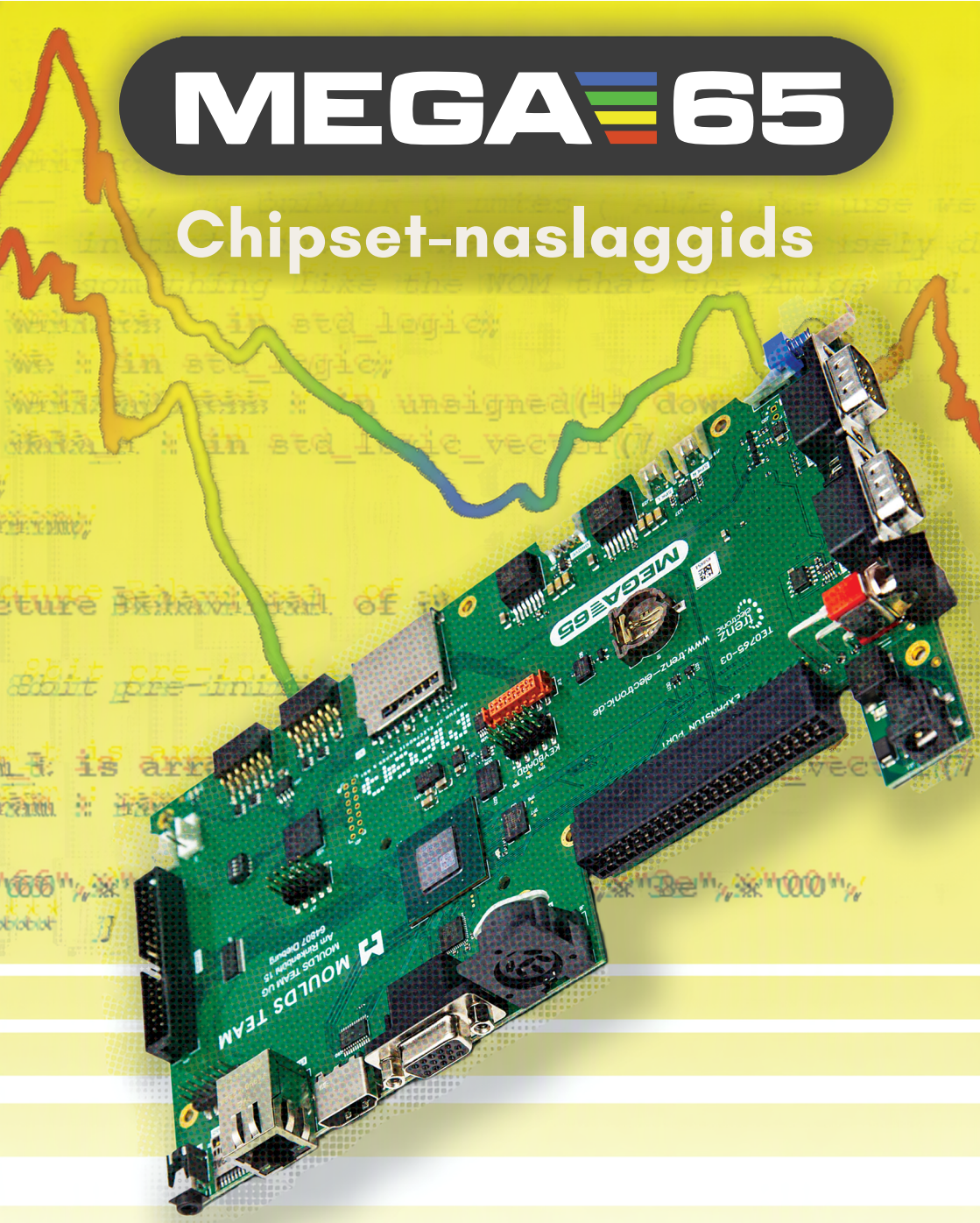
MEGA 65

Chipset-naslaggids



MUSEUM OF ELECTRONIC GAMES & ART

MEGA65-TEAM

Paul Gardner-Stephen, prof.
(highlander)
Oprichter
Software- en virtuele hardwarearchitect
Woordvoerder en hoofdwetenschapper

Martin Streit
(seriously)
Video- en fotoproductie
Fiscaal en organisatie
Sociale media

Dan Sanderson
(dddaaannn)
Media en documentatie
MEGA65.ROM

Edilbert Kirk, dr.
(Bit Shifter)
MEGA65.ROM
Handleiding en gereedschappen

Gábor Lénárt
(LGB)
Emulator

Farai Aschwanden
(Tayger)
Bestands-hosting en gereedschappen
Financieel adviseur

Falk Rehwagen
(bluewaysw)
GEOS

Robert Steffens
(kibo)
Netwerktechnologie
Jacht op bugs in onze cores

Detlef Hastik
(deft)
Mede-oprichter
Projectaansturing
Marketing en verkoop

Oliver Graf
(lydon)
Releasemanagement
VHDL- en platformverbeteringen

Antti Lukats
(antti-brain)
Host-hardwareontwerp en -productie

Dieter Penner
(doubleflash)
Host-hardwareondersteuning

Mirko H.
(sy2002)
Aanvullende platforms en advies

Gürçe Işıkyıldız
(gurce)
Gereedschappen en verbeteringen

Daniel England
(Mew Pokémon)
Aanvullende code en gereedschappen

Hernán Di Pietro
(indiocolifa)
Aanvullende emulatie en gereedschappen

Roman Standzikowski
(FeralChild)
Open ROMs

Anton Schneider-Michallek
(adtbm)
Presentatie en ondersteuning

Fouten en onvolkomenheden melden

Dit boek wordt voortdurend verbeterd door de MEGA65-gemeenschap. De versie van deze editie is:

```
commit c598652ac7f860872b8be62911f59ad5ad752169
date: Sat Oct 24 17:02:30 2026 +0200
```

We willen dat dit boek zo goed mogelijk wordt. Als u fouten ontdekt, iets mist of graag meer informatie wilt, meld dit dan via het issue-tracker van de MEGA65 User's Guide:

https://github.com/mega65/mega65-user-guide/issues

U kunt daar ook controleren of iemand anders al een vergelijkbaar probleem heeft gemeld, terwijl u wacht op een update van dit boek.

Ten slotte kunt u altijd de nieuwste versies van onze boekenserie downloaden via:

- https://mega65.org/mega65-book
- https://mega65.org/user-guide
- https://mega65.org/developer-guide
- https://mega65.org/basic65-ref
- https://mega65.org/chipset-ref
- https://mega65.org/docs

MEGA65 CHIPSET NASLAGGIDS

Uitgegeven door
het MEGA Museum of Electronic Games & Art e.V., Duitsland.

IN AANBOUW

Copyright ©2019 - 2026 Paul Gardner-Stephen, het MEGA Museum of Electronic Games & Art e.V. en medewerkers.

Dit boek is beschikbaar onder de GNU Free Documentation License v1.3 of latere versie. Dit betekent dat u dit boek vrij mag wijzigen, vermenigvuldigen en herdistribueren, onder bepaalde voorwaarden. De volledige tekst van de GNU Free Documentation License v1.3 is te vinden op https://www.gnu.org/licenses/fdl-1.3.en.html. Een Nederlandse vertaling van de licentie is te vinden op https://meta.wikimedia.org/wiki/GNU_Free_Documentation_License/nl.

Deze copyrightlicentie omvat ook de toestemming om dit document geheel of gedeeltelijk te kopiëren en/of te verspreiden voor gebruik in onderwijsomgevingen. Wij willen het onderwijs van toekomstige generaties ondersteunen, dus als u vragen of bezwaren heeft, neem dan contact met ons op.

oktober 29, 2026

Inhoudsopgave

1 Systeemgeheugenindeling 1

Inleiding . . . 3

Eigen geheugenindeling van de MEGA65 . . . 4

De eerste zestien 64KB-banken . . . 4

Kleurengeheugen . . . 5

Extra RAM . . . 5

28-bits adresruimte . . . 5

$D000 - $DFFF I/O-persoonlijkheden . . . 7

Banking van het CPU-geheugen . . . 10

C64/C65-ROM-emulatie . . . 11

Indeling van de C65-compatibiliteits-ROM . . . 12

2 VIC-IV Video Interface Controller 15

Kenmerken . . . 19

Toegang tot VIC-III/IV-registers . . . 20

VIC-II/III/IV detecteren . . . 21

Video-uitvoerformaten, timing en compatibiliteit . . . 23

Integrated Marvellous Digital Hookup™ (IMDH™) Digitale video-uitvoer 23

Aansluiten op eigenzinnige propriëtaire digitale videostandaarden . . . 23

Frametiming . . . 24

Fysieke en logische rasters . . . 27

Bad lines . . . 27

Geheugeninterface . . . 28

Schermgeheugen lokaliseren en verplaatsen . . . 28

Tekengeneratorgegevens verplaatsen . . . 29

Het kleuren-/attribuutgeheugen verplaatsen . . . 29

Spritepointers en sprite-afbeeldingen verplaatsen . . . 30

Hete registers . . . 31

Nieuwe modi . . . 34

Waarom de nieuwe VIC-IV-modi teken- en bitmapmodi zijn en geen bitplanemodi . . . 34

Weergeven van meer dan 256 unieke tekens via “Super-Uitgebreide Attribuutmodus” . . . 35
Bitvelden indien GOTOX_flag is gewist: . . . 38
Bitvelden indien GOTOX_flag is ingesteld: . . . 39
Super-Uitgebreide Attribuutmodus gebruiken . . . 41
Volkleur-tekstmodus (256 kleuren per teken) (FCM) . . . 45
Nibble-kleuren-tekstmodus (16 kleuren per teken) (NCM) . . . 45
Alphablending / Anti-aliasing . . . 46
Tekens omdraaien . . . 46
Variabele-breedte lettertypen . . . 46
Raster Re-write Buffer . . . 47
Sprites . . . 48
VIC-II/III-spritebesturing . . . 48
Uitgebreide sprite-afbeeldingssets . . . 48
Variabele spritegrootte . . . 49
Variabele sprite-resolutie . . . 49
Sprite-paletbank . . . 50
Vollekleuren-spritemodus . . . 50
VIC-III errata-niveau . . . 53
VIC-III errata-niveaus . . . 53
VIC-II / C64-registers . . . 54
VIC-III / C65-registers . . . 57
VIC-IV / MEGA65-specifieke registers . . . 59

3 Sound Interface Device (SID) 65
SID-registers . . . 67

4 F018-compatibele controller voor Direct Memory Access (DMA) 71
F018A/B-DMA-taken . . . 73
Indeling van de F018-DMA-taaklijst . . . 74
Structuur van de 11-byte F018-DMA-lijst . . . 75
Structuur van de 12-byte F018B-DMA-lijst . . . 75

Eenvoudige DMA-bewerkingen uitvoeren 76
MEGA65 Enhanced DMA Jobs 81
Textuurschaling en lijntekening 84
Inline-DMA-lijsten 86
Audio-DMA 87
Adresbeheer voor samples 87
Afspeelfrequentie en volume van samples 88
Zuivere sinusgolf 88
Regeling van het afspelen van samples 89
F018 "DMAgic" DMA-controller 89
Uitbreidingen van de MEGA65-DMA-controller 90
Niet-geïmplementeerde functionaliteit 94

5 6526 Complex Interface Adapter (CIA)-registers 95
CIA 6526-registers 97
CIA 6526-hypervisorregisters 100

6 4551 UART, GPIO en utiliteitscontroller 103
C65 6551 UART-registers 105
Algemene I/O & diverse interface-registers van de 4551 106

7 45E100 Fast Ethernet-controller 111
Overzicht 113
Verschillen met de RR-NET en vergelijkbare oplossingen 113
Werking: het ontvangen van frames 114
Toegang tot de Ethernet-framebuffers 116
Werking: het verzenden van frames 117
Geavanceerde functies 117
Broadcast- en multicast-verkeer en promiscue modus 117
Foutopsporings- en diagnosemogelijkheden 118
In het geheugen gemapte registers 118
Waarden van het COMMAND-register 120

Voorbeeldprogramma's 121

8 45IO27 multifunctionele I/O-controller 123

Overzicht 125
F011-compatibele schijfstationcontroller 125
Ondersteuning voor meerdere schijfstations 125
Bewerkingen met gebufferde sectoren 126
Sectoren van een schijf lezen 127
Automatische spoorcorrectie 127
Sector-scheefstand en de Target Any-modus 128
Schijfindeling en logische 1581-sectoren 128
FD2000-diskettes 129
HD-diskettes en diskettes met variabele dichtheid 129
Track Information Blocks 130
Schijven formatteren 131
Schrijfvoorcompensatie 132
Gebufferd sectorschrijven 132
Floppy-track-DMA 133
Floppy-track-DMA gebruiken 133
De beperkingen van schijfstations begrijpen 134
F011-registers van de schijfstationcontroller 135
SD-kaartcontroller en F011-virtualisatiefuncties 137
Toegang tot schijfimages op basis van de SD-kaart 137
F011-virtualisatie 139
Dual-bus SD-kaartcontroller 140
Write-gate 141
Vulmodus 141
Selectie tussen meerdere SD-kaarten 141
Commandotabel van de SD-controller 141
Touchpadinterface 144
Ondersteunende functies voor audio 146

Overige audiofuncties 148
Mixer-feedbackregisters 148
8/16-bits stereo digitale audio 148
Pulsbreedte versus pulsdichtheidsmodulatie 149
Diverse I/O-functies 150

9 4541 seriële buscontroller 151

Overzicht 153
Mogelijkheden van de 4541 153
Ondersteuning van versnelde seriële protocolvarianten 153
Processorontlasting via interruptondersteuning 153
Onafhankelijkheid van processorsnelheid 153
Coëxistentie via open-collectorlogica 153
Werkingsprincipe 154
Voorbeelden 155
De status van het DOS-kanaal lezen 155
Commandoreferentie 157
Registertabel 161
Seriële bustiming 162
Een byte onder Attention verzenden 162
JiffyDOS™-protocolaanvraag 165
Verzenden van controller naar apparaat onder JiffyDOS™ 167
Ontvangst van randapparaat naar controller onder JiffyDOS™ 169
Omschakelen van spreker naar luisteraar 171
Byte verzenden met einde-aanduiding (EOI) 173
Byte ontvangen 176
Optionele geïntegreerde datalogger 176
Gegevens uit de datalogger halen 178

10 Registers van de real-time klok (RTC) 181
Algemene RTC-registers (alle revisies) 183
Uitgebreide RTC-registers (moederbord R4 en nieuwer) 184
Registers op de Grove-RTC-vervangingsunit 188

11 Speciale registers 191
UUID64 193

12 Naslagtabellen 195
Opslageenheden 197
Getalstelselomrekening 198

13 Steuners & donateurs 203
Organisaties 205
Bijdragers 206
Steuners 207

Index 217

HOOFDSTUK 1

Systeemgeheugenindeling

- Inleiding
- Eigen geheugenindeling van de MEGA65
- $D000 – $DFFF I/O-persoonlijkheden
- Banking van het CPU-geheugen
- C64/C65-ROM-emulatie

INLEIDING

De MEGA65-computer heeft een grote 28-bits adresruimte, waarmee tot 256 MB aan geheugen en in het geheugen gemapte apparaten kunnen worden aangesproken. Deze geheugenindeling kent verschillende verschijningsvormen, afhankelijk van de modus waarin de computer draait. In grove lijnen zijn er vijf hoofdmodi: (1) hypervisormodus; (2) C64-compatibiliteitsmodus; (3) C65-compatibiliteitsmodus; (4) UltiMAX-compatibiliteitsmodus; en (5) MEGA65-modus, of een van de andere modi waarbij de programmeur gebruikmaakt van de uitgebreide functies van de MEGA65.

Het is belangrijk te begrijpen dat de C65 en MEGA65, in tegenstelling tot de C128, toegang bieden tot alle uitgebreide functies vanuit C64-modus, indien de programmeur dat wenst. Dit betekent dat wij, hoewel wij vaak over "C64-modus," "C65-modus" en "MEGA65-modus" spreken, eenvoudigweg gemaksnamen gebruiken voor de MEGA65 waarvan de geheugenindeling (en soms andere functies) zodanig is ingesteld dat zij een omgeving biedt die overeenkomt met de desbetreffende modus. De kern hiervan is de flexibele geheugenindeling van de MEGA65.

In deze bijlage beschrijven wij eerst de eigen geheugenindeling van de MEGA65, dat wil zeggen waar al het geheugen, de I/O-apparaten en andere functies hun plaats hebben in de 28-bits adresruimte. Vervolgens leggen wij uit hoe C64- en C65-compatibele geheugenindelingen vanuit deze 28-bits adresruimte worden benaderd.

EIGEN GEHEUGENINDELING VAN DE MEGA65

De eerste zestien 64KB-banken

De MEGA65 gebruikt voor het eerste MB geheugen een soortgelijke geheugenindeling als de C65, te weten 16 geheugenbanken van 64 KB elk. Dit komt doordat de 4510-CPU van de C65 slechts 1 MB adresruimte kan benaderen. Deze banken kunnen vanuit BASIC 65 worden benaderd met de commando's **BANK**, **DMA**, **PEEK** en **POKE**. De volgende tabel vat de inhoud van de eerste 16 banken samen:

HEX	DEC	Adres	Inhoud
0	0	$0xxxx	Eerste 64KB RAM. Dit is het RAM-geheugen dat in de C64-modus zichtbaar is.
1	1	$1xxxx	Tweede 64KB RAM. Dit is het tweede 64KB RAM-geheugen dat op een C65 aanwezig is.
2	2	$2xxxx	Eerste helft van de C65-ROM (C64-modus en gedeelde onderdelen) *of* RAM
3	3	$3xxxx	Tweede helft van de C65-ROM (C65-modus-onderdelen) *of* RAM
4	4	$4xxxx	Extra RAM (384KB of grotere chip-RAM-modellen)
5	5	$5xxxx	Extra RAM (384KB of grotere chip-RAM-modellen)
6	6	$6xxxx	Extra RAM (*512KB of grotere chip-RAM-modellen)
7	7	$7xxxx	Extra RAM (*512KB of grotere chip-RAM-modellen)
8	8	$8xxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
9	9	$9xxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
A	10	$Axxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
B	11	$Bxxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
C	12	$Cxxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
D	13	$Dxxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
E	14	$Exxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)
F	15	$Fxxxx	Extra RAM (*1MB of grotere chip-RAM-modellen)

* Let op: de MEGA65 biedt momenteel alleen een model met 384KB chip-RAM. Toekomstige modellen kunnen grotere hoeveelheden chip-RAM bevatten (zoals 512KB en 1MB).

De belangrijkste kenmerken van deze adresruimte zijn de 128KB RAM in de eerste twee banken, die ook op de C65 aanwezig is. Indien u programma's wilt schrijven die ook op een C65 kunnen draaien, dient u alleen deze twee banken RAM te gebruiken.

Op alle modellen is het mogelijk om alle of een deel van de 128KB "ROM"-ruimte als RAM te gebruiken. Daartoe moet u eerst de Hypervisor verzoeken de schrijfbeveiliging van dit gebied op te heffen, voordat u de inhoud kunt wijzigen. Indien u een

programma schrijft dat vanuit C64-modus start, of anderszins overschakelt naar het C64-deel van de ROM in plaats van het C65-deel, dan kan de tweede helft van die ruimte, te weten BANK 3, veilig voor uw programma's worden gebruikt. Dit geeft een totaal van 192KB RAM, dat op alle modellen van de MEGA65 beschikbaar is.

Op modellen met 384KB of meer chip-RAM zijn BANK 4 en 5 eveneens beschikbaar. Modellen die 1MB of meer chip-RAM bieden, hebben daarnaast BANK 6 tot en met 15 beschikbaar, wat een totaal van 896KB (of 960KB, indien alleen het C64-deel van de ROM nodig is) RAM oplevert voor uw programma's. Let op: de ingebouwde freeze-cartridge van de MEGA65 bevriest momenteel alleen de eerste 384KB RAM.

Kleurengeheugen

De VIC-IV-videokaart van de MEGA65 ondersteunt veel grotere schermen dan de VIC-II of VIC-III. Zij heeft daarom toegang tot een apart kleurengeheugen (colour-RAM), vergelijkbaar met dat van de C64. Voor compatibiliteit met de C65 zijn de eerste twee kilobytes hiervan toegankelijk op $1F800 - $1FFFF. Het volledige 32KB kleurengeheugen bevindt zich op $FF80000 - $FF87FFF. Dit is het eenvoudigst toegankelijk via geavanceerde DMA-bewerkingen of de 32-bits indirecte adresseermethode op de basispagina (zero page) van de processor.

Op het moment van schrijven kunnen de **BANK**- en **DMA**-commando's niet worden gebruikt om het resterende kleurengeheugen te bereiken, omdat het kleurengeheugen zich niet in het eerste megabyte van de adresruimte bevindt. Dit kan in een toekomstige revisie van de MEGA65 worden gecorrigeerd, waardoor toegang tot het volledige kleurengeheugen via BANK 15 of een equivalente DMA-taak mogelijk wordt.

Extra RAM

Naast de 384KB chip-RAM die standaard op alle MEGA65-modellen aanwezig is, hebben de meeste modellen (devkit, release-borden en xemu, maar momenteel NIET op Nexys-borden) ook 8MB extra RAM vanaf $8000000, aangeduid als 'ATTIC RAM'. Dit is niet zichtbaar voor de andere chips (vic/sid/enz.) en kan niet worden gebruikt voor audio-DMA, maar er kan code op worden uitgevoerd (weliswaar langzamer) of het kan worden gebruikt om inhoud op te slaan en via DMA in en uit het chip-RAM te verplaatsen.

Er zijn ook plannen om een PMOD hyperRAM-module te ondersteunen (geïnstalleerd via het klepje onder de MEGA65) om nog eens 8MB RAM vanaf $8800000 te bieden, aangeduid als 'CELLAR RAM'.

28-bits adresruimte

Naast de C65-stijl 1MB adresruimte breidt de MEGA65 deze uit tot 256MB met 28-bits adressen. De volgende tabel toont de indeling op hoofdlijnen van deze adresruimte.

HEX	DEC	Grootte	Inhoud
0000000	0	1	Richtingsregister van de CPU-I/O-poortgegevens
0000001	1	1	CPU-I/O-poortgegevens
0000002 - 005FFFF	2 - 384KB	384KB	Snel chip-RAM (40MHz)
0060000 - 0FFFFFF	384KB - 16MB	15.6MB	Gereserveerd voor toekomstige chip-RAM-uitbreiding
1000000 - 3FFFFFF	16MB - 64MB	48MB	Gereserveerd
4000000 - 7FFFFFF	64MB - 128MB	64MB	Cartridgepoort en andere apparaten op de langzame bus (1 - 10 MHz)
8000000 - 87FFFFF	128MB - 135MB	8MB	8MB ATTIC-RAM (momenteel op alle modellen behalve Nexys)
8800000 - 8FFFFFF	135MB - 144MB	8MB	8MB CELLAR-RAM (geplande PMOD-module, geïnstalleerd via het klepje)
9000000 - EFFFFFF	144MB - 240MB	96MB	Gereserveerd voor toekomstig uitbreidings-RAM
F000000 - FF7DFFF	240MB - 255.49MB	15.49MB	Gereserveerd voor toekomstige I/O-uitbreiding
FF7E000 - FF7EFFF	255.49MB - 255.49MB	4KB	VIC-IV-teken-ROM (alleen schrijven)
FF80000 - FF87FFF	255.5MB - 255.53MB	32KB	VIC-IV-kleurengeheugen (32KB kleurengeheugen - beschikbaar op alle modellen)
FF88000 - FF8FFFF	255.53MB - 255.57MB	32KB	Gereserveerd
FF90000 - FFCAFFF	255.53MB - 255.80MB	216KB	Gereserveerd
FFCB000 - FFCBFFF	255.80MB - 255.80MB	4KB	Geëmuleerd C1541-RAM
FFCC000 - FFCFFFF	255.80MB - 255.81MB	16KB	Geëmuleerde C1541-ROM
FFD0000 - FFD0FFF	255.81MB - 255.81MB	4KB	C64-$Dxxx-I/O-persoonlijkheid
FFD1000 - FFD1FFF	255.81MB - 255.82MB	4KB	C65-$Dxxx-I/O-persoonlijkheid
FFD2000 - FFD2FFF	255.82MB - 255.82MB	4KB	MEGA65-$Dxxx-Ethernet-I/O-persoonlijkheid
FFD3000 - FFD3FFF	255.82MB - 255.82MB	4KB	MEGA65-$Dxxx-normale-I/O-persoonlijkheid

Vervolgd ...

...Vervolgd

HEX	DEC	Grootte	Inhoud
FFD4000 - FFD5FFF	255.82MB - 255.83MB	8KB	Gereserveerd
FFD6000 - FFD67FF	255.83MB - 255.83MB	2KB	Tijdelijk geheugen voor de hypervisor
FFD6000 - FFD6BFF	255.83MB - 255.83MB	3KB	Tijdelijk geheugen voor de hypervisor
FFD6C00 - FFD6DFF	255.83MB - 255.83MB	512	Sectorbuffer van de F011-diskettecontroller
FFD6E00 - FFD6FFF	255.83MB - 255.83MB	512	Sectorbuffer van de SD-kaartcontroller
FFD7000 - FFD70FF	255.83MB - 255.83MB	256	I2C-apparaten van de MEGAphone r1
FFD7100 - FFD71FF	255.83MB - 255.83MB	256	I2C-apparaten van de MEGA65 r2
FFD7200 - FFD72FF	255.83MB - 255.83MB	256	MEGA65-HDMI-I2C-registers (alleen voor R2 en oudere modellen met de ADV7511-HDMI-driverchip)
FFD7300 - FFD7FFF	255.83MB - 255.84MB	3.25KB	Gereserveerd voor toekomstige I2C-apparaten
FFD8000 - FFDBFFF	255.83MB - 255.86MB	16KB	Hypervisor-ROM (alleen zichtbaar in de hypervisormodus)
FFDC000 - FFDDFFF	255.86MB - 255.87MB	8KB	Gereserveerd voor ROM-uitbreiding van de hypervisormodus
FFDE000 - FFDE7FF	255.87MB - 255.87MB	2KB	Gereserveerd voor uitbreiding van de Ethernet-buffer
FFDE800 - FFDEFFF	255.87MB - 255.87MB	2KB	Ethernet-frameleesbuffer (alleen lezen) en Ethernet-frameschrijfbuffer (alleen schrijven)
FFDF000 - FFDFFFF	255.87MB - 255.87MB	4KB	Virtuele FPGA-registers (alleen op geselecteerde modellen)
FFE0000 - FFFFFFF	255.87MB - 256MB	128KB	Gereserveerd

$D000 – $DFFF I/O-PERSOONLIJKHEDEN

De MEGA65 ondersteunt vier verschillende I/O-persoonlijkheden. Deze worden geselecteerd door de juiste waarden naar het $D02F KEY-register te schrijven, dat in alle

vier de I/O-persoonlijkheden zichtbaar is. In *het grote Mega65-boek*, C64-, C65- en MEGA65-modi (hoofdstuk 12) vindt u meer informatie over het gebruik van het KEY-register.

De volgende tabel toont welke I/O-apparaten in elk van deze I/O-modi zichtbaar zijn, evenals de KEY-registerwaarden die worden gebruikt om de I/O-persoonlijkheid te selecteren.

HEX	C64	C65	MEGA65 ETHERNET	MEGA65
KEY	$00	$A5, $96	$45, $54	$47, $53
$D000 - $D02F	VIC-II	VIC-II	VIC-II	VIC-II
$D030 - $D07F	VIC-II[1]	VIC-III	VIC-III	VIC-III
$D080 - $D08F	VIC-II	F011	F011	F011
$D090 - $D09F	VIC-II	-	SD-kaart	SD-kaart
$D0A0 - $D0FF	VIC-II	RAM EXPAND CONTROL	-	-
$D100 - $D1FF	VIC-II	ROOD PALET	ROOD PALET	ROOD PALET
$D200 - $D2FF	VIC-II	GROEN PALET	GROEN PALET	GROEN PALET
$D300 - $D3FF	VIC-II	BLAUW PALET	BLAUW PALET	BLAUW PALET
$D400 - $D41F	SID rechts #1	SID rechts #1	SID rechts #1	SID rechts #1
$D420 - $D43F	SID rechts #2	SID rechts #2	SID rechts #2	SID rechts #2
$D440 - $D45F	SID links #1	SID links #1	SID links #1	SID links #1
$D460 - $D47F	SID links #2	SID links #2	SID links #2	SID links #2
$D480 - $D49F	SID rechts #1	SID rechts #1	SID rechts #1	SID rechts #1
$D4A0 - $D4BF	SID rechts #2	SID rechts #2	SID rechts #2	SID rechts #2
$D4C0 - $D4DF	SID links #1	SID links #1	SID links #1	SID links #1
$D4E0 - $D4FF	SID links #2	SID links #2	SID links #2	SID links #2
$D500 - $D5FF	SID-kopieën	-	Gereser-veerd	Gereser-veerd
$D600 - $D63F	-	UART	UART	UART
$D640 - $D67F	-	UART-kopieën	HyperTrap-registers	HyperTrap-registers
$D680 - $D6FF	-	-	MEGA65-apparaten	MEGA65-apparaten
$D700 - $D7FF	-	-	MEGA65-apparaten	MEGA65-apparaten
$D800 - $DBFF	KLEURENGE-HEUGEN	KLEURENGE-HEUGEN	ETHERNET-buffer	KLEURENGE-HEUGEN
$DC00 - $DDFF	CIAs	CIAs / KLEURENGE-HEUGEN	ETHERNET-buffer	CIAs / KLEURENGE-HEUGEN
$DE00 - $DFFF	CART I/O	CART I/O	ETHERNET-buffer	CART I/O / SD-sectorbuffer

Vervolgd ...

...Vervolgd

HEX	C64	C65	MEGA65 ETHERNET	MEGA65

[1] In de C64-I/O-persoonlijkheid gedraagt $D030 zich zoals op de C128, waarmee kan worden geschakeld tussen 1MHz en 2MHz CPU-snelheid.
[2] De extra MEGA65-SID's zijn zichtbaar in alle I/O-persoonlijkheden.
[3] Sommige modellen kunnen de herhaalde kopieën van de eerste vier SID's vervangen door vier extra SID's, voor een totaal van 8 SID's.

BANKING VAN HET CPU-GEHEUGEN

De 45GS02-processor kan, net als de 6502, slechts 64 KB geheugen tegelijk "zien". Toegang tot aanvullend geheugen verloopt via een reeks bankswitchingmechanismen. Voor backward-compatibiliteit met de C64 en C65 worden de geheugenbankmechanismen van beide computers ondersteund op de MEGA65:

1. Banking via de MAP-instructie in C65-stijl
2. $D030-banking in C65-stijl
3. Cartridge-banking in C64-stijl
4. $00 / $01-banking in C64-stijl

Het MAP-register overschrijdt alle andere bankmechanismen. Dit mechanisme selecteert welke van de acht 8 KB-gebieden van de 16-bits adresruimte $0000 - $FFFF via een offset op andere adressen worden gemapt. Wanneer een gebied gemapt is, zijn de overige bankmechanismen niet van toepassing. Dit geldt zelfs wanneer de offset 0 is, waardoor de 16-bits adressen toegang krijgen tot het RAM in bank 0 (zoals adres 0.D000).

C65-stijl $D030-banking en C64-stijl $00 / $01-banking selecteren beide gebieden die naar bank 2 worden gemapt, die standaard de C64-ROM-code bevat. Deze twee mechanismen overlappen in de gebieden die zij naar ROM kunnen mappen. Wanneer een van de twee mechanismen een gebied naar ROM mappt (en het niet elders door het MAP-register is gemapt), wordt het naar ROM gemapt.

De volgende figuur toont de verschillende vormen van banking die van toepassing kunnen zijn op de verschillende gebieden van de 64 KB die de CPU kan zien.

MAP	MAP LO (4 x 8KB-gebieden)	MAP HI (4 x 8KB-gebieden)				
I/O/CART		CART ROMLO	CART ROMHI		I/O	CART ROMHI
D030		CHARROM	BASIC	INTER-FACE		KERNAL
C64			BASIC		CHAR ROM	KERNAL
RAM	RAM	RAM	RAM	RAM	RAM	RAM
	$0000 - $7FFF	$8000 - $9FFF	$A000 - $BFFF	$C000 - $CFFF	$D000 - $DFFF	$E000 - $FFFF

Er zijn enkele bijkomende complicaties. Wanneer de cartridge bijvoorbeeld de UltiMAX™-gamemodus selecteert, is slechts de eerste 4 KB RAM zichtbaar en wordt de rest van de adresruimte niet gemapt, zodat deze door de cartridge kan worden aangeboden.

C64/C65-ROM-EMULATIE

De C64 en C65 gebruiken ROM-geheugen om het KERNAL- en BASIC-systeem op te slaan. De MEGA65 werkt anders: zij gebruikt 128 KB van de 384 KB snelle chip-RAM op $20000 - $3FFFF (banken 2 en 3) om deze systeemprogramma's op te slaan. Hierdoor is het mogelijk de "ROM" die de MEGA65 gebruikt te wijzigen of bij te werken zonder de computer te openen. Het is zelfs mogelijk het Freeze-menu van de MEGA65 te gebruiken om de gebruikte "ROM" te wijzigen terwijl een programma draait.

De geheugenbankmethoden van de C64 en C65 gebruiken dit 128 KB-gebied wanneer zij ROM-banken zichtbaar maken. Wanneer de RAM-banken gemapt zijn, zijn zij altijd alleen-lezen. Wanneer echter de MAP-instructie of DMA wordt gebruikt om dat adresgebied te bereiken, is het mogelijk ernaar te schrijven. Voor verbeterde backward-compatibiliteit is het volledige 128 KB-geheugengebied standaard op alleen-lezen ingesteld.

Een programma kan evenwel lees-schrijftoegang tot dit 128 KB-geheugengebied aanvragen, zodat het programma het volledige 384 KB chip-RAM van de MEGA65 kan benutten. Dit wordt gerealiseerd door de *ROM Write-enable* systeemtrap van de hypervisor te activeren. Het volgende codefragment demonstreert hoe dit werkt.

```
LDA #$02
STA $D641
CLV
```

Dit fragment werkt door subfunctie $02 (ROM write-enable) van hypervisortrap $01 aan te roepen. Subfunctie $00 van dezelfde trap schakelt de schrijfbeveiliging weer in. De instructie na de STA moet een instructie met één cyclus zijn, doorgaans CLV: de CPU voert die bij het terugkeren uit de Hypervisor al dan niet uit. NOP wordt niet

aanbevolen, omdat deze op de 45GS02 als prefix voor de volgende instructie kan werken. De MEGA65 I/O-persoonlijkheid moet eerst worden geselecteerd, zodat de hypervisor-trap-registers ($D640 - $D67F) zichtbaar worden.

De huidige schrijfbeveiligingstoestand kan worden getest door te proberen naar dit geheugengebied te schrijven. Daarnaast kunt u de huidige toestand inspecteren en wijzigen in het Freeze-menu van de MEGA65.

Opmerking: wanneer u het programma vanuit C65-modus start, moet u er eerst voor zorgen dat het I/O-gebied zichtbaar is op $D000-$DFFF. De eenvoudigste manier om dit te doen is de MAP-instructie te gebruiken met alle waarden nul in de registers. Het volgende fragment demonstreert dit en zorgt er bovendien voor dat de MEGA65 I/O-context actief is, zodat de hypervisortrap kan worden geactiveerd:

```
; Wis de C65-geheugenindeling
LDA #$00
TAX
TAY
TAZ
MAP
; Bank I/O in via het C64-mechanisme
LDA #$35
STA $01
; Activeer de MEGA65 / VIC-IV-I/O-persoonlijkheid
LDA #$47
STA $D02F
LDA #$53
STA $D02F
; Einde van de MAP-reeks, zodat interrupts weer kunnen optreden
EOM
; Hypervisor-aanroep om de schrijfbeveiliging van het ROM-gebied op te heffen
LDA #$02
STA $D641
CLV
```

Indeling van de C65-compatibiliteits-ROM

De indeling van het 128KB C65-compatibiliteits-ROM-gebied komt overeen met die van de C65:

HEX	Inhoud
$3E000 -- $3FFFF	C65 KERNAL
$3D000 -- $3DFFF	CHARSET B
$3C000 -- $3CFFF	GERESERVEERD
$38000 -- $3BFFF	C65-BASIC-grafische routines
$32000 -- $37FFF	C65-BASIC
$30000 -- $31FFF	MONITOR (wordt op $6000 -- $7FFF gemapt)
$2E000 -- $2FFFF	C64 KERNAL
$2D000 -- $2DFFF	CHARSET C
$2C800 -- $2CFFF	INTERFACE
$2C000 -- $2C7FF	GERESERVEERD
$2A000 -- $2BFFF	C64-BASIC
$29000 -- $29FFF	CHARSET A
$24000 -- $28FFF	GERESERVEERD
$20000 -- $23FFF	DOS (wordt op $8000 -- $BFFF gemapt)

Het INTERFACE-programma is een reeks routines die door de C65 worden gebruikt om te wisselen tussen C64-modus, C65-modus en het ingebouwde DOS van de C65. Het DOS bevindt zich in het onderste achtste van het ROM.

HOOFDSTUK 2

VIC-IV Video Interface Controller

- Kenmerken
- Toegang tot VIC-III/IV-registers
- Video-uitvoerformaten, timing en compatibiliteit
- Geheugeninterface
- Hete registers
- Nieuwe modi
- Sprites
- VIC-III errata-niveau

- VIC-II / C64-registers
- VIC-III / C65-registers
- VIC-IV / MEGA65-specifieke registers

KENMERKEN

De VIC-IV is een vierde generatie Video Interface Controller die speciaal voor de MEGA65 is ontwikkeld en beschikt over een zeer goede backwards compatibility met de VIC-II die in de C64 werd gebruikt en de VIC-III die in de C65 werd gebruikt. De VIC-IV kan geprogrammeerd worden alsof het een van deze voorgangersystemen betreft. Daarnaast ondersteunt het een aantal nieuwe functies. Het is eenvoudig om oudere VIC-II/III-functies te combineren met de nieuwe VIC-IV-functies, waardoor de overstap van de VIC-II of VIC-III naar de VIC-IV net zo eenvoudig is als de overstap van de VIC-II naar de VIC-III destijds. Enkele van de nieuwe functies en verbeteringen van de VIC-IV zijn:

- **Directe toegang tot 384KB RAM** (ten opzichte van 16KB/64KB met de VIC-II en 128KB met de VIC-III).
- Ondersteuning voor **32KB 8-bits kleuren-/attribuutgeheugen** (ten opzichte van 2KB op de VIC-III), om zeer grote schermen te ondersteunen.
- **HDTV 720×576 / 800×600 native resolutie** bij zowel 50 Hz als 60 Hz voor **PAL en NTSC**, met **VGA en digitale video**uitvoer.
- **81 MHz pixelklok** (ten opzichte van ∼ 8 MHz met de VIC-II/III), waardoor een breed scala aan nieuwe functies mogelijk wordt.
- Nieuwe 16-kleuren (16×8 pixels per tekencel) en 256-kleuren (8×8 pixels per tekencel) **volkleur-tekstmodi**.
- Ondersteuning voor maximaal **8.192 unieke tekens in een tekenset**.
- **Vier 256-kleuren paletbanken** (ten opzichte van de enkele paletbank van de VIC-III), die elk een **23-bits kleurdiepte** ondersteunen (ten opzichte van de 12-bits kleurdiepte van de VIC-III) en die snel kunnen worden gewisseld om nog kleurrijkere afbeeldingen te produceren dan mogelijk is met de VIC-III.
- Scherm-, bitmap-, kleur- en tekengegevens kunnen op elk **adres met byte-nauwkeurigheid** worden gepositioneerd (in tegenstelling tot de vaste 1KB - 16KB-grenzen van de VIC-II/III).
- **Virtuele schermafmetingen**, die gecombineerd met byte-nauwkeurige gegevensposities **hardware-ondersteuning voor scrollen en pannen in zowel X- als Y-richting** bieden.
- **Nieuwe sprite-modi**: bitplanewijziging, **volkleuren** (15 voorgrondkleuren + transparantie) en getegelde modi, die een breed scala aan nieuwe en spannende sprite-gebaseerde effecten mogelijk maken.
- De mogelijkheid om sprites stapelvormig in bitplane-vorm te rangschikken om **sprites met maximaal 256 kleuren** te produceren.

- Sprites kunnen 64 bits gegevens per rasterlijn gebruiken, waardoor **sprites 64 pixels breed** kunnen zijn bij gebruik van de VIC-II/III mono-/multicolour-modus, of 16 pixels breed bij gebruik van de nieuwe VIC-IV volkleur-spritemodus.
- **Sprite-tegelmodus**, waarmee een sprite horizontaal kan worden herhaald over de volledige rasterlijn, waardoor sprites kunnen worden gebruikt om geanimeerde achtergronden te creëren op een geheugenefficiënte manier.
- Sprites kunnen worden geconfigureerd om een **apart 256-kleurenpalet** te gebruiken in plaats van het palet dat voor andere tekst en afbeeldingen wordt gebruikt, waardoor een kleurrijker beeld mogelijk is.
- **Super-uitgebreide attribuutmodus** die per tekenmodus twee schermgeheugenbytes en twee kleurengeheugenbytes gebruikt en een breed scala aan nieuwe functies ondersteunt, waaronder **alfa-menging/anti-aliasing**, **hardware-kerning/letters met variabele breedte**, hardwarematig horizontaal/verticaal spiegelen, selectie van een alternatief palet en andere krachtige functies die het gemakkelijk maken om zeer dynamische en kleurrijke weergaven te creëren.
- **Raster-Rewrite Buffer** waarmee **hardwarematig gegenereerde pseudo-sprites** mogelijk zijn, vergelijkbaar met "bobs" op Amiga™-computers, maar met het voordeel dat ze in de weergavepijplijn worden gerenderd en dus niet hoeven te worden gewist en opnieuw getekend om te animeren.
- Ook **verschillende 8-bits kleuren-playfields** zijn mogelijk via de Raster-Rewrite Buffer.

Kortom, de VIC-IV is een krachtige evolutie van de VIC-II/III, waarbij het karakter en de eigenaardigheden van de VIC-reeks videocontrollers behouden zijn gebleven.

Voor een volledige beschrijving van de extra registers die de VIC-IV biedt, alsmede documentatie van de legacy VIC-II- en VIC-III-registers, verwijst u naar de betreffende secties van deze bijlage. Het resterende deel van de bijlage richt zich op het beschrijven van de mogelijkheden en het gebruik van veel van de nieuwe functies van de VIC-IV.

TOEGANG TOT VIC-III/IV-REGISTERS

Omdat de nieuwe functies van de VIC-IV allemaal uitbreidingen zijn op de bestaande VIC-II/III-ontwerpen, bestaat er geen concept van het moeten selecteren van de modus waarin de VIC-IV zal werken: deze bevindt zich altijd in VIC-IV-modus. Voor backwards compatibility met software kunnen de vele extra registers van de VIC-IV echter verborgen worden, zodat de VIC-IV zich voordoet als een VIC-II of VIC-III. Dit gebeurt op dezelfde wijze als waarop de VIC-III zijn nieuwe functies verbergt voor legacy VIC-II-software.

Het mechanisme is het write-only KEY-register van de VIC-III ($D02F, 53295 decimaal). De VIC-III verbergt standaard zijn nieuwe functies totdat een "klop"-sequentie

wordt uitgevoerd. Deze bestaat uit het achtereenvolgens schrijven van twee speciale waarden naar $D02F. De volgende samenvattende tabel toont de ondersteunde klopsequenties van de VIC-IV en geeft aan welke VIC-IV-specifiek zijn en welke ook door de VIC-III worden ondersteund:

Eerste waarde Hex (Decimaal)	**Tweede waarde Hex (Decimaal)**	**Effect**	**VIC-IV-specifiek?**
$00 (0)	$00 (0)	Alleen VIC-II-registers zichtbaar (alle nieuwe VIC-III- en VIC-IV-registers zijn verborgen)	Nee
$A5 (165)	$96 (150)	Nieuwe VIC-III-registers zichtbaar	Nee
$47 (71)	$53 (83)	Zowel nieuwe VIC-III- als VIC-IV-registers zichtbaar	Ja
$45 (69)	$54 (84)	Geen VIC-II/III/IV-registers zichtbaar. In plaats daarvan zijn de buffers van de 45E100-Ethernetcontroller zichtbaar	Ja

VIC-II/III/IV detecteren

Het bepalen van welke generatie VIC-II/III/IV in een computer is geïnstalleerd, kan belangrijk zijn voor programma's die slechts bepaalde generaties ondersteunen of die hun grafische weergave willen laten variëren op basis van de mogelijkheden van de computer. Hoewel er hiervoor veel mogelijkheden bestaan, volgt hier een eenvoudige en effectieve methode. Deze berust op het feit dat de VIC-III en VIC-IV de VIC-II-registers niet door de hele I/O-adresruimte herhalen. Terwijl $D000 en $D100 synoniem zijn wanneer een VIC-II aanwezig is (of een VIC-III/IV zijn extra registers verbergt), is dit niet het geval wanneer een VIC-III of VIC-IV alle registers zichtbaar maakt. De aanwezigheid van een VIC-III/IV kan daarom worden vastgesteld door te testen of deze twee locaties aliasen zijn van hetzelfde register of afzonderlijke registers vertegenwoordigen. De detectiesequentie bestaat uit het gebruik van het KEY-register om extra VIC-IV- of VIC-III-registers zichtbaar te maken. Indien een van beide lukt, kan worden aangenomen dat de overeenkomstige VIC-generatie is geïnstalleerd. Omdat de VIC-IV de VIC-III-klopsequenties ondersteunt, moet eerst op de aanwezigheid van een VIC-IV worden getest. Tevens gaan we ervan uit dat de MEGA65 in VIC-IV-modus start, zelfs wanneer C65 BASIC wordt uitgevoerd. De test kan daarom vanuit BASIC worden uitgevoerd, zowel vanuit C64- als C65-modus als volgt:

```
0 REM IN C65-MODUS KUNNEN WE NIET VEILIG NAAR $D02F SCHRIJVEN,
1 REM DUS TESTEN WE HET OP EEN ANDERE MANIER
10 IF PEEK($D018) AND 32 THEN GOTO 65
20 POKE $D000,1:POKE $D02F,71:POKE $D02F,83
30 POKE $D000+256,0:IF PEEK($D000)=1 THEN PRINT"VIC-IV AANWEZIG":END
40 POKE $D000,1:POKE $D02F,165:POKE $D02F,150
50 POKE $D000+256,0:IF PEEK($D000)=1 THEN PRINT"VIC-III AANWEZIG":END
60 PRINT "VIC-II AANWEZIG":END
65 REM WE GAAN ERVAN UIT DAT WE HIER EEN C65 HEBBEN
70 V1=PEEK($D050):V2=PEEK($D050):V3=PEEK($D050)
80 IF V1<>V2 OR V1<>V3 OR V2<>V3 THEN PRINT "VIC-IV PRESENT":END
90 GOTO 40
```

Regel 10 van dit programma controleert of het scherm een veelvoud van 2KB is. Omdat het scherm op de C64 op 1KB staat, slaagt deze test niet en gaat de uitvoering verder naar regel 20. Regel 20 schrijft 1 naar een van de VIC-II-spritepositieregisters, 53248, en vervolgens de MEGA65-klop naar het KEY-register, 53295. Regel 30 schrijft naar 53248 + 256, wat op de C64 een spiegel is van 53248, maar op een MEGA65 met VIC-IV I/O ingeschakeld een van de rode paletregisters is. Na het schrijven naar 53248 + 256 controleert het programma of het register op 53248 is gewijzigd door de schrijfactie naar 53248 + 256. Indien dat het geval is, verwijzen de twee adressen naar hetzelfde register. Dit gebeurt op zowel een C64 als een C65, maar niet op een computer met een VIC-IV. Wanneer 53248 dus niet is gewijzigd, melden we dat we een VIC-IV hebben gedetecteerd. Indien het schrijven naar 53248 + 256 de waarde in register 53248 wel heeft gewijzigd, gaat het programma verder naar regel 40, die opnieuw naar 53248 schrijft en ditmaal de VIC-III-klop naar het KEY-register schrijft. Regel 50 is gelijk aan regel 30, maar omdat deze na een VIC-III-klop verschijnt, maakt ze de detectie van een VIC-III mogelijk. Indien ten slotte noch een VIC-IV noch een VIC-III wordt gedetecteerd, concluderen we dat er alleen een VIC-II aanwezig moet zijn.

Omdat de MEGA65 de enige C64-klassecomputer is die over een VIC-IV beschikt, kan dit als een *de facto*-test voor de aanwezigheid van een MEGA65-computer worden gebruikt. Detectie van een VIC-III kan op vergelijkbare wijze worden aangenomen dat deze de aanwezigheid van een C65 aanduidt.

VIDEO-UITVOERFORMATEN, TIMING EN COMPATIBILITEIT

Integrated Marvellous Digital Hookup™ (IMDH™) Digitale video-uitvoer

De MEGA65 beschikt over VGA-analoge video-uitvoer en Integrated Marvellous Digital Hookup™ (IMDH™). Dit wijkt in een aantal opzichten af van bestaande gangbare digitale videostandaarden:

1. We hebben er geen nieuwe connector voor bedacht: we hebben in plaats daarvan de meest gebruikte digitale videoconnector gebruikt die al in omloop is. Uw bestaande kabels zouden dus probleemloos moeten werken!
2. We hebben hem niet opzettelijk incompatibel gemaakt met een bestaande digitale videostandaard. Uw bestaande televisies en monitoren zouden dus probleemloos moeten werken!
3. We eisen geen woekerprijs van andere leveranciers voor het gebruik van de IMDH™-digitale videostandaard, door hen jaarlijks $10.000 in rekening te brengen, slechts voor de toestemming om één apparaat te mogen verkopen. Dit betekent dat de MEGA65 goedkoper is voor u!
4. De IMDH™-standaard staat geen inhoudsbescherming of andere soevereiniteitsaantastende ongein toe. Als u de video hebt geproduceerd, mag u ermee doen wat u wilt!

Aansluiten op eigenzinnige propriëtaire digitale videostandaarden

Er bestaan digitale videostandaarden die volkomen achterhaald zijn in vergelijking met IMDH™. Dankzij de open aanpak van IMDH™op het gebied van interoperabiliteit zouden deze in de meeste gevallen echter zonder problemen met de MEGA65 moeten werken. Zoek eenvoudigweg een videokabel die past op de IMDH™-connector op de achterkant van uw MEGA65 en sluit deze aan op uw MEGA65 en een televisie, monitor of projector die over dezelfde connector beschikt.

Helaas hebben niet alle fabrikanten hun apparaten ingediend voor de IMDH™-compliantietest bij het MEGA65-team. Dit betekent dat sommige televisies en monitoren helaas niet IMDH™-conform zijn. Hoewel de meeste televisies en monitoren dus met de MEGA65 zullen werken, zou het kunnen dat u er een paar moet uitproberen om een bevredigend resultaat te bereiken. Indien u een monitor vindt die niet met de MEGA65 werkt, horen we dat graag van u. Meld het probleem tevens bij de monitorfabrikant en beveel hen aan hun apparaten in te dienen voor de IMDH™-compliantietest.

De VIC-IV is ontworpen voor gebruik in de MEGA65 en verwante systemen, waaronder de MEGAphone-familie van draagbare apparaten. De VIC-IV ondersteunt zowel VGA- als digitale video-uitgang via de niet-propriëtaire IMDH™-interface. Daarnaast

ondersteunt de VIC-IV parallelle digitale video-uitgang voor het aansturen van LCD-beeldschermpanelen. Er is veel zorg besteed aan het ontwerpen van een gemeenschappelijke video-voorkant die deze drie uitgangsmodi ondersteunt.

Voor eenvoud en nauwkeurigheid van de frametiming voor bestaande software is het videoformaat doorgaans gebaseerd op de HDTV PAL- en NTSC-modi met 720×576/480 (576p en 480p) en een pixelklok van 27 MHz. Dit formaat is ideaal voor digitale video en LCD-beeldschermpanelen. Niet alle VGA-beeldschermen ondersteunen deze modi echter, met name 720×576 bij 50 Hz.

Met het oog op de achterwaartse compatibiliteit met de VIC-II en VIC-III heeft dit weergaveformaat verschillende gevolgen die voor de meeste programma's geen problemen veroorzaken, maar wel tot enkele gedragsverschillen kunnen leiden:

1. Omdat de VIC-IV-weergave progressief is in plaats van geïnterlinieerd, worden er twee fysieke rasterlijnen geproduceerd voor elke logische VIC-II- of VIC-III-rasterlijn. Dit betekent dat er 63 of 65 cycli per logische dubbele rasterlijn zijn, in plaats van per fysieke 576p/480p-rasterlijn. Dit kan enkele kleine visuele artefacten veroorzaken wanneer programma's aannemen op welk punt op een horizontale lijn de VIC-IV op dat moment tekent, bijvoorbeeld bij het wijzigen van de rand- of schermkleur.

2. De VIC-IV volgt niet het gedrag van de VIC-III, die wijzigingen in videomodi (bijv. tussen tekst- en bitmapmodus) op tekenen toestond, evenmin als de VIC-II-strategie waarbij dergelijke wijzigingen onmiddellijk van kracht werden. De VIC-IV past wijzigingen daarentegen toe aan het begin van elke rasterlijn. Dit kan enkele kleine artefacten veroorzaken.

3. De VIC-IV gebruikt een enkel raster-weergavebuffer die wordt gevuld met behulp van de interne 81 MHz-pixelklok van de VIC-IV, waarna deze wordt weergegeven met de 27 MHz-pixelklok voor de uitgang. Hierdoor wordt de weergave-inhoud van een rasterlijn doorgaans veel eerder in de rasterlijn gerenderd dan op de VIC-II of VIC-III. Dit kan artefacten veroorzaken op beeldschermen, met name in demo's die specifiek gedrag van de VIC-II op bepaalde cycli in een rasterlijn benutten, bijvoorbeeld voor effecten als VSP of FLI. Dergelijke effecten worden op de huidige revisie van de VIC-IV waarschijnlijk niet correct weergegeven. Er is verbeterde ondersteuning voor deze functies gepland voor een toekomstige revisie van de VIC-IV.

4. De 1280×200- en 1280×400-weergavemodi van de VIC-III worden momenteel niet ondersteund, omdat ze niet zinvol op een modern beeldscherm kunnen worden weergegeven en er geen software bekend is die deze functie ondersteunt of gebruikt.

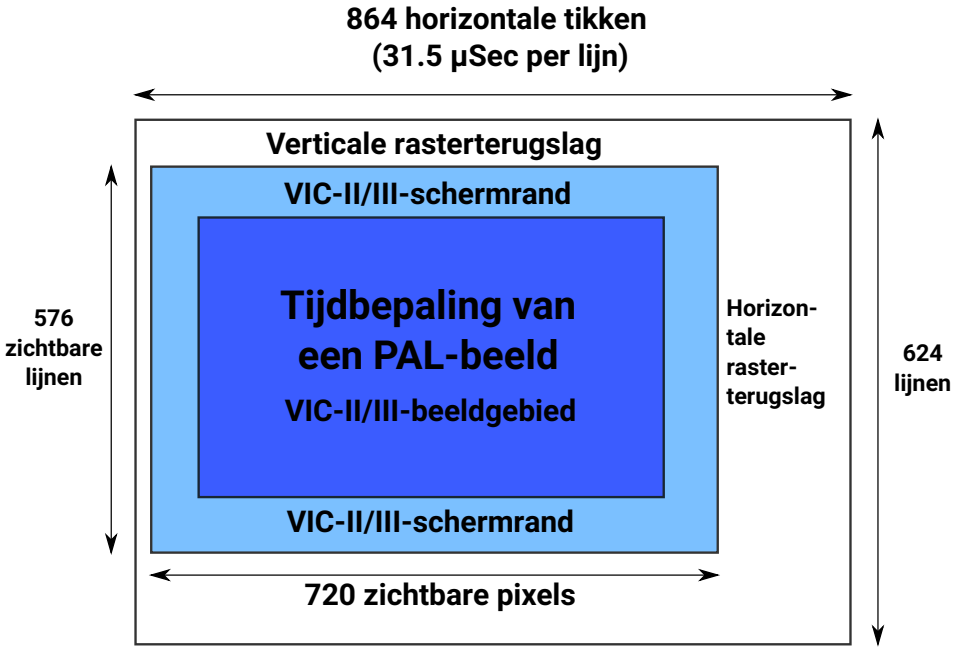
Frametiming

De frametiming is ontworpen om overeen te komen met die van de combinatie 6502 + VIC-II van de C64. Zowel PAL- als NTSC-timing wordt ondersteund, en het aantal cycli per logische rasterlijn, het aantal rasterlijnen per frame en het aantal cycli per frame worden dienovereenkomstig aangepast. Daartoe gebruikt de VIC-IV doorgaans

de HDTV-modi 576p 50 Hz (PAL) en 480p 60 Hz (NTSC), met timing die zo dicht mogelijk bij de dubbelgescande PAL- en NTSC-composiet-tv-modi van de VIC-II wordt afgestemd.

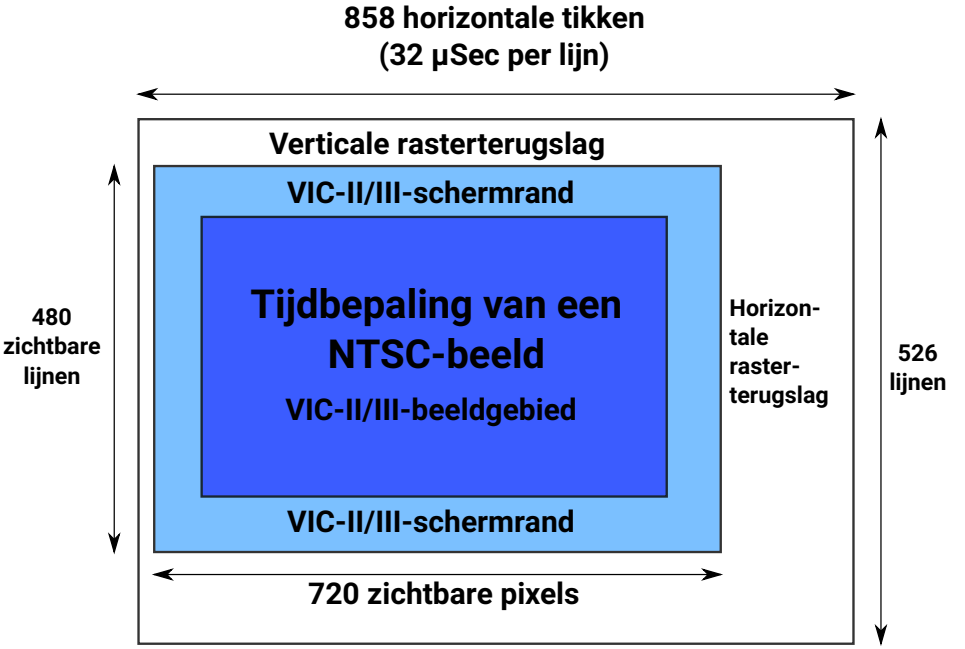
De VIC-IV genereert timingimpulsen van ongeveer 1 MHz die door de 45GS02-processor worden gebruikt, zodat de juiste effectieve frequentie wordt geleverd bij werking in de C64-, C128- en C65-compatibiliteitsmodi met respectievelijk 1 MHz, 2 MHz en 3.5 MHz. Hierdoor kan het enkele systeem overschakelen tussen nauwkeurige PAL- en NTSC-CPU-timing, evenals tussen videomodi. De exacte frequentie varieert tussen PAL- en NTSC-modi om het gedrag van PAL versus NTSC bij de C64-, C128- en C65-processor- en videotiming na te bootsen.

Het PAL-frame bestaat uit 624 fysieke rasterlijnen met elk 864 pixelklokken. De pixelklok is 27 MHz, oftewel 1/3 van de VIC-IV-pixelklok. Het zichtbare frame is 720×576 pixels, waarvan het geheel in de VIC-IV-modus kan worden gebruikt. In de VIC-II- en VIC-III-modi verkleint het randgebied de bruikbare grootte tot 640×400 pixels. In de VIC-II-modus en VIC-III 200H-modi wordt de weergave dubbelgescand, waarbij twee fysieke rasterlijnen van 31.5 microseconden overeenkomen met één VIC-II-stijl rasterlijn van 63 microseconden. Elk frame bestaat dus uit 312 VIC-II-rasterlijnen van elk 63 microseconden, exact gelijk aan die van een PAL-C64.



Het NTSC-frame bestaat uit 526 fysieke rasterlijnen met elk 858 pixelklokken. De pixelklok is 27 MHz, oftewel 1/3 van de VIC-IV-pixelklok. Het zichtbare frame is 720×480 pixels, waarvan het geheel in de VIC-IV-modus kan worden gebruikt. In de VIC-II- en VIC-III-modi verkleint het randgebied de bruikbare grootte tot 640×400

pixels. In de VIC-II-modus en VIC-III 200H-modi wordt de weergave dubbelgescand, waarbij twee fysieke rasterlijnen van 32 microseconden overeenkomen met één VIC-II-stijl rasterlijn van 64 microseconden. Elk frame bestaat dus uit 263 VIC-II-rasterlijnen van elk 64 microseconden, overeenkomstig de meest voorkomende NTSC-videotiming van de C64.



