Turbo Chameleon 64

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

The Programmers Manual

Peter Wendrich
pwsoft@syntiac.com

March 5, 2011

Draft version!

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 cores. 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. It can emulate 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

1.3 Chameleon core for the C-One Reconfigurable Computer

Both the Chameleon cartrigde and the C-One extender board are based on the same type of FPGA. Therefore it made sense to release a Chameleon core for the C-One with extender board. The C-One version of Chameleon behaves like the standalone mode of the Chameleon Cartridge. Because the hardware is different there are some small differences between the two cores. The most important difference are in the amount of memory available and the layout of the memory map.

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
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 7 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 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.

3.1 Allocated memory ranges

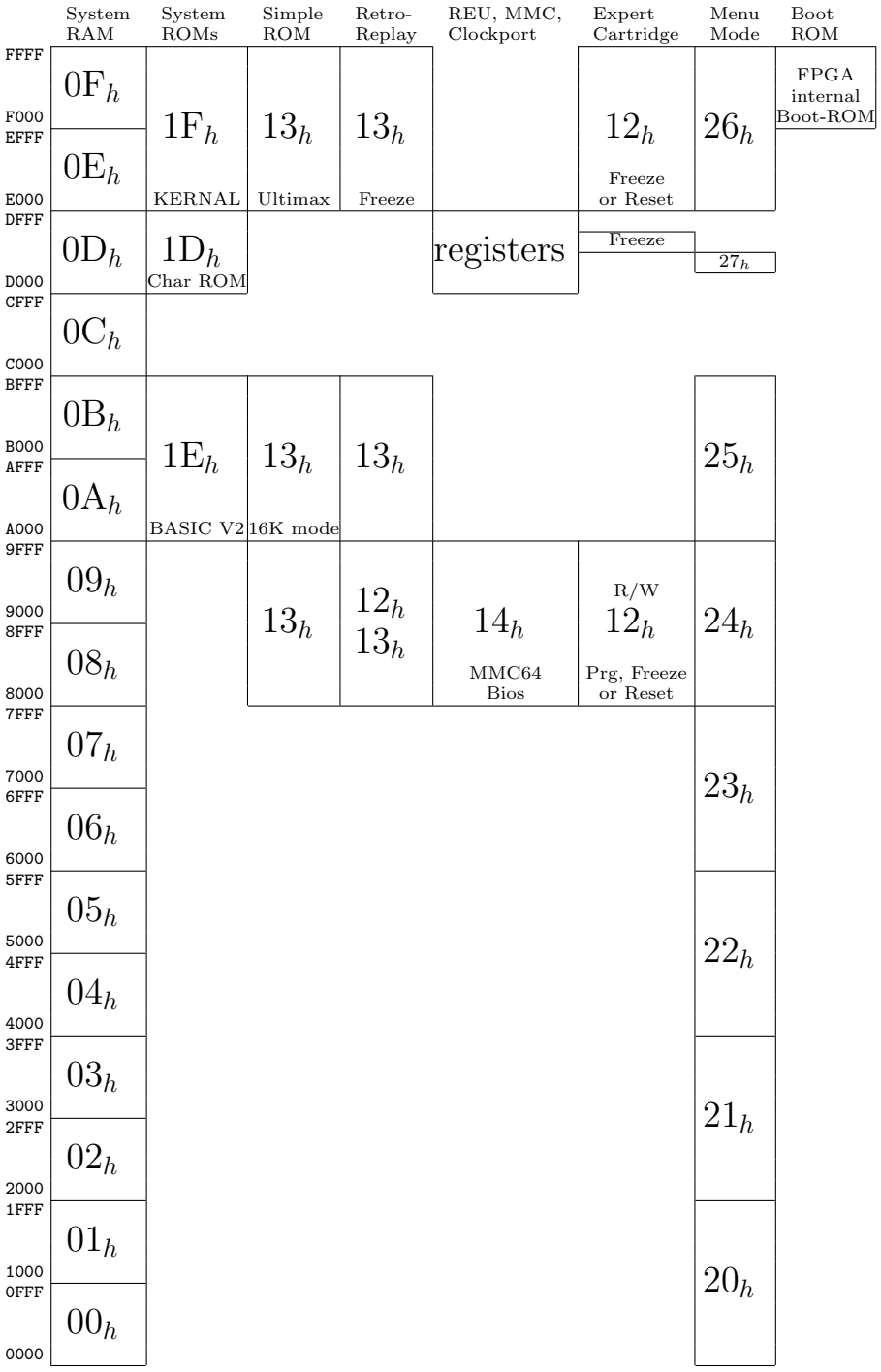
3.1.1 32 MByte Layout

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

3.2 Memory overlays



3.3 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		Address offset bits A_7–A_0 of current MMU slot
$D0A1_h$	53409		Address offset bits A_{15}–A_8 of current MMU slot
$D0A2_h$	53410		Address offset bits A_{23}–A_{16} of current MMU slot
$D0A3_h$	53411		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}	

$D0A4_h$ –$D0AE_h$	53412	–53422	Reserved for future use
$D0AF_h$	53423		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 Buttons

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

- Left, Turbo On/Off toggle
- Middle, short press is Freeze, long press accesses Chameleon menu
- Right, short press is Reset, long press restarts to core selection menu

4.1 Buttons Configuration Registers

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	Buttons configuration	

	Left		Middle		Right	
	short	long	short	long	short	long
0000	Menu	–	Freeze	–	Reset	Reboot from Flash
others	*** reserved ***					

5 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. If compatibility with a stock C64 is not required, the VGA controller can be reprogrammed to provide higher resolutions and more colors.

5.1 VGA Resolution and Sync Registers

Address (Hex)	Address (Dec)	Name	Description
$D040_h$			VGA Visual X-size$_{7..0}$
$D041_h$			VGA Visual Y-size$_{7..0}$
$D042_h$			VGA Visual size upper bits

