APPENDIX J

System Memory Map

- Introduction
- MEGA65 Native Memory Map
- $D000 – $DFFF I/O Personalities
- CPU Memory Banking
- C64/C65 ROM Emulation

INTRODUCTION

The MEGA65 computer has a large 28-bit address space, which allows it to address up to 256MB of memory and memory-mapped devices. This memory map has several different views, depending on which mode the computer is operating in. Broadly, there are five main modes: (1) Hypervisor mode; (2) C64 compatibility mode; (3) C65 compatibility mode; (4) UltiMAX compatibility mode; and (5) MEGA65-mode, or one of the other modes, where the programmer has made use of MEGA65 enhanced features.

It is important to understand that, unlike the C128, the C65 and MEGA65 allow access to all enhanced features from C64-mode, if the programmer wishes to do so. This means that while we frequently talk about “C64-mode,” “C65-mode” and “MEGA65-mode,” these are simply terms of convenience for the MEGA65 with its memory map (and sometimes other features) configured to provide an environment that matches the appropriate mode. The heart of this is the MEGA65’s flexible memory map.

In this appendix, we will begin by describing the MEGA65’s native memory map, that is, where all of the memory, I/O devices and other features appear in the 28-bit address space. We will then explain how C64 and C65 compatible memory maps are accessed from this 28-bit address space.

MEGA65 NATIVE MEMORY MAP

The First Sixteen 64KB Banks

The MEGA65 uses a similar memory map to that of the C65 for the first MB of memory, i.e., 16 memory banks of 64KB each. This is because the C65's 4510 CPU can access only 1MB of address space. These banks can be accessed from BASIC 65 using the **BANK**, **DMA**, **PEEK** and **POKE** commands. The following table summarises the contents of the first 16 banks:

HEX	DEC	Address	Contents
0	0	$0xxxx	First 64KB RAM. This is the RAM visible in C64-mode.
1	1	$1xxxx	Second 64KB RAM. This is the 2nd 64KB of RAM present on a C65.
2	2	$2xxxx	First half of C65 ROM (C64-mode and shared components) *or* RAM
3	3	$3xxxx	Second half of C65 ROM (C65-mode components) *or* RAM
4	4	$4xxxx	Additional RAM (384KB or larger chip-RAM models)
5	5	$5xxxx	Additional RAM (384KB or larger chip-RAM models)
6	6	$6xxxx	Additional RAM (*512KB or larger chip-RAM models)
7	7	$7xxxx	Additional RAM (*512KB or larger chip-RAM models)
8	8	$8xxxx	Additional RAM (*1MB or larger chip-RAM models)
9	9	$9xxxx	Additional RAM (*1MB or larger chip-RAM models)
A	10	$Axxxx	Additional RAM (*1MB or larger chip-RAM models)
B	11	$Bxxxx	Additional RAM (*1MB or larger chip-RAM models)
C	12	$Cxxxx	Additional RAM (*1MB or larger chip-RAM models)
D	13	$Dxxxx	Additional RAM (*1MB or larger chip-RAM models)
E	14	$Exxxx	Additional RAM (*1MB or larger chip-RAM models)
F	15	$Fxxxx	Additional RAM (*1MB or larger chip-RAM models)

* Note that the MEGA65 presently only provides a model featuring 384KB of chip-RAM. Future models may feature larger amounts of chip-RAM (such as 512KB and 1MB).

The key features of this address space are the 128KB of RAM in the first two banks, which is also present on the C65. If you intend to write programs which can also run on a C65, you should only use these two banks of RAM.

On all models it is possible to use all or part of the 128KB of “ROM” space as RAM. To do this, you must first request that the Hypervisor removes the read-only protection on this area, before you will be able to change its contents. If you are writing a program which will start from C64-mode, or otherwise switch to using the C64 part of the ROM, instead of the C65 part), then the second half of that space, i.e., BANK 3, can be safely used for your programs. This gives a total of 192KB of RAM, which is available on all models of the MEGA65.

On models that have 384KB or more of chip RAM, BANK 4 and 5 are also available. Similarly, models which provide 1MB or more of chip RAM will have BANK 6 through 15 also available, giving a total of 896KB (or 960KB, if only the C64 part of the ROM is required) of RAM available for your programs. Note that the MEGA65’s built-in freeze cartridge currently freezes only the first 384KB of RAM.

