

# VERA module

Author: Frank van den Hoef.

***This is preliminary documentation and the specification can still change at any point.***

This document describes the **Video Enhanced Retro Adapter** video-module.

## 1. External address space

Reg	Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	\$9F20	VERA_ADDR_LO	Address (7:0)								
2	\$9F21	VERA_ADDR_MID	Address (15:8)								
1	\$9F22	VERA_ADDR_HI	Increment				Address (19:16)				
3	\$9F23	VERA_DATA0	Data port 0								
4	\$9F24	VERA_DATA1	Data port 1								
5	\$9F25	VERA_CTRL	RESET	-							ADDRSEL
6	\$9F26	VERA_IEN	-				UART	SPRCOL	LINE	VSYNC	
7	\$9F27	VERA_ISR	-				UART	SPRCOL	LINE	VSYNC	

When RESET is set to 1, the FPGA will reconfigure itself. All registers will be reset. The palette RAM will be set to its default values.

If ADDR\_SEL = 0, register 0/1/2 contain address of data port 0, otherwise register 0/1/2 contain address of data port 1.

After each access of one of the data ports the corresponding address is increment by the value associated with the corresponding increment field:

Increment value	Increment amount
0	0
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128
9	256
10	512
11	1024
12	2048
13	4096
14	8192
15	16384

Interrupts will be generated for the interrupt sources set in VERA\_IEN. VERA\_ISR will indicate interrupts that have occurred. Writing a 1 to a position in VERA\_ISR will clear that interrupt status.

## 2. Internal address space

Address range	Description
\$00000 - \$1FFFF	Video RAM
\$F0000 - \$F001F	Display composer registers
\$F1000 - \$F11FF	Palette
\$F2000 - \$F200F	Layer 0 registers
\$F3000 - \$F300F	Layer 1 registers
\$F4000 - \$F400F	Sprite registers
\$F5000 - \$F53FF	Sprite attributes
\$F6000 - \$F6xxx	Audio
\$F7000 - \$F7001	SPI
\$F8000 - \$F8003	UART

## 3. Display composer

Register	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	DC_VIDEO	CURRENT_FIELD (RO)	-				CHROMA_DISABLE	OUT_MODE		
1	DC_HSCALE	HSCALE								
2	DC_VSCALE	VSCALE								
3	DC_BORDER_COLOR	BORDER_COLOR								
4	DC_HSTART_L	HSTART (7:0)								
5	DC_HSTOP_L	HSTOP (7:0)								
6	DC_VSTART_L	VSTART (7:0)								
7	DC_VSTOP_L	VSTOP (7:0)								
8	DC_STARTSTOP_H	-	VSTOP (8)	VSTART (8)	HSTOP (9:8)		HSTART (9:8)			
9	DC_IRQ_LINE_L	IRQ_LINE (7:0)								
10	DC_IRQ_LINE_H	-							IRQ_LINE (8)	

OUT_MODE	Description
0	Video disabled
1	VGA output
2	NTSC composite
3	RGB interlaced, composite sync (via VGA output)

Setting **CHROMA\_DISABLE** disables output of chroma in NTSC composite mode and will give a better picture on a monochrome display.

**CURRENT\_FIELD** is a read-only field which reflects the active interlaced field in composite and RGB modes. (0: even, 1: odd)

**HSCALE** and **VSCALE** will set the fractional scaling factor of the display. Setting this value to 128 will output 1 output pixel for every input pixel. Setting this to 64 will output 2 output pixels for every input pixel.

**BORDER\_COLOR** determines the palette index which is used for the non-active area of the screen.

**HSTART/HSTOP** and **VSTART/VSTOP** determines the active part of the screen. The values here are specified in the native 640x480 display space. **HSTART=0**, **HSTOP=640**, **VSTART=0**, **VSTOP=480** will set the active area to the full resolution.

