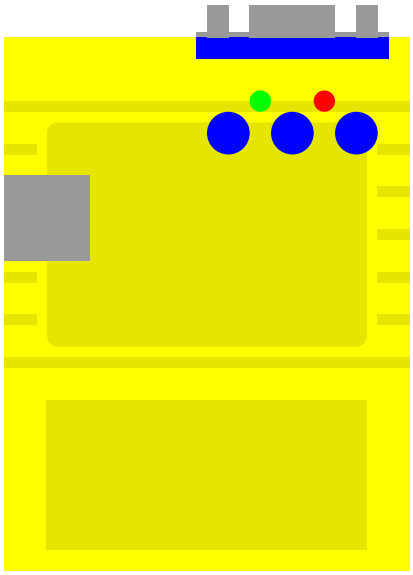
Turbo Chameleon 64

VGA, turbo, freezer and memory expansion for the Commodore-64

The Programmers Manual

Peter Wendrich
pwsoft@syntiac.com

June 22, 2014



Draft version! (BETA-9a)

Contents

1 Introducing the Chameleon core

The Turbo Chameleon FPGA core can run in a few different configurations and so can be used in various ways. The Chameleon cartridge itself can also run other FPGA images (that can provide other machine emulations). This documentation however only covers the C64 mode of the cartridge.

1.1 Turbo Chameleon Cartridge for the C64

This is the main purpose of the core and also where the name 'Chameleon' is comming from. As it is capable of emulating various cartridges and peripherals in a way that is invisible to the software. Most of the functions of the original C64 hardware is taken over by an enhanced emulation in the cartridge. This gives access to all data and address lines, but also internal registers and various control signals normally not accessable on the cartridge port. The CPU can be made to run faster, memory is expanded and various cartridges can be mapped into the address space without changing anything to the main machine.

1.2 Standalone Mode

The Chameleon can also run standalone. That means without being plug'ed into a C64 machine. The cartridge is powered in this mode by the USB connector on the break-out cable. A PS/2 keyboard and/or CDTV remote is required to be able to use the cartridge. The Chameleon function and behavior is almost identical in standalone mode and cartridge mode as all chips of the C64 are cycle accurately emulated.

1.3 Docking Station

The docking-station is an optional module to provide additional connections if the Chameleon is used standalone. The docking-station connects to the same expansion slot used for cartridge mode. The docking-station has connectors for 4 joysticks. Is also allows to use C64 and Amiga keyboards, ofcourse the PS/2 keyboard is also usable in this mode.

2 Configuration Mode

Configuration mode is where the required functionality is selected and additional reigsters and extensions are switched on. The configuration registers are located at 53488 ($D0F0_h$) to 53503 ($D0FF_h$). It is recommended to deactivate configuration mode after the required setting have been made, as some programs could overwrite these registers by accident.

2.1 Detecting a Chameleon

Because the Chameleon can emulate a variety of cartridges and even combinations of those, the normal cartridge type detection method by probing $DExx_h$ or $DFxx_h$ fails to reliably detect it. However if the Chameleon is active, a few extra registers are visible in one of the VIC-II mirror areas. Reading at address 53502 ($D0FE_h$) on a stock machine always results in 255 (FF_h). On the Chameleon the value can be unequal to 255 (FF_h) if the configuration mode is active. Use the following sequence to reliably detect the presence of the Chameleon: Read and backup the current value at 53502 ($D0FE_h$). Write 42 ($2A_h$) at 53502 ($D0FE_h$) and read at the same location. The value represents the FPGA core version. If the backuped value was 255 (FF_h) store it into 53502 ($D0FE_h$) to restore previous mode.

Core version number		Configuraion or Mode
1	01_h	C64 with Chameleon cartridge present
161	$A1_h$	Chameleon running in standalone mode (C64 emulation)
193	$C1_h$	Chameleon core for the C-One reconfigurable computer
209	$D1_h$	Chameleon with docing station (C64 emulation with joysticks)
255	FF_h	C64 without Chameleon

2.2 Activating Configuration Mode

To enter configuration mode and make the setup registers available write the value 42 ($2A_h$) in memory location 53502 ($D0FE_h$). To disable the configuration registers write 255 (FF_h) at this location. Any value written to either 53501 ($D0FD_h$) or 53503 ($D0FF_h$) also leaves configuration mode. Activation of configuration mode is very unlikely to happen by accident as sequencial writes will never or only briefly activate the registers. During configuration mode the extra registers are visible from 53488 ($D0F0_h$) to 53503 ($D0FF_h$). With these registers other memory areas can be configured and additional registers mapped into the CPU address space.

2.3 Reconfigure the FPGA core

The onboard flash has room for upto sixteen FPGA cores. On power-up the core in the first slot (slot 0) is loaded into the FPGA. This core is the Chameleon core and provides the functions for use as expansion cartridge and for standalone emulation of a Commodore 64. Other cores can be loaded however by writing a new slot number into the configuration register at 53502 ($D0FE_h$) or'ed with 16.

To reconfigure the FPGA first enter configuration mode by writing 42 ($2A_h$) at 53502 ($D0FE_h$) followed by a write of the slot number (0-15) or'ed with 16. So 16 (10_h) reloads the Chameleon core, while values 17 to 31 (11_h to $1F_h$) load other cores from the other slots in the onboard Flash. For more information about cores and the flash filesystem refer to the "Core Developers Manual".

2.4 Force menu mode

While configuration mode is active, writing 32 (20_h) at 53502 ($D0FE_h$) enables menu mode. Refer to chapter 9 for more information about this mode. As almost all RAM and ROM can change on the switch to menu mode, the program performing the switch should be running in the memory area from $C000_h$ to $DFFF_h$.

2.5 Force reset from software

To force a reset from software write the value 165 ($A5_h$) into the register at 53502 ($D0FE_h$). Alternatively the value of 166 ($A6_h$) can be used that not only performs a reset, but also disables configuration mode. Both values only work if either configuration or menu mode is active.

3 Core version information

The core version and build date information is available through configuration registers. This information can be used by utility software to determine if the correct core with the required functionality is running. It can also be displayed by the menu-system to help users determining the version when reporting errors.

There are two registers mapped in the MMU configuration space. The MMU registers are located at $D0A0_h$ to $D0AF_h$ and can be activated by setting bit 1 in the configuration register $D0FA_h$. The

VERIDX register at D0A8$_h$ selects which of the configuration characters to read. The VERDAT register at D0A9$_h$ will contain the actual character (encoded in ASCII).

Currently only the first 16 characters have a defined value.

Index	String returned
0 to 7	Core version number padded with spaces (ascii value 32 or 20$_h$). For example "Beta-8j "
8 to 15	Build date of the core in the format YYYYMMDD.
16 to 255	Reserved for future use (returned character for these indexes is undefined).

3.1 Version Registers

Address (Hex)	Address (Dec)	Name	Description
D0A8$_h$	53416	VERIDX	Version data character index.
D0A9$_h$	53417	VERDAT	Read out of character indexed by VERIDX.

4 Memory

The Chameleon Cartridge brings its own memory. The internal memory of the C64 is not used except for the color ram. Because the CPU and VIC-II chip can only access 64 Kbyte at a time, a few tricks are required to address more. There are different methods implemented, so the best one can be choosen for each purpose.

By far the fastest method to move large amount of data around in a compatible way is using the REU emulation. The REU is a DMA engine that can copy or compare blocks of data at a speed of 1 Mbyte per second.

A MMU is provided to allows splitting the C64 memory into sixteen segments (banks) of 4 Kbyte each. Each of these 4 Kbyte pages can be placed at any byte offset in memory. There are additional MMU slots for specifing the location of ROMs. This gives support for ROM replacements, emulation of various freezer cartridges and can even be used to implement multitasking.

The CPU is not the only device using memory. The REU emulation was already mentioned, which can use upto 16 Mbyte of storage space. The drive emulation needs memory for the disk images and a bit of work memory. Also the VGA video port uses quite a bit of memory for the framebuffer.

4.1 32 MByte Memory Layout

This is the memory layout used by the Turbo Chameleon Cartridge both as cartridge and in standalone mode. The area 0100000$_h$ to 019FFFF$_h$ is read from onboard Flash-ROM during startup (640 KByte total) by bootloader code inside the FPGA. After loading from Flash-ROM, the Chameleon is switched to either menu-mode or standard C64 mode depending on the status of the arrow-left key.

The MMU blocks 00$_h$ to 0F$_h$ are mapped to memory from 000 0000$_h$ to 000 FFFF$_h$. MMU block 1F$_h$(Kernal ROM) is mapped to 010 E000$_h$ and 1E$_h$(BASIC ROM) is mapped to 010 A000$_h$. MMU block 1D$_h$(character ROM) is mapped to 010 D000$_h$. This emulates the standard C64 memory layout. For the menu mode, four additional blocks are mapped (20$_h$, 24$_h$, 25$_h$ and 26$_h$). See following table where they map. Any additional setup for the menu system must be done by the code located at 010 0000$_h$–010 7FFF$_h$.

Address (Hex)	Address (Dec)	Name	Description
000 0000_h –000 $FFFF_h$			64 KByte RAM for C64 mode (MMU banks 00_h–$0F_h$)
001 0000_h –00F $FFFF_h$			960 Kbyte RAM free
010 0000_h –010 $1FFF_h$			Initial Menu 0000-1FFF (MMU bank 20_h)
010 2000_h –010 $3FFF_h$			Initial Menu 8000-9FFF (MMU bank 24_h)
010 4000_h –010 $5FFF_h$			Initial Menu A000-BFFF (MMU bank 25_h)
010 6000_h –010 $7FFF_h$			Initial Menu E000-FFFF (MMU bank 26_h)
010 8000_h –010 $9FFF_h$			MMC64 BIOS image
010 $A000_h$–010 $BFFF_h$			BASIC V2 ROM image (MMU $1E_h$)
010 $C000_h$–010 $CFFF_h$			*** reserved 4K ****
010 $D000_h$–010 $DFFF_h$			Character ROM image (MMU $1C_h$)
010 $E000_h$–010 $FFFF_h$			Kernal ROM image (MMU $1F_h$)
011 0000_h –0FF $FFFF_h$			*** reserved for menu system ****
100 0000_h –1FF $FFFF_h$			16 MByte REU memory

4.2 16 MByte Memory Layout

This is the memory layout used by the C-One Chameleon core. The C-One extender board has 16 MByte of memory.

Address (Hex)	Address (Dec)	Name	Description
00 0000_h –00 $FFFF_h$			64 KByte RAM for C64 mode
01 0000_h –0F $FFFF_h$			960 Kbyte RAM free for user programs
10 0000_h –10 $9FFF_h$			*** unused ***
10 $A000_h$–10 $BFFF_h$			BASIC V2 ROM image (MMU $1E_h$)
10 $C000_h$–10 $CFFF_h$			*** reserved 4K ****
10 $D000_h$–10 $DFFF_h$			Character ROM image (MMU $1C_h$)
10 $E000_h$–10 $FFFF_h$			Kernal ROM image (MMU $1F_h$)
11 0000_h –3F $FFFF_h$			*** unused ***
40 0000_h –7F $FFFF_h$			4 Mbyte video RAM
80 0000_h –FF $FFFF_h$			8 MByte REU memory

4.2.1 Bootrom on C-One

The bootrom on C-One needs access to the kernal, character and basic roms to copy them into SDRAM. As there is no flash chip like on the Chameleon platform they are included in the fpga image. When the bootrom is active the C64 layout is changed to the following to get access to the ROMs. Once the boot phase is completed these ROMs become invisible to the running system.

Address (Hex)	Address (Dec)	Name	Description
0000_h –$5FFF_h$			Normal layout
6000_h –$7FFF_h$			Kernal ROM (8k)
8000_h –$9FFF_h$			Normal layout
$A000_h$–$BFFF_h$			Basic ROM (8k)
$C000_h$–$DFFF_h$			Normal layout
$E000_h$–$EFFF_h$			Character ROM (4k)
$F000_h$–$FFFF_h$			Boot ROM (4k)

4.3 MMU Registers

The MMU has 256 slots that store 25 bit wide address offsets in memory. These offsets are the start address in SDRAM memory of the corresponding bank. This start address of each bank can be positioned anywere in memory at any byte offset. This makes the MMU more flexible as a simple banking scheme as they don't have to start at a multiple of the bank size. This allows MMU banks to overlap and point to shared memory (with each MMU bank at a possibly different offset).

Some slots have specific functions or regions, others are free assignable. By updating the offsets everything can be moved and relocated freely in memory. The MMU registers are located at $D0A0_h$ to $D0AF_h$ and can be activated by setting bit 1 in the configuration register $D0FA_h$. When changing an offset, first select the required slot by writing the slot-number into $D0AF_h$. Then the offset can be read or changed with the registers from $D0A0_h$ to $D0A3_h$.

The first 16 slots have offsets for the 64Kbyte memory that the 6510 and VIC-II can see. Each of the 16 slots specifies the location of a 4 KByte segment.

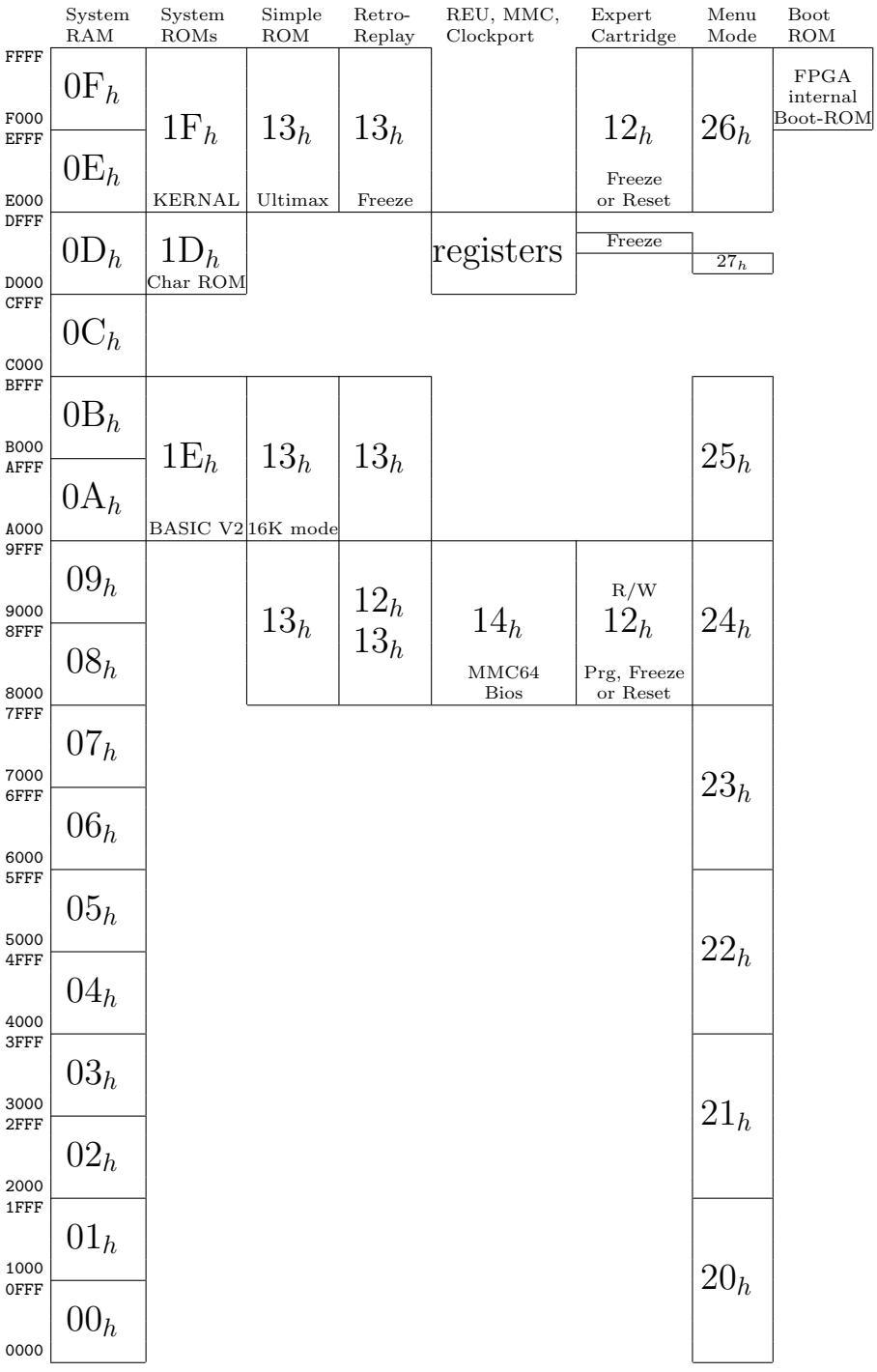
Address (Hex)	Address (Dec)	Name	Description
$D0A0_h$	53408	MMUA0	Address offset bits A_7–A_0 of current MMU slot
$D0A1_h$	53409	MMUA1	Address offset bits A_{15}–A_8 of current MMU slot
$D0A2_h$	53410	MMUA2	Address offset bits A_{23}–A_{16} of current MMU slot
$D0A3_h$	53411	MMUA3	Address offset bit A_{24} of current MMU slot

bit	settings	description
7	read-only	0 = Block of memory can be read and written 1 = Block of memory is read-only
6–1	Reserved for address extension, must be set to 0	
0	Address offset bit A_{24}	