Colour RAM

The MEGA65’s VIC-IV video controller supports much larger screens than the VIC-II or VIC-III. For this reason, it has access to a separate colour RAM, similar to on the C64. For compatibility with the C65, the first two kilo-bytes of this are accessible at $1F800 – $1FFFF. The full 32KB or 64KB of colour RAM is located at $FF80000. This is most easily accessed through the use of advanced DMA operations, or the 32-bit base-page indirect addressing mode of the processor.

At the time of writing, the **BANK** and **DMA** commands cannot be used to access the rest of the colour RAM, because the colour RAM is not located in the first mega-byte of address space. This may be corrected in a future revision of the MEGA65, allowing access to the full colour RAM via BANK 15 or an equivalent DMA job.

Additional RAM

Apart from the 384kb of chip-RAM found as standard on all MEGA65 models, most models (devkit, release boards and xemu, but NOT on Nexys boards currently) also have an extra 8MB of RAM starting at $8000000, referred to as ’ATTIC RAM’. It is not visible to the other chips (vic/sid/etc) and can’t be used for audio DMA, but code can run from it (more slowly) or it can be used to store content and DMA it in/out of the chip-RAM.

There are also plans underway to support a PMOD hyperRAM module (installed via the trapdoor beneath the MEGA65) in order to provide a further 8MB of RAM starting at $8800000, referred to as 'CELLAR RAM'.

28-bit Address Space

In addition to the C65-style 1MB address space, the MEGA65 extends this to 256MB, by using 28-bit addresses. The following shows the high-level layout of this address space.

HEX	DEC	Size	Contents
0000000	0	1	CPU I/O Port Data Direction Register
0000001	1	1	CPU I/O Port Data
0000002 – 005FFFF	2 – 384KB	384KB	Fast chip RAM (40MHz)
0060000 – 0FFFFFF	384KB – 16MB	15.6MB	Reserved for future chip RAM expansion
1000000 – 3FFFFFF	16MB – 64MB	48MB	Reserved
4000000 – 7FFFFFF	64MB – 128MB	64MB	Cartridge port and other devices on the slow bus (1 – 10 MHz)
8000000 – 87FFFFF	128MB – 135MB	8MB	8MB ATTIC RAM (all models apart from Nexys, presently)
8800000 – 8FFFFFF	135MB – 144MB	8MB	8MB CELLAR RAM (planned PMOD module installed via trapdoor)
9000000 – EFFFFFF	144MB – 240MB	96MB	Reserved for future expansion RAM
F000000 – FF7DFFF	240MB – 255.49MB	15.49MB	Reserved for future I/O expansion
FF7E000 – FF7EFFF	255.49MB – 255.49MB	4KB	VIC-IV Character ROM (write only)
FF80000 – FF87FFF	255.5MB – 255.53MB	32KB	VIC-IV Colour RAM (32KB colour RAM – available on all models)
FF88000 – FF8FFFF	255.53MB – 255.57MB	32KB	Additional VIC-IV Colour RAM (64KB colour RAM – planned to be available on R3 models and beyond)
FF90000 – FFCAFFF	255.53MB – 255.80MB	216KB	Reserved
FFCB000 – FFCBFFF	255.80MB – 255.80MB	4KB	Emulated C1541 RAM
FFCC000 – FFCFFFF	255.80MB – 255.81MB	16KB	Emulated C1541 ROM

continued ...