**IRQ\_LINE** specifies at which line the **LINE** interrupt will be generated. For interlaced modes the interrupt will be generated each field and the LSB of **IRQ\_LINE** is ignored.

TODO:

- Hardware ID
- Palette selection
- Per layer active area
- Per layer scaling
- Remapping transparent index 0 to other entry

## 4. Palette

The palette translate 8-bit color indexes into 12-bit output colors. The palette has 256 entries, each with the following format:

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Green				Blue			
1	-				Red			

At reset, the palette will contain a predefined palette:

Color indexes 0-15 contain the C64 color palette.

Color indexes 16-31 contain a grayscale ramp.

Color indexes 32-255 contain various hues, saturation levels, brightness levels.

## 5. Layer 0/1 registers

Register	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	Ln_CTRL0	MODE			-					EN
1	Ln_CTRL1	-		TILEH	TILEW	MAPH		MAPW		
2	Ln_MAP_BASE_L	MAP_BASE (9:2)								
3	Ln_MAP_BASE_H	MAP_BASE (17:10)								
4	Ln_TILE_BASE_L	TILE_BASE (9:2)								
5	Ln_TILE_BASE_H	TILE_BASE (17:10)								
6	Ln_HSCROLL_L	HSCROLL (7:0)								
7	Ln_HSCROLL_H	-				HSCROLL (11:8)				
8	Ln_VSCROLL_L	VSCROLL (7:0)								
9	Ln_VSCROLL_H	-				VSCROLL (11:8)				

In bitmap modes (5/6/7), the following changes apply:

Register	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7	Ln_BM_PAL_OFFS	-				BM_PALETTE_OFFSET			

The layer can be enabled or disabled by setting or clearing the **EN** bit.

**MAP\_BASE** specifies the base address where tile map data is fetched from. (Note that the registers don't specify the lower 2 bits, so the address is always aligned to a multiple of 4 bytes.)

**TILE\_BASE** specifies the base address where tile data is fetched from. (Note that the registers don't specify the lower 2 bits, so the address is always aligned to a multiple of 4 bytes.)

**HSCROLL** specifies the horizontal scroll offset. A value between 0 and 4095 can be used. Increasing the value will cause the picture to move left, decreasing will cause the picture to move right.

**YSCROLL** specifies the vertical scroll offset. A value between 0 and 4095 can be used. Increasing the value will cause the picture to move up, decreasing will cause the picture to move down.

**MAPW**, **MAPH** specify the map width and map height respectively:

Value	Map width / height
0	32 tiles
1	64 tiles
2	128 tiles
3	256 tiles

**TILEW, TILEH** specify the tile width and tile height respectively:

Value	Tile width / height
0	8
1	16

## 5.1. Layer display modes

Each layer supports a few different display modes, which can be selected using the **MODE** field:

Mode	Description
0	Tile mode 1bpp (per-tile 16 color foreground and background color)
1	Tile mode 1bpp (per-tile 256 color foreground color and fixed background color 0)
2	Tile mode 2bpp
3	Tile mode 4bpp
4	Tile mode 8bpp
5	Bitmap mode 2bpp
6	Bitmap mode 4bpp
7	Bitmap mode 8bpp

## 5.2. Mode 0 – 16 color text mode

**MAP\_BASE** points to a tile map containing tile map entries, which are 2 bytes each:

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Character index							
1	Background color				Foreground color			

**TILE\_BASE** points to the character data. This data is organized as 8 bytes per character entry. Each byte represents 1 line of character data, where bit 7 represents the left-most pixel and bit 0 the right-most pixel. If the bit is set the foreground color is used, otherwise the background color.

### 5.3. Mode 1 – 256 color text mode

**MAP\_BASE** points to a tile map containing tile map entries, which are 2 bytes each:

<b>Offset</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
<b>0</b>	<b>Character index</b>							
<b>1</b>	<b>Foreground color</b>							

**TILE\_BASE** points to the character data. This data is organized as 8 bytes per character entry. Each byte represents 1 line of character data, where bit 7 represents the left-most pixel and bit 0 the right-most pixel. If the bit is set the foreground color is used, otherwise color 0 is used.