$D0AF_h$	53423	MMUSLT	Select MMU slot

bit	settings	description
7–0	Current slot	00_h= C64 r/w memory at $0xxx_h$ 01_h= C64 r/w memory at $1xxx_h$ 02_h= C64 r/w memory at $2xxx_h$ 03_h= C64 r/w memory at $3xxx_h$ 04_h= C64 r/w memory at $4xxx_h$ 05_h= C64 r/w memory at $5xxx_h$ 06_h= C64 r/w memory at $6xxx_h$ 07_h= C64 r/w memory at $7xxx_h$ 08_h= C64 r/w memory at $8xxx_h$ 09_h= C64 r/w memory at $9xxx_h$ $0A_h$= C64 r/w memory (under basic) at $Axxx_h$ $0B_h$= C64 r/w memory (under basic) at $Bxxx_h$ $0C_h$= C64 r/w memory at $Cxxx_h$ $0D_h$= C64 r/w memory (under I/O) at $Dxxx_h$ $0E_h$= C64 r/w memory (under kernal) at $Exxx_h$ $0F_h$= C64 r/w memory (under kernal) at $Fxxx_h$ 10_h= REU internal memory (upto 16 MByte) 11_h= geoRAM internal memory (upto 4 MByte) 12_h= Freezer/Game cartridge RAM 13_h= Freezer/Game cartridge ROM 14_h= MMC64 cartridge ROM (8 KByte) 15_h= *** reserved *** 16_h= *** reserved *** 17_h= *** reserved for tape *** 18_h= Drive 8 RAM/ROM (64 KByte) 19_h= Drive 9 RAM/ROM (64 KByte) $1A_h$= *** reserved for drive 9 *** $1B_h$= *** reserved *** $1C_h$= VIC-II Frame-buffer location $1D_h$= character ROM (4 KByte) $1E_h$= ROM at $A000_h$-$BFFF_h$ (BASIC, 8 KByte) $1F_h$= ROM at $E000_h$-$FFFF_h$ (KERNAL, 8 KByte) 20_h= C64 r/w memory at 0000_h-$1FFF_h$ in menu-mode 21_h= C64 r/w memory at 2000_h-$3FFF_h$ in menu-mode 22_h= C64 r/w memory at 4000_h-$5FFF_h$ in menu-mode 23_h= C64 r/w memory at 6000_h-$7FFF_h$ in menu-mode 24_h= C64 r/w memory at 8000_h-$9FFF_h$ in menu-mode 25_h= C64 r/w memory at $A000_h$-$BFFF_h$ in menu-mode 26_h= C64 r/w memory at $E000_h$-$FFFF_h$ in menu-mode 27_h= ROM or RAM at $D700_h$-$D7FF_h$ 28_h= Drive 8 Disk tracks for virtual floppy 1 29_h= Drive 8 Disk tracks for virtual floppy 2 $2A_h$= Drive 8 Disk tracks for virtual floppy 3 $2B_h$= Drive 8 Disk tracks for virtual floppy 4 $2C_h$= Drive 9 Disk tracks for virtual floppy 1 $2D_h$= Drive 9 Disk tracks for virtual floppy 2 $2E_h$= Drive 9 Disk tracks for virtual floppy 3 $2F_h$= Drive 9 Disk tracks for virtual floppy 4 30_h–FF_h= *** Free for applications ***

4.4 Memory Overlays (6510 CPU)



The large hexidecimal number represent the MMU bank that is assigned to that segment. Some MMU banks are assigned to two segments. In that case the segments are located behind each other in that bank with the segment with the lowest address first.

5 Buttons

There are three buttons on the Chameleon. The functions they perform, can be changed by software. The default functions are:

- Left, Chameleon menu

- Middle, short press is Freeze, long press accesses Chameleon menu
- Right, short press is Reset, long press restarts the boot-ROM and reloads OS

5.1 Buttons Configuration Register

Address (Hex)	Address (Dec)	Name	Description
$D0FB_h$	53499	CFGBTN	Debug info and Buttons

bit	settings	description
7-6	Debug info on VGA	00 = No debug information 01 = Show memory and cache load and also main 6510 CPU state on the top of the screen. 10 = Show memory, cache, 6510 and drive CPU state. 11 = Show all debug information (note this uses a considerable amount of screen space).
5–4	Reserved, must be 0	
3–0	Left button configuration	

	short	long
0000	Menu	–
0001	Cartridge On/Off	Cartridge Prg (expert)
0010	Toggle Turbo On/Off	–
0100	Disk change drive 8 (next)	Disk change drive 8 (first)
0101	Disk change drive 9 (next)	Disk change drive 9 (first)
others	*** reserved ***	

5.2 Last Button Pressed

There are different ways to enter the menu. The LSTBTN register reports the last button pressed by the user. It allows the menu to perform different actions depending on the button used to enter the menu system. The register is located at 53422 ($D0AE_h$) and can be activated by setting bit 1 in the configuration register $D0FA_h$.

A write to the register (any value) will reset the register back to 0.

Address (Hex)	Address (Dec)	Name	Description
$D0AE_h$	53422	LSTBTN	Informs menu of the last button pressed

bit	settings	description
7–3	reserved, must be 0	–
2–0	Last button	000 = Nothing or unknown 010 = Left button pressed short 011 = Left button pressed long 100 = Middle (freezer) button pressed short 101 = Middle (freezer) button pressed long 110 = Right (reset) button pressed short 111 = Right (reset) button pressed long

6 VGA Output

One of the major features on the Turbo Chameleon Cartridge is the VGA connector. This interface allows rendering of the C64 picture on a VGA monitor in high quality. It doesn't use the original PAL or NTSC output, but generates the picture by monitoring the address and databus of the expansion connector. This results in a crisp and perfect stable picture on the monitor.

6.1 VGA Sync

The VGA controller can be asynchronous with the VIC-II chip where the vertical frequency of the picture can be almost anything from 50 Hertz upto 85 Hertz in the current firmware. This gives a lot of choices for the VGA resolutions, but gives some artifacts (jittery movement) for fast moving objects on the screen. Therefore the VGA controller can also run synchronously with the VIC-II

chip. Here one frame of the VIC-II corresponds with one frame on the VGA. Animations will be smooth and certain time depended graphic effects (like interlace) always will look correct. However VGA vertical frequency is now fixed and depends on the type of VIC-II chip. For PAL machines the VGA will need to run at 50 Hertz vertical frequency and for NTSC machines at 60 Hertz. Make sure the correct VGA mode is active when enabling synchronisation. The VGA controller can normally synchronise on any selected mode, but if the choosen VGA vertical frequency is not correct the height of the picture will be reduced (resulting in a wrong aspect ratio).

6.2 Frame buffers

When the VGA controller is in asynchronous mode there is a frame-buffer to prevent screen tearing artifacts. Standard is double-buffering where the screen is updated in a second buffer and swapped when the VGA starts a new redraw. Some C64 demos and applications use interlace effects where two screens are quickly alternated to get more colors or higher resolutions. When the VGA is not synchronised it start flickering horribly as the two different screens are not displayed at the proper intervals. For these cases there is the option to do tripple-buffering. In this mode two buffers are combined by blending them together. This prevents the tearing effect and also displays interlaced pictures correctly. There is a trade-off here as quick moving objects will look a bit fussy (motion blur) due to the averaging over two frames.

6.3 Scaling modes

As the VGA has a higher resolution as the C64 screen the picture is scaled up to fit the screen. The method for scaling can be configured. The default is "nearest neighbor" that simply repeats the pixels and lines. A more advanced scaler is the "Scale-2x" algorithm. This algorithm invents new pixels so the graphics appear to have a higher resolution. The "Scale-2x" algorithm only works correctly for video modes that double the pixels (which is true for 800x600). For any resolution other than 800x600 it is recommended to always use nearest neighbor scaling to prevent scaling artifacts.

The 640x480 resolutions use a scale factor of 1.5, which does not deliver a very good quality. Only use this resolution if the monitor used does not support any higher resolution.

6.4 Scanline emulation

Duplicated lines on the VGA can be rendered at reduced brightness. This emulates the behavior of old CRT screens. The lines are only dimmed and not completely black so the VGA does not become too dim. The bits 5 and 6 of the config register 53313 ($D041_h$) determine the brightness of the duplicated lines. The scanline effect can be enabled together with the Scale-2x scaling algorithm if so desired.

6.5 VGA Registers

Address (Hex)	Address (Dec)	Name	Description
$D040_h$	53312	VGAMOD	Set VGA mode

bit	settings	description
7–0	Current VGA mode	00_h= 800x600 72 Hz (default) 01_h= 800x600 50 Hz 02_h= 800x600 60 Hz 03_h= 800x600 75 Hz 04_h= 640x480 50 Hz 05_h= 640x480 60 Hz 06_h= 640x480 72 Hz 07_h= 640x480 75 Hz 08_h= 640x480 85 Hz 09_h= 1024x768 50 Hz $0A_h$= 1024x768 60 Hz $0B_h$= 1024x768 72 Hz $0C_h$= 1024x768 75 Hz $0D_h$= 1024x768 85 Hz $0E_h$= 1152x864 50 Hz $0F_h$= 1152x864 60 Hz 10_h= 1152x864 72 Hz 11_h= 1152x864 75 Hz 12_h= 1152x864 85 Hz 13_h= 1280x1024 50 Hz 14_h= 1280x1024 60 Hz 15_h= 1280x1024 72 Hz 16_h= 1280x1024 75 Hz 17_h= 1280x1024 85 Hz 18_h= 1600x1200 50 Hz 19_h= 1600x1200 60 Hz

$D041_h$	53313	VGACFG	Set VGA configuration

bit	settings	description
7	VIC-II Sync	0 = VGA asynchronous 1 = VGA synchronized to VIC-II chip.
6-5	Scanlines emulation	00 = All lines at full brightness. 01 = Doubled lines have 75% brightness to simulate scan-lines. 10 = Doubled lines have 50% brightness to simulate scan-lines. 11 = Doubled lines have 25% brightness to simulate scan-lines.
4–2	Scaling mode	000_b= Nearest Neighbor scaling. 001_b= Scale-2x 010_b= Alien space ship 011_b= Reserved for future use 100_b= Reserved for future use 101_b= Reserved for future use 110_b= Reserved for future use 111_b= Reserved for future use
1–0	Frame buffering mode	00_b= Reserved for future use 01_b= Double buffering 10_b= Tripple buffering (50%,50% blending for IFLI) 11_b= Reserved for future use

$D042_h$	53314	VGACOL	Set VGA color mode

bit	settings	description
7-1	Reserved, must be 0	–
0	Color palette	0 = Use standard color palette. 1 = Use custom palette.

$D043_h$	53315	VGARQT	Request info about VGA modes (result in $D044_h$–$D047_h$)
$D044_h$	53316	VGAW	VGA mode info $vga\_width_{7..0}$
$D045_h$	53317	VGAH	VGA mode info $vga\_lines_{7..0}$
$D046_h$	53318	VGAHWH	VGA mode info width and lines high bits

bit	settings	description
7–4	total $vga\_lines_{11..8}$	
3–0	total $vga\_width_{11..8}$	

$D047_h$	53319	VGAHZ	VGA mode vertical frequency in Hertz
$D048_h$	53320		Reserved
$D049_h$	53321		Reserved
$D04A_h$	53322		Reserved
$D04B_h$	53323		Reserved
$D04C_h$	53324		Reserved
$D04D_h$	53325		Reserved
$D04E_h$	53326		Reserved
$D04F_h$	53327		Reserved

6.6 Palette Registers

NOTE DUE TO RECENT CHANGES IN THE VGA CONTROLLER THE CURRENT BETA FIRMWARE (BETA 8) DOES NOT SUPPORT PALETTE CHANGES. The 768 registers are still implemented for software compatibility.

To support higher color depths on the VGA, a set of registers is added to store custom colors. The 'palette offset' register in the Object-Processor select which of the colors of the palette are being used. The first 32 entries in the color palette are fixed. Entries 32 (020_h) to 287 ($11F_h$) are software redefinable by using the palette registers (note that entries 271 to 287 are only reachable in 256 color mode).

When bit 0 of configuration register $D0FA_h$ is set, an additional 768 registers become available at memory locations $D100_h$ to $D3FF_h$. The registers at $D1xx_h$ store the red color intensities. The next 256 registers at $D2xx_h$ store the green intensity of the colors and the last 256 at $D3xx_h$ store the blue intensity value of the RGB triplets. Although the color resolution is limited to 5 bits (bit 7–3), all 8 bits are stored so the palette registers can also be used as 768 bytes of extra memory.

Address (Hex)	Address (Dec)	Name	Description
$D100_h$–$D1FF_h$	53504 –53759	PALRED	256 entry color palette **Red** intensity
$D200_h$–$D2FF_h$	53760 –54015	PALGRN	256 entry color palette **Green** intensity
$D300_h$–$D3FF_h$	54016 –54271	PALBLU	256 entry color palette **Blue** intensity

6.7 Fixed Palette Entries

NOTE DUE TO RECENT CHANGES IN THE VGA CONTROLLER THE CURRENT BETA FIRMWARE (BETA 8) DOES NOT SUPPORT PALETTE CHANGES.

The first 32 entries in the color palette are fixed. They contain VIC-II and VDC compatible color definitions. Palette entries 0 (000_h) to 15 ($00F_h$) contain VIC-II compatible colors. Palette entires 16 (010_h) to 31 ($01F_h$) contain RGBI entries compatible with the VDC chip that is found in Commodore 128 machines. Custom color entries start at palette index 32 (020_h) with the last entry at index 287 ($11F_h$).

Palette Index	Color (VIC-II)	Palette Index	Color (VDC)
0 (000_h)	black	16 (010_h)	black
1 (001_h)	white	17 (011_h)	dark gray
2 (002_h)	red	18 (012_h)	dark blue
3 (003_h)	cyan	19 (013_h)	light blue
4 (004_h)	purple	20 (014_h)	dark green
5 (005_h)	green	21 (015_h)	light green
6 (006_h)	blue	22 (016_h)	dark cyan
7 (007_h)	yellow	23 (017_h)	light cyan
8 (008_h)	orange	24 (018_h)	dark red
9 (009_h)	brown	25 (019_h)	light red
10 ($00A_h$)	light red	26 ($01A_h$)	dark purple
11 ($00B_h$)	dark gray	27 ($01B_h$)	light purple
12 ($00C_h$)	mid gray	28 ($01C_h$)	dark yellow (brown)
13 ($00D_h$)	light green	29 ($01D_h$)	yellow
14 ($00E_h$)	light blue	30 ($01E_h$)	light gray
15 ($00F_h$)	light gray	31 ($01F_h$)	white

7 VGA Status Lines

Chameleon can display up to three status lines on the VGA screen. When one or more status lines are enable also two colord bars appear that show the memory performance in a graphical representation. The color bars are split in sixteen equal segment, so each segment represents $6\frac{1}{4}$

percent. The top bar represents the current load placed by the system on the SDRAM memory. The bottom bar represents the cache miss rate in percent. When the cache controller has many misses it will cause the SDRAM load to increase as well. Idle values (e.g. READY prompt in basic) for the two bars are around three segments for the top bar (18 percent) and no more than one segment for the cache miss rate. When running graphic intensive applications the system load can increase significantly. Also the turbo function can cause many cache misses as the CPU will perform about 10 times as many memory accesses compared to 1 Mhz mode.

The first status line displays the status of the main CPU (6510). From left to right it displays the following values:

- The current Program Counter (PC)
- The opcode currently executed (IR)
- Contents of the accumulator (A)
- Contents of the index X register (X)
- Contents of the index Y register (Y)
- Position of the stack-pointer (S)
- Processor flags zero is flag cleared, N = negative flag set, V=Overflow flag set, D=Decimal flag set, I=Interrupts disabled, Z=Zero flag set, C=Carry set
- Values of memory locations 0 and 1, which control the IO lines on the CPU (IO). Both addresses are combined into a single value that represents the real memory layout (calculation is IO_1 OR (NOT IO_0)). Clearing the direction register (set to input) causes the output to become high due to pull-ups in the machine.
- CPU speed in percentage relative to the phi-2 clock. Numbers below 100 percent indicate that the CPU is slowed down by badlines or sprite fetches. If the turbo mode is active, numbers much larger as 100 can be seen.

The next two lines (if enabled) show the status for the two drive CPUs. The first fields are the same as for the main CPU. The two digits with a slash in between represent the current selected disk-image and maximum loaded disk-images. The last number is the current disk track represented as decimal number in the range 1 to 40.