Omdat niet alle VGA-beeldschermen deze HDTV-videomodi ondersteunen, is er een compatibiliteitsmodus beschikbaar die een VGA-stijl modus van 640×480 biedt. Omdat de pixelklok van de MEGA65 vaststaat op 27 MHz, werkt deze modus op 63 Hz. Dit zou op de overgrote meerderheid van VGA-beeldschermen moeten werken. Er zullen geen problemen optreden met de PAL/NTSC-modi bij gebruik van de digitale video-uitgang van de MEGA65 op de overgrote meerderheid van IMDH™-compatibele beeldschermen en televisies.

Om te bepalen of de MEGA65 in PAL- of NTSC-modus werkt, kunt u het Freezer-menu openen, dat de huidige videomodus weergeeft, of vanuit een programma het PALNTSC-signaal controleren (bit 7 van $D06F, decimaal 53359). Als deze bit gezet is, werkt het systeem in NTSC-modus; als deze gewist is, in PAL-modus. Deze bit kan worden gewijzigd om tussen de modi te schakelen, bijv.:

```
10 REM C65+MEGA65-I/O INSCHAKELEN
20 IF PEEK($D018)<32 THEN POKE $D02F,ASC("G"):POKE $D02F,ASC("S")
30 REM NTSC-bit controleren
40 NTSC=PEEK($D06F) AND 128
50 REM STATUS WEERGEVEN EN OM OMSCHAKELING VRAGEN
60 PRINT"MEGA65 IS IN ";:IF NTSC THEN PRINT"NTSC MODE":ELSE PRINT"PAL MODE"
70 INPUT"SWITCH MODES (Y/N)? ",A$
80 REM NTSC-BIT OMSCHAKELEN
90 IF A$="Y" THEN POKE $D06F,PEEK($D06F) XOR 128:ELSE END
100 REM NIEUWE STATUS WEERGEVEN
110 NTSC=PEEK($D06F) AND 128
120 PRINT"MEGA65 IS IN ";:IF NTSC THEN PRINT"NTSC MODE":ELSE PRINT"PAL MODE"
```

Fysieke en logische rasters

Fysieke rasterlijnen per frame verwijzen naar het aantal daadwerkelijke rasterlijnen in de PAL- of NTSC-EDTV-videomodi die door de MEGA65 worden gebruikt. Logische rasterlijnen verwijzen naar het aantal VIC-II-stijl rasterlijnen per frame. Elke logische rasterlijn bestaat uit twee fysieke rasterlijnen per regel, aangezien EDTV-modi dubbelgescande modi zijn vergeleken met de oorspronkelijke PAL- en NTSC-SDTV-modi van de C64. De frame-parameters van de VIC-IV voor PAL en NTSC zijn als volgt:

Standaard	Cycli per rasterlijn	Fysieke rasterlijnen per frame	Logische rasterlijnen per frame
PAL	63	626	312
NTSC	65	526	263

Het resultaat is dat de frames op de VIC-IV exact hetzelfde aantal ~1 MHz-CPU-cycli bevatten als op de VIC-II.

Bad lines

De VIC-IV kent van nature geen "bad lines", omdat de VIC-IV eigen geheugenbussen heeft naar het hoofdgeheugen en het kleurengeheugen (colour RAM) van de MEGA65. Hierdoor kunnen zowel de processor als de VIC-IV tegelijkertijd toegang krijgen tot het geheugen, in tegenstelling tot de C64 of C65, waarbij de toegang wordt afgewisseld.

Om de compatibiliteit te verbeteren, geeft de VIC-IV echter aan wanneer een "bad line" zou hebben plaatsgevonden op de VIC-II. De 45GS02-processor van de MEGA65 aanvaardt deze bad-line-signalen en onderbreekt de CPU gedurende 40 klokcycli, tenzij de processor op volle snelheid draait, in welk geval deze worden genegeerd. Dit verbetert de timingcompatibiliteit met de VIC-II aanzienlijk. De timing is echter niet exact, omdat de huidige revisie van de 45GS02 precies 40 cycli onderbreekt in plaats van 40-43 cycli, afhankelijk van de instructie die op dat moment

wordt uitgevoerd. Daarnaast onderbreken de VIC-IV en 45GS02 momenteel niet voor sprite-ophaaloperaties.

De bad-line-emulatie wordt beheerd door bit 0 van $D710: het zetten van deze bit schakelt bad-line-emulatie in, en het wissen ervan voorkomt dat een bad line tijd van de processor steelt.

GEHEUGENINTERFACE

De VIC-IV ondersteunt tot 16 MB direct toegankelijk RAM voor videogegevens, maar momenteel bieden alle bestaande modellen slechts 384 KB adresseerbaar RAM. In MEGA65-systemen wordt het tweede blok van 128 KB RAM (in het geheugengebied van 128 KB tot 256 KB) doorgaans gebruikt om een C65-compatibele ROM op te slaan, waardoor 256 KB RAM voor de gebruiker beschikbaar blijft. Indien de software zo is geschreven dat het gebruik van C65-ROM-routines niet nodig is, kan de volledige 384 KB RAM door het programma worden gebruikt.

Alle MEGA65-modellen ondersteunen momenteel 32 KB kleurengeheugen (colour RAM), maar er zijn plannen om het nieuwste R3-bord 64 KB kleurengeheugen (of mogelijk zelfs 128 KB) te laten ondersteunen.

De VIC-IV ondersteunt alle methoden van de oudere VIC-II en VIC-III voor de toegang tot dit geheugen, waaronder het gebruik van 16KB-banken door de VIC-II en de Display Address Translator (DAT) van de VIC-III. Dit extra geheugen kan voor teken- en bitmapweergaven worden gebruikt, evenals voor sprites. De bitplanemodi van de VIC-III blijven echter beperkt tot het gebruik van slechts de eerste 128KB van het geheugen, omdat de VIC-IV de bitplanemodus niet uitbreidt.

Schermgeheugen lokaliseren en verplaatsen

De VIC kan worden geconfigureerd om het tekstschermgeheugen overal in de eerste 384KB te zoeken. Uw programma kan bepalen waar het schermgeheugen komt te staan en het startadres in de hardware-registers instellen. Dit adres wordt opgeslagen in de SCRNPTR-registers ($D060 - $D063, 53344 - 53347 decimaal). Om het schermgeheugen bijvoorbeeld op adres 12345 in te stellen:

```
REM C65+MEGA65-I/O INSCHAKELEN
IF PEEK($D018)<32 THEN POKE $D02F,ASC("G"):POKE $D02F,ASC("S")

WPOKE $D060,$2345:POKE $D062,$1
```

Om een bestaand scherm te gebruiken dat door de KERNAL is ingesteld (het scherm dat de gebruiker bij het opstarten ziet), kunt u het startadres uit deze registers lezen. Houd er rekening mee dat de KERNAL in sommige omstandigheden de locatie van het schermgeheugen wijzigt. De beste praktijk is daarom dit adres binnen het programma uit de registers te lezen en de locatie ervan niet te veronderstellen.

```
10 SC=WPEEK($D060)+(PEEK($D062)*65536)
20 POKE SC,1 : REM SCHRIJFT EEN "A" NAAR DE LINKERBOVENHOEK
```

Tekengeneratorgegevens verplaatsen

De locatie van de tekengeneratorgegevens kan eveneens met precisie op byteniveau worden ingesteld via de CHARPTR-registers op $D068 - $D06A (53352 - 53354 decimaal). Zoals gebruikelijk bevat het eerste van deze registers de minst significante byte en het laatste de meest significante byte. De drie bytes maken plaatsing van tekengeneratorgegevens overal in de eerste 16MB van het geheugen mogelijk. Voor systemen met minder dan 16MB geheugen waartoe de VIC-IV toegang heeft, dienen de bovenste adresbits op nul te staan.

Om bijvoorbeeld aan te geven dat de tekengeneratorgegevens vanaf $41200 (266752 decimaal) moeten worden gelezen, kan het volgende worden gebruikt. Houd er rekening mee dat het commando WPOKE kan worden gebruikt om twee bytes als een woord naar een geheugen- of I/O-locatie te schrijven. Wij gebruiken daarom WPOKE om $00 naar $D068 en $12 naar $D069 te schrijven, en een extra POKE om de hoge byte $A naar $D06A te schrijven door het adres door 65536 te delen:

```
REM C65+MEGA65-I/O INSCHAKELEN
IF PEEK($D018)<32 THEN POKE $D02F,ASC("G"):POKE $D02F,ASC("$")
REM HEX $41200 IS MAKKELIJK VERDEELD IN ZIJN 3 BYTES $00, $12, $4
REM WPOKE ZET DE ONDERSTE TWEE BYTES IN EEN COMMANDO EN
REM DE VOLGENDE POKE ZET DE BOVENSTE BYTE
A=$41200
WPOKE $D068,A
POKE $D06A,A/65536
```

Het kleuren-/attribuutgeheugen verplaatsen

Het gebruikte deel van het kleurengeheugen kan op vergelijkbare wijze worden ingesteld met de COLPTR-registers ($D064 - $D065, 53348 - 53349 decimaal). De waarde is namelijk een verschuiving vanaf het begin van het kleuren-/attribuutgeheugen. Dit komt omdat het kleuren-/attribuutgeheugen van de MEGA65, net als op de C64, een afzonderlijk geheugenonderdeel is met een eigen verbinding met de VIC-IV. Standaard zijn de COLPTR's op nul ingesteld, wat het gedrag van de VIC-II/III nabootst. Om de weergave het kleuren-/attribuutgeheugen vanaf verschuiving $4000 te laten gebruiken, kan men bijvoorbeeld iets als het volgende gebruiken:

```
REM C65+MEGA65-I/O INSCHAKELEN
IF PEEK($D018)<32 THEN POKE $D02F,ASC("G"):POKE $D02F,ASC("$")
REM STEL COLPTR IN OP $4000, VERDEELD IN $00-LSB en $40-MSB
POKE $D064,$00
POKE $D065,$40
```

Spritepointers en sprite-afbeeldingen verplaatsen

De locatie van de spritepointers kan eveneens worden verplaatst, en sprites kunnen hun gegevens overal in de eerste 4MB van het geheugen laten hebben. Dit wordt bereikt door eerst de locatie van de spritepointers in te stellen via de SPRPTRADR-registers ($D06C - $D06E, 53356 - 53358 decimaal; houd er echter rekening mee dat alleen de onderste 7 bits van $D06E worden gebruikt, omdat de hoogste bit voor het SPRPTR16-signaal wordt gebruikt). Hierdoor kan de lijst van acht spritepointers van het einde van het schermgeheugen naar een willekeurige locatie in de eerste 8MB van het geheugen worden verplaatst. SPRPTRADR moet op een 16-byte-grens (een veelvoud van 16) zijn uitgelijnd.

Om sprites zelf overal in de eerste 4MB van het geheugen te kunnen plaatsen, moet de SPRPTR16-bit in $D06E worden gezet. In deze modus worden twee bytes gebruikt om de locatie van elke sprite aan te geven, in plaats van één byte. De lijst van spritepointers is daardoor 16 bytes lang, in plaats van 8 bytes zoals op de VIC-II/III. Wanneer SPRPTR16 is ingeschakeld, moet de locatie van de spritepointers altijd expliciet via de SPRPTRADR-registers worden ingesteld.

Om de spritepointers bijvoorbeeld op locatie 800 - 815 te plaatsen, kunt u min of meer de volgende code gebruiken. Houd er rekening mee dat enige kunstgrepen nodig zijn om de SPRPTR16-bit ongewijzigd te laten, en ook om de AND-binaire operator, die niet met waarden groter dan 65535 werkt, te omzeilen:

```
REM C65+MEGA65-I/O INSCHAKELEN
IF PEEK($D018)<32 THEN POKE $D02F,ASC("G"):POKE $D02F,ASC("S")
POKE $D06C,(800-INT(800/65536)*65536) AND 255
POKE $D06D,INT(800/256) AND 255
POKE $D06E,(PEEK($D06E) AND 128)+INT(800/65536)
```

De locatie van elke sprite-afbeelding blijft een veelvoud van 64 bytes, waardoor er op elk moment tot 65,536 unieke sprite-afbeeldingen kunnen worden gebruikt, indien het systeem over voldoende geheugen beschikt (4MB of meer). In deze modus wordt de 16KB-banking van de VIC-II genegeerd, en is de locatie van de spritegegevens eenvoudigweg 64 $\times$ de pointerwaarde. Om de gegevens voor een sprite op $C000 (49152 decimaal) te hebben, zou dit bijvoorbeeld sprite-locatie 768 zijn, omdat 49152 gedeeld door 64 = 768. Vervolgens moeten wij 768 in hoge en lage bytes splitsen om de twee pointerbytes in te stellen: 768 = 256$\times$3, met rest 0. Dit zou dus vereisen dat de twee spritepointerbytes 0 (lage byte, die als eerste komt) en 3 (hoge byte) zijn. Indien de spritepointers zich op $7F8 (2040 decimaal) bevonden, zou het instellen van de eerste sprite op sprite-afbeelding 768 kunnen worden gedaan met zoiets als:

```
POKE 2040,768-256*INT(768/256)
POKE 2041,INT(768/256)
```

HETE REGISTERS

Sommige VIC-IV-registers ondersteunen functies die vergelijkbaar zijn met die van de VIC-II en VIC-III, maar met uitgebreide mogelijkheden. Voor achterwaartse compatibiliteit zorgt het schrijven naar specifieke VIC-II- en VIC-III-registers er ook voor dat de bijbehorende VIC-IV-registers worden gereset met consistente waarden. Dit gedrag kan worden geconfigureerd met de HOTREG-vlag in bit 7 van $D05D.

De onderste vier bits van register $D018 (**CB**) stellen bijvoorbeeld het tekensetadres van de VIC-II in, als een veelvoud van 1 KiB. De VIC-IV kan de tekenset op elk 24-bits adres plaatsen met behulp van $D06A (**CHARPTRBNK**), $D068 (**CHARPTRLSB**) en $D069 (**CHARPTRMSB**). Indien u CB instelt, worden de VIC-IV-registers ook bijgewerkt om hiermee overeen te komen.

De volledige reeks "hete" VIC-II-registers die van invloed zijn op de VIC-IV-registers omvat:

- $D011 (53265): RB8, ECM, BMM, BLNK, RSEL, YSCL
- $D016 (53270): RST, MCM, CSEL, XSCL
- $D018 (53272): VS, CB
- $D031 (53297): VIC-III-modi: H640, FAST, ATTR, BPM, V400, H1280, MONO, INT
- De VIC-II-bankbits van $DD00 (56576) (CIA 2 PORTA)

Telkens wanneer een van die registers wordt gewijzigd, zelfs door de bestaande waarde er weer in terug te schrijven, worden alle bijbehorende VIC-IV-registers bijgewerkt op basis van de VIC-II-registerwaarden, zelfs die welke geen verband houden met het register waarnaar is geschreven. Dit omvat de FAST-vlag, welke zich op een byte-locatie van een heet register ($D031) bevindt, maar geen verband houdt met de videoparameters en geen bijbehorend VIC-IV-register heeft.

De registers die tijdens dit proces worden gewijzigd, staan hieronder vermeld. Houd er rekening mee dat sommige van deze registers intern zijn aan de VIC-IV en niet rechtstreeks door de gebruiker kunnen worden opgevraagd of gewijzigd. Waar dit het geval is, worden er geen adressen voor de registers vermeld.

- **X-positie van de linkerrandgrens.** Dit interne register wordt bijgewerkt om de linkerrand in te stellen op de breedte die in de "Single Side Border Width"-registers is aangegeven ($D05C bevat de LSB, en bits 0 – 5 van $D05D bevatten de MSB van de breedte van de zijrand. Houd er rekening mee dat de breedte van de zijrand is gebaseerd op de afmetingen van het videoframe op laag niveau, niet op de weergavegrootte van het scherm van de videomodus. Wanneer het 38/40-kolommenveld van $D016 op 38 kolommen is ingesteld, verschijnt de linkerrandgrens 14 pixels rechts van zijn normale positie.
- **X-positie van de rechterrandgrens.** Dit is hetzelfde als de linkerrandgrens, maar dan voor de rechterrand van het scherm. Houd er rekening mee dat wan-

neer de 38/40-kolommenvlag op 38 kolommen is ingesteld, de rechterrandgrens 17 pixels naar links van zijn normale positie wordt verplaatst.

- **Y-positie van de bovenste randgrens ($D048 LSB, bits 0 – 3 van $D049 MSB).** Dit interne register wordt ingesteld op de normale bovenste positie van het scherm, minus de waarde van het RASLINE0-veld in bits 0 – 5 van $D06F. Wanneer het 24/25-rijenveld van $D011 op 24 rijen is ingesteld, wordt de bovenste randgrens met 8 rasterlijnen verlaagd.
- **Y-positie van de onderste randgrens ($D04A LSB, bits 0 – 3 van $D04B MSB).** Dit interne register wordt ingesteld op de normale bovenste positie van het scherm, plus 400 rasterlijnen, om het normale 400px hoge primaire weergavegebied binnen de randen te creëren. Wanneer het 24/25-rijenveld van $D011 op 24 rijen is ingesteld, wordt de bovenste randgrens met 8 rasterlijnen verhoogd.
- **Verticale schaal van de tekengenerator ($D05B).** Dit register wordt ingesteld op 0 voor de V200- of 1 voor de V400-modus, zodat elke pixelrij in een teken respectievelijk 1 of 2 pixels hoog wordt, volgens de V400-vlag.
- **Aantal weer te geven tekenrijen ($D07B).** Dit register wordt ingesteld op 25-1 = 24 of 50-1 = 49 om respectievelijk 25 of 50 tekstrijen weer te geven. Houd er rekening mee dat wanneer $D011 wordt gebruikt om de verticale randen naar binnen te brengen om het aantal zichtbare tekenrijen te verminderen, de VIC-IV nog steeds alle 25 of 50 rijen tekent.
- **X-positie waar de tekenweergave begint ($D04C LSB, bits 0 – 3 van $D04D MSB).** Dit register wordt ingesteld op een positie ten opzichte van de rand van het 40-kolommen brede tekstscherm, plus 2× de positie van de vloeiende verschuiving die in $D016 is aangegeven.
- **Y-positie waar de tekenweergave begint ($D04E LSB, bits 0 – 3 van $D04F MSB).** Dit register wordt ingesteld op de bovenrand van de verticale rand, minus het VIC-II First Raster-aanpassingsregister (bits 0 – 5 van $D06F), plus eventuele verschuiving als gevolg van de verticale bits voor vloeiende verschuiving in $D011.
- **Virtuele rijbreedte ($D058 LSB, $D059 MSB), d.w.z. het aantal bytes scherm- en kleurengeheugen waarmee de VIC-IV vooruitgaat bij het weergeven van elke opeenvolgende tekenrij.** Dit register wordt ingesteld op 40 als de H640-vlag is gewist, of op 80 als de H640-vlag is gezet, waardoor de voortgang overeenkomt met het aantal weer te geven tekens.
- **Breedte van de weergaverij ($D05E LSB, bits 4 – 5 van $D063 MSB).** Dit register wordt ingesteld op 40 als de H640-vlag is gewist, of op 80 als de H640-vlag is gezet.
- **Basisadres van het schermgeheugen ($D060 – $D062, dat een 24-bits adres voorstelt).** Dit adres wordt gereset naar het adres zoals berekend aan de hand van $D018 en $DD00, zoals op de C64.

- **Spritepointer-adres van de VIC-II ($D06C - $D06E, dat een 24-bits adres voorstelt).** Dit register wordt gereset naar de normale locatie aan het einde van het schermgeheugen van de huidige modus. Indien de H640-vlag is gezet, gebeurt dit aan het einde van het 2KB-gebied van het schermgeheugen, of indien de H640-vlag niet is gezet, wijst het naar het einde van het 1KB-gebied van het schermgeheugen, zoals op een C64.
- **Basisadres van de tekenset ($D068 - $D06A, dat een 24-bits adres voorstelt).** Houd er rekening mee dat de hete registerfunctie alleen de onderste 16 bits van het tekensetadres instelt. Dat wil zeggen dat $D06A niet wordt gewist. Dit is bewust gedrag, waardoor het gemakkelijker wordt om in bestaande VIC-II-georiënteerde software de tekenset te vervangen door een andere tekenset in een andere geheugenbank.
- **Basisadres van het kleurengeheugen ($D064 LSB, $D065 MSB).** Deze registers worden naar nul gereset, waardoor de VIC-IV verwacht dat het kleurengeheugen voor het scherm zich in het eerste deel van het kleurengeheugen bevindt, om compatibel te zijn met de VIC-II en VIC-III.

Dit gedrag van de VIC-II-registers is vooral bedoeld voor software uit het verleden die geen weet heeft van (en daarom nooit naar) VIC-IV-registers schrijft. De heteregisterpropagatie kan worden uitgeschakeld door het HOTREG- ("hete register") signaal te wissen: bit 7 van $D05D (53341). Indien u de HOTREG-vlag wist, moet u de VIC-IV-registers rechtstreeks bijwerken wanneer u de parameters van de videomodus wilt wijzigen.

Indien u een wijziging in een heet register aanbrengt terwijl HOTREG is uitgeschakeld en HOTREG vervolgens weer inschakelt, worden alle hete registers onmiddellijk bijgewerkt. Er zijn zeldzame gevallen waarin u een wijziging in een heet register wilt aanbrengen zonder de VIC-IV-registers bij te werken, maar de hete registers na de wijziging toch ingeschakeld wilt hebben. Om dit te doen, kunt u de openstaande bijwerking annuleren door vlak voordat u de hete registers weer inschakelt opnieuw een 0 naar HOTREG te schrijven. De volledige procedure is:

1. Schrijf 0 naar HOTREG om hete registers uit te schakelen.
2. Werk het VIC-II-register bij dat u wilt wijzigen.
3. Schrijf 0 naar HOTREG om de openstaande bijwerking te annuleren.
4. Schrijf 1 naar HOTREG om hete registers weer in te schakelen.

In assembleertaal:

```
lda #%10000000
trb $d05d        ; hete registers uitschakelen

lda #%01000000
tsb $d031        ; een VIC-II/III-register bijwerken

lda #%10000000
trb $d05d        ; openstaande bijwerking wissen
tsb $d05d        ; hete registers weer inschakelen
```

NIEUWE MODI

Waarom de nieuwe VIC-IV-modi teken- en bitmapmodi zijn en geen bitplanemodi

De nieuwe videomodi van de VIC-IV zijn afgeleid van de teken- en bitmapmodi van de VIC-II, in plaats van de bitplanemodi van de VIC-III. Deze keuze was gebaseerd op diverse praktische aspecten van het programmeren van een geheugenbeperkte 8-bit thuiscomputer:

1. Bitplanes vereisen dat dezelfde hoeveelheid geheugen aan elk gebied op het scherm wordt toegewezen, ongeacht of het lege ruimte of complexe afbeeldingen weergeeft. Met bitplanes is het niet mogelijk om inhoud uit een afbeelding opnieuw te gebruiken in een ander deel van die afbeelding. De meeste C64-spellen maken echter gebruik van sterk repetitieve schermen, waarbij veelgebruikte elementen op diverse plaatsen op het scherm voorkomen, waarvan Boulder Dash en Super Giana Sisters goede voorbeelden zijn.
2. Bitplanes maken het ook moeilijk om een beeldscherm bij te werken, omdat elke pixel uniek is: er is geen mogelijkheid om een wijziging, bijvoorbeeld in de animatie van een schermelement, door te voeren op alle plaatsen tegelijk. De diamantanimaties in Boulder Dash zijn een goed voorbeeld van dit probleem. Het moeten wijzigen van meerdere afzonderlijke bytes in elke bitplane vormt een verhoogde rekenkundige belasting, hetgeen de reden was dat er voor de Amiga AAA-chipset werd gevraagd zogenaamde “chunky”-modi op te nemen, in plaats van uitsluitend op bitplanes gebaseerde modi. Hoewel de Display Address Translator (DAT) en DMAgic van de C65 enige verlichting bieden voor dit probleem, is die verlichting slechts gedeeltelijk.
3. Scrollen met behulp van de bitplanes van de C65 vereist het kopiëren van de gehele bitplane, omdat de hardwareondersteuning voor vloeiend scrollen zich niet uitstrekt tot het op fijne wijze wijzigen van het bronadres van de bitplane. Zelfs met gebruikmaking van de DMAgic vereist het scrollen van een 320×200 256-kleurenscherm in het gunstigste geval 128.000 klokcycli (het lezen en schrijven van 320×200 = 64.000 bytes). Op 3.5 MHz op de C65 zou dit ongeveer 36 milliseconden kosten, oftewel ongeveer 2 volledige videoframes. Voor vloeiend

scrollen van een dergelijk scherm is daarom een dubbelgebufferde regeling nodig, die 128.000 van de 131.072 bytes geheugen zou verbruiken.

De bekende tekenmodi van de VIC-II worden daarentegen veelvuldig in spellen gebruikt, omdat ze met een geringe hoeveelheid schermgeheugen kunnen bepalen welk blok van 8×8 pixels wordt weergegeven, wat zeer snel scrollen, minder geheugengebruik en effectieve hardwareversnelling van animatie van veelgebruikte elementen mogelijk maakt. De verbeteringen in de VIC-IV zijn dan ook vooral op de tekenmodus gericht. Omdat de bitmapmodus op de VIC-II feitelijk een speciaal geval van de tekenmodus is, met impliciete tekennummers, wordt deze op de VIC-IV als het ware gratis meegeleverd en zal alleen in het kader van een zeer klein aantal bitmapmodus-specifieke verbeteringen worden genoemd die triviaal te implementeren waren en waarvan het dus onverstandig leek ze niet te realiseren, voor het geval ze toch van pas komen.

Weergeven van meer dan 256 unieke tekens via "Super-Uitgebreide Attribuutmodus"

De belangrijkste innovatie is de toevoeging van de Super-Uitgebreide Attribuutmodus (SEAM). De VIC-II gebruikt al 12 bits per teken: elke cel van 8×8 pixels wordt gedefinieerd door 12 bits gegevens: 8 bits schermgeheugengegevens, standaard van $0400 - $07E7 (1024 - 2023 decimaal), die aangeven welke tekens moeten worden weergegeven, en 4 bits kleurgegevens van het 1 KB nibble kleurengeheugen op $D800 - $DBFF (55296 - 56319 decimaal). De VIC-III van de C65 gebruikt 16 bits, omdat het kleurengeheugen nu 8 bits telt in plaats van 4, waarbij de extra 4 bits kleurengeheugen worden gebruikt om attributen te ondersteunen (knipperen, vet, onderstrepen en inverse weergave). Het wordt aanbevolen om de werking hiervan door te nemen voordat u verder leest. Een goede inleiding op de tekstmodus van de VIC-II is op veel plaatsen te vinden. Super-Uitgebreide Attribuutmodus verdubbelt het aantal bits per teken van de 16 van de VIC-III tot 32: twee bytes schermgeheugen en twee bytes kleuren-/attribuutgeheugen.

Super-Uitgebreide Attribuutmodus wordt ingeschakeld door bit 0 in $D054 (53332 decimaal) te zetten. Vergeet niet eerst de VIC-IV-modus in te schakelen om dit register toegankelijk te maken. Wanneer deze bit is gezet, worden twee bytes gebruikt voor zowel het schermgeheugen als het kleurengeheugen voor elk teken dat op het scherm wordt weergegeven. In vergelijking met de 12 bits informatie die de C64 per teken gebruikt en de 16 bits van de VIC-III, heeft de VIC-IV 32 bits informatie. Hoe deze 32 bits worden gebruikt, varieert enigszins per modus, zoals beschreven in de volgende tabellen, inclusief of de GOTOX-bit is gezet.

Let er ook op dat het tegelijkertijd inschakelen van de BOLD- en REVERSE-attributen op de MEGA65 een alternatief palet selecteert, waardoor feitelijk 512 kleuren op het scherm mogelijk zijn, maar elk 8×8-teken slechts kleuren uit één palet van 256 kleuren kan gebruiken.

De twee tabellen die volgen beschrijven de bitvelden die voor dit teken beschikbaar zijn, die variëren afhankelijk van of de **GOTOX_flag** is gezet of gewist:

- **GOTOX_flag = 0** creëert een zichtbaar teken (blauwe bitvelden) met attributen die dit teken beïnvloeden.
- **GOTOX_flag = 1** creëert een onzichtbaar teken (rode bitvelden) met attributen die volgende tekens beïnvloeden.

Bitvelden indien GOTOX_flag is gewist:

Bit(s)	Functie wanneer GOTOX_flag is gewist
Schermgeheugen-byte 0	
Bits 7 - 0	**char_number (lo)**: (bits 7 - 0) Laagste byte van tekennummer (hetzelfde als de VIC-II en VIC-III)
Schermgeheugen-byte 1	
Bits 7 - 5	**rhs_trim (lo)**: (bits 2 - 0) Pixels aan de rechterkant van het teken wegsnijden)
Bits 4 - 0	**char_number (hi)**: (bits 12 - 8) Bovenste 5 bits van tekennummer, waardoor 8.192 unieke tekens kunnen worden aangesproken
Kleurengeheugen-byte 0	
Bit 7	**vert_flip**: het teken verticaal spiegelen
Bit 6	**horz_flip**: het teken horizontaal spiegelen
Bit 5	**alpha_mode**: de alpha-blendmodus inschakelen. De pixelwaarden worden dan als alpha-waarden behandeld die tussen de voorgrondkleur en de achtergrondkleur mengen (hiervoor moet $D054.7 - ALPHEN worden ingesteld)
Bit 4	**GOTOX_flag** is **gewist** **(op 0 gesteld)** Verschaft de blauwe bitvelden die in deze tabel worden getoond. Dit zijn attributen die invloed hebben op dit teken.
Bit 3	**ncm_flag**: • 1 = NCM-teken: gebruikt 4 bits per pixel en is 16 pixels breed (minus eventuele trim-bits aan de rechterkant) • 0 = FCM-teken: gebruikt 8 bits per pixel en is 8 pixels breed (minus eventuele trim-bits aan de rechterkant)
Bit 2	**rhs_trim (hi)**: (bit 3) pixels aan de rechterkant van het teken wegsnijden
Bits 1 - 0	GERESERVEERD, op 0 gesteld
Kleurengeheugen-byte 1	
Indien de VIC-II-multicolourmodus is ingeschakeld: ($D016.4 - MCM)	
Bits 7 - 4	**colour (hi)**: bovenste 4 bits van de voorgrondkleur van het teken
Indien de uitgebreide attributen van de VIC-III zijn ingeschakeld: ($D031.5 - ATTR)	
Bit 7	**underline**: Hardwarematige onderstreping van het teken
Bit 6	**bold**: Vetattribuut van het teken (hardware) *
Bit 5	**reverse**: Omgekeerde weergave van het teken (hardware) *
Bit 4	**blink**: Knipperen van het teken (hardware)
Het overgebleven bitveld is gemeenschappelijk:	
Bits 3 - 0	**colour (lo)**: onderste 4 bits van de voorgrondkleur van het teken

Bitvelden indien GOTOX_flag is ingesteld:

Bit(s)	Functie wanneer GOTOX_flag is ingesteld
Schermgeheugen-byte 0	
Bits 7 - 0	**gotox_pos (lo)**: (bits 7 - 0) onderste 8 bits van de nieuwe X-positie waar volgende tekens moeten beginnen met tekenen, ten opzichte van het begin van het tekenen van het teken. Door dit op 0 te stellen wordt het volgende teken over het meest linkse teken heen getekend.
Schermgeheugen-byte 1	
Bits 7 - 5	**fcm_yoffs**: Y-offset van FCM-tekengegevens: wanneer niet nul, worden FCM-tekengegevens opgehaald vanaf het tekenadres + 8 × de Y-waarde, wat equivalent is aan één pixellijn FCM-tekengegevens per Y. Door zorgvuldig plannen kan dit worden gebruikt om meerdere lagen RRB-inhoud soepel verticaal te scrollen.
Bit 4	**fcm_yoffs_dir**: wanneer ingesteld, wordt het **fcm_yoffs**-veld in plaats daarvan in mindering gebracht.
Bits 3 - 2	GERESERVEERD, op 0 ingesteld
Bits 1 - 0	**gotox_pos (hi)**: (bits 9 - 8) bovenste 2 bits van de nieuwe X-positie (het hoogste bit is het tekenbit van het tweecomplement)
Kleurengeheugen-byte 0	
Bit 7	**transparent_flag**: wanneer ingesteld, worden achtergrond-/transparante pixels niet getekend voor volgende tekens, waardoor lagen mogelijk zijn
Bit 6	**background_flag**: wanneer ingesteld, worden volgende tekens als achtergrond gerenderd, waardoor sprites ervoor kunnen verschijnen, zelfs wanneer sprites op achtergrond zijn ingesteld.
Bit 5	GERESERVEERD, op 0 ingesteld
Bit 4	**GOTOX_flag** is **op 1 ingesteld**. Biedt de in deze tabel getoonde rode bitvelden, dit zijn attributen die volgende tekens beïnvloeden. Eén opvallend veld is **gotox_pos**, dat later in de sectie Raster-Rewrite Buffer wordt besproken.)
Bit 3	**rowmask_flag**: wanneer ingesteld, beslist het **rowmask**-veld welke pixellijnen van volgende tekens moeten worden gerenderd. Dit kan worden gebruikt om tekens verticaal te scrollen met RRB, door elk teken op twee rijen te tekenen: verschoven naar beneden op de bovenste rij, en verschoven omhoog op de onderste rij, en gemaskeerd zodat alleen de pixellijnen van het gescrollde teken worden weergegeven, en niet de gegevens van vóór/na de gegevens van dat teken.
Bit 2	**foreground_flag**: wanneer ingesteld, worden volgende tekens als voorgrond getekend, ongeacht hun kleur, waardoor sprites erachter kunnen verschijnen.

vervolg ...

...vervolg

Bit(s)	Functie wanneer GOTOX_flag is ingesteld
Bits 1 - 0	GERESERVEERD, op 0 ingesteld
Kleurengeheugen-byte 1	
Bits 7 - 0	**rowmask**: maskerwaarden voor pixellijnen. Bits 7 - 0 = Rijen 7 - 0. 1 = rij weergeven, 0 = rij verbergen

We zien dat we nog steeds de onderste 8 bits van het tekennummer in de C64-stijl hebben in de eerste schermgeheugen-byte. De tweede byte van het schermgeheugen krijgt daarvoor vijf extra bits, waardoor er 2^{13} = 8.192 verschillende tekens op één scherm kunnen worden gebruikt. Dat is meer dan voldoende voor unieke tekens die een 80×50-scherm bestrijken (wat mogelijk is om te maken met de VIC-IV). De resterende bits maken het bijsnijden van het teken mogelijk. Hierdoor kunnen tekens met variabele breedte worden gebruikt, waarmee dingen gedaan kunnen worden die normaal niet mogelijk zouden zijn, zoals het gebruik van tekstmodus voor vrije horizontale plaatsing van tekens (of delen daarvan). Dit werd oorspronkelijk toegevoegd om hardwarematige ondersteuning voor proportionele breedte-lettertypen mogelijk te maken.

Voor het kleurengeheugen is de tweede byte (byte 1) gelijk aan die van de C65, oftewel de onderste helft biedt vier bits voor de voorgrondkleur, zoals op de C64, plus de optionele VIC-III uitgebreide attributen. Het C65-specificatiedocument beschrijft het gedrag wanneer meer dan één van deze tegelijk wordt gebruikt, waarvan de meeste logisch zijn, maar er zijn een paar combinaties die zich anders gedragen dan verwacht. Het combineren van vetgedrukt met knipperen zorgt er bijvoorbeeld voor dat het teken schakelt tussen de vetgedrukte en de normale modus. Vetgedrukte modus zelf wordt geïmplementeerd door effectief te fungeren als bit 4 van de voorgrondkleurwaarde, waardoor de kleur van andere paletposities wordt getekend dan normaal.

Indien u de VIC-III uitgebreide attributen niet nodig heeft, kunt u in plaats daarvan de bovenste vier bits van de tweede byte van het kleurengeheugen gebruiken om meer bits van de kleurindex te bevatten, waardoor een selectie uit het volledige bereik van 256 kleurinstellingen mogelijk is. Deze modus wordt geactiveerd door de multicolour-modus van de VIC-II in te schakelen terwijl de volledige kleurmodus actief is.

De C65 / VIC-III-attributen en het gebruik van 256-kleuren 8-bit-waarden voor verschillende VIC-II-kleurregisters wordt ingeschakeld door bit 5 van $D031 (53297 decimaal) in te stellen. Het wordt sterk aanbevolen dit te doen bij het gebruik van de VIC-IV-modus, omdat anders bepaalde functies niet naar behoren werken. Merk op dat BOLD+REVERSE samen de betekenis heeft van het selecteren van een alternatief palet op de MEGA65, wat verschilt van de C65.

Dankzij Super-Uitgebreide Attribuutmodus zijn veel effecten mogelijk. Een paar mogelijkheden worden in de volgende subsecties uitgelegd.

Super-Uitgebreide Attribuutmodus gebruiken

Super-Uitgebreide Attribuutmodus vereist twee keer zoveel schermgeheugen en kleurengeheugen als de VIC-II/III-tekstmodi. Dit komt doordat er twee bytes van elk nodig zijn om elk teken te definiëren, in plaats van één. Het schermgeheugen kan op elke willekeurige locatie in het 384KB hoofdgeheugen worden geplaatst met behulp van de registers $D060 - $D062 (53344 - 53346 decimaal). Het kleurengeheugen kan op elke willekeurige locatie in het 32KB kleurengeheugen worden geplaatst. Alleen de eerste 1 of 2KB van het kleurengeheugen is zichtbaar op $D800 - $DBFF of $D800 - $DFFF (indien het *CRAM2K*-signaal is ingesteld in bit 0 van $D030, 53296 decimaal). Indien een scherm groter dan 40×25 tekens wordt gebruikt, is het gebruik van de DMA-controller of een andere manier vereist om het volledige kleurengeheugen te benaderen. We bespreken daarom eerst het gebruik van Super-Uitgebreide Attribuutmodus met een 40x25-tekenweergave.

De eerste stap is het inschakelen van Super-Uitgebreide Attribuutmodus door de *FCLRHI*- en *CHR16*-signalen te activeren, door bits 2 en 0 van $D054 (53332 decimaal) in te stellen. Aangezien dit een VIC-IV-register is, moet eerst de VIC-IV I/O-modus worden ingeschakeld. De VIC-IV moet ook worden geconfigureerd naar 40-kolomsmodus, door het *H640*-signaal uit te schakelen door bit 7 van $D031 (53297 decimaal) te wissen. Dit komt doordat elk tekenpaar samen één teken op het scherm vormt, waarbij één teken twee schermgeheugenbytes vereist, waardoor er 80 schermgeheugenbytes nodig zijn om 40 tekens weer te geven. Eveneens zijn er 80 kleurengeheugenbytes nodig.

Om dit visueel te begrijpen, is het nuttig eerst de normale C64-schermgeheugenindeling te bekijken:

$400	$401	$402	$403	$404	$405	$406	$407	$408	$409	$40a	$40b	$40c	$40d	$40e	$40f	$410	$411	$412	$413	$414	$415	$416	$417	$418	$419	$41a	$41b	$41c	$41d	$41e	$41f	$420	$421	$422	$423	$424	$425	$426	$427
$428	$429	$42a	$42b	$42c	$42d	$42e	$42f	$430	$431	$432	$433	$434	$435	$436	$437	$438	$439	$43a	$43b	$43c	$43d	$43e	$43f	$440	$441	$442	$443	$444	$445	$446	$447	$448	$449	$44a	$44b	$44c	$44d	$44e	$44f
$450	$451	$452	$453	$454	$455	$456	$457	$458	$459	$45a	$45b	$45c	$45d	$45e	$45f	$460	$461	$462	$463	$464	$465	$466	$467	$468	$469	$46a	$46b	$46c	$46d	$46e	$46f	$470	$471	$472	$473	$474	$475	$476	$477
$478	$479	$47a	$47b	$47c	$47d	$47e	$47f	$480	$481	$482	$483	$484	$485	$486	$487	$488	$489	$48a	$48b	$48c	$48d	$48e	$48f	$490	$491	$492	$493	$494	$495	$496	$497	$498	$499	$49a	$49b	$49c	$49d	$49e	$49f
$4a0	$4a1	$4a2	$4a3	$4a4	$4a5	$4a6	$4a7	$4a8	$4a9	$4aa	$4ab	$4ac	$4ad	$4ae	$4af	$4b0	$4b1	$4b2	$4b3	$4b4	$4b5	$4b6	$4b7	$4b8	$4b9	$4ba	$4bb	$4bc	$4bd	$4be	$4bf	$4c0	$4c1	$4c2	$4c3	$4c4	$4c5	$4c6	$4c7
$4c8	$4c9	$4ca	$4cb	$4cc	$4cd	$4ce	$4cf	$4d0	$4d1	$4d2	$4d3	$4d4	$4d5	$4d6	$4d7	$4d8	$4d9	$4da	$4db	$4dc	$4dd	$4de	$4df	$4e0	$4e1	$4e2	$4e3	$4e4	$4e5	$4e6	$4e7	$4e8	$4e9	$4ea	$4eb	$4ec	$4ed	$4ee	$4ef
$4f0	$4f1	$4f2	$4f3	$4f4	$4f5	$4f6	$4f7	$4f8	$4f9	$4fa	$4fb	$4fc	$4fd	$4fe	$4ff	$500	$501	$502	$503	$504	$505	$506	$507	$508	$509	$50a	$50b	$50c	$50d	$50e	$50f	$510	$511	$512	$513	$514	$515	$516	$517
$518	$519	$51a	$51b	$51c	$51d	$51e	$51f	$520	$521	$522	$523	$524	$525	$526	$527	$528	$529	$52a	$52b	$52c	$52d	$52e	$52f	$530	$531	$532	$533	$534	$535	$536	$537	$538	$539	$53a	$53b	$53c	$53d	$53e	$53f
$540	$541	$542	$543	$544	$545	$546	$547	$548	$549	$54a	$54b	$54c	$54d	$54e	$54f	$550	$551	$552	$553	$554	$555	$556	$557	$558	$559	$55a	$55b	$55c	$55d	$55e	$55f	$560	$561	$562	$563	$564	$565	$566	$567
$568	$569	$56a	$56b	$56c	$56d	$56e	$56f	$570	$571	$572	$573	$574	$575	$576	$577	$578	$579	$57a	$57b	$57c	$57d	$57e	$57f	$580	$581	$582	$583	$584	$585	$586	$587	$588	$589	$58a	$58b	$58c	$58d	$58e	$58f
$590	$591	$592	$593	$594	$595	$596	$597	$598	$599	$59a	$59b	$59c	$59d	$59e	$59f	$5a0	$5a1	$5a2	$5a3	$5a4	$5a5	$5a6	$5a7	$5a8	$5a9	$5aa	$5ab	$5ac	$5ad	$5ae	$5af	$5b0	$5b1	$5b2	$5b3	$5b4	$5b5	$5b6	$5b7
$5b8	$5b9	$5ba	$5bb	$5bc	$5bd	$5be	$5bf	$5c0	$5c1	$5c2	$5c3	$5c4	$5c5	$5c6	$5c7	$5c8	$5c9	$5ca	$5cb	$5cc	$5cd	$5ce	$5cf	$5d0	$5d1	$5d2	$5d3	$5d4	$5d5	$5d6	$5d7	$5d8	$5d9	$5da	$5db	$5dc	$5dd	$5de	$5df
$5e0	$5e1	$5e2	$5e3	$5e4	$5e5	$5e6	$5e7	$5e8	$5e9	$5ea	$5eb	$5ec	$5ed	$5ee	$5ef	$5f0	$5f1	$5f2	$5f3	$5f4	$5f5	$5f6	$5f7	$5f8	$5f9	$5fa	$5fb	$5fc	$5fd	$5fe	$5ff	$600	$601	$602	$603	$604	$605	$606	$607
$608	$609	$60a	$60b	$60c	$60d	$60e	$60f	$610	$611	$612	$613	$614	$615	$616	$617	$618	$619	$61a	$61b	$61c	$61d	$61e	$61f	$620	$621	$622	$623	$624	$625	$626	$627	$628	$629	$62a	$62b	$62c	$62d	$62e	$62f
$630	$631	$632	$633	$634	$635	$636	$637	$638	$639	$63a	$63b	$63c	$63d	$63e	$63f	$640	$641	$642	$643	$644	$645	$646	$647	$648	$649	$64a	$64b	$64c	$64d	$64e	$64f	$650	$651	$652	$653	$654	$655	$656	$657
$658	$659	$65a	$65b	$65c	$65d	$65e	$65f	$660	$661	$662	$663	$664	$665	$666	$667	$668	$669	$66a	$66b	$66c	$66d	$66e	$66f	$670	$671	$672	$673	$674	$675	$676	$677	$678	$679	$67a	$67b	$67c	$67d	$67e	$67f
$680	$681	$682	$683	$684	$685	$686	$687	$688	$689	$68a	$68b	$68c	$68d	$68e	$68f	$690	$691	$692	$693	$694	$695	$696	$697	$698	$699	$69a	$69b	$69c	$69d	$69e	$69f	$6a0	$6a1	$6a2	$6a3	$6a4	$6a5	$6a6	$6a7
$6a8	$6a9	$6aa	$6ab	$6ac	$6ad	$6ae	$6af	$6b0	$6b1	$6b2	$6b3	$6b4	$6b5	$6b6	$6b7	$6b8	$6b9	$6ba	$6bb	$6bc	$6bd	$6be	$6bf	$6c0	$6c1	$6c2	$6c3	$6c4	$6c5	$6c6	$6c7	$6c8	$6c9	$6ca	$6cb	$6cc	$6cd	$6ce	$6cf
$6d0	$6d1	$6d2	$6d3	$6d4	$6d5	$6d6	$6d7	$6d8	$6d9	$6da	$6db	$6dc	$6dd	$6de	$6df	$6e0	$6e1	$6e2	$6e3	$6e4	$6e5	$6e6	$6e7	$6e8	$6e9	$6ea	$6eb	$6ec	$6ed	$6ee	$6ef	$6f0	$6f1	$6f2	$6f3	$6f4	$6f5	$6f6	$6f7
$6f8	$6f9	$6fa	$6fb	$6fc	$6fd	$6fe	$6ff	$700	$701	$702	$703	$704	$705	$706	$707	$708	$709	$70a	$70b	$70c	$70d	$70e	$70f	$710	$711	$712	$713	$714	$715	$716	$717	$718	$719	$71a	$71b	$71c	$71d	$71e	$71f
$720	$721	$722	$723	$724	$725	$726	$727	$728	$729	$72a	$72b	$72c	$72d	$72e	$72f	$730	$731	$732	$733	$734	$735	$736	$737	$738	$739	$73a	$73b	$73c	$73d	$73e	$73f	$740	$741	$742	$743	$744	$745	$746	$747
$748	$749	$74a	$74b	$74c	$74d	$74e	$74f	$750	$751	$752	$753	$754	$755	$756	$757	$758	$759	$75a	$75b	$75c	$75d	$75e	$75f	$760	$761	$762	$763	$764	$765	$766	$767	$768	$769	$76a	$76b	$76c	$76d	$76e	$76f
$770	$771	$772	$773	$774	$775	$776	$777	$778	$779	$77a	$77b	$77c	$77d	$77e	$77f	$780	$781	$782	$783	$784	$785	$786	$787	$788	$789	$78a	$78b	$78c	$78d	$78e	$78f	$790	$791	$792	$793	$794	$795	$796	$797
$798	$799	$79a	$79b	$79c	$79d	$79e	$79f	$7a0	$7a1	$7a2	$7a3	$7a4	$7a5	$7a6	$7a7	$7a8	$7a9	$7aa	$7ab	$7ac	$7ad	$7ae	$7af	$7b0	$7b1	$7b2	$7b3	$7b4	$7b5	$7b6	$7b7	$7b8	$7b9	$7ba	$7bb	$7bc	$7bd	$7be	$7bf
$7c0	$7c1	$7c2	$7c3	$7c4	$7c5	$7c6	$7c7	$7c8	$7c9	$7ca	$7cb	$7cc	$7cd	$7ce	$7cf	$7d0	$7d1	$7d2	$7d3	$7d4	$7d5	$7d6	$7d7	$7d8	$7d9	$7da	$7db	$7dc	$7dd	$7de	$7df	$7e0	$7e1	$7e2	$7e3	$7e4	$7e5	$7e6	$7e7

Dat wil zeggen: elk tekencel gebruikt één byte van het schermgeheugen en de adressen nemen geleidelijk toe, zowel binnen regels als tussen regels. Super-Uitgebreide Attribuutmodus vereist twee bytes per tekencel. Indien u $D054 bijvoorbeeld op $05 instelt, krijgt u schermadressen als volgt:

$400	$402	$404	$406	$408	$40a	$40c	$40e	$410	$412	$414	$416	$418	$41a	$41c	$41e	$420	$422	$424	$426	$428	$42a	$42c	$42e	$430	$432	$434	$436	$438	$43a	$43c	$43e	$440	$442	$444	$446	$448	$44a	$44c	$44e
$428	$42a	$42c	$42e	$430	$432	$434	$436	$438	$43a	$43c	$43e	$440	$442	$444	$446	$448	$44a	$44c	$44e	$450	$452	$454	$456	$458	$45a	$45c	$45e	$460	$462	$464	$466	$468	$46a	$46c	$46e	$470	$472	$474	$476
$450	$452	$454	$456	$458	$45a	$45c	$45e	$460	$462	$464	$466	$468	$46a	$46c	$46e	$470	$472	$474	$476	$478	$47a	$47c	$47e	$480	$482	$484	$486	$488	$48a	$48c	$48e	$490	$492	$494	$496	$498	$49a	$49c	$49e
$478	$47a	$47c	$47e	$480	$482	$484	$486	$488	$48a	$48c	$48e	$490	$492	$494	$496	$498	$49a	$49c	$49e	$4a0	$4a2	$4a4	$4a6	$4a8	$4aa	$4ac	$4ae	$4b0	$4b2	$4b4	$4b6	$4b8	$4ba	$4bc	$4be	$4c0	$4c2	$4c4	$4c6
$4a0	$4a2	$4a4	$4a6	$4a8	$4aa	$4ac	$4ae	$4b0	$4b2	$4b4	$4b6	$4b8	$4ba	$4bc	$4be	$4c0	$4c2	$4c4	$4c6	$4c8	$4ca	$4cc	$4ce	$4d0	$4d2	$4d4	$4d6	$4d8	$4da	$4dc	$4de	$4e0	$4e2	$4e4	$4e6	$4e8	$4ea	$4ec	$4ee
$4c8	$4ca	$4cc	$4ce	$4d0	$4d2	$4d4	$4d6	$4d8	$4da	$4dc	$4de	$4e0	$4e2	$4e4	$4e6	$4e8	$4ea	$4ec	$4ee	$4f0	$4f2	$4f4	$4f6	$4f8	$4fa	$4fc	$4fe	$500	$502	$504	$506	$508	$50a	$50c	$50e	$510	$512	$514	$516
$4f0	$4f2	$4f4	$4f6	$4f8	$4fa	$4fc	$4fe	$500	$502	$504	$506	$508	$50a	$50c	$50e	$510	$512	$514	$516	$518	$51a	$51c	$51e	$520	$522	$524	$526	$528	$52a	$52c	$52e	$530	$532	$534	$536	$538	$53a	$53c	$53e
$518	$51a	$51c	$51e	$520	$522	$524	$526	$528	$52a	$52c	$52e	$530	$532	$534	$536	$538	$53a	$53c	$53e	$540	$542	$544	$546	$548	$54a	$54c	$54e	$550	$552	$554	$556	$558	$55a	$55c	$55e	$560	$562	$564	$566
$540	$542	$544	$546	$548	$54a	$54c	$54e	$550	$552	$554	$556	$558	$55a	$55c	$55e	$560	$562	$564	$566	$568	$56a	$56c	$56e	$570	$572	$574	$576	$578	$57a	$57c	$57e	$580	$582	$584	$586	$588	$58a	$58c	$58e
$568	$56a	$56c	$56e	$570	$572	$574	$576	$578	$57a	$57c	$57e	$580	$582	$584	$586	$588	$58a	$58c	$58e	$590	$592	$594	$596	$598	$59a	$59c	$59e	$5a0	$5a2	$5a4	$5a6	$5a8	$5aa	$5ac	$5ae	$5b0	$5b2	$5b4	$5b6
$590	$592	$594	$596	$598	$59a	$59c	$59e	$5a0	$5a2	$5a4	$5a6	$5a8	$5aa	$5ac	$5ae	$5b0	$5b2	$5b4	$5b6	$5b8	$5ba	$5bc	$5be	$5c0	$5c2	$5c4	$5c6	$5c8	$5ca	$5cc	$5ce	$5d0	$5d2	$5d4	$5d6	$5d8	$5da	$5dc	$5de
$5b8	$5ba	$5bc	$5be	$5c0	$5c2	$5c4	$5c6	$5c8	$5ca	$5cc	$5ce	$5d0	$5d2	$5d4	$5d6	$5d8	$5da	$5dc	$5de	$5e0	$5e2	$5e4	$5e6	$5e8	$5ea	$5ec	$5ee	$5f0	$5f2	$5f4	$5f6	$5f8	$5fa	$5fc	$5fe	$600	$602	$604	$606
$5e0	$5e2	$5e4	$5e6	$5e8	$5ea	$5ec	$5ee	$5f0	$5f2	$5f4	$5f6	$5f8	$5fa	$5fc	$5fe	$600	$602	$604	$606	$608	$60a	$60c	$60e	$610	$612	$614	$616	$618	$61a	$61c	$61e	$620	$622	$624	$626	$628	$62a	$62c	$62e
$608	$60a	$60c	$60e	$610	$612	$614	$616	$618	$61a	$61c	$61e	$620	$622	$624	$626	$628	$62a	$62c	$62e	$630	$632	$634	$636	$638	$63a	$63c	$63e	$640	$642	$644	$646	$648	$64a	$64c	$64e	$650	$652	$654	$656
$630	$632	$634	$636	$638	$63a	$63c	$63e	$640	$642	$644	$646	$648	$64a	$64c	$64e	$650	$652	$654	$656	$658	$65a	$65c	$65e	$660	$662	$664	$666	$668	$66a	$66c	$66e	$670	$672	$674	$676	$678	$67a	$67c	$67e
$658	$65a	$65c	$65e	$660	$662	$664	$666	$668	$66a	$66c	$66e	$670	$672	$674	$676	$678	$67a	$67c	$67e	$680	$682	$684	$686	$688	$68a	$68c	$68e	$690	$692	$694	$696	$698	$69a	$69c	$69e	$6a0	$6a2	$6a4	$6a6
$680	$682	$684	$686	$688	$68a	$68c	$68e	$690	$692	$694	$696	$698	$69a	$69c	$69e	$6a0	$6a2	$6a4	$6a6	$6a8	$6aa	$6ac	$6ae	$6b0	$6b2	$6b4	$6b6	$6b8	$6ba	$6bc	$6be	$6c0	$6c2	$6c4	$6c6	$6c8	$6ca	$6cc	$6ce
$6a8	$6aa	$6ac	$6ae	$6b0	$6b2	$6b4	$6b6	$6b8	$6ba	$6bc	$6be	$6c0	$6c2	$6c4	$6c6	$6c8	$6ca	$6cc	$6ce	$6d0	$6d2	$6d4	$6d6	$6d8	$6da	$6dc	$6de	$6e0	$6e2	$6e4	$6e6	$6e8	$6ea	$6ec	$6ee	$6f0	$6f2	$6f4	$6f6
$6d0	$6d2	$6d4	$6d6	$6d8	$6da	$6dc	$6de	$6e0	$6e2	$6e4	$6e6	$6e8	$6ea	$6ec	$6ee	$6f0	$6f2	$6f4	$6f6	$6f8	$6fa	$6fc	$6fe	$700	$702	$704	$706	$708	$70a	$70c	$70e	$710	$712	$714	$716	$718	$71a	$71c	$71e
$6f8	$6fa	$6fc	$6fe	$700	$702	$704	$706	$708	$70a	$70c	$70e	$710	$712	$714	$716	$718	$71a	$71c	$71e	$720	$722	$724	$726	$728	$72a	$72c	$72e	$730	$732	$734	$736	$738	$73a	$73c	$73e	$740	$742	$744	$746
$720	$722	$724	$726	$728	$72a	$72c	$72e	$730	$732	$734	$736	$738	$73a	$73c	$73e	$740	$742	$744	$746	$748	$74a	$74c	$74e	$750	$752	$754	$756	$758	$75a	$75c	$75e	$760	$762	$764	$766	$768	$76a	$76c	$76e
$748	$74a	$74c	$74e	$750	$752	$754	$756	$758	$75a	$75c	$75e	$760	$762	$764	$766	$768	$76a	$76c	$76e	$770	$772	$774	$776	$778	$77a	$77c	$77e	$780	$782	$784	$786	$788	$78a	$78c	$78e	$790	$792	$794	$796
$770	$772	$774	$776	$778	$77a	$77c	$77e	$780	$782	$784	$786	$788	$78a	$78c	$78e	$790	$792	$794	$796	$798	$79a	$79c	$79e	$7a0	$7a2	$7a4	$7a6	$7a8	$7aa	$7ac	$7ae	$7b0	$7b2	$7b4	$7b6	$7b8	$7ba	$7bc	$7be
$798	$79a	$79c	$79e	$7a0	$7a2	$7a4	$7a6	$7a8	$7aa	$7ac	$7ae	$7b0	$7b2	$7b4	$7b6	$7b8	$7ba	$7bc	$7be	$7c0	$7c2	$7c4	$7c6	$7c8	$7ca	$7cc	$7ce	$7d0	$7d2	$7d4	$7d6	$7d8	$7da	$7dc	$7de	$7e0	$7e2	$7e4	$7e6
$7c0	$7c2	$7c4	$7c6	$7c8	$7ca	$7cc	$7ce	$7d0	$7d2	$7d4	$7d6	$7d8	$7da	$7dc	$7de	$7e0	$7e2	$7e4	$7e6	$7e8	$7ea	$7ec	$7ee	$7f0	$7f2	$7f4	$7f6	$7f8	$7fa	$7fc	$7fe	$800	$802	$804	$806	$808	$80a	$80c	$80e

Er zijn twee zaken op te merken in de bovenstaande tabel: ten eerste vordert het adres met twee bytes per tekencel, omdat twee bytes nodig zijn om elk teken te definiëren. Ten tweede neemt het startadres van elke schermregel nog steeds slechts met 40 ($28 in hexadecimaal) toe. Dit is niet wat we eigenlijk willen, want het betekent dat de eerste helft van de vorige regel opnieuw wordt weergegeven op elke huidige regel. Dit wordt opgelost door het aantal bytes dat per schermregel wordt overgeslagen in te stellen via $D058 (LSB) en $D059 (MSB). In dit geval willen we het aantal over te slaan bytes per regel te verhogen van 40 bytes naar 80 bytes. Dit kan door $D058 in te stellen op 80 ($50 in hexadecimaal) en $D059 op 0. Dit geeft een scherminrichting als volgt:

$400	$402	$404	$406	$408	$40a	$40c	$40e	$410	$412	$414	$416	$418	$41a	$41c	$41e	$420	$422	$424	$426	$428	$42a	$42c	$42e	$430	$432	$434	$436	$438	$43a	$43c	$43e	$440	$442	$444	$446	$448	$44a	$44c	$44e
$450	$452	$454	$456	$458	$45a	$45c	$45e	$460	$462	$464	$466	$468	$46a	$46c	$46e	$470	$472	$474	$476	$478	$47a	$47c	$47e	$480	$482	$484	$486	$488	$48a	$48c	$48e	$490	$492	$494	$496	$498	$49a	$49c	$49e
$4a0	$4a2	$4a4	$4a6	$4a8	$4aa	$4ac	$4ae	$4b0	$4b2	$4b4	$4b6	$4b8	$4ba	$4bc	$4be	$4c0	$4c2	$4c4	$4c6	$4c8	$4ca	$4cc	$4ce	$4d0	$4d2	$4d4	$4d6	$4d8	$4da	$4dc	$4de	$4e0	$4e2	$4e4	$4e6	$4e8	$4ea	$4ec	$4ee
$4f0	$4f2	$4f4	$4f6	$4f8	$4fa	$4fc	$4fe	$500	$502	$504	$506	$508	$50a	$50c	$50e	$510	$512	$514	$516	$518	$51a	$51c	$51e	$520	$522	$524	$526	$528	$52a	$52c	$52e	$530	$532	$534	$536	$538	$53a	$53c	$53e
$540	$542	$544	$546	$548	$54a	$54c	$54e	$550	$552	$554	$556	$558	$55a	$55c	$55e	$560	$562	$564	$566	$568	$56a	$56c	$56e	$570	$572	$574	$576	$578	$57a	$57c	$57e	$580	$582	$584	$586	$588	$58a	$58c	$58e
$590	$592	$594	$596	$598	$59a	$59c	$59e	$5a0	$5a2	$5a4	$5a6	$5a8	$5aa	$5ac	$5ae	$5b0	$5b2	$5b4	$5b6	$5b8	$5ba	$5bc	$5be	$5c0	$5c2	$5c4	$5c6	$5c8	$5ca	$5cc	$5ce	$5d0	$5d2	$5d4	$5d6	$5d8	$5da	$5dc	$5de
$5e0	$5e2	$5e4	$5e6	$5e8	$5ea	$5ec	$5ee	$5f0	$5f2	$5f4	$5f6	$5f8	$5fa	$5fc	$5fe	$600	$602	$604	$606	$608	$60a	$60c	$60e	$610	$612	$614	$616	$618	$61a	$61c	$61e	$620	$622	$624	$626	$628	$62a	$62c	$62e
$630	$632	$634	$636	$638	$63a	$63c	$63e	$640	$642	$644	$646	$648	$64a	$64c	$64e	$650	$652	$654	$656	$658	$65a	$65c	$65e	$660	$662	$664	$666	$668	$66a	$66c	$66e	$670	$672	$674	$676	$678	$67a	$67c	$67e
$680	$682	$684	$686	$688	$68a	$68c	$68e	$690	$692	$694	$696	$698	$69a	$69c	$69e	$6a0	$6a2	$6a4	$6a6	$6a8	$6aa	$6ac	$6ae	$6b0	$6b2	$6b4	$6b6	$6b8	$6ba	$6bc	$6be	$6c0	$6c2	$6c4	$6c6	$6c8	$6ca	$6cc	$6ce
$6d0	$6d2	$6d4	$6d6	$6d8	$6da	$6dc	$6de	$6e0	$6e2	$6e4	$6e6	$6e8	$6ea	$6ec	$6ee	$6f0	$6f2	$6f4	$6f6	$6f8	$6fa	$6fc	$6fe	$700	$702	$704	$706	$708	$70a	$70c	$70e	$710	$712	$714	$716	$718	$71a	$71c	$71e
$720	$722	$724	$726	$728	$72a	$72c	$72e	$730	$732	$734	$736	$738	$73a	$73c	$73e	$740	$742	$744	$746	$748	$74a	$74c	$74e	$750	$752	$754	$756	$758	$75a	$75c	$75e	$760	$762	$764	$766	$768	$76a	$76c	$76e
$770	$772	$774	$776	$778	$77a	$77c	$77e	$780	$782	$784	$786	$788	$78a	$78c	$78e	$790	$792	$794	$796	$798	$79a	$79c	$79e	$7a0	$7a2	$7a4	$7a6	$7a8	$7aa	$7ac	$7ae	$7b0	$7b2	$7b4	$7b6	$7b8	$7ba	$7bc	$7be
$7c0	$7c2	$7c4	$7c6	$7c8	$7ca	$7cc	$7ce	$7d0	$7d2	$7d4	$7d6	$7d8	$7da	$7dc	$7de	$7e0	$7e2	$7e4	$7e6	$7e8	$7ea	$7ec	$7ee	$7f0	$7f2	$7f4	$7f6	$7f8	$7fa	$7fc	$7fe	$800	$802	$804	$806	$808	$80a	$80c	$80e
$810	$812	$814	$816	$818	$81a	$81c	$81e	$820	$822	$824	$826	$828	$82a	$82c	$82e	$830	$832	$834	$836	$838	$83a	$83c	$83e	$840	$842	$844	$846	$848	$84a	$84c	$84e	$850	$852	$854	$856	$858	$85a	$85c	$85e
$860	$862	$864	$866	$868	$86a	$86c	$86e	$870	$872	$874	$876	$878	$87a	$87c	$87e	$880	$882	$884	$886	$888	$88a	$88c	$88e	$890	$892	$894	$896	$898	$89a	$89c	$89e	$8a0	$8a2	$8a4	$8a6	$8a8	$8aa	$8ac	$8ae
$8b0	$8b2	$8b4	$8b6	$8b8	$8ba	$8bc	$8be	$8c0	$8c2	$8c4	$8c6	$8c8	$8ca	$8cc	$8ce	$8d0	$8d2	$8d4	$8d6	$8d8	$8da	$8dc	$8de	$8e0	$8e2	$8e4	$8e6	$8e8	$8ea	$8ec	$8ee	$8f0	$8f2	$8f4	$8f6	$8f8	$8fa	$8fc	$8fe
$900	$902	$904	$906	$908	$90a	$90c	$90e	$910	$912	$914	$916	$918	$91a	$91c	$91e	$920	$922	$924	$926	$928	$92a	$92c	$92e	$930	$932	$934	$936	$938	$93a	$93c	$93e	$940	$942	$944	$946	$948	$94a	$94c	$94e
$950	$952	$954	$956	$958	$95a	$95c	$95e	$960	$962	$964	$966	$968	$96a	$96c	$96e	$970	$972	$974	$976	$978	$97a	$97c	$97e	$980	$982	$984	$986	$988	$98a	$98c	$98e	$990	$992	$994	$996	$998	$99a	$99c	$99e
$9a0	$9a2	$9a4	$9a6	$9a8	$9aa	$9ac	$9ae	$9b0	$9b2	$9b4	$9b6	$9b8	$9ba	$9bc	$9be	$9c0	$9c2	$9c4	$9c6	$9c8	$9ca	$9cc	$9ce	$9d0	$9d2	$9d4	$9d6	$9d8	$9da	$9dc	$9de	$9e0	$9e2	$9e4	$9e6	$9e8	$9ea	$9ec	$9ee
$9f0	$9f2	$9f4	$9f6	$9f8	$9fa	$9fc	$9fe	$a00	$a02	$a04	$a06	$a08	$a0a	$a0c	$a0e	$a10	$a12	$a14	$a16	$a18	$a1a	$a1c	$a1e	$a20	$a22	$a24	$a26	$a28	$a2a	$a2c	$a2e	$a30	$a32	$a34	$a36	$a38	$a3a	$a3c	$a3e
$a40	$a42	$a44	$a46	$a48	$a4a	$a4c	$a4e	$a50	$a52	$a54	$a56	$a58	$a5a	$a5c	$a5e	$a60	$a62	$a64	$a66	$a68	$a6a	$a6c	$a6e	$a70	$a72	$a74	$a76	$a78	$a7a	$a7c	$a7e	$a80	$a82	$a84	$a86	$a88	$a8a	$a8c	$a8e
$a90	$a92	$a94	$a96	$a98	$a9a	$a9c	$a9e	$aa0	$aa2	$aa4	$aa6	$aa8	$aaa	$aac	$aae	$ab0	$ab2	$ab4	$ab6	$ab8	$aba	$abc	$abe	$ac0	$ac2	$ac4	$ac6	$ac8	$aca	$acc	$ace	$ad0	$ad2	$ad4	$ad6	$ad8	$ada	$adc	$ade
$ae0	$ae2	$ae4	$ae6	$ae8	$aea	$aec	$aee	$af0	$af2	$af4	$af6	$af8	$afa	$afc	$afe	$b00	$b02	$b04	$b06	$b08	$b0a	$b0c	$b0e	$b10	$b12	$b14	$b16	$b18	$b1a	$b1c	$b1e	$b20	$b22	$b24	$b26	$b28	$b2a	$b2c	$b2e
$b30	$b32	$b34	$b36	$b38	$b3a	$b3c	$b3e	$b40	$b42	$b44	$b46	$b48	$b4a	$b4c	$b4e	$b50	$b52	$b54	$b56	$b58	$b5a	$b5c	$b5e	$b60	$b62	$b64	$b66	$b68	$b6a	$b6c	$b6e	$b70	$b72	$b74	$b76	$b78	$b7a	$b7c	$b7e
$b80	$b82	$b84	$b86	$b88	$b8a	$b8c	$b8e	$b90	$b92	$b94	$b96	$b98	$b9a	$b9c	$b9e	$ba0	$ba2	$ba4	$ba6	$ba8	$baa	$bac	$bae	$bb0	$bb2	$bb4	$bb6	$bb8	$bba	$bbc	$bbe	$bc0	$bc2	$bc4	$bc6	$bc8	$bca	$bcc	$bce

Het is mogelijk om Super-Uitgebreide Attribuutmodus te gebruiken vanuit C65-modus, door het scherm op 80 kolommen in te stellen, omdat de C65-ROM 2 KB reserveert voor zowel het schermgeheugen als het kleurengeheugen, en automatisch $D058 en $D059 op de juiste waarde zet voor $40 \times 2 = 80$ bytes per schermregel. De gebruiker hoeft elk tekenpaar slechts als één Super-Uitgebreid Attribuutteken te behandelen en Super-Uitgebreide Attribuutmodus in te schakelen, zoals hierboven beschreven.

