moet zijn, krijgen we:

een slot voor de BIOS en de MSX1-ROM;
 een slot voor de MSX2-ROM en de DISK-ROM;
 een slot voor de 64K RAM
 en nog twee sloten voor de cartridge poorten.

Dat maakt samen vijf en dat is volgens het model van de standaard MSX1 er een teveel.

Het slot-model bij de standaard MSX1 is echter niet volledig. De MSX-norm houdt namelijk de mogelijkheid open om een slot uit te breiden naar vier slots. Als dat met elk slot zou gebeuren, zou dat leiden tot een MSX computer met 16 slots. Daarom is het ook mogelijk om in theorie een MSX machine te maken met 15 cartridge poorten. Deze techniek wordt 'expanderen' genoemd.

'Slot 3 geëxpandeerd' betekent dan ook dat we met een MSX te maken hebben met de vier slots 0, 1, 2 en 3 - welke hier nu primaire slots worden genoemd - waarvan de laatste is gesplitst in de 'secundaire'

slots 3-0, 3-1, 3-2 en 3-3. In totaal dus zeven slots, zie figuur 5.

Bij elke MSX2 is er minstens een slot geëxpandeerd. Ook komen er bij enkele MSX1 computers geëxpandeerde slots voor, onder andere bij de Spectravideo X'press en de Mitsubishi ML-FX1.

Secundaire slots kiezen

Deze geëxpandeerde slots worden op dezelfde manier aangesproken als de gewone primaire slots, alleen moet daarbij bepaald worden hoe dat geëxpandeerde slot geschakeld staat.

Laten we dat weer eens aan de hand van een voorbeeld gaan bekijken. In figuur 6 staat de situatie geschetst van de Philips VG8235. Voor de overzichtelijkheid zijn de slots 1 en 2, welke de cartridge-poorten zijn, weggelaten. In slot 0 is niets aan de hand, gewoon de BIOS en de MSX1-ROM. Slot 3 is geëxpandeerd. Daarin bevindt zich de RAM (slot 3-2), de Disk-ROM (slot 3-3) en de MSX2-ROM (3-0), welke voor de specifieke MSX2-Basic bevelen zorgt.

Stel, we willen nu page 1 weer in RAM hebben staan. De computer is opgestart met page 0 en 1 in slot 0 en page 2 en 3 in slot 3. Daarvoor echter is de verdeling in secundaire slo-

ten voor slot 3 al ingesteld. Page 3 en 2 van slot 3 staan in slot 3-2, page 1 van slot 3 in slot 3-3 (de Disk-ROM) en page 0 van slot 3 in slot 3-0 (de MSX2-ROM).

Schakelt nu de routine page 1 zonder meer in slot 3 - waar hij veronderstelt RAM te vinden - dan vindt hij daar geen RAM. Dit komt omdat page 1 in slot 3 niet in slot 3-2 staat maar in 3-3.

Waar de computer dus geen RAM vindt maar de Disk-ROM aantreft.

Toch staat in deze situatie de RAM ook recht onder elkaar, dus er valt wel weer een routine te maken welke uitzoekt in welk secundair slot page 2 en 3 staan en deze te kopiëren voor de instelling van page 1, zoals we dat al eerder gezien hebben voor niet-geëxpandeerde machines.

Risikant

Ook hier is echter weer het gevaar aanwezig dat er machines kunnen zijn waar de RAM niet recht onder elkaar staat. Bij de 64K MSX1-computers komt dat probleem niet voor, behalve als we dus een 32K of 16K machine hadden uitgebreid. Maar er bestaan wel zeker MSX2-computers waarbij de standaard ingebouwde 64K RAM niet recht onder elkaar staat. Dit is namelijk het geval bij de Sony HB-F500p, die me-

de daarom tot standaard redactie-computer verheven is. Op die manier komen we op de redactie de meeste grappen en grollen wel tegen.

Bij deze HB-F500p namelijk steekt alle ROM en RAM in slot 0, wat op zichzelf heel handig is, omdat nu slot 1, 2 en 3 helemaal vrij zijn voor cartridge-poorten. De RAM die ter hoogte van page 2 en 3 zit vindt zijn plaats in slot 0-0, terwijl de RAM van page 0 en 1 zich achter de ROM in slot 0-2 bevindt. Het kopiëren van de eventuele secundaire slotindeling van page 2 en 3 lost in dit geval niets op.