7.1 VGA Status Configuration Register

Address (Hex)	Address (Dec)	Name	Description
$D0FB_h$	53499	CFGBTN	Debug info and Buttons

bit	settings	description
7-6	Debug info on VGA	00 = No debug information 01 = Show memory and cache load and also main 6510 CPU state on the top of the screen. 10 = Show memory, cache, 6510 and drive CPU state. 11 = Show all debug information (note this uses a considerable amount of screen space).
5–4	Reserved, must be 0	
3–0	Left button configuration	

	short	long
0000	Menu	–
0001	Cartridge On/Off	Cartridge Prg (expert)
0010	Toggle Turbo On/Off	–
0100	Disk change drive 8 (next)	Disk change drive 8 (first)
0101	Disk change drive 9 (next)	Disk change drive 9 (first)
others	*** reserved ***	

8 Cartridge Emulation

The Chameleon occupies the expansion connector and can not share it with any other cartridge(s). Fortunately it can emulate many types of cartridges. Even some combinations of different cartridges can be emulated. A few of these combinations are special as such that these are normally not possible to be used in a single machine without tricks.

8.1 Freezer Logic

Chameleon contains a generic freezer implementation, the same logic is used for all the available freezer cartridge emulations. The freezer logic in Chameleon is often more reliable as the kludgy logic used originally in many of the cartridges.

The freezer emulation can successfully freeze programs that have interrupts disabled or force the NMI line low. It also properly waits for the acknowledge of the interrupt before enabling the freezer ROMs. Although some programs might not function correctly after a restart, it is impossible for an application to prevent the freeze action itself.

8.2 Clock port

This is not really a separate cartridge type, but a part of other cartridges. The clockport is an interface that originally comes from the Amiga 1200 computer, but can be found on many Commodore 64 cartridges as well. It allows small add-on cards to be easily connected to the machine. The shape of the Chameleon PCB and casing is designed for an optional RR-Net ethernet adapter. Many other addons don't fit properly as they were designed for different hardware.

As the Chameleon can emulate multiple cartridges that have their own (conflicting) clockport settings, the configuration for this port is moved to a Chameleon specific register. The clockport configuration bits in the register maps of various cartridges are therefore not emulated.

8.3 Simple ROM cartridges

These are simple cartridges with an EPROM, an optional on/off switch and sometimes one logic chip. These type of cartridges are often used for utilities like tape speeders, assemblers and machine-monitors or for small games. There are three different cartridge layouts that can be configured.

- 8 Kbyte ROM at 8000_h to $9FFF_h$. If the ROM doesn't support autostart, the machine will report 30719 basic bytes free.
- 16 Kbyte ROM at 8000_h to $BFFF_h$. This type of cartridge replaces the BASIC interpreter ROM to get 8 Kbyte more ROM space.
- 16 Kbyte ROM at 8000_h to $9FFF_h$ and $E000_h$ to $FFFF_h$. This type of cartridge replaces Kernal ROM to get 8 Kbyte more ROM space. This configuration is known as ultimax and changes the memory layout as well.

The 8 Kbyte configuration is the most common. Some games cartridges are using the 16 Kbyte variants if they need more as 8 Kbyte of ROM space. Some Kernal ROM replacement cartridges also use a 16 Kbyte ROM layout (Ultimax), but have some extra logic on the PCB to keep the normal RAM layout. These type of cartridges can not be emulated, but the Kernal ROM can be replaced much easier on Chameleon by simply reprogramming the MMU. Changing the address for slot $1F_h$ in the MMU has the same effect as replacing the ROM inside the machine and is completely transparent for any software.

CRT files containing Simple 8 or 16 KByte ROMs should have 0 (00_h) as CRT ID.

8.4 MMC64

Address (Hex)	Address (Dec)	Name	Description
$DF10_h$	57104	MMCSPI	SPI transfer register. Write in this register sends byte to SPI bus, read is last retrieved byte.
$DF11_h$	57105	MMCCTL	MMC64 Control register.

bit	settings	description
7	MMC64 active	0 = MMC64 is active 1 = MMC64 is disabled Bit can only be modified when unlocked
6	SPI trigger mode	0 = Trigger SPI transfer on write to register $DF10_h$ 1 = Trigger SPI transfer on read of register $DF10_h$
5	External ROM	0 = Allow external ROM when BIOS is disabled 1 = Disable external ROM
4	Flash mode	0 = Normal mode 1 = Flash update mode Not implemented, must be set to 0
3	Clock port address	Not implemented, must be set to 0
2	Clock Speed	0 = 250 KHz SPI clock 1 = 8 Mhz SPI clock
1	MMC cart select	0 = Cart selected 1 = Cart not selected
0	MMC64 Bios	0 = MMC64 BIOS ROM active 1 = BIOS ROM disabled (external ROM active)

$DF12_h$	57106	MMCST	MMC64 Status register (read-only).

bit	settings	description
5	Flash jumper	Not implemented reads always as 0
4	MMC Write Protect	0 = Cart can be written 1 = Cart is write protected
3	MMC Cart Detect	0 = Cart inserted 1 = No cart present, slot empty
2	External EXROM line	
1	External GAME line	
0	Busy	0 = SPI bus ready 1 = SPI bus busy (only for 250 Khz mode)

8.4.1 MMC64 additional SPI devices

Same as MMC64, but with unused bit 4 combined with card select to have access to three SPI devices: MMC cart, FlashRom or RTC (Real Time Clock). When accessing the RTC, the transfer speed must be set to 250 Khz. The RTC device is too slow to accept data at 8 Mhz. The FlashRom is fast enough to be accessed in 8 Mhz mode.

Refer to the datasheets of the flash and RTC chips for programming information and usage.

Address (Hex)	Address (Dec)	Name	Description
$DF11_h$	57105	MMCCTL	MMC64 Control register.

bit	settings	description
4,1	Select	00 = MMC cart selected 01 = Nothing selected 10 = Flash ROM selected 11 = RTC (Real Time Clock) selected

8.5 RAM expansions

The standard internal memory of 64 Kbyte of the Commodore 64 is not always enough. Therefore some memory expansions have been developed.

- RAM Expansion Unit (REU)
- geoRAM

The operating system GEOS was one of the first programs to support the REU. Because the REU was difficult to obtain, the company behind GEOS made their own expansion called geoRAM. The REU has a buildin DMA engine that the geoRAM module lacks. This makes the REU the better expansion option and has also slightly more software support. Chameleon can emulate both the

REU and geoRAM. The registers of the two expansions don't overlap and therefore can even be activated at the same time.

8.5.1 REU (Ram Expansion Unit) 1700, 1750, 1764

The memory of the REU is not directly visible in the address space of the C64. The REU transfers blocks of data to or from its onboard RAM by using DMA. While the transfer is in progress the CPU is stopped. The REU copies and compares at a speed of 1 Mbyte/second (memory swaps run at half that speed), but like the processor it will stop on badlines when the VIC-II video chip needs the extra memory cycles.

8.5.2 REU Emulated Quirks

The REU was only designed to handle upto 512k. However some custom modifications have allowed REUs to handle larger sizes (1 and 2 Mbyte), however the internal address counter still counts only 19 bits. So on these REUs when crossing the 512k mark the 19 bit counter wraps around and the upper address bits stay at their value. This wraparound is emulated in the Chameleon for REUs sizes from 128k to 2 Mbyte. When the REU is set to 4,8 or 16 Mbyte the Chameleon emulation provides a full counter (22, 23 or 24 bits) to easy software development.

TODO: Describe 16 bit reload bug when writing half of address or length registers.

8.5.3 REU Registers

Address (Hex)	Address (Dec)	Name	Description
$DF00_h$	57088	DMAST	REU Status register (read-only)

bit	settings	description
7	1 = IRQ pending	
6	1 = End of block	
5	1 = Fault	Compare operation detected a difference
4	Size	0 = 128 KByte 1 = 256 or 512 KByte A single bit can't represent all possible memory sizes. So software should probe for the amount that is really available.
3–0	Version	Always 0000

Address (Hex)	Address (Dec)	Name	Description
$DF01_h$	57089	DMACMD	REU Command register

bit	settings	description
7	1 = Execute	
6	Reserved	–
5	1 = Auto load	When autoload is enabled. The memory pointers and length registers are reloaded at the end of the transfer
4	$FF00_h$ flag	0 = Wait for write to $FF00_h$ before starting transfer 1 = Start immediately when bit 7 becomes set
3–2	Reserved	–
1–0	Transfer type	00 = C64 to REU 01 = REU to C64 10 = Swap 11 = Compare / verify

$DF02_h$	57090	DMA64L	C64 memory pointer low
$DF03_h$	57091	DMA64H	C64 memory pointer high
$DF04_h$	57092	DMAINL	REU memory pointer low
$DF05_h$	57093	DMAINM	REU memory pointer mid
$DF06_h$	57094	DMAINH	REU memory pointer high
$DF07_h$	57095	DMACNL	Transfer length low
$DF08_h$	57096	DMACNH	Transfer length high
$DF09_h$	57097	DMAINT	Interrupt mask register

bit	settings	description
7	Interrupt enable	1 = enabled
6	End Of Block mask	1 = interrupt after transfer
5	Verify mask	1 = interrupt on verify error
4–0	Reserved	Read as 1

Address (Hex)	Address (Dec)	Name	Description
$DF0A_h$	57098	DMACTL	Address control register

bit	settings	description
7	C64 Address control	0 = Increment C64 address 1 = Fixed C64 address
6	REU Address control	0 = Increment REU address 1 = Fixed REU address
5–0	Reserved	Read as 1

8.5.4 GeoRAM registers

There is a 256 byte large window at $DE00_h$–$DEFF_h$ to access the GeoRAM memory. Two registers at $DFFE_h$ and $DFFF_h$ select the position of the memory window. The registers are write only. A read can return any random value. On reset the registers are cleared to zero.

The GeoRAM register layout allows upto 4 Mbyte of internal memory (its location in memory is determined by MMU bank 17 (11_h). The real GeoRAM cartridge has only 512 KByte, but some clone hardware have appeared with more memory (NeoRAM). The emulation in Chameleon can be configured from 64KByte upto 4 MByte in powers of two.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$ -$DEFF_h$	56832 –57087	GEOBUF	geoRAM 256 byte memory window
$DFFE_h$	57342	GEOLOW	geoRAM address A_{13}–A_8

bit	settings	description
7–6	Unused	must be set to 0
5–0	geoRAM A_{13}–A_8	

$DFFF_h$	57343	GEOHI	geoRAM address A_{21}–A_{14}

8.6 Action Replay / RetroReplay

Chameleon can emulate the RetroReplay hardware. This is a freezer cartridge developed by Individual Computers and is an improvement on and backwards compatible with the Action Replay. The RetroReplay cartridge provides access to 64 KByte of ROM and 32 KByte of RAM. The real cartridge has two ROMs of 64 KByte that can be selected with a hardware jumper, this is not emulated as the MMU in Chameleon can provide similar functionality.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$	56832	RRCTRL	RR control register (on write)

bit	settings	description
7	A15	ROM address line 15
6		
5	ROM/RAM	0 = ROM 1 = RAM
4	A14	ROM/RAM address line 14
3	A13	ROM/RAM address line 13
2	Disable	Write 1 to disable cartridge
1	EXROM	
0	GAME (inverted)	

$DE01_h$	56833	RREXTD	RR extended control register (on write)

bit	settings	description
7	A15	ROM address line 15 (mirror of $DE00_h$)
6	REU Compatibility	0 = Standard memory map 1 = REU compatible memory map
5		Not implemented, must be set to 0
4	A14	ROM/RAM address line 14 (mirror of $DE00_h$)
3	A13	ROM/RAM address line 13 (mirror of $DE00_h$)
2		Not implemented, must be set to 0
1	AllowBank	0 = no RAM banking in $DE02_h$–$DFFF_h$ area 1 = Enable RAM banking in $DE02_h$–$DFFF_h$ area
0		Not implemented, must be set to 0

$DE00_h$–$DE01_h$	56832 –56833	RRSTAT	RR status (on read)

bit	settings	description
7	A15	ROM address line 15
6		
5		Not implemented, reads 0
4	A14	ROM/RAM address line 14
3	A13	ROM/RAM address line 13
2		
1		
0		Not implemented, reads 0

The ROM/RAM switch determines if memory or ROM is visible at 8000_h–$9FFF_h$ and at $DE00_h$–$DFFF_h$. The memory locations $A000_h$–$BFFF_h$ and $E000_h$–$FFFF_h$ always map ROM and are not affected by this bit. Although the ROM can be located at different memory addresses only 8 Kbyte is available at any time. When more as one location is activated they are mirrors of each other. The ROM has 8 banks of 8 Kbyte for a total of 64 KByte ROM. The RAM only has 4 banks for a total of 32 KByte RAM.

The lowest two bits of the configuration register at $DE00_h$ determine where in memory the ROM or RAM of the cartridge is visible. Take note that the control of the GAME line is inverted. After reset the register is cleared so the first 8 Kbyte of the ROM is visible at 8000_h-$9FFF_h$. The "CBM80" signature in the ROM makes the kernel jump into the cartridge and this will display the startup menu.

EXROM bit 1	GAME (inverted) bit 0	ROM Mapping
0	0	8 KByte at 8000_h–$9FFF_h$
0	1	8 KByte ROM/RAM at 8000_h–$9FFF_h$ and 8 Kbyte ROM at $A000_h$–$BFFF_h$
1	0	Cartridge ROM/RAM disabled.
1	1	Ultimax mode, ROM/RAM at 8000_h–$9FFF_h$ and ROM at $E000_h$–$FFFF_h$.

The RetroReplay has a slight incompatibility (by design) compared to the original Action-Replay cartridge. When writing to the RAM on the Action Replay at 8000_h–$9FFF_h$ it will also write to the internal C64 memory at the same address. On the RetroReplay (and its emulation in Chameleon) a write operation will only write to the cartridge RAM, leaving the C64 memory below it intact.

CRT files containing ActionReplay or RetroReply ROMs should have 1 (01_h) as CRT ID.

8.7 KCS Power Cartridge

The KCS Power catridge is an utility, disk-speed and freezer cartridge. It has (only) 16 KByte ROM and 128 bytes of RAM. For the actually limited amount of ROM the cartridge feature set was OK-ish. However the limited amount of RAM meant you can not restart the program after entering the machine monitor after freezing.

It doesn't have any configuration registers in the usual sense of the word. Mode switching of the cartridge is controlled by address lines when accessing I/O space at $DE00_h$–$DFFF_h$. So jumps to certain locations in I/O space allowed the cartridge to become active and jump into its ROM.

Writing to any address in $DE00_h$–$DEFF_h$ enables 16K mode. Reading at $DE00_h$–$DEFF_h$ reads from ROM. If the address line A_1 during read is low the cart switches to 8K mode and if A_1 is high the ROM is switched off.

Reading or writing at $DF00_h$–$DFFF_h$ access the RAM ($DF80_h$-$DFFF_h$ is a mirror of $DF00_h$-$DF7F_h$). If the address line A_7 is high during any access the cartridge is switched to Ultimax mode and that also resets the freezer logic.

The original hardware has some other unexplained logic functions, but those don't seem to be required for the software in its ROM. Pressing the freeze button switches the cartridge to Ultimax mode.

CRT files containing KCS Power Cartridge ROMs should have 2 (02_h) as CRT ID.

8.8 Final Cartridge 3

Chameleon can emulate the Final Cartridge 3 hardware. The cartridge provides 64 KByte of ROM containing disk and tape speeders, basic extensions, machine monitor and a freezer. A unique feature of the cartridge is its graphical menu system that can be controlled with a mouse. The ROM is divided into four banks of 16 KByte each. A control register at $DFFF_h$ allows selection of the required bank. The register can only be written as any reads in the $DE00_h$–$DFFF_h$ address range always access the ROMs. As the cartridge occupies all addresses in the IO space at $DE00_h$–$DFFF_h$ no other emulations can be active at the same time.

By setting bit 7 of the control register at $DFFF_h$ disables the cartridge and makes any hidden registers available again. This makes it possible for example to use the clock-port or REU. Use the freeze button to re-activate the Final Cartridge 3 emulation.

On a system reset the control register at $DFFF_h$ is cleared to zero. This maps the first 16 KByte of the ROM into 8000_h–$BFFF_h$ and makes the control register writable. The "CBM80" signature at the beginning of the ROM will make the Kernal jump into the cartridge on reset to let it initialize

and active the graphic desktop environment. The control register also has two individual bits for GAME and EXROM so it can enable either 8 Kbyte or 16 KByte of the current ROM bank.

GAME bit 5	EXROM bit 4	ROM Mapping
0	0	16 KByte at 8000_h–$BFFF_h$
0	1	Ultimax mode, ROM at 8000_h–$9FFF_h$ and $E000_h$–$FFFF_h$.
1	0	8 KByte at 8000_h–$9FFF_h$
1	1	Cartridge ROM disabled.

Pressing the freeze button on a Final Cartridge 3, pulls GAME low (activating Ultimax mode) and pulls NMI low as well to force an interrupt. In Chameleon freezing is handled by the generic freezing logic that can for example also freeze while NMI is already low. The behavior of the control register during a freeze is therefore slightly different from the original hardware.