bit	settings	description
7–4	visual Y-size$_{11..8}$	
3–0	visual X-size$_{11..8}$	

$D043_h$			VGA total X-size$_{7..0}$
$D044_h$			VGA total Y-size$_{7..0}$
$D045_h$			VGA total size upper bits

bit	settings	description
7–4	total Y-size$_{11..8}$	
3–0	total X-size$_{11..8}$	

$D046_h$			VGA HSync start$_{7..0}$
$D047_h$			VGA HSync end$_{7..0}$
$D048_h$			VGA HSync upper bits

bit	settings	description
7–4	HSync end$_{11..8}$	
3–0	HSync start$_{11..8}$	

$D049_h$			VGA VSync start$_{7..0}$
$D04A_h$			VGA VSync end$_{7..0}$
$D04B_h$			VGA VSync upper bits

bit	settings	description
7–4	VSync end$_{11..8}$	
3–0	VSync start$_{11..8}$	

$D04C_h$			Select current object Object registers are at $D050_h$-$D05F_h$
$D04D_h$			First object to render
$D04E_h$			Last object to render
$D04F_h$			Polarity and Pixel-clock

bit	settings	description
7	VSync polarity	0 = negative sync 1 = positive sync
6	HSync polarity	0 = negative sync 1 = positive sync
5	Enable VGA VSync Interrupt	0 = disabled 1 = enabled
4	VGA VSync Interrupt status	0 = no interrupt 1 = pending Interrupt status is cleared on any write to $D04F_h$
3–0	Pixel-clock frequency	0000 = 25.175 Mhz 0001 = 31.5 Mhz 0010 = Reserved for 36 Mhz 0011 = Reserved for 40 Mhz 0100 = 50 Mhz 0101 = Reserved for 65 Mhz 0110 = Reserved for 75 Mhz 0111 = Reserved for 78.8 Mhz 1000 = 108 Mhz others = Reserved for future use

5.2 Example settings for standard VGA modes

Screen mode	Visual Size		Total Size		HSync	VSync	Polarity
	W	H	W	H	start / end	start / end	& Pixelclock
640x480 @ 60Hz	640	480	?				H=n / V=p / 25.175
800x600 @ 72Hz	800	600	1040	666	860 / 980	625 / 631	H=p / V=p / 50
1024x768 @ 70Hz	1024	768	1328	806			H=n / V=n / 75
1024x768 @ 75Hz	1024	768	1312	800			H=p / V=p / 78.8
1280x1024 @ 60Hz	1280	1024	1688	1066			H=p / V=p / 108

5.3 Chameleon Object Processor

The COP (Chameleon Object Processor) is a separate processor designed to perform graphic tasks. It can render multiple moving objects to the screen, these objects can take the form of sprites/MOBs or even complete bitmaps (with smooth scrolling in all directions). Objects of different resolutions (pixels can be stretched both horizontally and vertically) and different bit-depths can be combined on the same screen. A object can be configured as a window showing only a part of a larger bitmap. Combining this feature with smooth scroll and byte wise addressing allows the window to be positioned anywhere on the bitmap. Objects that overlap other objects can have one of their colors set to transparent to show the objects below it.