De konklusie kan dus niet anders luiden dat de methode om RAM in page 1 te vinden door exact de plaats van page 2 en 3 te kopiëren gewoon niet veilig is. Veel beter is het om een routine te schrijven die gewoon in de computer zoekt in welk slot - of dat nu een primair of een secundair slot is - zich RAM ter hoogte van de gewenste page bevindt.

Memory-mapper

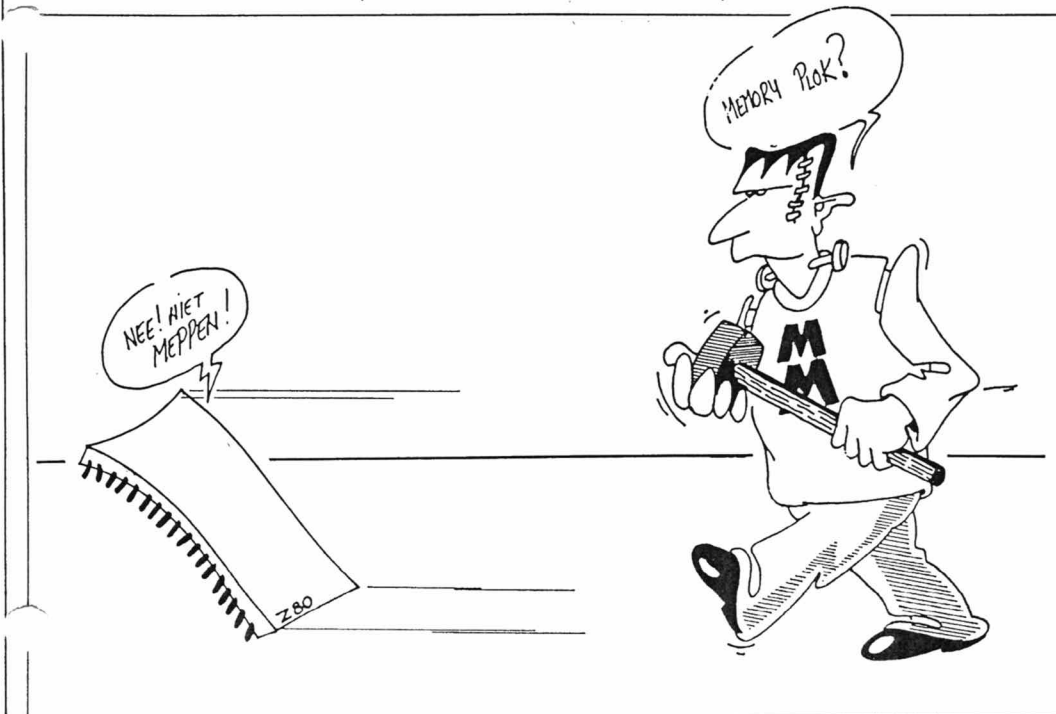
Voordat we gaan kijken hoe we nu door middel van de software deze slots kunnen gaan schakelen eerst nog dit:

In figuur 6 is de geheugenopbouw van de VG8235 schematisch weergegeven, we zien daarin 4 pages met 16K RAM. Maar deze MSX2-computer heeft echter 128K RAM. Rijst de vraag waar dan die tweede 64K gebleven is.

Nu, deze zit ook in slot 3-2. Dit is mogelijk omdat er een aanvulling op de MSX2-norm is gekomen. Dit extra geheugensysteem is de zogenaamde 'Memory Mapper'.

Het in de computer aanwezige RAM, met uitzondering van het videogeheugen, wordt net als bij het slotslect-systeem in blokken van 16K verdeeld. Deze blokken krijgen allemaal een nummer, vanaf 0 tot en met het aantal blokken min 1.

Bij de VG8235 zijn dat dan 8 blokken, want de computer bezit in totaal 128K RAM. De Sony HB F700p heeft er zelfs tweemaal zoveel: 16 blokken en dus 256K RAM.