CRT files containing Final Cartridge 3 ROMs should have 3 (03_h) as CRT ID.

8.8.1 Final Cartridge 3 registers

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$ –$DFFF_h$	56832 –57343		Reads will read cartridge ROM at $1E00_h$–$1FFF_h$, $5E00_h$–$5FFF_h$, $9E00_h$–$9FFF_h$ or $DE00_h$–$DFFF_h$ depending on the current selected bank.
$DFFF_h$	57343	FC3BNK	On write

bit	settings	description
7	register enable	0 = Banking register writable at DFFF. 1 = Banking register invisible. On Chameleon setting this bit to 1 also disables the ROM mirror at $DE00_h$–$DFFF_h$.
6	NMI	0 = Force NMI line low 1 = Normal operation
5	GAME	State of the GAME line
4	EXROM	State of the EXROM line
3	unused	
2	unused	
1	A15	ROM address line 15
0	A14	ROM address line 14

8.9 Simons Basic

The Simons Basic cartridge is a Basic extension. It adds 16 KByte of ROM to the machine. As the cartridge is an extension it needs access to the original Basic V2 ROM. The cartridge has logic to disable the upper 8K of its ROM that overlays the Basic ROM at $A000_h$–$BFFF_h$, while keeping the lower 8K active at 8000_h–$9FFF_h$.

A write action to $DE00_h$ switches to 16K mode, while a read at $DE00_h$ switches to 8K mode. After reset the cartridge emulation is in 16 KByte mode.

Chameleon Offset	C64 location
0000_h–$1FFF_h$	8000_h–$9FFF_h$
2000_h–$3FFF_h$	$A000_h$–$BFFF_h$

CRT files containing the Simons Basic ROMs should have 4 (04_h) as CRT ID.

8.10 Ocean type 1

The Ocean type 1 cartridge can have upto 64 banks of either 8 KByte or 16 KByte. The 64 banks version only uses 8 KByte so the maximum size offered by real cartridges is 512 KByte.

There is one register at $DE00_h$. The lowest 6 bits selects one of the 64 banks (0-63). These 6 bits generate the address lines 19 to 14 of the ROM image in memory. Address bit 13 (that switches

between the two 8 KBytes low and high) comes from ROML and ROMH. The first 8 Kbyte is mapped at 8000_h-$9FFF_h$ and the second 8 Kbyte is mapped at $A000_h$-$BFFF_h$.

The 8 Kbyte cartridges have a mirror of 8000_h at $A000_h$-$BFFF_h$. The Chameleon always asumes 16 Kbyte and any mirrors of the banks are the responsibility of the software loading the ROM images into memory. So when loading 8 KByte images the loader should copy the 8 KByte of each bank twice to get the mirroring correct.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$	56832	BANK	Banking register

bit	settings	description
7–6	–	–
5	bank bit 5	ROM address A_{19}
4	bank bit 4	ROM address A_{18}
3	bank bit 3	ROM address A_{17}
2	bank bit 2	ROM address A_{16}
1	bank bit 1	ROM address A_{15}
0	bank bit 0	ROM address A_{14}

CRT files containing Ocean type 1 ROMs should have 5 (05_h) as CRT ID.

8.11 The Expert Cartridge

The Expert Cartridge is a RAM based freezer cartridge. Being RAM based, the software needs to be loaded from disk before the cartridge can be used. The advantage is that the software on disk can be upgraded to add new functions or fix bugs.

The cartridge has a 3 position switch that is emulated by the left button on the Chameleon. The button configuration must be set to "cartridge mode" when using the expert emulation. A short press toggles the expert emulation between ON and OFF (green LED is lit when the cartridge is on). A long press puts the expert in programming mode PRG (green LED is flashing when the cartridge is in programming mode). In programming mode the 8 KByte RAM is visible in 8000_h–$9FFF_h$ and can be read and written.

The cartridge doesn't have any registers or banking logic. When the expert switch is in the ON position, it will activate after reset and also on an NMI (e.g pressing the RESTORE key). If active it forces ultimax mode putting the 8 KByte RAM at both 8000_h–$9FFF_h$ and $E000_h$–$FFFF_h$. It can turn itself off again by reading or writing at any location in the range $DE00_h$–$DEFF_h$. Because it doesn't have any registers or trampoline area it generates NMIs during operation to bank its own RAM in and out of C64 memory. The freezing function doesn't really work well when the frozen program uses NMIs as well.

Some versions of the Expert Cartridge have an extra button to force an NMI even when the line is held low by the CIA chip. The Chameleon freeze button offers the same kind of function (but without the need to force high on the physical NMI line).

CRT files containing The Expert Cartridge RAM snapshots should have 6 (06_h) as CRT ID.

8.12 Fun Play

Cartridge can have upto 256 KByte (16 banks of 16 KByte). The ROM is mapped at 8000_h–$BFFF_h$. Writing special value of 86_h to the banking register turns the ROM off.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$	56832	BANK	Banking register

bit	settings	description
7–6	–	–
5	bank bit 2	ROM address A_{16}
4	bank bit 1	ROM address A_{15}
3	bank bit 0	ROM address A_{14}
2–1	–	–
0	bank bit 3	ROM address A_{17}

CRT files containing Fun Play ROMs should have 7 (07_h) as CRT ID.

8.13 Super Games

The Super Games cartridge has 4 banks of 16 KByte. Banking is done by writing to a register at DF00. Bit 0 and 1 select a 16K bank. Bit 2 controls GAME and EXROM. Cartridge is enabled when bit 2 is cleared to 0 and disabled when set to 1.

Address (Hex)	Address (Dec)	Name	Description
$DF00_h$	57088	BANK	Banking register

bit	settings	description
7–3	–	–
2	Exrom, Game	0_b = 16K mode 8000-BFFF 1_b = Cartridge off
1–0	Select bank	00_b = Bank 0 01_b = Bank 1 10_b = Bank 2 11_b = Bank 3

CRT files containing Super Games ROMs should have 8 (08_h) as CRT ID.

8.14 Epyx Fastload

Catridge with disk turbo. The cartridge has 8 KByte ROM that is mapped to 8000_h–$9FFF_h$. The ROM contents at $9F00_h$–$9FFF_h$ is mirrored at $DF00_h$–$DFFF_h$. Read accesses to address range $DE00_h$–$DEFF_h$ activate the ROM. Also read accesses to 8000_h–$9FFF_h$ keep the ROM active. After a certain time period the the ROM is turned off again (based on an analog circuit on the cartridge). The emulation in Chameleon counts 512 cpu cycles without any access until the ROM is switched off.

CRT files containing Epyx Fastload ROMs should have 10 ($0A_h$) as CRT ID.

8.15 Westermann

Utility cartridge with 16 KByte ROM. After reset cartridge is in 16 KByte mode mapping the complete ROM to 8000_h–$BFFF_h$. Any read access in the range $DF00_h$–$DFFF_h$ switches the cartridge into 8 KByte mode. In 8 Kbyte mode the first 8 KByte is mapped to 8000_h–$9FFF_h$.

CRT files containing Westermann ROMs should have 11 ($0B_h$) as CRT ID.

8.16 Game System (GS), System 3

The cartridge offsers multiple banks of 8 KByte each. The ROM can be accessed at 8000_h–$9FFF_h$. A bank is selected by writing to address $DE00_h$+bank number. For example to activate bank 3, a write to address $DE03_h$ is required. Reading from anywhere in IO space at $DExx_h$ resets to bank 0. The bank that is selected after reset is bank 0.

CRT files containing Game System ROMs should have 15 ($0F_h$) as CRT ID.

8.17 WarpSpeed

The cartridge maps 16 KByte of ROM at 8000_h–$BFFF_h$ The ROM contents at $9E00_h$–$9FFF_h$ is mirrored at $DE00_h$–$DFFF_h$. Writing to any address in the range $DE00_h$–$DEFF_h$ enables the ROM at 8000_h. Writing to any address in the range $DF00_h$–$DFFF_h$ turns the ROM at 8000_h off.

CRT files containing WarpSpeed ROMs should have 16 (10_h) as CRT ID.

8.18 Dinamic

Game cartridge with 128 KByte of ROM. The ROM contains 16 banks (numbered 0–15) of 8 KByte each mapped at 8000_h–$9FFF_h$. Bank selection is done by reading from $DE00_h$+bank number. For example to activate bank 3, a read from address $DE03_h$ is required. The bank that is selected after reset is bank 0.

CRT files containing Dinamic ROMs should have 17 (11_h) as CRT ID.

8.19 Zaxxon and Super Zaxxon

The Zaxxon and Super Zaxxon cartridges consists of one 4K ROM at 8000_h–$9FFF_h$ (two mirrors) and two 8K ROM banks at $A000_h$–$BFFF_h$. Reading at 8000_h–$8FFF_h$ activates bank 0 and reading at 9000_h–$9FFF_h$ activates bank 1.

The Chameleon hardware can only emulate 8 Kbyte ROMs. So when the Zaxxon ROM is loaded the first 4K needs to be duplicated in memory for the mirror at 9000_h–$9FFF_h$.

Chameleon Offset	Bank	C64 location
0000_h–$0FFF_h$		8000_h–$8FFF_h$
1000_h–$1FFF_h$		9000_h–$9FFF_h$ (should be copy of 8000_h–$8FFF_h$)
2000_h–$3FFF_h$	0	$A000_h$–$BFFF_h$
4000_h–$5FFF_h$	1	$A000_h$–$BFFF_h$

CRT files containing (Super) Zaxxon ROMs should have 18 (12_h) as CRT ID.

8.20 Magic Desk

Cartridge that can offer upto 64 banks of 8 KByte mapped at 8000_h–$9FFF_h$.

Address (Hex)	Address (Dec)	Name	Description
$DF00_h$	57088	BANK	Banking register

bit	settings	description
7	Game line	0 = cartridge enabled 1 = cartridge off
6	–	–
5–0	Select bank	Select one of a possible 64 banks.

CRT files containing Magic Desk ROMs should have 19 (13_h) as CRT ID.

8.21 Super Snapshot 5

The Super Snapshot 5 is an ultility cartridge with freezer. It has 64 KByte of ROM and 32 KByte of RAM. The RAM is mapped at 8000_h–$9FFF_h$ when the Ultimax mode is active. There are 4 banks of RAM each 8 KByte in size. Two banking bit select which of the 4 banks of the RAM are visible. The ROM can map to 8000_h–$9FFF_h$, $A000_h$–$BFFF_h$ and $E000_h$–$FFFF_h$ depending on the mode. There ROM consists of 4 banks of 16 KByte each.

When the freeze button is pressed the cartrigde switches to Ultimax mode. Any access to $DF00_h$–$DFFF_h$ resets freezer logic. Reads in this I/O area always read from ROM.

Writes to $DE00_h$ or $DE01_h$ select bank and mode of the cartridge.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$	56832	BANK	Banking and mode register

bit	settings	description
7–5	–	–
4	bank bit 1	Selects ROM/RAM bank
3	–	not emulated in Chameleon
2	bank bit 0	Selects ROM/RAM bank
1–0	mode	00 = Ultimax (RAM at 8000_h–$9FFF_h$, ROM at $E000_h$-$FFFF_h$) 01 = 16K ROM mode (ROM visible from 8000_h–$BFFF_h$) 10 = 8K ROM mode (Lower 8K of ROM visible from 8000_h–$9FFF_h$) 11 = Cartridge ROM is off (still accesible at $DFxx_h$)

CRT files containing Super Snapshot 5 ROMs should have 20 (14_h) as CRT ID.

8.22 Comal-80

The Comal-80 cartridge has 4 banks of 16 Kb each that map at 8000_h–$BFFF_h$. The required bank can be selected by writing to any location in the range $DE00_h$–$DEFF_h$.

Value Written	Bank Selected	Chameleon Offset
80_h	0	0000_h–$3FFF_h$
81_h	1	4000_h–$7FFF_h$
82_h	2	8000_h–$BFFF_h$
83_h	3	$C000_h$–$FFFF_h$

CRT files containing Comal-80 ROMs should have 21 (15_h) as CRT ID.

8.23 Ross

Ross cartridge has 16 KByte or 32 KByte ROM mapped at 8000_h–$BFFF_h$. Reading at $DE00_h$–$DEFF_h$ enables the second 16 Kbyte bank. Reading at $DF00_h$–$DFFF_h$ disables the cartridge.

CRT files containing Ross ROMs should have 23 (17_h) as CRT ID.

8.24 Mikro Assembler

This is 8 Kbyte ROM cartridge containing an assembler and machine monitor. The ROM is mapped at 8000_h–$9FFF_h$. The last 512 bytes ($9E00_h$–$9FFF_h$) of the ROM are also mirrored in the I/O space at $DE00_h$–$DFFF_h$.

CRT files containing Mikro Assembler ROMs should have 28 ($1C_h$) as CRT ID.

8.25 StarDos

This is a 16 KByte kernal replacement. It replaces the kernal at $E000_h$–$FFFF_h$ to provide extended disk functions. The lower 8 KByte of the ROM contains utilities and can be mapped at 8000_h–$9FFF_h$ when required.

The cartridge uses an unusual way to switch the lower ROM on and off. Repeated read accesses to the first I/O space ($DE00_h$–$DEFF_h$) enable the utility ROM. Repeated read accesses to the second I/O space ($DF00_h$–$DFFF_h$) disable the utility ROM. An analog RC circuit on the PCB performs lowpass filtering on the control signal, requiring multiple reads before the ROM is switched. The code performs 256 accesses to the I/O space in a row (as a single access will not switch the ROM on or off).

The Chameleon emulation is implemented with counters that increase with 16 on each I/O read and decrease with 1 on all other CPU cycles. A counter must reach 512 for the switch to be made, closely emulating the real hardware behaviour.

Chameleon Offset	C64 location
0000_h–$1FFF_h$	8000_h–$9FFF_h$
2000_h–$3FFF_h$	$E000_h$–$EFFF_h$

CRT files containing StarDos ROMs should have 31 ($1F_h$) as CRT ID.

8.26 EasyFlash

FlashROM based cartridge with two 512 KByte chips. The bankswitching on the cartridge is similar to ocean (64 banks of 16K), but it has an additional register to configure the actual type of cartridge to emulate. The two registers are write only. EasyFlash can emulate standard 8 KByte, 16 KByte and Ultimax cartridges and some of the ocean type 1 cartridges. The EasyFlash cartridge starts in ultimax mode. It maps 256 bytes of RAM at $DF00_h$–$DFFF_h$, this is always active even if the ROM is switched off.

The LED and jumper on the cartridge are not emulated in the Chameleon.

The following table shows how the ROM contents is layout in Chameleon memory.

Chameleon Offset	EasyFlash bank	C64 location
0000_h–$1FFF_h$	0	8000_h–$9FFF_h$
2000_h–$3FFF_h$	0	$A000_h$–$BFFF_h$ or $E000_h$–$FFFF_h$
4000_h–$5FFF_h$	1	8000_h–$9FFF_h$
6000_h–$7FFF_h$	1	$A000_h$–$BFFF_h$ or $E000_h$–$FFFF_h$
...		
$FC000_h$–$FDFFF_h$	63	8000_h–$9FFF_h$
$FE000_h$–$FFFFF_h$	63	$A000_h$–$BFFF_h$ or $E000_h$–$FFFF_h$

CRT files containing EasyFlash ROMs should have 32 (20_h) as CRT ID.

8.26.1 EasyFlash registers

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$	56832	BANK	Banking register

bit	settings	description
7–6	–	–
5–0	Select bank	Select one of the 64 banks.

Address (Hex)	Address (Dec)	Name	Description
$DE02_h$	56834	CTRL	Control register

bit	settings	description
7–3	–	–
2	Jumper	Not emulated. Set this bit to 1
1–0	Mode	00 = ROM off 01 = Ultimax (startup) 10 = 8 KByte ROM 11 = 16 KByte ROM

8.27 Capture

A freezer cartridge with 8 Kbyte ROM and 8 KByte RAM. Uses a diskdrive to store the frozen program. It writes the program in separete files to disk each representing 2 Kbyte of memory. Together with some special files containing I/O and CPU register contents. As a freezer the cartridge is special as it doesn't have any software visible registers, so is (almost) impossible to

detect by software for protection against freezing. The cartridge is famous for the ability to make ROMable versions of frozen programs.

The cartridge is disabled and invisible from software after reset. When the freeze button is pressed the cartridge switches (like all freezers) to ultimax mode to freeze the program. The 8 Kbyte ROM is mapped at $E000_h$–$FFFF_h$ and the RAM can be read and written at 6000_h–$7FFF_h$. The cartridge uses two memory locations to enable and disable itself. These become active after freeze and are only disabled again with a reset. Any read or write access to $FFF7_h$ disables the cartridge. Any read or write to $FFF8_h$ enables the cartridge (in ultimax mode).

CRT files containing Capture ROMs should have 34 (22_h) as CRT ID.