Programming the COP is done with 19 registers, 16 of these are mapped at $D050_h$ and are used to configure the objects. The register at $D04C_h$ selects one of a possible 256 objects to configure. Registers $D04D_h$ and $D04E_h$ allow selection of a sub-range of objects. Only the objects within this range are rendered to the screen. The drawing order is fixed, an object with lower index number is always behind an object with a higher index number. All other objects outside the range are invisible (and won't use any memory nor object processor bandwidth).

Address (Hex)	Address (Dec)	Name	Description
$D050_h$	53328	COPXL	X position$_{7..0}$
$D051_h$	53329	COPYL	Y position$_{7..0}$
$D052_h$	53330	COPYXH	position upper bits

bit	settings	description
7–4	Y position$_{11..8}$	
3–0	X position$_{11..8}$	

$D053_h$	53331	COPWL	X size$_{7..0}$
$D054_h$	53332	COPHL	Y size$_{7..0}$
$D055_h$	53333	COPHWH	size upper bits

bit	settings	description
7–4	Y size$_{11..8}$	
3–0	X size$_{11..8}$	

$D056_h$	53334	COPLIL	Line increment low
$D057_h$	53335	COPLIH	Line increment high
$D058_h$	53336	COPMMU	MMU slot
$D059_h$	53337		Stretch and flip

bit	settings	description
7	Vertical flip	0 = normal 1 = flipped / mirror
6–4	stretch	000 = normal size 001 = double height pixels 010 = 4x height pixels 011 = 8x height pixels 100 = 16x height pixels others = Reserved for future use
3	Horizontal flip	0 = normal 1 = flipped / mirror
2–0	stretch	000 = normal size 001 = double width pixels 010 = 4x width pixels 011 = 8x width pixels 100 = 16x width pixels others = Reserved for future use

$D05A_h$	53338	Horizontal smooth scroll (in pixels)
$D05B_h$	53339	Vertical smooth scroll (in lines)
$D05C_h$	53340	Palette offset
$D05D_h$	53341	*** free ***
$D05E_h$	53342	Group and Alpha

bit	settings	description
7	–	
6–4	Collision group	Selects group for collision detection
3–0	Alpha	Alpha-blending value in 6% steps (1/16th) 0000 = Fully opaque (100% new) .... 1111 = 6% of the new color and 94% of background

$D05F_h$	53343	Mode selection

bit	settings	description
7	Command	0 = Render bitmap graphics 1 = Define tile-set or mask
When bit 7 (command) is 0		
6	Clip rectangle	0 = use existing clip rectangle 1 = Set new clip rectange from object position and dimensions. All following objects will clip to the boundary of this object.
5	Use masking	0 = No masking 1 = Use previously set mask
4	Enable color dither	0 = 5 bits color channels (truncated) 1 = 8 bits color channels (dithered)
3	Color keying	0 = object is fully opaque. 1 = color 0 is transparent.
2–0	Color depth	000 = Solid color 001 = 1 bit/pixel, 2 palette colors 010 = 2 bits/pixel, 4 palette colors 011 = 4 bits/pixel, 16 palette colors 100 = 8 bits/pixel, 256 palette colors 101 = 16 bits/pixel, 32768 color mode 110 = 8 bits/tile, 256 tiles 111 = 16 bits/tile, 256 tiles with palette offset
When bit 7 (command) is 1		
TBD		

5.4 Palette Registers

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

5.5 Fixed Palette Entries

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

6 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.

6.1 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.

6.2 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 map of various cartridges are therefore not emulated.

6.3 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.

6.4 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)

Address (Hex)	Address (Dec)	Name	Description
$DF12_h$	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)

6.4.1 MMC64 + SPI

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.

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

6.5 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.

6.5.1 REU (Ram Expansion Unit) 1700, 1750, 1764 registers

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.