Het slotslect principe blijft bij een memory-mapper gewoon van kracht, maar het is nu mogelijk om binnen het slot waar de RAM zit steeds - per page - het te gebruiken RAM-blok te kiezen. Dit gebeuren is onafhankelijk van in welk(e) slot(s) de computer geschakeld staat of wordt.

Als we de computer aanzetten, wordt in page 0 van het slot waar de RAM staat, RAM-blok nummer 3 geïnitieerd. In page 1 blok 2, in page 2 blok 1 en in page 3 tenslotte blok 0. Nu kunnen we in de Basic-situatie rustig in page 0 en 1 van het RAM-slot stuuivertje gaan wisselen met die RAM blokken, omdat page 0 en 1 van het RAM-slot op dat moment niet ingeschakeld staan. De enige bijkomstigheid is dan dat de RAM-disk wat in de war kan raken.

Overigens moet er bij dat 'mappen' wel uitgekeken worden dat de systeemvariabelen niet uitgeschakeld worden - laat dus blok 0 gewoon in page 3 staan - en dat we niet onbedoeld twee pages in het zelfde 16K RAM blok zetten. Door zelf te gaan schakelen kunnen we bijvoorbeeld page 2 en page 3 allebei in het zelfde RAM-blok zetten.

Geven we dan POKE &H9000,X dan staat de waarde X ook in RAM op geheugenplaats &HD000. Daar zijn leuke dingen mee mogelijk, maar we moeten wel goed op letten dat de huishouding van het systeem niet de draad kwijtraakt!

Waarom een Memory Mapper

De achterliggende redenen van de ontwerpers van MSX om voor zo'n memory-mapper te kiezen zijn met wat nadenken wel te doorgronden. In eerste instantie lijkt het echter wat dubbel op, want in een enkel geëxpandeerd slot zou toch ook $4 \times 64 = 256K$ kunnen? De extra capaciteit die de memory-mapper dan nog biedt is weliswaar veel en veel groter, maar zal toch niet snel nodig zijn.

Dit is juist, maar zoals al eerder duidelijk geworden is, zijn vooral de pages 0 en 3 moeilijk

te wisselen, omdat de BIOS en de systeem-variabelen daar hun plaatsje hebben. Schakelen met de pages 0 en 3 moet dan ook erg voorzichtig gebeuren en dat gaat vaak ten koste van snelheid en flexibiliteit van het programma.

Met de memory mapper daarentegen zijn we niet aan geheugenadressen gebonden om de verschillende RAM-blokken aan te spreken. Een voorbeeld ter verduidelijking:

In page 0 staat de BIOS.
In pages 1, 2 en 3 staat RAM.

Als we dan het programma in de pages 2 en 3 neerzetten en de data van het programma - denk hierbij bijvoorbeeld aan een tekstverwerker of een database - in page 1, dan kunnen we het gehele geheugen van de memory mapper op de twee blokken waarin het eigenlijke programma staat na - page 2 en 3 - door page 1 laten 'rouleren'. Op deze wijze is er een programma ontstaan die op de VG8235 96K en op de HB-F700p zelfs 224K aan gegevens kwijt kan!

Overigens, de memory mapper kan in theorie 256 blokken bevatten. Dat betekent dan een geheugen van 256 maal 16, dus 4 Megabyte!

De memory mapper is nooit de oorzaak van eventuele MSX-incompatibiliteit, want voor bijna alle bestaande software ziet zo'n memory mapper er als een 64K RAM slot uit. Jammer genoeg zullen de meeste bestaande programma's daardoor ook geen profijt van het extra mapper geheugen kunnen trekken.

Alleen de bijgeleverde programma-pakketten als 'Home Office' en 'Hi-brid' zijn er al op ingesteld. Maar bijvoorbeeld MT-Base, die wel voorbeeldig met primaire en secundaire slots kan werken - althans versie 1.1 - ondersteunt de extra RAM van de memory mapper (nog) niet.