Omdat er paren van kleurengeheugen- en schermgeheugenbytes worden gebruikt om elk teken te definiëren, moet voorzichtig te werk worden gegaan bij het initialiseren en bewerken van het scherm. Een goede aanpak is om de tekstkleur op zwart in te stellen (kleurcode 0) en vervolgens het scherm te vullen met @-tekens, omdat dat tekencode 0 is. Vervolgens zijn er verschillende manieren om het scherm te bewerken. U kunt het normale PRINT-commando gebruiken en zorgvuldig tekenreeksen samenstellen die de juiste waarden in elk scherm- en kleurenbytepaar schrijven. Een andere aanpak is om de BANK- en POKE-commando's te gebruiken om de inhoud van het schermgeheugen en kleurengeheugen direct te instellen.

Het beheren van een scherm in Super-Uitgebreide Attribuutmodus op deze manier met BASIC 65 is natuurlijk eerder een noodoplossing en wordt alleen voorgesteld als een relatief eenvoudige manier om te beginnen met experimenteren. U wilt bijna zeker snel overschakelen op aangepaste schermverwerkingscode, het liefst in assembleertaal, om schermen in Super-Uitgebreide Attribuutmodus te bewerken. De aanpak met BASIC 65 kan echter behoorlijk krachtig zijn, omdat het bestaande scrollen van het scherm en andere bewerkingen beschikbaar blijven.

XXX Voorbeeldprogramma

De volgende beschrijvingen gaan ervan uit dat u een van de hierboven beschreven methoden heeft geïmplementeerd om het schermgeheugen en kleurengeheugen in te stellen.

Volkleur-tekstmodus (256 kleuren per teken) (FCM)

In de normale VIC-II/III-tekstmodus wordt één byte gebruikt voor elke pixelrij in een teken. Ter herinnering aan de werking van die modes: in hoge-resolutiemodus is elke pixel ofwel de achtergrond- ofwel de voorgrondkleur, gebaseerd op één bit in de byte. Kleurenmodus gebruikt twee bits om tussen vier mogelijke kleuren te kiezen, maar omdat er nog steeds slechts 8 bits beschikbaar zijn om elke rij van 8 pixels te beschrijven, heeft elk pixelpaar dezelfde kleur. De volkleur-tekstmodus van de VIC-IV schaft deze beperkingen op en maakt het mogelijk om elke pixel van een teken te kiezen uit 256 kleuren van de primaire of alternatieve paletbank, zonder horizontale resolutie op te offeren.

Hiertoe heeft elk teken nu 64 bytes aan gegevens nodig. Het adres van de gegevens is 64 $\times$ het tekennummer, ongeacht het adres van de tekenset. FCM hoort normaal gesproken te worden gebruikt met Super-Uitgebreide Attribuutmodus (SEAM), zodat meer dan 256 unieke tekens kunnen worden aangesproken. Omdat SEAM de keuze van 8.192 unieke tekens mogelijk maakt, kunnen FCM-tekengegevens overal in de eerste 512 KB chip-RAM worden geplaatst (let er echter op dat de meeste MEGA65-modellen slechts 384 KB chip-RAM hebben).

Let erop dat pixelwaarde $ff niet direct de bijbehorende kleurcode selecteert. In plaats daarvan selecteert deze de kleurcode die is gedefinieerd door het kleurengeheugen.

Nibble-kleuren-tekstmodus (16 kleuren per teken) (NCM)

De Nibble-kleuren-modus (NCM) voor tekst lijkt op Volkleur-tekstmodus, met als verschil dat elke byte gegevens twee pixels beschrijft met elk 4 bits. Dit maakt NCM uniek, omdat de tekens 16 pixels breed zullen zijn in plaats van de gebruikelijke 8 pixels. Dit kan worden gebruikt om kleurrijke weergaven te maken, zonder zoveel geheugen te gebruiken als FCM, omdat er minder tekens nodig zijn om het scherm te bedekken. In tegenstelling tot de multicolourmodus (MCM) van de VIC-II heeft deze modus geen verlies van horizontale resolutie tot gevolg.

Bij NCM komen de onderste vier bits van de pixelkleur van de bovenste of onderste vier bits van de pixelgegevens. De bovenste vier bits van de kleurcode komen van de kleurengeheugengegevens van het weergegeven teken. Hierdoor is het mogelijk om alle paletingangen in NCM te gebruiken, hoewel de beperking van 16 kleuren per teken blijft bestaan. Vergelijkbaar met het gedrag van FCM selecteert pixelgegevenswaarde $f de pixelkleur die is ingesteld in het kleurengeheugen.

Een verder voordeel van NCM is dat het minder buscycli per pixel gebruikt dan FCM, omdat er minder tekengegevens-ophaalbewerkingen per rasterlijn hoeven plaats te vinden. Samen met de verminderde geheugenbehoeften maakt dit NCM bijzonder

geschikt voor het maken van kleurrijke grafische lagen. Hiermee kan de VIC-IV arcade-achtige weergaven met meer kleuren tonen dan bij veel 16-bits computers.

XXX

Alphablending / Anti-aliasing

De VIC-IV ondersteunt het mengen van tekens met de achtergrondkleur, waardoor effecten zoals anti-aliased lettertype-weergave mogelijk zijn. Het mengen is mogelijk op per-teken-basis. Het wordt ingeschakeld voor een specifiek teken als aan de volgende voorwaarden wordt voldaan:

1. Het *ALPHAEN*-signaal is ingesteld in bit 7 van $D054.
2. Het *CHR16*-signaal is ingesteld in bit 0 van $D054 om Super-Uitgebreide Attributmodus (SEAM) in te schakelen.
3. Volkleur-tekstmodus (FCM) is ingeschakeld voor het teken.

Indien alphablending is ingeschakeld voor een teken, worden de 8-bit pixelwaarden behandeld als alphawaarden in plaats van paletindices. De werkelijke pixelkleur wordt bepaald door de achtergrondkleur te mengen met de voorgrondkleur van het teken zoals gedefinieerd in het kleurengeheugen. Een alphawaarde van 0 staat voor volledige transparantie, waarbij alleen de achtergrondkleur van die pixel wordt weergegeven.

Merk op dat de alphablending alleen wordt toegepast tussen de achtergrondkleur en de voorgrondkleur van het teken. Dit betekent dat tekens achter het huidige teken feitelijk niet zichtbaar zullen zijn (tekenlagen kunnen worden samengesteld via GOTOX-herpositionering). Programmeurs moeten er echter van uitgaan dat het mengen met eerdere lagen in een toekomstige implementatie zal worden ondersteund. Om problemen met een dergelijke wijziging te voorkomen, plaatst u geen tekens achter een teken waarvoor alphablending is ingeschakeld.

Tekens omdraaien

XXX

Variabele-breedte lettertypen

Er zijn 4 bits waarmee pixels van de rechterrand van tekens kunnen worden weggesneden wanneer deze worden weergegeven. Dit heeft het effect dat tekens smaller worden. Dit kan nuttig zijn voor aantrekkelijkere tekstweergave, waarbij smallere tekens zoals "i" minder ruimte innemen dan bredere tekens zoals "m", zonder dat een bitmap-display nodig is. Deze functie kan worden gebruikt om variabele-breedte tekstweergave zeer efficiënt te maken, zowel wat betreft geheugengebruik als verwerkingstijd.

Deze functie kan worden gecombineerd met de volkleur-tekstmodus, de alfablendmodus en de 4-bits-per-pixel-modus om tekens mogelijk te maken met 15 intensiteitsniveaus tussen de achtergrond- en voorgrondkleur, en tot 16 pixels breed. Bovendien kan het GOTO-bit worden gebruikt om negatieve kerning toe te passen,

zodat tekenparen zoals A en T geen overbodige witruimte tussen zich hebben wanneer ze naast elkaar worden afgedrukt. Het zorgvuldig gebruik van deze functies kan leiden tot indrukwekkende tekstweergave, vergelijkbaar met die op moderne 32-bits en 64-bits systemen, maar die nog steeds efficiënt genoeg is om te worden geïmplementeerd op een relatief beperkt systeem zoals de MEGA65. De presentatiesoftware "MegaWAT!?"" voor de MEGA65 maakt gebruik van verschillende van deze functies om zijn aantrekkelijke, antialiased proportionele tekstweergave op dia's te produceren.

XXX MEGAWat!?!-screenshot

XXX Voorbeeldprogramma

Raster Re-write Buffer

Als het GOTO-bit is gezet voor een teken in Super-Uitgebreide Attribuutmodus, wordt in plaats van het teken te tekenen de positie op de rasterlijn teruggedraaid (of vooruitgeschoven) naar de pixelpositie die is opgegeven in de onderste 10 bits van de schermgeheugenbytes. Als het vertical flip-bit is gezet, heeft dit de alternatieve betekenis dat de achtergrondkleur niet wordt getekend. Deze combinatie kan worden gebruikt om tekstmateriaal over ander tekstmateriaal te plaatsen, wat een grove aanvulling vormt op de 8 hardware-sprites. De hoeveelheid materiaal is alleen beperkt door de rastertijd van de VIC-IV. Enige experimentatie is vereist om te bepalen hoeveel er in PAL- en NTSC-modus kan worden bereikt.

Als het GOTO-bit is gezet voor een teken en de tekenbreedte-verminderingsbits ook zijn gezet, worden deze geïnterpreteerd als een Y-offset die wordt opgeteld bij het tekengeheugenadres, maar alleen in Volkleurmodus. Instellen van Y=1 zorgt ervoor dat de tekendata worden opgehaald van 8 bytes verder, d.w.z. de eerste rij tekendata komt van het adres waar normaal de tweede rij tekendata zou worden opgehaald. Voor hogere waarden worden de tekendata opgehaald van verderop in de tekens. Door tekens zorgvuldig in het geheugen te rangschikken, kan deze functie worden gebruikt om vrije verticale positionering van soft sprites mogelijk te maken, zonder dat de tekendata hoeven te worden gekopieerd.

Als het GOTO-bit is gezet voor een teken en het Nybl Colour Mode (NCM)-bit ook is gezet, wordt het tweede kleurengeheugenbyte voor dat teken gebruikt om de rijmaskerbits in te stellen. Voor elk bit dat in het rijmasker is gezet, wordt de overeenkomstige rij tekens in de regel niet weergegeven. Dit kan worden gecombineerd met de Y-offsetfunctie om effectief een soepele verticale schuifregeling per teken mogelijk te maken.

De mogelijkheid om meerdere lagen tekst en grafisch materiaal te tekenen is zeer krachtig. Dit kan bijvoorbeeld worden gebruikt om meerdere overlappende lagen van apart schuifbare grafische elementen te bieden. Dit geeft veel van de voordelen van bitplane-gebaseerde playfields op andere computers, zoals de Amiga, maar zonder de nadelen van bitplanes.

Een goede introductie van de Raster Re-write Buffer en het gebruik ervan is te vinden in deze video:

https://www.youtube.com/watch?v=00bm5uBeBos&feature=youtu.be

Een belangrijk aspect van de RRB is dat de VIC-IV alleen de tekendata weergeeft die zich links van, en inclusief, het laatst getekende teken bevindt. Dit betekent dat wanneer u de GOTO-token gebruikt om meerdere lagen grafisch materiaal te overschrijven, u ofwel moet zorgen dat de laatste laag tot aan de rechterrand van het scherm reikt, ofwel een GOTO-token moet toevoegen dat de renderpositie naar de rechterrand van het scherm verplaatst.

XXX Voorbeeldprogramma

SPRITES

VIC-II/III-spritebesturing

De besturing van sprites voor C64 / VIC-II/III-compatibiliteit is ongewijzigd ten opzichte van de C64. De enige praktische verschillen zijn zeer gering. In het bijzonder gebruikt de VIC-IV een ringbuffer voor de gegevens van elke sprite bij het renderen van een rasterlijn. Dit betekent dat een sprite meerdere keren per rasterlijn kan worden weergegeven, wat horizontale multiplexing mogelijk maakt.

Uitgebreide sprite-afbeeldingssets

Op de VIC-II en VIC-III moeten alle sprites hun afbeeldingsdata op een bepaald moment uit één enkel 16KB-geheugengebied halen. Dit beperkt het aantal verschillende sprite-afbeeldingen tot 256, omdat elke sprite-afbeelding 64 bytes in beslag neemt. In de praktijk moet hetzelfde 16KB-geheugengebied ook bitmap-, tekst- of bitplane-data bevatten, wat het aantal sprite-afbeeldingen dat tegelijk kan worden gebruikt aanzienlijk beperkt.

De VIC-IV heft deze beperking op doordat spritedata overal in het geheugen kunnen worden geplaatst, zij het nog steeds op 64-byte-grenzen. Dit wordt gedaan door het SPRPTR16-signaal in te stellen (bit 7, $D06E, decimaal 53358), wat de VIC-IV vertelt dat er twee bytes per spritepointer in plaats van één moeten worden verwacht. Deze adressen zijn dan absolute adressen en negeren de 16KB VIC-II-bankselectielogica. Er zijn dus 16 bytes nodig in plaats van 8 bytes. De lijst met pointers kan ook overal in het geheugen worden geplaatst door de SPRPTRADR ($D06C - $D06D, 53356 - 53357 decimaal) en SPRPTRBNK-signalen (bits 0 - 6, $D06E, 53358 decimaal) in te stellen. Hierdoor kunnen spritedata overal in de eerste 4MB RAM worden geplaatst en de spritepointerlijst overal in de eerste 8MB RAM. Let op dat typische installaties van de VIC-IV slechts 384KB aangesloten RAM hebben, zodat deze beperkingen geen praktisch effect hebben. De bovenste bits van het SPRPTRBNK-signaal moeten echter op nul worden gezet om problemen met voorwaartse compatibiliteit te voorkomen.

Een reden voor ondersteuning van meer sprite-afbeeldingen is dat sprites op de VIC-IV meer dan één 64-byte-afbeeldingsslot kunnen vereisen. Het inschakelen van de modus voor extra-brede sprites betekent bijvoorbeeld dat een sprite $8 \times 21 = 168$ bytes

nodig heeft en dus vier VIC-II-stijl 64-byte-spriteafbeeldingsslots inneemt. Wanneer sprites met variabele hoogte worden gebruikt, kan dit oplopen tot 8×255 = 2.040 bytes per sprite.

Variabele spritegrootte

Sprites kunnen met de VIC-IV een van drie breedtes hebben:

1. Normale VIC-II-breedte (24 pixels breed).
2. Extra Breed, waarbij 64 bits (8 bytes) data per rasterlijn worden gebruikt in plaats van de 24 van de VIC-II. Dit resulteert in sprites die 64 pixels breed zijn, tenzij Volkleur-spritemodus is geselecteerd voor een sprite, in welk geval de sprite 64 bits ÷ 4 bits per pixel = 16 pixels breed zal zijn.
3. Tegelmodus, waarbij de sprite wordt herhaald getekend tot het einde van de rasterlijn.
   Tegelmodus moet normaal gesproken alleen worden gebruikt met extra-brede sprites, omdat het tegelen altijd plaatsvindt met 64 bits spritedata per regel. Het inschakelen van tegelmodus met normale 24 bits brede mono- of multicolour-spritedata tekent 2 en 2/3 rijen spritedata als één regel, zelfs als de sprite niet in extra-brede modus staat, wat tot verwarrende weergave leidt.

Om een sprite 64 pixels breed te maken (of 16 pixels als Volkleur-spritemodus is ingeschakeld), zet u het overeenkomstige bit voor de sprite in het SPRX64EN-register op ($D057, 53335 decimaal). Het inschakelen van Volkleurmodus voor een sprite schakelt impliciet de uitgebreide breedtemodus in en zorgt ervoor dat deze sprites 16 pixels breed zijn.

Op vergelijkbare wijze kunnen sprites diverse hoogtes hebben: sprites zijn ofwel 21 pixels hoog zoals bij de VIC-II, ofwel, als het overeenkomstige bit voor de sprite is ingeschakeld in het SPRHGTEN-signaal ($D055, 53333 decimaal), heeft die sprite de hoogte in pixels die is ingesteld in het SPRHGT-register ($D056, 53334 decimaal), van 0 tot 255. Let op dat alle sprites met SPRHGTEN ingeschakeld dezelfde hoogte delen. Een sprite kan altijd de onderkant van zijn afbeeldingsdata transparant laten.

Om tegelmodus in te schakelen voor een sprite, zet u het bijbehorende bit van SPRTILEN. Voor sprites 0 tot en met 3 zet u bits 4 tot en met 7 van $D04D (53325 decimaal). Voor sprites 4 tot en met 7 zet u bits 4 tot en met 7 van $D04F (55327 decimaal).

Variabele sprite-resolutie

Standaard hebben sprites dezelfde resolutie als op de VIC-II, oftewel elke sprite-pixel is twee fysieke pixels breed en hoog. Het is echter mogelijk om sprites de native resolutie te laten gebruiken, waarbij sprite-pixels één fysiek pixel breed en/of hoog zijn. Dit wordt bereikt door het relevante bit voor de sprite in het SPRENV400-register ($D076, 53366 decimaal) te zetten, zodat de verticale resolutie per sprite wordt vergroot. De horizontale resolutie van alle sprites is ofwel de normale VIC-II-resolutie, of wanneer

het SPR640-signaal is gezet (bit 4 van $D054, 53332 decimaal), hebben sprites dezelfde horizontale resolutie als de fysieke pixels van het beeldscherm.

Sprite-paletbank

De VIC-IV heeft vier paletbanken, in vergelijking met de enkele paletbank van de VIC-III. De VIC-IV maakt het mogelijk om afzonderlijke paletbanken te selecteren voor bitmap-/tekstgraphics en voor sprites. Dit maakt het eenvoudig om zeer kleurrijke beeldschermen te creëren, waarbij sprites andere kleuren hebben dan de rest van het beeldscherm, of om paletanimatie te gebruiken voor interessante visuele effecten in sprites, zonder het palet dat door andere elementen van het beeldscherm wordt gebruikt te verstoren.

De sprite-paletbank wordt geselecteerd door het SPRPALSEL-signaal in bits 2 en 3 van het register $D070 (53360 decimaal) te zetten. Het is mogelijk om deze op dezelfde bank als de bitmap-/tekstweergave in te stellen, of om een andere paletbank te selecteren. De selectie van de paletbank wordt onmiddellijk van kracht. Vergeet niet dat u, om een palet te kunnen wijzigen, het palet ook in het bereik van de paletbankregisters op $D100 - $D3FF moet plaatsen door het MAPEDPAL-signaal in bits 6 en 7 van $D070 te zetten.

Vollekleuren-spritemodus

Naast de monochrome en multicolour-modi ondersteunt de VIC-IV een nieuwe vollekleuren-spritemodus. In deze modus worden vier bits gebruikt om elke sprite-pixel te coderen. In tegenstelling tot de multicolour-modus, waarin paren van bits paren van pixels coderen, behouden de pixels in de vollekleurenmodus hun normale horizontale resolutie. De kleur nul wordt als transparant beschouwd. Indien u zwart wilt gebruiken in een vollekleurensprite, moet u de paletbank die voor sprites is geselecteerd zodanig configureren dat een van de 15 kleuren voor de betreffende sprite de kleur zwart codeert.

De vollekleuren-spritemodus kan per sprite worden geselecteerd door het bijbehorende bit in het SPR16EN-register ($D06B, 53355 decimaal) te zetten.

Om de acht sprites elk 15 unieke kleuren te laten hebben, wordt de spritekleur weergegeven met behulp van de paletingang die overeenkomt met: $spritenummer \times 16 + nibblewaarde$, waarbij $spritenummer$ het nummer van de sprite is (van 0 tot en met 7) en $nibblewaarde$ de waarde is van het halfbyte dat de spritegegevens voor de pixel bevat. Indien bovendien de bitplane-modus voor deze sprite is ingeschakeld, wordt 128 opgeteld bij de kleurwaarde, waardoor het eenvoudig wordt om tussen twee kleurenschema's voor een gegeven sprite te schakelen door slechts één bit in het SPRBPMEN-register te wijzigen.

Omdat de vollekleuren-spritemodus vier bits per pixel vereist, zijn sprites slechts zes pixels breed, tenzij extra-breed voor een sprite is ingeschakeld, in welk geval de sprite 16 pixels breed is. Tegelmodus kan gecombineerd worden met de vollekleuren-spritemodus, en heeft dan tot gevolg dat de 16 vollekleuren-pixels van de sprite worden herhaald tot aan het einde van de rasterlijn.

Het volgende BASIC-programma tekent een vollekleurensprite in C64- of C65-modus:

```
10 PRINT CHR$(147)
20 REM C65-/C64-MODUS DETECTEREN
30 IF PEEK(53272) AND 32 THEN GOTO 100
40 POKE 53295,ASC("6"):POKE 53295,ASC("$")
100 REM Sprite instellen
110 AD=4096                          :REM $1000-spriteadres
120 TC=10                            :REM transparante kleur
130 SPR=PEEK(53356)+PEEK(53357)*256 :REM sprite-tabeladres ophalen
140 POKE SPR,AD/64                   :REM spriteadres instellen
150 FOR I=AD TO AD+168               :REM sprite met TC wissen
160 POKE I,TC+TC*16                  :REM een byte = 2 pixels
170 NEXT
180 POKE 53287,TC                    :REM transparante kleur instellen
190 POKE 53248,100                   :REM sprite op 100,100 zetten...
200 POKE 53249,100                   :REM op het scherm op 100,100
210 POKE 53355,1                     :REM sprite 0 16-kleurig maken
220 POKE 53335,1                     :REM sprite 0 16X4-bits laten gebruiken
230 POKE 53269,1                     :REM sprite 0 inschakelen
240 GOSUB 900                        :REM meerkleurensprite lezen
250 END

900 REM Sprite uit gegevens laden
910 READ N$:IF N$="END" THEN RETURN
920 GOSUB 1000                       :REM regel decoderen
930 GOTO 910
```

```
1000 REM Tekenreeks van nibbles in N$ op adres AD decoderen
1010 IF LEN(N$)<>16 THEN PRINT "ILLEGAL SPR DATA!":END
1020 FOR I=1 TO 16 STEP 2
1030 N=(ASC(MID$(N$,I,1))-ASC("@"))    :REM hoge nibble
1040 IF N<0 THEN N=TC                  :REM . is transparant
1050 M=(ASC(MID$(N$,I+1,1))-ASC("@"))  :REM lage nibble
1060 IF M<0 THEN M=TC                  :REM . is transparant
1070 POKE AD,(N AND 15)*16 + (M AND 15):REM 2 pixels instellen
1080 AD=AD+1                           :REM AD ophogen
1090 NEXT I
1100 RETURN

1998 REM Spritegegevens
1999 REM . = transparant, @-O = kleuren 0 tot 15
2000 DATA "..AAFF...HHCC..."
2010 DATA ".AAFF.....HHCC.."
2020 DATA "AAFF.......HHCC."
2030 DATA "AFF...@@@...HHC."
2040 DATA "FF..@@GGG@@..HH."
2050 DATA "..@@GGGGGGG@@..."
2060 DATA ".@GGGGGGGGGG@.."
2070 DATA ".@GGGGGGGGGG@.."
2080 DATA "@GGG@@GGG@@GGG@."
2090 DATA "@GG@GGGGGG@GG@."
2100 DATA "@GGGGGGGGGGGG@."
2110 DATA "@GGGGGBGBGGGGG@."
2120 DATA "@GGGBBBBBBBGGG@."
2130 DATA ".@GGGBBBGGGG@.."
2140 DATA ".@GGGGGBGGGGG@.."
2150 DATA "..@@GGGGGGG@@..."
2160 DATA "II..@@GGG@@..KK."
2170 DATA "DII...@@@...KKE."
2180 DATA "DDII.......KKEE."
2190 DATA ".DDII.....KKEE.."
2200 DATA "..DDII...KKEE..."
2210 DATA "END"
```

VIC-III ERRATA-NIVEAU

Er zijn een aantal gevallen waarin de VIC-III-chip in de bekende Commodore 65-prototypen niet overeenkomstig de specificatie functioneert, de specificatie onvoldoende detail bevat en de implementatie opmerkelijk inconsistent is, of het ontwerp zelf gebreken of inconsistenties vertoont. Het standaardgedrag van de VIC-IV is om de VIC-III in deze gevallen zo nauwkeurig mogelijk te emuleren. In sommige gevallen waarin het gedrag van de VIC-III tekortschiet, biedt de VIC-IV verbeterd gedrag, dat door software kan worden geselecteerd met behulp van het HWERRATA-register op $D08F.

Omdat deze correcties achterwaarts onverenigbaar zijn met de VIC-III en met eerdere versies van de VIC-IV en de MEGA65-core, moet software deze correcties expliciet activeren door het HWERRATA-register te zetten. Dit beschermt software tegen verdere wijzigingen die in toekomstige versies van de MEGA65-core kunnen worden geïntroduceerd. De beginstand van dit register is $00, wat volledige compatibiliteit met de VIC-III aanvraagt, inclusief het foutieve gedrag. De MEGA65-ROM zal dit register bij het starten van programma's altijd op $00 laten staan. Een programma kan HWERRATA instellen om een set correcties te activeren die de ontwikkelaar bij het schrijven van het programma kende, en achterwaarts onverenigbare correcties uit te sluiten die in latere versies kunnen worden geïntroduceerd.

Het aanvragen van een errata-niveau activeert alle correcties tot dat niveau. Er is bewust geen mogelijkheid om een niveau aan te vragen en specifieke correcties op lagere niveaus uit te sluiten. U moet uw programma zodanig schrijven dat het alle correcties tot het gevraagde niveau ondersteunt.

In gevallen waarin het activeren van een correctie het gedrag van een heet register wijzigt, zal het instellen van het errata-niveau hete-registerpropagatie *niet* activeren. Het programma moet de hete-registerpropagatie zelf activeren door een 1 te schrijven naar het HOTREG-register.

De tot nu toe geïmplementeerde errata-niveaus zijn als volgt:

VIC-III errata-niveaus

Niveau	Gecorrigeerd gedrag	Geïntroduceerd in
0	Geen correcties. Volledig VIC-III-compatibel.	N/A
1	**X-scrollpositie verschoven in de H640-modus.** De VIC-III geeft de X-scrollpositie voor het vloeiend scrollen van tekst ($D016 XSCL) onjuist weer in de H640-modus. De correctie verschuift de scrollpositie met 1 logische pixel (2 fysieke pixels) naar rechts. Dit wordt pas van kracht na de hot register-propagatie.	v0.96

vervolg ...

Niveau	Gecorrigeerd gedrag	Geïntroduceerd in
X	**Gedrag van de busy-vlag van de SD-kaart.** De busy-vlag van de SD-kaart (bit 1 van $D680) geeft aan of de SD-kaartcontroller op laag niveau bezig is. Door de toevoeging van de read-ahead-functionaliteit van de SD-kaartcontroller zal oudere software, die erop rekent dat dit bit vrijkomt na voltooiing van een leesbewerking, ten onrechte wachten totdat de volledige read-ahead-sequentie is afgerond. Dit kan er op zijn beurt toe leiden dat software ten onrechte meent dat het lezen van de sector is mislukt vanwege de langere benodigde tijd, of dat een ander leesverzoek niet onmiddellijk kan worden ingediend. Daarom geeft de busy-vlag van de SD-kaart onder dit errata-niveau niet aan of de SD-kaartcontroller een read-ahead-bewerking op de achtergrond voor een sector uitvoert. Op of boven dit errata-niveau wordt deze informatie niet verborgen: bit 0 komt vrij wanneer de gevraagde sector beschikbaar is, maar bit 1 blijft gezet zolang een read-ahead-bewerking doorgaat.	Nog niet geïmplementeerd

VIC-II / C64-REGISTERS

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D000	53248	S0X							
D001	53249	S0Y							
D002	53250	S1X							
D003	53251	S1Y							
D004	53252	S2X							
D005	53253	S2Y							
D006	53254	S3X							
D007	53255	S3Y							
D008	53256	S4X							
D009	53257	S4Y							
D00A	53258	S5X							
D00B	53259	S5Y							
D00C	53260	S6X							
D00D	53261	S6Y							
D00E	53262	S7X							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D00F	53263	S7Y							
D010	53264	SXMSB							
D011	53265	RC8	ECM	BMM	BLNK	RSEL	YSCL		
D012	53266	RC							
D013	53267	LPX							
D014	53268	LPY							
D015	53269	SE							
D016	53270	-		RST	MCM	CSEL	XSCL		
D017	53271	SEXY							
D018	53272	VS					CB		-
D019	53273	-				ILP	ISSC	ISBC	RIRQ
D01A	53274	-					MISSC	MISBC	MRIRQ
D01B	53275	BSP							
D01C	53276	SCM							
D01D	53277	SEXX							
D01E	53278	SSC							
D01F	53279	SBC							
D020	53280	-				BORDERCOL			
D021	53281	-				SCREENCOL			
D022	53282	-				MC1			
D023	53283	-				MC2			
D024	53284	-				MC3			
D025	53285	SPRMC0							
D026	53286	SPRMC1							
D027	53287	SPR0COL							
D028	53288	SPR1COL							
D029	53289	SPR2COL							
D02A	53290	SPR3COL							
D02B	53291	SPR4COL							
D02C	53292	SPR5COL							
D02D	53293	SPR6COL							
D02E	53294	SPR7COL							
D02F	53295	KEY							
D030	53296	-							C128-FAST

- **BLNK** Weergave inschakelen: 0 = weergave uitschakelen, 1 = weergave inschakelen
- **BMM** Bitmapmodus
- **BORDERCOL** Randkleur van het scherm (16 kleuren)
- **BSP** Bitstelling sprite-/achtergrondprioriteit

- **C128FAST** 2 MHz-selectie (voor C128 2 MHz-emulatie)
- **CB** Adreslocatie van de tekenset (× 1 KiB)
- **CSEL** 38/40-kolommenselectie
- **ECM** Uitgebreide achtergrondmodus
- **ILP** Lichtpen indicator of bevestiging
- **ISBC** Sprite-/bitmapbotsing indicator of bevestiging
- **ISSC** Sprite-/spritebotsing indicator of bevestiging
- **KEY** $00 en dan $00 schrijven om C64/VIC-II I/O-registers in te schakelen
- **LPX** Grove horizontale bundelpositie (was lichtpen X)
- **LPY** Grove verticale bundelpositie (was lichtpen Y)
- **MC1** Multikleur 1 (16 kleuren)
- **MC2** Multikleur 2 (16 kleuren)
- **MC3** Multikleur 3 (16 kleuren)
- **MCM** Multikleurmodus
- **MISBC** Masker sprite-/bitmapbotsing IRQ
- **MISSC** Masker sprite-/spritebotsing IRQ
- **MRIRQ** Masker rasterIRQ
- **RC** Rastervergelijking bits 0 tot 7
- **RC8** Rastervergelijking bit 8
- **RIRQ** Rastervergelijking indicator of bevestiging
- **RSEL** 24/25-rijselectie
- **RST** Schakelt de video-uitvoer op de MAX Machine(tm) VIC-II 6566 uit. Wordt genegeerd op normale C64-computers en de MEGA65
- **SBC** Indicatiebits voor sprite/voorgrondbotsing
- **SCM** Inschakelbits voor de sprite-multikleur
- **SCREENCOL** Schermkleur (16 kleuren)
- **SE** Inschakelbits voor sprites
- **SEXX** Inschakelbits voor horizontale sprite-vergroting
- **SEXY** Inschakelbits voor verticale sprite-vergroting
- **SNX** Horizontale positie van sprite N
- **SNY** Verticale positie van sprite N

- **SPRMC0** Sprite-multikleur 0
- **SPRMC1** Sprite-multikleur 1
- **SPRNCOL** Kleur van sprite N / transparantiekleur van de sprite in 16-kleurenmodus (lagere nibble)
- **SSC** Indicatiebits voor sprite/spritebotsing
- **SXMSB** MSB's van de horizontale sprite-positie
- **VS** Schermadres (× 1KiB)
- **XSCL** Horizontaal vloeiend scrollen
- **YSCL** 24/25 verticaal vloeiend scrollen

VIC-III / C65-REGISTERS

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D020	53280	BORDERCOL							
D021	53281	SCREENCOL							
D022	53282	MC1							
D023	53283	MC2							
D024	53284	MC3							
D025	53285	SPRMC0							
D026	53286	SPRMC1							
D02F	53295	KEY							
D030	53296	ROME	CROM9	ROMC	ROMA	ROM8	PAL	EXTSYNC	CRAM2K
D031	53297	H640	FAST	ATTR	BPM	V400	H1280	MONO	INT
D032	53298	BPENABLE							
D033	53299	B0ADODD			-	B0ADEVN			-
D034	53300	B1ADODD			-	B1ADEVN			-
D035	53301	B2ADODD			-	B2ADEVN			-
D036	53302	B3ADODD			-	B3ADEVN			-
D037	53303	B4ADODD			-	B4ADEVN			-
D038	53304	B5ADODD			-	B5ADEVN			-
D039	53305	B6ADODD			-	B6ADEVN			-
D03A	53306	B7ADODD			-	B7ADEVN			-
D03B	53307	BPCOMP							
D03C	53308	BPX							
D03D	53309	BPY							
D03E	53310	HPOS							
D03F	53311	VPOS							
D040	53312	B0PIX							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D041	53313	B1PIX							
D042	53314	B2PIX							
D043	53315	B3PIX							
D044	53316	B4PIX							
D045	53317	B5PIX							
D046	53318	B6PIX							
D047	53319	B7PIX							
D100 - D1FF	53504 - 53759	PALRED							
D200 - D2FF	53760 - 54015	PALGREEN							
D300 - D3FF	54016 - 54271	PALBLUE							

- **ATTR** Uitgebreide attributen en 8-bits kleuringangen inschakelen
- **BNPIX** Display Address Translater (DAT) Bitplane N-poort
- **BORDERCOL** Weergaverandkleur (256 kleuren)
- **BPCOMP** Complement-vlaggen voor bitplanes
- **BPENABLE** Inschakelbits voor bitplanes
- **BPM** Bitplane-modus
- **BPX** Bitplane X
- **BPY** Bitplane Y
- **BXADEVN** Even rijen van even Bitplane X-adres[15:13] (voor v200 wordt de even rij tweemaal getekend)
- **BXADODD** Oneven rijen van even Bitplane X-adres[15:13] (wordt alleen gebruikt in de v400-modus)
- **BYADEVN** Even rijen van oneven Bitplane Y-adres[15:13] (+64KB) (voor v200 wordt de even rij tweemaal getekend)
- **BYADODD** Oneven rijen van oneven Bitplane Y-adres[15:13] (+64KB) (wordt alleen gebruikt in de v400-modus)
- **CRAM2K** Mapt de 2e KB van het kleurengeheugen $DC00-$DFFF
- **CROM9** Selecteert tussen de C64- en C65-tekenset (niet geïmplementeerd)
- **EXTSYNC** Externe videosynchronisatie inschakelen (genlock-ingang)
- **FAST** C65 FAST-modus inschakelen (~3.5MHz)
- **H1280** 1280 horizontale pixels inschakelen (niet geïmplementeerd, schakelt PALEMU uit)

- **H640** C64 640 horizontale pixels / 80-kolommenmodus inschakelen
- **HPOS** Bitplane X-offset
- **INT** Interlaced modus van de VIC-III inschakelen
- **KEY** $A5 en dan $96 schrijven om de C65/VIC-III I/O-registers in te schakelen
- **MC1** Multikleur 1 (256 kleuren)
- **MC2** Multikleur 2 (256 kleuren)
- **MC3** Multikleur 3 (256 kleuren)
- **MONO** VIC-III MONO-composietvideo-uitvoer inschakelen (kleur indien uitgeschakeld)
- **PAL** Gebruikt ingangen van het PALETTE ROM (0) of het RAM (1) voor kleuren 0 - 15
- **PALBLUE** Blauwe paletwaarden (omgekeerde nibble-volgorde)
- **PALGREEN** Groene paletwaarden (omgekeerde nibble-volgorde)
- **PALRED** Rode paletwaarden (omgekeerde nibble-volgorde)
- **ROM8** Mapt de C65-ROM $8000
- **ROMA** Mapt de C65-ROM $A000
- **ROMC** Mapt de C65-ROM $C000
- **ROME** Mapt de C65-ROM $E000
- **SCREENCOL** Schermkleur (256 kleuren)
- **SPRMC0** Sprite-multikleur 0 (8-bits voor selectie van elke paletkleur)
- **SPRMC1** Sprite-multikleur 1 (8-bits voor selectie van elke paletkleur)
- **V400** 400 verticale pixels inschakelen (schakelt PALEMU uit)
- **VPOS** Bitplane Y-offset

VIC-IV / MEGA65-SPECIFIEKE REGISTERS

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D020	53280	BORDERCOL							
D021	53281	SCREENCOL							
D022	53282	MC1							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D023	53283	MC2							
D024	53284	MC3							
D025	53285	SPRMC0							
D026	53286	SPRMC1							
D02F	53295	KEY							
D048	53320	TBDRPOS							
D049	53321	SPRBPMEN				TBDRPOS			
D04A	53322	BBDRPOS							
D04B	53323	SPRBPMEN				BBDRPOS			
D04C	53324	TEXTXPOS							
D04D	53325	SPRTILEN				TEXTXPOS			
D04E	53326	TEXTYPOS							
D04F	53327	SPRTILEN				TEXTYPOS			
D050	53328	XPOSLSB							
D051	53329	NORRDEL	DBLRR	XPOSMSB					
D052	53330	FNRASTERLSB							
D053	53331	FNRST	SHDEMU	UPSCALE	BOL-DISALT	FNRASTERMSB			
D054	53332	ALPHEN	VFAST	PALEMU	SPRH640	SMTH	FCLRHI	FCLRLO	CHR16
D055	53333	SPRHGTEN							
D056	53334	SPRHGHT							
D057	53335	SPRX64EN							
D058	53336	LINESTEPLSB							
D059	53337	LINESTEPMSB							
D05A	53338	CHRXSCL							
D05B	53339	CHRYSCL							
D05C	53340	SDBDRWDLSB							
D05D	53341	HOTREG	RSTDELEN	SDBDRWDMSB					
D05E	53342	CHRCOUNT							
D05F	53343	SPRXSMSBS							
D060	53344	SCRNPTRLSB							
D061	53345	SCRNPTRMSB							
D062	53346	SCRNPTRBNK							
D063	53347	EXGLYPH	FCOLMCM	CHRCOUNT		SCRNPTRMB			
D064	53348	COLPTRLSB							
D065	53349	COLPTRMSB							
D068	53352	CHARPTRLSB							
D069	53353	CHARPTRMSB							
D06A	53354	CHARPTRBNK							
D06B	53355	SPR16EN							
D06C	53356	SPRPTRADRLSB							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D06D	53357	SPRPTRADRMSB							
D06E	53358	SPRPTR16	SPRPTRBNK						
D06F	53359	PALNTSC	VGAHDTV	RASLINE0					
D070	53360	MAPEDPAL		BTPALSEL		SPRPALSEL		ABTPALSEL	
D071	53361	BP16ENS							
D072	53362	SPRYADJ							
D073	53363	RASTERHEIGHT				ALPHADELAY			
D074	53364	SPRENALPHA							
D075	53365	SPRALPHAVAL							
D076	53366	SPRENV400							
D077	53367	SPRYMSBS							
D078	53368	SPRYSMSBS							
D079	53369	RASCMP							
D07A	53370	FNRST-CMP	EXTIRQS	NOBUG-COMPAT	CHARY16	SPTR-CONT	RASCMPMSB		
D07B	53371	DISPROWS							
D07C	53372	DEBUGC		VSYNCP	HSYNCP	NORRB-WRAP	BITPBANK		

- **ABTPALSEL** Bitmap-/tekstpaletbank van de VIC-IV (alternatief palet)
- **ALPHADELAY** Alfa-vertraging voor de compositor
- **ALPHEN** Alfa-compositor inschakelen
- **BBDRPOS** Positie van de onderrand
- **BITPBANK** Bepaalt welke 128KB-bank de bitplanes gebruiken
- **BOLDISALT** Indien gezet, selecteert de BOLD-attribuutbit het alternatieve palet, zonder dat REVERSE eveneens gezet hoeft te worden.
- **BORDERCOL** Weergaverandkleur (256 kleuren)
- **BP16ENS** Inschakelvlaggen voor de 16-kleurenbitplanes van de VIC-IV
- **BTPALSEL** Bitmap-/tekstpaletbank
- **CHARPTRBNK** Nauwkeurig basisadres van de tekenset (bits 23 - 16)
- **CHARPTRLSB** Nauwkeurig basisadres van de tekenset (bits 0 - 7)
- **CHARPTRMSB** Nauwkeurig basisadres van de tekenset (bits 15 - 8)
- **CHARY16** Alternatieve teken-ROM-bank op afwisselende rasterlijnen in V200
- **CHR16** 16-bits teekennummers inschakelen (twee schermbytes per teken)
- **CHRCOUNT** Aantal weer te geven tekens per rij (LSB)

- **CHRXSCL** Horizontale hardwarematige schaal van de tekstmodus (een honderdtwintigste pixel per pixel)
- **CHRYSCL** Verticale schaling van de tekstmodus (aantal fysieke rasterlijnen per tekstrij)
- **COLPTRLSB** Basisadres van het kleurengeheugen (bits 0 - 7)
- **COLPTRMSB** Basisadres van het kleurengeheugen (bits 15 - 8)
- **DBLRR** Wanneer deze is gezet, wordt de Raster Rewrite Buffer slechts om de tweede rasterlijn bijgewerkt, waardoor de resolutie wordt beperkt tot V200 maar er meer cycli voor Raster-Rewrite-acties overblijven.
- **DEBUGC** VIC-IV-debugpixelselectie: het rode (01), groene (10) of blauwe (11) kanaal is zichtbaar in $D07D
- **DISPROWS** Aantal tekstrijen dat moet worden weergegeven
- **EXGLYPH** Haalt full-colour-tekengegevens uit het uitbreidingsgeheugen
- **EXTIRQS** Schakelt extra IRQ-bronnen in, bijv. de raster-X-positie
- **FCLRHI** Schakelt de full-colour-modus in voor tekennummers >$FF
- **FCLRLO** Schakelt de full-colour-modus in voor tekennummers <=$FF
- **FCOLMCM** Schakelt 256 kleuren in de multikleur-tekstmodus in
- **FNRASTERLSB** Leest de fysieke rasterpositie
- **FNRASTERMSB** Leest de fysieke rasterpositie
- **FNRST** Leest de bron van de rastervergelijking (0=VIC-IV-fijne raster, 1=VIC-II-raster) en levert dezelfde waarde als in FNRSTCMP is ingesteld
- **FNRSTCMP** De rastervergelijking vindt plaats in fysieke rasterlijnen indien dit bit is gewist, of in VIC-II-lijnen indien het is gezet
- **HOTREG** Schakelt de VIC-II hot registers in. Wanneer deze zijn ingeschakeld, zorgt het aanraken van veel VIC-II-registers ervoor dat de VIC-IV de weergaveparameters herberekent, zoals randposities en -afmetingen. Het aanraken van registers terwijl deze functie is uitgeschakeld leidt tot een wijziging wanneer deze weer wordt ingeschakeld. Deze waarde op 0 instellen wist de herberekeningsvlag en annuleert de herberekening.
- **HSYNCP** hsync-polariteit
- **KEY** Schrijf $47 en dan $53 om de MEGA65/VIC-IV I/O-registers in te schakelen
- **LINESTEPLSB** Aantal bytes waarmee tussen elke tekstrij wordt opgeschoven (LSB)
- **LINESTEPMSB** Aantal bytes waarmee tussen elke tekstrij wordt opgeschoven (MSB)

- **MAPEDPAL** Ingemapte paletbank op $D100-$D3FF
- **MC1** Multikleur 1 (256 kleuren)
- **MC2** Multikleur 2 (256 kleuren)
- **MC3** Multikleur 3 (256 kleuren)
- **NOBUGCOMPAT** *VEROUDERD*, gebruik HWERRATA. Schakelt de VIC-III / C65-bugcompatibiliteitsmodus uit indien deze is gezet
- **NORRBWRAP** Schakelt de RRB-adreswrap-around uit wanneer deze is gezet
- **NORRDEL** Wanneer deze is gewist, wordt dubbelbuffering van de raster rewrite gebruikt
- **PALEMU** Schakelt PAL-CRT-achtige scanlijn-emulatie in (uitgeschakeld door V400 of H1280)
- **PALNTSC** NTSC-emulatiemodus (maximaal raster = 262)
- **RASCMP** Fysieke rastervergelijkingwaarde die moet worden gebruikt indien FNRSTCMP is gewist
- **RASCMPMSB** MSB van de rastervergelijkingwaarde
- **RASLINE0** Eerste VIC-II-rasterlijn
- **RASTERHEIGHT** Fysieke rasterlijnen per VIC-II-raster (1 tot 16)
- **RSTDELEN** Schakelt de rastervertraging in (vertraagt de rasterteller en interrupts met één regel om aan te sluiten bij de latentie van de uitvoerpipeline)
- **SCREENCOL** Schermkleur (256 kleuren)
- **SCRNPTRBNK** Nauwkeurig basisadres van het scherm-RAM (bits 23 - 16)
- **SCRNPTRLSB** Nauwkeurig basisadres van het scherm-RAM (bits 7 - 0)
- **SCRNPTRMB** Nauwkeurig basisadres van het scherm-RAM (bits 27 - 24)
- **SCRNPTRMSB** Nauwkeurig basisadres van het scherm-RAM (bits 15 - 8)
- **SDBDRWDLSB** Breedte van de zijrand (LSB)
- **SDBDRWDMSB** Breedte van de zijrand (MSB)
- **SHDEMU** Gesimuleerd schaduwmasker inschakelen (PALEMU moet ook zijn ingeschakeld)
- **SMTH** Horizontale vloeiendheid van de video-uitvoer inschakelen
- **SPR16EN** Inschakelbits voor de 16-kleurenmodus van sprites
- **SPRALPHAVAL** Alpha-vermengingswaarde van sprites
- **SPRBPMEN** Inschakelbits voor de sprite-bitplane-wijzigingsmodus
- **SPRENALPHA** Alpha-vermenging van sprites inschakelen

- **SPRENV400** Inschakelbits voor de V400-modus van sprites
- **SPRH640** H640-modus van sprites inschakelen
- **SPRHGHT** Grootte van de uitgebreide sprite-hoogte (in sprite-pixels)
- **SPRHGTEN** Uitgebreide sprite-hoogte inschakelen (één bit per sprite)
- **SPRMC0** Sprite-multikleur 0 (8-bits voor de keuze van elke paletkleur)
- **SPRMC1** Sprite-multikleur 1 (8-bits voor de keuze van elke paletkleur)
- **SPRPALSEL** Spritepaletbank
- **SPRPTR16** 16-bits sprite-pointermodus (ermee kunnen sprites op elke 64-bytegrens in chip-RAM worden geplaatst)
- **SPRPTRADRLSB** Sprite-pointeradres (bits 7 - 0)
- **SPRPTRADRMSB** Sprite-pointeradres (bits 15 - 8)
- **SPRPTRBNK** Sprite-pointeradres (bits 23 - 16)
- **SPRTILEN** Inschakelbits voor horizontale sprite-tegels
- **SPRX64EN** Inschakelbits voor de uitgebreide sprite-breedte (8 bytes per sprite-rij = 64 pixels breed voor normale sprites of 16 pixels breed voor de 16-kleuren-sprite-modus)
- **SPRXSMSBS** Super-MSB's van de X-positie van sprites in de H640-modus
- **SPRYADJ** Aanpassing van de Y-positie van sprites
- **SPRYMSBS** MSB's van de Y-positie van sprites in de V400-modus
- **SPRYSMSBS** Super-MSB's van de Y-positie van sprites in de V400-modus
- **SPTRCONT** Bewaakt voortdurend de sprite-pointer zodat de gegevensbron van de sprite kan worden gewijzigd terwijl een sprite wordt getekend
- **TBDRPOS** Positie van de bovenrand
- **TEXTXPOS** Horizontale positie van de tekengenerator
- **TEXTYPOS** Verticale positie van de tekengenerator
- **UPSCALE** Ingebouwde 720p-upscaler met lage latentie (130 usec) inschakelen
- **VFAST** MEGA65 FAST-modus (40.5MHz)
- **VGAHDTV** Selecteert indien gezet een meer VGA-compatibele modus in plaats van de cyclus-exacte frame-timing van de HDMI/HDTV-VIC-II. Kan helpen om een functioneel beeld op oudere VGA-monitoren te produceren.
- **VSYNCP** vsync-polariteit
- **XPOSLSB** Leest de LSB van de horizontale raster-scanpositie
- **XPOSMSB** Leest de MSB van de horizontale raster-scanpositie

HOOFDSTUK 3

Sound Interface Device (SID)

- SID-registers

SID-REGISTERS

De MEGA65 heeft vier ingebouwde SID's, bereikbaar via de registerbereiken die beginnen bij $D400, $D420, $D440 en $D460. De registers in elk van deze bereiken zijn exact hetzelfde, dus de volgende tabel vermeldt alleen de eerste SID. Tel er 32 bij op om bij de volgende SID te komen.

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D400	54272	VOICE1FRQLO							
D401	54273	VOICE1FRQHI							
D402	54274	VOICE1PWLO							
D403	54275	VOICE1UNSD				VOICE1PWHI			
D404	54276	VOICE1-CTRLRNW	VOICE1-CTRLPUL	VOICE1-CTRLSAW	VOICE1-CTRLTRI	VOICE1-CTRLTST	VOICE1-CTRLRMO	VOICE1-CTRLRMF	VOICE1-CTRLGATE
D405	54277	ENV1ATTDUR				ENV1DECDUR			
D406	54278	ENV1SUSLEV				ENV1RELDUR			
D407	54279	VOICE2FRQLO							
D408	54280	VOICE2FRQHI							
D409	54281	VOICE2PWLO							
D40A	54282	VOICE2UNSD				VOICE2PWHI			
D40B	54283	VOICE2-CTRLRNW	VOICE2-CTRLPUL	VOICE2-CTRLSAW	VOICE2-CTRLTRI	VOICE2-CTRLTST	VOICE2-CTRLRMO	VOICE2-CTRLRMF	VOICE2-CTRLGATE
D40C	54284	ENV2ATTDUR				ENV2DECDUR			
D40D	54285	ENV2LEVDUR				ENV2RELDUR			
D40E	54286	VOICE3FRQLO							
D40F	54287	VOICE3FRQHI							
D410	54288	VOICE3PWLO							
D411	54289	VOICE3UNSD				VOICE3PWHI			
D412	54290	VOICE3-CTRLRNW	VOICE3-CTRLPUL	VOICE3-CTRLSAW	VOICE3-CTRLTRI	VOICE3-CTRLTST	VOICE3-CTRLRMO	VOICE3-CTRLRMF	VOICE3-CTRLGATE
D413	54291	ENV3ATTDUR				ENV3DECDUR			
D414	54292	ENV3LEVDUR				ENV3RELDUR			
D415	54293	FLTRCUTFRQLO							
D416	54294	FLTRCUTFRQHI							
D417	54295	FLTRRESON				FLTR-EXTINP	FLTR-V3OUT	FLTR-V2OUT	FLTR-V1OUT
D418	54296	FLTR-CUTV3	FLTR-HIPASS	FLTR-BDPASS	FLTR-LOPASS	FLTRVOL			
D419	54297	PADDLE1							
D41A	54298	PADDLE2							
D41B	54299	OSC3RNG							
D41C	54300	ENV3OUT							
D41D	54301	SIDPANLEFT							
D41E	54302	SIDPANRIGHT							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D63C	54844	-			SIDEXENA	SIDMODE			

- **ENV3OUT** Uitgang van envelopegenerator 3
- **ENVXATTDUR** Duur van de aanzetcyclus van envelopegenerator X
- **ENVXDECDUR** Duur van de terugvalcyclus van envelopegenerator X
- **ENVXRELDUR** Duur van de wegstervenscyclus van envelopegenerator X
- **ENVXSUSLEV** Aanhoudniveau van envelopegenerator X
- **FLTRBDPASS** Filter banddoorlaatmodus
- **FLTRCUTFRQHI** Hoge afsnijfrequentie van het filter
- **FLTRCUTFRQLO** Lage afsnijfrequentie van het filter
- **FLTRCUTV3** Filter uitschakelen op de uitgang van stem 3 (1 = uit)
- **FLTREXTINP** Externe ingang van het filter
- **FLTRHIPASS** Filter hoogdoorlaatmodus
- **FLTRLOPASS** Filter laagdoorlaatmodus
- **FLTRRESON** Filterresonantie
- **FLTRVOL** Uitgangsvolume van het filter
- **FLTRVXOUT** Filteruitgang van stem X
- **OSC3RNG** Generator voor willekeurige getallen van oscillator 3
- **PADDLE1** Analoog/digitaalomzetter: paddle 1 (0-255)
- **PADDLE2** Analoog/digitaalomzetter: paddle 2 (0-255)
- **SIDEXENA** Extra SID-registers inschakelen
- **SIDMODE** Selecteert de SID-modus: 0=6581, 1=8580
- **SIDPANLEFT** Volume van het linkerkanaal van de SID (vereist dat SIDEXENA is ingesteld)
- **SIDPANRIGHT** Volume van het rechterkanaal van de SID (vereist dat SIDEXENA is ingesteld)
- **VOICE1CTRLRMF** Stem 1 synchroniseert oscillator 1 met de frequentie van oscillator 3
- **VOICE1CTRLRMO** Stem 1 ringmoduleert oscillator 1 met de uitgang van oscillator 3

- **VOICE2CTRLRMF** Stem 2 synchroniseert oscillator 2 met de frequentie van oscillator 1
- **VOICE2CTRLRMO** Stem 2 ringmoduleert oscillator 2 met de uitgang van oscillator 1
- **VOICE3CTRLRMF** Stem 3 synchroniseert oscillator 3 met de frequentie van oscillator 2
- **VOICE3CTRLRMO** Stem 3 ringmoduleert oscillator 3 met de uitgang van oscillator 2
- **VOICEXCTRLGATE** Stem X gate-bit (1 = start, 0 = wegsterven)
- **VOICEXCTRLPUL** Pulsgolfvorm van stem X
- **VOICEXCTRLRNW** Willekeurige ruisgolfvorm van stem X
- **VOICEXCTRLSAW** Zaagtandgolfvorm van stem X
- **VOICEXCTRLTRI** Driehoeksgolf van stem X
- **VOICEXCTRLTST** Testbit van stem X - de oscillator uitschakelen
- **VOICEXFRQHI** Frequentie van stem X (hoge byte)
- **VOICEXFRQLO** Frequentie van stem X (lage byte)
- **VOICEXPWHI** Pulsgolfbreedte van stem X (hoge byte)
- **VOICEXPWLO** Pulsgolfbreedte van stem X (lage byte)
- **VOICEXUNSD** Ongebruikt

HOOFDSTUK 4

F018-compatibele controller voor Direct Memory Access (DMA)

- F018A/B-DMA-taken
- MEGA65 Enhanced DMA Jobs
- Textuurschaling en lijntekening
- Inline-DMA-lijsten
- Audio-DMA
- F018 “DMAgic” DMA-controller
- Uitbreidingen van de MEGA65-DMA-controller
- Niet-geïmplementeerde functionaliteit

De MEGA65 bevat een F018/F018A-compatibele DMA-controller. Anders dan in de C65, waar de DMA-controller als een afzonderlijke chip bestaat, maakt deze in de MEGA65 deel uit van de 45GS02-processor. Omdat het gebruik van de DMA-controller echter een logisch afzonderlijk onderwerp vormt, wordt het apart in deze bijlage besproken.