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 or 256 KByte 1 = 512 KByte A single bit can't represent all 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 = Fix C64 address
6	REU Address control	0 = Increment REU address 1 = Fix REU address
5–0	Reserved	Read as 1

6.5.2 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 geoRAM register layout allows upto 4 Mbyte of internal memory. The real geoRAM cartridge has 512 KByte.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$ -$DEFF_h$	? ?	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}

6.6 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
$DE00_h$	?	?	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)	

$DE01_h$	?	?	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

$DE00_h$–$DE01_h$	?	?	?	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.

6.7 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 agian. This makes it possible for example to use the MMC64 BIOS to load a program from the MMC card or for example use the 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.

Address (Hex)	Address (Dec)	Name	Description
$DE00_h$ –$DFFF_h$			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

6.8 The Expert Cartridge

6.9 Game cartridges

There are game cartridges that need (much) more ROM space as the 16 Kbyte provided by the standard memory layouts. These cartridges use bank switching logic inside the cartridge to map different parts of the ROM into memory. A couple of these bank switching methods can be emulated by Chameleon. This allows the games to be played without unplugging the Chameleon. All that is needed to play the game is a copy of the game catridge ROM contents on a MMC card or in the onboard flash chip.

6.10 Configuration registers

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

bit	settings	description
7–0	Freezer/Game Type	00000000 = Off 00000001 = RetroReplay 00000011 = Final Cartridge 3 00000110 = Expert Cartridge 11111100 = 16K ROM cartridge at 8000_h–$BFFF_h$ 11111101 = 16K ROM cartridge in Ultimax mode 11111110 = 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
7–6	Reserved, must be 0	
5–4	Clock port	00 = Off 01 = Clock port at $DE00_h$-$DE0F_h$ 10 = Clock port at $DF20_h$-$DF2F_h$ 11 = reserved
3	Reserved, must be 0	
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 = Off 01 = MMC64 Emulation 10 = reserved 11 = 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	1 = Enable REU	Enable REU emulation and activate registers at $DF00_h$–$DF0A_h$.
6	1 = Enable geoRAM	Enable geoRAM emulation and activate registers at $DE00_h$–$DEFF_h$, $DFFE_h$and $DFFF_h$
5–3	geoRAM size	000 = 64 KByte 001 = 128 KByte 010 = 256 KByte 011 = 512 KByte 100 = 1 MByte 101 = 2 MByte 110 = 4 MByte 111 = reserved for future use
2–0	REU memory size	000 = 128 KByte 001 = 256 KByte 010 = 512 KByte 011 = 1 MByte 100 = 2 MByte 101 = 4 MByte 110 = 8 MByte 111 = 16 MByte (Note there is not enough RAM on C-One for this setting)

6.11 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.

7 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.

7.1 Entering menu mode

Menu mode can be entered in three different ways:

- The menu mode is active after a reset
- Pressing the freeze button longer as 0.7 seconds
- Writing 32 (20_h) into 53502 ($D0FE_h$) while in configuration mode

7.2 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 $0C_h$.

7.3 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.

7.4 Differences between Menu and Configuration modes

TODO!

7.5 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 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, it stays active even in C64 mode. This allows basic or kernal vector hooks to point into this area, where extra 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.

7.6 Leaving menu mode

7.6.1 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 $D0FF_h$.

7.6.2 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 ($D0FE_h$).

7.7 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 ($D0FE_h$).

8 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 two speeds either double speed or maximum. At double 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.

8.1 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	Turbo switch	0 = Manual setting 1 = Turbo mode is switchable with button
5	VIC-II turbo bit	0 = Off 1 = "Turbo Enable" is mirrored at bit 0 of $D030_h$
4–1	Reserved, must be 0	
0	Turbo speed limit	0 = maximum speed 1 = CPU limited to 2 Mhz

9 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 D64 without going though 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.

9.1 Drive Memory Map

9.2 Disk track layout

9.3 Drive Configuration Registers

10 SID Emulation

10.1 SID Configuration Register

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

bit	settings	description
7	Reserved, must be 0	
6	SID type	0 = Emulate 6581 SID-Chip(s) 1 = Emulate 8580 SID-Chip(s) Not implemented in beta firmware, must be set to 0!
5–3	Stereo SID in C64	000 = Single SID in C64 001 = Second SID in C64 at $D420_h$ 010 = reserved 011 = reserved 100 = Second SID in C64 at $D500_h$ 101 = Second SID in C64 at $D700_h$ 110 = Second SID in C64 at $DE00_h$ 111 = 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.
2–0	SID emulation	000 = Emulate single SID 001 = Emulate stereo SID use A5 for selection ($D420_h$, $D460_h$, $D4A0_h$ ...) 110 = Emulate stereo SID use IO1 for second SID ($DE00_h$) 111 = 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.