…continued

HEX	DEC	Size	Contents
FFD0000 – FFD0FFF	255.81MB – 255.81MB	4KB	C64 $Dxxx I/O Personality
FFD1000 – FFD1FFF	255.81MB – 255.82MB	4KB	C65 $Dxxx I/O Personality
FFD2000 – FFD2FFF	255.82MB – 255.82MB	4KB	MEGA65 $Dxxx Ethernet I/O Personality
FFD3000 – FFD3FFF	255.82MB – 255.82MB	4KB	MEGA65 $Dxxx Normal I/O Personality
FFD4000 – FFD5FFF	255.82MB – 255.83MB	8KB	Reserved
FFD6000 – FFD67FF	255.83MB – 255.83MB	2KB	Hypervisor scratch space
FFD6000 – FFD6BFF	255.83MB – 255.83MB	3KB	Hypervisor scratch space
FFD6C00 – FFD6DFF	255.83MB – 255.83MB	512	F011 floppy controller sector buffer
FFD6E00 – FFD6FFF	255.83MB – 255.83MB	512	SD Card controller sector buffer
FFD7000 – FFD70FF	255.83MB – 255.83MB	256	MEGAphone r1 I2C peripherals
FFD7100 – FFD71FF	255.83MB – 255.83MB	256	MEGA65 r2 I2C peripherals
FFD7200 – FFD72FF	255.83MB – 255.83MB	256	MEGA65 HDMI I2C registers (only for R2 and older models fitted with the ADV7511 HDMI driver chip)
FFD7300 – FFD7FFF	255.83MB – 255.84MB	3.25KB	Reserved for future I2C peripherals
FFD8000 – FFDBFFF	255.83MB – 255.86MB	16KB	Hypervisor ROM (only visible in Hypervisor Mode)
FFDC000 – FFDDFFF	255.86MB – 255.87MB	8KB	Reserved for Hypervisor Mode ROM expansion
FFDE000 – FFDE7FF	255.87MB – 255.87MB	2KB	Reserved for Ethernet buffer expansion
FFDE800 – FFDEFFF	255.87MB – 255.87MB	2KB	Ethernet frame read buffer (read only) and Ethernet frame write buffer (write only)
FFDF000 – FFDFFFF	255.87MB – 255.87MB	4KB	Virtual FPGA registers (selected models only)
FFE0000 – FFFFFFF	255.87MB – 256MB	128KB	Reserved

$D000 – $DFFF I/O PERSONALITIES

The MEGA65 supports four different I/O personalities. These are selected by writing the appropriate values to the $D02F KEY register, which is visible in all four I/O personalities. There is more information in Chapter/Appendix 11 on page 11-3 about the use of the KEY register.

The following table shows which I/O devices are visible in each of these I/O modes, as well as the KEY register values that are used to select the I/O personality.

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 card	SD card
$D0A0 - $D0FF	VIC-II	RAM EXPAND CONTROL	-	-
$D100 - $D1FF	VIC-II	RED Palette	RED Palette	RED Palette
$D200 - $D2FF	VIC-II	GREEN Palette	GREEN Palette	GREEN Palette
$D300 - $D3FF	VIC-II	BLUE Palette	BLUE Palette	BLUE Palette
$D400 - $D41F	SID Right #1	SID Right #1	SID Right #1	SID Right #1
$D420 - $D43F	SID Right #2	SID Right #2	SID Right #2	SID Right #2
$D440 - $D45F	SID Left #1	SID Left #1	SID Left #1	SID Left #1
$D460 - $D47F	SID Left #2	SID Left #2	SID Left #2	SID Left #2
$D480 - $D49F	SID Right #1	SID Right #1	SID Right #1	SID Right #1
$D4A0 - $D4BF	SID Right #2	SID Right #2	SID Right #2	SID Right #2
$D4C0 - $D4DF	SID Left #1	SID Left #1	SID Left #1	SID Left #1
$D4E0 - $D4FF	SID Left #2	SID Left #2	SID Left #2	SID Left #2
$D500 - $D5FF	SID images	-	Reserved	Reserved
$D600 - $D63F	-	UART	UART	UART
$D640 - $D67F	-	UART images	HyperTrap Registers	HyperTrap Registers
$D680 - $D6FF	-	-	MEGA65 Devices	MEGA65 Devices
$D700 - $D7FF	-	-	MEGA65 Devices	MEGA65 Devices
$D800 - $DBFF	COLOUR RAM	COLOUR RAM	ETHERNET Buffer	COLOUR RAM
$DC00 - $DDFF	CIAs	CIAs / COLOUR RAM	ETHERNET Buffer	CIAs / COLOUR RAM
$DE00 - $DFFF	CART I/O	CART I/O	ETHERNET Buffer	CART I/O / SD SECTOR

[1] In the C64 I/O personality, $D030 behaves as on C128, allowing toggling between 1MHz and 2MHz CPU speed.

[2] The additional MEGA65 SIDs are visible in all I/O personalities.

[3] Some models may replace the repeated images of the first four SIDs with four additional SIDs, for a total of 8 SIDs.

CPU MEMORY BANKING

The 45GS02 processor, like the 6502, can only "see" 64KB of memory at a time. Access to additional memory is via a selection of bank-switching mechanisms. For backward-compatibility with the C64 and C65, the memory banking mechanisms for both of these computers are supported on the MEGA65:

1. C65-style MAP instruction banking
2. C65-style $D030 banking
3. C64-style cartridge banking
4. C64-style $00 / $01 banking

The MAP register overrides all other banking mechanisms. This mechanism selects which of the eight 8KB regions of the 16-bit address space $0000 – $FFFF are mapped to other addresses via an offset. If a region is mapped, then the other banking mechanisms do not apply. This is true even if the offset is 0, allowing the 16-bit addresses to access RAM in bank 0 (such as address 0.D000).