8.28 Prophet 64

Cartridge with various music, sound and sequencing tools. It has 256 Kbyte of ROM in 32 blocks of 8 Kbyte. The ROM is mapped at 8000_h–$9FFF_h$. Bank selection is done with a write to any address in the range $DF00_h$–$DFFF_h$.

Address (Hex)	Address (Dec)	Name	Description
$DF00_h$	57088	BANK	Banking register

bit	settings	description
7–6	–	–
5	Game line	0 = cartridge enabled 1 = cartridge off
4–0	Select bank	Select one of the 32 banks.

CRT files containing Prophet 64 ROMs should have 43 ($2B_h$) as CRT ID.

8.29 Mach 5

The Mach 5 cartridge offers a disk turbo (5 times speedup). It consists of an 8Kbyte ROM mapped at 8000_h–$9FFF_h$. The last 512 bytes ($9E00_h$–$9FFF_h$) of the ROM are also mirrored in the I/O space at $DE00_h$–$DFFF_h$.

Writing to any address in the range $DE00_h$–$DEFF_h$ enables the ROM at 8000_h. Writing to any address in the range $DF00_h$–$DFFF_h$ turns the ROM at 8000_h off. The mirrors in I/O space stay accessable even if the ROM at 8000_h is turned off.

CRT files containing Mach 5 ROMs should have 51 (33_h) as CRT ID.

8.30 PageFox

A cartridge offering a full desktop publishing system for the C64. It has 64K ROM (in two 32K banks) and 32K RAM. Using the RAM however is a bit tricky. The ROMs and RAM are mapped to 8000_h–$BFFF_h$. Having RAM in this location means writing to the cartridge RAM also writes to the C64 memory. Also disabling the cartridge by setting bit 4 in the control register doesn't block writing to the RAM.

The banking register is write only and mapped from $DE80_h$–$DEFF_h$. The other I/O spaces are free for other cartridges to use (like a REU). The banking register is set to 00_h on a reset (enabling the first ROM that contains the CBM80 signature). To completely disable the cartridge, write FF_h to the banking register.

Address (Hex)	Address (Dec)	Name	Description
$DE80_h$–$DEFF_h$	56960 –57087	BANK	Banking register

bit	settings	description
7–5	–	–
4	Disable	0 = Cartridge enabled 1 = Cartridge disabled
3–2	CS	00 = Eprom "79" 01 = Eprom "ZS3" 10 = 32K RAM 11 = None selected (open databus)
1	A14	Select upper or lower 16K from the 32K ROMs or RAM.
0	–	–

CRT files containing Business Basic ROMs should have 53 (35_h) as CRT ID.

8.31 Business Basic

CRT files containing Business Basic ROMs should have 54 (36_h) as CRT ID.

8.32 Cartridge Configuration Registers

Address (Hex)	Address (Dec)	Name	Description
$D0F0_h$	53488	CFGCRT	Cartridge emulation

bit	settings	description
7–0	Cartridge Type	00000000_b, 00_h = Off 00000001_b, 01_h = RetroReplay 00000010_b, 02_h = KCS Power Cartridge 00000011_b, 03_h = Final Cartridge 3 00000100_b, 04_h = Simons Basic 00000101_b, 05_h = Ocean type 1 00000110_b, 06_h = Expert Cartridge 00000111_b, 07_h = Fun Play 00001000_b, 08_h = Super Games 00001010_b, $0A_h$ = Epyx Fastload 00001011_b, $0B_h$ = Westermann 00001111_b, $0F_h$ = Game System (GS), System 3 00010000_b, 10_h = WarpSpeed 00010001_b, 11_h = Dinamic 00010010_b, 12_h = (Super) Zaxxon 00010011_b, 13_h = Magic Desk 00010100_b, 14_h = Super Snapshot 5 00010101_b, 15_h = Comal-80 00010111_b, 17_h = Ross 00011100_b, $1C_h$ = Mikro Assembler 00011111_b, $1F_h$ = StarDos 00100000_b, 20_h = EasyFlash 00100010_b, 22_h = Capture 00101011_b, $2B_h$ = Prophet 64 00110011_b, 33_h = Mach 5 00110101_b, 35_h = PageFox 00110110_b, 36_h = Business Basic 11111100_b, FC_h = 16K ROM cartridge at 8000_h–$BFFF_h$ 11111101_b, FD_h = 16K ROM cartridge in Ultimax mode 11111110_b, FE_h = 8K ROM cartridge at 8000_h–$9FFF_h$ others = reserved for future use

$D0F1_h$	53489	CFGSPI	Clock-port and MMC64 Emulation

bit	settings	description
7	Reserved, must be 0	
6	Ext NMI	0 = Clock-port NMI is disabled in cartridge and docking-station mode. 1 = Clock-port NMI is always enabled (might require an extra pull-up on the NMI line in some cases)
5–4	Clock port	00_b = Off 01_b = Clock port at $DE00_h$-$DE0F_h$ 10_b = Clock port at $DF20_h$-$DF2F_h$ 11_b = reserved
3	ROM source	0 = ROMs are banked with MMU at $D0A0_h$-$D0AF_h$ 1 = C64 original Basic and Kernal ROMs are used This bit is only functional in cartridge mode. In standalone mode and on the C-One this bit should always be clear. Note that the character ROM is always emulated and never the C64 original.
2	MMC64 active	0 = MMC64 active (Copy of bit 7 in $DF11_h$) 1 = MMC64 disabled ($DF1x_h$ registers are invisible) This bit can only be toggled in $DF11_h$ after unlocking, while it can be accessed here at any time. On reset this bit is set to 1 if MMC64 emulation is disabled (bits 1–0 are zero) and 0 when emulation is enabled.
1–0	MMC64 Emulation, SPI	00_b = Off 01_b = MMC64 Emulation 10_b = reserved 11_b = MMC64 Emulation with extra bits combinations defined for access to RTC (Real Time Clock) and FlashRom.

$D0F5_h$	53493	CFGREU	REU (Ram Expansion Unit) and geoRAM Emulation Config

bit	settings	description
7	Enable REU	0 = REU is disabled (off) 1 = Enable REU emulation and activate registers at $DF00_h$–$DF0A_h$
6	Enable geoRAM	0 = geoRAM is disabled (off) 1 = Enable geoRAM emulation and activate registers at $DE00_h$–$DEFF_h$, $DFFE_h$ and $DFFF_h$
5–3	geoRAM size	000_b = 64 KByte 001_b = 128 KByte 010_b = 256 KByte 011_b = 512 KByte 100_b = 1 MByte 101_b = 2 MByte 110_b = 4 MByte 111_b = reserved for future use
2–0	REU memory size	000_b = 128 KByte 001_b = 256 KByte 010_b = 512 KByte 011_b = 1 MByte 100_b = 2 MByte 101_b = 4 MByte 110_b = 8 MByte 111_b = 16 MByte (Note there is not enough RAM on C-One for this setting)

8.33 Cartridge stacks and combinations

Unless a port expander is used only a single cartridge can be used in the Commodore 64 at any time. In Chameleon there are a variety of functions integrated into a single device. This makes it a lot more likely that multiple functions are selected and active at the same time. However the original cartridges on which the Chameleon functions are based on, were never designed to be used at the same time. So not all possible combinations make sense. There are some overlaps in memory areas and registers that each cartridge uses. So some functions will hide the registers and ROM images used by other functions when enabled.

The cartridge emulator engine in Chameleon assigns highest priority to the MMC64 registers and boot ROM. The freezer (or game) emulation has next priority followed by the clockport, any ram expander registers and simple ROMs. The internal ROMs (BASIC and Kernal) and system RAM have the lowest priority.

9 Menu mode

Menu mode is similar in function to a freezer cartridge, but is separate from the normal cartridge emulation logic. The mode is designed for configuration and control of the various aspects of the cartridge. Because it functions as a freezer it is possible to enter menu mode at any time and in most cases return to the original application again when done.

9.1 Entering menu mode

Menu mode can be entered in the following ways:

- Pressing the freeze button longer as 0.7 seconds
- On reset if bit 2 of 53500 ($D0FC_h$ is set)
- Writing 32 (20_h) or 33 (21_h) into 53502 ($D0FE_h$) while in configuration mode
- Menu mode is also active after power-up

9.2 Programming for menu mode

In menu mode the I/O space is always active (the CPU banking registers at address 0 and 1 have no effect in menu mode). Some Chameleon specific settings in the configuration registers at $D0F0_h$–$D0FF_h$ are also over-ridden. Any changes made to those configuration registers therefore will only take effect when leaving the menu mode. In menu mode the REU and MMC64 emulations are always active and any freezers or game emulations are (temporarily) disabled.

In menu mode a total of 56 KByte of ROM and RAM memory is replaced. This allows utility functions to operate without disturbing the frozen program. The MMU can be used for banking additional memory in and out of the address space. The 56 KByte is build up from seven banks of 8 KByte each. The area $C000_h$–$CFFF_h$ keeps it original mapping to MMU bank $0C_h$.

When entering the menu by writing to 53502 ($D0FE_h$) there are two values that can be used. Writing 32 (20_h) only switches the memory layout, while a value of 33 (21_h) also disables NMI interrupts.

9.3 VIC-II memory access in menu mode

In menu mode the VIC-II accesses also go to the seven new 8K banks. However the character ROM accesses at 1000_h-$1FFF_h$ and 9000_h–$9FFF_h$ stay intact and access MMU bank $1D_h$. As there is no new memory at $C000_h$–$DFFF_h$ in menu mode, the VIC-II will gets it data from normal C64 MMU banks $0C_h$ and $0D_h$ on these adresses. Note that the CPU is not able to access any data

(character ROM or RAM) at the memory range $D000_h$-$DFFF_h$ directly as the I/O space is always on top.

9.4 Differences between Menu and Configuration modes

In menu mode only the registers $D0FD_h$ and $D0FF_h$ are readable. All other configuration registers read as 255 (FF_h), unless the configuration mode is active at the same time as well. Also writes to $D0FE_h$ are possible without first enabling configuration mode. This allow the menu system to perform soft-resets and reconfiguration commands. It is therefore possible to detect menu mode by disabling the configuration mode and check if $D0FD_h$ is unequal to 255 (FF_h).

9.5 Extra 256 bytes of ROM or RAM

To facilitate the use of menu mode for other functions normally implemented in (freezer) cartridges an extra I/O area can be enabled. An extra space of 256 bytes can be enabled at $D700_h$–$D7FF_h$, which normally is an unused mirror of the SID registers. This keeps the $DE00_h$–$DFFF_h$ memory area (which often performs similair functions) empty for use by the catridge emulations. Once the $D7xx_h$ area is enabled with bit 5 of 53498 ($D0FA_h$), it stays active even in C64 mode. This allows basic or kernal vector hooks to point into this area, where extra trampoline-code can be placed to jump into menu mode (by writing 20_h into $D0FE_h$) perform the function and then leave menu mode again.

Take note that it is possible to have a stereo SID emulation mapped at $D700_h$ as well. The RAM or ROM has priority over any SID registers, making the stereo SID unavailable at this address. The recommended location for the stereo SID is $D420_h$, which doesn't overlap with any other Chameleon function.

9.5.1 Blocking NMI interrupts

After entering the menu with the freezer or by writing 33 (21_h) into 53502 ($D0FE_h$), the NMI interrupt line is disabled. Write the value of 34 (22_h) into 53502 ($D0FE_h$) to enable NMI interrupts. Writing a value of 35 (23_h) disables NMI interrupts again. The disabling of the NMI interrupt line is done with the menu freezer logic (without triggering an actual freeze). When the NMIs are disabled the menu should perform an unfreeze sequence to reenable the NMIs or write 34 (22_h) into 53502 ($D0FE_h$) before leaving.

Although the NMI blocking can also be enabled from normal applications (in configuration mode), it is not recommended to do so, as it could block the user from using the menu system. To block NMIs in user code use the standard trick by keeping the second CIA unacknowledged keeping the NMI line low. This trick will not block the use of the menu system or menu button (the menu freezer will still work even if the NMI line is permanently low).

9.6 Leaving menu mode

9.6.1 Leaving menu mode with RTI

To leave menu mode perform a read at address 53503 ($D0FF_h$). The configuration disable register at 53503 ($D0FF_h$) always contains the RTI opcode (64 or 40_h) when the menu (or configuration) mode is active. A read on that location turns off menu mode. One way to leave menu mode is to jump to $D0FF_h$ and while the RTI opcode is fetched the memory configuration is restored. So the machine is in a same state before the menu mode was activated. The RTI instruction will fetch the program-counter and CPU state from the original stack and continues execution. However when menu mode is entered with a reset or under software control, the menu software is responsible to initialize the stack in such a way that the RTI can be used to leave the menu. Alternatively the menu application can run in a memory area that is uneffected by the switch, which is the range $C000_h$–$DFFF_h$ by default and perform a load operation at $D0FF_h$.

Take note that the NMI interrupt processing is re-enabled again if it was previously disabled (by entering through the freezer or by writing either 33 (21_h) or 35 (23_h) to 53502 ($D0FE_h$)).

9.6.2 Leaving menu mode with reset

The other way to leave the menu mode is by performing a software reset by writing the value 165 or 166 ($A5_h$or $A6_h$) to 53502 ($D0FE_h$). This also resets the menu freezer and re-enables the NMI interrupt if it was disabled.

9.7 Limitations

It is not recommended to change any cartridge emulation settings while in menu mode unless a (soft) reset is performed on exit. As changing cartridge type while it is in use might result in undefined behavior. This is especially true for freezer cartridges as these might have hooks installed into the basic and system vectors for basic enhancements and turbo loaders. Removing the cartridge without a reset will leave the vectors pointing into the void and crashing the machine.

To force a reset from software write the value 165 or 166 ($A5_h$ or $A6_h$) into the register at 53502 ($D0FE_h$).

10 Timers

To support the menu system there are 4 timers available. The first timer is a high speed timer running at 1 Khz for implementing short delays. The timer will overflow after 256 milliseconds. The second and third timers run at 100 Hz (10 millisecond ticks) and overflow after 2.56 seconds. The last timer runs at only 10 Hz and can be used to time longer periods (overflows after 25.6 seconds).

The timer registers are located at $D0AA_h$ to $D0AD_h$ and can be activated by setting bit 1 in the configuration register $D0FA_h$. The registers can both be read and written. So the timer can be set to a certain start value. The timers can also be used by application programs, although they where added primarily for supporting the menu system (which can't use the CIA timers). As applications can also fully use the four normal CIA timers, the practical use of these additional timers might be limited.

There are no configuration registers. The timers are always running at the specified speed.

10.1 Timers Registers

Address (Hex)	Address (Dec)	Name	Description
$D0AA_h$	53418	TIMER1	Timer 1, counts up each 1 millisecond (1 Khz)
$D0AB_h$	53419	TIMER2	Timer 2, counts up each 10 milliseconds (100 Hz)
$D0AC_h$	53420	TIMER3	Timer 3, counts up each 10 milliseconds (100 Hz)
$D0AD_h$	53421	TIMER4	Timer 4, counts up each 100 milliseconds (10 Hz)

11 CPU Turbo/Accelerator

The 6510 CPU emulation inside the Chameleon can run faster as the normal 1 Mhz. Only CPU cycles where memory is accessed can be speed-up. When any registers are accessed the CPU needs to slow down and resynchronize to the C64 clock. This is true for all regster accesses in the $D000_h$-$DFFF_h$ area. This includes color ram, CIA, VIC-II, SID and also the Chameleon specific registers.

The turbo can run at either limited speed (configurable between 2x and 6x) or maximum speed. At 2x speed there are two accesses in a single system cycle, when VIC-II accesses are turned off the timing is almost identical to the C128 in 2 Mhz mode. When the turbo is set to maximum the CPU uses all remaining free SDRAM cycles. The exact speed of the CPU will depend on the type of code executed and which RAM locations are accessed (which influences the cache hit rate) and how many other devices and controllers are activated inside Chameleon. Turning off unused blocks inside Chameleon will allow the CPU to run at faster speed. Especially the amount, depth and size of graphic layers active on the VGA display can have a great effect on the available memory bandwidth.

11.1 Turbo I/O

Most I/O operations need to happen at orignal speed (1 Mhz). I/O accesses are all the reads and writes the CPU performs in address range $D000_h$ to $DFFF_h$. The CPU needs to wait until the beginning of a machine cycle before it can start the I/O operation. This waiting slows the turbo down. The "Turbo I/O" feature peforms certain I/O accesses at maximum speed. The following table shows which accesses benefit from "Turbo I/O"

Addresses	Access	I/O device
$D0A0_h$–$D0AF_h$	R/W	MMU / Timers
$D800_h$–$DCFF_h$	Read	Color RAM

11.2 Auto Speed

The turbo has the option (default on) to automatically slow down after any CIA chip accesses that control the IEC bus. If any register is accessed that could read or control the IEC lines, the turbo will automatically switch to normal speed for about 10 milliseconds (10000 CPU cycles). This gives the software enough time to run timing loops that depend on the correct CPU speed. After 10 milliseconds without any IEC accesses the turbo switches on again.