De DMA-controller van de MEGA65 biedt diverse belangrijke verbeteringen ten opzichte van de F018/F018A-DMAgic-chips van de C65:

- **Snelheid** De MEGA65 voert DMA-bewerkingen uit met 40 MHz, waardoor per seconde 40 MB kan worden gevuld of 20 MB kan worden gekopieerd. Zo is het bijvoorbeeld mogelijk om een volledige bitmapafbeelding van 8 KB in C64-stijl in circa 200 microseconden te kopiëren, wat overeenkomt met minder dan vier rasterlijnen!
- **Toegang tot het volledige geheugen** De DMA-controller van de MEGA65 biedt toegang tot alle 256 MB van de adresruimte.
- **Ondersteuning voor het kopiëren van texturen** De DMA-controller van de MEGA65 kan fractionele adresberekeningen uitvoeren ter ondersteuning van hardwarematige textuurschaling, evenals adresstriding, waardoor het in principe mogelijk is om een textuur gelijktijdig te schalen en vanuit het geheugen naar het scherm te tekenen. Dat zou nuttig zijn als iemand gek genoeg zou zijn om een spel in de stijl van Wolfenstein of Doom op de MEGA65 te implementeren.
- **Ondersteuning voor transparantie- en maskerwaarden** De DMA-controller van de MEGA65 kan de opdracht krijgen om bij het kopiëren van geheugen een speciale waarde te negeren, waardoor de inhoud van het bestemmingsgeheugen ongewijzigd blijft. Dit maakt het maskeren van transparante gebieden bij een DMA-kopie mogelijk, wat het blitten van grafische vormen aanzienlijk vereenvoudigt.
- **Optielijst per taak** Voor elke taak in een geketende lijst van DMA-taken kunnen diverse opties worden geconfigureerd, zoals het selecteren van de F018- of F018B-modus, het wijzigen van de transparantiewaarde, fractionele adresstappen of het bron- of bestemmingsgeheugengebied.
- **Achtergrond-audio-DMA** De MEGA65 beschikt over mogelijkheden voor achtergrond-audio-DMA, vergelijkbaar met de Amiga™-serie computers. Belangrijke verschillen zijn dat de MEGA65 samples van 8 of 16 bits kan gebruiken, zeer hoge samplerates tot ongeveer 1 MHz ondersteunt, 256 volume-instellingen per kanaal heeft en geen modulatie tussen de kanalen kent.

F018A/B-DMA-TAKEN

Om een DMA-taak met de F018-serie DMA-controllers uit te voeren, moet u de lijst met DMA-taken in het geheugen opbouwen en vervolgens het adres van deze lijst

in de DMA-adresregisters schrijven. De DMA-taak wordt uitgevoerd wanneer u naar het register ADDRLSBTRIG ($D700) schrijft. Om die reden moet u eerst het MSB en het banknummer van de DMA-lijst in $D701 en $D702 schrijven en pas daarna het LSB, nadat u deze beide registers hebt ingesteld. Indien u met dezelfde lijststructuur in het geheugen meerdere DMA-taken wilt uitvoeren, kunt u gewoon opnieuw naar ADDRLSBTRIG schrijven nadat u de inhoud van de lijst hebt bijgewerkt, mits geen ander programma de inhoud van $D701 of $D702 heeft gewijzigd. Houd er rekening mee dat BASIC 65 de DMA-controller gebruikt om het scherm te laten scrollen. Het is dan ook doorgaans het veiligst om altijd naar alle drie de registers te schrijven.

Nadat er naar ADDRLSBTRIG is geschreven, wordt de DMA-taak onmiddellijk afgerond. Anders dan in de C65 maakt de DMA-controller in de MEGA65 deel uit van de processor. Dit betekent dat de processor stopt met het uitvoeren van instructies totdat de DMA-taak is voltooid. De enige uitzondering hierop is dat audio-DMA doorgaat en cycli onttrekt aan elke andere DMA-activiteit, zodat de geluidsweergave niet wordt beïnvloed.

Dit gedrag betekent dat DMA-taken, anders dan op de C65, niet kunnen worden onderbroken. Heeft uw programma gevoelige timingvereisten, dan moet u grotere DMA-taken wellicht opsplitsen in verscheidene kleinere taken. Deze zorg wordt enigszins gecompenseerd door de hoge snelheid van de DMA in de MEGA65, die per seconde 40.5 MB geheugen kan vullen en 20.25 MB kan kopiëren, vergeleken met circa 3.5 MB en 1.7 MB per seconde op een C65. Daardoor kunnen grotere DMA-taken worden uitgevoerd zonder dat u zich zorgen hoeft te maken over de gevolgen voor realtime-elementen van een programma. Zo is het bijvoorbeeld mogelijk om met de DMA-controller van de MEGA65 een tekstscherm van 80 kolommen bij 50 rijen te vullen in slechts 200 microseconden.

Indeling van de F018-DMA-taaklijst

De DMA-controller van de MEGA65 ondersteunt de twee formaten voor DMA-taaklijsten die de oorspronkelijke F018 gebruikte, welke in de vroege C65-prototypen werd toegepast (tot revisie 2B), en die de F018B en latere revisies gebruikten, welke in de C65-prototypen van revisie 3 - 5 werden toegepast. Het belangrijkste verschil is de toevoeging van een tweede commandobyte, zoals de volgende tabellen tonen:

Het is belangrijk om te weten welke stijl de DMA-controller verwacht. De Hypervisor van de MEGA65 stelt de modus in op basis van de gedetecteerde versie van de C65-ROM, indien er één wordt uitgevoerd. Indien het een oudere versie betreft, wordt de F018-stijl verwacht; anders wordt de nieuwere F018B-stijl verwacht. U kunt nagaan welke stijl is geselecteerd door bit 0 van $D703 op te vragen: staat deze op 1, dan wordt het nieuwere F018B-lijstformaat van 12 bytes verwacht; staat deze op 0, dan wordt het oudere F018-lijstformaat van 11 bytes verwacht. De verwachte stijl kan worden ingesteld door naar dit register te schrijven.

Tenzij u software schrijft die ook op een C65-prototype moet draaien, kunt u waarschijnlijk het beste gebruikmaken van de Enhanced DMA-taken van de MEGA65, waarvan het lijstformaat expliciet in de lijst zelf wordt opgegeven. Omdat de Enhanced

DMA-taken een uitbreiding zijn van de F018/F018B-DMA-taken, moet u het volgende toch lezen, tenzij u het gedrag van de F018-DMA-controller al kent.

Structuur van de 11-byte F018-DMA-lijst

Offset	Inhoud
$00	Commando LSB
$01	Aantal LSB
$02	Aantal MSB
$03	Bronadres LSB
$04	Bronadres MSB
$05	Bronadres BANK en FLAGS
$06	Bestemmingsadres LSB
$07	Bestemmingsadres MSB
$08	Bestemmingsadres BANK en FLAGS
$09	Modulo LSB
$0a	Modulo MSB

* Het commando-MSB is $00 bij gebruik van dit lijstformaat.

Structuur van de 12-byte F018B-DMA-lijst

Offset	Inhoud
$00	Commando LSB
$01	Aantal LSB
$02	Aantal MSB
$03	Bronadres LSB
$04	Bronadres MSB
$05	Bronadres BANK en FLAGS
$06	Bestemmingsadres LSB
$07	Bestemmingsadres MSB
$08	Bestemmingsadres BANK en FLAGS
$09	Commando MSB
$0a	Modulo LSB / Modus
$0b	Modulo MSB / Modus

De structuur van het commandowoord is als volgt:

Bit(s)	Inhoud
0 - 1	DMA-bewerkingstype
2	Ketting (d.w.z. er volgt nog een DMA-lijst)
3	Toegeven aan interrupts
4	MINTERM -SA,-DA bit
5	MINTERM -SA,DA bit
6	MINTERM SA,-DA bit
7	MINTERM SA,DA bit
8 - 9	Adresseermethode van de bron
10 - 11	Adresseermethode van de bestemming
12 - 15	GERESERVEERD. Altijd op 0 instellen

Het commando-veld kent de volgende vier waarden:

Waarde	Inhoud
%00 (0)	Kopiëren
%01 (1)	Mengen (via MINTERMs)
%10 (2)	Wisselen
%11 (3)	Vullen

* Alleen Kopiëren en Vullen zijn op het moment van schrijven geïmplementeerd.

De velden voor de adresseermethode kennen de volgende vier waarden:

Waarde	Inhoud
%00 (0)	Lineair (normaal) adresseren
%01 (1)	Modulo (rechthoekig) adresseren
%10 (2)	Vasthouden (constant adres)
%11 (3)	XY MOD (rechthoekige bitmap) adresseren

* Alleen Lineair, Modulo en Vasthouden zijn op het moment van schrijven geïmplementeerd.

Het BANK- en FLAGS-veld voor de bron- en bestemmingsadressen maakt het mogelijk adressen binnen een adresruimte van 1 MB te selecteren. Om geheugen buiten de eerste 1 MB te benaderen, is het noodzakelijk een Enhanced DMA-taak te gebruiken met de juiste optiebytes om de MB van het bron- en/of bestemmingsgeheugen te selecteren. Het BANK- en FLAGS-veld heeft de volgende structuur:

Bit(s)	Inhoud
0 - 3	Geheugenbank binnen de geselecteerde MB
4	HOLD, d.w.z. wijzig het adres niet
5	MODULO, d.w.z. pas het MODULO-veld toe voor de wrap-around binnen een beperkte geheugenruimte
6	DIRECTION. Indien gezet, wordt het adres verlaagd in plaats van verhoogd.
7	I/O. Indien gezet, zijn de I/O-registers tijdens de DMA zichtbaar op $D000 - $DFFF.

Eenvoudige DMA-bewerkingen uitvoeren

Voor informatie over het gebruik van de DMA-controller vanuit BASIC 65, zie de BASIC-opdracht **DMA** in *het grote Mega65-boek*, DMA (deelparagraaf B).

Om de DMA-controller vanuit assembleertaal te gebruiken, bouwt u een gegevens-structuur met de DMA-lijst en stelt u daarna $D702 - $D700 in op het adres van de lijst. Om bijvoorbeeld het scherm in de C65-modus te wissen door het met spaties te vullen, zou de volgende routine kunnen worden gebruikt:

```
; De DMA-lijst bevindt zich in BANK 0
LDA #$00
STA $D702
; Stel het MSB van het DMA-lijstadres in
```

```
    LDA #>dmalist
    STA $D701
    ; Stel het LSB van het DMA-lijstadres in en voer de DMA uit
    LDA #<dmalist
    STA $D700
    RTS

  dmalist:
    .byte $03   ; Commando-laagbyte: FILL
    .word 2000  ; Aantal: 80x25 = 2000 bytes
    .word $0020 ; Vul met waarde $20
    .byte $00   ; Bronbank (genegeerd bij een FILL-bewerking)
    .word $0800 ; Bestemmingsadres van de schermgegevens
    .byte $00   ; Het scherm bevindt zich in bank 0
    .byte $00   ; Commando-hoogbyte
    .word $0000 ; Modulo (genegeerd vanwege het gekozen commando)
```

Het is ook mogelijk meer dan één DMA-taak tegelijk uit te voeren door de CHAIN-bit in het lage byte van het commando te zetten. Om bijvoorbeeld het scherm te wissen zoals hiervoor en daarnaast ook het kleurengeheugen (colour-RAM) van het scherm te wissen, zou u iets kunnen gebruiken als:

```
    ; De DMA-lijst bevindt zich in BANK 0
    LDA #$00
    STA $D702
    ; Stel het MSB van het DMA-lijstadres in
    LDA #>dmalist
    STA $D701
    ; Stel het LSB van het DMA-lijstadres in en voer de DMA uit
    LDA #<dmalist
    STA $D700
    RTS

  dmalist:
    .byte $07   ; Commando-laagbyte: FILL + CHAIN
    .word 2000  ; Aantal: 80x25 = 2000 bytes
    .word $0020 ; Vul met waarde $20
    .byte $00   ; Bronbank (genegeerd bij een FILL-bewerking)
    .word $0800 ; Bestemmingsadres van de schermgegevens
    .byte $00   ; Het scherm bevindt zich in bank 0
    .byte $00   ; Commando-hoogbyte
    .word $0000 ; Modulo (genegeerd vanwege het gekozen commando)

    ; De tweede DMA-taak volgt direct op de eerste
    .byte $03   ; Commando-laagbyte: FILL
    .word 2000  ; Aantal: 80x25 = 2000 bytes
    .word $0001 ; Vul met waarde $01 = wit
    .byte $00   ; Bronbank (genegeerd bij een FILL-bewerking)
    .word $F800 ; Bestemmingsadres van het kleurengeheugen
    .byte $01   ; Het kleurengeheugen bevindt zich in
                ; bank 1 ($1F800-$1FFFF)
```

```
    .byte $00   ; Commando-hoogbyte
    .word $0000 ; Modulo (genegeerd vanwege het gekozen commando)
```

Geheugen kopiëren lijkt sterk op het vullen van geheugen, met dit verschil dat het lage byte van het commando moet worden aangepast en dat het bronadresveld correct moet worden geïnitialiseerd. Om bijvoorbeeld de tekenset uit de ROM op $2D000 - $2DFFF te kopiëren naar $5000, zou u iets kunnen gebruiken als:

```
    ; De DMA-lijst bevindt zich in BANK 0
    LDA #$00
    STA $D702
    ; Stel het MSB van het DMA-lijstadres in
    LDA #>dmalist
    STA $D701
    ; Stel het LSB van het DMA-lijstadres in en voer de DMA uit
    LDA #<dmalist
    STA $D700
    RTS

  dmalist:
    .byte $00   ; Commando-laagbyte: COPY
    .word $1000 ; Aantal: 4KB = 4096
    .word $D000 ; Kopieer vanaf $xD000
    .byte $02   ; Bronbank = $02 voor $2xxxx
    .word $5000 ; Bestemmingsadres van de schermgegevens
    .byte $00   ; Het scherm bevindt zich in bank 0
    .byte $00   ; Commando-hoogbyte
    .word $0000 ; Modulo (genegeerd vanwege het gekozen commando)
```

Het is ook mogelijk vanuit BASIC 2 in de C64-modus een DMA-bewerking uit te voeren door de benodigde waarden met POKE te schrijven, nadat u er eerst voor hebt gezorgd dat de MEGA65- of C65-I/O-modus is geselecteerd door de juiste waarden naar $D02F (53295) te schrijven. Om bijvoorbeeld het scherm in C64 BASIC 2 te wissen met behulp van de DMA-controller, zou u iets kunnen gebruiken als:

```
10 rem schakel de MEGA65-I/O in
20 poke53295,asc("g"):poke53295,asc("s")
30 rem DMA-lijst in data-regels
40 data 3: rem commando-lsb = vullen
50 data 232,3 : rem scherm is 1000 bytes = 3*256+232
60 data 32,0: rem vullen met spatie = 32
70 data 0: rem bronbank (niet gebruikt bij vullen)
80 data 0,4: rem schermadres = 1024 = 4*256
90 data 0: rem scherm bevindt zich in bank 0
100 data 0: rem hoge byte van commando
110 data 0,0: rem modulo (niet gebruikt in deze opgave)
120 rem zet de DMA-lijst op $c000 = 49152
130 for i=0to11:reada:poke49152+i,a:next
140 rem voer de taak uit
150 poke55042,0: rem DMA-lijst staat in bank 0
160 poke55041,192: rem DMA-lijst staat in $c0xx
170 poke55040,0: rem DMA-lijst staat in $xx00, en voer uit
```

Hoewel dit elke keer vrij omslachtig is, hoeft u, indien u het scherm opnieuw wilt wissen, niets anders te doen dan opnieuw **POKE 55040,0** uit te voeren, ervan uitgaande dat de DMA-lijst en de registers van de DMA-controller niet zijn gewijzigd sinds de vorige keer dat de DMA-taak is uitgevoerd.

De HOLD-, I/O- en andere opties kunnen ook worden gebruikt om interessante effecten te bereiken. Om bijvoorbeeld zeer snel een nieuwe waarde naar de achtergrondkleur van het scherm te schrijven, kunt u een geheugengebied kopiëren naar $D021, met de I/O-vlag gezet zodat het I/O-register in de DMA-taak zichtbaar wordt voor schrijven, en de HOLD-vlag gezet zodat herhaaldelijk naar hetzelfde adres wordt geschreven. Hiermee wordt met een snelheid van 20.5 MHz naar de achtergrondkleur geschreven, bijna zo snel als de videopixelklok (27 MHz). Zo kunnen wij de kleur vrijwel per pixel wijzigen.

Met een beetje zorg kunnen wij deze routine zo maken dat de uitvoering ervan precies één rasterlijn duurt en zo verticale rasterbalken trekken, of een soort frankenstein-videomodus creëren die een lineaire geheugenindeling gebruikt – ten koste van alle verwerkingstijd gedurende het actieve deel van het beeld.

Het volgende voorbeeld doet dit om verticale rasterbalken op het scherm te trekken. Het programma gaat ervan uit dat de MEGA65 op PAL is ingesteld. Voor NTSC moet de omvang van de DMA-overdracht enigszins worden verkleind. Verder valt bij dit programma op dat met MEGA65 Enhanced-DMA-optie $81 de bestemmingsmegabyte van het geheugen op $FFxxxxx wordt ingesteld, de bank op $D en het bestemmingsadres op $0021, om zo het volledige adres $FFD0021 te vormen. Dit is de werkelijke locatie van het randkleurregister van de VIC-IV. Het programma is geschreven met ACME-compatibele syntaxis.

```
basicheader:
  ;; 2020 SYS 2061
```

```
  !word $80A,2020
  !byte $9E,$32,$30,$36,$31,0,0,0

  ;; Werkelijke code begint op $080d = 2061
main:
  SEI
  LDA #$47      ; Schakel de MEGA65-I/O in
  STA $D02F
  LDA #$53
  STA $D02F
  LDA #65       ; Stel de CPU-snelheid in op snel
  STA 0
  LDA #0        ; Schakel het scherm uit, alles is schermrandkleur
  STA $D011

  LDA $D012     ; Wacht tot het begin van de volgende rasterlijn
raster_sync:    ; vóór het begin van de lus voor horizontale
  CMP $D012     ; uitlijning
  BEQ raster_sync

  ;; De volgende lus duurt precies één rasterlijn
  ;; bij 40,5 MHz in PAL
loop:
  JSR triggerdma
  JMP loop

triggerdma:
  LDA #0      ; Zorg dat het F018-lijstformaat actief is
  STA $D703

  LDA #0      ; DMA-lijstbank
  STA $D702
  LDA #>rasterdmalist
  STA $D701
  LDA #<rasterdmalist
  STA $D705
  RTS

rasterdmalist:
  !byte $81,$FF,$00
  !byte $00     ; COPY
  !word 619     ; De DMA-overdracht is 619 bytes lang
  !word rastercolours  ; bronadres
  !byte $00     ; bronbank
  !word $0020   ; bestemmingsadres
  !byte $1D     ; bestemmingsbank + HOLD
  ;; ongebruikt moduloveld
  !word $0000

rastercolours:
  !byte 0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

```
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
!byte 0,0,0,11,11,11,12,12,12,15,15,15
!byte 1,1,1,15,15,15,12,12,12,11,11,11,0,0,0
!byte 0,0,0,6,6,6,4,4,4,14,14,14,3,3,3
!byte 1,1,1,3,3,3,14,14,14,4,4,4,6,6,6,0,0,0
```

MEGA65 ENHANCED DMA JOBS

De implementatie van de DMAgic in de MEGA65 ondersteunt aanzienlijk uitgebreidere DMA-taken. Een uitgebreide DMA-taak wordt aangegeven door het lage byte van het adres van de DMA-lijst naar $D705 te schrijven in plaats van naar $D700. De MEGA65 zoekt vervolgens aan het begin van de DMA-lijst naar één of meer *taakoptietokens*. Die tokens worden geïnterpreteerd voordat de DMA-taak wordt uitgevoerd die onmiddellijk volgt op het *einde van de taakopties*-token ($00).

Taakoptietokens die een argument nodig hebben, hebben de meest significante bit gezet en worden altijd gevolgd door een optie van 1 byte. Taakoptietokens die

geen argument nodig hebben, hebben de meest significante bit niet gezet. Niet-ondersteunde taakoptietokens worden eenvoudigweg genegeerd. Zo kunnen toekomstige revisies van de DMAgic ondersteuning voor aanvullende opties toevoegen, zonder dat de achterwaartse compatibiliteit wordt verbroken.

Opties die door een taak zijn ingesteld blijven gelden voor de volgende taken in een geketende lijst. Ze worden alleen teruggezet naar hun standaardwaarden wanneer de laatste taak in de keten klaar is. Een taak in een keten die een optie weer op zijn standaardwaarde wil hebben, moet die optie dus opnieuw in zijn eigen optielijst instellen; het weglaten van de optie zet hem niet terug.

Deze opties worden ook gebruikt om geavanceerde functies te bereiken, zoals hardwarematige textuurschaling met snelheden tot 20 megapixels per seconde en hardwarematig lijntekenen met snelheden tot 40 megapixels per seconde. Deze geavanceerde functies zijn geïmplementeerd doordat tijdens de uitvoering complexe berekeningen kunnen worden toegepast op het bron- en/of bestemmingsadres van DMA-jobs.

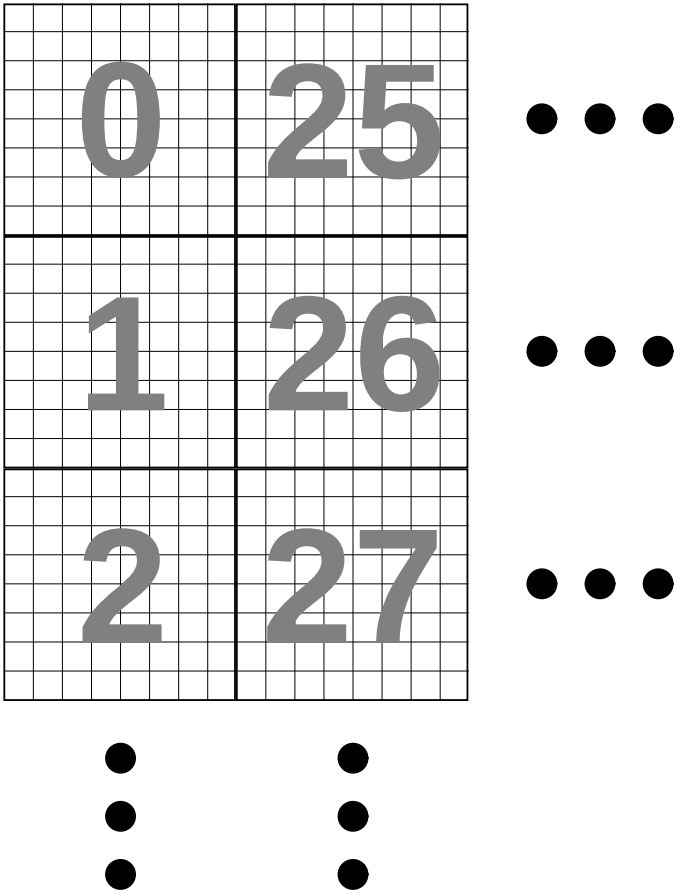
De lijst met geldige taakoptietokens is:

$00	Einde van de taakoptielijst
$06	Gebruik van de transparante waarde uitschakelen
$07	Gebruik van de transparante waarde inschakelen
$0A	Gebruik het 11-byte-F011A-DMA-lijstformaat
$0B	Gebruik het 12-byte-F011B-DMA-lijstformaat
$0D	Ruwe flux naar het diskettestation schrijven
$0E	Ruwe flux van het diskettestation lezen
$0F	Ruwe flux van het diskettestation lezen
$53	De modus 'Shallan Spiral' inschakelen
$80	Bronadres bits 20 - 27
$81	Bestemmingsadres bits 20 - 27
$82	Bron-skipwaarde (256ste van een byte)
$83	Bron-skipwaarde (gehele bytes)
$84	Bestemmings-skipwaarde (256ste van een byte)
$85	Bestemmings-skipwaarde (gehele bytes)
$86	Transparante waarde (bytes met een overeenkomende waarde worden niet geschreven)
$87	X-kolombytes instellen (LSB) voor het bestemmingsadres bij lijntekening
$88	X-kolombytes instellen (MSB) voor het bestemmingsadres bij lijntekening
$89	Y-rijbytes instellen (LSB) voor het bestemmingsadres bij lijntekening
$8A	Y-rijbytes instellen (MSB) voor het bestemmingsadres bij lijntekening
$8B	Helling (LSB) voor het bestemmingsadres bij lijntekening
$8C	Helling (MSB) voor het bestemmingsadres bij lijntekening
$8D	Beginfractie van de hellingaccumulator (LSB) voor het bestemmingsadres bij lijntekening
$8E	Beginfractie van de hellingaccumulator (MSB) voor het bestemmingsadres bij lijntekening
$8F	Lijntekenmodus inschakelen en opties voor het bestemmingsadres (in te stellen in het argumentbyte): bit 7 = lijnmodus inschakelen, bit 6 = richting X of Y kiezen, bit 5 = de helling is negatief.
$97	X-kolombytes instellen (LSB) voor het bronadres bij lijntekening
$98	X-kolombytes instellen (MSB) voor het bronadres bij lijntekening
$99	Y-rijbytes instellen (LSB) voor het bronadres bij lijntekening
$9A	Y-rijbytes instellen (MSB) voor het bronadres bij lijntekening
$9B	Helling (LSB) voor het bronadres bij lijntekening
$9C	Helling (MSB) voor het bronadres bij lijntekening
$9D	Beginfractie van de hellingaccumulator (LSB) voor het bronadres bij lijntekening
$9E	Beginfractie van de hellingaccumulator (MSB) voor het bronadres bij lijntekening
$9F	Lijntekenmodus inschakelen en opties voor het bronadres (in te stellen in het argumentbyte): bit 7 = lijnmodus inschakelen, bit 6 = richting X of Y kiezen, bit 5 = de helling is negatief.

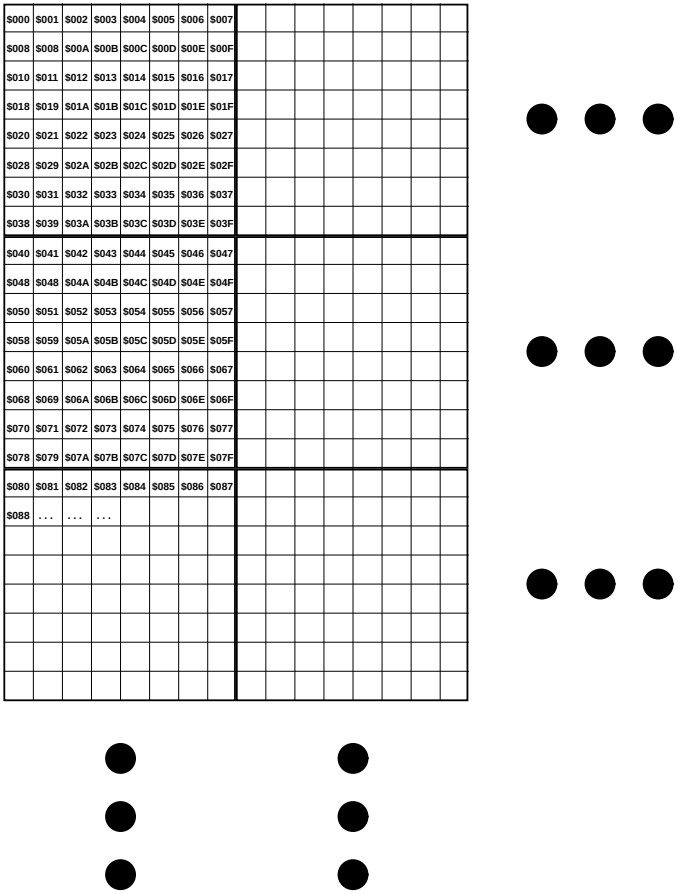
TEXTUURSCHALING EN LIJNTEKENING

De DMAgic beschikt over een geavanceerde interne adresrekenmachine waarmee hij geschaalde texturen en lijnen met willekeurige hellingen kan tekenen op VIC-IV-FCM-beeldschermen.

Voor het schalen van texturen moet het FCM-scherm verticaal worden gerangschikt, zoals hieronder is weergegeven:



Door de tekens zo in verticale kolommen te rangschikken, levert een verticale verplaatsing van één pixel telkens een constante toename van 8 bytes op, zoals hieronder wordt weergegeven:



De skipwaarden voor bron en bestemming maken het ook mogelijk de schaalfactoren in te stellen. Een skipwaarde van $0100 komt overeen met een verplaatsing van $01.00 pixel. Wilt u de verticaal gestapelde FCM-indeling gebruiken als doel voor het kopiëren van verticale textuurlijnen, dan moet de bestemmingsskipwaarde $0800 zijn, d.w.z. 8.0 bytes per pixel. Daarmee wordt een verticale lijn textuurgegevens zonder schaling gekopieerd. Door de bronverplaatsing lager dan $0100 in te stellen, worden sommige pixels herhaald, waardoor de textuur effectief wordt ingezoomd; stelt u de bronverplaatsing boven $0100 in, dan worden sommige pixels overgeslagen, waardoor de textuur effectief wordt uitgezoomd. De bestemmingsverplaatsing hoeft gewoonlijk niet te worden aangepast. Let op: de textuurgegevens moeten zo worden opgeslagen dat elke verticale strook aaneengesloten staat, opdat deze modus kan worden gebruikt.

Voor het tekenen van lijnen moet de DMA-controller de schermindeling kennen. Hij moet in het bijzonder weten welk getal moet worden opgeteld bij het adres van de meest rechtse pixel in een kolom FCM-tekens om het adres te berekenen van de pixel

die onmiddellijk rechts daarvan verschijnt. Evenzo moet hij weten hoeveel er moet worden opgeteld bij het adres van de onderste pixel in een rij FCM-tekens om het adres te berekenen van de pixel die onmiddellijk onder die pixel verschijnt. Dit biedt flexibele mogelijkheden voor de schermindeling en willekeurige schermafmetingen. U moet daarna ook de helling van de lijn opgeven, of de lijn de X-as of de Y-as als hoofdas heeft en of de helling positief of negatief is.

Het bestand `test_290.c` in de repository https://github.com/mega65/mega65-tools bevat een voorbeeld van het gebruik van deze voorzieningen om hardware-versnelde lijntekening te implementeren. Deze aanpak is *zeer* snel, want de lijnen worden getekend met de volledige DMA-vulsnelheid, d.w.z. ongeveer 40.5 miljoen pixels per seconde.

INLINE-DMA-LIJSTEN

Gewoonlijk moet u een apart geheugengebied inrichten dat de DMA-lijst bevat en vervolgens het adres van dat gebied in de DMA-adresregisters op $D70x laden. Omdat de DMA-controller van de MEGA65 deel uitmaakt van de CPU, ondersteunt deze bovendien een uiterst handige extra modus: de inline-DMA-lijstmodus.

Deze modus werkt zoals de Enhanced-DMA-modus, met dit verschil dat de DMA-lijst wordt gelezen vanaf de huidige waarde van het Program Counter (PC)-register. Om deze modus te gebruiken, schrijft u een willekeurige waarde naar $D707. Hiermee start u onmiddellijk een DMA-taak, waarvan de lijst staat in de bytes die direct volgen op de instructie die naar $D707 schrijft.

De DMA-lijst kan uit één enkele taak bestaan of, zoals bij elke andere DMA-taak, uit meerdere gekoppelde taken. Het echte kunststuk is dat de programmateller wordt gezet op het eerstvolgende adres na het einde van de DMA-lijst en dat de DMA-lijst wordt gelezen uit de huidige geheugenindeling van de CPU. Dit betekent dat u code met DMA-lijsten vanuit elke geheugenbank kunt uitvoeren, zonder dat u hoeft na te gaan in welke bank die code staat.

Het volgende codevoorbeeld wist bijvoorbeeld het scherm in de C65-modus en laat de rand daarna eindeloos flikkeren:

```
    STA $D707
    .byte $00   ; end of job options
    .byte $03   ; fill
    .word 2000  ; count
    .word $0020 ; value
    .byte $00   ; src bank
    .word $0800 ; dst
    .byte $00   ; dst bank
    .byte $00   ; cmd hi
    .word $0000 ; modulo / ignored
foo:
    INC $D020
    JMP foo
```

AUDIO-DMA

De MEGA65 beschikt over vier kanalen voor DMA-aangestuurde geluidsweergave, die in plaats van de directe digitale audioregisters op $D6F8-$D6FB gebruikt kunnen worden. U moet dus kiezen welke van deze twee bronnen u aan de audio-crossbar-mixer wilt toevoeren. Die keuze wordt gemaakt via het AUDEN-signaal (bit 7 van $D711), dat tegelijkertijd de audio-DMA-functie in de processor inschakelt en de audio-crossbar-mixer opdraagt de audio van deze bron te gebruiken in plaats van de digitale audioregisters op $D6F8-$D6FB. Indien u naast de audio-DMA-kanalen geen ander geluid wenst, kan de audio-crossbar-mixer worden omzeild en kan de DMA-audio op vol volume worden afgespeeld door het NOMIX-signaal te zetten (bit 4 van $D711). De omzeiling geldt alleen voor de PWM-audio-uitgangen (de 3,5mm-audioaansluiting op R3 en oudere borden): op die uitgangen is er geen geluid van de SID's, FM, microfoons of andere bronnen beschikbaar. HDMI-audio, en de audio-DAC op R4 en nieuwere borden, dragen altijd de uitvoer van de audio-crossbar-mixer. Alle overige bits van $D711 moeten gewoonlijk nul blijven; schrijf dus $80 naar $D711 om audio-DMA in te schakelen.

Twee kanalen vormen het linker digitale audiokanaal en de twee andere het rechter digitale audiokanaal. Deze linker- en rechterkanalen worden vervolgens aan de audio-crossbar-mixer van de MEGA65 toegevoerd.

Omdat de DMA-controller deel uitmaakt van de processor van de MEGA65 en de MEGA65 geen gereserveerde busesleuven heeft voor multimedia-bewerkingen, gebruikt de MEGA65 ongebruikte CPU-cycli om DMA op de achtergrond uit te voeren. Hiervoor moet de CPU van de MEGA65 in de modus "full speed" staan, d.w.z. op ongeveer 40 MHz. In deze modus treedt bij het lezen van een operand uit het geheugen telkens een wachtcyclus op. Elke instructie die een byte uit het geheugen laadt, creëert daardoor één impliciete audio-DMA-sleuf. In de praktijk is dit zelden een probleem, behalve wanneer de processor in een zeer strakke lus wacht. Om ervoor te zorgen dat de audio op de achtergrond blijft spelen, moeten dergelijke lussen een leesinstructie bevatten, zoals:

```
    ; Ensure loop has at least one idle cycle for
    ; audio DMA
loop:
    LDA $1234
    JMP loop
```

Elk van de vier DMA-kanalen wordt geconfigureerd met een blok van 16 registers op respectievelijk $D720, $D730, $D740 en $D750. Wij lichten de registers van het eerste kanaal, kanaal 0, toe op $D720 - $D72F.

Adresbeheer voor samples

Om een audiosample af te spelen, moet u eerst het startadres van het sample opgeven. Dit is een 24-bits adres dat in het hoofd-chipgeheugen van de MEGA65 moet liggen. U schrijft het adres daartoe in $D72A - $D72C. Dat is het adres van de eerste

samplewaarde die wordt afgespeeld. Vervolgens moet u het eindadres van het sample opgeven in $D727 - $D728. Let echter op: dit is slechts 16 bits. Dat komt doordat de MEGA65 bij de controle of het einde van een sample is bereikt, alleen de onderste 16 bits van het adres vergelijkt. In de praktijk betekent dit dat samples niet groter dan 64 KB kunnen zijn. Bevat het sample een gedeelte dat herhaald moet worden, dan moet het startadres van dat herhalende gedeelte in $D721 - $D723 worden geladen en moet de CH0LOOP-bit worden gezet (bit 6 van $D720).

U kunt op elk moment het huidige sampleadres bepalen door de registers op $D72A - $D72C te lezen. Let echter op: deze registers worden niet vergrendeld, zodat de waarden tijdens het lezen kunnen worden bijgewerkt, tenzij u het kanaal eerst stopt door het CH0EN-signaal te wissen.

Afspeelfrequentie en volume van samples

De MEGA65 regelt de afspeelsnelheid van audio-DMA-samples met een 24-bits teller. Telkens wanneer de 24-bits teller overloopt, wordt de volgende samplewaarde opgevraagd. De samplesnelheid stelt u in door de waarde te kiezen die elke CPU-cyclus aan deze teller wordt toegevoegd. Een waarde van $FFFFFF levert derhalve een samplerate van bijna 40.5 MHz op. In de praktijk zijn samplerates boven enkele megahertz echter niet mogelijk, omdat daarvoor onvoldoende ongebruikte CPU-cycli beschikbaar zijn en er anders vervormd geluid ontstaat. Ook onder die grens moet u ervoor zorgen dat de ongebruikte cycli voldoende vaak en voldoende verspreid door de instructiestroom van de processor voorkomen, om vervorming te vermijden. Bij typische samplerates onder 16 kHz en het gebruik van 8-bits samples zijn deze effecten voor normale instructiestromen doorgaans verwaarloosbaar, zodat voor normaal audio-afspelen gewoonlijk geen speciale maatregelen nodig zijn.

Aan de andere kant van de schaal zijn samplefrequenties van slechts 40.5 MHz/2^{24}= 2.4 samples per seconde mogelijk. Dat is laag genoeg voor zelfs de meest veeleisende infrageluidstoepassingen.

Het volume wordt geregeld door $D729 in te stellen. Het maximale volume wordt bereikt met de waarde $FF, terwijl een waarde van $00 het kanaal volledig dempt. De eerste twee audiokanalen worden normaal gesproken aan de linkerzijde toegewezen en de tweede twee aan de rechterzijde. De MEGA65 beschikt echter over afzonderlijke volumeregelaars voor de tegenoverliggende kanalen. Om audio-DMA-kanaal 0 bijvoorbeeld op vol volume aan zowel de linker- als de rechterkant van de audio-uitgang af te spelen, stelt u zowel $D729 als $D71C in op $FF. Dit maakt het pannen van de vier audio-DMA-kanalen mogelijk.

Zowel de frequentie als het volume kunnen tijdens het afspelen van een sample vrij worden aangepast om verschillende effecten te verkrijgen.

Zuivere sinusgolf

Wanneer het nodig is een stabiele sinusgolf te produceren, met name bij hogere frequenties, voorziet een speciale modus hierin. Door het CH0SINE-signaal te zetten,

speelt het audiokanaal een sinusgolfpatroon van 32 bytes met 16-bits samples af. De sample-adressen moeten nog steeds worden ingesteld, alsof de sinusgolftabel zich in de onderste 64 bytes van het geheugen bevindt; de normale adresgeneratielogica wordt namelijk ook in deze modus gebruikt. In deze modus worden geen audio-DMA-fetches uitgevoerd, zodat alle vervormingsbronnen worden vermeden die het gevolg zijn van de onregelmatige spreiding van inactieve cycli in de instructiestroom van de processor.

Dit kan worden gebruikt om sinusgolven te produceren in zowel het hoorbare bereik als ver daarboven in het ultrasone bereik, bij frequenties die 60000 Hz overschrijden, mits de MEGA65 is aangesloten op een geschikte luidsprekeropstelling.

Regeling van het afspelen van samples

Om een kanaal een sample te laten afspelen, zet u het CH0EN-signaal ($D720 bit 7). Het sample wordt afgespeeld tot het zijn einde bereikt, tenzij het CH0LOOP-signaal eveneens is gezet. Wanneer een sample klaar is met afspelen, wordt de CH0STP-vlag gezet. Het audio-DMA-subsysteem kan op dit moment geen interrupts genereren.

Anders dan de Amiga™ biedt het audio-DMA-systeem van de MEGA65 ondersteuning voor zowel 8- als 16-bits samples. Het ondersteunt ook samengepakte 4-bits samples, waarbij van elk sample-byte de lagere of de hogere nibble wordt afgespeeld. Hierdoor kunnen twee afzonderlijke samples hetzelfde byte bezetten, waarmee de benodigde opslagruimte voor twee samples van gelijke lengte effectief wordt gehalveerd.

F018 "DMAGIC" DMA-CONTROLLER

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D700	55040	ADDRLSBTRIG							
D701	55041	ADDRMSB							
D702	55042	WITHIO	ADDRBANK						

- **ADDRBANK** Adresbank van de DMA-lijst (bits 16 - 22). Schrijven wist $D704.
- **ADDRLSBTRIG** DMAgic-LSB van het adres van de DMA-lijst (bits 0 - 7); start DMA wanneer erin wordt geschreven
- **ADDRMSB** Hoge byte van het adres van de DMA-lijst (bits 8 - 15)
- **WITHIO** DMA-withio-vlag (onbenut)

UITBREIDINGEN VAN DE MEGA65-DMA-CONTROLLER

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D703	55043	-						NOMB-WRAP	EN018B
D704	55044	ADDRMB							
D705	55045	ETRIG							
D706	55046	ETRIGMAPD							
D707	55047	ETRIGINLINE							
D70E	55054	ADDRLSB							
D711	55057	AUDEN	BLKD	AUD-WRBLK	NOMIX	-	AUDBLKTO		
D712	55058	DBG-LFTCLP	DBG-RGTCLP	-				CHANS-WAP	-
D713	55059	IRQ-CH0FLG	IRQ-CH1FLG	IRQ-CH2FLG	IRQ-CH3FLG	IRQ-CH0ENA	IRQ-CH1ENA	IRQ-CH2ENA	IRQ-CH3ENA
D714	55060	IRQCH0OFFS							
D715	55061	IRQCH1OFFS							
D716	55062	IRQCH2OFFS							
D717	55063	IRQCH3OFFS							
D71C	55068	CH0RVOL							
D71D	55069	CH1RVOL							
D71E	55070	CH2LVOL							
D71F	55071	CH3LVOL							
D720	55072	CH0EN	CH0LOOP	CH0SGN	CH0SINE	CH0STP	-	CH0SBITS	
D721	55073	CH0BADDRLSB							
D722	55074	CH0BADDRMSB							
D723	55075	CH0BADDRBNK							
D724	55076	CH0FREQ0							
D725	55077	CH0FREQ1							
D726	55078	CH0FREQ2							
D727	55079	CH0TADDRLSB							
D728	55080	CH0TADDRMSB							
D729	55081	CH0VOLUME							
D72A	55082	CH0CURADDRLSB							
D72B	55083	CH0CURADDRMSB							
D72C	55084	CH0CURADDRBNK							
D72D	55085	CH0TMGCNT0							
D72E	55086	CH0TMGCNT1							
D72F	55087	CH0TMGCNT2							
D730	55088	CH1EN	CH1LOOP	CH1SGN	CH1SINE	CH1STP	-	CH1SBITS	

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D731	55089	CH1BADDRLSB							
D732	55090	CH1BADDRMSB							
D733	55091	CH1BADDRBNK							
D734	55092	CH1FREQ0							
D735	55093	CH1FREQ1							
D736	55094	CH1FREQ2							
D737	55095	CH1TADDRLSB							
D738	55096	CH1TADDRMSB							
D739	55097	CH1VOLUME							
D73A	55098	CH1CURADDRLSB							
D73B	55099	CH1CURADDRMSB							
D73C	55100	CH1CURADDRBNK							
D73D	55101	CH1TMGCNT0							
D73E	55102	CH1TMGCNT1							
D73F	55103	CH1TMGCNT2							
D740	55104	CH2EN	CH2LOOP	CH2SGN	CH2SINE	CH2STP	–	CH2SBITS	
D741	55105	CH2BADDRLSB							
D742	55106	CH2BADDRMSB							
D743	55107	CH2BADDRBNK							
D744	55108	CH2FREQ0							
D745	55109	CH2FREQ1							
D746	55110	CH2FREQ2							
D747	55111	CH2TADDRLSB							
D748	55112	CH2TADDRMSB							
D749	55113	CH2VOLUME							
D74A	55114	CH2CURADDRLSB							
D74B	55115	CH2CURADDRMSB							
D74C	55116	CH2CURADDRBNK							
D74D	55117	CH2TMGCNT0							
D74E	55118	CH2TMGCNT1							
D74F	55119	CH2TMGCNT2							
D750	55120	CH3EN	CH3LOOP	CH3SGN	CH3SINE	CH3STP	–	CH3SBITS	
D751	55121	CH3BADDRLSB							
D752	55122	CH3BADDRMSB							
D753	55123	CH3BADDRBNK							
D754	55124	CH3FREQ0							
D755	55125	CH3FREQ1							
D756	55126	CH3FREQ2							
D757	55127	CH3TADDRLSB							
D758	55128	CH3TADDRMSB							
D759	55129	CH3VOLUME							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D75A	55130	CH3CURADDRLSB							
D75B	55131	CH3CURADDRMSB							
D75C	55132	CH3CURADDRBNK							
D75D	55133	CH3TMGCNT0							
D75E	55134	CH3TMGCNT1							
D75F	55135	CH3TMGCNT2							

- **ADDRLSB** Lage byte van het DMA-lijstadres (bits 0 - 7) ZONDER EEN DMA-TAAK TE STARTEN (door de Hypervisor gebruikt om taken die DMA gebruiken te ontdooien)
- **ADDRMB** Mega-byte van het adres van de DMA-lijst (bits 20 - 27). Overlapt met ADDRBANK!
- **AUDBLKTO** Time-out van het audio-DMA-blok (alleen-lezen) DEBUG
- **AUDEN** Schakelt audio-DMA in
- **AUDWRBLK** Audio-DMA-blokschrijfbewerkingen (samples worden nog steeds gelezen)
- **BLKD** Audio-DMA geblokkeerd (alleen-lezen) DEBUG
- **CH0RVOL** Volume van het rechterkanaal van audio-DMA-kanaal 0
- **CH1RVOL** Volume van het rechterkanaal van audio-DMA-kanaal 1
- **CH2LVOL** Volume van het linkerkanaal van audio-DMA-kanaal 2
- **CH3LVOL** Volume van het linkerkanaal van audio-DMA-kanaal 3
- **CHANSWAP** Omwisseling van de audio-DMA-kanalen
- **CHXBADDRBNK** Basisadres van audio-DMA-kanaal X (bits 16 - 23)
- **CHXBADDRLSB** Basisadres van audio-DMA-kanaal X (bits 0 - 7)
- **CHXBADDRMSB** Basisadres van audio-DMA-kanaal X (bits 8 - 15)
- **CHXCURADDRBNK** Huidig adres van audio-DMA-kanaal X (bits 16 - 23)
- **CHXCURADDRLSB** Huidig adres van audio-DMA-kanaal X (bits 0 - 7)
- **CHXCURADDRMSB** Huidig adres van audio-DMA-kanaal X (bits 8 - 15)
- **CHXEN** Schakelt audio-DMA-kanaal X in
- **CHXENA** Audio-DMA-IRQ van kanaal X ingeschakeld
- **CHXFLG** Vlag die wordt gezet wanneer een gebeurtenis de audio-DMA-IRQ van kanaal X activeert
- **CHXFREQ0** Frequentie van audio-DMA-kanaal X (bits 0 - 7)

- **CHXFREQ1** Frequentie van audio-DMA-kanaal X (bits 8 - 15)
- **CHXFREQ2** Frequentie van audio-DMA-kanaal X (bits 16 - 23)
- **CHXLOOP** Schakelt looping van audio-DMA-kanaal X in
- **CHXOFFS** IRQ-adresoffset van audio-DMA-kanaal X (bits 15 - 8)
- **CHXSBITS** Sample-bits van audio-DMA-kanaal X (11=16, 10=8, 01=hogere nibble, 00=lagere nibble)
- **CHXSINE** Audio-DMA-kanaal X speelt een sinusgolf van 32 samples af in plaats van DMA-gegevens
- **CHXSTP** Stopvlag van audio-DMA-kanaal X
- **CHXTADDRL** Topadres van audio-DMA-kanaal X (bits 0 - 7)
- **CHXTADDRM** Topadres van audio-DMA-kanaal X (bits 8 - 15)
- **CHXTMGCNT0** Timingteller van audio-DMA-kanaal X (bits 0 - 7)
- **CHXTMGCNT1** Timingteller van audio-DMA-kanaal X (bits 8 - 15)
- **CHXTMGCNT2** Timingteller van audio-DMA-kanaal X (bits 16 - 23)
- **CHXUSGN** Sample-teken van audio-DMA-kanaal X: 0=met teken, 1=zonder teken
- **CHXVOLUME** Afspeelvolume van audio-DMA-kanaal X
- **DBGLFTCLP** Audio-DMA links geklipt (alleen-lezen)
- **DBGRGTCLP** Audio-DMA rechts geklipt (alleen-lezen)
- **EN018B** Schakelt de F018B-modus van de DMA in (voegt een subcommandobyte toe)
- **ETRIG** Stel de lage byte van het DMA-lijstadres in (bits 0 - 7) en start de Enhanced DMA-taak met het lijstadres gespecificeerd als 28-bits flat adres (gebruikt de DMA-optielijst)
- **ETRIGINLINE** Start de Enhanced DMAgic-taak inline (d.w.z. beginnend bij de PC-waarde, waarbij de PC na afloop aan het einde van de taak wordt gezet)
- **ETRIGMAPD** Stel de lage byte van het DMA-lijstadres in (bits 0 - 7) en start de Enhanced DMA-taak met de lijst in de huidige CPU-geheugenindeling (gebruikt de DMA-optielijst)
- **NOMBWRAP** Schakelt de wrap-around van het bron- en doeladres op de MB-grenzen uit (wordt genegeerd in de Hypervisor-modus)
- **NOMIX** Audio-DMA omzeilt de audiomixer

NIET-GEÏMPLEMENTEERDE FUNCTIONALITEIT

De DMAgic van de MEGA65 ondersteunt op dit moment noch memory-swap- noch mini-term-bewerkingen.

Mini-terms waren bedoeld voor het blitten van bitplanes, hetgeen voor de MEGA65 niet nodig is, aangezien deze veel geavanceerdere tekstmodi en trapsgewijze en fractionele DMA-adresverhoging biedt, waarmee teksturen efficiënt kunnen worden gekopieerd en geschaald. Er is bovendien geen bekende software die ooit van deze voorziening gebruik heeft gemaakt, en het blijft onzeker of deze ooit in enige revisie van de DMAgic-chip in de C65-prototypen is geïmplementeerd.

De memory-swap-bewerking zal naar verwachting worden geïmplementeerd, maar kan in de tussentijd worden omzeild door het eerste gebied naar een derde gebied te kopiëren dat als tijdelijk buffer fungeert, vervolgens het tweede gebied naar het eerste, en het derde naar het tweede.

HOOFDSTUK 5

6526 Complex Interface Adapter (CIA)-registers

- CIA 6526-registers
- CIA 6526-hypervisorregisters

CIA 6526-REGISTERS

CIA 1-registers

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DC00	56320	PORTA							
DC01	56321	PORTB							
DC02	56322	DDRA							
DC03	56323	DDRB							
DC04	56324	TIMERA							
DC05	56325	TIMERA							
DC06	56326	TIMERB							
DC07	56327	TIMERB							
DC08	56328	-				TODJIF			
DC09	56329	-		TODSEC					
DC0A	56330	-		TODMIN					
DC0B	56331	TOD-AMPM	-		TODHOUR				
DC0C	56332	SDR							
DC0D	56333	IR	ISRCLR		FLG	SP	ALRM	TB	TA
DC0E	56334	TOD50	SPMOD	IMODA	-	RMODA	OMODA	PBONA	STRTA
DC0F	56335	TODEDIT	IMODB		LOAD	RMODB	OMODB	PBONB	STRTB

- **ALRM** TOD-alarm
- **DDRA** DDR van poort A
- **DDRB** DDR van poort B
- **FLG** FLAG-flank gedetecteerd
- **IMODA** Tickbron van Timer A
- **IMODB** Tickbron van Timer B
- **IR** Interruptvlag
- **ISRCLR** Plaatshouder - lezen wist de gebeurtenissen
- **LOAD** Strobe-invoer om de timers geforceerd te laden
- **OMODA** Omschakelen of puls voor Timer A
- **OMODB** Omschakelen of puls voor Timer B
- **PBONA** PB6-uitgang van timer A
- **PBONB** PB7-uitgang van timer B
- **PORTA** Poort A

- **PORTB** Poort B
- **RMODA** Timer A one-shotmodus
- **RMODB** Timer B one-shotmodus
- **SDR** Schuifregistergegevensregister (schrijven start het verzenden)
- **SP** Schuifregister vol/leeg
- **SPMOD** Richting van de seriële poort
- **STRTA** Start van timer A
- **STRTB** Start van timer B
- **TA** Onderloop van timer A
- **TB** Onderloop van timer B
- **TIMERA** Teller van timer A (16-bits)
- **TIMERB** Teller van timer B (16-bits)
- **TOD50** 50/60Hz-selectie voor de TOD-klok
- **TODAMPM** PM-vlag van de TOD
- **TODEDIT** TOD-alarm bewerken
- **TODHOUR** Uren van de TOD
- **TODJIF** Tienden van seconden van de TOD
- **TODMIN** Minuten van de TOD
- **TODSEC** Seconden van de TOD

CIA2-registers

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DD00	56576	PORTA							
DD01	56577	PORTB							
DD02	56578	DDRA							
DD03	56579	DDRB							
DD04	56580	TIMERA							
DD05	56581	TIMERA							
DD06	56582	TIMERB							
DD07	56583	TIMERB							
DD08	56584	-				TODJIF			
DD09	56585	-		TODSEC					
DD0B	56587	TOD-AMPM	-		TODHOUR				

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DD0C	56588	SDR							
DD0D	56589	IR	ISRCLR		FLG	SP	ALRM	TB	TA
DD0E	56590	TOD50	SPMOD	IMODA	–	RMODA	OMODA	PBONA	STRTA
DD0F	56591	TODEDIT	IMODB		LOAD	RMODB	OMODB	PBONB	STRTB

- **ALRM** TOD-alarm
- **DDRA** DDR van poort A
- **DDRB** DDR van poort B
- **FLG** FLAG-flank gedetecteerd
- **IMODA** Tickbron van Timer A
- **IMODB** Tickbron van Timer B
- **IR** Interruptvlag
- **ISRCLR** Plaatshouder - Lezen wist de gebeurtenissen
- **LOAD** Strobe-invoer om de timers met dwang te laden
- **OMODA** Omschakelen of puls voor Timer A
- **OMODB** Omschakelen of puls voor Timer B
- **PBONA** Timer A PB6-uitvoer
- **PBONB** Timer B PB7-uitvoer
- **PORTA** Poort A
- **PORTB** Poort B
- **RMODA** One-shot-modus van Timer A
- **RMODB** One-shot-modus van Timer B
- **SDR** Schuifregistergegevensregister (schrijven start het verzenden)
- **SP** Schuifregister vol/leeg
- **SPMOD** Richting van de seriële poort
- **STRTA** Start van timer A
- **STRTB** Start van timer B
- **TA** Onderloop van timer A
- **TB** Onderloop van timer B
- **TIMERA** Teller van timer A (16-bits)

- **TIMERB** Teller van timer B (16-bits)
- **TOD50** 50/60Hz-selectie voor de TOD-klok
- **TODAMPM** PM-vlag van TOD
- **TODEDIT** Bewerken van het TOD-alarm
- **TODHOUR** Uren van TOD
- **TODJIF** Tienden van seconden van TOD
- **TODSEC** Seconden van TOD

CIA 6526-HYPERVISORREGISTERS

Naast de standaard CIA-registers die op de C64 en C65 beschikbaar zijn, biedt de MEGA65 een extra reeks registers die alleen zichtbaar zijn wanneer de computer zich in de hypervisormodus bevindt. Deze extra registers maken het mogelijk om de interne toestand van de CIA bij het bevriezen volledig uit te lezen, waardoor meer programma's na het bevriezen correct blijven functioneren. Zij zijn niet zichtbaar wanneer u de MEGA65 op normale wijze gebruikt en kunnen veilig worden genegeerd door programmeurs die de MEGA65 niet in de hypervisormodus programmeren.

CIA1-hypervisorregisters

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DC10	56336	TALATCH							
DC11	56337	TALATCH							
DC12	56338	TALATCH							
DC13	56339	TALATCH							
DC14	56340	TALATCH							
DC15	56341	TALATCH							
DC16	56342	TALATCH							
DC17	56343	TALATCH							
DC18	56344	IMFLG	IMSP	IMALRM	IMTB	TODJIF			
DC19	56345	TODSEC							
DC1A	56346	TODMIN							
DC1B	56347	TOD-AMPM	TODHOUR						
DC1C	56348	DD00-DELAY	ALRMJIF						
DC1D	56349	ALRMSEC							
DC1E	56350	ALRMMIN							
DC1F	56351	ALRM-AMPM	ALRMHOUR						

- **ALRMAMPM** AM/PM-vlag van het TOD-alarm
- **ALRMHOUR** TOD-alarmwaarde voor uren
- **ALRMJIF** TOD-alarmwaarde voor tienden van seconden (eigenlijk alle 8 bits)
- **ALRMMIN** TOD-alarmwaarde voor minuten
- **ALRMSEC** TOD-alarmwaarde voor seconden
- **DD00DELAY** Schakel het vertragen van schrijfbewerkingen naar $DD00 met 3 cycli in zodat dit overeenkomt met de timing van de echte 6502
- **IMALRM** Interruptmasker voor het TOD-alarm
- **IMFLG** Interruptmasker voor de FLAG-lijn
- **IMSP** Interruptmasker voor het schuifregister (seriële poort)
- **IMTB** Interruptmasker voor Timer B
- **TALATCH** Latchwaarde van Timer A (16 bit)
- **TODAMPM** TOD-vlag voor AM/PM
- **TODHOUR** TOD-waarde voor uren
- **TODJIF** TOD-waarde voor tienden van seconden
- **TODMIN** TOD-alarmwaarde voor minuten
- **TODSEC** TOD-alarmwaarde voor seconden

CIA2-hypervisorregisters

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DD10	56592	TALATCH							
DD11	56593	TALATCH							
DD12	56594	TALATCH							
DD13	56595	TALATCH							
DD14	56596	TALATCH							
DD15	56597	TALATCH							
DD16	56598	TALATCH							
DD17	56599	TALATCH							
DD18	56600	IMFLG	IMSP	IMALRM	IMTB	TODJIF			
DD19	56601	TODSEC							
DD1A	56602	TODMIN							
DD1B	56603	TOD-AMPM	TODHOUR						
DD1C	56604	DD00-DELAY	ALRMJIF						
DD1D	56605	ALRMSEC							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DD1E	56606	ALRMMIN							
DD1F	56607	ALRM-AMPM	ALRMHOUR						

- **ALRMAMPM** AM/PM-vlag van het TOD-alarm
- **ALRMHOUR** Urenwaarde van het TOD-alarm
- **ALRMJIF** Tienden van seconden-waarde van het TOD-alarm (in feite alle 8 bits)
- **ALRMMIN** Minutenwaarde van het TOD-alarm
- **ALRMSEC** Secondenwaarde van het TOD-alarm
- **DD00DELAY** Uitstel van schrijfbewerkingen naar $DD00 met 3 cycli om de timing van een echte 6502 te benaderen
- **IMALRM** Interruptmasker voor het TOD-alarm
- **IMFLG** Interruptmasker voor de FLAG-lijn
- **IMSP** Interruptmasker voor het schuifregister (seriële poort)
- **IMTB** Interruptmasker voor timer B
- **TALATCH** Latchwaarde van timer A (16 bits)
- **TODAMPM** AM/PM-vlag van TOD
- **TODHOUR** Urenwaarde van TOD
- **TODJIF** Tienden van seconden-waarde van TOD
- **TODMIN** Minutenwaarde van het TOD-alarm
- **TODSEC** Secondenwaarde van het TOD-alarm

HOOFDSTUK 6

4551 UART, GPIO en utiliteitscontroller

- C65 6551 UART-registers
- Algemene I/O & diverse interface-registers van de 4551

C65 6551 UART-REGISTERS

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D600	54784	DATA							
D601	54785	-				FRMERR	PTYERR	RXOVR-RUN	RXRDY
D602	54786	TXEN	RXEN	SYNCMOD		CHARSZ		PTYEN	PTYEVEN
D603	54787	DIVISOR							
D604	54788	DIVISOR							
D605	54789	IMTXIRQ	IMRXIRQ	IMTXNMI	IMRXNMI	-			
D606	54790	IFTXIRQ	IFRXIRQ	IFTXNMI	IFRXNMI	-			

- **CHARSZ** Tekengrootte van de UART: 00=8, 01=7, 10=6, 11=5 bits per byte
- **DATA** Gegevensregister van de UART (lezen of schrijven)
- **DIVISOR** UART-baudratedeler (16 bit). Baudrate = 7.09375MHz / DIVISOR, tenzij de snelle UART-modus van de MEGA65 is ingeschakeld, in welk geval baudrate = 80MHz / DIVISOR
- **FRMERR** UART RX-framingfoutvlag (wissen door $D600 te lezen)
- **IFRXIRQ** UART-interruptvlag: IRQ op RX (nog niet geïmplementeerd op de MEGA65)
- **IFRXNMI** UART-interruptvlag: NMI op RX (nog niet geïmplementeerd op de MEGA65)
- **IFTXIRQ** UART-interruptvlag: IRQ op TX (nog niet geïmplementeerd op de MEGA65)
- **IFTXNMI** UART-interruptvlag: NMI op TX (nog niet geïmplementeerd op de MEGA65)
- **IMRXIRQ** UART-interruptmasker: IRQ op RX (nog niet geïmplementeerd op de MEGA65)
- **IMRXNMI** UART-interruptmasker: NMI op RX (nog niet geïmplementeerd op de MEGA65)
- **IMTXIRQ** UART-interruptmasker: IRQ op TX (nog niet geïmplementeerd op de MEGA65)
- **IMTXNMI** UART-interruptmasker: NMI op TX (nog niet geïmplementeerd op de MEGA65)
- **PTYEN** UART-pariteit inschakelen: 1=ingeschakeld
- **PTYERR** UART-vlag voor RX-pariteitsfout (wissen door $D600 te lezen)

- **PTYEVEN** UART-pariteit: 1=even, 0=oneven
- **RXEN** UART ontvangen inschakelen
- **RXOVRRUN** UART-vlag voor RX-overrun (wissen door $D600 te lezen)
- **RXRDY** UART-vlag dat RX-byte gereed is (wissen door $D600 te lezen)
- **SYNCMOD** UART-vlaggen voor synchronisatiemodus (00=RX & TX beide asynchroon, 01=RX synchroon, TX asynchroon, 1x=TX synchroon, RX asynchroon (ongebruikt op de MEGA65)
- **TXEN** UART verzenden inschakelen

ALGEMENE I/O & DIVERSE INTERFACE-REGISTERS VAN DE 4551

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D609	54793	-							UFAST
D60A	54794	KEY-QUEUE	MODKEY-CAPS	MODKEY-SCRL	MODKEY-ALT	MODKEY-MEGA	MOD-KEYCTRL	MOD-KEYRSHFT	MOD-KEYLSHFT
D60B	54795	OSKZEN	OSKZON	PORTF					
D60C	54796	PORTFDDR		PORTFDDR					
D60D	54797	HDSCL	HDSDA	SDBSH	SDCS	SDCLK	SDDATA	RST41	CONN41
D60E	54798	BASHDDR							
D60F	54799	ACCES-SKEY	OSKDIM	REALHW	-			KEYUP	KEYLEFT
D610	54800	ASCIIKEY							
D611	54801	MDISABLE	MCAPS	MSCRL	MALT	MMEGA	MCTRL	MRSHFT	MLSHFT
D612	54802	LJOYB	LJOYA	JOYSWAP	OSKDE-BUG	-			
D613	54803	KEYMATRIXPEEK							
D614	54804	KEYMATRIXSEL							
D615	54805	OSKEN	VIRTKEY1						
D616	54806	OSKALT	VIRTKEY2						
D617	54807	OSKTOP	VIRTKEY3						
D618	54808	KSCNRATE							
D619	54809	PETSCIIKEY							
D61A	54810	SYSCTL							
D61D	54813	KEYLED-ENA	KEYLEDREG						
D61E	54814	KEYLEDVAL							
D620	54816	POTAX							
D621	54817	POTAY							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D622	54818	POTBX							
D623	54819	POTBY							
D625	54821	J21L							
D626	54822	J21H							
D627	54823	J21LDDR							
D628	54824	BOARDMINOR				J21HDDR			
D629	54825	M65MODEL							

- **ACCESSKEY** Toegankelijke toetsenbordinvoer inschakelen via de vuurknop van joystickpoort 2
- **ASCIIKEY** Bovenkant van de typgebeurteniswachtrij als ASCII. Schrijven om de gebeurtenis te wissen en de volgende gereed te maken.
- **BASHDDR** Data Direction Register (DDR) voor de $D60D-bitbashingpoort.
- **BOARDMINOR** PCB-minorrevisie lezen (enkel R5+, anders worden nullen gelezen)
- **CONN41** Interne 1541-stationskoppeling (1=het interne 1541-station op de IEC-bus aansluiten)
- **HDSCL** HDMI I2C-besturingsinterface SCL-klok
- **HDSDA** HDMI I2C-besturingsinterface SDA-datalijn
- **J21H** J21-pinnen 11 - 14 invoer/uitvoerwaarden
- **J21HDDR** J21-pinnen 11 - 14 data direction register
- **J21L** J21-pinnen 1 - 6, 9 - 10 invoer/uitvoerwaarden
- **J21LDDR** J21-pinnen 1 - 6, 9 - 10 data direction register
- **JOYSWAP** Joystickpoorten 1 & 2 verwisselen
- **KEYLEDENA** Besturing van de toetsenbord-LED's inschakelen
- **KEYLEDREG** Registerkeuze voor de toetsenbord-LED's (R,G,B-kanalen x 4 = 0 tot 11)
- **KEYLEDVAL** Waarde van het toetsenbord-LED-register (enkel schrijven)
- **KEYLEFT** De C65 Cursor links-toets rechtstreeks lezen
- **KEYMATRIXPEEK** 8-bits segment van de gecombineerde toetsenbordmatrix (LEZEN)
- **KEYMATRIXSEL** Bepaalt welk 8-bits segment van de gecombineerde toetsenbordmatrix gelezen wordt.

- **KEYQUEUE** 1 = De typgebeurteniswachtrij is niet leeg. Schrijf 0 naar deze bit om de wachtrij te spoelen.
- **KEYUP** De C65 Cursor omhoog-toets rechtstreeks lezen
- **KSCNRATE** Fysieke scanfrequentie van het toetsenbord ($00=50MHz, $FF=∼200KHz)
- **LJOYA** Roteert de invoer van joystick A met 180 graden (voor linkshandig gebruik)
- **LJOYB** Roteert de invoer van joystick B met 180 graden (voor linkshandig gebruik)
- **M65MODEL** MEGA65-model-ID. Kan worden gebruikt om het model van de MEGA65 te bepalen waarop een programma wordt uitgevoerd, bijv. om aanraakbediening op de MEGAphone in te schakelen.
- **MALT** Toestand van de ALT-toets (direct; alleen lezen).
- **MCAPS** Toestand van de CAPS LOCK-toets (direct; alleen lezen).
- **MCTRL** Toestand van de CTRL-toets (direct; alleen lezen).
- **MDISABLE** Schakelt de modificatietoetsen uit.
- **MLSHFT** Toestand van de linker SHIFT-toets (direct; alleen lezen).
- **MMEGA** Toestand van de MEGA/C=-toets (direct; alleen lezen).
- **MODKEYALT** Toestand van de ALT-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MODKEYCAPS** Toestand van de CAPS LOCK-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MODKEYCTRL** Toestand van de CTRL-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MODKEYLSHFT** Toestand van de linker SHIFT-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MODKEYMEGA** Toestand van de MEGA/C=-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MODKEYRSHFT** Toestand van de rechter SHIFT-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MODKEYSCRL** Toestand van de NOSCRL-toets bovenaan de wachtrij van typegebeurtenissen. 1 = ingedrukt tijdens de gebeurtenis.
- **MRSHFT** Toestand van de rechter SHIFT-toets (direct; alleen lezen).
- **MSCRL** Toestand van de NOSCRL-toets (direct; alleen lezen).
- **OSKALT** Toont de alternatieve indeling van het schermtoetsenbord (gewoonlijk het nummerpad van de MEGA65-telefoon)

- **OSKDEBUG** Debugt de OSK-overlay (alleen schrijven)
- **OSKDIM** Dimt het achtergrondmateriaal achter het schermtoetsenbord licht of sterk
- **OSKEN** Schakelt de weergave van de samengestelde overlay van het schermtoetsenbord in
- **OSKTOP** 1=toont het schermtoetsenbord bovenaan, 0=toont het schermtoetsenbord onderaan het scherm.
- **OSKZEN** Toont de hardwarezoom van het gebied onder het eerste aanraakpunt voor het schermtoetsenbord
- **OSKZON** Toont altijd de hardwarezoom van het gebied onder het eerste aanraakpunt
- **PETSCIIKEY** Bovenaan de wachtrij van typegebeurtenissen als PETSCII. Schrijf erin om de gebeurtenis te wissen en gereed te maken voor de volgende.
- **PORTF** PMOD-poort A op het FPGA-bord (gegevens) (alleen Nexys4-borden)
- **PORTFDDR** PMOD-poort A op het FPGA-bord (DDR)
- **POTAX** Leest de paddle X van poort A, zonder dat u met de SID/CIA-instellingen hoeft te rommelen.
- **POTAY** Leest de paddle Y van poort A, zonder dat u met de SID/CIA-instellingen hoeft te rommelen.
- **POTBX** Leest de paddle X van poort B, zonder dat u met de SID/CIA-instellingen hoeft te rommelen.
- **POTBY** Leest de paddle Y van poort B, zonder dat u met de SID/CIA-instellingen hoeft te rommelen.
- **REALHW** Zet op 1 indien de MEGA65 op echte hardware wordt uitgevoerd, zet op 0 indien geëmuleerd (Xemu) of gesimuleerd (ghdl).
- **RST41** Interne reset van het 1541-schijfstation (1=reset, 0=operationeel)
- **SDBSH** Bitbash-modus van de SD-kaart inschakelen
- **SDCLK** SD-kaart SCLK
- **SDCS** SD-kaart CS_BO
- **SDDATA** SD-kaart MOSI/MISO
- **SYSCTL** Systeembeheervlagen (doelapparaatspecifiek)
- **UFAST** UART-baudklokbron van de C65: 1 = 7,09375 MHz, 0 = 80 MHz (VIC-IV-pixelpitch)
- **VIRTKEY1** Op $7F zetten voor geen toetsaanslag; geef anders het virtuele toetsaanslag aan

- **VIRTKEY2** Op $7F zetten voor geen toetsaanslag; geef anders de 2e virtuele toetsaanslag aan
- **VIRTKEY3** Op $7F zetten voor geen toetsaanslag; geef anders de 3e virtuele toetsaanslag aan

HOOFDSTUK 7

45E100 Fast Ethernet-controller

- Overzicht
- In het geheugen gemapte registers
- Voorbeeldprogramma's

OVERZICHT

De 45E100 is een eenvoudige Fast Ethernet-controller die speciaal voor de MEGA65 en voor 8-bit computers in het algemeen is ontworpen. Naast het ondersteunen van 100 Mbit Fast Ethernet verschilt hij radicaal van andere Ethernet-controllers, zoals de RR-NET.

De 45E100 heeft 32 ontvangstbuffers op MEGA65 R3- en latere borden (andere borden kunnen er minder hebben, bijvoorbeeld vier op de Nexys4 DDR), waardoor tot 31 frames ontvangen kunnen worden terwijl een ander wordt verwerkt, of waardoor interrupts minder frequent verwerkt hoeven te worden. Deze ontvangstbuffers kunnen in het geheugen gemapt worden en ook direct benaderd worden via de DMA-controller van de MEGA65. Samen met automatische CRC32-controle bij ontvangst en automatische CRC32-generatie bij verzending, verlagen deze mogelijkheden de belasting van de processor aanzienlijk en maken ze het schrijven van programma's met Ethernet-ondersteuning veel eenvoudiger.

De 45E100 ondersteunt tevens echte full-duplex-werking met 100 Mbit per seconde, waardoor de totale bidirectionele doorvoer de 100 Mbit per seconde kan overschrijden. Het MAC-adres is softwarematig configureerbaar en promiscue modus wordt ondersteund, evenals het individueel aansturen van de ontvangst van broadcast- en multicast Ethernet-frames.

De 45E100 ondersteunt tevens zowel verzend- als ontvangstinterrupts, wat de prestaties in de praktijk aanzienlijk verbetert. Indien bijzonder lage latentie vereist is, kan de verzending van het huidige Ethernet-frame ook onmiddellijk afgebroken worden, zodat een frame met een hogere prioriteit onmiddellijk verstuurd kan worden. Deze mogelijkheden samen maken roundtriplatenties van minder dan een milliseconde mogelijk, wat vooral waardevol is voor interactieve toepassingen, zoals netwerkspellen voor meerdere spelers.

Verschillen met de RR-NET en vergelijkbare oplossingen

De RR-NET en andere Ethernet-controllers voor de serie 8-bit-homecomputers van de Commodore™ gebruiken doorgaans een Ethernet-controller die voor 16-bit pc's is ontworpen maar ook een zogenaamde "8-bit-modus" ondersteunt, die kampt met een aantal nadelen. Deze nadelen omvatten het ontbreken van werkende interrupts evenals processorintensieve toegang tot de Ethernet-framebuffers. Het gebrek aan interrupts dwingt programma's ertoe te pollen om de aankomst van nieuwe Ethernet-frames te controleren. Dit, gecombineerd met de complexiteit van het benaderen van de buffers, resulteert in een Ethernet-interface die zeer traag is en waarvan de praktijkdoorvoer aanzienlijk lager is dan de theoretische 10 Mbit per seconde. Zelfs een Commodore 64 met REU haalt geen snelheden boven enkele tientallen kilobytes per seconde.

Daarentegen ondersteunt de 45E100 zowel RX- (Ethernet-frame ontvangen) als TX- (klaar om te verzenden) interrupts, waarbij de processor het apparaat niet langer hoeft te pollen. Omdat de 45E100 RX-interrupts ondersteunt, is er geen behoefte aan grote

aantallen ontvangstbuffers, en daarom heeft de 45E100 slechts een klein aantal RX-buffers nodig om zeer hoge prestatieniveaus te bereiken.

Bovendien ondersteunt de 45E100 directe geheugentoegang tot de Ethernet-framebuffers, wat aanzienlijk efficiënter benaderen mogelijk maakt, ook via DMA. Met de geïntegreerde DMA-controller van de MEGA65 is het heel goed mogelijk transfersnelheden van enkele megabytes per seconde te bereiken – ongeveer 100 keer sneller dan de RR-NET.

Werking: het ontvangen van frames

De 45E100 is eenvoudig te bedienen: om het ontvangen van Ethernet-frames te beginnen, hoeft de programmeur alleen de RST- en TXRST-bits (bit 0 van register $D6E0) te wissen om de Ethernet-controller te resetten en vervolgens deze bits op 1 te zetten om de controller uit de resettoestand te bevrijden. De controller onderhandelt vervolgens automatisch een verbinding op de hoogst beschikbare snelheid, doorgaans 100 Mbit, full-duplex.

Indien u eenvoudig door pollen wilt controleren of er Ethernet-frames zijn aangekomen, controleer dan de RXQ-bit (bit 5 van $D6E1). Indien deze gezet is, is er ten minste één frame ontvangen. Om het volgende ontvangen frame te benaderen, schrijf $01 naar $D6E1 en vervolgens $03 naar $D6E1. Hiermee draait u de ring van ontvangstbuffers, zodat het volgende ontvangen frame door de processor benaderd kan worden. De ontvangstbuffer die voorheen door de processor benaderd kon worden, wordt als vrij gemarkeerd, en de 45E100 zal deze gebruiken om een volgend Ethernet-frame te ontvangen wanneer dat nodig is.

Omdat de 45E100 meerdere ontvangstbuffers heeft, is het mogelijk om meerdere frames achter elkaar te verwerken door deze procedure te volgen. Indien alle ontvangstbuffers ontvangen frames bevatten en de processor deze niet heeft aanvaard, dan wordt het RXBLKD-signaal geactiveerd, zodat de processor weet dat verdere ontvangen frames verloren zullen gaan. Programmeurs moeten voorzichtig zijn om deze situatie te vermijden. Omdat de 45E100 ontvangstinterrupts ondersteunt, is dit doorgaans eenvoudig te beheren – onderschat echter niet hoe vaak Ethernet-frames op een 100 Mbit Fast Ethernet-verbinding kunnen binnenkomen: wanneer een zender een continue stroom van frames van minimale lengte stuurt, kunnen deze elke 6 microseconden aankomen! Hoewel het onwaarschijnlijk is dat u met zo'n hoge ontvangstfrequentie te maken krijgt, moet u er rekening mee houden dat pakketten ten minste elke milliseconde verwerkt moeten worden. Met name een CIA- of raster-IRQ die eenmaal per frame optreedt, kan ertoe leiden dat enkele pakketten verloren gaan wanneer er meer dan drie in een videoframe van 16-20 ms aankomen. Het RXBLKD-signaal kan gebruikt worden om te bepalen of deze situatie zich waarschijnlijk heeft voorgedaan. Let er echter op dat het alleen aangeeft wanneer alle ontvangstbuffers bezet zijn, en niet of er nog extra frames arriveerden terwijl er geen vrije ontvangstbuffers beschikbaar waren.

De ontvangstbuffers zijn elk 2 KB groot en kunnen slechts één ontvangen Ethernet-frame tegelijk bevatten. Dit verschilt van enkele Ethernet-controllers die hun totale ontvangstbuffergeheugen als één simpele ringbuffer gebruiken. De reden hiervoor is

het mechanisme voor programmeurs zo eenvoudig mogelijk te houden. Door de vaste buffers kan de controller de ontvangen Ethernet-frames elke keer op exact dezelfde locatie in het geheugen mappen, waardoor het mogelijk is veel eenvoudigere ontvangstprogramma's te schrijven, omdat de locatie van de ontvangen Ethernet-frames constant verondersteld kan worden.

De structuur van een ontvangstbuffer met een Ethernet-frame is vrij eenvoudig: de eerste twee bytes geven de lengte van het ontvangen frame aan. Het frame volgt dan onmiddellijk. De effectieve Maximum Transport Unit (MTU)-lengte is 2042 bytes, aangezien de laatste vier bytes bezet zijn door de CRC32-controlesom van het ontvangen Ethernet-frame. De indeling van de ontvangstbuffers is als volgt:

HEX	DEC	Lengte	Omschrijving
0000	0	1	Het lage byte van de lengte van het ontvangen Ethernet-frame.
0001	1	1	Bits 0-2 bevatten de bovenste drie bits van de lengte van het ontvangen Ethernet-frame. Bit 3 is altijd 0. Bit 4 is gezet wanneer het ontvangen Ethernet-frame een multicast-frame is. Bit 5 is gezet wanneer het een broadcast-frame betreft. Bit 6 is gezet wanneer het bestemmingsadres van het frame overeenkomt met het geprogrammeerde MAC-adres van de 45E100. Bit 7 is gezet wanneer de CRC32-controle voor het ontvangen frame is mislukt, d.w.z. dat het frame of is afgekapt of tijdens de verzending beschadigd is geraakt.
0002 - 07FB	2 - 2,043	2,042	Het ontvangen frame. Frames korter dan 2042 bytes beginnen op offset 2.
07FC - 07FF	2,044 - 2,047	4	Gereserveerde ruimte voor het bewaren van de CRC32-code tijdens het ontvangen. De CRC32-code bevindt zich echter altijd direct na het ontvangen frame en neemt deze ruimte dus alleen in wanneer het ontvangen frame langer is dan 2038 bytes.

Vanwege het zeer hoge tempo waarin Fast Ethernet-frames kunnen worden ontvangen, dient een programmeur gebruik te maken van de ontvangstinterrupt-voorziening, die wordt ingeschakeld door RXQEN (bit 7 van $D6E1) te zetten. Polling is als alternatief mogelijk, maar wordt bij de 45E100 niet aangeraden, omdat bij de 100 Mbit-snelheid van Fast Ethernet pakketten om de 5 microseconden kunnen aankomen. Gelukkig is het mogelijk om dergelijke hoge gegevenssnelheden bij te houden in de volledige snelheidsmodus van 40 MHz van de MEGA65 en met behulp van de DMA-kopieerfunctie van 20 MB per seconde.

Toegang tot de Ethernet-framebuffers

Anders dan bij de RR-NET kunnen de Ethernet-framebuffers van de 45E100 in het geheugen worden gemapt, waardoor snelle toegang via DMA of via assembleertaalprogramma's mogelijk is. Het is eveneens mogelijk de buffers met enige voorzichtigheid vanuit BASIC te benaderen.

De frame-buffers kunnen ofwel worden benaderd vanaf hun natuurlijke locatie in de uitgebreide adresruimte van de MEGA65 op adres $FFDE800 - $FFDEFFF, ofwel kunnen ze worden gemapt in de normale C64/C65-I/O-adresruimte op $D800 - $DFFF. Hierbij is voorzichtigheid geboden, aangezien het mappen van de Ethernet-framebuffers in de I/O-adresruimte ertoe leidt dat alle andere I/O-apparaten gedurende die tijd niet beschikbaar zijn. Om die reden dienen CIA-gebaseerde interrupts te worden uitgeschakeld voordat u dit doet, of u nu BASIC of machinetaal gebruikt. Voor het programmeren in assembleertaal of machinetaal wordt daarentegen aangeraden de natuurlijke locatie te gebruiken en dit geheugengebied te benaderen via één van de drie mechanismen voor toegang tot de uitgebreide adresruimte, welke worden beschreven in *het grote Mega65-boek*, Toegang tot geheugen buiten het 1MB-punt (deelparagraaf K).

De methode om interrupts uit te schakelen verschilt al naar gelang de context waarin een programma wordt geschreven. Voor programma's die worden geschreven met BASIC 2 van de C64-modus werkt het volgende:

```
POKE56333,127: REM SCHAKEL TIMER-IRQS VAN CIA UIT
```

Terwijl voor BASIC 65 van de MEGA65 in plaats daarvan het volgende moet worden gebruikt, omdat een VIC-III-rasterinterrupt wordt gebruikt in plaats van een op CIA gebaseerde timer-interrupt:

```
POKE53274,0: REM SCHAKEL RASTER-IRQS VAN VIC-II/III/IV UIT
```

Wanneer dit eenmaal is gebeurd, kan de I/O-context voor de Ethernet-controller worden geactiveerd door $45 (69 in decimaal, gelijk aan het teken 'E' in PETSCII) en $54 (84 in decimaal, gelijk aan het teken 'T' in PETSCII) in het KEY-register van de VIC-IV ($D02F, 53295 in decimaal) te schrijven, bijvoorbeeld:

```
POKE53295,ASC("E"):POKE53295,ASC("T")
```

Op dit punt kan de Ethernet-RX-buffer worden gelezen vanaf locatie $D800 (55296 in decimaal) en kan in de TX-buffer op hetzelfde adres worden geschreven. Raadpleeg "Werking: het ontvangen van frames" hierboven voor nadere uitleg hierover.

Wanneer u klaar bent met het benaderen van de Ethernet-framebuffer, kunt u de normale C64-, C65- of MEGA65-I/O-context herstellen door in het KEY-register van de VIC-III/IV te schrijven. In de meeste gevallen is het het meest zinvol om terug te keren naar de I/O-context van de MEGA65 door eerst $47 (71 in decimaal) en vervolgens $53 (83 in decimaal) in het KEY-register te schrijven, bijvoorbeeld:

```
POKE53295,ASC("G"):POKE53295,ASC("S")
```

Tot slot dient u vervolgens de interrupts weer in te schakelen, hetgeen opnieuw afhangt van de vraag of u programmeert vanuit de C64- of de C65-modus. Voor de C64-modus:

```
POKE56333,129
```

Voor de C65-modus zou dit zijn:

```
POKE53274,129
```

Werking: het verzenden van frames

Het verzenden van frames is eveneens eenvoudig: het programma moet het te verzenden frame eenvoudigweg in de zendbuffer laden, de lengte ervan in de TXSZLSB- en TXSZMSB-registers schrijven en vervolgens $01 in het COMMAND-register schrijven. Het frame begint daarna met verzenden, zodra de zender inactief is. Het is niet nodig een Ethernet-CRC32-veld te berekenen en toe te voegen, aangezien de 45E100 dit automatisch doet.

In tegenstelling tot bij de ontvanger is er voor de zender slechts één frame-buffer (dit kan in een toekomstige revisie worden gewijzigd). Dit betekent dat u het volgende frame niet kunt voorbereiden totdat het vorige frame reeds is verzonden. Dit vermindert de maximale gegevensdoorvoer enigszins, in ruil voor een zeer eenvoudige architectuur.

Houd er tevens rekening mee dat de zendbuffer vanuit de processorbusinterface uitsluitend schrijfbaar is. Dit betekent dat u de inhoud van de zendbuffer niet direct kunt lezen, maar dat u de waarden "blind" moet laden. Tot slot stelt de 45E100 u in staat om Ethernet-frames te verzenden.

Geavanceerde functies

Naast het functioneren als een eenvoudige en efficiënte Ethernet-framezender en -ontvanger, beschikt de 45E100 over een aantal geavanceerde functies die hier worden beschreven.

Broadcast- en multicast-verkeer en promiscue modus

De 45E100 ondersteunt filtering op basis van het Ethernet-bestemmingsadres, d.w.z. het MAC-adres. Standaard worden alleen frames ontvangen waarvan het Ethernet-bestemmingsadres overeenkomt met het Ethernet-adres dat in de registers MACADDR1 – MACADDR6 is geprogrammeerd. Indien de MCST-bit is gezet, worden echter ook multicast-Ethernet-frames ontvangen. Evenzo staat het instellen van de BCST-bit alle broadcast-frames toe, d.w.z. frames met het MAC-adres ff:ff:ff:ff:ff:ff. Tenslotte wordt het filter volledig uitgeschakeld indien de NOPROM-bit is gewist; de 45E100 ontvangt dan alle geldige Ethernet-frames.

Foutopsporings- en diagnosemogelijkheden

De 45E100 ondersteunt tevens een aantal functies om bij te dragen aan het opsporen van Ethernet-problemen. Indien de NOCRC-bit is gezet, worden ook Ethernetframes met ongeldige CRC32-waarden ontvangen. Dit kan helpen bij het opsporen van defecte Ethernet-apparaten in een netwerk.

Indien de STRM-bit is gezet, verzendt de Ethernet-zender een continue stroom van foutopsporingsframes via een speciale logging-interface met hoge bandbreedte. Standaard genereert de 45E100 een stroom van Ethernet-frames van ongeveer 2200 bytes, die gecomprimeerde video bevatten die wordt aangeboden door een VIC-IV of compatibele videocontroller die de video-over-Ethernet-interface van de MEGA65 ondersteunt. Door een aangepaste decoder voor deze stroom van Ethernet-frames te schrijven, is het mogelijk een extern beeldscherm van de MEGA65 via Ethernet te realiseren. Zo'n extern beeldscherm kan bijvoorbeeld worden gebruikt om de weergave van een MEGA65 digitaal vast te leggen.

De grootte en inhoud van de foutopsporingsframes kunnen worden beheerd door specifieke waarden in het COMMAND-register te schrijven. Het schrijven van $F1 staat de selectie toe van frames met een lengte van ongeveer 1200 bytes. Hoewel dit de prestaties van de foutopsporings- en streamingmogelijkheden vermindert, maakt het de ontvangst van deze frames mogelijk op systemen waarvan de Ethernet-controllers niet kunnen worden geconfigureerd om frames van ongeveer 2200 bytes te ontvangen.

Indien de STRM-bit is gezet en bit 2 van $D6E1 ook is gezet, wordt in plaats daarvan een gecomprimeerd logboek van de door de 45GS02-processor uitgevoerde instructies gestreamd, mits een compatibele processor op deze interface is aangesloten. Dit mechanisme oefent tegendruk uit en vertraagt de 45GS02-processor zodat de instructiegegevens kunnen worden uitgezonden. Doorgaans beperkt dit de snelheid van de aangesloten 45GS02-processor tot ongeveer 5 MHz, afhankelijk van de specifieke instructiemix.

Opmerking: de status van bit 2 van $D6E1 kan momenteel niet direct worden uitgelezen. Dit wordt mogelijk in een toekomstige revisie gecorrigeerd.

Tenslotte maakt het inschakelen van de videofunctionaliteit ook de ontvangst van synthetische toetsenbordgebeurtenissen via Ethernet mogelijk. Deze worden aangeboden aan de Keyboard Complex Interface Adapter (KCIA) van de MEGA65, waardoor volledige interactie op afstand met een MEGA65 via de Ethernet-interface mogelijk is. Deze functie is primair bedoeld voor ontwikkelingsdoeleinden.

IN HET GEHEUGEN GEMAPTE REGISTERS

De 45E100 Fast Ethernet-controller is een MEGA65-specifieke voorziening. Hij is daarom alleen beschikbaar in de MEGA65 I/O-context. Deze wordt ingeschakeld

door $47 en vervolgens $53 te schrijven naar het VIC-IV-register $D02F. Bij het programmeren in BASIC kan dit als volgt worden gedaan:

```
POKE53295,ASC("G"):POKE53295,ASC("S")
```

De 45E100 Fast Ethernet-controller heeft de volgende registers

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D02F	53295	KEY							
D6E0	55008	TXIDLE	RXBLKD	-	RCENAB-LED	DRXDV	DRXD	TXRST	RST
D6E1	55009	RXQEN	TXQEN	RXQ	TXQ	STRM	RXBF		-
D6E2	55010	TXSZLSB							
D6E3	55011	TXSZMSB							
D6E4	55012	COMMAND							
D6E5	55013	RXPH		MCST	BCST	TXPH		NOCRC	NOPROM
D6E6	55014	MIIMPHY			MIIMREG				
D6E7	55015	MIIMVLSB							
D6E8	55016	MIIMVMSB							
D6E9	55017	MACADDR1							
D6EA	55018	MACADDR2							
D6EB	55019	MACADDR3							
D6EC	55020	MACADDR4							
D6ED	55021	MACADDR5							
D6EE	55022	MACADDR6							

- **BCST** Aanvaard broadcast-frames
- **COMMAND** Ethernet-commandoregister (alleen schrijven)
- **DRXD** Lees de ethernet-RX-bits die op dit moment op de draad staan
- **DRXDV** Lees of de ethernet-RX-gegevens geldig zijn (debug)
- **KEY** Schrijf $45 en daarna $54 om de buffers van de 45E100-ethernetcontroller naar $D000-$DFFF te mappen
- **MACADDRX** Ethernet-MAC-adres
- **MCST** Aanvaard multicast-frames
- **MIIMPHY** Ethernet-MIIM-PHY-nummer (gebruik 0 voor Nexys4 en 1 voor MEGA65-r1-PCB's)
- **MIIMREG** Ethernet-MIIM-registernummer
- **MIIMVLSB** Ethernet-MIIM-registerwaarde (LSB)
- **MIIMVMSB** Ethernet-MIIM-registerwaarde (MSB)
- **NOCRC** Schakel de CRC-controle voor ontvangen pakketten uit

- **NOPROM** Schakel de promiscue modus van ethernet uit
- **RCENABLED** (alleen lezen) inschakelstatus van de ethernet-afstandsbediening
- **RST** Schrijf 0 om de ethernetcontroller in reset te houden
- **RXBF** Aantal vrije ontvangstbuffers
- **RXBLKD** Geeft aan of ethernet-RX is geblokkeerd totdat RX-buffers vrijkomen
- **RXPH** Ethernet-RX-klokfase-instelling
- **RXQ** Ethernet-RX-IRQ-status
- **RXQEN** Schakel de ethernet-RX-IRQ in
- **STRM** Schakel het streamen van de CPU-instructiestroom of de VIC-IV-weergave via ethernet in
- **TXIDLE** De ethernet-zendzijde is inactief, d.w.z. er kan een pakket worden verzonden.
- **TXPH** Ethernet-TX-klokfase-instelling
- **TXQ** Ethernet-TX-IRQ-status
- **TXQEN** Schakel de ethernet-TX-IRQ in
- **TXRST** Schrijf 0 om het zendsubsysteem van de ethernetcontroller in reset te houden
- **TXSZLSB** TX-pakketgrootte (lage byte)
- **TXSZMSB** TX-pakketgrootte (hoge byte)

Waarden van het COMMAND-register

De volgende waarden kunnen in het COMMAND-register worden geschreven om de beschreven functies uit te voeren. In normaal gebruik is alleen het STARTTX-commando nodig, bijvoorbeeld door de volgende **POKE** uit te voeren:

```
POKE55012,1
```

HEX	DEC	Signaal	Omschrijving
00	0	STOPTX	Stop onmiddellijk met het verzenden van het huidige Ethernet-frame. De zender stuurt een gedeeltelijk verzonden frame, waardoor dit frame zeer waarschijnlijk verloren gaat.
01	1	STARTTX	Zend pakket
D0	208	RXNORMAL	Schakel de effecten van RXONLYONE uit

vervolgd ...

...vervolgd

HEX	DEC	Signaal	Omschrijving
D4	212	DEBUGVIC	Selecteer de VIC-IV-foutopsporingsstroom via Ethernet wanneer $D6E1.3 is gezet
DC	220	DEBUGCPU	Selecteer de CPU-foutopsporingsstroom via Ethernet wanneer $D6E1.3 is gezet
DE	222	RXONLYONE	Ontvang precies één Ethernet-frame en behoud alle signaaltoestanden (voor het opsporen van fouten in het Ethernet-subsysteem)
F1	241	FRAME1K	Selecteer frames van 1 KiB voor de video-/CPU-foutopsporingsstroom (voor ontvangers die MTU's groter dan 2 KiB niet ondersteunen)
F2	242	FRAME2K	Selecteer frames van 2 KiB voor de video-/CPU-foutopsporingsstroom, voor optimale prestaties.

VOORBEELDPROGRAMMA'S

Voorbeeldprogramma's voor de Ethernet-controller bevinden zich in de MEGA65 Core-repository op GitHub in de mappen src/tests en src/examples, in wisselende staat van voltooiing.

Indien u de Ethernet-controller wilt gebruiken voor TCP/IP-verkeer, kunt u de port van WeeIP naar de MEGA65 bekijken op https://github.com/mega65/mega65-weeip. De code die de Ethernet-controller aanstuurt, bevindt zich in `eth.c`.

HOOFDSTUK 8

45IO27 multifunctionele I/O-controller

- Overzicht
- F011-compatibele schijfstationcontroller
- SD-kaartcontroller en F011-virtualisatiefuncties
- Touchpadinterface
- Ondersteunende functies voor audio
- Diverse I/O-functies

OVERZICHT

De 45IO27 is een multifunctionele I/O-controller die de functies van de F011-schijfstationcontroller van de C65 verenigt met de interface van de SD-kaartcontroller van de MEGA65 en een aantal andere uiteenlopende I/O-functies.

Elk van deze hoofdfuncties wordt in een afzonderlijke sectie van dit hoofdstuk behandeld.

F011-COMPATIBELE SCHIJFSTATIONCONTROLLER

De MEGA65-computer is een van de zeer weinige moderne computers die nog steeds volwaardige ondersteuning bieden voor magnetische schijfstations. Hij bevat een schijfstationcontroller die achterwaarts compatibel is met de F011D-schijfstationcontroller van de C65.

Anders dan de F011D ondersteunt de schijfstationcontroller van de MEGA65 HD- en ED-media. Net als het 1541-schijfstation ondersteunt deze ook variabele datatempo's, zodat een vastberaden gebruiker schijfformaten kan ontwikkelen die meer gegevens opslaan, robuuste kopieerbeveiligingsschema's bevatten of beide.

Ook andere schijfformaten kunnen worden gebruikt, zij het met wisselende inspanning, zoals het GCR-schema van de 1541 en de 1571 of het track-at-once-MFM-schema dat de Amiga™-familie van computers gebruikt. Dit is mogelijk dankzij de track-DMA-functionaliteit die de 45IO27 en de 45GS02-processor gezamenlijk bieden. Deze twee apparaten werken samen om ruwe fluxsignalen van en naar diskettes te lezen of te schrijven, waardoor vrijwel elk schijfformaat kan worden gelezen of gereproduceerd.

Voor Amiga™-diskettes bevat de 45IO27 bovendien experimentele speciale schakelingen die het lezen van dergelijke diskettes vergemakkelijken. Zo kan de 45IO27 de unieke sectorsynchronisatiesignalen herkennen die op Amiga™-diskettes worden gebruikt en de gegevenssectoren lezen. Hij haalt echter niet de 16 bytes besturingssysteemspecifieke gegevens per sector op en voert ook geen de-interleaving uit van de gescheiden vectoren van oneven en even bits ten opzichte van de volgorde waarin de Amiga™-computer ze op de schijf schrijft. Het gebruik van deze speciale schakelingen is optioneel. De track-DMA-functionaliteit kan dus ook worden gebruikt om Amiga™-diskettes te lezen.

Ondersteuning voor meerdere schijfstations

Net als de F011-schijfstationcontroller van de C65 ondersteunt de 45IO27 maximaal acht stations. Van die stations wordt aangenomen dat de eerste twee, station 0 en station 1, zijn aangesloten op een standaard 34-pins diskettekabel, dezelfde als in

standaard-pc's, dus met een draai in de kabel zodat er twee stations zonder jumpers kunnen worden gebruikt.

Zoals in latere secties wordt beschreven, kunnen de posities van station 0 en station 1 worden omgewisseld zonder dat de bekabeling hoeft te worden gewijzigd. Evenzo kan elk van de eerste twee stations verwijzen naar een echt schijfstation of naar een op een aangesloten SD-kaart opgeslagen D81-schijfimage, en kan het ook worden omgeleid naar de virtualisatiedienst voor schijfstations, zodat de sectortoegangen door een aangesloten computer kunnen worden afgehandeld, bijvoorbeeld als onderdeel van een comfortabele en efficiënte cross-development-omgeving.

De overige zes stations worden alleen ondersteund in combinatie met een toekomstige C1565-compatibele externe stationpoort.

Bewerkingen met gebufferde sectoren

De 45IO27 ondersteunt twee manieren om sectoren van een schijf te lezen: byte voor byte en via een in het geheugen ingemapte sectorbuffer.

Het mechanisme byte voor byte bestaat uit een lus die wacht totdat het DRQ-signaal actief wordt en daarna het databyte uit het DATA-register ($D087) leest.

Bij de methode met de in het geheugen ingemapte sectorbuffer wacht u totdat de BUSY-vlag wordt gewist. Wanneer deze gewist is, is de volledige sector gelezen en kunt u direct de sectorbuffer op $FFD6C00 - $FFD6DFF benaderen. Let erop dat het BUFSEL-signaal (bit 7 van $D689) is gewist, zodat voor andere programma's dan de Hypervisor de diskettesectorbuffer zichtbaar is in plaats van de SD-kaart-sectorbuffer. Dit komt doordat alleen de Hypervisor toegang heeft tot de volledige 4KB-bufferruimte van de SD-controller: gewone programma's zien ofwel de diskettesectorbuffer ofwel de SD-kaart-sectorbuffer, acht keer herhaald tussen $FFD6000 en $FFD6FFF.

Alternatief kan de sectorbuffer worden ingemapt op $DE00 - $DFFF, d.w.z. in het 4KB-I/O-gebied, door $81 naar het SD-commandoregister op $D680 te schrijven. Eventuele I/O-perifierieën die dit gebied anders gebruiken, zoals cartridges of REU-emulatie, worden dan verborgen. Deze functie kan weer worden uitgeschakeld door $82 naar het SD-commandoregister te schrijven. Net als bij de normale geheugentoewijzing van de sectorbuffer op $FFD6xxx, bepaalt het BUFSEL-signaal (bit 7 van $D689) voor software die niet in hypervisormodus draait of de FDC-sectorbuffer of de SD-kaart-sectorbuffer zichtbaar is. Let op: wanneer u de Matrix Mode / serial-monitor-interface gebruikt om de inhoud van de sectorbuffer te bekijken, gebeurt dat in de hypervisorcontext. Het BUFSEL-signaal wordt dan genegeerd en de volledige 4KB-buffer is zichtbaar.

De in het geheugen ingemapte sectorbuffer heeft als voordeel dat hij via DMA kan worden benaderd, waardoor zeer efficiënte kopieën mogelijk zijn. Bovendien kan een sector op de achtergrond worden geladen terwijl uw programma zich in de tussentijd met interessantere zaken bezighoudt.

Sectoren van een schijf lezen

Om een sector van een schijf te kunnen lezen, moet u een aantal stappen doorlopen. Volgt u deze aanwijzingen op, dan werkt uw code zowel met fysieke schijven als met D81-schijfimages die zich op de SD-kaart bevinden:

- Schakel eerst de motor in en selecteer het juiste station. De F011 ondersteunt maximaal acht fysieke stations, hoewel het zelden voorkomt dat er meer dan twee fysiek zijn aangesloten. Om de motor in te schakelen schrijft u $60 naar $D080. Schrijf daarna een SPINUP-commando ($20) naar $D081 en wacht totdat de BUSY-vlag (bit 7 van $D082) wordt gewist. Het station draait dan op snelheid en is klaar om aanvragen af te handelen.
- Selecteer vervolgens de juiste zijde van de schijf door de SIDE1-vlag (bit 3 van $D080) te zetten of te wissen. Dit heeft onmiddellijk effect.
- Gebruik daarna, afhankelijk van de behoefte, de step-in- en step-out-commando's (door $10 en $18 naar $D081 te schrijven) om de leeskop naar het juiste spoor te verplaatsen. Ook na elk commando hoort u te wachten totdat de BUSY-vlag (bit 7 van $D082) wordt gewist voordat u het volgende commando geeft.

  U kunt controleren of de leeskop zich op spoor 0 bevindt door de TRACK0-vlag te controleren, maar er is geen onfeilbare manier om te weten of u zich op enig ander specifiek spoor bevindt. U kunt de registers op $D6A3 - $D6A5 gebruiken om de spoor-, sector- en zijdewaarde van de laatste sectorheader te zien die onder de leeskop is gepasseerd en zo een geïnformeerde schatting te maken van het momenteel geselecteerde spoor. Dit werkt echter alleen voor echte schijven, omdat schijfimages niet onder de leeskop ronddraaien. Houd er ook rekening mee dat sporen sectoren kunnen bevatten waarvan de spoorwaarden in de sectorheaders opzettelijk of per ongeluk onjuist zijn.
- Laad als vierde stap het gewenste spoor-, sector- en zijdenummer in de TRACK-, SECTOR- en SIDE-registers ($D084, $D085 en $D086). De FDC is nu gereed om een sector te lezen.
- Ten vijfde moet u een passende leesopdrachtwaarde in $D081 schrijven. Normaal is dit $40 (64). Daarna wacht u tot het RDREQ-signaal ($D083, bit 7) hoog wordt, wat aangeeft dat de sector gevonden is. Vervolgens wacht u ofwel op elke keer dat DRQ hoog wordt en leest u in zo'n lus byte voor byte, ofwel wacht u tot de BUSY-vlag gewist is en de DRQ- en EQ-vlaggen hoog worden, wat aangeeft dat de volledige sector in de buffer is gelezen.

Automatische spoorcorrectie

De 45IO27 bevat tevens een "auto-tune"-functie voor sporen, die wordt ingeschakeld door bit 7 van $D696 te wissen. Die functie leest de sectorheaders om te bepalen waar de kop zich momenteel bevindt, en stapt de kop in of uit om zo het juiste spoor te bereiken. Auto-tune is standaard ingeschakeld.

Sector-scheefstand en de Target Any-modus

Het is ook het vermelden waard dat het TARGANY-signaal geactiveerd kan worden om de schijfstationcontroller te laten weten dat hij eenvoudig de volgende sector moet lezen die onder de kop passeert. Dit geldt alleen bij het gebruik van echte diskettes, waar het aanzienlijke voordeel biedt dat u de sectoren kunt lezen in de volgorde waarin ze zich op de schijf bevinden. Zo kunt u in één keer een volledig spoor lezen, zonder te hoeven wachten tot het indexgat voorbij is, of te hoeven weten welke sector als volgende onder de kop zal passeren.

De C65-DOS formatteert bijvoorbeeld diskettes met een scheefstandsfactor van 7, terwijl PC's een andere scheefstandsfactor kunnen gebruiken. Als u de scheefstandsfactor van de diskette niet kent, kunt u het lezen van de sectoren op het spoor in een suboptimale volgorde plannen. Dit kan leiden tot overdrachtsnelheden van slechts 5 sectoren per seconde, vergeleken met het optimale geval van 50 sectoren per seconde. Met de juiste sectorvolgorde, of door de target any-modus te gebruiken, is het dus mogelijk om ongeveer twee volledige sporen per seconde te lezen, d.w.z. twee zijden $\times$ twee sporen, oftewel ongeveer 20 KB/seconde op DD-diskettes, of het dubbele daarvan op HD-diskettes, rond 40 KB/seconde. Dit steekt zeer gunstig af tegen de laadsnelheid van de C65-DOS, die in de C64-modus doorgaans dichter bij 1 KB/sec ligt.

Schijfindeling en logische 1581-sectoren

Het 1581-schijfformaat is ongebruikelijk in die zin dat de fysieke sectoren op de schijf een andere grootte hebben dan de datablokken die het aan de gebruiker presenteert. Concreet gebruiken de diskettes sectoren van 512 bytes, terwijl de 1581- (en C65-) DOS datablokken van 256 bytes presenteren. In elke fysieke sector worden twee blokken opgeslagen. De fysieke spoornummers lopen ook van 0 tot 79, terwijl de logische spoornummers van de DOS van 1 tot 80 lopen. Fysieke sectoren zijn ook genummerd van 1 tot 10, terwijl logische bloknummers lopen van 0 tot 39.

Dit betekent dat u, als u een logische 1581-sector wilt vinden, moet weten in welke fysieke sector deze te vinden is. Om de fysieke sector te bepalen die een blok bevat, trekt u eerst één af van het spoornummer en deelt u vervolgens het sectornummer door twee. Logische sectoren 0 tot 19 van elk spoor bevinden zich in fysieke sectoren 1 tot 10 op de eerste zijde van de schijf. Logische sectoren 20 tot 39 bevinden zich in fysieke sectoren 1 tot 10 op de achterzijde van de schijf.

We kunnen dus een logisch spoor en sector t,s als volgt afbeelden op het fysieke spoor, de zijde en de sector:

$track = t - 1$

$sector = (s/2) + 1, IFFs < 20, ELSE = ((s - 20)/2) + 1$

$side = 0IFFsector < 20$

Het is ook het vermelden waard dat de 45IO27 van sporen voorbij spoor 80 kan lezen, mits het schijfstation daartoe in staat is. Bijna alle 3.5-inchschijfstations kunnen

ten minste één extra spoor lezen, omdat fabrikanten van diskettes historisch gezien informatie over de schijf opsloegen op het 81ste spoor. Naar onze ervaring kunnen bijna alle stations ook een 82ste spoor benaderen.

FD2000-diskettes

De CMD™ FD2000™-schijfstations met hoge dichtheid van 3.5" voor Commodore™-computers gebruiken een ongebruikelijke schijfindeling die heel anders is dan die van PC's: Ze gebruiken 10 sectoren, net zoals op 720KB-diskettes met dubbele dichtheid, maar verdubbelen de *sectorgrootte* van 512 bytes naar 1024 bytes. De 45IO27 ondersteunt deze grotere sectoren momenteel niet voor gebufferde sectorbewerkingen. Ten minste ondersteuning voor alleen-lezen is gepland; deze zal in de toekomst via een core-update worden toegevoegd. FD2000™- en FD4000™-diskettes kunnen echter wel gelezen en geschreven worden met behulp van de track-DMA-functie.

Voor het aanmaken van nieuwe diskettes ondersteunt de 45IO27 *bovendien* al schijven en stations met hoge dichtheid, met veel hogere capaciteiten dan de FD2000 kon bieden.

HD-diskettes en diskettes met variabele dichtheid

De 45IO27 ondersteunt variabele datatempo's, waardoor het mogelijk is HD-stations en -media te gebruiken, met een flexibele benadering van schijfformaten voor gebruikersexperimenten en een gemakkelijke omgang met formaten met hoge capaciteit voor softwaredistributie.

U wordt eigenlijk alleen beperkt door uw verbeeldingskracht, de beschikbare tijd en het beperkte aantal mensen dat nog geïnteresseerd is in het in hun computer plaatsen van een diskette!

Het standaardformaat voor hoge dichtheid (HD) is "1.44MB", met 18 sectoren per spoor over 80 sporen. Dit levert 80 sporen × 18 sectoren × 2 zijden = 2880 sectoren op. Aangezien elke sector 512 bytes groot is, komt dit overeen met 1440KB. Dit voert ons naar het interessante wonderland van "floppy-disk marketing megabytes", een verschijnsel dat lang voorafging aan fabrikanten van SD-kaarten en harde schijven die megabytes van 1000000 bytes gebruiken.

Vreemd genoeg werd voor diskettes de "megabyte" van 1024000 bytes gebruikt, d.w.z. "1MB" = 1KB × 1KB, een vreemde hybride van binaire en decimale conventies. Misschien kwam het doordat de vorige standaard 720KB was en men dacht dat mensen het vreemd zouden vinden als het dubbele van 720KB 1.41MB was en zouden klagen over de ontbrekende kilo-bytes. We blijven voor de consistentie met deze historische ongerijmdheid de floppy-disk-marketing-mega-byte van 1024KB = 1000KB gebruiken.

HD-diskettes zijn echter fundamenteel in staat om veel meer dan 1.44MB te bevatten. De FD2000 sloeg bijvoorbeeld 1.6MB op door dubbelgrote sectoren te gebruiken om het equivalent van 20 sectoren per spoor eruit te halen, en de Amiga ging verder door track-at-once-schrijven te gebruiken om er 22 sectoren per spoor in te passen. Beide

formaten gebruikten over alle sporen een constant datatempo en dus een constant aantal sectoren per spoor.

De omtrek van de sporen op een 3.5"-diskette varieert echter behoorlijk: Het binnenste spoor op een straal van ongeveer 25 mm heeft een lengte van 157 mm, terwijl het buitenste spoor op een straal van 40 mm een lengte van 251 mm heeft, 1.6 maal zo lang. Het 1.44MB-formaat is zo ontworpen dat de gegevens betrouwbaar worden opgeslagen op die kortere binnenste sporen. Dat betekent dat we 60% meer gegevens op het buitenste spoor zouden moeten kunnen plaatsen dan op het binnenste spoor, binnen een aantal voorwaarden van de wetten van de fysica, het ontwerp van de schijfstation-elektronica, de kwaliteit van het gebruikte medium en diverse andere hinderlijke zaken. Vanwege deze variabiliteit en onzekerheid ondersteunt de schijfstationcontroller van de MEGA65 een volledig variabel datatempo per spoor afzonderlijk.

Track Information Blocks

Om variabele datatempo's te ondersteunen gebruikt de 45IO27 de zogeheten *Track Information Blocks* (TIB's) die informatie bevatten over het datatempo en de codering die op het spoor gebruikt worden. Zo kunnen gebruikers experimenteren met verschillende dichtheden op verschillende sporen en tóch functioneren de diskettes automatisch voor gebufferde sectorbewerkingen.

Het Track Information Block wordt automatisch aangemaakt bij gebruik van de functie voor het automatisch formatteren van sporen, maar moet handmatig worden aangemaakt bij gebruik van ongebufferde opmaak. De TIB zelf bestaat uit de volgende gegevens:

1. 3× $A1-syncbytes (geschreven met het klokbyte $FB)
2. het $65-marker van het MEGA65 Track Information Block (geschreven met het klokbyte $FF, evenals alle volgende bytes in het blok)
3. het spoornummer
4. de datatempo-deler, in hetzelfde formaat als $D6A2, d.w.z. datatempo = 40.5 MHz / waarde.
5. de track-coderingsinformatie: bit 7 = Track-at-once-vlag, 1 = geen gaps tussen sectoren (Amiga-stijl), 0 = met gaps tussen sectoren (normaal), bit 6 = de gegevenscodering, 0 = MFM, 1 = RLL2,7. De overige bits zijn gereserveerd en moeten bij het schrijven 0 zijn.
6. het aantal sectoren, d.w.z. het aantal sectoren op de track.
7. CRC-byte 1, volgens het normale CRC-algoritme voor diskettes.
8. CRC-byte 2, volgens het normale CRC-algoritme voor diskettes.

Het Track Information Block wordt altijd geschreven met het datatempo voor een 720KB-diskette met dubbele dichtheid, zodat het op elke diskette aanwezig kan zijn. Het schrijven van het Track Information Block en van de start-of-track-gaps met het DD-datatempo zorgt er ook voor dat de kop bij zeer hoge datatempo's nog voldoende

tijd heeft om naar de schrijfmodus over te schakelen, waardoor een van de vele problemen wordt vermeden die ontstaan bij het schrijven van gegevens met zeer hoge datatempo's.

Indien u diskettes ongebufferd formatteert, is het de verantwoordelijkheid van de programmeur om het datatempo om te schakelen nadat het Track Information Block is geschreven en enkele bytes meer zijn geschreven, zodat de floppy-coderingspijplijn de laatste byte van het Track Information Block kan doorspoelen. Dit wordt allemaal automatisch afgehandeld wanneer u de automatische trackformatteringsfunctie gebruikt.

De opname van het TIB stelt gebruikers in staat om op verschillende stations en media te spelen en de mogelijkheden van verschillende datatempo's te verkennen, terwijl alles in alle MEGA65's toch automatisch leesbaar blijft, omdat het TIB de controller in staat stelt om naar het juiste datatempo en de juiste codering over te schakelen. Het is waarschijnlijk dat na verloop van tijd enigszins gestandaardiseerde formaten zullen ontstaan, zeer waarschijnlijk in het bereik van 2 MB tot 3.5 MB - en daarmee de capaciteit van ED-media in ED-stations benaderen, zonder die stations of media nodig te hebben.

Schijven formatteren

Het formatteren van schijven is nu mogelijk met de 45IO27, ongebufferd of volledig automatisch. Om een track te formatteren geeft u een van de volgende commando's aan $D081:

- $A0 - automatisch formatteren, met gaps tussen sectoren en schrijfvoorcompensatie uitgeschakeld.
- $A1 - handmatig formatteren, schrijfvoorcompensatie uitgeschakeld.
- $A4 - automatisch formatteren, met gaps tussen sectoren en schrijfvoorcompensatie ingeschakeld.
- $A5 - handmatig formatteren, schrijfvoorcompensatie ingeschakeld.
- $A8 - automatisch formatteren, Amiga-stijl track-at-once en schrijfvoorcompensatie uitgeschakeld.
- $AC - automatisch formatteren, Amiga-stijl track-at-once en schrijfvoorcompensatie ingeschakeld.

Handmatig formatteren wordt niet aanbevolen, tenzij u track-at-once-geformatteerde diskettes voor softwaredistributie aanmaakt, vanwege de relatieve complexiteit daarvan. Ook moeten bij de hogere datatempo's bytes zo vaak als elke 20 cycli aan de floppy-controller worden geleverd, wat aanzienlijke zorg vereist bij het schrijven van de formatteringsroutine. Zie voor meer informatie over het handmatig formatteren van tracks, evenals voor voorbeelden van handmatig formatteren, de C64 Specifications Manual of de C65 ROM-DOS-broncode.

De automatische modi daarentegen formatteren een track met één enkel commando en zijn daardoor veel eenvoudiger te gebruiken, en worden voor algemeen gebruik aanbevolen. Schrijfvoorcompensatie moet normaliter worden ingeschakeld, omdat die bij hogere datatempo's vereist is en bij lagere datatempo's geen problemen veroorzaakt.

Schrijfvoorcompensatie

Schrijfvoorcompensatie is een familie van algoritmen die wordt gebruikt bij het schrijven van signalen met een hoog datatempo naar diskettes. Zij dient om de voorspelbare component van de tijdsvariatie van magnetische opnamen te anticiperen en teniet te doen. Er zijn diverse bronnen van deze tijdsvariatie, die het onderwerp zijn geweest van promotieonderzoeken en van veel bedrijfseigen onderzoek door fabrikanten van harde schijven. Wat voor ons belangrijk is om te begrijpen, is dat aangrenzende pulsen (in feite magnetische ompolingen) naar elkaar toe worden gedrukt wanneer ze door langere pulsen worden omgeven, en de neiging hebben uit elkaar te spreiden wanneer ze door kortere pulsen worden omgeven.

Er zijn ook andere fascinerend complexe en moeilijk te voorspellen factoren, die zaken veroorzaken zoals de "negatieve verschuiving van midlange pulsen", de "inverse F-verdeling van pulsaankomsttijden" en wie weet wat nog meer. Die laten we echter over aan de fabrikanten van harde schijven. Wij beperken ons tot het door het gegevenspatroon veroorzaakte effect dat in de vorige alinea is beschreven.

De 45IO27 ondersteunt twee instelbare coëfficiënten voor kleine en grote correcties hierop, die met behulp van een interne opzoek (look-up) tabel worden toegepast. Dit wordt echter allemaal automatisch afgehandeld wanneer u schrijfvoorcompensatie inschakelt. Dit maakt datatempo's mogelijk die veel dichter bij de verwachte limiet van HD-media liggen, hoewel de werkelijke limiet vanwege de andere verschrikkingen van magnetische media-opnamen die hierboven zijn aangestipt, niet wordt bereikt.

Gebufferd sectorschrijven

De 45IO27 kan naar schijfimages schrijven die zich op de SD-kaart bevinden, of wanneer u gevirtualiseerde schijftoegang gebruikt.

Om een sector te schrijven doorloopt u een soortgelijk proces als bij het lezen, behalve dat u $84 in het commandobyt schrijft in plaats van $40. De $80 geeft een schrijfbewerking aan en de $04 activeert de schrijfvoorcompensatie. Dit is belangrijk bij het schrijven naar echte diskettes, vooral HD- en ED-diskettes. Schrijfvoorcompensatie zorgt ervoor dat bits iets te vroeg of iets te laat worden geschreven, met behulp van een algoritme dat modelleert hoe de magnetische domeinen op een schijf de neiging hebben te bewegen nadat ze zijn geschreven.

Indien u de sectorbuffer niet wilt gebruiken, maar tijdens de schrijfbewerking elk byte afzonderlijk tegelijk wilt leveren, moet u $01 aan de commandocode toevoegen. Dit wordt op de MEGA65 echter niet aanbevolen, omdat bij het schrijven naar de SD-kaart of bij het gebruik van gevirtualiseerde schijfimages de volledige sectorbewerking vanuit het perspectief van uw programma ogenblikkelijk kan plaatsvinden. Dat betekent

dat het in deze modus niet mogelijk is om gegevens betrouwbaar te leveren. Behalve dat het minder handig is, is het dus ook minder betrouwbaar.

Zodra een schrijfbewerking is geactiveerd, geeft het DRQ-signaal aan wanneer u het volgende byte moet leveren indien u byte voor byte schrijft. In overige gevallen wordt aangenomen dat u de sectorbuffer vooraf hebt gevuld met de volledige benodigde 512 bytes aan gegevens.

Om naar schijven te schrijven die Track Information Blocks bevatten, moet u eerst wachten tot het TIB is gelezen bij het wisselen van tracks. Dit doet u door te wachten tot $D6A9 (aantal sectoren per track uit het TIB) een waarde ongelijk aan nul bevat.

Floppy-track-DMA

Zoals eerder beschreven, verbindt de 45IO27 zich met de 45GS02-CPU, en in het bijzonder met de interne DMA-controller daarvan, om een eenvoudig mechanisme te bieden voor het lezen en schrijven van de ruwe magnetische fluxovergangen van diskettes. Naast het mogelijk maken van software die elk mogelijk schijfformaat kan lezen, maakt het ook het schrijven en dupliceren van vrijwel elk schijfformaat mogelijk. Wij zeggen 'vrijwel', omdat het mogelijk is dat diskettes worden geschreven met speciale machines die mogelijkheden hebben die die van het floppy-stationmechanisme van de MEGA65 te boven gaan. Fat tracks is hiervan een voorbeeld. Een ander voorbeeld is dat de magnetische fluxovergangen zodanig worden geplaatst en/of gespreid dat een normaal floppy-station ze niet precies kan reproduceren.

Floppy-track-DMA gebruiken

De volgende DMA-opties vertellen de DMAgic-controller om in plaats van een geheugenlocatie ruwe fluxwaarden te lezen of schrijven. In deze modus wacht de DMAgic-controller op het ontvangen van elk opeenvolgend pulsinterval van de 45IO27. De gegevens die hij aldus leest (of schrijft) zijn de duur tussen elke opeenvolgende fluxompoling, in eenheden van 50ns.

- $0D - Schrijf ruwe fluxintervallen. Stel de DMA in op COPY-modus om dit te gebruiken.
- $0E - Lees ruwe fluxintervallen en negeer onwaarschijnlijk lange intervallen. Stel de DMA in op FILL-modus om dit te gebruiken.
- $0F - Lees ruwe fluxintervallen en retourneer onwaarschijnlijk lange intervallen als $FF. Stel de DMA in op FILL-modus om dit te gebruiken.

Zie floppytest.c in de mega65-tools-repository van de MEGA65-githubsite op https://github.com/mega65-toolsvoor voorbeeldcode die sporen leest met behulp van deze functionaliteit.

De beperkingen van schijfstations begrijpen

Bij het schrijven van ruwe fluxovergangen moet u rekening houden met de beperkingen van standaard schijfstations. Er zijn drie belangrijke factoren waarmee rekening moet worden gehouden:

- Twee overgangen kunnen niet te dicht bij elkaar worden geplaatst. Het station weigert ze namelijk eenvoudigweg te schrijven als ze minder dan een zekere afstand uit elkaar liggen. Zelfs wanneer u ze toch weet te schrijven, voegt het leesfiltercircuit van het schijfstation pulsen die te dicht bij elkaar staan samen.
- Twee overgangen kunnen niet te ver uit elkaar worden geplaatst. In dat geval schrijft het station ze zonder bezwaar, maar het leesfiltercircuit raakt ongerust als het een tijdje geen puls ziet, begint te denken dat de achtergrondruis echte signalen zijn en introduceert valse pulsen in de leesstroom die in werkelijkheid niet op de schijf staan.
- Magnetische pulsen verplaatsen zich op de schijf wanneer u ze schrijft! De natuurkunde hierachter is erg complex en hangt af van de reeks afstanden tussen opeenvolgende pulsen, van de gegevens die eerder op het spoor zijn geschreven en, zo voelt het, ook van de maanstand en de kleur van uw ondergoed. Er zijn enkele vuistregels die bij "typische"-pulsafstandsintervallen kunnen worden gebruikt om dit gedeeltelijk te beperken. Deze regels heten schrijfvoorcompensatie: de pulsen worden vóór het schrijven verplaatst, zodat ze na het schrijven op de juiste plaats terechtkomen. De 45IO27 past al schrijfvoorcompensatie toe voor HD-geformatteerde schijven. Het effect van het verplaatsen van pulsen is vooral uitgesproken bij pulsen die dichter bij elkaar liggen.
- Hoe hoger genummerd een spoor is, des te korter het is. Dat betekent dat de pulsen in de tijd verder uit elkaar moeten liggen om op een spoor op dezelfde afstand van elkaar te staan. Daarom past de 1541 minder gegevens op hoger genummerde sporen. Dit is vooral uitgesproken bij hogere datatempo's, waar de "magnetische resolutie" van de schijven een probleem wordt. Om het nog erger te maken is de sterkte van de magnetische signalen evenredig aan de snelheid waarmee het spoor passeert. Dat betekent dat bij hoger genummerde sporen, die dichter bij het midden van de schijf liggen, niet alleen de "magnetische resolutie" lager is door de kortere sporen, maar ook de lineaire snelheid van de schijf onder de kop lager is, wat resulteert in zwakkere signalen.

Het is de combinatie van deze factoren die doorgaans beperkt hoe dicht u gegevens op een diskette kunt pakken – samen uiteraard met de kwaliteit en toestand van de diskette en het schijfstation. De HD+-schijfformaten van de MEGA65, die rond de 3MiB op een standaard diskette van 1.44MiB kunnen passen, besteden bijzondere aandacht aan het beheersen van deze factoren. Daarom gebruiken die formaten veel verschillende dichtheidszones en gebruiken ze bovendien RLL27-codering in plaats van MFM-codering, omdat die de minimale afstand tussen pulsen vergroot.

F011-registers van de schijfstationcontroller

De volgende registers zijn de set F011-compatibiliteitsregisters van de 45IO27. Registers die verband houden met het gebruik van opslag op basis van SD-kaarten vindt u in de overeenkomstige sectie hieronder.

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D080	53376	IRQ	LED	MOTOR	SWAP	SIDE	DS		
D081	53377	WRCMD	RDCMD	FREE	STEP	DIR	ALGO	ALT	NOBUF
D082	53378	BUSY	DRQ	EQ	RNF	CRC	LOST	PROT	TK0
D083	53379	RDREQ	WTREQ	RUN	WGATE	DISKIN	INDEX	IRQ	DSKCHG
D084	53380	TRACK							
D085	53381	SECTOR							
D086	53382	SIDE							
D087	53383	DATA							
D088	53384	CLOCK							
D089	53385	STEP							
D08A	53386	PCODE							

- **ALGO** Selecteert het lees- en schrijfalgoritme (wordt momenteel genegeerd).
- **ALT** Selecteert de alternatieve DPLL-leesherstelmethode (niet geïmplementeerd)
- **BUSY** F011 FDC-busyvlag (commando wordt uitgevoerd) (alleen lezen)
- **CLOCK** Stelt het klokpatroon in dat wordt gebruikt bij het schrijven van adres- en datamerken. Normaal $FF laten staan
- **COMMAND** F011 FDC-commandoregister
- **CRC** F011 FDC CRC-controlefoutvlag (alleen lezen)
- **DATA** F011 FDC-gegevensregister (lezen/schrijven) voor toegang tot de 512 byte sectorbuffer van de schijfstationcontroller
- **DIR** Stelt de staprichting in (naar binnen ten opzichte van
- **DISKIN** F011 Schijfgevoeligheid (alleen lezen)
- **DRQ** F011 FDC DRQ-vlag (een of meer bytes gegevens zijn gereed) (alleen lezen)
- **DS** Stationselectie (0 tot 7). Intern station is 0. Tweede schijfstation op de interne kabel is 1. Andere waarden zijn gereserveerd voor de C1565 externe stationinterface.
- **DSKCHG** F011 Schijfwisselgevoeligheid (alleen lezen)
- **EQ** F011 FDC CPU- en diskschijfwaarden naar de sectorbuffer zijn gelijk, wat aangeeft dat de sectorbuffer vol of leeg is. (alleen lezen)

- **FREE** Commando is een vrij formaat (laag niveau) bewerking
- **INDEX** F011 Indexgatgevoeligheid (alleen lezen)
- **IRQ** De schijfstationcontroller heeft een interrupt gegenereerd (alleen lezen). Opmerking: interrupts worden momenteel niet geïmplementeerd op de 45IO27.
- **LED** Station-LED knippert wanneer ingesteld
- **LOST** F011 LOST-vlag (gegevens zijn verloren gegaan tijdens de overdracht, d.w.z. de CPU las de gegevens niet snel genoeg) (alleen lezen)
- **MOTOR** Schakelt de stationmotor en LED in (tenzij het LED-signaal ook is ingesteld, waardoor de station-LED knippert)
- **NOBUF** Herstel de lees-/schrijfpunten van de sectorbuffer
- **PCODE** (Alleen lezen) geeft de beschermingscode van het meest recent gelezen sector terug. Was bedoeld voor elementaire kopieerbescherming. Niet geïmplementeerd.
- **PROT** F011 Schrijfbeveiligingsvlag van de schijf (alleen lezen)
- **RDCMD** Commando is een leesbewerking wanneer ingesteld
- **RDREQ** F011 Leesverzoekvlag, d.w.z. het gevraagde sector werd gevonden tijdens een leesbewerking (alleen lezen)
- **RNF** F011 FDC-verzoek niet gevonden (RNF), d.w.z. een lees- of schrijfbewerking heeft het gevraagde sector niet gevonden (alleen lezen)
- **RUN** F011 Opeenvolgende overeenkomst. Een synoniem van RDREQ op de 45IO27 (alleen lezen)
- **SECTOR** F011 FDC-sectorselectieregister
- **SIDE** Stuurt het SIDE-signaal naar de schijfstationcontroller direct, d.w.z. selecteert welke zijde van het medium actief is.
- **STEP** Het schrijven van 1 laat de kop in de aangegeven richting stappen
- **SWAP** Wissel de boven- en onderste helft van de gegevensbuffer (d.w.z. inverteer bit 8 van de sectorbuffer)
- **TK0** F011 Kop bevindt zich boven spoor 0-vlag (alleen lezen)
- **TRACK** F011 FDC-spoorselectieregister
- **WGATE** F011 Schrijfpoortvlag. Geeft aan dat het station momenteel naar het medium schrijft. Er kunnen problemen optreden als een schrijfbewerking wordt afgebroken (alleen lezen)
- **WRCMD** Commando is een schrijfbewerking wanneer ingesteld
- **WTREQ** F011 Schrijfverzoekvlag, d.w.z. het gevraagde sector werd gevonden tijdens een schrijfbewerking (alleen lezen)

De volgende registers zijn uitsluitend van toepassing op de 45IO27, d.w.z. ze zijn specifiek voor de MEGA65:

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D6A0	54944	DENSITY	DBGMO-TORA	DBGMO-TORA	DBGDIR	DBGDIR	DBGW-DATA	DBGW-GATE	DBGW-GATE
D6A2	54946	DATARATE							

- **DATARATE** Stelt het aantal buscycli per diskettemagnetisch interval in (verlaag om het datatempo te verhogen)
- **DBGDIR** Stuurt de STEPDIR-lijn van de schijfstationcontroller
- **DBGMOTORA** Stuurt de MOTOR-lijn van het schijfstation aan
- **DBGWDATA** Stuurt de WDATA-lijn van het schijfstation aan
- **DBGWGATE** Stuurt de WGATE-lijn van het schijfstation aan
- **DENSITY** Stuurt de selectielijn van de dichtheid van het schijfstation aan

SD-KAARTCONTROLLER EN F011-VIRTUALISATIEFUNCTIES

Voor situaties waarin u geen echte diskettes wilt gebruiken, ondersteunt de 45IO27 twee complementaire alternatieve modi:

- Toegang tot schijfimages op basis van de SD-kaart.
- Gevirtualiseerde toegang tot schijfimages.

Dit komt bovenop de directe toegang tot een dual-bus-SD-kaartinterface.

Toegang tot schijfimages op basis van de SD-kaart

De 45IO27 is tegelijk een schijfstation en een SD-kaartcontroller. Hierdoor kan het apparaat transparant toegang bieden tot D81-schijfimages die op de SD-kaart zijn opgeslagen. Verder is het, omdat de controller is gecombineerd, mogelijk om het schijfstation nog steeds te laten stappen en draaien alsof het wordt gebruikt, wat voor aanzienlijke sfeer en een gevoel van realisme zorgt, zelfs wanneer u schijfimages gebruikt.

De 45IO27 ondersteunt zowel standaard D81-schijfimages van 800KB als "MEGA Images" van 64MB. Hoewel een besturingssysteem beperkingen kan opleggen op basis van de naam van een bestand, is de 45IO27 blind voor deze eisen. In plaats daarvan vereist het apparaat enkel dat een aaneengesloten blok van 800KB of 64MB van de SD-kaart wordt gebruikt om een schijfimage te bevatten.

Wanneer een schijfimage wordt ingeschakeld, worden de overeenkomstige sectoren op de SD-kaart feitelijk onder de controle van de gebruiker geplaatst en kan het besturingssysteem het lezen of schrijven van die sectoren niet langer verhinderen. U moet daarom nooit toegang inschakelen tot een image dat korter is dan de vereiste grootte, omdat het de gebruiker anders in staat stelt onbedoeld of kwaadwillig gegevens te openen en/of te wijzigen die geen deel uitmaken van het imagebestand.

Om dezelfde reden kan alleen de hypervisor het sectornummer wijzigen waar een schijfimage begint (de D?STARTSEC?-signalen), of het gebruik van schijfimages in plaats van het echte schijfstation toestaan (de USEREAL0- en USEREAL1-signalen). Zodra de hypervisor de beginsector van een schijfimage heeft ingesteld en het USEREAL0- of USEREAL1-signaal heeft gewist, kan de gebruiker nog steeds bepalen of een toegang naar het echte schijfstation of naar het schijfimage gaat, door het betreffende signaal respectievelijk te wissen of te zetten. Voor station 0 is dit D0IMG, en voor station 1 is het D1IMG.

Er zijn ook signalen die bepalen of een schijfimage een D81-image van 800KB of een MEGA Disk-image van 64MB is, of een schijfimage aanwezig is en of het schrijfbeveiligd is. Deze bevinden zich allemaal in het $D68B-register. Omdat manipulatie van deze registers gegevens kan beschadigen of er op ongeoorloofde wijze toegang toe kan geven, zijn deze signalen allemaal alleen-lezen, behalve vanuit de hypervisor.

De volgende tabel geeft de registers weer die worden gebruikt om de toegang tot de schijfimages op de SD-kaart te regelen:

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D68A	54922	D1D64	D0D64	-					
D68B	54923	D1MD	D0MD	D1WP	D1P	D1IMG	D0WP	D0P	D0IMG
D68C	54924	D0STARTSEC0							
D68D	54925	D0STARTSEC1							
D68E	54926	D0STARTSEC2							
D68F	54927	D0STARTSEC3							
D690	54928	D1STARTSEC0							
D691	54929	D1STARTSEC1							
D692	54930	D1STARTSEC2							
D693	54931	D1STARTSEC3							
D6A1	54945	-				SILENT	USE-REAL1	TARGANY	USE-REAL0

- **D0D64** Het schijfimage van F011-station 0 is D64 indien ingesteld, anders D81 (zie ook SDFDC:D0MD)
- **D0IMG** F011-station 0 gebruikt een schijfimage indien ingesteld, anders een echte diskette.
- **D0MD** Het schijfimage van F011-station 0 is D65 indien ingesteld, anders D81 (zie ook SDFDC:D0D64)
- **D0P** F011-station 0 medium aanwezig

- **D0STARTSEC0** Adres van het schijfimage van F011-station 0 op de SD-kaart (LSB)
- **D0STARTSEC1** F011-station 0 schijfimageadres op SD-kaart (2e byte)
- **D0STARTSEC2** F011-station 0 schijfimageadres op SD-kaart (3e byte)
- **D0STARTSEC3** F011-station 0 schijfimageadres op SD-kaart (MSB)
- **D0WP** Schrijven inschakelen voor F011-station 0
- **D1D64** F011-station 1 schijfimage is D64 indien gezet, anders D81 (zie ook SDFDC:D1MD)
- **D1IMG** F011-station 1 gebruikt schijfimage indien gezet, anders een echt schijfstation.
- **D1MD** F011-station 1 schijfimage is D65 indien gezet, anders D81 (zie ook SDFDC:D1D64)
- **D1P** F011-station 1 medium aanwezig
- **D1STARTSEC0** F011-station 1 schijfimageadres op SD-kaart (LSB)
- **D1STARTSEC1** F011-station 1 schijfimageadres op SD-kaart (2e byte)
- **D1STARTSEC2** F011-station 1 schijfimageadres op SD-kaart (3e byte)
- **D1STARTSEC3** F011-station 1 schijfimageadres op SD-kaart (MSB)
- **D1WP** Schrijven inschakelen voor F011-station 1
- **SILENT** Schakelt het draaien en tracken van de diskette uit voor SD-kaartbewerkingen.
- **TARGANY** Leest het volgende sector onder de kop indien gezet, waarbij de gevraagde zijde, track en sectornummer worden genegeerd.
- **USEREAL0** Gebruikt een echt schijfstation voor station 0 indien gezet (alleen-lezen, behalve vanuit de hypervisor)
- **USEREAL1** Gebruikt een echt schijfstation voor station 1 indien gezet (alleen-lezen, behalve vanuit de hypervisor)

F011-virtualisatie

Naast het toestaan van automatische lees- en schrijftoegang tot D81-images op basis van de SD-kaart, is het mogelijk een programma op de seriële monitorinterface aan te sluiten dat gegevens verschaft en aanvaardt alsof het de diskette was.

Dit wordt gewoonlijk gebruikt in een cross-ontwikkelomgeving, waarin u een schijfimage vaak wilt wijzigen dat door een programma wordt gebruikt dat u ontwikkelt - zonder dat u telkens nieuwe versies van het schijfimage eerst op de SD-kaart van de MEGA65 hoeft te zetten. Het heeft bovendien het bijkomende voordeel dat u

gemakkelijk kunt visualiseren welke sectoren worden gelezen en geschreven, wat de ontwikkeling en het opsporen van fouten in uw programma kan versnellen.

Deze functie werkt samen met de hypervisor van de MEGA65 door hyperrupts te activeren (dat zijn interrupts die de hypervisor activeren). In de hypervisor zit dan speciale code die via de seriële monitorinterface met het `m65`-programma communiceert.

Als dat allemaal tamelijk ingewikkeld klinkt: alles wat u hoeft te weten is dat u deze functie gebruikt door het `m65`-hulpprogramma uit te voeren met argumenten als `-d image.d81`. Hiermee zou de verbinding met de MEGA65 automatisch tot stand moeten komen. Als de BASIC-interpreter niet meer reageert, drukt u op de resetknop (niet op de netschakelaar) aan de linkerkant van uw MEGA65, en zou deze moeten terugkeren naar de `READY.`-prompt van BASIC - en als uw meegeleverde schijfimage een C65-auto-bootfunctie heeft, zou deze automatisch moeten beginnen met opstarten.

Deze functie werkt uitstekend indien de hostcomputer Linux draait en maakt laden mogelijk met een snelheid van ongeveer 60KB per seconde. Op Windows- of Apple OSX-systemen kan de snelheid echter aanzienlijk lager liggen.

Om dit te gebruiken heeft u bovendien een interface-module en/of kabel nodig om uw cross-ontwikkelingssysteem op de seriële monitorinterface van de MEGA65 aan te sluiten. Dit doet u het eenvoudigst met een Trenz TE0790-03 JTAG-adapter en een mini-USB-kabel.

Meer informatie over het gebruik van deze interface en het `m65`-hulpprogramma vindt u in *het grote Mega65-boek*, Gegevensoverdracht en foutopsporing (hoofdstuk 18).

Dual-bus SD-kaartcontroller

De 45IO27 bevat een snelle dual-bus SD-kaartcontroller. Deze controller werkt in de SPI x1-modus met een kloksnelheid van 20MHz en biedt een maximale doorvoer van ongeveer 2MB per seconde. De kwaliteit van de SD-kaart is bepalend voor de prestaties: sommige kaarten leveren doorgaans 1.7MB per seconde, terwijl andere 1MB per seconde of minder halen. Over het algemeen presteren nieuwere kaarten die geschikt zijn voor video-opnamen beter. De controller ondersteunt SDHC-kaarten en heeft experimentele ondersteuning voor SDXC-kaarten. Oudere SD-kaarten met een capaciteit van 2GB of minder worden niet ondersteund, omdat deze een andere adresseringsmethode gebruiken.

De SD-controller zelf is eenvoudig aan te sturen: geef het sectornummer op in $D681-$D684 en geef vervolgens een lees- of schrijfcommando aan het commandoregister ($D680). De SD-controller ondersteunt alleen sectorgebaseerde gebufferde bewerkingen via de sectorbuffer. In hypervisormodus bevindt de sectorbuffer zich op $FFD6E00-$FFD6FFF, terwijl de SD-kaart en de schijfstationcontroller in de normale werkingsmodus van de computer één adres delen voor zowel de diskettesectorbuffer als de SD-kaartsectorbuffer. Welke buffer op dat adres zichtbaar is, wordt bepaald door het BUFSEL-signaal. Bij een waarde van 1 is de SD-kaartbuffer zichtbaar, bij 0 de diskettesectorbuffer. Zie ook subsectie 8 op bladzijde 126 voor een nadere beschrijving van het precieze gedrag van deze buffer in de normale modus ten opzichte van de hypervisormodus, en hoe deze ook op $DE00 kan worden gemapt.

Write-gate

Bij het schrijven van een sector moet u eerst de write-gate openen. Dit is een mechanisme dat onbedoelde gegevenscorruptie op de SD-kaart voorkomt: u moet twee verschillende waarden achtereenvolgens naar het commandoregister ($D680) schrijven. Na het openen van de write-gate heeft u ongeveer één milliseconde om het schrijfcommando te geven, waarna de write-gate weer sluit en de SD-kaart wordt beschermd tegen schrijven totdat de write-gate opnieuw wordt geopend. Er zijn twee aparte write-gates: één voor het master boot record (sector 0) en één voor alle overige sectoren, beide vermeld in de commandotabel hieronder. Dit biedt extra bescherming van het zeer belangrijke master boot record tegen programma's die per ongeluk sector 0 als doel van een gewone schrijfoperatie berekenen.

Vulmodus

Wanneer u sectoren met een constante waarde wilt vullen, ondersteunt de 45IO27 hiervoor een speciale modus, zodat u de inhoud van de sectorbuffer niet hoeft te overschrijven. U activeert dit door de gewenste vulwaarde in het FILLVAL-register ($D686) te plaatsen, vervolgens het commando om de vulmodus in te schakelen ($83) te geven, de schrijfbewerkingen uit te voeren en tenslotte het commando om de vulmodus uit te schakelen ($84) te geven.

Selectie tussen meerdere SD-kaarten

De controller ondersteunt twee SD-kaartinterfaces, en het is mogelijk om in beide interfaces tegelijkertijd een kaart te hebben. Elke kaart moet echter apart worden gereset en aangestuurd. Er kan tegelijkertijd slechts één kaart worden aangestuurd. U kunt wel elke kaart eenmaal resetten en vervolgens tussen de kaarten wisselen om individuele bewerkingen uit te voeren.

Om de eerste SD-kaartslot te selecteren, schrijft u $C0 naar het SD-controllercommandoregister ($D680). Om de tweede SD-kaartslot te selecteren, schrijft u in plaats daarvan $C1.

Commandotabel van de SD-controller

De SD-controller ondersteunt de volgende commando's die naar het commandoregister op $D680 geschreven kunnen worden:

Commando	Functie
$00 (0)	Zet de SD-kaart in reset (verouderd. Gebruik in plaats daarvan commando $10)
$01 (1)	Haal de SD-kaart uit reset
$02 (2)	Lees een sector van de SD-kaart
$03 (3)	Schrijf één sector naar de SD-kaart
$04 (4)	Schrijf de eerste sector van een meervoudige schrijfbewerking naar de SD-kaart

vervolgd ...

...vervolgd

Commando	Functie
$05 (5)	Schrijf een volgende sector van een meervoudige schrijfbewerking naar de SD-kaart
$06 (6)	Schrijf de laatste sector van een meervoudige schrijfbewerking naar de SD-kaart
$0C (12)	Vraag om het doorspoelen van de schrijfbuffers van de SD-kaart (experimenteel)
$0E (14)	Trek de SD-handshakelijn laag (alleen voor foutopsporing)
$0F (15)	Trek de SD-handshakelijn hoog (alleen voor foutopsporing)
$10 (16)	Zet de SD-kaart in reset met ingestelde vlaggen (voorkeursmethode)
$11 (17)	Haal de SD-kaart uit reset (alternatieve methode)
$40 (64)	Wis de SDHC/SDXC-vlag en selecteer daarmee de verouderde SD-kaartmodus (verouderd)
$41 (65)	Zet de SDHC/SDXC-modusvlag
$44 (68)	Beëindig het geforceerd wissen van de foutvlag van de SD-kaart-toestandsmachine
$45 (69)	Begin het geforceerd wissen van de foutvlag van de SD-kaart-toestandsmachine
$4D (77)	Open de write-gate voor sector 0 (master boot record) gedurende ongeveer 1 milliseconde
$57 (87)	Open de write-gate voor alle sectoren > 0 gedurende ongeveer 1 milliseconde
$81 (129)	Schakel het mappen van de SD/FDC-sectorbuffer op $DE00 - $DFFF in
$82 (130)	Schakel het mappen van de SD/FDC-sectorbuffer op $DE00 - $DFFF uit
$83 (131)	Schakel de vulmodus van de SD-kaart in
$84 (132)	Schakel de vulmodus van de SD-kaart uit
$C0 (192)	Selecteer SD-kaartslot 0
$C1 (193)	Selecteer SD-kaartslot 1

Merk op dat de hypervisor de rechtstreekse toegang tot de SD-controller kan in- of uitschakelen. Het besturingssysteem van de hypervisor kan een mechanisme bieden om toestemming te vragen voor toegang tot de SD-kaartcontroller, bijvoorbeeld voor hulpprogramma's voor schijfbeheer.

De registers van de SD-kaartcontroller zijn als volgt:

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D680	54912	CMDANDSTAT							
D681	54913	SECTOR0							
D682	54914	SECTOR1							
D683	54915	SECTOR2							

vervolgd ...

...vervolgd

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D684	54916	SECTOR3							
D686	54918	FILLVAL							
D68A	54922	-				VFDC1	VFDC0	VICIII	CDC00
D6AE	54958	FDCTIBEN	FDC-2XSEL	FDC-VARSPD	AUTO-2XSEL	FDCENC			
D6AF	54959	-		VLOST	VDRQ	VRNF	VEQINH	VW-FOUND	VRFOUND

- **AUTO2XSEL** Selecteert automatisch de DD- of HD-decoder voor de weergave van de laatste sector
- **CDC00** (alleen-lezen) Ingesteld indien kleurengeheugen op $DC00
- **CMDANDSTAT** SD-controllerstatus/commando
- **FDC2XSEL** Selecteert de HD-decoder voor de weergave van de laatste sector
- **FDCENC** Selecteert de diskettecodering (0=MFM, 1=RLL2,7, F=Raw-codering)
- **FDCTIBEN** Schakelt het gebruik van Track Info Block-instellingen in
- **FDCVARSPD** Schakelt automatische variabele snelheidsselectie in voor de diskettecontroller met behulp van Track Information Blocks op MEGA65 HD-diskettes
- **FILLVAL** ALLEENSCHRIJVEN: stelt het vulbyte in voor gebruik in de vulmodus, in plaats van SD-buffergegevens
- **SECTOR0** SD-controller SD-sectoradres (LSB)
- **SECTOR1** SD-controller SD-sectoradres (2e byte)
- **SECTOR2** SD-controller SD-sectoradres (3e byte)
- **SECTOR3** SD-controller SD-sectoradres (MSB)
- **VDRQ** Handmatig instellen van het f011_drq-signaal (uitsluitend bedoeld voor de virtuele F011-modus)
- **VEQINH** Handmatig instellen van het f011_eq_inhibit-signaal (uitsluitend bedoeld voor de virtuele F011-modus)
- **VFDC0** (alleen-lezen) Ingesteld indien station 0 is gevirtualiseerd (sectoren geleverd via de seriële monitorinterface)
- **VFDC1** (alleen-lezen) Ingesteld indien station 1 is gevirtualiseerd (sectoren geleverd via de seriële monitorinterface)
- **VICIII** (alleen-lezen) Ingesteld indien de VIC-IV- of Ethernet I/O-bank zichtbaar is
- **VLOST** Handmatig instellen van het f011_lost-signaal (uitsluitend bedoeld voor de virtuele F011-modus)

- **VRFOUND** Handmatig instellen van het f011_rsector_found-signaal (uitsluitend bedoeld voor de virtuele F011-modus)
- **VRNF** Handmatig instellen van het f011_rnf-signaal (uitsluitend bedoeld voor de virtuele F011-modus)
- **VWFOUND** Handmatig instellen van het f011_wsector_found-signaal (uitsluitend bedoeld voor de virtuele F011-modus)

TOUCHPADINTERFACE

Bepaalde MEGA65-varianten zijn uitgerust met een LCD-touchpad, met name de handheld MEGAphone-versie van de MEGA65. De aanraakinterface ondersteunt het detecteren van twee gelijktijdige aanraakgebeurtenissen. Sommige varianten ondersteunen mogelijk ook gebarenherkenning, hoewel deze nog zeer experimenteel is.

De aanraakdetectie-interface in de 45IO27 wordt aangevuld door de schermtoetsenbordinterface van de 4551-UART- en GPIO-controller. Raadpleeg paragraaf 6 voor verdere informatie. Bijzonder relevant is bit 7 van de registers $D615 -$D617, die het mogelijk maakt de schermtoetsenbordinterface te activeren, te kiezen of het schermtoetsenbord in het bovenste of onderste gedeelte van het scherm wordt geplaatst, en of het primaire of secundaire schermtoetsenbord wordt weergegeven.

De rechtstreekse verbindingen tussen de 4551 en de 45IO27 combineren informatie over het op dat moment weergegeven schermtoetsenbord en de aanraakinterfacecontroller, waardoor synthetische toetsenbordgebeurtenissen automatisch worden geactiveerd wanneer u op het schermtoetsenbordgedeelte van de aanraakinterface drukt. Hierdoor kan de aanraakinterface worden gebruikt om het schermtoetsenbord aan te sturen, zonder dat daarvoor ondersteuning van gebruikersprogramma's nodig is. Dit werkt zelfs wanneer het schermtoetsenbord beweegt tijdens de activering of wanneer het tussen de boven- en onderkant van het scherm wordt verplaatst.

Omdat aanraakinterfaces kalibratie kunnen vereisen, biedt de 45IO27 een lineaire transformatie van zowel de X- als de Y-coördinaten van een aanraakgebeurtenis. Er zijn specifiek schaalregisters (TCHXSCALE en TCHYSCALE) en offsetregisters (TCHXDELTA en TCHYDELTA) die deze transformatie verzorgen. Het is ook mogelijk de coördinaten van het touchpad te spiegelen op de X-as, de Y-as, of op beide assen. Deze kalibratieregisters beïnvloeden ook de werking van het schermtoetsenbord.

Er dient ook te worden opgemerkt dat sommige aanraakinterfaces geen constante horizontale of verticale resolutie hebben. Sommige panelen hebben bijvoorbeeld een gedeelte met een lage horizontale resolutie in het midden van het paneel, waarvoor enige zorg nodig kan zijn.

Om de primaire aanraakgebeurtenis te detecteren, kunnen de registers TOUCH1XLSB, TOUCH1XMSB, TOUCH1YLSB en TOUCH1YMSB worden gelezen. Voor de tweede aanraakgebeurtenis bestaan vergelijkbare registers: TOUCH2XLSB, TOUCH2XMSB, TOUCH2YLSB en TOUCH2YMSB. Elke aanraakgebeurtenis heeft een vlag van één bit

die aangeeft of de aanraakgebeurtenis op dat moment geldig is: de EV1- en EV2-bits van het register $D6B0. Er zijn ook overeenkomstige bitvelden die aangeven of een bepaalde aanraakgebeurtenis is ingedrukt of losgelaten, waardoor kan worden gedetecteerd wanneer een vinger zowel contact met het scherm maakt als het contact verbreekt. De UPDN1- en UPDN2-signalen verschaffen deze informatie. Binaire waarden van respectievelijk 01 en 10 geven aan of de vinger van het touchpad is verwijderd of ertegen is gedrukt. De waarden 00 en 11 betekenen dat een vinger al dan niet tegen het touchpad wordt gehouden.

De primaire aanraakgebeurtenis wordt tevens ingevoerd in de lichtpen-ingang van de VIC-IV en kan worden gedetecteerd met de gewone lichtpenregisters van de VIC-IV.

De registers voor de touchpadinterface zijn als volgt:

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D6B0	54960	YINV	XINV	UPDN2		UPDN1		EV2	EV1
D6B1	54961	CALXSCALELSB							
D6B2	54962	CALXSCALEMSB							
D6B3	54963	CALYSCALELSB							
D6B4	54964	CALYSCALEMSB							
D6B5	54965	CALXDELTALSB							
D6B7	54967	CALYDELTALSB							
D6B8	54968	CALYDELTAMSB							
D6B9	54969	TOUCH1XLSB							
D6BA	54970	TOUCH1YLSB							
D6BB	54971	-		TOUCH1YMSB		-		TOUCH1XMSB	
D6BC	54972	TOUCH2XLSB							
D6BD	54973	TOUCH2YLSB							
D6BE	54974	-		TOUCH2YMSB		-		TOUCH2XMSB	
D6C0	54976	GESTUREID				GESTUREDIR			

- **CALXDELTALSB** X-delta van de touchpad (LSB)
- **CALXSCALELSB** X-schaling van de touchpad (LSB)
- **CALXSCALEMSB** X-schaling van de touchpad (MSB)
- **CALYDELTALSB** Y-delta van de touchpad (LSB)
- **CALYDELTAMSB** Y-delta van de touchpad (MSB)
- **CALYSCALELSB** Y-schaling van de touchpad (LSB)
- **CALYSCALEMSB** Y-schaling van de touchpad (MSB)
- **EV1** Aanraakgebeurtenis 1 is geldig
- **EV2** Aanraakgebeurtenis 2 is geldig

- **GESTUREDIR** Gebaarrichtingen van de touchpad (links, rechts, omhoog, omlaag)
- **GESTUREID** Gebaren-ID van de touchpad
- **TOUCH1XLSB** X-coördinaat van aanraking #1 op de touchpad (LSB)
- **TOUCH1XMSB** X-coördinaat van aanraking #1 op de touchpad (MSBs)
- **TOUCH1YLSB** Y-coördinaat van aanraking #1 op de touchpad (LSB)
- **TOUCH1YMSB** Y-coördinaat van aanraking #1 op de touchpad (MSBs)
- **TOUCH2XLSB** X-coördinaat van aanraking #2 op de touchpad (LSB)
- **TOUCH2XMSB** X-coördinaat van aanraking #2 op de touchpad (MSBs)
- **TOUCH2YLSB** Y-coördinaat van aanraking #2 op de touchpad (LSB)
- **TOUCH2YMSB** Y-coördinaat van aanraking #2 op de touchpad (MSBs)
- **UPDN1** Status van aanraking 1 (losgelaten/ingedrukt)
- **UPDN2** Status van aanraking 2 (losgelaten/ingedrukt)
- **XINV** Inverteert de horizontale as
- **YINV** Inverteert de verticale as

ONDERSTEUNENDE FUNCTIES VOOR AUDIO

De 45IO27 verschaft de primaire interface naar de volledige cross-bar-audiomixer van de MEGA65. Dit omvat de interface voor het lezen of wijzigen van de mixercoëfficiënten, de toegang tot de feedbackregisters van de mixer en het instellen van de 16-bits digitale samplewaarden, welke tevens twee van de ingangskanalen van de audiomixer vormen.

De audiomixer bestaat uit 128 coëfficiënten, die elk 16 bits groot zijn. Elk audio-uitgangskanaal, zoals de luidspreker links, de luidspreker rechts, de hoofdtelefoon links, de hoofdtelefoon rechts, de cellulaire modem 1 (alleen op MEGAphone-modellen) enzovoort, wordt gegenereerd door elk audio-ingangskanaal te nemen, het met de passende coëfficiënt te vermenigvuldigen en het aan de totale uitvoer van het audio-uitgangskanaal toe te voegen.

Omdat elk audio-uitgangskanaal zijn eigen reeks coëfficiënten heeft die op alle audio-ingangskanalen worden toegepast, betekent dit dat het mogelijk is om volstrekt verschillende audio uit te voeren over elk audiokanaal: Zo is het bijvoorbeeld mogelijk om uw favoriete quadrafonische SID-muziek af te spelen via de hoofdtelefoon terwijl u voorbijgangers trakteert op Amiga-stijl MOD-audio. Dit is de reden dat de audiomixer een `full cross-bar-mixer` wordt genoemd, omdat er geen beperkingen zijn aan de

wijze waarop u elk audio-uitgangskanaal mixt. In dit opzicht lijkt hij sterk op een volwaardige audiomengtafel, en maakt hij verschillende mengniveaus voor verschillende luidsprekers mogelijk.

Omdat de audiocoëfficiënten elk 16 bits groot zijn, wordt elke coëfficiënt gevormd door twee opeenvolgende bytes van de audiocoëfficiëntenruimte. Wijzigingen aan de audiocoëfficiënten worden onmiddellijk van kracht, dus dient er voorzichtigheid te worden betracht bij het wijzigen van coëfficiënten om hoorbare klikken en plofjes te voorkomen. U moet bovendien 32 cycli laten verlopen voordat u de geselecteerde audiocoëfficiënt wijzigt; indien u dit niet doet, kan de wijziging worden verworpen indien de accumulator van de audiomixer die coëfficiënt nog niet opnieuw heeft verwerkt.

De audiobronnen op de MEGA65- en MEGAphone-apparaten zijn als volgt:

Kanaal-ID v/d ingang	**Aansluiting**
$0 (0)	Linker SID's
$1 (1)	Rechter SID's
$2 (2)	Modem-bay 1 (alleen MEGAphone)
$3 (3)	Modem-bay 2 (alleen MEGAphone)
$4 (4)	Bluetooth™ Links
$5 (5)	Bluetooth™ Rechts
$6 (6)	Hoofdtelefoon-interface 1
$7 (7)	Hoofdtelefoon-interface 2
$8 (8)	Digitale audio Links
$9 (9)	Digitale audio Rechts
$A (10)	MEMs-microfoon 0 (alleen Nexys4 en MEGAphone)
$B (11)	MEMs-microfoon 1 (alleen MEGAphone)
$C (12)	MEMs-microfoon 2 (alleen MEGAphone)
$D (13)	MEMs-microfoon 3 (alleen MEGAphone)
$E (14)	Hoofdtelefoon-audiojack-microfoon (alleen Nexys4 en MEGAphone)
$F (15)	OPL-compatibele FM-audio (*deelt de coëfficiënt met ingang 14*)

De OPL-compatibele FM-audio op bron 15 wordt aangestuurd door de coëfficiënt voor bron 14. De reden hiervoor is dat de coëfficiënt voor bron 15 het mastervolumeniveau voor elke uitvoer bepaalt. Het OPL-compatibele FM-audioapparaat is momenteel niet functioneel in de MEGA65-core.

De cross-bar-audiomixer van het audiosysteem ondersteunt de volgende acht uitgangskanalen:

Kanaal-ID v/d uitgang	Aansluiting
$0 (0)	Linker primaire luidspreker (digitale audio op de MEGA65 R2/R3, fysieke luidspreker op de MEGAphone, audiojack op de Nexys4)
$1 (1)	Rechter primaire luidspreker (digitale audio op de MEGA65 R2/R3, fysieke luidspreker op de MEGAphone, audiojack op de Nexys4)
$2 (2)	Modem-bay 1 audio-uitvoer (alleen MEGAphone)
$3 (3)	Modem-bay 2 audio-uitvoer (alleen MEGAphone)
$4 (4)	Bluetooth-audio links (alleen MEGAphone)
$5 (5)	Bluetooth-audio rechts (alleen MEGAphone)
$6 (6)	Hoofdtelefoon-uitvoer links (MEGA65 R2/R3 en MEGAphone. Op Nexys4-borden stuurt de primaire luidspreker de 3.5 mm-audiojack aan)
$7 (7)	Hoofdtelefoon-uitvoer rechts (MEGA65 R2/R3 en MEGAphone. Op Nexys4-borden stuurt de primaire luidspreker de 3.5 mm-audiojack aan)

Om het coëfficiëntregisternummer voor een bepaalde bron en uitvoer te bepalen, vermenigvuldigt u het uitvoernummer met 32 en het bronnummer met 2. Dit geeft het registernummer van de LSB van de 16-bits coëfficiënt. De MSB bevindt zich in het volgende register. Om bijvoorbeeld de coëfficiënt van de rechter SID's in te stellen op de tweede modem bay-audio-uitvoer, wordt de coëfficiënt $32\times3+1\times2 = 96+2 = 98$.

Overige audiofuncties

Het audiosysteem ondersteunt verscheidene andere functies, waarvan de documentatie nog gereed moet komen.

Mixer-feedbackregisters

Deze registers maken het mogelijk dat de processor de gemengde audio van een bepaald uitgangskanaal uitleest. Dit kan worden gebruikt om weergaven van audiogolfformen te maken of om bepaalde audio-effecten, zoals reverb, te realiseren.

8/16-bits stereo digitale audio

Er zijn registers beschikbaar om 8- of 16-bits samplewaarden rechtstreeks in specifieke invoerkanalen van de audiomixer in te voeren. Dit biedt een eenvoudige manier om digitaal audiogegevens af te spelen. Deze voorziening is met name nuttig voor procedureel gegenereerde audiogegevens. Voor opgenomen samples is het doorgaans eenvoudiger en beter om de Audio DMA-functionaliteit te gebruiken die het afspelen van digitaal audio volledig automatiseert. Audio DMA is vooral nuttig op hogere samplefrequenties, omdat het de belasting van de CPU vermindert en sample-jitter aanzienlijk verkleint.

Pulsbreedte versus pulsdichtheidsmodulatie

Voor MEGA65-modellen die een 1-bits audio-uitvoer met oversampling gebruiken (tot en met het R3/R3A-model) is het mogelijk te kiezen tussen deze twee verschillende oversamplingmethoden. Beide hebben vergelijkbare prestaties, maar gebruikers hebben wellicht een voorkeur voor de ene boven de andere. Deze keuze heeft geen effect op het R4 of nieuwere modellen die een dedicated audio-DAC gebruiken om audio naar de 3.5 mm-audiojack te voeren en die van nature een betere audiokwaliteit hebben. Voor eigenaren van oudere modellen kan de geplande interne uitbreidingskaart van de MEGA65 een verbeterd audio-uitvoercircuit bevatten, afhankelijk van de definitieve configuratie van die kaart. Er is momenteel nog geen termijn bekend waarop deze kaart beschikbaar komt. In de tussentijd is het gebruik van HDMI-audio-uitvoer de aanbevolen oplossing, omdat de audio digitaal over de HDMI-kabel wordt gecodeerd.

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D6F4	55028	MIXREGSEL							
D6F5	55029	MIXREGDATA							
D6F8	55032	DIGILLSB							
D6F9	55033	DIGILMSB							
D6FA	55034	DIGIRLSB							
D6FB	55035	DIGIRMSB							
D6FC	55036	READBACKLSB							
D6FD	55037	READBACKMSB							
D711	55057	-				PWMPDM	-		

- **DIGILEFTLSB** Digitale audio, linkerkanaal, LSB
- **DIGILEFTMSB** Digitale audio, linkerkanaal, MSB
- **DIGILLSB** 16-bits digitale audio-uitgang (linker LSB)
- **DIGILMSB** 16-bits digitale audio-uitgang (linker MSB)
- **DIGIRIGHTLSB** Digitale audio, linkerkanaal, LSB
- **DIGIRIGHTMSB** Digitale audio, linkerkanaal, MSB
- **DIGIRLSB** 16-bits digitale audio-uitgang (rechter LSB)
- **DIGIRMSB** 16-bits digitale audio-uitgang (rechter MSB)
- **MIXREGDATA** Leespoort van het audiomixerregister
- **MIXREGSEL** Selectie van het audiomixerregister
- **PWMPDM** Selectie van de PWM/PDM-audio-encodering
- **READBACKLSB** De LSB van de audio-read-back (de bron wordt geselecteerd door $D6F4)
- **READBACKMSB** De MSB van de audio-read-back (de bron wordt geselecteerd door $D6F4)

DIVERSE I/O-FUNCTIES

HOOFDSTUK 9

4541 seriële buscontroller

- Overzicht
- Mogelijkheden van de 4541
- Werkingsprincipe
- Voorbeelden
- Commandoreferentie
- Registertabel
- Seriële bustiming
- Optionele geïntegreerde datalogger

OVERZICHT

De 4541 is een buscontroller die compatibel is met de seriële randapparatuurbus van Commodore™ en die de inspanning die nodig is om met apparaten op deze bus te communiceren sterk vermindert.

MOGELIJKHEDEN VAN DE 4541

Ondersteuning van versnelde seriële protocolvarianten

De 4541 ondersteunt de JiffyDOS™-uitbreidingen op dit protocol, waarmee gegevensoverdrachten ongeveer 10× sneller verlopen dan met het oorspronkelijke protocol. Van een toekomstige revisie van de 4541 wordt verwacht dat deze het snelle seriële protocol van Commodore ondersteunt, dat in de schijfstations 1571 en 1581 aanwezig is.

Processorontlasting via interruptondersteuning

De 4541 voert seriële communicatie uit onafhankelijk van de microprocessor. Samen met een IRQ-functionaliteit stelt dit software in staat om door te gaan met andere taken terwijl seriële communicatie met randapparaten plaatsvindt, waarbij slechts een korte onderbreking nodig is wanneer het volgende gebeurtenis op de seriële randapparatuurbus optreedt.

Onafhankelijkheid van processorsnelheid

Een groot voordeel van de 4541 is dat deze ook alle tijdsvereisten van communicatie op deze bus afhandelt, waardoor de bus kan worden aangestuurd door een processor die op verschillende snelheden kan werken, zonder de software van de buscontroller aan te hoeven passen.

Coëxistentie via open-collectorlogica

Omdat de 4541 open-collectorlogica gebruikt, kan deze parallel worden gebruikt met bestaande softwarematige implementaties van het busprotocol, waarmee compatibiliteit met bestaande software wordt gewaarborgd. In de MEGA65 is de 4541 in deze configuratie geïnstalleerd, waardoor de bestaande softwarematige communicatiesoftware voor de seriële randapparatuurbus ongewijzigd kan blijven werken.

WERKINGSPRINCIPE

De 4541 biedt een eenvoudige interface: u geeft opdrachten, wacht op een reactie en haalt vervolgens de gegevens op die de opdracht heeft verzameld. Sommige opdrachten vereisen ook een gegevensbyte, die via een apart register wordt aangeboden. Er is ook een register met apparaatinformatie, waarmee u kunt zien wat de 4541 van de huidige status van het meest recent gevraagde apparaat vindt, waaronder of het aanwezig is en of het een van beide of beide uitbreidingen van het JiffyDOS™- of Commodore™ 128-protocol ondersteunt.

Om bijvoorbeeld de Attention-lijn vrij te geven, schrijft u de juiste commandobytewaarde (65 = $41) naar het commandoregister op $D698 en controleert u vervolgens de voltooiingsstatus in het statusregister op $D697, zoals weergegeven in het volgende voorbeeld in BASIC65:

```
10 POKE $D698,$41
20 IF ( PEEK($D697) AND $20 ) = $00 GOTO 20
30 PRINT "GEREED"
```

De 4541 implementeert een reeks opdrachten die nauw aansluiten bij de KERNAL-aanroepen die worden gebruikt om de IEC-bus op de C64 en verwante computers te besturen. In de meeste gevallen is er één bijbehorende opdracht voor de 4541, hoewel u in een aantal gevallen twee opdrachten moet geven, zoals samengevat in de volgende tabel:

KERNAL-aanroep	**Betekenis en bijbehorende 4541-opdracht(en)**
$FF93 LSTNSA	Verzend het secundaire LISTEN-adres. 4541-equivalent: Gegevens = Binair OR van $60 en het gewenste secundaire adres. Opdracht $30. Vervolgens opdracht $41 om de Attention-lijn vrij te geven.
$FF96 TALKSA	Verzend het secundaire TALK-adres. 4541-equivalent: Gegevens = Binair OR van $60 en het gewenste secundaire adres. Opdracht $30. Vervolgens opdracht $41 om de Attention-lijn vrij te geven.
$FFA5 IECIN	Ontvang een byte van de seriële randapparatuurbus. 4541-equivalent : Opdracht $32. De ontvangen byte is beschikbaar in het gegevensregister na voltooiing.
$FFA8 IECOUT	Verzend een byte naar de seriële randapparatuurbus. 4541-equivalent : Gegevens = de te verzenden byte. Opdracht $31 (of $30 indien de byte onder Attention wordt verzonden).
$FFA8 UNTALK	Verzend UNTALK-opdracht naar de seriële randapparatuurbus. 4541-equivalent : Gegevens = $5F. Opdracht $30.

KERNAL-aanroep	Betekenis en bijbehorende 4541-opdracht(en)
$FFAB UNLISTN	Verzend UNLISTEN-opdracht naar de seriële randapparatuurbus. 4541-equivalent : Gegevens = $3F. Opdracht $30.
$FFB1 LISTEN	Verzend het LISTEN-commando naar de seriële randapparatuurbus. 4541-equivalent: Gegevens = $20 plus het apparaatnummer. Opdracht $30.
$FFB4 TALK	Verzend het TALK-commando naar de seriële randapparatuurbus. 4541-equivalent: Gegevens = $40 plus het apparaatnummer. Opdracht $30.
$FFB7 READSS	Lees de status van de seriële randapparatuurbus. 4541-equivalent: Lees de statusbits vanaf $D698. Voor het gemak hebben de bovenste bits van deze statusbyte dezelfde indeling als in de KERNAL.

De 4541 is zeer gehoorzaam: vraagt u hem iets nieuws, dan begint hij daar onmiddellijk mee, zelfs midden in een andere taak – zelfs al is hij halverwege het verzenden van een byte gegevens naar de seriële randapparatuurbus!

U dient dan ook altijd te wachten totdat de IRQREADY-bit in $D697 is gezet voordat u een commando geeft of de statusbits leest, zodat u er zeker van bent dat de controller de laatste gevraagde taak heeft afgerond.

VOORBEELDEN

De status van het DOS-kanaal lezen

Het volgende BASIC65-programma kan worden gebruikt om met een apparaat op de seriële randapparatuurbus te communiceren. Omdat het de 4541 gebruikt en niet de CBDOS van de C65/MEGA65, negeert dit programma het interne 3.5"-schijfstation, omdat dat station niet op de seriële randapparatuurbus is aangesloten.

```
10 REM BREEK EEN EVENTUEEL LOPEND 4541-COMMANDO af
20 POKE $D698,$00
30 REM RESET DE SERIELE RANDAPPARATUURBUS DOOR
40 REM DE RESETPIN 1 SECOND LANG LAAG TE HOUDEN
50 POKE $D698,$72 : SLEEP 1 : POKE $D698,$52
60 REM GUN AANGESLOTEN APPARATEN NA RESET TIJD OM GEREED TE KOMEN
70 SLEEP 2: REM HET DUURT EVEN VOOR DE 1541 GERESET IS!
80 REM SELECTEER HET APPARAATNUMMER WAARNAAR GELUISTERD WORDT
90 D = 8
100 PRINT"DEVICE"; D; "TALK"
110 POKE $D699,$40 + D : POKE $D698,$30 : GOSUB 1000
120 PRINT "SECUNDAIR ADDRES 15"
130 POKE $D699,$6F : POKE $D698,$30 : GOSUB 1000
140 PRINT"OMKEREREN NAAR LUISTEREN"
150 POKE $D698,$35 : GOSUB 1000
160 PRINT "LEES DOSSTATUS"
170 POKE $D698,$32 : GOSUB 1000
180 PRINT CHR$( PEEK($D699) );
190 CC = CC + 1 : IF CC < 200 GOTO 180
999 END
1000 REM WACHT TOT DE 4541 HET COMMANDO HEEFT AFGEROND
1010 S=PEEK($D697) : IF (S AND 32) = 32 GOTO 1020
1020 GOTO 1000
1030 IF S AND 128 THEN PRINT "DEVICE NIET AANWEZIG" : END
1040 RETURN
```

Als u een 1581 met een JiffyDOS™-ROM heeft aangesloten, toont dit programma bij uitvoering het volgende:

```
READY.
RUN
DEVICE 8 TALK
SECUNDAIR ADRES 15
OMKEREN NAAR LUISTEREN
LEES DOSSTATUS
73,(C) 1989 JIFFYDOS 6.0 1581,00,00
00, OK,00,00
00, OK,00,00
00, OK,00,00
...
00, OK,0
READY.
```

COMMANDOREFERENTIE

De volgende tabel geeft een overzicht van de door de 4541 ondersteunde commandocodes.

Commandobyte	Actie
Lopende commando's onderbreken	
$00	Onderbreek elk uitgevoerd commando.
$01	Herstel de toestand van de controller: ATN, CLK, DATA en SRQ loslaten en de standaard protocolselectie inschakelen.
Bit-bashing-commando's	
$41	Attention-lijn (ATN) loslaten naar 5V.
$61	Attention-lijn (ATN) naar 0V trekken.
$43	Kloklijn (CLK) loslaten naar 5V.
$63	Kloklijn (CLK) naar 0V trekken.
$44	Gegevenslijn (DATA) loslaten naar 5V.
$64	Gegevenslijn (DATA) naar 0V trekken.
$53	Snelle seriële kloklijn (SRQ) loslaten naar 5V.
$73	Snelle seriële kloklijn (SRQ) naar 0V trekken.
$52	Laat de resetlijn (RESET) los op 5V.
$72	Trek de resetlijn (RESET) naar 0V.
$4C	Laat alle lijnen (ATN, CLK, DATA, SRQ, RESET) los op 5V.
Protocolvariant-besturingscommando's	
$4A	Sta het aanvragen van de JiffyDOS™ uitbreiding voor het seriële protocol toe.
$6A	Sta het aanvragen van de JiffyDOS™ uitbreiding voor het seriële protocol niet toe.
$46	Sta het aanvragen van de Commodore™ 1571/1581 snelle seriële uitbreiding voor het seriële protocol toe.
$66	Sta het aanvragen van de Commodore™ 1571/1581 snelle seriële uitbreiding voor het seriële protocol niet toe.

Commandobyte	Actie
$50	Sta het aanvragen van alle seriële uitbreidingen toe.
$70	Sta het aanvragen van alle seriële uitbreidingen niet toe.

Seriële busprotocolcommando's

$30	Verzend een byte in $D699 onder Attention. De Attention-lijn wordt aan het einde van de transactie op 0V gehouden.
$31	Verzend een byte in $D699 zonder Attention. De Attention-lijn wordt losgelaten indien deze voorafgaand was geactiveerd.
$32	Ontvang een byte van de seriële randapparatuurbus. Er moet een vooraf geactiveerde spreker ("**talk**er") aanwezig zijn. De ontvangen byte wordt opgeslagen in $D699.
$33	GERESERVEERD
$34	Verzend een byte en geef einde-aanduiding (EOI) aan. De Attention-lijn wordt voorafgaand aan de verzending losgelaten indien deze was geactiveerd.
$35	Draai om van spreker ("**talk**er") naar luisteraar ("**listen**er").

Commando's voor de protocoltiming

$80	Reset de protocoltiming naar de standaardwaarden.
$81	Stel de T_R-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$82	Stel de T_{TK}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$83	Stel de T_{DC}-time-out in (1 - 255 milliseconden). De oude waarde kan worden gelezen uit $D699.
$84	Stel de T_{BB}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$85	Stel de T_{HA}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$86	Stel de T_{ST}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.

Commandobyte	Actie
$87	Stel de T_{VT}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$88	Stel de T_{AL}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$89	Stel de T_{AC}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$8A	Stel de T_{AT}-vertraging in (1 - 255 milliseconden). De oude waarde kan worden gelezen uit $D699.
$8B	Stel de T_H-vertraging in (1 - 255 milliseconden). De oude waarde kan worden gelezen uit $D699. *Deze waarde heeft momenteel geen effect, omdat de 4541 een werkelijk oneindige wachttijd door een luisteraar mogelijk maakt, met uitzondering van bytes die onder Attention worden verzonden, waarbij in plaats daarvan T_{HA} wordt gebruikt.*
$8C	Stel de T_{NE}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$8D	Stel de T_F-vertraging in (4 - 1020 microseconden). De oude waarde kan worden gelezen uit $D699, geschaald met een factor 4.
$8E	Stel de T_{YE}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$8F	Stel de T_{EI}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$90	Stel de T_{AR}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$91	Stel de T_{JT}-vertraging in (4 - 1020 microseconden). De oude waarde kan worden gelezen uit $D699, geschaald met een factor 4.
$92	Stel de T_{J0}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$93	Stel de T_{J1}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$94	Stel de T_{J2}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$95	Stel de T_{J3}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.

Commandobyte	Actie
$96	Stel de T_{J4}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$97	Stel de T_{J5}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$98	Stel de T_{J6}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$99	Stel de T_{J7}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$9A	Stel de T_{J8}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$9B	Stel de T_{J9}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$9C	Stel de T_{J10}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$9D	Stel de T_{J11}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$9E	Stel de T_{JR}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$9F	Stel de T_{FS}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$A0	Stel de T_{FF}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$A1	Stel de T_{PULLUP}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.
$A2	Stel de T_{JD}-vertraging in (4 - 1020 microseconden). De oude waarde kan worden gelezen uit $D699, geschaald met een factor 4.
$A3	Stel de T_{J12}-vertraging in (1 - 255 microseconden). De oude waarde kan worden gelezen uit $D699.

Diagnostische opdrachten

$D0	Activeer de optionele geïntegreerde datalogger indien ingeschakeld.
$D1	Begin met het uitzenden van een 1 kHz-pulstrein op de CLK- en DATA-lijnen. Dit duurt voort totdat een ander commando het onderbreekt.

REGISTERTABEL

De 4541 heeft de volgende registers:

HEX	DEC	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
D694	54932	DATALOG0							
D695	54933	DATALOG1							
D697	54935	IRQFLAG	IRQRX	IRQRDY	IRQTO	IRQEN	IRQRXEN	IRQRDYEN	IRQTOEN
D698	54936	STNODEV	STNOEOI	STSRQ	STVERIFY	STC	STD	STTO	STDDIR
D699	54937	DATA							
D69A	54938	PRESENT	PROT		DIATN	DEVNUM			

- **DATA** De gegevensbyte die van de IEC-bus wordt gelezen
- **DATALOG0** Toegang tot de geïntegreerde datalogger in de IEC-controller
- **DATALOG1** Toegang tot de geïntegreerde datalogger in de IEC-controller
- **DEVNUM** De onderste 4 bits van het op dat moment geselecteerde apparaatnummer
- **DIATN** Het apparaat wordt momenteel in de attention-toestand gehouden
- **IRQEN** Schakelt de interrupts in indien deze is gezet
- **IRQFLAG** Interruptvlag. Deze wordt gezet indien een IRQ-gebeurtenis wordt geactiveerd.
- **IRQRDY** Gezet indien gereed om een opdracht te verwerken
- **IRQRDYEN** Schakelt de TX-interruptbron in indien deze is gezet
- **IRQRX** Gezet indien een byte van een luisteraar is ontvangen.
- **IRQRXEN** Schakelt de RX-interruptbron in indien deze is gezet
- **IRQTO** Gezet indien een protocol-timeout is opgetreden, bijv. wanneer het apparaat niet wordt gevonden.
- **IRQTOEN** Schakelt de timeout-interruptbron in indien deze is gezet
- **PRESENT** Het apparaat is aanwezig
- **PROT** Protocolondersteuning van het apparaat (5=C128/C65 FAST, bit 6 = JiffyDOS(tm))
- **STC** Toestand van de CLK-lijn
- **STD** Toestand van de DATA-lijn

- **STDDIR** De gegevensrichting op het moment dat een timeout optreedt.
- **STNODEV** Het apparaat is niet aanwezig
- **STNOEOI** End of Indicate (EOI/EOF)
- **STSRQ** Toestand van de SRQ-lijn
- **STTO** Er is een timeout opgetreden
- **STVERIFY** Er is een verificatiefout opgetreden

SERIËLE BUSTIMING

Dit hoofdstuk beschrijft de tijdsvereisten voornamelijk vanuit het perspectief van een buscontroller. Indien u seriële busapparaten ontwerpt, besteed dan zorgvuldig aandacht aan de manier waarop deze tijdsvereisten de apparaten beïnvloeden.

Een byte onder Attention verzenden

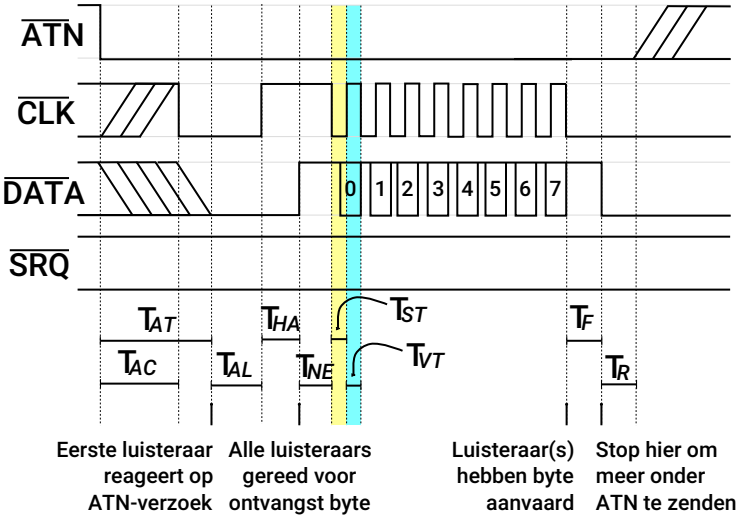
Wanneer de controller de aandacht van één of meer apparaten op de bus wil trekken, gebruikt hij de ATN-lijn om dit aan te geven: hij trekt deze laag naar 0V en stuurt vervolgens één of meer bytes die door alle apparaten op de bus worden ontvangen.

De SRQ-lijn is niet actief in deze transactie.

Samengevat:

1. De controller trekt de ATN naar 0V en laat de CLK-, DATA- en SRQ-lijnen los indien hij een van deze op 0V hield.
2. Alle apparaten op de bus merken op dat de ATN-lijn op 0V staat en geven dit aan door de DATA-lijn naar 0V te trekken. De controller kan niet zien noch hoeveel apparaten er hebben gereageerd, noch of alle apparaten op de bus hebben gereageerd; hij kan enkel vaststellen dat ten minste één apparaat heeft gereageerd.
3. Indien geen enkel apparaat reageert, blijft DATA op 5V, omdat niemand deze naar beneden trekt, en geeft de controller een DEVICE NOT PRESENT-fout aan.
4. Terwijl de controller wacht op reactie van de apparaten, trekt hij ook de CLK-lijn naar 0V, doorgaans kort nadat hij de ATN-lijn naar 0V heeft getrokken.
5. Nadat de controller enige tijd heeft gewacht, doorgaans 1 milliseconde, neemt hij aan dat alle apparaten die op het ATN-verzoek zullen reageren, dat inmiddels hebben gedaan.
6. Vervolgens laat de controller de CLK-lijn los naar 5V om aan te geven dat hij een byte gegevens op de bus wil verzenden. Dit vormt het begin van het verzenden van een byte onder Attention. Al deze voorgaande stappen hadden tot doel de ATN-communicatie tot stand te brengen.

7. De controller wacht nu tot alle apparaten hun bereidheid om een byte te ontvangen aangeven door de DATA-lijn terug naar 5V te laten gaan. Omdat de bus open collector is, kan zelfs één enkel apparaat dat nog niet gereed is, de DATA-lijn naar 0V blijven trekken, waardoor de controller moet wachten. Indien dit te lang duurt, kan de controller opgeven en een time-out aangeven.
8. Zodra het laatste apparaat heeft aangegeven gereed te zijn om gegevens te ontvangen, wacht de controller nog iets langer om er zeker van te zijn dat alle apparaten hun laatste voorbereidingen voor het ontvangen van een byte kunnen afronden. Dit duurt niet lang.
9. Na die wachttijd begint de controller met het verzenden van de 8 bits gegevens, te beginnen met het minst significante bit (LSB), oftewel bit 0. Voor elk bit trekt hij eerst de CLK-lijn laag om aan te geven dat er gegevens op de bus worden geladen, trekt de DATA-lijn naar 0V als hij een 0-bit wenst te verzenden, of laat deze naar 5V zweven als hij een 1-bit wenst te verzenden, en laat vervolgens de CLK-lijn weer los naar 5V en houdt deze daar enige tijd. De timing van dit proces is van kritiek belang: als de CLK te kort laag of hoog is, raken de apparaten in de war en kunnen zij een of meer bits missen, wat op de bus tot volledige chaos leidt.
10. Nadat de controller het laatste bit heeft verzonden, houdt hij de CLK-lijn laag en laat de DATA-lijn los. Hij verwacht dat een of meer apparaten binnen korte tijd de DATA-lijn naar 0V trekken om de ontvangst van de byte te bevestigen. Indien dit niet gebeurt, kan de controller een time-out of een DEVICE NOT PRESENT-fout melden.
11. Aan het einde hiervan houdt de controller de CLK-lijn op 0V en houden de apparaten de DATA-lijn op 0V. Deze combinatie dient om elkaar te vertellen dat de controller nog geen volgende byte wenst te verzenden en dat de apparaten nog niet gereed zijn om de volgende byte te ontvangen.
12. Indien de controller meer bytes onder Attention wenst te verzenden, herhaalt hij dit proces vanaf de stap waarin hij de CLK-lijn naar 5V liet los.
13. Indien het echter de laatste byte betrof die onder deze Attention-aanvraag werd verzonden, wacht de controller een korte tijd en laat vervolgens de ATN-lijn los.



Symbool	Min	4541	Max	Omschrijving
T_{AT}	-	1000	∞	Reactietijd van het apparaat op Attention. *Een apparaat moet binnen deze tijd op het ATN-verzoek reageren. Indien geen enkel apparaat binnen deze tijd reageert, volgt een DEVICE NOT PRESENT-fout. Apparaten dienen derhalve de typische waarde van 1000 microseconden aan te houden.*
T_{AC}	-	20	∞?	Tijd tussen het laag trekken van ATN en het laag trekken van CLK.
T_{AL}	-	1000	∞?	Tijd tussen de reactie van het eerste apparaat op ATN en het loslaten van CLK door de controller naar 5V.
T_{HA}	-	64 ms	∞	Luisteraar-wachttijd. *Het protocol staat een luisteraar toe om gedurende een willekeurige periode of helemaal niet te wachten. De 4541 staat dit niet toe bij ATN-verzoeken.*
T_{NE}	-	40	200	Niet-EOI-tijdskanaalreactie op "Gereed voor gegevens". *Indien dit interval te lang is, wordt de byte als een EOI-byte geïnterpreteerd en zullen de apparaten een EOI-reactie vereisen (zie T_{EI}).*
T_{ST}	20*	35	∞	Bitinsteltijd. *Wordt aangeduid als T_S in de C64 Programmer's Reference Guide.*

Symbool	Min	4541	Max	Omschrijving
T_{VT}	20*	35	∞	Bitgeldigheid vasthoudtijd. *Wordt aangeduid als T_V in de C64 Programmer's Reference Guide.*
T_F	-	1000	1000	Frame-handshake (bevestiging) time-out.
T_R	20	200	∞	Tijd tussen het ontvangen van de framebevestiging en het loslaten van de ATN-lijn naar 5V.

Alle tijdseenheden zijn uitgedrukt in microseconden, tenzij anders aangegeven.

* De Commodore™ 64 Programmer's Reference Guide stelt dat T_S en T_V elk een minimumduur van 20 microseconden dienen te hebben. Onze onderzoeken wijzen uit dat bij communicatie met een 1541 veilige waarden voor T_{ST} en T_{VT} dichter bij 70 microseconden liggen.

JiffyDOS™-protocolaanvraag

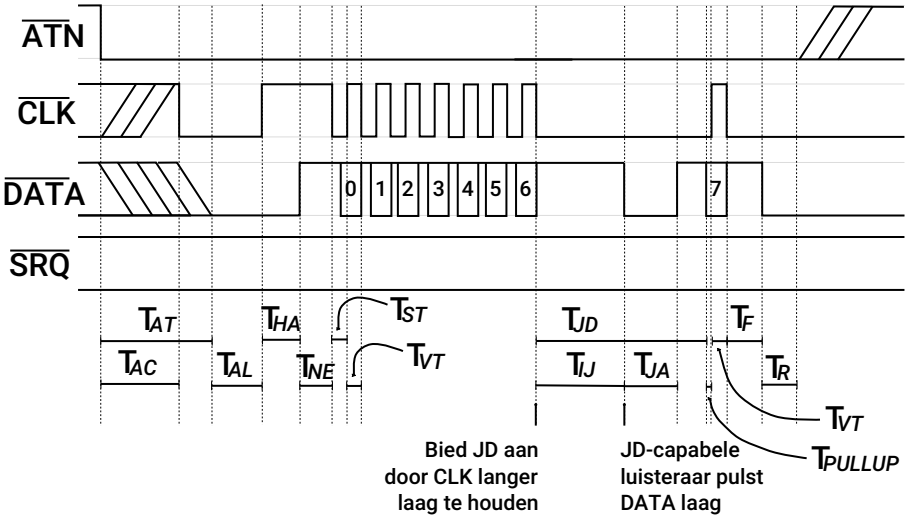
Indien ondersteuning voor de JiffyDOS™-protocoluitbreiding is ingeschakeld, vraagt de 4541 deze slechts aan tijdens de verzending van TALK- en LISTEN-commando's. Dat wil zeggen: wanneer een byte onder Attention wordt gestuurd die valt in het bereik $20 - $5F.

De SRQ-lijn is niet actief in deze transactie.

Samengevat:

1. De byte wordt normaal verzonden, tot het T_{ST}-interval vlak voor het laatste bit (bit 7) van de byte.
2. In plaats van T_{ST} af te wachten, moet de controller een veel langere periode afwachten, doorgaans 300 microseconden, zodat de luisteraar herkent dat hem gevraagd wordt of hij het JiffyDOS™ protocol ondersteunt. De controller laat ook de DATA-lijn naar 5V los.
3. Indien de luisteraar het JiffyDOS™ protocol ondersteunt, trekt hij DATA naar 0V wanneer hij opmerkt dat de CLK-lijn gedurende de langere periode laag is gehouden.
4. Nadat de luisteraar DATA lang genoeg op 0V heeft gehouden om er zeker van te zijn dat de controller dit heeft kunnen opmerken, laat hij de DATA-lijn los naar 5V. Het apparaat zal nu het JiffyDOS™ gebruiken voor de gevraagde TALK- of LISTEN-sessie.
5. De controller noteert of de DATA-lijn naar 0V is getrokken, en registreert in dat geval dat de luisteraar het JiffyDOS™-protocol heeft geselecteerd en dit protocol zal gebruiken voor de gevraagde TALK- of LISTEN-sessie.

6. De controller stuurt nu het laatste bit van de byte.
7. De controller zorgt ervoor dat de bitwaarde van het databit zichtbaar is op DATA voordat hij CLK activeert, doorgaans 1 - 4 microseconden.
8. De controller laat de CLK-lijn los naar 5V om aan de luisteraar aan te geven dat het laatste databit gereed is.
9. De rest van de byte wordt normaal overgedragen.



Symbool	Min	4541	Max	Omschrijving
T_{JD}	320	320	∞	Tijd dat CLK op 0V wordt gehouden tussen de twee laatste bits van de byte om het nevenkanaal te markeren; via het nevenkanaal geeft de controller aan dat hij de JiffyDOS™-protocoluitbeiding wenst te gebruiken.
T_{IJ}	100	-	295	Tijd voordat de luisteraar de selectie van de JiffyDOS™-protocoluitbreiding aanvaardt en de bevestiging ervan begint.
T_{IJ}	4/100*	-	200	Tijd dat de luisteraar de DATA-lijn laag houdt om de selectie van de JiffyDOS™-protocoluitbreiding te bevestigen.

Alle tijdseenheden zijn uitgedrukt in microseconden, tenzij anders aangegeven.

* De 4541 kan pulsbreedtes van slechts 4 microseconden detecteren. Software-implementaties van het protocol vereisen echter een grotere waarde. JiffyDOS™gebruikt 100 microseconden.

Verzenden van controller naar apparaat onder JiffyDOS™

In tegenstelling tot het standaard seriële protocol maakt het JiffyDOS™-protocol gebruik van verschillende protocollen voor verzenden en ontvangen. Dit is noodzakelijk bij software-implementaties om de maximale snelheid te bereiken, omdat de seriële apparaatlijnen op verschillende registerbits zijn gemapt, en een onderdeel van de kracht van het JiffyDOS™-protocol bestaat uit de slimme manipulatie van de ontvangen bits om het complete byte in de minst mogelijke tijd te reconstrueren, en eveneens bij het verzenden.

Het resultaat is opmerkelijk: JiffyDOS™ kan een byte verzenden in ongeveer dezelfde tijd die het originele protocol nodig heeft om één bit te verzenden!

Let op dat het JiffyDOS™-protocol databits met de tegenovergestelde polariteit verzendt vergeleken met het standaard protocol: een 1-bit wordt aangegeven met 0V, terwijl een 0-bit wordt aangegeven met 5V.

De SRQ-lijn is niet actief bij deze transactie.

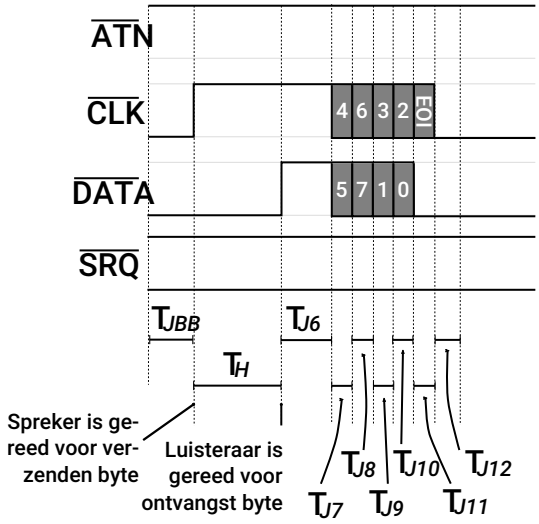
Samengevat:

1. Afhankelijk van de toestand van de controller op het moment dat deze met verzenden begint, houdt hij mogelijk de CLK-lijn op 0V. Zo ja, dan laat hij deze onmiddellijk los.
2. Vervolgens geeft het apparaat na een willekeurige tijd aan dat het gereed is om een byte van de controller te ontvangen, door de DATA-lijn naar 5V los te laten.
3. De timing voor het resterende deel van de byteoverdracht is nu cruciaal, omdat het apparaat direct na het loslaten van DATA de CLK- en DATA-signalen gaat uitlezen om de byte over te dragen.
4. De controller wacht enkele microseconden (T_{J6}).
5. De controller zet CLK op 0V als databit 4 gelijk is aan 1, en laat deze anders naar 5V los. De controller zet DATA op 0V als databit 5 gelijk is aan 1, en laat deze anders naar 5V los.
6. De controller wacht enkele microseconden (T_{J7}).
7. De controller zet CLK op 0V als databit 6 gelijk is aan 1, en laat deze anders naar 5V los. De controller zet DATA op 0V als databit 7 gelijk is aan 1, en laat deze anders naar 5V los.
8. De controller wacht enkele microseconden (T_{J8}).
9. De controller zet CLK op 0V als databit 3 gelijk is aan 1, en laat deze anders naar 5V los. De controller zet DATA op 0V als databit 1 gelijk is aan 1, en laat deze anders naar 5V los.
10. De controller wacht enkele microseconden (T_{J9}).

11. De controller zet CLK op 0V als databit 2 gelijk is aan 1, en laat deze anders naar 5V los. De controller zet DATA op 0V als databit 0 gelijk is aan 1, en laat deze anders naar 5V los.
12. De controller wacht enkele microseconden (T_{J10}).
13. De controller trekt de DATA-lijn naar 0V. Hij laat ook de CLK-lijn naar 5V los, tenzij hij EOI wil aangeven, in welk geval hij de CLK-lijn naar 0V trekt.
14. De controller wacht enkele microseconden, gedurende welke tijd het apparaat de EOI-waarde uitleest, en trekt vervolgens DATA naar 0V. (T_{J11}).
15. De byteoverdracht is voltooid.

Let er bij het verzenden van de eerste byte onder het JiffyDOS™protocol op dat het CLK-signaal op 0V kan staan. De tijd tussen bytes (T_{JBB}) is niet vereist (wordt op nul gezet), omdat de korte ontvanglus op het apparaat eenvoudigweg aangeeft wanneer het gereed is om de volgende byte te ontvangen, wat ook bijdraagt aan de hoge snelheid.

Let ook op dat de onderstaande timing gebaseerd is op het moment waarop de signalen ingesteld moeten worden. Bij software-implementaties moet de latentie van de benodigde processorinstructies in aanmerking worden genomen. Dit vormt geen probleem bij de 4541, omdat deze een latentie van minder dan 50 ns heeft.



Symbool	Min	4541	Max	Omschrijving
T_{JBB}	0	0	∞	Tijd tussen bytes. Er is geen minimale tijd tussen bytes bij het JiffyDOS™-protocol, mits T_{J12} voldoende groot is, omdat dat de latentie van de ontvangroutine effectief verbergt.
T_{J6}	10	10	10	JiffyDOS™ verzenden: insteltijd ontvanger.
T_{J7}	13	13	13	JiffyDOS™ verzenden: vasthoudtijd, eerste di-bit.
T_{J8}	11	11	11	JiffyDOS™ verzenden: vasthoudtijd, tweede di-bit.
T_{J9}	11	11	11	JiffyDOS™ verzenden: vasthoudtijd, derde di-bit.
T_{J10}	12	12	12	JiffyDOS™ verzenden: vasthoudtijd, vierde di-bit.
T_{J11}	15	15	15	JiffyDOS™ verzenden: vasthoudtijd, statusbits. CLK draagt de EOI-melding, en DATA wordt op 0V gezet om een succesvolle verzending aan te geven. De ontvanger kan een framefout aangeven indien DATA 5V wordt waargenomen.
T_{J12}	18	18	18	JiffyDOS™ hersteltijd na verzending. CLK wordt naar 0V getrokken, en DATA wordt op 0V gezet om een succesvolle verzending aan te geven. De ontvanger kan een framefout aangeven indien DATA 5V wordt waargenomen.
T_{JR}	15	15	15	

De overige tijden zijn gelijk aan die bij het verzenden van een byte onder Attention.

Alle tijdseenheden zijn uitgedrukt in microseconden, tenzij anders aangegeven.

Ontvangst van randapparaat naar controller onder JiffyDOS™

In tegenstelling tot het standaard seriële protocol gebruikt het JiffyDOS™-protocol verschillende protocollen voor verzenden en ontvangen. Dit is noodzakelijk bij software-implementaties om de maximale snelheid te bereiken, omdat de seriële randapparatuurbus-lijnen op verschillende registerbits worden gemapt. Een essentieel onderdeel van het JiffyDOS™-protocol is hoe het op intelligente wijze de ontvangen bits manipuleert om het volledige byte in de kortst mogelijke tijd te reconstrueren, en op vergelijkbare wijze bij verzending.

Het resultaat is opmerkelijk: JiffyDOS™ kan een byte verzenden in ongeveer dezelfde tijd die het originele protocol nodig heeft om één bit te verzenden.

Bij verzending via het JiffyDOS™-protocol worden de databits geïnverteerd; bij ontvangst van een randapparaat dat het JiffyDOS™-protocol gebruikt, worden de bits op natuurlijke wijze weergegeven, oftewel met 1 = 5V en 0 = 0V.

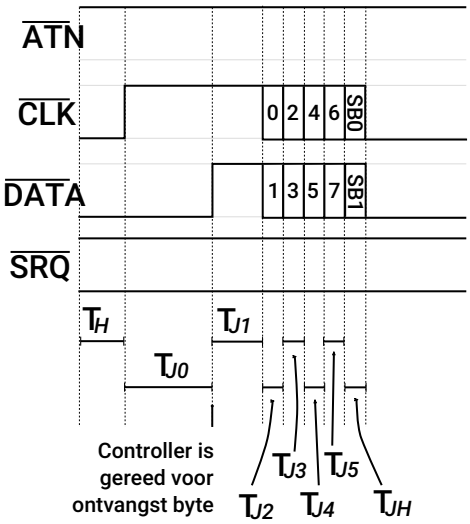
De SRQ-lijn is niet actief in deze transactie.

Ter samenvatting:

1. Het randapparaat kan een willekeurige periode wachten voordat het een byte naar de controller kan verzenden. Wanneer het zover is, laat het de CLK-lijn los naar 5V.
2. Wanneer de controller dit opmerkt, laat hij de DATA-lijn los naar 5V en begint het kritieke tijdsgebied van het protocol.
3. Het randapparaat wacht kort. Voor apparaten met een software-implementatie, oftewel alle Commodore™-stations, is de vertraging een natuurlijk gevolg van de processorinstructie die nodig is om het verzenden van de byte in te luiden; hetzelfde geldt voor de volgende stappen.
4. Het randapparaat plaatst het eerste di-bit van gegevens op de CLK- en DATA-lijnen en wacht kort.
5. Het randapparaat plaatst het tweede di-bit van gegevens op de CLK- en DATA-lijnen en wacht kort.
6. Het randapparaat plaatst het derde di-bit van gegevens op de CLK- en DATA-lijnen en wacht kort.
7. Het randapparaat plaatst het vierde di-bit van gegevens op de CLK- en DATA-lijnen en wacht kort.
8. Het randapparaat plaatst de statusbits van de byte op de CLK- en DATA-lijnen en wacht kort.
9. De controller trekt DATA naar 0V om de ontvangst van de byte te bevestigen en te voorkomen dat het randapparaat nog een byte verzendt voordat hij gereed is.
10. Het randapparaat trekt de CLK-lijn naar 0V indien het niet onmiddellijk nog een byte gereed heeft om te verzenden; in het andere geval blijft de lijn op 5V.

De statusbits hebben de volgende betekenis:

CLK	**DATA**	**Betekenis**
0V	0V	Er is een fout opgetreden.
0V	5V	Byte ontvangen met einde-aanduiding (EOI).
5V	0V	Normale byte ontvangen (geen EOI).
5V	5V	Er is een fout opgetreden.



Symbool	Min	4541	Max	Omschrijving
T_{JH}	10	–	∞	Wachttijd van het randapparaat totdat het gegevens heeft om te verzenden.
T_{J0}	0	–	∞	Wachttijd van de controller totdat deze gereed is om een byte te ontvangen.
T_{J1}	37	37	∞	Deze tijd wordt door software-implementaties gebruikt om de byte voor te bereiden voor onmiddellijke verzending.
T_{J2}	14	14	14	Eerste di-bit van gegevens verzenden
T_{J3}	10	10	10	Tweede di-bit van gegevens verzenden
T_{J4}	11	11	11	Derde di-bit van gegevens verzenden
T_{J5}	11	11	11	Vierde di-bit van gegevens verzenden
T_{JH}	13	13	13	Statusbits verzenden

De overige tijden zijn gelijk aan die bij het verzenden van een byte onder Attention.

Alle tijdseenheden zijn uitgedrukt in microseconden, tenzij anders aangegeven.

Omschakelen van spreker naar luisteraar

Wanneer de buscontroller een apparaat beveelt om te verzenden, wordt de transactie afgesloten met ATN op 0V. De controller houdt de CLK-lijn op 0V, en het apparaat dat

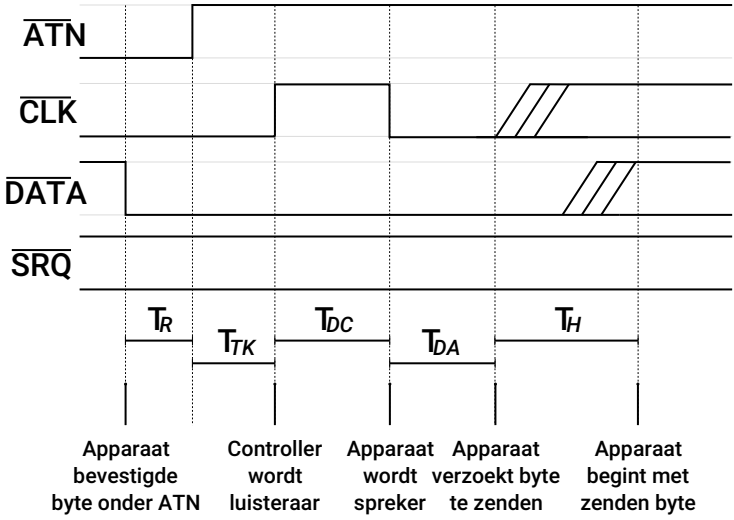
verzond hield de DATA-lijn vast. Deze situatie moet worden omgekeerd: de controller moet de besturing van de bus afstaan en deze overdragen aan het apparaat.

De SRQ-lijn is niet actief bij deze transactie.

Ter samenvatting:

1. De controller wacht lang genoeg voordat hij de ATN-lijn vrijgeeft, om er zeker van te zijn dat het randapparaat de bevestiging van de byte die onder Attention werd verzonden, heeft afgerond. Dit is de vertraging T_R.
2. De controller geeft de ATN-lijn vrij naar 5V.
3. De controller wacht enige tijd. Dit is de vertraging T_{TK}.
4. De controller geeft de CLK-lijn vrij naar 5V en trekt de DATA-lijn naar 0V. Vanaf dit moment heeft de controller de besturing van de bus afgestaan en wordt hij luisteraar.
5. Na een bepaalde tijdsduur trekt het randapparaat dat op het punt staat te spreken, de CLK-lijn naar 0V en geeft de DATA-lijn vrij, die op 0V blijft staan omdat de controller deze nog steeds naar 0V trekt ter voorbereiding op het ontvangen. Vanaf dit moment is het randapparaat de enige spreker op de bus. De vertraging voordat de CLK-lijn door het randapparaat naar 0V wordt getrokken, is de time-out T_{DC}.
6. Het randapparaat wacht enige tijd voordat het met spreken mag beginnen. Dit is de vertraging T_{DA}.

Na voltooiing van dit proces zijn de rollen van controller en randapparaat omgekeerd, waarbij het randapparaat nu de rol van spreker ("**talk**er") heeft en de controller (en mogelijk andere apparaten op de bus) die van luisteraar ("**listen**er").



Symbool	Min	4541	Max	Omschrijving
T_{DA}	4/80*	-	∞	Duur van de bevestiging dat apparaat nu spreker is. De controller houdt CLK.
T_{DC}	0	64 ms	∞	Duur van de zendbevestiging, d.w.z. de tijd die een randapparaat heeft voordat het spreker wordt. *Kan onmiddellijk beginnen zodra CLK op 5V wordt waargenomen.*
T_H	-	N/A	∞	Luisteraar-wachttijd. *De luisteraar kan de ontvangstbevestiging willekeurig lang uitstellen of ook helemaal niet.*
T_R	0/20^	200	-	Tijd van frame tot vrijgave van ATN.
T_{TK}	20	40	100	Vrijgave van de zendvoorwaarde.

Alle tijdseenheden zijn uitgedrukt in microseconden, tenzij anders aangegeven.

* T_{DA} wordt geleverd door het randapparaat, niet door de controller. De 4541 is in staat veel sneller op seriële bus-gebeurtenissen te reageren dan een C64 en heeft derhalve slechts 4 microseconden nodig om te garanderen dat de CLK-lijn de tijd heeft om naar 5V te stijgen. Voor controllers met een softwarematige implementatie van het protocol, zoals de C64, bedraagt het minimum 80 microseconden. Randapparaten dienen derhalve steeds een waarde van ten minste 80 microseconden voor T_{DA} te gebruiken.

^ Het Commodore™ 64 Programmer's Reference Guide vermeldt dat T_R een minimumduur van 20 microseconden heeft. Wij zien geen aanleiding om te veronderstellen dat enige implementatie een vertraging vereist voordat de ATN-lijn kan worden vrijgegeven na het overdragen van een byte.

Byte verzenden met einde-aanduiding (EOI)

Na het verzenden van een stroom gegevens wil een apparaat de luisteraar ("**listen**er") er vaak van op de hoogte brengen dat het einde van de gegevens is bereikt, bijvoorbeeld zodat deze met de verwerking kan beginnen. Dit gebeurt door de byte op een bijzondere manier te markeren met het einde-aanduiding (EOI)-attribuut.

Over de betekenis van de afkorting EOI bestaat spraakverwarring. Op de parallele IEE-488 bus heet één van de pinnen in de aansluiting *end or identify*. Deze pin heeft inderdaad een dubbele functie: Als ATN niet actief is, dan dient de pin als einde-aanduiding (*end*). Is ATN wel actief, dan dient de pin om parallel de status van apparaten op te vragen (*identify*). Op de seriële Commodore-bus is de *identify*-functie niet aanwezig en is EOI te allen tijde een einde-aanduiding. Wellicht hierom wordt EOI, zelfs in officiële Commodore-documenten, vaak beschreven als *end of indicator*. De meest triviale manier om dit te lezen, "einde van (een) aanduiding", is logisch onzinnig. De juiste manier om dit te lezen is "eind van"-aanduiding, de aanduiding van het

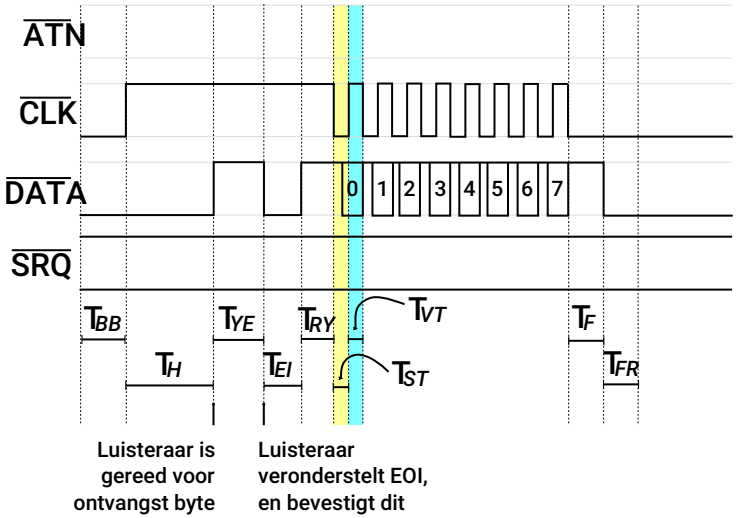
eind van gegevensoverdracht. Dit is weer een mooi voorbeeld waarin begrippen in de Engelse taal voor meerdere uitleg vatbaar zijn. We zullen deze spraakverwarring niet meenemen naar het Nederlands en in de handleiding stelselmatig "einde-aanduiding (EOI)" gebruiken.

De einde-aanduiding wordt gerealiseerd met behulp van een timing-nevenkanaal, zodat er geen afzonderlijke ader in de kabel voor nodig is. Wanneer de spreker de CLK-lijn naar 5V heeft gebracht om aan te geven dat hij een byte wil verzenden, en de luisteraar de DATA-lijn naar 5V heeft gebracht om zijn bereidheid om te ontvangen aan te geven, kan de spreker ten minste 200 microseconden wachten om de luisteraar aan te geven dat de byte met EOI-status moet worden ontvangen. De luisteraar bevestigt dit aan de spreker door de DATA-lijn enige tijd naar 0V te trekken, waarna de verzending vrijwel identiek aan een normale transactie verloopt.

De SRQ-lijn is niet actief bij deze transactie.

Ter samenvatting:

1. De spreker geeft de CLK-lijn vrij naar 5V om aan te geven dat hij een byte wil verzenden.
2. Na een willekeurige tijdsduur geeft de luisteraar de DATA-lijn vrij naar 5V om zijn bereidheid om een byte te ontvangen aan te geven.
3. Normaal zou de spreker op dit punt na een korte tijd beginnen met het verzenden van de bits. Om EOI aan te geven, wacht hij evenwel stilzwijgend totdat de luisteraar doorheeft dat deze byte afwijkend is, hetgeen ertoe leidt dat de luisteraar de DATA-lijn terug naar 0V trekt.
4. De luisteraar houdt de DATA-lijn lang genoeg op 0V om er zeker van te zijn dat de spreker dit heeft opgemerkt, en geeft deze vervolgens weer vrij.
5. Zodra de spreker heeft waargenomen dat de luisteraar de DATA-lijn naar 0V heeft getrokken en weer heeft vrijgegeven, begint hij de byte na een korte vertraging normaal te verzenden.
6. Indien de spreker te lang wacht, neemt de luisteraar aan dat de spreker EOI wilde aangeven zonder daadwerkelijk een byte te verzenden.
7. De verzending van de byte wordt op dezelfde manier afgerond als bij een niet-EOI-byte.



Symbool	Min	4541	Max	Omschrijving
T_{BB}	100	100	∞	Tijd tussen bytes. Sprekers moeten lang genoeg wachten om de luisteraar de tijd te geven het einde van de verzending van de vorige byte te registreren, voordat de volgende byte kan worden verzonden.
T_H	–	N/A	∞	Luisteraar-wachttijd. *De luisteraar kan de ontvangstbevestiging willekeurig lang uitstellen of ook helemaal niet.*
T_{YE}	200	250	∞	EOI-indicatievertraging. De zender moet ten minste 200 microseconden wachten zodat de ontvanger kan herkennen dat een byte met EOI wordt verzonden. Indien de vertraging korter is, herkent de ontvanger de EOI niet.
T_{EI}	60	80	∞	EOI-bevestiging vasthoudtijd. De ontvanger moet de DATA-lijn ten minste 60 microseconden weer op 0V trekken, zodat de zender deze EOI-bevestigingspuls kan detecteren.
T_{RY}	60	80	100	De spreker dient binnen een strak tijdvenster te reageren. Indien hij te vroeg is, is het mogelijk dat de luisteraar zich na bevestiging van de EOI, nog niet heeft voorbereid op ontvangst en kan de synchronisatie verloren gaan. Indien hij te laat is, dan kan de luisteraar concluderen dat de spreker gestopt is met spreken.

Symbool	Min	4541	Max	Omschrijving
T_{ST}	20*	35	∞	Databit-insteltijd. *In de C64 Programmer's Reference Guide aangeduid als T_S.*
T_{VT}	20*	35	∞	Databit-geldigheidsvasthoudtijd. *In de C64 Programmer's Reference Guide aangeduid als T_V.*
T_F	-	1000	1000	Frame-handshake (bevestiging) time-out.
T_{FR}	0/60	0	∞	EOI-frame-bevestiging. De luisteraar trekt DATA na een korte periode naar 0V om de met EOI verzonden byte te bevestigen.

Alle tijdseenheden zijn uitgedrukt in microseconden, tenzij anders aangegeven.

* T_{ST} wordt geleverd door het randapparaat, niet door de controller. De 4541 is in staat veel sneller te reageren op seriële bus-gebeurtenissen dan een C64, en heeft derhalve slechts 4 microseconden nodig om er zeker van te zijn dat de CLK-lijn de tijd heeft gehad naar 5V te stijgen. Voor controllers die een softwarematige implementatie van het protocol gebruiken, zoals de C64, bedraagt het minimum 80 microseconden. Randapparaten dienen derhalve altijd een waarde van ten minste 80 microseconden voor T_{DA} te gebruiken.

^ De Commodore™ 64 Programmer's Reference Guide vermeldt T_{FR} met een minimumduur van 60 microseconden. Wij zien geen enkel bewijs dat er een vertraging nodig zou zijn voordat de DATA-lijn door de luisteraar laag kan worden getrokken om ontvangst van een byte te bevestigen.

Byte ontvangen

Het ontvangen van bytes op de IEC-bus verloopt identiek aan het verzenden. Het belangrijkste verschil is dat de 4541 veel ruimere marginale waarden voor tijdsparameters toelaat dan softwarematige implementaties van het bus-protocol. De 4541 heeft niet meer dan 4 microseconden nodig voor een willekeurige bus-toestand; dit is de tijd die de pull-up-weerstanden op de bus nodig hebben om de lijnen te laten terugkeren naar 5V. Intern is de 4541 in staat bit-tijden korter dan 1 microseconde te detecteren.

OPTIONELE GEÏNTEGREERDE DATALOGGER

De 4541 is verkrijgbaar in een variant die een ingebouwde seriële randapparatuurbus-datalogger bevat. Deze is ontworpen om het oplossen van protocolfouten op deze

bus te vergemakkelijken. Het vastleggen van gegevens begint zodra opdracht $D0 aan de 4541 wordt gegeven, en duurt ongeveer 4 milliseconden.

- **DATALOG0** Het lage byte van de optionele geïntegreerde seriële randapparatuurbus-datalogger. Het schrijven van $00 naar dit register reset het leespunt van de datalogger naar het begin van de opname. Het schrijven van $01 naar dit register verplaatst het leespunt van de datalogger met één tijdstap. De tijdstappen bedragen ongeveer 1 microseconde, met extra tijdstappen wanneer een van de signalen op de seriële perifere bus van toestand verandert.
- **DATALOG1** Het hoge byte van de optionele geïntegreerde seriële randapparatuurbus-datalogger. Verschillende velden kunnen op dit adres worden gelezen door de volgende waarden naar het DATALOG0-register te schrijven:
  - $02 - Lees lage byte van het statemachine-toestandsnummer van de IEC-controller.
  - $03 - Lees hoge byte van het statemachine-toestandsnummer van de IEC-controller.
  - $04 - Lees lage byte van het aantal cycli dat de vorige bustoestand is vastgehouden.
  - $05 - Lees hoge byte van het aantal cycli dat de vorige bustoestand is vastgehouden.

Elke tijdstap legt 16 bits informatie vast over de seriële perifere bus:

- **DATALOG0 Bit 0** DATA-lijn invoerwaarde
- **DATALOG0 Bit 1** CLK-lijn invoerwaarde
- **DATALOG0 Bit 2** SRQ-lijn invoerwaarde
- **DATALOG0 Bit 3** DATA-lijn uitvoertoestand
- **DATALOG0 Bit 4** CLK-lijn uitvoertoestand
- **DATALOG0 Bit 5** SRQ-lijn uitvoertoestand
- **DATALOG0 Bit 6** ATN-lijn uitvoertoestand
- **DATALOG0 Bit 7** RESET-lijn uitvoertoestand
- **DATALOG1** Toegang tot de aanvullende datalogger-velden.

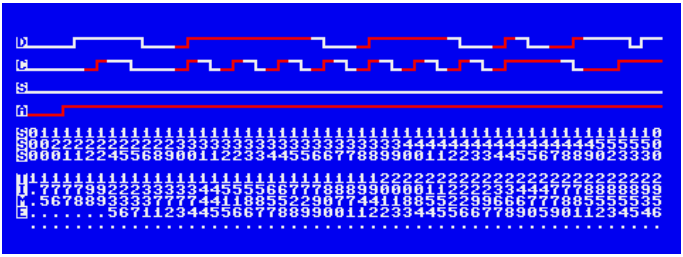
De invoerwaarden zijn de spanningen die de 4541 op de betreffende pinnen uitleest. Los daarvan legt de datalogger vast of de 4541 een van die lijnen laag trekt. Dit maakt het mogelijk te bepalen of een aangesloten randapparaat of de 4541 een bepaald signaal laag trekt. Dit biedt veel meer informatie dan het simpelweg signalen op de seriële randapparatuurbus meten, waarbij niet eenvoudig is vast te stellen wie een bepaalde lijn laag trekt.

De volgende waarheidstabel legt uit hoe deze signalen moeten worden geïnterpreteerd:

Invoerwaarde	Uitvoerwaarde	Betekenis
1	1	Het signaal zweeft op 5V. Geen enkel apparaat trekt het laag.
0	1	Het signaal staat op 0V. De 4541 trekt het niet laag, ofwel een aangesloten randapparaat, ofwel de CIA trekt het laag.
0	0	Het signaal staat op 0V. De 4541 trekt het laag. Andere apparaten kunnen het ook laag trekken, maar tussen deze twee situaties kan niet worden onderscheiden.
1	0	Het signaal zweeft op 5V, maar de 4541 trekt het naar 0V. Uw computer is waarschijnlijk defect, of iemand heeft 5V op het signaal aangesloten zonder stroombeperkende weerstand ertussen; in dat geval is uw computer op het punt defect te raken.

Gegevens uit de datalogger halen

Met het volgende programma wordt de inhoud van de datalogger opgehaald en als PETSCII-golfvorm weergegeven, zoals de hier getoonde:



```
10 WK=$50000
20 NL=PEEK($ED)
30 FETCH 3, 4096*2, 0, WK
40 PRINT"VDUMPING..."
50 POKE$D694,0
60 FORI=0 TO 4095*2 STEP 2
70 POKEWK+I,PEEK($D694):POKE WK+1+I,PEEK($D695)
80 POKE$D694,1
90 NEXT I
100 PRINT"VPROCESSING..."
110 PRINT"VRDQQ|CQQ|SQQ|AQQ|SQ|SQ|SQQ|TQ|IQ|MQ|EOOOOOOOOOOOOOOOOO_";
120 PV%=0:PS%=0:C%=0:DV=1:CV=1:SV=1:AV=1
130 FORD=0 TO 4095*2 STEP 2
140 V%=PEEK(WK+D):S%=PEEK(WK+D+1)
150 IF V%=PV% AND S%=PS% THEN GOTO 460
160 C%=C%+1
170 DD=C%>1 AND DV<>(V%AND$01) : DV=(V% AND $01)
180 CD=C%>1 AND CV<>(V%AND$02) : CV=(V% AND $02)
190 SD=C%>1 AND SV<>(V%AND$04) : SV=(V% AND $04)
200 AD=C%>1 AND AV<>(V%AND$40) : AV=(V% AND $40)
210 IFDV=0THEN BEGIN:IFDD=0 THEN D$="‾":ELSE D$="┌":BEND
220 IFDV>0THEN BEGIN:IFDD=0 THEN D$="_":ELSE D$="└":BEND
230 IFCV=0THEN BEGIN:IFCD=0 THEN C$="‾":ELSE C$="┌":BEND
240 IFCV>0THEN BEGIN:IFCD=0 THEN C$="_":ELSE C$="└":BEND
250 IFSV=0THEN BEGIN:IFSD=0 THEN S$="‾":ELSE S$="┌":BEND
260 IFSV>0THEN BEGIN:IFSD=0 THEN S$="_":ELSE S$="└":BEND
270 IFAV=0THEN BEGIN:IFAD=0 THEN A$="‾":ELSE A$="┌":BEND
280 IFAV>0THEN BEGIN:IFAD=0 THEN A$="_":ELSE A$="└":BEND
290 PRINTCHR$(27);"K";
300 IF(V% AND $08)=0 THEN PRINT"2"; :ELSE PRINT"E";
310 PRINT D$
320 PRINT"":PRINTCHR$(27);"K";
330 IF(V% AND $10)=0 THEN PRINT"2"; :ELSE PRINT"E";
340 PRINT C$
350 PRINT"":PRINTCHR$(27);"K";
360 IF(V% AND $20)=0 THEN PRINT"2"; :ELSE PRINT"E";
370 PRINT S$
380 PRINT"":PRINTCHR$(27);"K";
390 PRINT"2";
```