11 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.

11.1 PS/2 Keyboard layout

PS/2 keyboard key	C64 function	PS/2 keyboard key	C64 function
ESCAPE	RUN/STOP	NUM-LOCK	Select port 1 or port 2
F1	F1	Numeric 0	Joystick Fire Button
F2	RShift + F1	Numeric 1	Joystick Left + Down
F3	F3	Numeric 2	Joystick Down
F4	RShift + F3	Numeric 3	Joystick Right + Down
F5	F5	Numeric 4	Joystick Left
F6	RShift + F5	Numeric 6	Joystick Right
F7	F7	Numeric 7	Joystick Left + Up
F8	RShift + F8	Numeric 8	Joystick Up
∼	Left-Arrow	Numeric 9	Joystick Right + Up

12 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. Although both two buttons mice and three buttons mice with scroll-wheel (known as intelli-mouse) can be used, the scroll-wheel does not have any function.

The Chameleon emulates a commodore 1351 mouse. Therefore the PS/2 mouse is usable together with most exisiting software packages that have mouse support.

13 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)
POWER	Right push button (Reset)
Vol Up	+
Vol Down	-
Switch in MOUSE position	Joystick 1
Switch in JOY position	Joystick 2
A	Fire
B	Auto fire (8 Hz)

14 Complete register map

Address (Hex)	Address (Dec)	Name	Description
$D040_h$			VGA Visual X-size$_{7..0}$
$D041_h$			VGA Visual Y-size$_{7..0}$
$D042_h$			VGA Visual size upper bits

bit	settings	description
7–4	visual Y-size$_{11..8}$	
3–0	visual X-size$_{11..8}$	

$D043_h$			VGA total X-size$_{7..0}$
$D044_h$			VGA total Y-size$_{7..0}$
$D045_h$			VGA total size upper bits

bit	settings	description
7–4	total Y-size$_{11..8}$	
3–0	total X-size$_{11..8}$	

$D046_h$			VGA HSync start$_{7..0}$
$D047_h$			VGA HSync end$_{7..0}$
$D048_h$			VGA HSync upper bits

bit	settings	description
7–4	HSync end$_{11..8}$	
3–0	HSync start$_{11..8}$	

$D049_h$			VGA VSync start$_{7..0}$
$D04A_h$			VGA VSync end$_{7..0}$
$D04B_h$			VGA VSync upper bits

bit	settings	description
7–4	VSync end$_{11..8}$	
3–0	VSync start$_{11..8}$	

$D04C_h$	Select current object
	Object registers are at $D050_h$-$D05F_h$
$D04D_h$	First object to render
$D04E_h$	Last object to render
$D04F_h$	Polarity and Pixel-clock

bit	settings	description
7	VSync polarity	0 = negative sync 1 = positive sync
6	HSync polarity	0 = negative sync 1 = positive sync
5	Enable VGA VSync Interrupt	0 = disabled 1 = enabled
4	VGA VSync Interrupt status	0 = no interrupt 1 = pending Interrupt status is cleared on any write to $D04F_h$
3–0	Pixel-clock frequency	0000 = 25.175 Mhz 0001 = 31.5 Mhz 0010 = Reserved for 36 Mhz 0011 = Reserved for 40 Mhz 0100 = 50 Mhz 0101 = Reserved for 65 Mhz 0110 = Reserved for 75 Mhz 0111 = Reserved for 78.8 Mhz 1000 = 108 Mhz others = Reserved for future use