The auto speed option can be disabled by setting bit 4 of the configuration register at 53491 ($D0F3_h$).

11.3 VIC-II register

If bit 5 of the turbo config register ($D0F3_h$) is set then bit 0 of the VIC-II register at 53296 ($D030_h$) controls the turbo. This makes the turbo switchable by software that was written to use the 2 Mhz mode of the C128 to speedup the program. When bit 0 of 53296 ($D030_h$) is set the turbo is on. And when the bit is cleared the turbo is off. It might be necessary to set a speed limit as well as the program might require CPU speed of 2 Mhz and not something much faster.

Take note that on a C128 machine the VIC-II screen shows garbage in 2 Mhz, so the program would most likely only enable the turbo in the borders. Another posibility is that the program uses the VDC chip (80 columns mode). On a C128 this chip is accessable even in C64 mode. This chip is however not supported nor emulated by the Chameleon cartridge.

11.4 Turbo Configuration Register

Address (Hex)	Address (Dec)	Name	Description
$D0F3_h$	53491	CFGTUR	Turbo configuration

bit	settings	description
7	Turbo Enable	0 = 1 Mhz mode 1 = Turbo mode active
6	Reserved, must be 0	
5	VIC-II turbo bit	0 = Off 1 = "Turbo Enable" is mirrored at bit 0 of $D030_h$
4	Auto Speed	0 = Turbo is slowed down on IEC bus accesses. 1 = Auto speed is disabled. It is recommended to keep the Auto Speed setting at 0. Otherwise accessing drives and other peripherals on the serial IEC bus might be impossible with the turbo active.
3–0	Turbo speed limit	0000_b = CPU not limited, runs at maximum speed possible. 0001_b = CPU limited to 2x normal speed 0010_b = CPU limited to 3x normal speed 0011_b = CPU limited to 4x normal speed 0100_b = CPU limited to 5x normal speed 0101_b = CPU limited to 6x normal speed 1100_b = CPU limited to 100% speed (slow-down mode) 1101_b = CPU limited to 75% speed (slow-down mode) 1110_b = CPU limited to 50% speed (slow-down mode) 1111_b = CPU limited to 25% speed (slow-down mode) others = reserved for future use Setting a speed limit determines the maximum speed the CPU is allowed to run, it can actually run slower if other factors slow it down (like I/O accesses or many cache misses). The limit is caculated using the average speed over the last 250 CPU cycles. The last 4 modes are not turbo modes, but slowdown modes. They can slowdown the CPU in 25% steps.

12 Disk Drive Emulation

The Chameleon can emulate upto two 1541 disk-drives. These are known as drive 8 and drive 9, although the ID can be changed when there is also an external drive connected. Drive 8 emulates a standard 1541 drive with optional 8 Kbyte RAM expansion. The ROM size can be upto 32 Kbyte.

Drive 9 can also emulate a standard 1541 drive with optional 8 KByte RAM expansion, but can also switched into an enhanced mode. In this mode it has additional registers to access the MMU, MMC card and it can also control parts of the drive 8 emulation. This advanced drive is able to mount a D64 without going through the menu. This can have advantages when using the Chameleon as standalone drive emulator. Ofcourse other functions could be assigned that use MMC and IEC bus (printer emulation). This however will require additional software effort.

The current beta firmware doesn't support the enhanced mode of drive 9!

12.1 Drive Memory Map

12.2 Disk track layout

Each disk image in memory uses 336 KByte of memory. Each track is allocated 8 Kbyte and there can be upto 42 disk tracks. Although not all of the 8 Kbyte is used, it is easier to manage if each track starts on a power of 2 boundary.

The actual track lengths are as follows:

Track	Number of bytes
1–17	7696
18–24	7144
25–30	6672
31–42	6256

12.3 Drive Configuration Registers

Address (Hex)	Address (Dec)	Name	Description
$D0F6_h$	53494	CFGDWR	Floppy-disk image write bits

bit	settings	description
7		1 = Writes have been done by drive 9 to floppy image 4.
6		1 = Writes have been done by drive 9 to floppy image 3.
5		1 = Writes have been done by drive 9 to floppy image 2.
4		1 = Writes have been done by drive 9 to floppy image 1.
3		1 = Writes have been done by drive 8 to floppy image 4.
2		1 = Writes have been done by drive 8 to floppy image 3.
1		1 = Writes have been done by drive 8 to floppy image 2.
0		1 = Writes have been done by drive 8 to floppy image 1.
		Bits are set when the emulated drive writes to the floppy image. Menu software can use these bits to write updated image back to sd card. Bits can be reset by writing 0 to the register.

$D0F7_h$	53495	CFGDSK	Disk images

bit	settings	description
7-6	Disk 9 floppy range	Number of floppy images for drive 9 00_b = 1 image 01_b = 2 images 10_b = 3 images 11_b = 4 images
5-4	Disk 9 floppy select	Select floppy image for drive 9 00_b = floppy image 1 selected 01_b = floppy image 2 selected 10_b = floppy image 3 selected 11_b = floppy image 4 selected
3-2	Disk 8 floppy range	Number of floppy images for drive 8 00_b = 1 image 01_b = 2 images 10_b = 3 images 11_b = 4 images
1-0	Disk 8 floppy select	Select floppy image for drive 8 00_b = floppy image 1 selected 01_b = floppy image 2 selected 10_b = floppy image 3 selected 11_b = floppy image 4 selected

$D0F8_h$	53496	CFGFD0	Drive emulation

bit	settings	description
7–6	Enable virtual-drive CPU	00_b = drive cpu stopped 01_b = drive cpu running
5	Drive door	0_b = Drive door closed 1_b = Drive door open
4–3	Reserved, must be 0	–
2	Drive memory size	0 = 2 Kbyte (default) 1 = 8 Kbyte (not implemented in beta firmware!)
1–0	Drive ID jumpers	00_b = drive device id is 8 01_b = drive device id is 9 10_b = drive device id is 10 11_b = drive device id is 11

$D0F9_h$	53497	CFGFD1	Reserved for second drive

bit	settings	description
7–6	Enable virtual-drive CPU	00_b = drive cpu stopped 01_b = drive cpu running
5	Drive door	0_b = Drive door closed 1_b = Drive door open
4–3	Reserved, must be 0	–
2	Drive memory size	0 = 2 Kbyte (default) 1 = 8 Kbyte (not implemented in beta firmware!)
1–0	Drive ID jumpers	00_b = drive device id is 8 01_b = drive device id is 9 10_b = drive device id is 10 11_b = drive device id is 11

13 SID Emulation

The Chameleon can emulate one or two SID chips. The two SID emulation is there to support stereo sid-tunes. Stereo SID can be enabled in cartridge mode even if the Commodore 64 only has one chip installed. In that case the internal SID chip plays the left channel only (writes to the right channel will not reach the chip). The second SID can be placed at a number of possible

addresses in the memory map. The current supported choices are $D420_h$, $D500_h$, $D700_h$, $DE00_h$ or $DF00_h$. The recommanded value is $D420_h$ as that will not cause any conflicts with the various cartridge emulations.

13.1 Using a Second SID Chip

When the Commodore 64 has a second chip installed it can be activated as well. Now writes for the left channel go to the first SID chip and writes for the right channel will go to the second SID chip inside the machine. Because the machine was designed with only one SID, there is no standard for the address range for a second chip. Each stereo SID modification will be different. Chameleon supports many possible address locations that can be configured to accomodate most existing stereo SID configurations. Note that this address is only used when addressing the chip and that address can be completely different from where the stereo SID is visible in Chameleon address space.

This dual addressing allows an easy modification to the machine by using one of the IOe or IOf lines as chip-select for the second SID, whithout causing any address conflicts for freezer emulation in the Chameleon. Note that some freezer cartridge emulations or memory expansions might make the second SID unaddressable if it is mapped at $DE00_h$ or $DF00_h$. The cartridge emulation always has higher priority. Here the remapping comes at the rescue as Chameleon can map the second chip at a different (not conflicting) address. The recommanded value is $D420_h$ as that will not cause any conflicts with the various cartridge emulations.

13.2 Filter curves

The SID emulation can emulate two different filter cutoff curves. In 6581 mode the filter cutoff curve is exponential. In 8580 mode the filter cutoff curve is linear. Choosing the correct filter-curve to the actual type the SID music was originally composed for, can greatly improve the quality of the playback.

The two emulated SIDs have each a seperate configuration bit to select the required filter curve. So the two SIDs can be configured with a different curve if desired.

13.3 SID Configuration Register

Address (Hex)	Address (Dec)	Name	Description
$D0F4_h$	53492	CFGSID	SID emulation

bit	settings	description
7	Second SID type	0 = Emulate 6581 SID-Chip 1 = Emulate 8580 SID-Chip Selects the type of SID emulated for the second SID in stereo mode.
6	First SID type	0 = Emulate 6581 SID-Chip 1 = Emulate 8580 SID-Chip Selects the type of SID emulated for the first SID at $D400_h$.
5–3	Stereo SID in C64	Specify where the second SID chip is located inside the C64 memory space. 000_b = Single SID in C64 001_b = Second SID in C64 at $D420_h$ 010_b = reserved 011_b = reserved 100_b = Second SID in C64 at $D500_h$ 101_b = Second SID in C64 at $D700_h$ 110_b = Second SID in C64 at $DE00_h$ 111_b = Second SID in C64 at $DF00_h$ For C-One use "000" when zero or one SID-Chip is placed and "001" when both SID-Chips are present. For stereo SID chip to properly work you also need to turn Chameleon SID emulation in stereo mode.
2–0	Stereo SID (emulation)	Specify if and where the second SID must be placed in chameleon memory space. This doesn't have to be the same address as a second SID chip inside the C64 as Chameleon can translate the address automatically. 000_b = Emulate single SID 001_b = Emulate stereo SID use A_5 for selection ($D420_h$, $D460_h$, $D4A0_h$ ...) 010_b = reserved 011_b = reserved 100_b = Emulate stereo SID use A_8 for selection ($D5xx_h$, $D7xx_h$) 101_b = Same as setting 100_b ($D5xx_h$, $D7xx_h$) 110_b = Emulate stereo SID use IO1 for second SID ($DE00_h$) 111_b = Emulate stereo SID use IO2 for second SID ($DF00_h$) others = reserved for future use When only a single SID chip is emulated (setting "000") it is played through both audio output channels in mono. For any of the stereo settings, the first SID-Chip emulation is played through the left audio output channel and is always located in memory at $D400_h$. The second SID-Chip emulation is played through the right audio output channel.

14 VIC-II Emulation

Inside the Chameleon is a replica of the VIC-II chip. This is the chip that generates the video picture. The replica allows the picture to be captured in a framebuffer and then shown on the VGA screen.

In cartridge mode the Chameleon also send the data fetched from its memory to the VIC-II inside the Commodore 64 machine. The machine is put into Ultimax mode and then one of the highest address lines is driven low during VIC-II fetches. This disables all internal Commodore 64 memory and ROMs accesses. Now data from the Chameleon memory can be send to the VIC-II. This trick uses a previously undocumented mode of the Commodore 64. It allows the memory map to be changed with the MMU and still keep an identical picture on both the original video output and the VGA connector. Everything can be moved and relocated in memory except for the color-RAM, as that is a separate SRAM inside the machine and only accessable through reads and writes at $D800_h$–$DBFF_h$.

14.1 Commodore 128 Incompatibility

The undocumented mode used by the Chameleon to feed the VIC-II chip is unfortunately not available on any of the Commodore 128 machines. It makes Chameleon strictly a Commodore 64 only cartridge even though the cartridge port and the signals on it are defined the same for all the different machine types.

The equivalent PLA logic equations required are simply not there in the Commodore 128 logic chips. Even if the machine is put into "GO64" mode by holding EXROM low on the cartridge

port it will fail to accept the external data. The resulting bus-conflicts are dangerous for both the Chameleon cartridge and the logic chips inside the Commodore 128. It is unwise to experiment as unrepairable damage can occur to your equipment.

14.2 Framebuffer

The VIC-II emulation writes the graphics into a framebuffer. The location of the framebuffer in memory is controlled by MMU bank 28 ($1C_h$). The framebuffer has a fixed size of 256 KByte and must be placed in memory on an 8 byte boundary (lowest 3 bits must be 0). It is 512 pixels wide and 1024 lines high, each pixel uses 4 bits to store one out of 16 colors. Not all memory locations are used and which ones depends on the VIC-II type (PAL or NTSC) and if double buffering is enabled in the core. For the current beta cores the double buffer logic is disabled, currently leaving half of the framebuffer unused.

14.3 VIC-II Emulation Registers

Address (Hex)	Address (Dec)	Name	Description
$D0F2_h$	53490	CFGVIC	VIC-II Emulation Config

bit	settings	description
7	VIC-II Read Enable	0 = Off 1 = Perform memory accesses for VIC-II
6	Frame buffer Enable	0 = Off 1 = VIC-II emulation writes graphics to framebuffer (MMU slot $1C_h$)
5	reserved, must be 0	–
4	Force side-border open	0 = Not forced open 1 = Side-border is forced open (turbo mode must be on!)
3	reserved, must be 0	–
2–0	VIC-II type	000 = PAL (63 columns, 312 lines) 001_b = Reserved 010_b = NTSC (65 columns, 263 lines) 011_b = Old-NTSC (64 columns, 262 lines) $1xx_b$ = Reserved These bits are read-only in cartridge mode. They can be changed in standalone mode and on the C-One.

15 Using the Onboard Flash Memory

16 Using the RTC (Real Time Clock) Chip

The realtime clock is accessed by using the MMC64 registers as it sits on the same SPI bus as the MMC/SD-card iterface. The MMC64 emulation must be set to mode 11_b in configureation register $D0F1_h$ to get access to the required additional chip-select signals. The RTC chip has a rather slow SPI bus. Using the fast clock speed mode of the MMC64 interface (8 Mhz) will result in communication errors. So the MMC64 must be run set to 250 Khz clock speed mode when communicating with the RTC chip by clearing bit 2 of register $DF10_h$ to zero.

Refer to the datasheets of the RTC chip (PCF2123 from NXP) for programming information and usage. Please note that re-programming the RTC chip can cause conflicts with way the menu-system uses the chip. The menu expects the RTC to be in 24 hour mode and have a valid date and time programmed. If the menu-system detects an inconsistency, the RTC chip will be reinitialized possibly loosing configured RTC data. Keep this in mind when experimenting with the RTC chip settings.

17 RTC access registers (using MMC64 emulation)

Address (Hex)	Address (Dec)	Name	Description
$DF10_h$	57104	MMCSPI	SPI transfer register. Write in this register sends byte to SPI bus, read is last retrieved byte.
$DF11_h$	57105	MMCCTL	MMC64 Control register.

bit	settings	description
7	MMC64 active	0 = MMC64 is active 1 = MMC64 is disabled Bit can only be modified when unlocked
6	SPI trigger mode	0 = Trigger SPI transfer on write to register $DF10_h$ 1 = Trigger SPI transfer on read of register $DF10_h$
5	External ROM	0 = Allow external ROM when BIOS is disabled 1 = Disable external ROM
4	Flash mode	0 = Normal mode 1 = Flash update mode Not implemented, must be set to 0
3	Clock port address	Not implemented, must be set to 0
2	Clock Speed	0 = 250 KHz SPI clock 1 = 8 Mhz SPI clock
1	MMC cart select	0 = Cart selected 1 = Cart not selected
0	MMC64 Bios	0 = MMC64 BIOS ROM active 1 = BIOS ROM disabled (external ROM active)

$DF12_h$	57106	MMCST	MMC64 Status register (read-only).

bit	settings	description
5	Flash jumper	Not implemented reads always as 0
4	MMC Write Protect	0 = Cart can be written 1 = Cart is write protected
3	MMC Cart Detect	0 = Cart inserted 1 = No cart present, slot empty
2	External EXROM line	
1	External GAME line	
0	Busy	0 = SPI bus ready 1 = SPI bus busy (only for 250 Khz mode)

$DF11_h$	57105	MMCCTL	MMC64 Control register.

bit	settings	description
4,1	Select	00 = MMC cart selected 01 = Nothing selected 10 = Flash ROM selected 11 = RTC (Real Time Clock) selected

18 PS/2 Keyboard connector

A PS/2 compatible keyboard can be connected to the Chameleon by using the purple connector on the break-out cable. The keyboard should be connected before applying power to the Chameleon, PS/2 devices are not hot-pluggable.

In cartridge mode the PS/2 keyboard can be used in parallel with the C64 keyboard, both operate at the same time. Besides the keyboard function it also emulates a joystick on the numeric-keypad. The NUM-LOCK key toggles between emulating a joystick on port 1 or port 2.