Slotslect en memory mapper in de praktijk

Bij het zelf besturen van de geheugen-indeling van de MSX-computer, door middel van de software, kan er gebruik ge-

maakt worden van zowel Basic als machinetaal. De voorkeur gaat echter naar de laatste uit, om de volgende redenen:

Bij slotslect in Basic moet er buiten de MSX-norm om geprogrammeerd worden.

De schakelmogelijkheden in Basic zijn zeer beperkt ten opzichte van machinetaal.

Volgens de MSX-norm moet de slotslect-procedure namelijk altijd plaatsvinden in de BIOS, iets wat in BASIC niet mogelijk is.

Daarbij is een van de meest gebruikte wisseling van pages juist het uitschakelen van de BASIC-ROM en deze te vervangen door RAM. Dit kan wel in BASIC gebeuren, maar als de computer klaar is met z'n opdracht, kan hij nooit met een volgende Basic-opdracht verder gaan, omdat hij z'n besturing zelf net uitgeschakeld heeft. Resultaat: op tilt of reset.

Om te vertellen hoe het wel moet zullen we eerst eventjes wat over de hardware van de MSX moeten vertellen. In elke MSX-computer die tot nu toe op de markt is verschenen, zit de Intel i-8255 interface-chip.

Deze bouwsteen neemt naast functies zoals het uitlezen van het toetsenbord, ook de primaire slot-selektor voor z'n rekening. De chip heeft drie poorten, te weten A, B en C, waarvan poort A het slotslect signaal afgeeft.

Deze poort A heeft een breedte van 8 bits, welke gesplitst is in vier groepjes van twee bits. De twee 'most-significant' bits - bit 7 en 6 - zijn de bits voor de plaats van page 3. Bit 5 en 4 staan voor page 2, bit 3 en 2 voor page 1 en - logischerwijs - bit 1 en 0 voor page 0. Twee bits tezamen kunnen de waarden 00, 01, 10 en 11 aannemen, de nummers 0 tot en met 3 dus. Deze getallen staan voor de slotnummers. Een voorbeeld:

Page 3 en 2 in slot 2;
Page 1 en 0 in slot 0.

De inhoud van poort A wordt dan binair 10100000, of in de meer gebruikelijke hexadecimale schrijfwijze &HA0.

Poort A is aan te sturen door met Z80 opdracht:

OUT &HA8,data

Daarbij staat die &HA8 voor het poortnummer, waarmee we poort A van de Intel i8255 kunnen aansturen. Om poort A de waarde &HA0 te geven gebruiken we de opdracht:

OUT &HA8,&HA0

Niet alleen in machinetaal bestaat de opdracht 'OUT', ook in Basic hebben we dit kommando tot onze beschikking. De opdracht OUT heeft daarnaast nog een zusje, namelijk de opdracht

IN poortnummer

welke de inhoud van de poort in de accumulator plaatst. Ook hiervan bestaat een Basic-versie:

variabele=INP(poortnummer)

Bij dit alles is echter wel een maar. De MSX-norm schrijft voor, dat de output-poorten - zo worden deze poorten meestal genoemd - niet rechtstreeks mogen worden aangesproken door de opdrachten IN en OUT. Dit is gedaan om er voor te zorgen dat de MSX-compatibiliteit gehandhaafd blijft.

Het is namelijk mogelijk dat er in toekomstige MSX-computers output-poorten van functie veranderen of uitgebreid worden. Niet alleen de slot-selektor zit namelijk aan een output-poort vast, maar alle chips zijn zo met de Z80 verbonden. Om de aansturing van alle chips toch compatibel te kunnen houden is de BIOS ontworpen. Bij de (primaire) slot-selektor betekent dat we - in plaats van rechtstreeks poort &HA8 te lezen of te schrijven - via subroutines in de ROM moeten werken.

Deze routines zijn in dit geval WSLREG (Write SLOtselect REGISTER) en RSLREG (Read SLOtselect REGISTER), welke achtereenvolgens de OUT &HA8 en de IN &HA8 uitvoeren. Wordt nu de slotslect-poort vervangen door een andere, dan hoeft alleen de BIOS aangepast te worden en blijft alle - althans de volgens de norm geschreven - software gewoon werken.

Expanded slots

Ook de secundaire slots moeten volgens de MSX-standaard door middel van de BIOS geselecteerd worden. Vanuit BASIC zijn ze echter ook te kiezen.