$D050_h$	53328	COPXL	X $position_{7..0}$
$D051_h$	53329	COPYL	Y $position_{7..0}$
$D052_h$	53330	COPYXH	position upper bits

bit	settings	description
7–4	Y $position_{11..8}$	
3–0	X $position_{11..8}$	

$D053_h$	53331	COPWL	X $size_{7..0}$
$D054_h$	53332	COPHL	Y $size_{7..0}$
$D055_h$	53333	COPHWH	size upper bits

bit	settings	description
7–4	Y $size_{11..8}$	
3–0	X $size_{11..8}$	

$D056_h$	53334	COPLIL	Line increment low
$D057_h$	53335	COPLIH	Line increment high
$D058_h$	53336	COPMMU	MMU slot
$D059_h$	53337		Stretch and flip

bit	settings	description
7	Vertical flip	0 = normal 1 = flipped / mirror
6–4	stretch	000 = normal size 001 = double height pixels 010 = 4x height pixels 011 = 8x height pixels 100 = 16x height pixels others = Reserved for future use
3	Horizontal flip	0 = normal 1 = flipped / mirror
2–0	stretch	000 = normal size 001 = double width pixels 010 = 4x width pixels 011 = 8x width pixels 100 = 16x width pixels others = Reserved for future use

$D05A_h$	53338	Horizontal smooth scroll (in pixels)
$D05B_h$	53339	Vertical smooth scroll (in lines)
$D05C_h$	53340	Palette offset
$D05D_h$	53341	*** free ***
$D05E_h$	53342	Group and Alpha

bit	settings	description
7	–	
6–4	Collision group	Selects group for collision detection
3–0	Alpha	Alpha-blending value in 6% steps (1/16th) 0000 = Fully opaque (100% new) .... 1111 = 6% of the new color and 94% of background

$D05F_h$ 53343 Mode selection

bit	settings	description
7	Command	0 = Render bitmap graphics 1 = Define tile-set or mask
When bit 7 (command) is 0		
6	Clip rectangle	0 = use existing clip rectangle 1 = Set new clip rectange from object position and dimensions. All following objects will clip to the boundary of this object.
5	Use masking	0 = No masking 1 = Use previously set mask
4	Enable color dither	0 = 5 bits color channels (truncated) 1 = 8 bits color channels (dithered)
3	Color keying	0 = object is fully opaque. 1 = color 0 is transparent.
2–0	Color depth	000 = Solid color 001 = 1 bit/pixel, 2 palette colors 010 = 2 bits/pixel, 4 palette colors 011 = 4 bits/pixel, 16 palette colors 100 = 8 bits/pixel, 256 palette colors 101 = 16 bits/pixel, 32768 color mode 110 = 8 bits/tile, 256 tiles 111 = 16 bits/tile, 256 tiles with palette offset
When bit 7 (command) is 1		
TBD		

$D0A0_h$ 53408 Address offset bits A_7–A_0 of current MMU slot

$D0A1_h$ 53409 Address offset bits A_{15}–A_8 of current MMU slot

$D0A2_h$ 53410 Address offset bits A_{23}–A_{16} of current MMU slot

$D0A3_h$ 53411 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}	

$D0A4_h$ –$D0AE_h$ 53412 –53422 Reserved for future use

$D0AF_h$ 53423 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	Freezer/Game Type	00000000 = Off 00000001 = RetroReplay 00000011 = Final Cartridge 3 00000110 = Expert Cartridge 11111100 = 16K ROM cartridge at 8000_h–$BFFF_h$ 11111101 = 16K ROM cartridge in Ultimax mode 11111110 = 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
7–6	Reserved, must be 0	
5–4	Clock port	00 = Off 01 = Clock port at $DE00_h$-$DE0F_h$ 10 = Clock port at $DF20_h$-$DF2F_h$ 11 = reserved
3	Reserved, must be 0	
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 = Off 01 = MMC64 Emulation 10 = reserved 11 = 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 = Reserved 010 = NTSC (65 columns, 263 lines) 011 = Old-NTSC (64 columns, 262 lines) 1xx = 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	Turbo switch	0 = Manual setting 1 = Turbo mode is switchable with button
5	VIC-II turbo bit	0 = Off 1 = "Turbo Enable" is mirrored at bit 0 of $D030_h$
4–1	Reserved, must be 0	
0	Turbo speed limit	0 = maximum speed 1 = CPU limited to 2 Mhz