18.1 PS/2 Keyboard layout

PS/2 keyboard	C64 function
ALT	C= key
ESCAPE	RUN/STOP
F1	F1
F2	RShift + F1
F3	F3
F4	RShift + F3
F5	F5
F6	RShift + F5
F7	F7
F8	RShift + F8
F9	£
F10	+
PAUSE	RESTORE
~	⇐
_	–
= / +	=
Home	HOME/CLR
Backspace	DEL/INST
[/ {	@
] / }	*
\ / |	⇑

PS/2 keyboard	C64 function
NUM-LOCK	Select port 1 or port 2
Numeric 0	Joystick Fire Button
Numeric 1	Joystick Left + Down
Numeric 2	Joystick Down
Numeric 3	Joystick Right + Down
Numeric 4	Joystick Left
Numeric 6	Joystick Right
Numeric 7	Joystick Left + Up
Numeric 8	Joystick Up
Numeric 9	Joystick Right + Up
F11	Left cartridge button
F12	Middle (Freeze) cartridge button
Print Screen	Right (Reset) cartridge button
Page Up	F1
Page Down	F7

Take note that it is possible with the keyboard to press both "Joystick Left" and "Joystick Right" at the same time (same is true for up and down). There are a few games that crash when you do so. Don't blame Chameleon for the crash, but the programmer that wrote the fragile game code.

19 PS/2 Mouse connector

A PS/2 compatible mouse can be connected to the Chameleon by using the green connector on the break-out cable. The mouse should be connected before applying power to the Chameleon, PS/2 devices are not hot-pluggable. Both two buttons mice and three buttons mice with scroll-wheel (known as intelli-mouse) can be used. The type of mouse connected is automatically detected by the Chameleon hardware.

The Chameleon emulates a commodore 1351 mouse. The optional scroll-wheel is mapped compatible with the Micromys PS/2 mouse adapter. Therefore the mouse emulation is compatible with most exisiting software packages that have mouse support.

Mouse emulation can be used both in cartridge and standalone mode. The mouse normally plugs into joystick port 1 and uses the paddle inputs (and corresponding converters in the SID) for X and Y movement information. Therefore if used in cartridge mode, any paddles or other analog controllers connected to the joystick ports will not function. To prevent conflicts the mouse emulation can be completely switched off by setting bit 5 of configuration register 53500 ($D0FC_h$). Set bit 4 of the configuration register if the software requires the mouse on joystick port 2.

19.1 Emulation Behavior

The range of possible values for the emulated mouse is 82 to 209 when reading the SID registers for the potX and potY. The following table shows how the mouse maps to the joystick port.

Mouse action	Bit in CIA register	Joystick Movement	Comment
Left button	4	Fire	
Right button	0	Up	
Middle button	1	Down	
Scroll up	2	Left	50 ms long pulse (pulses are at least 50 ms apart)
Scroll down	3	Right	50 ms long pulse (pulses are at least 50 ms apart)

20 Infrared remote (CDTV)

Chameleon can be controlled with an Amiga CDTV compatible IR remote. The keys on the remote are mapped to C64 joystick and key presses. See following table for the mapping of the keys.

Infrared CDTV remote key	C64 function
1	F1
2	RShift + F1
3	F3
4	RShift + F3
5	F5
6	RShift + F5
7	F7
8	RShift + F7
9	RUN/STOP
0	Spacebar
ESCAPE	arrow left
ENTER	RETURN
REW	cursor left (RShift + right)
PLAY/PAUSE	cursor up (RShift + down)
FF	cursor right
STOP	cursor down
GENLOCK	Left push button
CD/TV	Middle push button (Freeze/Menu)
POWER	Right push button (Reset/Reboot)
Vol Up	+
Vol Down	-
Switch in MOUSE position	Joystick 1
Switch in JOY position	Joystick 2
A	Fire
B	Auto fire (8 Hz)

21 Complete register map

Address (Hex)	Address (Dec)	Name	Description

$D040_h$	53312	VGAMOD	Set VGA mode

bit	settings	description
7–0	Current VGA mode	00_h= 800x600 72 Hz (default) 01_h= 800x600 50 Hz 02_h= 800x600 60 Hz 03_h= 800x600 75 Hz 04_h= 640x480 50 Hz 05_h= 640x480 60 Hz 06_h= 640x480 72 Hz 07_h= 640x480 75 Hz 08_h= 640x480 85 Hz 09_h= 1024x768 50 Hz $0A_h$= 1024x768 60 Hz $0B_h$= 1024x768 72 Hz $0C_h$= 1024x768 75 Hz $0D_h$= 1024x768 85 Hz $0E_h$= 1152x864 50 Hz $0F_h$= 1152x864 60 Hz 10_h= 1152x864 72 Hz 11_h= 1152x864 75 Hz 12_h= 1152x864 85 Hz 13_h= 1280x1024 50 Hz 14_h= 1280x1024 60 Hz 15_h= 1280x1024 72 Hz 16_h= 1280x1024 75 Hz 17_h= 1280x1024 85 Hz 18_h= 1600x1200 50 Hz 19_h= 1600x1200 60 Hz

$D041_h$	53313	VGACFG	Set VGA configuration

bit	settings	description
7	VIC-II Sync	0 = VGA asynchronous 1 = VGA synchronized to VIC-II chip.
6-5	Scanlines emulation	00 = All lines at full brightness. 01 = Doubled lines have 75% brightness to simulate scan-lines. 10 = Doubled lines have 50% brightness to simulate scan-lines. 11 = Doubled lines have 25% brightness to simulate scan-lines.
4–2	Scaling mode	000_b= Nearest Neighbor scaling. 001_b= Scale-2x 010_b= Alien space ship 011_b= Reserved for future use 100_b= Reserved for future use 101_b= Reserved for future use 110_b= Reserved for future use 111_b= Reserved for future use
1–0	Frame buffering mode	00_b= Reserved for future use 01_b= Double buffering 10_b= Tripple buffering (50%,50% blending for IFLI) 11_b= Reserved for future use

$D042_h$	53314	VGACOL	Set VGA color mode

bit	settings	description
7-1	Reserved, must be 0	–
0	Color palette	0 = Use standard color palette. 1 = Use custom palette.

$D043_h$	53315	VGARQT	Request info about VGA modes (result in $D044_h$–$D047_h$)
$D044_h$	53316	VGAW	VGA mode info $vga\_width_{7..0}$
$D045_h$	53317	VGAH	VGA mode info $vga\_lines_{7..0}$
$D046_h$	53318	VGAHWH	VGA mode info width and lines high bits

bit	settings	description
7–4	total $vga\_lines_{11..8}$	
3–0	total $vga\_width_{11..8}$	

$D047_h$	53319	VGAHZ	VGA mode vertical frequency in Hertz

$D048_h$	53320		Reserved
$D049_h$	53321		Reserved
$D04A_h$	53322		Reserved
$D04B_h$	53323		Reserved
$D04C_h$	53324		Reserved
$D04D_h$	53325		Reserved
$D04E_h$	53326		Reserved
$D04F_h$	53327		Reserved
$D0A0_h$	53408	MMUA0	Address offset bits A_7–A_0 of current MMU slot
$D0A1_h$	53409	MMUA1	Address offset bits A_{15}–A_8 of current MMU slot
$D0A2_h$	53410	MMUA2	Address offset bits A_{23}–A_{16} of current MMU slot
$D0A3_h$	53411	MMUA3	Address offset bit A_{24} of current MMU slot

bit	settings	description
7	read-only	0 = Block of memory can be read and written 1 = Block of memory is read-only
6–1	Reserved for address extension, must be set to 0	
0	Address offset bit A_{24}	

$D0A8_h$	53416	VERIDX	Version data character index.
$D0A9_h$	53417	VERDAT	Read out of character indexed by VERIDX.
$D0AA_h$	53418	TIMER1	Timer 1, counts up each 1 millisecond (1 Khz)
$D0AB_h$	53419	TIMER2	Timer 2, counts up each 10 milliseconds (100 Hz)
$D0AC_h$	53420	TIMER3	Timer 3, counts up each 10 milliseconds (100 Hz)
$D0AD_h$	53421	TIMER4	Timer 4, counts up each 100 milliseconds (10 Hz)
$D0AE_h$	53422	LSTBTN	Informs menu of the last button pressed

bit	settings	description
7–3	reserved, must be 0	–
2–0	Last button	000 = Nothing or unknown 010 = Left button pressed short 011 = Left button pressed long 100 = Middle (freezer) button pressed short 101 = Middle (freezer) button pressed long 110 = Right (reset) button pressed short 111 = Right (reset) button pressed long

D0AF$_h$ 53423 MMUSLT Select MMU slot

bit	settings	description
7–0	Current slot	00_h= C64 r/w memory at 0xxx$_h$ 01_h= C64 r/w memory at 1xxx$_h$ 02_h= C64 r/w memory at 2xxx$_h$ 03_h= C64 r/w memory at 3xxx$_h$ 04_h= C64 r/w memory at 4xxx$_h$ 05_h= C64 r/w memory at 5xxx$_h$ 06_h= C64 r/w memory at 6xxx$_h$ 07_h= C64 r/w memory at 7xxx$_h$ 08_h= C64 r/w memory at 8xxx$_h$ 09_h= C64 r/w memory at 9xxx$_h$ 0A$_h$= C64 r/w memory (under basic) at Axxx$_h$ 0B$_h$= C64 r/w memory (under basic) at Bxxx$_h$ 0C$_h$= C64 r/w memory at Cxxx$_h$ 0D$_h$= C64 r/w memory (under I/O) at Dxxx$_h$ 0E$_h$= C64 r/w memory (under kernal) at Exxx$_h$ 0F$_h$= C64 r/w memory (under kernal) at Fxxx$_h$ 10_h= REU internal memory (upto 16 MByte) 11_h= geoRAM internal memory (upto 4 MByte) 12_h= Freezer/Game cartridge RAM 13_h= Freezer/Game cartridge ROM 14_h= MMC64 cartridge ROM (8 KByte) 15_h= *** reserved *** 16_h= *** reserved *** 17_h= *** reserved for tape *** 18_h= Drive 8 RAM/ROM (64 KByte) 19_h= Drive 9 RAM/ROM (64 KByte) 1A$_h$= *** reserved for drive 9 *** 1B$_h$= *** reserved *** 1C$_h$= VIC-II Frame-buffer location 1D$_h$= character ROM (4 KByte) 1E$_h$= ROM at A000$_h$-BFFF$_h$ (BASIC, 8 KByte) 1F$_h$= ROM at E000$_h$-FFFF$_h$ (KERNAL, 8 KByte) 20_h= C64 r/w memory at 0000$_h$-1FFF$_h$ in menu-mode 21_h= C64 r/w memory at 2000$_h$-3FFF$_h$ in menu-mode 22_h= C64 r/w memory at 4000$_h$-5FFF$_h$ in menu-mode 23_h= C64 r/w memory at 6000$_h$-7FFF$_h$ in menu-mode 24_h= C64 r/w memory at 8000$_h$-9FFF$_h$ in menu-mode 25_h= C64 r/w memory at A000$_h$-BFFF$_h$ in menu-mode 26_h= C64 r/w memory at E000$_h$-FFFF$_h$ in menu-mode 27_h= ROM or RAM at D700$_h$-D7FF$_h$ 28_h= Drive 8 Disk tracks for virtual floppy 1 29_h= Drive 8 Disk tracks for virtual floppy 2 2A$_h$= Drive 8 Disk tracks for virtual floppy 3 2B$_h$= Drive 8 Disk tracks for virtual floppy 4 2C$_h$= Drive 9 Disk tracks for virtual floppy 1 2D$_h$= Drive 9 Disk tracks for virtual floppy 2 2E$_h$= Drive 9 Disk tracks for virtual floppy 3 2F$_h$= Drive 9 Disk tracks for virtual floppy 4 30_h–FF$_h$= *** Free for applications ***

D0F0$_h$ 53488 CFGCRT Cartridge emulation

bit	settings	description
7–0	Cartridge Type	00000000_b, 00_h = Off 00000001_b, 01_h = RetroReplay 00000010_b, 02_h = KCS Power Cartridge 00000011_b, 03_h = Final Cartridge 3 00000100_b, 04_h = Simons Basic 00000101_b, 05_h = Ocean type 1 00000110_b, 06_h = Expert Cartridge 00000111_b, 07_h = Fun Play 00001000_b, 08_h = Super Games 00001010_b, 0A$_h$ = Epyx Fastload 00001011_b, 0B$_h$ = Westermann 00001111_b, 0F$_h$ = Game System (GS), System 3 00010000_b, 10_h = WarpSpeed 00010001_b, 11_h = Dinamic 00010010_b, 12_h = (Super) Zaxxon 00010011_b, 13_h = Magic Desk 00010100_b, 14_h = Super Snapshot 5 00010101_b, 15_h = Comal-80 00010111_b, 17_h = Ross 00011100_b, 1C$_h$ = Mikro Assembler 00011111_b, 1F$_h$ = StarDos 00100000_b, 20_h = EasyFlash 00100010_b, 22_h = Capture 00101011_b, 2B$_h$ = Prophet 64 00110011_b, 33_h = Mach 5 00110101_b, 35_h = PageFox 00110110_b, 36_h = Business Basic 11111100_b, FC$_h$ = 16K ROM cartridge at 8000$_h$–BFFF$_h$ 11111101_b, FD$_h$ = 16K ROM cartridge in Ultimax mode 11111110_b, FE$_h$ = 8K ROM cartridge at 8000$_h$–9FFF$_h$ others = reserved for future use

$D0F1_h$ 53489 CFGSPI Clock-port and MMC64 Emulation

bit	settings	description
7	Reserved, must be 0	
6	Ext NMI	0 = Clock-port NMI is disabled in cartridge and docking-station mode. 1 = Clock-port NMI is always enabled (might require an extra pull-up on the NMI line in some cases)
5–4	Clock port	00_b = Off 01_b = Clock port at $DE00_h$-$DE0F_h$ 10_b = Clock port at $DF20_h$-$DF2F_h$ 11_b = reserved
3	ROM source	0 = ROMs are banked with MMU at $D0A0_h$-$D0AF_h$ 1 = C64 original Basic and Kernal ROMs are used This bit is only functional in cartridge mode. In standalone mode and on the C-One this bit should always be clear. Note that the character ROM is always emulated and never the C64 original.
2	MMC64 active	0 = MMC64 active (Copy of bit 7 in $DF11_h$) 1 = MMC64 disabled ($DF1x_h$ registers are invisible) This bit can only be toggled in $DF11_h$ after unlocking, while it can be accessed here at any time. On reset this bit is set to 1 if MMC64 emulation is disabled (bits 1–0 are zero) and 0 when emulation is enabled.
1–0	MMC64 Emulation, SPI	00_b = Off 01_b = MMC64 Emulation 10_b = reserved 11_b = MMC64 Emulation with extra bits combinations defined for access to RTC (Real Time Clock) and FlashRom.

$D0F2_h$ 53490 CFGVIC VIC-II Emulation Config

bit	settings	description
7	VIC-II Read Enable	0 = Off 1 = Perform memory accesses for VIC-II
6	Frame buffer Enable	0 = Off 1 = VIC-II emulation writes graphics to framebuffer (MMU slot $1C_h$)
5	reserved, must be 0	–
4	Force side-border open	0 = Not forced open 1 = Side-border is forced open (turbo mode must be on!)
3	reserved, must be 0	–
2–0	VIC-II type	000 = PAL (63 columns, 312 lines) 001_b = Reserved 010_b = NTSC (65 columns, 263 lines) 011_b = Old-NTSC (64 columns, 262 lines) $1xx_b$ = Reserved These bits are read-only in cartridge mode. They can be changed in standalone mode and on the C-One.

$D0F3_h$ 53491 CFGTUR Turbo configuration

bit	settings	description
7	Turbo Enable	0 = 1 Mhz mode 1 = Turbo mode active
6	Reserved, must be 0	
5	VIC-II turbo bit	0 = Off 1 = "Turbo Enable" is mirrored at bit 0 of $D030_h$
4	Auto Speed	0 = Turbo is slowed down on IEC bus accesses. 1 = Auto speed is disabled. It is recommended to keep the Auto Speed setting at 0. Otherwise accessing drives and other peripherals on the serial IEC bus might be impossible with the turbo active.
3–0	Turbo speed limit	0000_b = CPU not limited, runs at maximum speed possible. 0001_b = CPU limited to 2x normal speed 0010_b = CPU limited to 3x normal speed 0011_b = CPU limited to 4x normal speed 0100_b = CPU limited to 5x normal speed 0101_b = CPU limited to 6x normal speed 1100_b = CPU limited to 100% speed (slow-down mode) 1101_b = CPU limited to 75% speed (slow-down mode) 1110_b = CPU limited to 50% speed (slow-down mode) 1111_b = CPU limited to 25% speed (slow-down mode) others = reserved for future use Setting a speed limit determines the maximum speed the CPU is allowed to run, it can actually run slower if other factors slow it down (like I/O accesses or many cache misses). The limit is caculated using the average speed over the last 250 CPU cycles. The last 4 modes are not turbo modes, but slowdown modes. They can slowdown the CPU in 25% steps.