Dit gebeurt dan weer buiten de norm om, maar het kan in dit geval zijn nut wel bewijzen, omdat zo veel niet korrekt geschreven software toch op een MSX2 of Mitsubishi ML-FX1 te gebruiken is. De secundaire slots worden niet aangesproken via een output-poort, maar via de geheugen plaats **&HFFF**, welke hier geen RAM, maar een in het geheugen-gebied geplaatst 'schakelregister' is.

Dit 8-bits register werkt op exakt dezelfde wijze als de primaire slot-selektor, met de aantekening dat de groepen van twee-bits hiervoor de expanded slotnummers staan. Een voorbeeld:

Slot 3 is geëxpandeerd;
Page 3 en 2 in slot 3-2;
Page 1 in slot 3-3 en
Page 0 in slot 3-0.

De waarde in **&HFFFF** is dan 10101100, oftewel **&HAC**. Dit is bijvoorbeeld de situatie bij de Philips VG8235 na het opstarten.

Houdt hier echter wel bij in de gaten dat de pages 0 en 1 in slot 3 alleen nog maar 'voor-geselecteerd' zijn en waarschijnlijk niet ingeschakeld, omdat de primaire slot-selektor (zoals bij de Philips VG8235) nog de waarde **&HF0** oftewel binair 11110000 zal hebben.

MSX2 problemen oplossen

Nu is het mogelijk in BASIC het geëxpandeerde slot alvast in de goede positie te brengen door deze helemaal in RAM te schakelen en de computer daardoor in een soort 'MSX1-stand' te brengen. Bij de Philips VG8235 kan dat met:

```
POKE &HFFFF,&HAA
```

of, korter:

```
POKE -1,170
```

Deze pake geldt ook voor de Mitsubishi ML-FX1, de Philips VG8230 en de Sony HB-

F9p. Bij de Sony HB-F700p gebruiken we:

```
POKE &HFFFF,&HFF
```

```
POKE -1,255
```

Deze handigheid kan echter niet toegepast worden bij de Sony HB-F500p, de AVT CPC 300 of andere machines met hun RAM in slot 0, omdat 'voor-selektieren' hier niet mogelijk is. Wordt er bij deze machines een POKE gegeven om de hele RAM voor te selecteren, het zouden POKE **&HFFFF** en **&H0A** en POKE **&HFFFF** en **&HAA** geweest zijn, dan wordt meteen de hele ROM uitgeschakeld, omdat de pages allemaal al in slot 0 geschakeld stonden. Resultaat: alweer tilt of reset.

Uitlezen van **&HFFFF** gaat ook maar dan moeten alle enen en nullen wel omgekeerd worden gelezen. Dit heet ook wel het complement nemen. Bij de Philips VG8235:

```
PEEK(&HFFFF)=&H53=01010011
```

Het complement hiervan is 10101100, oftewel **&HAC**.

Bij het schrijven van echte MSX-software die aan de standaard voldoet moet deze PEEK en POKE-truuk wel achterwege gelaten worden. Maar ook voor het (voor)-selektieren van secundaire slots zijn BIOS-calls voorhanden.

De BIOS-Calls

ENASLT - het staat voor ENABLE SLoT - (voor)-selektiert primaire en/of secundaire slots; WRSLT (WRite SLoT) schrijft data in een primair of secundair slot.

RDSLTL (ReaD SLoT) leest data uit een primair of secundair slot en CALSLT (CALl SLoT) of CALLF (CALL-Far) roepen een subroutine in een ander primair of secundair slot aan.

Deze routines - uitgezonderd de CALL-routines - hebben een adres nodig in het HL-register om een page of adres te kunnen selecteren, de twee 'most-significant' bits van het

HL-register geven het pagenummer aan. In de accumulator wordt aangegeven welk primair en secundair slot gewenst is. Bit 7 is om aan te geven of het slot geëxpandeerd is of niet, 1=wel, 0=niet.

In bit 3 en 2 staan het secundaire slot nummer en in bit 1 en 0 het primaire slot nummer. Bij WRSLT wordt de weg te schrijven data in het E-register geplaatst en de accumulator heeft de data - die gelezen wordt door RDSLTL - opgenomen.