$D0F4_h$ 53492 CFGSID SID emulation

bit	settings	description
7	Reserved, must be 0	
6	SID type	0 = Emulate 6581 SID-Chip(s) 1 = Emulate 8580 SID-Chip(s) Not implemented in beta firmware, must be set to 0!
5–3	Stereo SID in C64	000 = Single SID in C64 001 = Second SID in C64 at $D420_h$ 010 = reserved 011 = reserved 100 = Second SID in C64 at $D500_h$ 101 = Second SID in C64 at $D700_h$ 110 = Second SID in C64 at $DE00_h$ 111 = 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.
2–0	SID emulation	000 = Emulate single SID 001 = Emulate stereo SID use A5 for selection ($D420_h$, $D460_h$, $D4A0_h$ ...) 110 = Emulate stereo SID use IO1 for second SID ($DE00_h$) 111 = 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	1 = Enable REU	Enable REU emulation and activate registers at $DF00_h$–$DF0A_h$.
6	1 = Enable geoRAM	Enable geoRAM emulation and activate registers at $DE00_h$–$DEFF_h$, $DFFE_h$and $DFFF_h$
5–3	geoRAM size	000 = 64 KByte 001 = 128 KByte 010 = 256 KByte 011 = 512 KByte 100 = 1 MByte 101 = 2 MByte 110 = 4 MByte 111 = reserved for future use
2–0	REU memory size	000 = 128 KByte 001 = 256 KByte 010 = 512 KByte 011 = 1 MByte 100 = 2 MByte 101 = 4 MByte 110 = 8 MByte 111 = 16 MByte (Note there is not enough RAM on C-One for this setting)

$D0F6_h$ 53494 CFG??? Reserved

$D0F7_h$ 53495 CFGDSK Disk images

bit	settings	description
7-6	Disk 9 floppy range	Number of floppy images for drive 9
5-4	Disk 9 floppy select	Select floppy image for drive 9
3-2	Disk 8 floppy range	Number of floppy images for drive 8
1-0	Disk 8 floppy select	Select floppy image for drive 8

$D0F8_h$ 53496 CFGFD0 Drive emulation

bit	settings	description
7–6	Enable virtual-drive CPU	00 = drive cpu stopped 01 = drive cpu running
5	Drive door	0 = Drive door closed 1 = Drive door open
4–3	Reserved, must be 0	–
2	Drive memory size	0 = 2 Kbyte (default) 1 = 8 Kbyte
1–0	Drive ID jumpers	00 = drive device id is 8 01 = drive device id is 9 10 = drive device id is 10 11 = drive device id is 11

$D0F9_h$ 53497 CFGFD1 Reserved for second drive

bit	settings	description
7–6	Enable virtual-drive CPU	00 = drive cpu stopped 01 = drive cpu running
5	Drive door	0 = Drive door closed 1 = Drive door open
4–3	Reserved, must be 0	–
2	Drive memory size	0 = 2 Kbyte (default) 1 = 8 Kbyte
1–0	Drive ID jumpers	00 = drive device id is 8 01 = drive device id is 9 10 = drive device id is 10 11 = drive device id is 11

$D0FA_h$ 53498 CFGREG Enable Chameleon registers

bit	settings	description
7	reserved, must be 0	–
6	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.
5	Chameleon banking ROM at $D700_h$	0 = No ROM 1 = ROM with banking and OS support routines is mapped at $D700_h$
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 Bank/MMU registers	0 = VIC-II chip mirrors at $D0A0_h$–$D0AF_h$ 1 = Chameleon Bank/MMU 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	Buttons configuration	

	Left		Middle		Right	
	short	long	short	long	short	long
0000	Menu	–	Freeze	–	Reset	Reboot from Flash
others	*** reserved ***					

$D0FC_h$ 53500 CFGCIA CIA 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 IEC bus and attached devices are held in reset
6-2	Reserved, must be 0	
1	CIA-2 emulation	0 = Old 1 = Type A This bit has no effect in cartridge mode
0	CIA-1 emulation	0 = Old 1 = Type A This bit has no effect in cartridge mode

$D0FD_h$ 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.
6–5	Reserved, always 0	
4	Flash slot valid	0 = Slot number is unknown 1 = Slot number is valid
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 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

$DE00_h$? ? 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)	

$DE01_h$? ? 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

$DE00_h$ –$DE01_h$? ? ? 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 or 256 KByte 1 = 512 KByte A single bit can't represent all 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 = Fix C64 address
6	REU Address control	0 = Increment REU address 1 = Fix 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)

GeoRAM

$DE00_h$ -$DEFF_h$? ? 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

$DE00_h$ –$DFFF_h$ 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