D0F4$_h$ 53492 CFGSID SID emulation

bit	settings	description
7	Second SID type	0 = Emulate 6581 SID-Chip 1 = Emulate 8580 SID-Chip Selects the type of SID emulated for the second SID in stereo mode.
6	First SID type	0 = Emulate 6581 SID-Chip 1 = Emulate 8580 SID-Chip Selects the type of SID emulated for the first SID at D400$_h$.
5–3	Stereo SID in C64	Specify where the second SID chip is located inside the C64 memory space. 000$_b$ = Single SID in C64 001$_b$ = Second SID in C64 at D420$_h$ 010$_b$ = reserved 011$_b$ = reserved 100$_b$ = Second SID in C64 at D500$_h$ 101$_b$ = Second SID in C64 at D700$_h$ 110$_b$ = Second SID in C64 at DE00$_h$ 111$_b$ = Second SID in C64 at DF00$_h$ For C-One use "000" when zero or one SID-Chip is placed and "001" when both SID-Chips are present. For stereo SID chip to properly work you also need to turn Chameleon SID emulation in stereo mode.
2–0	Stereo SID (emulation)	Specify if and where the second SID must be placed in chameleon memory space. This doesn't have to be the same address as a second SID chip inside the C64 as Chameleon can translate the address automatically. 000$_b$ = Emulate single SID 001$_b$ = Emulate stereo SID use A$_5$ for selection (D420$_h$, D460$_h$, D4A0$_h$ ...) 010$_b$ = reserved 011$_b$ = reserved 100$_b$ = Emulate stereo SID use A$_8$ for selection (D5xx$_h$, D7xx$_h$) 101$_b$ = Same as setting 100$_b$ (D5xx$_h$, D7xx$_h$) 110$_b$ = Emulate stereo SID use IO1 for second SID (DE00$_h$) 111$_b$ = Emulate stereo SID use IO2 for second SID (DF00$_h$) others = reserved for future use When only a single SID chip is emulated (setting "000") it is played through both audio output channels in mono. For any of the stereo settings, the first SID-Chip emulation is played through the left audio output channel and is always located in memory at D400$_h$. The second SID-Chip emulation is played through the right audio output channel.

D0F5$_h$ 53493 CFGREU REU (Ram Expansion Unit) and geoRAM Emulation Config

bit	settings	description
7	Enable REU	0 = REU is disabled (off) 1 = Enable REU emulation and activate registers at DF00$_h$–DF0A$_h$
6	Enable geoRAM	0 = geoRAM is disabled (off) 1 = Enable geoRAM emulation and activate registers at DE00$_h$–DEFF$_h$, DFFE$_h$ and DFFF$_h$
5–3	geoRAM size	000$_b$ = 64 KByte 001$_b$ = 128 KByte 010$_b$ = 256 KByte 011$_b$ = 512 KByte 100$_b$ = 1 MByte 101$_b$ = 2 MByte 110$_b$ = 4 MByte 111$_b$ = reserved for future use
2–0	REU memory size	000$_b$ = 128 KByte 001$_b$ = 256 KByte 010$_b$ = 512 KByte 011$_b$ = 1 MByte 100$_b$ = 2 MByte 101$_b$ = 4 MByte 110$_b$ = 8 MByte 111$_b$ = 16 MByte (Note there is not enough RAM on C-One for this setting)

D0F6$_h$ 53494 CFGDWR Floppy-disk image write bits

bit	settings	description
7		1 = Writes have been done by drive 9 to floppy image 4.
6		1 = Writes have been done by drive 9 to floppy image 3.
5		1 = Writes have been done by drive 9 to floppy image 2.
4		1 = Writes have been done by drive 9 to floppy image 1.
3		1 = Writes have been done by drive 8 to floppy image 4.
2		1 = Writes have been done by drive 8 to floppy image 3.
1		1 = Writes have been done by drive 8 to floppy image 2.
0		1 = Writes have been done by drive 8 to floppy image 1.
		Bits are set when the emulated drive writes to the floppy image. Menu software can use these bits to write updated image back to sd card. Bits can be reset by writing 0 to the register.

$D0F7_h$ 53495 CFGDSK Disk images

bit	settings	description
7-6	Disk 9 floppy range	Number of floppy images for drive 9 00_b = 1 image 01_b = 2 images 10_b = 3 images 11_b = 4 images
5-4	Disk 9 floppy select	Select floppy image for drive 9 00_b = floppy image 1 selected 01_b = floppy image 2 selected 10_b = floppy image 3 selected 11_b = floppy image 4 selected
3-2	Disk 8 floppy range	Number of floppy images for drive 8 00_b = 1 image 01_b = 2 images 10_b = 3 images 11_b = 4 images
1-0	Disk 8 floppy select	Select floppy image for drive 8 00_b = floppy image 1 selected 01_b = floppy image 2 selected 10_b = floppy image 3 selected 11_b = floppy image 4 selected

$D0F8_h$ 53496 CFGFD0 Drive emulation

bit	settings	description
7–6	Enable virtual-drive CPU	00_b = drive cpu stopped 01_b = drive cpu running
5	Drive door	0_b = Drive door closed 1_b = Drive door open
4–3	Reserved, must be 0	–
2	Drive memory size	0 = 2 Kbyte (default) 1 = 8 Kbyte (not implemented in beta firmware!)
1–0	Drive ID jumpers	00_b = drive device id is 8 01_b = drive device id is 9 10_b = drive device id is 10 11_b = drive device id is 11

$D0F9_h$ 53497 CFGFD1 Reserved for second drive

bit	settings	description
7–6	Enable virtual-drive CPU	00_b = drive cpu stopped 01_b = drive cpu running
5	Drive door	0_b = Drive door closed 1_b = Drive door open
4–3	Reserved, must be 0	–
2	Drive memory size	0 = 2 Kbyte (default) 1 = 8 Kbyte (not implemented in beta firmware!)
1–0	Drive ID jumpers	00_b = drive device id is 8 01_b = drive device id is 9 10_b = drive device id is 10 11_b = drive device id is 11

$D0FA_h$ 53498 CFGREG Enable Chameleon registers

bit	settings	description
7–6	reserved, must be 0	–
5	Chameleon RAM or ROM at $D700_h$	0 = $D700_h$–$D7FF_h$ has SID mirrors 1 = RAM or ROM with banking and trampoline-code for the menu is mapped at $D700_h$–$D7FF_h$
4	reserved, must be 0	–
3	Palette Registers Enable	0 = VIC-II chip mirrors at $D100_h$–$D3FF_h$ 1 = Palette registers are at $D100_h$–$D3FF_h$
2	reserved, must be 0	–
1	Enable MMU/Timer registers	0 = VIC-II chip mirrors at $D0A0_h$–$D0AF_h$ 1 = Chameleon MMU/Timer registers at $D0A0_h$–$D0AF_h$
0	Enable VGA Controller Registers	0 = VIC-II chip mirrors at $D040_h$–$D07F_h$ 1 = VGA/COP registers at $D040_h$–$D07F_h$

$D0FB_h$ 53499 CFGBTN Debug info and Buttons

bit	settings	description
7-6	Debug info on VGA	00 = No debug information 01 = Show memory and cache load and also main 6510 CPU state on the top of the screen. 10 = Show memory, cache, 6510 and drive CPU state. 11 = Show all debug information (note this uses a considerable amount of screen space).
5–4	Reserved, must be 0	
3–0	Left button configuration	

short		long
0000	Menu	–
0001	Cartridge On/Off	Cartridge Prg (expert)
0010	Toggle Turbo On/Off	–
0100	Disk change drive 8 (next)	Disk change drive 8 (first)
0101	Disk change drive 9 (next)	Disk change drive 9 (first)
others	*** reserved ***	

D0FC$_h$ 53500 CFGIO I/O and IEC configuration

bit	settings	description
7	IEC port	0 = Chameleon IEC bus connected to virtual CIAs 1 = Chameleon IEC bus and any emulated disk-drives are disconnected from the system By setting this bit, the Chameleon IEC bus is disconnected from the C64 side. In this mode the Chameleon can function as a 1541 drive emulator. This feature is not available on the C-One due to a hardware limitation.
6	IEC reset	0 = Normal operation 1 = Chameleon virtual drives are held in reset
5	PS/2 mouse enable	0 = Autodetect mouse on PS/2 port and activate 1351 emulation if found. 1 = 1351 emulation disabled (Paddle inputs on joystick ports can be used)
4	PS/2 mouse port	0 = Mouse emulation on port 1 1 = Mouse emulation on port 2
3	IR receiver	0 = IR is enabled (on) 1 = IR is disabled (off)
2	Menu-mode on reset	0 = Reset to C64 mode 1 = Reset to menu-mode. Menu will be displayed after pressing reset button.
1	Reserved, always 0	
0	C64 IEC bus	0 = C64 IEC bus active 1 = C64 IEC bus inactive This bit only has a function in cartridge mode. In standalone mode it is ignored.

D0FD$_h$ 53501 CFGDIS A write (any value) leaves configuration mode. A read returns current flash slot where the FPGA image is started from.

bit	settings	description
7	VIC-II emulation error	0 = VGA emulation in sync. with VIC-II chip 1 = Error, VGA and VIC-II chip not in sync. This bit has a valid value in cartridge mode only. It is always 0 in standalone mode.
6	Reserved, always 0	
5	Readback of reset line	0 = No reset pending (reset line is high) 1 = Line constantly held in reset. The clock-port can not be used due to missing pull-up resistor. The status of the reset can be checked to determine if the clock-port has the required pull-up in standalone and docking-station modes. If this flag is set the pull-up is missing and the clock-port can not be used. It should always read 0 in cartridge mode.
4	Flash slot valid	0 = Slot number is unknown 1 = Slot number is valid This bit should always be 1. If 0 it means the USB micro didn't respond to the initialization request. It might have crashed or there is a hardware fault preventing communication.
3–0	Flash slot	One of 16 slots where the FPGA started from.

D0FE$_h$ 53502 CFGENA Write 42 (2A$_h$) to enter configuration mode. When in configuration mode write 16 to 31 (10$_h$ to 1F$_h$) to reconfigure the FPGA with a new core. The 4 lower bits specify the slot number in the onboard flash. Write 32 (20$_h$) in configuration mode to force menu mode. Write 33 (21$_h$) in configuration mode to force menu mode with NMI interrupts disabled. Write 34 (22$_h$) to enable NMI interrupts after freezing to menu. Write 35 (23$_h$) to disable NMIs again. Write 165 (A5$_h$) to reset machine. Write 166 (A6$_h$) to reset and leave configuration mode.

D0FF$_h$ 53503 CFGRTI A write (any value) leaves configuration mode. A read leaves menu mode.

D100$_h$ –D1FF$_h$ 53504 –53759 PALRED 256 entry color palette **Red** intensity

D200$_h$ –D2FF$_h$ 53760 –54015 PALGRN 256 entry color palette **Green** intensity

D300$_h$ –D3FF$_h$ 54016 –54271 PALBLU 256 entry color palette **Blue** intensity

Action Replay / RetroReplay

DE00$_h$ 56832 RRCTRL RR control register (on write)

bit	settings	description
7	A15	ROM address line 15
6		
5	ROM/RAM	0 = ROM 1 = RAM
4	A14	ROM/RAM address line 14
3	A13	ROM/RAM address line 13
2	Disable	Write 1 to disable cartridge
1	EXROM	
0	GAME (inverted)	

$DE01_h$ 56833 RREXTD RR extended control register (on write)

bit	settings	description
7	A15	ROM address line 15 (mirror of $DE00_h$)
6	REU Compatibility	0 = Standard memory map 1 = REU compatible memory map
5		Not implemented, must be set to 0
4	A14	ROM/RAM address line 14 (mirror of $DE00_h$)
3	A13	ROM/RAM address line 13 (mirror of $DE00_h$)
2		Not implemented, must be set to 0
1	AllowBank	0 = no RAM banking in $DE02_h$–$DFFF_h$ area 1 = Enable RAM banking in $DE02_h$–$DFFF_h$ area
0		Not implemented, must be set to 0

$DE00_h$ –$DE01_h$ 56832 –56833 RRSTAT RR status (on read)

bit	settings	description
7	A15	ROM address line 15
6		
5		Not implemented, reads 0
4	A14	ROM/RAM address line 14
3	A13	ROM/RAM address line 13
2		
1		
0		Not implemented, reads 0

REU

$DF00_h$ 57088 DMAST REU Status register (read-only)

bit	settings	description
7	1 = IRQ pending	
6	1 = End of block	
5	1 = Fault	Compare operation detected a difference
4	Size	0 = 128 KByte 1 = 256 or 512 KByte A single bit can't represent all possible memory sizes. So software should probe for the amount that is really available.
3–0	Version	Always 0000

$DF01_h$ 57089 DMACMD REU Command register

bit	settings	description
7	1 = Execute	
6	Reserved	–
5	1 = Auto load	When autoload is enabled. The memory pointers and length registers are reloaded at the end of the transfer
4	$FF00_h$ flag	0 = Wait for write to $FF00_h$ before starting transfer 1 = Start immediately when bit 7 becomes set
3–2	Reserved	–
1–0	Transfer type	00 = C64 to REU 01 = REU to C64 10 = Swap 11 = Compare / verify

$DF02_h$ 57090 DMA64L C64 memory pointer low

$DF03_h$ 57091 DMA64H C64 memory pointer high

$DF04_h$ 57092 DMAINL REU memory pointer low

$DF05_h$ 57093 DMAINM REU memory pointer mid

$DF06_h$ 57094 DMAINH REU memory pointer high

$DF07_h$ 57095 DMACNL Transfer length low

$DF08_h$ 57096 DMACNH Transfer length high

$DF09_h$ 57097 DMAINT Interrupt mask register

bit	settings	description
7	Interrupt enable	1 = enabled
6	End Of Block mask	1 = interrupt after transfer
5	Verify mask	1 = interrupt on verify error
4–0	Reserved	Read as 1

$DF0A_h$ 57098 DMACTL Address control register

bit	settings	description
7	C64 Address control	0 = Increment C64 address 1 = Fixed C64 address
6	REU Address control	0 = Increment REU address 1 = Fixed REU address
5–0	Reserved	Read as 1

MMC64

$DF10_h$		57104	MMCSPI	SPI transfer register. Write in this register sends byte to SPI bus, read is last retrieved byte.
$DF11_h$		57105	MMCCTL	MMC64 Control register.

bit	settings	description
7	MMC64 active	0 = MMC64 is active 1 = MMC64 is disabled Bit can only be modified when unlocked
6	SPI trigger mode	0 = Trigger SPI transfer on write to register $DF10_h$ 1 = Trigger SPI transfer on read of register $DF10_h$
5	External ROM	0 = Allow external ROM when BIOS is disabled 1 = Disable external ROM
4	Flash mode	0 = Normal mode 1 = Flash update mode Not implemented, must be set to 0
3	Clock port address	Not implemented, must be set to 0
2	Clock Speed	0 = 250 KHz SPI clock 1 = 8 Mhz SPI clock
1	MMC cart select	0 = Cart selected 1 = Cart not selected
0	MMC64 Bios	0 = MMC64 BIOS ROM active 1 = BIOS ROM disabled (external ROM active)

$DF12_h$		57106	MMCST	MMC64 Status register (read-only).

bit	settings	description
5	Flash jumper	Not implemented reads always as 0
4	MMC Write Protect	0 = Cart can be written 1 = Cart is write protected
3	MMC Cart Detect	0 = Cart inserted 1 = No cart present, slot empty
2	External EXROM line	
1	External GAME line	
0	Busy	0 = SPI bus ready 1 = SPI bus busy (only for 250 Khz mode)

MMC64 additional SPI devices

$DF11_h$		57105	MMCCTL	MMC64 Control register.

bit	settings	description
4,1	Select	00 = MMC cart selected 01 = Nothing selected 10 = Flash ROM selected 11 = RTC (Real Time Clock) selected

GeoRAM

$DE00_h$ -$DEFF_h$		56832	–57087 GEOBUF	geoRAM 256 byte memory window
$DFFE_h$		57342	GEOLOW	geoRAM address A_{13}–A_8

bit	settings	description
7–6	Unused	must be set to 0
5–0	geoRAM A_{13}–A_8	

$DFFF_h$		57343	GEOHI	geoRAM address A_{21}–A_{14}

Final Cartridge 3

$DE00_h$ –$DFFF_h$		56832	–57343	Reads will read cartridge ROM at $1E00_h$–$1FFF_h$, $5E00_h$–$5FFF_h$, $9E00_h$–$9FFF_h$ or $DE00_h$–$DFFF_h$ depending on the current selected bank.
$DFFF_h$		57343	FC3BNK	On write

bit	settings	description
7	register enable	0 = Banking register writable at DFFF. 1 = Banking register invisible. On Chameleon setting this bit to 1 also disables the ROM mirror at $DE00_h$–$DFFF_h$.
6	NMI	0 = Force NMI line low 1 = Normal operation
5	GAME	State of the GAME line
4	EXROM	State of the EXROM line
3	unused	
2	unused	
1	A15	ROM address line 15
0	A14	ROM address line 14