Deze BIOS-routines hebben een vast adres waarmee ze aangeroepen kunnen worden. Een overzicht:

```
Routine Adres
RDSLTL 000C
WRSLTL 0014
CALSLTL 001C
ENASLT 0024
CALLF 0030
RSLREG 0138
WSLREG 013B
```

MSXMEM

Door middel van deze routines is het mogelijk na te gaan waar wel en waar niet RAM aanwezig is. In het programma MSXMEM, dat elders in deze MSX Computer Magazine staat, worden deze routines gebruikt. Ook de verbeterde SUPDIR-routines in het oktobernummer maken gebruik van deze routines.

Bekijk ze goed en let daarbij vooral op twee dingen:

de stack en de geheugenplaatsen **&HFCC1** tot en met **&HFCC9**.

Het blijkt namelijk nodig te zijn om voor het aanroepen van de BIOS routines de register-waarden te behouden voor de rest van het programma.

De geheugenplaatsen **&HFCC1** tot en met **&HFCC5** geven aan of de sloten 0, 1, 2 en 3 eventueel geëxpandeerd zijn. Bij de Philips VG8235 en de Sony HB-F700p is dat slot 3, wat er toe leidt dat het systeem in **&HFCC5** de waarde **&H80** zet. Merk daarbij de analogie op met de inhoud van de accumulator bij het aanroepen van bijvoorbeeld ENASLT.

De waarden van **&HFCC6** tot en met **&HFCC9** geeft de inhoud van het secundaire slotregister aan. Deze heeft echter alleen betekenis als de korresponderende waarde uit **&HFCC1** tot en met **&HFCC5** de waarde **&H80** bezit.

Nogmaals memory mapper

Bij de memory mapper is men afgeweken van de BIOS-routines. Hierbij zijn de output-poorten gestandaardiseerd en dus als vaste MSX2-norm vastgelegd. De memory mapper is overigens alleen een optie bij MSX2 en is niet in elke MSX2-computer te vinden. Voor het besturen van de memory mapper zijn er vier Z80 output-poorten: **&HFF**, **&HFE**, **&HFD** en **&HFC**. Deze schakelen respectievelijk in RAM-page 3, 2, 1 en 0, er is voor elk RAM-page een apart register.

Alle RAM-blokken in de memory mapper hebben een nummer. Bij het opstarten van de computer komt blok 0 in page 3, blok 1 in page 2, blok 2 in page 1 en blok 3 in page 0. De eventuele overgebleven blokken kunnen dan 'met de hand' gewisseld worden met de blokken 0, 1, 2 of 3. Zoals eerder al gezegd luidt het advies: laat page 3 liever ongemoeid, laat blok 0 er maar blijven staan.

Om bijvoorbeeld blok 6 - mits aanwezig natuurlijk, want een memory mapper heeft een minimum configuratie van 4 blokken oftewel 64K - in RAM-page 1 te plaatsen, moet de opdracht

```
OUT &HFD,6
```

gegeven worden. Zowel in Basic als in machinetaal kan de memory aangepast worden naar eigen wens. Uit deze constructie van registers en blokken valt duidelijk te zien hoe het mogelijk is dat het hele 64K RAM gebied in principe in een enkel 16K blok te plaatsen is. Door

```
OUT &HFE,0;
OUT &HFD,0;
OUT &HFC,0;
```

te geven staat de hele 64K geheugen in 16K RAM.

Bij het uitlezen van de outputpoorten &HFF, &HFE, &HFD en &HFC moeten wel even extra voorzorgen worden genomen. Uitlezen gaat als volgt:

waarde=INP(poortnummer)
MOD aantal aanstuurbare blokken

Dit aantal aanstuurbare blokken wordt verkregen door:

INP(&HFF) XOR &HFF

Dit aantal is niet per definitie gelijk aan het aantal aanwezige blokken. Bij de Philips VG8235 is dat wel het geval, maar bij de Sony HB-F700p niet. Deze laatste heeft 16 RAM blokken, maar kan er in principe 32 aansturen. Dit aantal is afhankelijk van hoe de memory mapper is ingebouwd in de computer.