```
400 PRINT A$;"■■";
410 GOSUB 550:GOSUB 490
420 PRINT"■■■■■■■■■■■■■■■■";
430 PS%=S%
440 PV%=V%
450 IF C%=77 THEN  0,18:C%=0
460 NEXT D
470 PRINT"■";:FORL=0TONL-3:PRINT"■";:NEXTL
480 END
490 N$=LEFT$(STR$(D/2)+".........",6)
500 PRINT"■";
510 FORI=1TO6
520 PRINTCHR$(27);"K";MID$(N$,I,1)
530 NEXT I
540 RETURN
550 S$=MID$(STR$(S%),2,LEN(STR$(S%))):N$=RIGHT$("000"+S$,3)
560 PRINT"■";
570 FORI=1TO3
580 PRINTCHR$(27);"K";MID$(N$,I,1)
590 NEXT I
600 RETURN
```

HOOFDSTUK 10

Registers van de real-time klok (RTC)

- Algemene RTC-registers (alle revisies)
- Uitgebreide RTC-registers (moederbord R4 en nieuwer)
- Registers op de Grove-RTC-vervangingsunit

ALGEMENE RTC-REGISTERS (ALLE REVISIES)

HEX	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
FFD7110	RTCSEC							
FFD7111	RTCMIN							
FFD7112	Onge-bruikt	RTCHOUR						
FFD7113	RTCDAY							
FFD7114	RTCMONTH							
FFD7115	RTCYEAR							
FFD7116	RTCDOW							
FFD7140 – FFD717F	NVRAM							

- **NVRAM** 64 byte niet-vluchtig RAM; kan worden gebruikt om de machineconfiguratie op te slaan
- **RTCDAY** Waarde die de dag van de maand aangeeft in de realtimeklok (binair gecodeerd decimaal)
- **RTCDOW** Waarde die de dag van de week aangeeft in de realtimeklok (0-6; binair gecodeerd decimaal)
- **RTCHOUR** Waarde die het uur aangeeft in de realtimeklok (binair gecodeerd decimaal)
- **RTCHOUR24EN** Schakelt de 24-uursmodus van de realtimeklok in
- **RTCHOURPM** PM-indicator van de realtimeklok (alleen in de 12-uursmodus)
- **RTCMIN** Waarde die de minuten aangeeft in de realtimeklok (binair gecodeerd decimaal)
- **RTCMONTH** Waarde die de maand aangeeft in de realtimeklok (binair gecodeerd decimaal)
- **RTCSEC** Waarde die de seconden aangeeft in de realtimeklok (binair gecodeerd decimaal)
- **RTCYEAR** Waarde die het jaar aangeeft in de realtimeklok (binair gecodeerd decimaal)
- **Ongebruikt** Ongebruikt; altijd 1

UITGEBREIDE RTC-REGISTERS (MOEDERBORD R4 EN NIEUWER)

Deze moederborden bevatten de RV-3032-C7, welke enkele extra registers biedt. Lees de documentatie voor deze RTC-chip zorgvuldig door voordat u deze registers wijzigt!

HEX	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
FFD7116	RTCDOW							
FFD7117	100THSEC							
FFD7118	AE_M	ALMINUTES						
FFD7119	AE_H	-	ALHOURS					
FFD711A	AE_D	-	ALDAY					
FFD711B	PCTCR0							
FFD711C	-				PCTCR1			
FFD711D	THF	TLF	UF	TF	AF	EVF	PORF	VLF
FFD711E	TEMPL				EEF	EEbusy	CLKF	BSF
FFD711F	TEMPH							
FFD7120	Onge-bruikt	Onge-bruikt	GP0	USEL	TE	EERD	TD	
FFD7121	-	CLKIE	UIE	TIE	AIE	EIE	GP1	STOP
FFD7122	Ongebruikt			BSIE	THE	TLE	THIE	TLIE
FFD7123	Ongebruikt		EVR	THR	TLR	EVOW	THOW	TLOW
FFD7124	CLKD	INTDE	CEIE	CAIE	CTIE	CUIE	CTHIE	CTLIE
FFD7125	CLKDE	EHL	ET		Ongebruikt			ESYN
FFD7126	TLT							
FFD7127	THT							
FFD7128	TSTLowCount							
FFD7129	-	TSTLowSec						
FFD712A	-	TSTLowMin						
FFD712B	-	TSTLowHrs						
FFD712C	-	TSTLowDat						
FFD712D	-	TSTLowMonth						
FFD712E	TSTLowYear							
FFD712F	TSTHighCount							
FFD7130	-	TSTHighSec						
FFD7131	-	TSTHighMin						
FFD7132	-	TSTHighHrs						
FFD7133	-	TSTHighDat						
FFD7134	-	TSTHighMonth						
FFD7135	TSTHighYear							
FFD7136	TSEVICount							

vervolgd ...

...vervolgd

HEX	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
FFD7137	TSEVI100th							
FFD7138	–	TSEVISec						
FFD7139	–	TSEVIMin						
FFD713A	–		TSEVIHrs					
FFD713B	–		TSEVIDat					
FFD713C	–			TSEVIMonth				
FFD713D	TSEVIYear							

- **100THSEC** Honderdsten van seconden van de realtimeklok
- **AE_D** Inschakelbit van het alarm (samen met AE_M en AE_H)
- **AE_H** Inschakelbit van het alarm (samen met AE_M en AE_D)
- **AE_M** Inschakelbit van het minuten-alarm; schakelt het alarm in samen met AE_H en AE_D
- **AF** Statusregister: 1 geeft het optreden van een alarminterrupt aan
- **AIE** Inschakelbit van het alarminterrupt
- **ALDAY** Datumwaarde van het alarm, 01-31 (binair gecodeerd decimaal)
- **ALHOURS** Uurwaarde van het alarm, 00-23 (binair gecodeerd decimaal)
- **ALMINUTES** Minutenwaarde van het alarm, 00-59 (binair gecodeerd decimaal)
- **BSF** 1 geeft aan dat een omschakeling van de hoofdvoeding naar de back-up heeft plaatsgevonden
- **BSIE** Inschakelbit van het interrupt voor de back-upomschakeling
- **CAIE** Klokuitvoer wanneer de alarm-interrupt is ingeschakeld
- **CEIE** Klokuitvoer wanneer de EVI-interrupt is ingeschakeld
- **CLKD** Inschakelbit voor de uitschakelvertragingswaarde van CLKOUT na I2C STOP
- **CLKDE** Inschakelbit van de uitschakelvertraging van CLKOUT na I2C-STOP
- **CLKF** 1 geeft het optreden van een interruptgestuurde klokuitvoer op de CLKOUT-pin aan
- **CLKIE** Inschakelbit van de interruptgestuurde klokuitvoer
- **CTHIE** Klokuitvoer wanneer de THigh-interrupt is ingeschakeld
- **CTIE** Klokuitvoer wanneer de interrupt van de periodieke afteltimer is ingeschakeld
- **CTLIE** Klokuitvoer wanneer de TLow-interrupt is ingeschakeld

- **CUIE** Klokuitvoer wanneer de Periodic Time Update-interrupt is ingeschakeld
- **EEF** 1 geeft aan dat de EEPROM-schrijftoegang is mislukt
- **EERD** Bit voor het uitschakelen van de EEPROM-geheugenverversing
- **EEbusy** 1 geeft aan dat de EEPROM momenteel bezig is en verdere commando's negeert
- **EHL** Selectie van het hoge/lage niveau van de gebeurtenis (stijgende/dalende flank) voor detectie
- **EIE** Inschakelbit voor de interrupt van een externe gebeurtenis
- **ESYN** Synchronisatiebit voor een externe gebeurtenis (EVI)
- **ET** Instelling van de filtertijd voor gebeurtenissen
- **EVF** Statusregister: 1 geeft aan dat een externe gebeurtenis is opgetreden
- **EVOW** Overschrijfbit van de EVI-tijdstempel
- **EVR** Resetbit van de EVI-tijdstempel
- **GP0** Registerbit voor algemeen gebruik
- **GP1** Registerbit voor algemeen gebruik
- **INTDE** Inschakelbit voor interruptvertraging na het inschakelen van CLKOUT
- **PCTCR0** Besturingsregister van periodieke afteltimer 0 (lage byte)
- **PCTCR1** Besturingsregister van periodieke afteltimer 1 (hoge byte)
- **PORF** Statusregister: 1 geeft aan dat gegevens ongeldig zijn vanwege een stroomprobleem; registers moeten opnieuw worden geïnitialiseerd
- **RTCDOW** Waarde van de dag van de week, 0-6 (binair gecodeerd decimaal)
- **STOP** 1 stopt de timer en herstelt de klokvoorscalerfrequenties
- **TD** Selectie van de timerklokfrequentie
- **TE** Inschakelbit van de periodieke afteltimer
- **TEMPH** Geheel deel van de temperatuurwaarde [11.0] in tweecomplementformaat
- **TEMPL** Fractiedeel van de temperatuurwaarde [11:0] in tweecomplementformaat
- **TF** Statusregister: 1 geeft het optreden van een interrupt van de periodieke afteltimer aan
- **THE** Inschakelbit voor hoge temperatuur
- **THF** Statusregister: 1 geeft aan dat de temperatuur de hoge drempelwaarde THT heeft overschreden

- **THIE** Inschakelbit voor de hoge-temperatuurinterrupt
- **THOW** Overschrijfbit van de THigh-tijdstempel
- **THR** Resetbit van de THigh-tijdstempel
- **THT** Temperatuurdrempelregister: hoge drempelwaarde (THigh)
- **TIE** Inschakelbit voor de interrupt van de periodieke afteltimer
- **TLE** Inschakelbit voor lage temperatuur
- **TLF** Statusregister: 1 geeft aan dat de temperatuur onder de lage drempelwaarde TLT is gezakt
- **TLIE** Inschakelbit voor de lage-temperatuurinterrupt
- **TLOW** Bit voor het overschrijven van TLow in de tijdstempel.
- **TLR** Bit voor het resetten van TLow in de tijdstempel.
- **TLT** Temperatuurdrempelregister: lage drempelwaarde (TLow)
- **TSEVI100th** Tijdstempelevenementregister: TS EVI honderdsten van seconden
- **TSEVICount** Tijdstempelevenementregister: TS EVI-telling
- **TSEVIDat** Tijdstempel-THigh-register: TS TLow-datum
- **TSEVIHrs** Tijdstempel-THigh-register: TS TLow-uren
- **TSEVIMin** Tijdstempel-THigh-register: TS TLow-minuten
- **TSEVIMonth** Tijdstempel-THigh-register: TS TLow-maand
- **TSEVISec** Tijdstempel-THigh-register: TS TLow-seconden
- **TSEVIYear** Tijdstempel-THigh-register: TS TLow-jaar
- **TSTHighCount** Tijdstempel-THigh-register: TS TLow-telling
- **TSTHighDat** Tijdstempel-THigh-register: TS TLow-datum
- **TSTHighHrs** Tijdstempel-THigh-register: TS TLow-uren
- **TSTHighMin** Tijdstempel-THigh-register: TS TLow-minuten
- **TSTHighMonth** Tijdstempel-THigh-register: TS TLow-maand
- **TSTHighSec** Tijdstempel-THigh-register: TS TLow-seconden
- **TSTHighYear** Tijdstempel-THigh-register: TS TLow-jaar
- **TSTLowCount** Tijdstempel-TLow-register: TS TLow-telling
- **TSTLowDat** Tijdstempel-TLow-register: TS TLow-datum
- **TSTLowHrs** Tijdstempel-TLow-register: TS TLow-uren
- **TSTLowMin** Tijdstempel-TLow-register: TS TLow-minuten

- **TSTLowMonth** Tijdstempel-TLow-register: TS TLow-maand
- **TSTLowSec** Tijdstempel-TLow-register: TS TLow-seconden
- **TSTLowYear** Tijdstempel-TLow-register: TS TLow-jaar
- **UF** Statusregister: 1 geeft aan dat er een periodieke timer-update-interruptgebeurtenis heeft plaatsgevonden.
- **UIE** Bit voor het inschakelen van de periodieke tijdupdate-interrupt.
- **USEL** Bit voor het selecteren van de update-interrupt
- **Ongebruikt** Niet geïmplementeerd. Geeft altijd 0 terug
- **VLF** Statusregister: 1 geeft aan dat er ongeldige gegevens zijn door een voedingsprobleem. De registers moeten opnieuw worden geïnitialiseerd.

REGISTERS OP DE GROVE-RTC-VERVANGINGSUNIT

Deze registers zijn beschikbaar indien een Grove-vervangingsunit is geïnstalleerd. Dit onderdeel is alleen nodig indien uw RTC defect is. Dit heeft alleen betrekking op de eerste R3-moederborden die werden verzonden.

HEX	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
FFD7400	EXTRTCSEC							
FFD7401	EXTRTCMIN							
FFD7402	-	EXTRTC-HOUR12EN	EXTRTC-HOURPM	EXTRTCHOUR				
FFD7403	EXTRTCDOW							
FFD7404	EXTRTCDAY							
FFD7405	EXTRTC-CENTURY	-		EXTRTCMONTH				
FFD7406	EXTRTCYEAR							
FFD7407	EXTRTC-A1M1	EXTRTCA1SEC						
FFD7408	EXTRTC-A1M2	EXTRTCA1MIN						
FFD7409	EXTRTC-A1M3	EXTRTC-A1HOUR12EN	EXTRTC-A1HOURPM	EXTRTCA1HOUR				
FFD740A	EXTRTC-A1M4	EXTRTC-A1DOWSEL	EXTRTCA1DATE					
FFD740B	EXTRTC-A2M2	EXTRTCA2MIN						
FFD740C	EXTRTC-A2M3	EXTRTC-A2HOUR12EN	EXTRTC-A2HOURPM	EXTRTCA2HOUR				

vervolgd ...

...vervolgd

HEX	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
FFD740D	EXTRTC-A2M4	EXTRTC-A2DOWSEL	EXTRTCA2DATE					
FFD740E	EXTRTCCTRL							
FFD740F	EXTRTCST							
FFD7410	EXTRTCAGINGOFS							
FFD7411	EXTRTCTEMPMSB							
FFD7412	EXTRTCTEMPMSB							

- **EXTRTCA1DATE** Datumwaarde van alarm 1 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA1DOW** Waarde die de dag van de week aangeeft voor alarm 1 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA1DOWSEL** Selecteert de overeenkomst van de dag van de week voor alarm 1 van de externe realtimeklok
- **EXTRTCA1HOUR** Uurwaarde van alarm 1 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA1HOUR12EN** Schakelt de 12-uursmodus van alarm 1 van de externe realtimeklok in
- **EXTRTCA1HOURPM** PM-indicator van het uur van alarm 1 van de externe realtimeklok
- **EXTRTCA1M1** Maskerbit 1 van alarm 1 van de externe realtimeklok
- **EXTRTCA1M2** Maskerbit 2 van alarm 1 van de externe realtimeklok
- **EXTRTCA1M3** Maskerbit 3 van alarm 1 van de externe realtimeklok
- **EXTRTCA1M4** Maskerbit 4 van alarm 1 van de externe realtimeklok
- **EXTRTCA1MIN** Minuutwaarde van alarm 1 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA1SEC** Secondenwaarde van alarm 1 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA2DATE** Datumwaarde van alarm 2 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA2DOW** Waarde die de dag van de week aangeeft voor alarm 2 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCA2DOWSEL** Selecteert de overeenkomst van de dag van de week voor alarm 2 van de externe realtimeklok
- **EXTRTCA2HOUR** Uurwaarde van alarm 2 van de externe realtimeklok (binair gecodeerd decimaal)

- **EXTRTCA2HOUR12EN** Schakelt de 12-uursmodus van alarm 2 van de externe realtimeklok in
- **EXTRTCA2HOURPM** PM-indicator van het uur van alarm 2 van de externe realtimeklok
- **EXTRTCA2M2** Maskerbit 2 van alarm 2 van de externe realtimeklok
- **EXTRTCA2M3** Maskerbit 3 van alarm 2 van de externe realtimeklok
- **EXTRTCA2M4** Maskerbit 4 van alarm 2 van de externe realtimeklok
- **EXTRTCA2MIN** Minuutwaarde van alarm 2 van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCAGINGOFS** Ouderdomscompensatie van de externe realtimeklok (niet wijzigen!)
- **EXTRTCCENTURY** Jaaroverdracht van de externe realtimeklok
- **EXTRTCCTRL** Besturing van de externe realtimeklok
- **EXTRTCDAY** Waarde die de dag van de maand aangeeft in de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCDOW** Waarde die de dag van de week aangeeft in de externe realtimeklok (1-7; binair gecodeerd decimaal)
- **EXTRTCHOUR** Uurwaarde van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCHOUR12EN** Schakelt de 12-uursmodus van de externe realtimeklok in
- **EXTRTCHOURPM** PM-indicator van de externe realtimeklok (alleen 12-uursmodus)
- **EXTRTCMIN** Minuutwaarde van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCMONTH** Maandwaarde van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCSEC** Secondenwaarde van de externe realtimeklok (binair gecodeerd decimaal)
- **EXTRTCST** Besturings- en statusregister van de externe realtimeklok
- **EXTRTCTEMPMSB** Temperatuur van de externe realtimeklok (MSB)
- **EXTRTCYEAR** Jaarwaarde van de externe realtimeklok (binair gecodeerd decimaal)

HOOFDSTUK 11

Speciale registers

- UUID64

UUID64

HEX	Signaal	Omschrijving
FFD7100 - FFD7107	UUID64	64-bits UUID. Kan worden gebruikt om het Ethernet-MAC-adres in te zaaien

HOOFDSTUK 12

Naslagtabellen

- Opslageenheden
- Getalstelselomrekening

OPSLAGEENHEDEN

Eenheid	Is gelijk aan	Afkorting
1 Bit		
1 Nibble	4 Bits	
1 Byte	8 bits	B
1 Kilobyte	1024 B	KB
1 Megabyte	1024KB of 1048576 B	MB

GETALSTELSELOMREKENING

Decimaal	Binair	Hexadecimaal
0	%0	$0
1	%1	$1
2	%10	$2
3	%11	$3
4	%100	$4
5	%101	$5
6	%110	$6
7	%111	$7
8	%1000	$8
9	%1001	$9
10	%1010	$A
11	%1011	$B
12	%1100	$C
13	%1101	$D
14	%1110	$E
15	%1111	$F
16	%10000	$10
17	%10001	$11
18	%10010	$12
19	%10011	$13
20	%10100	$14
21	%10101	$15
22	%10110	$16
23	%10111	$17
24	%11000	$18
25	%11001	$19
26	%11010	$1A
27	%11011	$1B
28	%11100	$1C
29	%11101	$1D
30	%11110	$1E
31	%11111	$1F

Decimaal	Binair	Hexadecimaal
32	%100000	$20
33	%100001	$21
34	%100010	$22
35	%100011	$23
36	%100100	$24
37	%100101	$25
38	%100110	$26
39	%100111	$27
40	%101000	$28
41	%101001	$29
42	%101010	$2A
43	%101011	$2B
44	%101100	$2C
45	%101101	$2D
46	%101110	$2E
47	%101111	$2F
48	%110000	$30
49	%110001	$31
50	%110010	$32
51	%110011	$33
52	%110100	$34
53	%110101	$35
54	%110110	$36
55	%110111	$37
56	%111000	$38
57	%111001	$39
58	%111010	$3A
59	%111011	$3B
60	%111100	$3C
61	%111101	$3D
62	%111110	$3E
63	%111111	$3F

Decimaal	Binair	Hexadecimaal
64	%1000000	$40
65	%1000001	$41
66	%1000010	$42
67	%1000011	$43
68	%1000100	$44
69	%1000101	$45
70	%1000110	$46
71	%1000111	$47
72	%1001000	$48
73	%1001001	$49
74	%1001010	$4A
75	%1001011	$4B
76	%1001100	$4C
77	%1001101	$4D
78	%1001110	$4E
79	%1001111	$4F
80	%1010000	$50
81	%1010001	$51
82	%1010010	$52
83	%1010011	$53
84	%1010100	$54
85	%1010101	$55
86	%1010110	$56
87	%1010111	$57
88	%1011000	$58
89	%1011001	$59
90	%1011010	$5A
91	%1011011	$5B
92	%1011100	$5C
93	%1011101	$5D
94	%1011110	$5E
95	%1011111	$5F

Decimaal	Binair	Hexadecimaal
96	%1100000	$60
97	%1100001	$61
98	%1100010	$62
99	%1100011	$63
100	%1100100	$64
101	%1100101	$65
102	%1100110	$66
103	%1100111	$67
104	%1101000	$68
105	%1101001	$69
106	%1101010	$6A
107	%1101011	$6B
108	%1101100	$6C
109	%1101101	$6D
110	%1101110	$6E
111	%1101111	$6F
112	%1110000	$70
113	%1110001	$71
114	%1110010	$72
115	%1110011	$73
116	%1110100	$74
117	%1110101	$75
118	%1110110	$76
119	%1110111	$77
120	%1111000	$78
121	%1111001	$79
122	%1111010	$7A
123	%1111011	$7B
124	%1111100	$7C
125	%1111101	$7D
126	%1111110	$7E
127	%1111111	$7F

Decimaal	Binair	Hexadecimaal
128	%10000000	$80
129	%10000001	$81
130	%10000010	$82
131	%10000011	$83
132	%10000100	$84
133	%10000101	$85
134	%10000110	$86
135	%10000111	$87
136	%10001000	$88
137	%10001001	$89
138	%10001010	$8A
139	%10001011	$8B
140	%10001100	$8C
141	%10001101	$8D
142	%10001110	$8E
143	%10001111	$8F
144	%10010000	$90
145	%10010001	$91
146	%10010010	$92
147	%10010011	$93
148	%10010100	$94
149	%10010101	$95
150	%10010110	$96
151	%10010111	$97
152	%10011000	$98
153	%10011001	$99
154	%10011010	$9A
155	%10011011	$9B
156	%10011100	$9C
157	%10011101	$9D
158	%10011110	$9E
159	%10011111	$9F

Decimaal	Binair	Hexadecimaal
160	%10100000	$A0
161	%10100001	$A1
162	%10100010	$A2
163	%10100011	$A3
164	%10100100	$A4
165	%10100101	$A5
166	%10100110	$A6
167	%10100111	$A7
168	%10101000	$A8
169	%10101001	$A9
170	%10101010	$AA
171	%10101011	$AB
172	%10101100	$AC
173	%10101101	$AD
174	%10101110	$AE
175	%10101111	$AF
176	%10110000	$B0
177	%10110001	$B1
178	%10110010	$B2
179	%10110011	$B3
180	%10110100	$B4
181	%10110101	$B5
182	%10110110	$B6
183	%10110111	$B7
184	%10111000	$B8
185	%10111001	$B9
186	%10111010	$BA
187	%10111011	$BB
188	%10111100	$BC
189	%10111101	$BD
190	%10111110	$BE
191	%10111111	$BF

Decimaal	Binair	Hexadecimaal
192	%11000000	$C0
193	%11000001	$C1
194	%11000010	$C2
195	%11000011	$C3
196	%11000100	$C4
197	%11000101	$C5
198	%11000110	$C6
199	%11000111	$C7
200	%11001000	$C8
201	%11001001	$C9
202	%11001010	$CA
203	%11001011	$CB
204	%11001100	$CC
205	%11001101	$CD
206	%11001110	$CE
207	%11001111	$CF
208	%11010000	$D0
209	%11010001	$D1
210	%11010010	$D2
211	%11010011	$D3
212	%11010100	$D4
213	%11010101	$D5
214	%11010110	$D6
215	%11010111	$D7
216	%11011000	$D8
217	%11011001	$D9
218	%11011010	$DA
219	%11011011	$DB
220	%11011100	$DC
221	%11011101	$DD
222	%11011110	$DE
223	%11011111	$DF

Decimaal	Binair	Hexadecimaal
224	%11100000	$E0
225	%11100001	$E1
226	%11100010	$E2
227	%11100011	$E3
228	%11100100	$E4
229	%11100101	$E5
230	%11100110	$E6
231	%11100111	$E7
232	%11101000	$E8
233	%11101001	$E9
234	%11101010	$EA
235	%11101011	$EB
236	%11101100	$EC
237	%11101101	$ED
238	%11101110	$EE
239	%11101111	$EF
240	%11110000	$F0
241	%11110001	$F1
242	%11110010	$F2
243	%11110011	$F3
244	%11110100	$F4
245	%11110101	$F5
246	%11110110	$F6
247	%11110111	$F7
248	%11111000	$F8
249	%11111001	$F9
250	%11111010	$FA
251	%11111011	$FB
252	%11111100	$FC
253	%11111101	$FD
254	%11111110	$FE
255	%11111111	$FF

HOOFDSTUK 13

Steuners & donateurs

- Organisaties
- Bijdragers
- Steuners

De MEGA65 zou niet tot stand zijn gekomen zonder de gulle steun van veel organisaties en individuen.

Wij stellen deze lijsten nog steeds samen; onze excuses indien u er nog niet in voorkomt. Indien u iemand kent die wij over het hoofd hebben gezien, laat het ons dan weten. Zo kunnen wij de bijdrage erkennen van iedereen die de MEGA65 mogelijk heeft gemaakt en die er het geweldige retrocomputingproject van heeft gemaakt dat de computer inmiddels is.

ORGANISATIES

Het MEGA Museum of Electronic Games & Art e.V. Duitsland
ALLES

Trenz Electronic, Duitsland
MOEDERBORD **FABRICAGE** **VERKOOP**

Hintsteiner, Oostenrijk
BEHUIZING

GMK, Duitsland
TOETSENBORD

KEVAG Telekom, Duitsland
WEBHOSTING

BIJDRAGERS

Andreas Liebeskind
(libi in paradize)
Financieel directeur MEGA eV

Thomas Hertzler
(grumpyninja)
Woordvoerder voor de VS

Russell Peake
(rdpeake)
Bugherder

Alexander Nik Petra
(n0d)
Vroeg ontwerp van de behuizing

Ralph Egas
(0-limits)
Zakelijk adviseur

Lucas Moss
PCB-ontwerp voor de MEGAphone

Daren Klamer
(Impakt)
Proeflezen van de handleiding

Daniël Mantione
(dmantione)
C64-hardwareguru

Dr. Canan Hastik
(indica)
Voorzitter MEGA eV

Simon Jameson
(Shallan)
Platformverbeteringen

Stephan Kleinert
(ubik)
Vernietiger van BASIC 10

Wayne Johnson
(sausage)
Aanvullingen op de handleiding

Raven L. A. Kleiss
(LAK132)
MegaWAT-presentatiesoftware

Maurice van Gils
(Maurice)
BASIC 65-voorbeeldprogramma's

Andrew Owen
(Cheveron)
Toetsenbord, Sinclair-ondersteuning

Adam Barnes
(amb5l)
HDMI-expert en printplaatherziening

Wayne Rittimann, Jr.
(johnwayner)
Bugverplettering op alle niveaus

STEUNERS

3c74ce64
8-Bit Classics
@11110110100
Aaron Smith
Achim Mrotzek
Adolf Nefischer
Adrian Esdaile
Adrien Guichard
Ahmed Kablaoui
Alan Bastian Witkowski
Alan Field
Alastair Paulin-Campbell
Alberto Mercuri
Alexander Haering
Alexander Kaufmann
Alexander Niedermeier
Alexander Soppart
Alfonso Ardire
Amiga On The Lake
André Kudra
André Simeit
André Wösten
Andrea Farolfi
Andrea Minutello
Andreas Behr
Andreas Freier
Andreas Grabski
Andreas Millinger
Andreas Nopper
Andreas Ochs
Andreas Wendel Manufaktur
Andreas Zschunke
Andrew Bingham
Andrew Dixon
Andrew Mondt
Andrzej Hłuchyj
Andrzej Sawiniec
Andrzej Śliwa
Anthony W. Leal
Arkadiusz Bronowicki
Arkadiusz Kwasny
Arnaud Léandre
Arne Drews
Arne Neumann
Arne Richard Tyarks
Axel Klahr
Balaz Ondrej
Barry Thompson
Bartol Filipovic
Benjamin Maas
Bernard Alaiz
Bernhard Zorn
Bieno Marti-Braitmaier
Bigby
Bill LaGrue
Bjoerg Stojalowski
Björn Johannesson
Bjørn Melbøe
Bo Goeran Kvamme
Boerge Noest
Bolko Beutner
Brett Hallen
Brian Gajewski
Brian Green
Brian Juul Nielsen
Brian Reiter
Bryan Pope
Burkhard Franke
Byron Goodman
Cameron Roberton (KONG)
Carl Angervall
Carl Danowski
Carl Stock
Carl Wall
Carlo Pastore
Carlos Silva
Carsten Sørensen
Cenk Miroglu Miroglu
Chang sik Park
Charles A. Hutchins Jr.
Chris Guthrey
Chris Hooper
Chris Stringer
Christian Boettcher
Christian Eick
Christian Gleinser
Christian Gräfe
Christian Heffner
Christian Kersting
Christian Schiller
Christian Streck
Christian Weyer
Christian Wyk
Christoph Haug
Christoph Huck
Christoph Pross
Christopher Christopher
Christopher Kalk
Christopher Kohlert
Christopher Nelson
Christopher Taylor
Christopher Whillock
Claudio Piccinini
Claus Skrepek
Collen Blijenberg
Constantine Lignos
Crnjaninja
Daniel Auger
Daniel Julien
Daniel Lobitz
Daniel O'Connor
Daniel Teicher
Daniel Tootill
Daniel Wedin
Daniele Benetti
Daniele Gaetano Capursi
Dariusz Szczesniak
Darrell Westbury
David Asenjo Raposo
David Dillard
David Gorgon
David Norwood
David Raulo
David Ross
de voughn accooe
Dean Scully
Dennis Jeschke
Dennis Schaffers
Dennis Schierholz

Dennis Schneck
denti
Dick van Ginkel
Diego Barzon
Dierk Schneider
Dietmar Krueger
Dietmar Schinnerl
Dirk Becker
Dirk Wouters
Domingo Fivoli
DonChaos
Donn Lasher
Douglas Johnson
Dr. Leopold Winter
Dusan Sobotka
Earl Woodman
Ed Reilly
Edoardo Auteri
Eduardo Gallardo
Eduardo Luis Arana
Eirik Juliussen Olsen
Emilio Monelli
EP Technical Services
Epic Sound
Erasmus Kuhlmann
ergoGnomik
Eric Hilaire
Eric Hildebrandt
Eric Hill
Eric Jutrzenka
Erwin Reichel
Espen Skog
Evangelos Mpouras
Ewan Curtis
Fabio Zanicotti
Fabrizio Di Dio
Fabrizio Lodi
FARA Gießen GmbH
FeralChild
First Choice Auto's
Florian Rienhardt
Forum64. de
Francesco Baldassarri
Frank Fechner
Frank Glaush
Frank Gulasch
Frank Haaland
Frank Hempel
Frank Koschel
Frank Linhares
Frank Sleeuwaert
Frank Wolf
FranticFreddie
Fredrik Ramsberg
Fridun Nazaradeh
Friedel Kropp
Garrick West
Gary Lake-Schaal
Gary Pearson
Gavin Jones
Geir Sigmund Straume
Gerd Mitlaender
Giampietro Albiero
Giancarlo Valente
Gianluca Girelli
Giovanni Medina
Glen Fraser
Glen R Perye III
Glenn Main
Gordon Rimac
GRANT BYERS
Grant Louth
Gregor Bubek
Gregor Gramlich
Guido Ling
Guido von Gösseln
Guillaume Serge
Gunnar Hemmerling
Günter Hummel
Guy Simmons
Guybrush Threepwood
Hakan Blomqvist
Hans Pronk
Hans-Jörg Nett
Hans-Martin Zedlitz
Harald Dosch
Harri Salokorpi
Harry Culpan
Harry Venema
Heath Gallimore
Heinz Roesner
Heinz Stampfli
Helge Förster
Hendrik Fensch
Henning Harperath
Henri Parfait
Henrik Kühn
Holger Burmester
Holger Sturk
Howard Knibbs
Hubert de Hollain
Huberto Kusters
Hugo Maria Gerardus v.d. Aa
Humberto Castaneda
Ian Cross
IDE64 Staff
Igor Ianov
Igor Kurtes
Immo Beutler
Ingo Katte
Ingo Keck
Insanely Interested Publishing
IT-Dienstleistungen Obsieger
Ivan Elwood
Jaap HUIJSMAN
Jace Courville
Jack Wattenhofer
Jakob Schönpflug
Jakub Tyszko
James Hart
James Marshburn
James McClanahan
James Sutcliffe
Jan Bitruff
Jan Hildebrandt
Jan Iemhoff
Jan Kösters
Jan Peter Borsje
Jan Schulze
Jan Stoltenberg-Lerche
Janne Tompuri
Jannis Schulte
Jari Loukasmäki
Jason Smith
Javier Gonzalez Gonzalez
Jean-Paul Lauque
Jeffrey van der Schilden
Jens Schneider

Jens-Uwe Wessling
Jesse DiSimone
Jett Adams
Johan Arneklev
Johan Berntsson
Johan Svensson
Johannes Fitz
John Cook
John Deane
John Dupuis
John Nagi
John Rorland
John Sargeant
John Traeholt
Jon Sandelin
Jonas Bernemann
Jonathan Prosise
Joost Honig
Jordi Pakey-Rodriguez
Jöre Weber
Jörg Jungermann
Jörg Schaeffer
Jörg Weese
Josef Hesse
Josef Soucek
Josef Stohwasser
Joseph Clifford
Joseph Gerth
Jovan Crnjanin
Juan Pablo Schisano
Juan S. Cardona Iguina
JudgeBeeb
Juliussen Olsen
Juna Luis Fernandez Garcia
Jürgen Endras
Jürgen Herm Stapelberg
Jyrki Laurila
Kai Pernau
Kalle Pöyhönen
Karl Lamford
Karl-Heinz Blum
Karsten Engstler
Karsten Westebbe
katarakt
Keith McComb
Kenneth Dyke
Kenneth Joensson
Kevin Edwards
Kevin Thomasson
Kim Jorgensen
Kim Rene Jensen
Kimmo Hamalainen
Konrad Buryło
Kosmas Einbrodt
Kurt Klemm
Lachlan Glaskin
Large bits collider
Lars Becker
Lars Edelmann
Lars Slivsgaard
Lasse Lambrecht
Lau Olivier
Lee Chatt
Loan Leray
Lorenzo Quadri
Lorenzo Travagli
Lorin Millsap
Lothar James Foss
Lothar Serra Mari
Luca Papinutti
Ludek Smetana
Lukas Burger
Lutz-Peter Buchholz
Luuk Spaetgens
Mad Web Skills
MaDCz
Magnus Wiklander
Maik Diekmann
Malte Mundt
Manfred Wittemann
Manuel Beckmann
Manzano Mérida
Marc "3D-vice" Schmitt
Marc Bartel
Marc Jensen
Marc Schmidt
Marc Theunissen
Marc Tutor
Marc Wink
Marcel Buchtmann
Marcel Kante
Marco Beckers
Marco Cappellari
Marco Rivela
Marco van de Water
Marcus Gerards
Marcus Herbert
Marcus Linkert
Marek Pernicky
Mario Esposito
Mario Fetka
Mario Teschke
Mariusz Tymków
Mark Adams
Mark Anderson
Mark Green
Mark Hucker
Mark Leitiger
Mark Spezzano
Mark Watkin
Marko Rizvic
Markus Bieler
Markus Bonet
Markus Dauberschmidt
Markus Fehr
Markus Fuchs
Markus Guenther-Hirn
Markus Liukka
Markus Merz
Markus Roesgen
Markus Uttenweiler
Martin Bauhuber
Martin Benke
Martin Gendera
Martin Groß
Martin Gutenbrunner
Martin Johansen
Martin Marbach
Martin Sonnleitner
Martin Steffen
Marvin Hardy
Massimo Villani
Mathias Dellacherie
Mathieu Chouinard
Matthew Adams
Matthew Browne
Matthew Carnevale
Matthew Palmer

Matthew Santos
Matthias Barthel
Matthias Dolenc
Matthias Fischer
Matthias Frey
Matthias Grandis
Matthias Guth
Matthias Lampe
Matthias Meier
Matthias Mueller
Matthias Nofer
Matthias Schonder
Maurice Al-Khaliedy
Max Ihlenfeldt
Meeso Kim
Michael Dailly
Michael Dötsch
Michael Dreßel
Michael Fichtner
Michael Fong
Michael Geoffrey Stone
Michael Gertner
Michael Grün
Michael Habel
Michael Härtig
Michael Haynes
Michael J Burkett
Michael Jensen
Michael Jurisch
Michael Kappelgaard
Michael Kleinschmidt
Michael Lorenz
Michael Mayerhofer
Michael Nurney
Michael Rasmussen
Michael Richmond
Michael Sachse
Michael Sarbak
Michael Schneider
Michael Scholz
Michael Timm
Michael Traynor
Michael Whipp
Michal Ursiny
Michele Chiti
Michele Perini
Michele Porcu
Miguel Angel Rodriguez Jodar
Mikael Lund
Mike Betz
Mike Kastrantas
Mike Pikowski
Mikko Hämäläinen
Mikko Suontausta
Mirko Roller
Miroslav Karkus
Morgan Antonsson
Moritz
Morten Nielsen
MUBIQUO APPS,SL
Myles Cameron-Smith
Neil Moore
Nelson
neoman
Nicholas Melnick
Nikolaj Brinch Jørgensen
Nils Andreas
Nils Eilers
Nils Hammerich
Nils77
Norah Smith
Norman King
Normen Zoch
Olaf Grunert
Ole Eitels
Oliver Boerner
Oliver Brüggmann
Oliver Graf
Oliver Smith
Olivier Bori
ONEPSI LLC
oRdYNe
Osaühing Trioflex
OSHA-PROS USA
Padawer
Patrick Becher
Patrick Bürckstümmer
Patrick de Zoete
Patrick Toal
Patrick Vogt
Paul Alexander Warren
Paul Gerhardt (KONG)
Paul Jackson
Paul Johnson
Paul Kuhnast (mindrail)
Paul Massay
Paul Westlake
Paul Woegerer
Pauline Brasch
Paulo Apolonia
Pete Collin
Pete of Retrohax.net
Peter Eliades
Peter Gries
Peter Habura
Peter Herklotz
Peter Huyoff
Peter Knörzer
Peter Leswell
Peter Weile
Petri Alvinen
Philip Marien
Philip Timmermann
Philipp Rudin
Pierre Kressmann
Pieter Labie
Piotr Kmiecik
Power-on.at
Przemysław Safonow
Que Labs
R Welbourn
R-Flux
Rafał Michno
Rainer Kappler
Rainer Kopp
Rainer Weninger
Ralf Griewel
Ralf Pöscha
Ralf Reinhardt
Ralf Schenden
Ralf Smolarek
Ralf Zenker
Ralph Bauer
Ralph Wernecke
Rédl Károly
Reiner Lanowski
Remi Veilleux
Riccardo Bianchi

Richard Englert
Richard Good
Richard Menedetter
Richard Sopuch
Rick Reynolds
Rico Gruninger
Rob Dean
Robert Bernardo
Robert Eaglestone
Robert Grasböck
Robert Miles
Robert Schwan
Robert Shively
Robert Tangmar
Robert Trangmar
Rodney Xerri
Roger Olsen
Roger Pugh
Roland Attila Kett
Roland Evers
Roland Schatz
Rolf Hass
Ronald Cooper
Ronald Hunn
Ronny Hamida
Ronny Preiß
Roy van Zundert
Rüdiger Wohlfromm
Ruediger Schlenter
Rutger WIllemsen
Sampo Peltonen
Sarmad Gilani
SAS74
Sascha Hesse
Scott Halman
Scott Hollier
Scott Robison
Sebastian Baranski
Sebastian Bölling
Sebastian Felzmann
Sebastian Lipp
Sebastian Rakel
Şemseddin Moldibi
Šeth Morabito
Shawn McKee
Siegfried Hartmann
Zytex Online Store
Sigurbjorn Larusson
Sigurdur Finnsson
Simon Lawrence
Simon Wolf
spreen.digital
Stefan Haberl
Stefan Kramperth
Stefan Richter
Stefan Schultze
Stefan Sonnek
Stefan Theil
Stefan Vrampe
Stefano Canali
Stefano Mozzi
Steffen Reiersen
Stephan Bielmann
Stephen Jones
Stephen Kew
Steve Gray
Steve Kurlin
Steve Lemieux
Steven Combs
Stewart Dunn
Stuart Marsh
Sven Neumann
Sven Stache
Sven Sternberger
Sven Wiegand
Szabolcs Bence
Tantrumedia Limited
Techvana Operations Ltd.
Teddy Turmeaux
Teemu Korvenpää
The Games Foundation
Thierry Supplisson
Thieu-Duy Thai
Thomas Bierschenk
Thomas Edmister
Thomas Frauenknecht
Thomas Gitzen
Thomas Gruber
Thomas Haidler
Thomas Jager
Thomas Karlsen
Thomas Laskowski
Thomas Marschall
Thomas Niemann
Thomas Scheelen
Thomas Schilling
Thomas Tahsin-Bey
Thomas Walter
Thomas Wirtzmann
Thorsten Knoll
Thorsten Nolte
Tim Krome
Tim Waite
Timo Weirich
Timothy Blanks
Timothy Henson
Timothy Prater
Tobias Butter
Tobias Heim
Tobias Köck
Tobias Lüthi
Tommi Vasarainen
Toni Ammer
Tore Olsen
Torleif Strand
Torsten Schröder
Tuan Nguyen
Uffe Jakobsen
Ulrich Hintermeier
Ulrich Nieland
Ulrik Kruse
Urban Lindeskog
Ursula Förstle
Uwe Anfang
Uwe Boschanski
Vedran Vrbanc
Verm Project
Wayne Rittimann
Wayne Sander
Wayne Steele
Who Knows
Winfried Falkenhahn
Wolfgang Becker
Wolfgang Stabla
Worblehat
www.patop69.net
Yan B
Zoltan Markus
Zsolt Zsila

Bibliografie

[1] L. Soares en M. Stumm, "Flexsc: Flexible system call scheduling with exception-less system calls." in *Osdi*, dl. 10, 2010, blz. 1–8.

[2] N. Montfort, P. Baudoin, J. Bell, I. Bogost, J. Douglass, M. C. Marino, M. Mateas, C. Reas, M. Sample, en N. Vawter, *10 PRINT CHR $(205.5+ RND (1));: GOTO 10*. MIT Press, 2012.

[3] Actraiser, "Vic-ii for beginners: Screen modes, cheaper by the dozen," 2013. [Online]. Beschikbaar: http://dustlayer.com/vic-ii/2013/4/26/vic-ii-for-beginners-screen-modes-cheaper-by-the-dozen

Index

$00 (STOPTX), 120
$01 (STARTTX), 120
$D0 (RXNORMAL), 120
$D4 (DEBUGVIC), 121
$DC (DEBUGCPU), 121
$DE (RXONLYONE), 121
$F1 (FRAME1K), 121
$F2 (FRAME2K), 121

Amiga™-stijl audio, 87
audio-cross-bar-schakelaar, 146
audiomixer, 146

BASIC 65-opdrachten
    BANK, 4
    DMA, 4

cross-bar-schakelaar, audio, 146

DEBUGCPU, 121
DEBUGVIC, 121
Digitale audio, 87
digitale video, 23
DMA
    Inline-DMA-lijsten, 86
DMA-audio, 87

Floppy-DMA, 133
FRAME1K, 121
FRAME2K, 121

Hete registers, 31

IEC-controlleropdrachten, 154
IMDH™, 23
Inline-DMA-lijsten, 86
Integrated Marvellous Digital Hookup™, 23

lichtpen, 145
Lijntekening, 82
    DMA-optiebytes, 82

mixer, audio, 146

MOD-stijl audio, 87

Registers
    $D000, 54
    $D001, 54
    $D002, 54
    $D003, 54
    $D004, 54
    $D005, 54
    $D006, 54
    $D007, 54
    $D008, 54
    $D009, 54
    $D00A, 54
    $D00B, 54
    $D00C, 54
    $D00D, 54
    $D00E, 54
    $D00F, 55
    $D010, 55
    $D011, 55
    $D012, 55
    $D013, 55
    $D014, 55
    $D015, 55
    $D016, 55
    $D017, 55
    $D018, 55
    $D019, 55
    $D01A, 55
    $D01B, 55
    $D01C, 55
    $D01D, 55
    $D01E, 55
    $D01F, 55
    $D020, 55, 57, 59
    $D021, 55, 57, 59
    $D022, 55, 57, 59
    $D023, 55, 57, 60
    $D024, 55, 57, 60
    $D025, 55, 57, 60
    $D026, 55, 57, 60
    $D027, 55

$D028, 55
$D029, 55
$D02A, 55
$D02B, 55
$D02C, 55
$D02D, 55
$D02E, 55
$D02F, 55, 57, 60, 119
$D030, 55, 57
$D031, 57
$D032, 57
$D033, 57
$D034, 57
$D035, 57
$D036, 57
$D037, 57
$D038, 57
$D039, 57
$D03A, 57
$D03B, 57
$D03C, 57
$D03D, 57
$D03E, 57
$D03F, 57
$D040, 57
$D041, 58
$D042, 58
$D043, 58
$D044, 58
$D045, 58
$D046, 58
$D047, 58
$D048, 60
$D049, 60
$D04A, 60
$D04B, 60
$D04C, 60
$D04D, 60
$D04E, 60
$D04F, 60
$D050, 60
$D051, 60
$D052, 60
$D053, 60
$D054, 60
$D055, 60
$D056, 60
$D057, 60
$D058, 60
$D059, 60
$D05A, 60
$D05B, 60
$D05C, 60
$D05D, 60
$D05E, 60
$D05F, 60
$D060, 60
$D061, 60
$D062, 60
$D063, 60
$D064, 60
$D065, 60
$D068, 60
$D069, 60
$D06A, 60
$D06B, 60
$D06C, 60
$D06D, 61
$D06E, 61
$D06F, 61
$D070, 61
$D071, 61
$D072, 61
$D073, 61
$D074, 61
$D075, 61
$D076, 61
$D077, 61
$D078, 61
$D079, 61
$D07A, 61
$D07B, 61
$D07C, 61
$D080, 135

$D081, 135
$D082, 135
$D083, 135
$D084, 135
$D085, 135
$D086, 135
$D087, 135
$D088, 135
$D089, 135
$D08A, 135
$D100 - $D1FF, 58
$D200 - $D2FF, 58
$D300 - $D3FF, 58
$D400, 67
$D401, 67
$D402, 67
$D403, 67
$D404, 67
$D405, 67
$D406, 67
$D407, 67
$D408, 67
$D409, 67
$D40A, 67
$D40B, 67
$D40C, 67
$D40D, 67
$D40E, 67
$D40F, 67
$D410, 67
$D411, 67
$D412, 67
$D413, 67
$D414, 67
$D415, 67
$D416, 67
$D417, 67
$D418, 67
$D419, 67
$D41A, 67
$D41B, 67
$D41C, 67
$D41D, 67
$D41E, 67
$D600, 105
$D601, 105
$D602, 105
$D603, 105
$D604, 105
$D605, 105
$D606, 105
$D609, 106
$D60A, 106
$D60B, 106
$D60C, 106
$D60D, 106
$D60E, 106
$D60F, 106
$D610, 106
$D611, 106
$D612, 106
$D613, 106
$D614, 106
$D615, 106
$D616, 106
$D617, 106
$D618, 106
$D619, 106
$D61A, 106
$D61D, 106
$D61E, 106
$D620, 106
$D621, 106
$D622, 107
$D623, 107
$D625, 107
$D626, 107
$D627, 107
$D628, 107
$D629, 107
$D63C, 68
$D680, 142
$D681, 142
$D682, 142

$D683, 142
$D684, 143
$D686, 143
$D68A, 138, 143
$D68B, 138
$D68C, 138
$D68D, 138
$D68E, 138
$D68F, 138
$D690, 138
$D691, 138
$D692, 138
$D693, 138
$D694, 161
$D695, 161
$D697, 161
$D698, 161
$D699, 161
$D69A, 161
$D6A0, 137
$D6A1, 138
$D6A2, 137
$D6AE, 143
$D6AF, 143
$D6B0, 145
$D6B1, 145
$D6B2, 145
$D6B3, 145
$D6B4, 145
$D6B5, 145
$D6B7, 145
$D6B8, 145
$D6B9, 145
$D6BA, 145
$D6BB, 145
$D6BC, 145
$D6BD, 145
$D6BE, 145
$D6C0, 145
$D6E0, 119
$D6E1, 119
$D6E2, 119
$D6E3, 119
$D6E4, 119
$D6E5, 119
$D6E6, 119
$D6E7, 119
$D6E8, 119
$D6E9, 119
$D6EA, 119
$D6EB, 119
$D6EC, 119
$D6ED, 119
$D6EE, 119
$D6F4, 149
$D6F5, 149
$D6F8, 149
$D6F9, 149
$D6FA, 149
$D6FB, 149
$D6FC, 149
$D6FD, 149
$D700, 89
$D701, 89
$D702, 89
$D703, 90
$D704, 90
$D705, 90
$D706, 90
$D707, 90
$D70E, 90
$D711, 90, 149
$D712, 90
$D713, 90
$D714, 90
$D715, 90
$D716, 90
$D717, 90
$D71C, 90
$D71D, 90
$D71E, 90
$D71F, 90
$D720, 90
$D721, 90

$D722, 90
$D723, 90
$D724, 90
$D725, 90
$D726, 90
$D727, 90
$D728, 90
$D729, 90
$D72A, 90
$D72B, 90
$D72C, 90
$D72D, 90
$D72E, 90
$D72F, 90
$D730, 90
$D731, 91
$D732, 91
$D733, 91
$D734, 91
$D735, 91
$D736, 91
$D737, 91
$D738, 91
$D739, 91
$D73A, 91
$D73B, 91
$D73C, 91
$D73D, 91
$D73E, 91
$D73F, 91
$D740, 91
$D741, 91
$D742, 91
$D743, 91
$D744, 91
$D745, 91
$D746, 91
$D747, 91
$D748, 91
$D749, 91
$D74A, 91
$D74B, 91
$D74C, 91
$D74D, 91
$D74E, 91
$D74F, 91
$D750, 91
$D751, 91
$D752, 91
$D753, 91
$D754, 91
$D755, 91
$D756, 91
$D757, 91
$D758, 91
$D759, 91
$D75A, 92
$D75B, 92
$D75C, 92
$D75D, 92
$D75E, 92
$D75F, 92
$DC00, 97
$DC01, 97
$DC02, 97
$DC03, 97
$DC04, 97
$DC05, 97
$DC06, 97
$DC07, 97
$DC08, 97
$DC09, 97
$DC0A, 97
$DC0B, 97
$DC0C, 97
$DC0D, 97
$DC0E, 97
$DC0F, 97
$DC10, 100
$DC11, 100
$DC12, 100
$DC13, 100
$DC14, 100
$DC15, 100

$DC16, 100
$DC17, 100
$DC18, 100
$DC19, 100
$DC1A, 100
$DC1B, 100
$DC1C, 100
$DC1D, 100
$DC1E, 100
$DC1F, 100
$DD00, 98
$DD01, 98
$DD02, 98
$DD03, 98
$DD04, 98
$DD05, 98
$DD06, 98
$DD07, 98
$DD08, 98
$DD09, 98
$DD0B, 98
$DD0C, 99
$DD0D, 99
$DD0E, 99
$DD0F, 99
$DD10, 101
$DD11, 101
$DD12, 101
$DD13, 101
$DD14, 101
$DD15, 101
$DD16, 101
$DD17, 101
$DD18, 101
$DD19, 101
$DD1A, 101
$DD1B, 101
$DD1C, 101
$DD1D, 101
$DD1E, 102
$DD1F, 102
$FFD7100 – $FFD7107, 193
$FFD7110, 183
$FFD7111, 183
$FFD7112, 183
$FFD7113, 183
$FFD7114, 183
$FFD7115, 183
$FFD7116, 183, 184
$FFD7117, 184
$FFD7118, 184
$FFD7119, 184
$FFD711A, 184
$FFD711B, 184
$FFD711C, 184
$FFD711D, 184
$FFD711E, 184
$FFD711F, 184
$FFD7120, 184
$FFD7121, 184
$FFD7122, 184
$FFD7123, 184
$FFD7124, 184
$FFD7125, 184
$FFD7126, 184
$FFD7127, 184
$FFD7128, 184
$FFD7129, 184
$FFD712A, 184
$FFD712B, 184
$FFD712C, 184
$FFD712D, 184
$FFD712E, 184
$FFD712F, 184
$FFD7130, 184
$FFD7131, 184
$FFD7132, 184
$FFD7133, 184
$FFD7134, 184
$FFD7135, 184
$FFD7136, 184
$FFD7137, 185
$FFD7138, 185
$FFD7139, 185

$FFD713A, 185
$FFD713B, 185
$FFD713C, 185
$FFD713D, 185
$FFD7140 - $FFD717F, 183
$FFD7400, 188
$FFD7401, 188
$FFD7402, 188
$FFD7403, 188
$FFD7404, 188
$FFD7405, 188
$FFD7406, 188
$FFD7407, 188
$FFD7408, 188
$FFD7409, 188
$FFD740A, 188
$FFD740B, 188
$FFD740C, 188
$FFD740D, 189
$FFD740E, 189
$FFD740F, 189
$FFD7410, 189
$FFD7411, 189
$FFD7412, 189
100THSEC, 184, 185
53504 - 53759, 58
53760 - 54015, 58
54016 - 54271, 58
53248, 54
53249, 54
53250, 54
53251, 54
53252, 54
53253, 54
53254, 54
53255, 54
53256, 54
53257, 54
53258, 54
53259, 54
53260, 54
53261, 54
53262, 54
53263, 55
53264, 55
53265, 55
53266, 55
53267, 55
53268, 55
53269, 55
53270, 55
53271, 55
53272, 55
53273, 55
53274, 55
53275, 55
53276, 55
53277, 55
53278, 55
53279, 55
53280, 55, 57, 59
53281, 55, 57, 59
53282, 55, 57, 59
53283, 55, 57, 60
53284, 55, 57, 60
53285, 55, 57, 60
53286, 55, 57, 60
53287, 55
53288, 55
53289, 55
53290, 55
53291, 55
53292, 55
53293, 55
53294, 55
53295, 55, 57, 60, 119
53296, 55, 57
53297, 57
53298, 57
53299, 57
53300, 57
53301, 57
53302, 57
53303, 57

53304, 57
53305, 57
53306, 57
53307, 57
53308, 57
53309, 57
53310, 57
53311, 57
53312, 57
53313, 58
53314, 58
53315, 58
53316, 58
53317, 58
53318, 58
53319, 58
53320, 60
53321, 60
53322, 60
53323, 60
53324, 60
53325, 60
53326, 60
53327, 60
53328, 60
53329, 60
53330, 60
53331, 60
53332, 60
53333, 60
53334, 60
53335, 60
53336, 60
53337, 60
53338, 60
53339, 60
53340, 60
53341, 60
53342, 60
53343, 60
53344, 60
53345, 60
53346, 60
53347, 60
53348, 60
53349, 60
53352, 60
53353, 60
53354, 60
53355, 60
53356, 60
53357, 61
53358, 61
53359, 61
53360, 61
53361, 61
53362, 61
53363, 61
53364, 61
53365, 61
53366, 61
53367, 61
53368, 61
53369, 61
53370, 61
53371, 61
53372, 61
53376, 135
53377, 135
53378, 135
53379, 135
53380, 135
53381, 135
53382, 135
53383, 135
53384, 135
53385, 135
53386, 135
54272, 67
54273, 67
54274, 67
54275, 67
54276, 67
54277, 67

54278, 67
54279, 67
54280, 67
54281, 67
54282, 67
54283, 67
54284, 67
54285, 67
54286, 67
54287, 67
54288, 67
54289, 67
54290, 67
54291, 67
54292, 67
54293, 67
54294, 67
54295, 67
54296, 67
54297, 67
54298, 67
54299, 67
54300, 67
54301, 67
54302, 67
54784, 105
54785, 105
54786, 105
54787, 105
54788, 105
54789, 105
54790, 105
54793, 106
54794, 106
54795, 106
54796, 106
54797, 106
54798, 106
54799, 106
54800, 106
54801, 106
54802, 106
54803, 106
54804, 106
54805, 106
54806, 106
54807, 106
54808, 106
54809, 106
54810, 106
54813, 106
54814, 106
54816, 106
54817, 106
54818, 107
54819, 107
54821, 107
54822, 107
54823, 107
54824, 107
54825, 107
54844, 68
54912, 142
54913, 142
54914, 142
54915, 142
54916, 143
54918, 143
54922, 138, 143
54923, 138
54924, 138
54925, 138
54926, 138
54927, 138
54928, 138
54929, 138
54930, 138
54931, 138
54932, 161
54933, 161
54935, 161
54936, 161
54937, 161
54938, 161

54944, 137
54945, 138
54946, 137
54958, 143
54959, 143
54960, 145
54961, 145
54962, 145
54963, 145
54964, 145
54965, 145
54967, 145
54968, 145
54969, 145
54970, 145
54971, 145
54972, 145
54973, 145
54974, 145
54976, 145
55008, 119
55009, 119
55010, 119
55011, 119
55012, 119
55013, 119
55014, 119
55015, 119
55016, 119
55017, 119
55018, 119
55019, 119
55020, 119
55021, 119
55022, 119
55028, 149
55029, 149
55032, 149
55033, 149
55034, 149
55035, 149
55036, 149
55037, 149
55040, 89
55041, 89
55042, 89
55043, 90
55044, 90
55045, 90
55046, 90
55047, 90
55054, 90
55057, 90, 149
55058, 90
55059, 90
55060, 90
55061, 90
55062, 90
55063, 90
55068, 90
55069, 90
55070, 90
55071, 90
55072, 90
55073, 90
55074, 90
55075, 90
55076, 90
55077, 90
55078, 90
55079, 90
55080, 90
55081, 90
55082, 90
55083, 90
55084, 90
55085, 90
55086, 90
55087, 90
55088, 90
55089, 91
55090, 91
55091, 91
55092, 91

55093, 91
55094, 91
55095, 91
55096, 91
55097, 91
55098, 91
55099, 91
55100, 91
55101, 91
55102, 91
55103, 91
55104, 91
55105, 91
55106, 91
55107, 91
55108, 91
55109, 91
55110, 91
55111, 91
55112, 91
55113, 91
55114, 91
55115, 91
55116, 91
55117, 91
55118, 91
55119, 91
55120, 91
55121, 91
55122, 91
55123, 91
55124, 91
55125, 91
55126, 91
55127, 91
55128, 91
55129, 91
55130, 92
55131, 92
55132, 92
55133, 92
55134, 92
55135, 92
56320, 97
56321, 97
56322, 97
56323, 97
56324, 97
56325, 97
56326, 97
56327, 97
56328, 97
56329, 97
56330, 97
56331, 97
56332, 97
56333, 97
56334, 97
56335, 97
56336, 100
56337, 100
56338, 100
56339, 100
56340, 100
56341, 100
56342, 100
56343, 100
56344, 100
56345, 100
56346, 100
56347, 100
56348, 100
56349, 100
56350, 100
56351, 100
56576, 98
56577, 98
56578, 98
56579, 98
56580, 98
56581, 98
56582, 98
56583, 98
56584, 98

56585, 98
56587, 98
56588, 99
56589, 99
56590, 99
56591, 99
56592, 101
56593, 101
56594, 101
56595, 101
56596, 101
56597, 101
56598, 101
56599, 101
56600, 101
56601, 101
56602, 101
56603, 101
56604, 101
56605, 101
56606, 102
56607, 102
ABTPALSEL, 61
ACCESSKEY, 107
ADDRBANK, 89
ADDRLSB, 90, 92
ADDRLSBTRIG, 89
ADDRMB, 90, 92
ADDRMSB, 89
AE_D, 185
AE_H, 185
AE_M, 185
AF, 185
AIE, 185
ALDAY, 184, 185
ALGO, 135
ALHOURS, 184, 185
ALMINUTES, 184, 185
ALPHADELAY, 61
ALPHEN, 61
ALRM, 97, 99
ALRMAMPM, 101, 102
ALRMHOUR, 100–102
ALRMJIF, 100–102
ALRMMIN, 100–102
ALRMSEC, 100–102
ALT, 135
ASCIIKEY, 106, 107
ATTR, 58
AUDBLKTO, 90, 92
AUDEN, 92
AUDWRBLK, 92
AUTO2XSEL, 143
B1PIX, 58
B2ADEVN, 57
B2ADODD, 57
B2PIX, 58
B3ADEVN, 57
B3ADODD, 57
B3PIX, 58
B4ADEVN, 57
B4ADODD, 57
B4PIX, 58
B5ADEVN, 57
B5ADODD, 57
B5PIX, 58
B6ADEVN, 57
B6ADODD, 57
B6PIX, 58
B7ADEVN, 57
B7ADODD, 57
B7PIX, 58
BASHDDR, 106, 107
BBDRPOS, 60, 61
BCST, 119
BITPBANK, 61
BLKD, 92
BLNK, 55
BMM, 55
BNPIX, 57, 58
BOARDMINOR, 107
BOLDISALT, 61
BORDERCOL, 55, 57–59, 61
BP16ENS, 61

BPCOMP, 57, 58
BPENABLE, 57, 58
BPM, 58
BPX, 57, 58
BPY, 57, 58
BSF, 185
BSIE, 185
BSP, 55
BTPALSEL, 61
BUSY, 135
BXADEVN, 57, 58
BXADODD, 57, 58
BYADEVN, 57, 58
BYADODD, 57, 58
C128FAST, 56
CAIE, 185
CALXDELTALSB, 145
CALXSCALELSB, 145
CALXSCALEMSB, 145
CALYDELTALSB, 145
CALYDELTAMSB, 145
CALYSCALELSB, 145
CALYSCALEMSB, 145
CB, 55, 56
CDC00, 143
CEIE, 185
CH0RVOL, 90, 92
CH1BADDRBNK, 91
CH1BADDRLSB, 91
CH1BADDRMSB, 91
CH1CURADDRBNK, 91
CH1CURADDRLSB, 91
CH1CURADDRMSB, 91
CH1FREQ0, 91
CH1FREQ1, 91
CH1FREQ2, 91
CH1RVOL, 90, 92
CH1SBITS, 90
CH1TADDRLSB, 91
CH1TADDRMSB, 91
CH1TMGCNT0, 91
CH1TMGCNT1, 91
CH1TMGCNT2, 91
CH1VOLUME, 91
CH2BADDRBNK, 91
CH2BADDRLSB, 91
CH2BADDRMSB, 91
CH2CURADDRBNK, 91
CH2CURADDRLSB, 91
CH2CURADDRMSB, 91
CH2FREQ0, 91
CH2FREQ1, 91
CH2FREQ2, 91
CH2LVOL, 90, 92
CH2SBITS, 91
CH2TADDRLSB, 91
CH2TADDRMSB, 91
CH2TMGCNT0, 91
CH2TMGCNT1, 91
CH2TMGCNT2, 91
CH2VOLUME, 91
CH3BADDRBNK, 91
CH3BADDRLSB, 91
CH3BADDRMSB, 91
CH3CURADDRBNK, 92
CH3CURADDRLSB, 92
CH3CURADDRMSB, 92
CH3FREQ0, 91
CH3FREQ1, 91
CH3FREQ2, 91
CH3LVOL, 90, 92
CH3SBITS, 91
CH3TADDRLSB, 91
CH3TADDRMSB, 91
CH3TMGCNT0, 92
CH3TMGCNT1, 92
CH3TMGCNT2, 92
CH3VOLUME, 91
CHANSWAP, 92
CHARPTRBNK, 60, 61
CHARPTRLSB, 60, 61
CHARPTRMSB, 60, 61
CHARSZ, 105
CHARY16, 61

CHR16, 61
CHRCOUNT, 60, 61
CHRXSCL, 60, 62
CHRYSCL, 60, 62
CHXBADDRBNK, 90, 92
CHXBADDRLSB, 90, 92
CHXBADDRMSB, 90, 92
CHXCURADDRBNK, 90, 92
CHXCURADDRLSB, 90, 92
CHXCURADDRMSB, 90, 92
CHXEN, 92
CHXENA, 92
CHXFLG, 92
CHXFREQ0, 90, 92
CHXFREQ1, 90, 93
CHXFREQ2, 90, 93
CHXLOOP, 93
CHXOFFS, 90, 93
CHXSBITS, 90, 93
CHXSINE, 93
CHXSTP, 93
CHXTADDRL, 90, 93
CHXTADDRM, 90, 93
CHXTMGCNT0, 90, 93
CHXTMGCNT1, 90, 93
CHXTMGCNT2, 90, 93
CHXUSGN, 93
CHXVOLUME, 90, 93
CLKD, 185
CLKDE, 185
CLKF, 185
CLKIE, 185
CLOCK, 135
CMDANDSTAT, 142, 143
COLPTRLSB, 60, 62
COLPTRMSB, 60, 62
COMMAND, 119, 135
CONN41, 107
CRAM2K, 58
CRC, 135
CROM9, 58
CSEL, 56
CTHIE, 185
CTIE, 185
CTLIE, 185
CUIE, 186
D0D64, 138
D0IMG, 138
D0MD, 138
D0P, 138
D0STARTSEC0, 138, 139
D0STARTSEC1, 138, 139
D0STARTSEC2, 138, 139
D0STARTSEC3, 138, 139
D0WP, 139
D1D64, 139
D1IMG, 139
D1MD, 139
D1P, 139
D1STARTSEC0, 138, 139
D1STARTSEC1, 138, 139
D1STARTSEC2, 138, 139
D1STARTSEC3, 138, 139
D1WP, 139
DATA, 105, 135, 161
DATALOG0, 161
DATALOG1, 161
DATARATE, 137
DBGDIR, 137
DBGLFTCLP, 93
DBGMOTORA, 137
DBGRGTCLP, 93
DBGWDATA, 137
DBGWGATE, 137
DBLRR, 62
DD00DELAY, 101, 102
DDRA, 97–99
DDRB, 97–99
DEBUGC, 61, 62
DENSITY, 137
DEVNUM, 161
DIATN, 161
DIGILEFTLSB, 149
DIGILEFTMSB, 149

DIGILLSB, 149
DIGILMSB, 149
DIGIRIGHTLSB, 149
DIGIRIGHTMSB, 149
DIGIRLSB, 149
DIGIRMSB, 149
DIR, 135
DISKIN, 135
DISPROWS, 61, 62
DIVISOR, 105
DRQ, 135
DRXD, 119
DRXDV, 119
DS, 135
DSKCHG, 135
ECM, 56
EEbusy, 186
EEF, 186
EERD, 186
EHL, 186
EIE, 186
EN018B, 93
ENV2ATTDUR, 67
ENV2DECDUR, 67
ENV2LEVDUR, 67
ENV2RELDUR, 67
ENV3ATTDUR, 67
ENV3DECDUR, 67
ENV3LEVDUR, 67
ENV3OUT, 67, 68
ENV3RELDUR, 67
ENVXATTDUR, 67, 68
ENVXDECDUR, 67, 68
ENVXRELDUR, 67, 68
ENVXSUSLEV, 67, 68
EQ, 135
ESYN, 186
ET, 184, 186
ETRIG, 90, 93
ETRIGINLINE, 90, 93
ETRIGMAPD, 90, 93
EV1, 145
EV2, 145
EVF, 186
EVOW, 186
EVR, 186
EXGLYPH, 62
EXTIRQS, 62
EXTRTCA1DATE, 188, 189
EXTRTCA1DOW, 189
EXTRTCA1DOWSEL, 189
EXTRTCA1HOUR, 188, 189
EXTRTCA1HOUR12EN, 189
EXTRTCA1HOURPM, 189
EXTRTCA1M1, 189
EXTRTCA1M2, 189
EXTRTCA1M3, 189
EXTRTCA1M4, 189
EXTRTCA1MIN, 188, 189
EXTRTCA1SEC, 188, 189
EXTRTCA2DATE, 189
EXTRTCA2DOW, 189
EXTRTCA2DOWSEL, 189
EXTRTCA2HOUR, 188, 189
EXTRTCA2HOUR12EN, 190
EXTRTCA2HOURPM, 190
EXTRTCA2M2, 190
EXTRTCA2M3, 190
EXTRTCA2M4, 190
EXTRTCA2MIN, 188, 190
EXTRTCAGINGOFS, 189, 190
EXTRTCCENTURY, 190
EXTRTCCTRL, 189, 190
EXTRTCDAY, 188, 190
EXTRTCDOW, 188, 190
EXTRTCHOUR, 188, 190
EXTRTCHOUR12EN, 190
EXTRTCHOURPM, 190
EXTRTCMIN, 188, 190
EXTRTCMONTH, 188, 190
EXTRTCSEC, 188, 190
EXTRTCST, 189, 190
EXTRTCTEMPMSB, 189, 190
EXTRTCYEAR, 188, 190

EXTSYNC, 58
FAST, 58
FCLRHI, 62
FCLRLO, 62
FCOLMCM, 62
FDC2XSEL, 143
FDCENC, 143
FDCTIBEN, 143
FDCVARSPD, 143
FILLVAL, 143
FLG, 97, 99
FLTRBDPASS, 68
FLTRCUTFRQHI, 67, 68
FLTRCUTFRQLO, 67, 68
FLTRCUTV3, 68
FLTREXTINP, 68
FLTRHIPASS, 68
FLTRLOPASS, 68
FLTRRESON, 67, 68
FLTRVOL, 67, 68
FLTRVXOUT, 68
FNRASTERLSB, 60, 62
FNRASTERMSB, 60, 62
FNRST, 62
FNRSTCMP, 62
FREE, 136
FRMERR, 105
GESTUREDIR, 145, 146
GESTUREID, 145, 146
GP0, 186
GP1, 186
H1280, 58
H640, 59
HDSCL, 107
HDSDA, 107
HOTREG, 62
HPOS, 57, 59
HSYNCP, 62
IFRXIRQ, 105
IFRXNMI, 105
IFTXIRQ, 105
IFTXNMI, 105
ILP, 56
IMALRM, 101, 102
IMFLG, 101, 102
IMODA, 97, 99
IMODB, 97, 99
IMRXIRQ, 105
IMRXNMI, 105
IMSP, 101, 102
IMTB, 101, 102
IMTXIRQ, 105
IMTXNMI, 105
INDEX, 136
INT, 59
INTDE, 186
IR, 97, 99
IRQ, 136
IRQCH1OFFS, 90
IRQCH2OFFS, 90
IRQCH3OFFS, 90
IRQEN, 161
IRQFLAG, 161
IRQRDY, 161
IRQRDYEN, 161
IRQRX, 161
IRQRXEN, 161
IRQTO, 161
IRQTOEN, 161
ISBC, 56
ISRCLR, 97, 99
ISSC, 56
J21H, 107
J21HDDR, 107
J21L, 107
J21LDDR, 107
JOYSWAP, 107
KEY, 55-57, 59, 60, 62, 119
KEYLEDENA, 107
KEYLEDREG, 106, 107
KEYLEDVAL, 106, 107
KEYLEFT, 107
KEYMATRIXPEEK, 106, 107
KEYMATRIXSEL, 106, 107

KEYQUEUE, 108
KEYUP, 108
KSCNRATE, 106, 108
LED, 136
LINESTEPLSB, 60, 62
LINESTEPMSB, 60, 62
LJOYA, 108
LJOYB, 108
LOAD, 97, 99
LOST, 136
LPX, 55, 56
LPY, 55, 56
M65MODEL, 107, 108
MACADDR2, 119
MACADDR3, 119
MACADDR4, 119
MACADDR5, 119
MACADDR6, 119
MACADDRX, 119
MALT, 108
MAPEDPAL, 61, 63
MC1, 55–57, 59, 63
MC2, 55–57, 59, 60, 63
MC3, 55–57, 59, 60, 63
MCAPS, 108
MCM, 56
MCST, 119
MCTRL, 108
MDISABLE, 108
MIIMPHY, 119
MIIMREG, 119
MIIMVLSB, 119
MIIMVMSB, 119
MISBC, 56
MISSC, 56
MIXREGDATA, 149
MIXREGSEL, 149
MLSHFT, 108
MMEGA, 108
MODKEYALT, 108
MODKEYCAPS, 108
MODKEYCTRL, 108
MODKEYLSHFT, 108
MODKEYMEGA, 108
MODKEYRSHFT, 108
MODKEYSCRL, 108
MONO, 59
MOTOR, 136
MRIRQ, 56
MRSHFT, 108
MSCRL, 108
NOBUF, 136
NOBUGCOMPAT, 63
NOCRC, 119
NOMBWRAP, 93
NOMIX, 93
NOPROM, 120
NORRBWRAP, 63
NORRDEL, 63
NVRAM, 183
OMODA, 97, 99
OMODB, 97, 99
Ongebruikt, 183, 184, 188
OSC3RNG, 67, 68
OSKALT, 108
OSKDEBUG, 109
OSKDIM, 109
OSKEN, 109
OSKTOP, 109
OSKZEN, 109
OSKZON, 109
PADDLE1, 67, 68
PADDLE2, 67, 68
PAL, 59
PALBLUE, 58, 59
PALEMU, 63
PALGREEN, 58, 59
PALNTSC, 63
PALRED, 58, 59
PBONA, 97, 99
PBONB, 97, 99
PCODE, 135, 136
PCTCR0, 184, 186
PCTCR1, 184, 186

PETSCIIKEY, 106, 109
PORF, 186
PORTA, 97–99
PORTB, 97–99
PORTF, 106, 109
PORTFDDR, 106, 109
POTAX, 106, 109
POTAY, 106, 109
POTBX, 107, 109
POTBY, 107, 109
PRESENT, 161
PROT, 136, 161
PTYEN, 105
PTYERR, 105
PTYEVEN, 106
PWMPDM, 149
RASCMP, 61, 63
RASCMPMSB, 61, 63
RASLINE0, 61, 63
RASTERHEIGHT, 61, 63
RC, 55, 56
RC8, 56
RCENABLED, 120
RDCMD, 136
RDREQ, 136
READBACKLSB, 149
READBACKMSB, 149
REALHW, 109
RIRQ, 56
RMODA, 98, 99
RMODB, 98, 99
RNF, 136
ROM8, 59
ROMA, 59
ROMC, 59
ROME, 59
RSEL, 56
RST, 56, 120
RST41, 109
RSTDELEN, 63
RTCDAY, 183
RTCDOW, 183, 184, 186
RTCHOUR, 183
RTCHOUR24EN, 183
RTCHOURPM, 183
RTCMIN, 183
RTCMONTH, 183
RTCSEC, 183
RTCYEAR, 183
RUN, 136
RXBF, 119, 120
RXBLKD, 120
RXEN, 106
RXOVRRUN, 106
RXPH, 119, 120
RXQ, 120
RXQEN, 120
RXRDY, 106
S1X, 54
S1Y, 54
S2X, 54
S2Y, 54
S3X, 54
S3Y, 54
S4X, 54
S4Y, 54
S5X, 54
S5Y, 54
S6X, 54
S6Y, 54
S7X, 54
S7Y, 55
SBC, 55, 56
SCM, 55, 56
SCREENCOL, 55–57, 59, 63
SCRNPTRBNK, 60, 63
SCRNPTRLSB, 60, 63
SCRNPTRMB, 60, 63
SCRNPTRMSB, 60, 63
SDBDRWDLSB, 60, 63
SDBDRWDMSB, 60, 63
SDBSH, 109
SDCLK, 109
SDCS, 109

SDDATA, 109
SDR, 97–99
SE, 55, 56
SECTOR, 135, 136
SECTOR0, 142, 143
SECTOR1, 142, 143
SECTOR2, 142, 143
SECTOR3, 143
SEXX, 55, 56
SEXY, 55, 56
SHDEMU, 63
SIDE, 135, 136
SIDEXENA, 68
SIDMODE, 68
SIDPANLEFT, 67, 68
SIDPANRIGHT, 67, 68
SILENT, 139
SMTH, 63
SNX, 54, 56
SNY, 54, 56
SP, 98, 99
SPMOD, 98, 99
SPR16EN, 60, 63
SPR1COL, 55
SPR2COL, 55
SPR3COL, 55
SPR4COL, 55
SPR5COL, 55
SPR6COL, 55
SPR7COL, 55
SPRALPHAVAL, 61, 63
SPRBPMEN, 60, 63
SPRENALPHA, 61, 63
SPRENV400, 61, 64
SPRH640, 64
SPRHGHT, 60, 64
SPRHGTEN, 60, 64
SPRMC0, 55, 57, 59, 60, 64
SPRMC1, 55, 57, 59, 60, 64
SPRNCOL, 55, 57
SPRPALSEL, 61, 64
SPRPTR16, 64
SPRPTRADRLSB, 60, 64
SPRPTRADRMSB, 61, 64
SPRPTRBNK, 61, 64
SPRTILEN, 60, 64
SPRX64EN, 60, 64
SPRXSMSBS, 60, 64
SPRYADJ, 61, 64
SPRYMSBS, 61, 64
SPRYSMSBS, 61, 64
SPTRCONT, 64
SSC, 55, 57
STC, 161
STD, 161
STDDIR, 162
STEP, 135, 136
STNODEV, 162
STNOEOI, 162
STOP, 186
STRM, 120
STRTA, 98, 99
STRTB, 98, 99
STSRQ, 162
STTO, 162
STVERIFY, 162
SWAP, 136
SXMSB, 55, 57
SYNCMOD, 105, 106
SYSCTL, 106, 109
TA, 98, 99
TALATCH, 100–102
TARGANY, 139
TB, 98, 99
TBDRPOS, 60, 64
TD, 184, 186
TE, 186
TEMPH, 184, 186
TEMPL, 184, 186
TEXTXPOS, 60, 64
TEXTYPOS, 60, 64
TF, 186
THE, 186
THF, 186

THIE, 187
THOW, 187
THR, 187
THT, 184, 187
TIE, 187
TIMERA, 97–99
TIMERB, 97, 98, 100
TK0, 136
TLE, 187
TLF, 187
TLIE, 187
TLOW, 187
TLR, 187
TLT, 184, 187
TOD50, 98, 100
TODAMPM, 98, 100–102
TODEDIT, 98, 100
TODHOUR, 97, 98, 100–102
TODJIF, 97, 98, 100–102
TODMIN, 97, 98, 100–102
TODSEC, 97, 98, 100–102
TOUCH1XLSB, 145, 146
TOUCH1XMSB, 145, 146
TOUCH1YLSB, 145, 146
TOUCH1YMSB, 145, 146
TOUCH2XLSB, 145, 146
TOUCH2XMSB, 145, 146
TOUCH2YLSB, 145, 146
TOUCH2YMSB, 145, 146
TRACK, 135, 136
TSEVI100th, 185, 187
TSEVICount, 184, 187
TSEVIDat, 185, 187
TSEVIHrs, 185, 187
TSEVIMin, 185, 187
TSEVIMonth, 185, 187
TSEVISec, 185, 187
TSEVIYear, 185, 187
TSTHighCount, 184, 187
TSTHighDat, 184, 187
TSTHighHrs, 184, 187
TSTHighMin, 184, 187
TSTHighMonth, 184, 187
TSTHighSec, 184, 187
TSTHighYear, 184, 187
TSTLowCount, 184, 187
TSTLowDat, 184, 187
TSTLowHrs, 184, 187
TSTLowMin, 184, 187
TSTLowMonth, 184, 188
TSTLowSec, 184, 188
TSTLowYear, 184, 188
TXEN, 106
TXIDLE, 120
TXPH, 119, 120
TXQ, 120
TXQEN, 120
TXRST, 120
TXSZLSB, 119, 120
TXSZMSB, 119, 120
UF, 188
UFAST, 109
UIE, 188
UPDN1, 145, 146
UPDN2, 145, 146
UPSCALE, 64
USEL, 188
USEREAL0, 139
USEREAL1, 139
UUID64, 193
V400, 59
VDRQ, 143
VEQINH, 143
VFAST, 64
VFDC0, 143
VFDC1, 143
VGAHDTV, 64
VICIII, 143
VIRTKEY1, 106, 109
VIRTKEY2, 106, 110
VIRTKEY3, 106, 110
VLF, 188
VLOST, 143
VOICE1CTRLRMF, 68

VOICE1CTRLRMO, 68
VOICE2CTRLRMF, 69
VOICE2CTRLRMO, 69
VOICE2FRQHI, 67
VOICE2FRQLO, 67
VOICE2PWHI, 67
VOICE2PWLO, 67
VOICE2UNSD, 67
VOICE3CTRLRMF, 69
VOICE3CTRLRMO, 69
VOICE3FRQHI, 67
VOICE3FRQLO, 67
VOICE3PWHI, 67
VOICE3PWLO, 67
VOICE3UNSD, 67
VOICEXCTRLGATE, 69
VOICEXCTRLPUL, 69
VOICEXCTRLRNW, 69
VOICEXCTRLSAW, 69
VOICEXCTRLTRI, 69
VOICEXCTRLTST, 69
VOICEXFRQHI, 67, 69
VOICEXFRQLO, 67, 69
VOICEXPWHI, 67, 69
VOICEXPWLO, 67, 69
VOICEXUNSD, 67, 69
VPOS, 57, 59
VRFOUND, 144
VRNF, 144
VS, 55, 57
VSYNCP, 64
VWFOUND, 144
WGATE, 136
WITHIO, 89
WRCMD, 136
WTREQ, 136
XINV, 146
XPOSLSB, 60, 64
XPOSMSB, 60, 64
XSCL, 55, 57
YINV, 146
YSCL, 55, 57
Rijmasker, 47
RXNORMAL, 120
RXONLYONE, 121

STARTTX, 120
STOPTX, 120

Textuurschaling, 84
TIB, 130, 133
Track at once, 133
Track Information Block, 130, 133
Track-DMA, 133