C65-style $D030 banking and C64-style $00 / $01 banking both select regions to map to bank 2, which (by default) contains C64 ROM code. These two mechanisms overlap in which regions they can map to ROM. If either mechanism maps a region to ROM (and it is not mapped elsewhere by the MAP register), then it is mapped to ROM.

The following diagram shows the different types of banking that can apply to the different areas of the 64KB that the CPU can see.

MAP	MAP LO (4 x 8KB slabs)	MAP HI (4 x 8KB slabs)				
I/O/CART		CART ROMLO	CART ROMHI		I/O	CART ROMHI
D030		CHARROM	BASIC	INTERFACE		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

There are actually a few further complications. For example, if the cartridge selects the UltiMAX™ game mode, then only the first 4KB of RAM will be visible, and the remaining address space will be un-mapped, and able to be supplied by the cartridge.

C64/C65 ROM EMULATION

The C64 and C65 use ROM memories to hold the KERNAL and BASIC system. The MEGA65 is different: It uses 128KB of its 384KB fast chip RAM at $20000 – $3FFFF (banks 2 and 3) to hold these system programs. This makes it possible to change or upgrade the “ROM” that the MEGA65 is running, without having to open the computer. It is even possible to use the MEGA65’s Freeze Menu to change the “ROM” being used while a program is running.

The C64 and C65 memory banking methods use this 128KB of area when making ROM banks visible. When the RAM banks are mapped, they are always read-only. However, if the MAP instruction or DMA is used to access that address area, it is possible to write to it. For improved backward compatibility, the whole 128KB region of memory is normally set to read-only.

A program can, however, request read-write access to this 128KB area of memory, so that it can make full use of the MEGA65’s 384KB of chip RAM. This is accomplished by triggering the *Toggle Rom Write-protect* system trap of the hypervisor. The following code-fragment demonstrates how to do this. Calling it a second time will re-activate the write-protection.

```
LDA #$70
STA $D640
NOP
```

This fragment works by calling sub-function $70 (toggle ROM write-protect) of Hypervisor trap $00. Note that the NOP is mandatory. The MEGA65 I/O personality must be first selected, so that the $D640 register is un-hidden.

The current write-protection state can be tested by attempting to write to this area of memory. Also, you can examine and toggle the current state in the MEGA65 Freeze Menu.

NOTE: If you are starting your program from C65-mode, you must first make sure that the I/O area is visible at $D000-$DFFF. The simplest way to do this is to use the MAP instruction with all zero values in the registers. The following fragment demonstrates this, and also makes sure that the MEGA65 I/O context is active, so that the hypervisor trap will be able to trigger:

```
; Clear C65 memory map
LDA #$00
TAX
TAY
TAZ
MAP
; Bank I/O in via C64 mechanism
LDA #$35
STA $01
; Do MEGA65 / VIC-IV I/O knock
LDA #$47
STA $D02F
LDA #$53
STA $D02F
; End MAP sequence, thus allowing interrupts to occur again
EOM
; Do Hypervisor call to un-write-protect the ROM area
LDA #$70
STA $D640
NOP
```

C65 Compatibility ROM Layout

The layout of the C65 compatibility 128KB ROM area is identical to that of the C65:

HEX	Contents
$3E000 -- $3FFFF	C65 KERNAL
$3D000 -- $3DFFF	CHARSET B
$3C000 -- $3CFFF	RESERVED
$38000 -- $3BFFF	C65 BASIC GRAPHICS ROUTINES
$32000 -- $37FFF	C65 BASIC
$30000 -- $31FFF	MONITOR (gets mapped at $6000 -- $7FFF)
$2E000 -- $2FFFF	C64 KERNAL
$2D000 -- $2DFFF	CHARSET C
$2C000 -- $2C7FF	RESERVED
$2C800 -- $2CFFF	INTERFACE
$2A000 -- $2BFFF	C64 BASIC
$29000 -- $29FFF	CHARSET A
$24000 -- $28FFF	RESERVED
$20000 -- $23FFF	DOS (gets mapped at $8000 -- $BFFF)

The INTERFACE program is a series of routines that are used by the C65 to switch between C64-mode, C65-mode and the C65's built-in DOS. The DOS is located in the lower-eighth of the ROM.