Bij schakelen met RAM en ROM met de memory mapper en de slotselect-procedure in

een programma moet er wel rekening mee gehouden worden dat bij een aanroep van ENASLT de memory mapper weer in z'n beginsituatie wordt teruggezet. Dit is daarentegen niet het geval bij WRSLT en RDSLTL.

Leve de reset!

Diegenen die nu de proef op de som willen nemen mogen wel blij zijn dat er op elke MSX2-computer tegenwoordig standaard een resetknop zit en dat er voor de MSX1-machines die er geen hebben in MCM een methode is gepubliceerd om er zelf een in te zetten.

Want vooral als men begint met slotselecting en de memory mapper worden op de speurtocht naar RAM veel BOKken geschoten. Wees niet verbaasd als daarbij uw computer tijdelijk - gelukkig een reset is genoeg - het loodje legt!

MSX SHOP KEERBERGEN

(15 km van Leuven, 15 km van Mechelen, 15 km van Aarschot)

ONZE DRIE TROEVEN ZIJN: -46 UUR PER WEEK KUNT U BIJ ONS TERECHT

dinsdag 13.00-19.00u
woe. t/m zat. 9.00-12.00u - 13.00-19.00u
zondag 9.00 - 13.00u

-ALS EERSTE MSX SPECIAALZAAK HEBBEN WIJ HET MEEST UITGEBREIDE ASSORTIMENT SOFT- EN HARDWARE IN MSX, TEGEN ZEER VOORDELIGE PRIJZEN.

Hardware: AVT - DAEWOO-GOLDSTAR-PHILIPS
SONY - SPECTRAVIDEO - YAMAHA
Software: meer dan 300 titels van programma's en MSX-boeken van diverse merken in voorraad.

-Verzending in gans België van onze software
Gratis toegestuurd bij vooruitbetaling - Vraag onze prijslijst aan.

Alle HANDY-KAP MSX-beschermkappen te verkrijgen

Gesloten wegens Hemelbeurs te Mechelen van 8-11-'88 t/m 16-11-'88

MSX SHOP
Gemeenteplein 9 - 2850 Keerbergen
BELGIË - Tel.: 015/517529

Dat is gemakkelijk... Een echte MSX--specialist voor software en boeken

Tientallen programma's voor f 9,95 p.st.
zoals: Molecule Man, Speed King,
Oh Shit, Formula 1, enz.

Ook de programma's uit
MSX COMPUTER MAGAZINE
zijn bij ons verkrijgbaar

Vraag de gratis prijslijst
met honderden titels

**TIME
SOFT**

Beukenweg 7
1092 AX Amsterdam
Tel.: 020 - 659393

(Bij het Onze Lieve Vrouwen Gasthuis en het Oosterpark)

SPEEDSAVE 4000 TURBOLADER EN BACK-UP UTILITY

Tot vier maal sneller cassettesaven en -laden.
Laadt eenmaal Speedsave en verander de baudrates in de gewenste snelheid tussen 435 and 4600 baud. Daarna kunt U al Uw programma's saven en laden met de ideale snelheid. Tot 4x sneller, afhankelijk van de recorder en de tape.

Back-up utility voor mcode programma's
Speedsave laadt alle machinetaal programma's, ook zonder header en beneden basic gebied. Met tape headerreader en filesdirectory voor namen, soort programma, adressen en lengte van alle files op disk of tape. Screencopyroutine. Op cassette voor back-ups tot 43K, tape naar tape.

Disk versie met herstel van verwijderde programma's.
Disk versie voor overzetten van mcode programma's tot 40K van tape naar disk. Met automatische aanpassing en slotselectie als het diskgebied overgeschreven wordt. Filesrecovery voor herstel van 'killed' files op disk tot elke lengte. Met sektorreader voor uitlezen van sectoren.

**op cassette f 24,50
op 3.5" disk f 37,50**

Bestel via postgiro 5099419 tnv W.H.Hultink, Bergum
Incl. verzendkosten. Updates f 6,- op originele tape of disk.

ARCKSOFT
De Wylch 26, 9251 PC Bergum