#### 5.4. Mode 2/3/4 – Tile mode 2/4/8bpp

**MAP\_BASE** points to a tile map containing tile map entries, which are 2 bytes each:

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Tile index (7:0)							
1	Palette offset				V-flip	H-flip	Tile index (9:8)	

**TILE\_BASE** points to the tile data.

Each pixel in the tile data gives a color index of either 0-3 (2bpp), 0-15 (4bpp), 0-255 (8bpp). This color index is modified by the palette offset in the tile map data using the following logic:

- Color index 0 (transparent) and 16-255 are unmodified.
- Color index 1-15 is modified by adding 16 x palette offset.

TODO: explanation of tile data memory organization

#### 5.5. Mode 5/6/7 – Bitmap mode 2/4/8bpp

**MAP\_BASE** isn't used in these modes.

**TILE\_BASE** points to the bitmap data.

**TILEW** specifies the bitmap width. **TILEW**=0 results in 320 pixels width and **TILEW**=1 results in 640 pixels width.

**BM\_PALETTE\_OFFSET** modifies the color indexes of the bitmap in the same way as in the tile modes.

TODO: explanation of bitmap data memory organization

## 6. Sprite registers

Register	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SPR_CTRL								EN
1	SPR_COLLISION					Collision mask			

At the start of the vertical blank **Collision mask** is updated. This field indicates which groups of sprites have collided. If the field is non-zero the **SPRCOL** interrupt will be set. The interrupt is generated once per field / frame and can be cleared by making sure the sprites no longer collide.

Collisions are only detected on lines that are actually rendered.

## 7. Sprite attributes

128 entries of the following format:

Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Address (12:5)							
1	Mode	-			Address (16:13)			
2	X (7:0)							
3	-						X (9:8)	
4	Y (7:0)							
5	-						Y (9:8)	

6	Collision mask		Z-depth	V-flip	H-flip
7	Sprite height	Sprite width	Palette offset		

Mode	Description
0	4 bpp
1	8 bpp

Z-depth	Description
0	Sprite disabled
1	Sprite between background and layer 0
2	Sprite between layer 0 and layer 1
3	Sprite in front of layer 1

Sprite width / height	Description
0	8 px
1	16 px
2	32 px
3	64 px

**Palette offset** works in the same way as with the layers.

## 8. SPI registers

Register	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SPI_DATA	DATA							
1	SPI_CTRL	-						BUSY	SELECT

The SPI controller is connected to the SD card connector. The data rate is fixed at 12.5MHz.

Start a transfer by writing to SPI\_DATA. While the transfer is in progress the BUSY bit will be set. After the transfer is done, the result can be read from the SPI\_DATA register.

The chip select can be controlled by writing the SELECT bit. Writing 1 will assert the chip-select (logic-0) and writing 0 will release the chip-select (logic-1).

## 9. UART registers

Register	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	UART_DATA	DATA							
1	UART_STATUS	-						TXBUSY	RXFIFO not empty
2	UART_BAUDDIV_L	BAUDDIV (7:0)							
3	UART_BAUDDIV_H	BAUDDIV (15:8)							

The UART has a 512 bytes receive FIFO and a single byte transmit buffer. The BAUDDIV has to be set to select the desired baudrate. The resulting baudrate is  $25000000 / (\text{BAUDDIV} + 1)$ . The default baudrate at reset is 1Mbps.

Bit 0 in UART\_STATUS indicates whether DATA is available. If so, data can be read by reading from UART\_DATA. After this read bit 0 is updated.

To send a byte, first check if TXBUSY is clear, then write the data to UART\_DATA. While the UART is transmitting TXBUSY will be set.

An interrupt can be generated when data is available in the RX FIFO. Use the UART bit in the VERA\_IEN and VERA\_ISR registers for this.