

# **SuperVidel**

## **Memory Map**

Rev 1  
2012-03-12

## ***History***

<b>Date</b>	<b>Changed</b>
12-03-12	Initial release

## DDR2 RAM

Address	Size	Access types	Description
0xA0000000	16MB	-	Shadow of ST-RAM in DDR2 SDRAM
0xA1000000	112MB	Byte/ Word/ Long Re+Wr	DDR2 SDRAM general purpose graphics RAM

## Super Videl settings registers

Address	Size	Access types	Description																																																																																																																																																
0x80010000	Long	Long	<p><b>General purpose settings/status register</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Bit</th> <th colspan="8">31:24</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td colspan="8">Reserved</td> </tr> <tr> <td>Read/Write</td> <td colspan="8">R</td> </tr> <tr> <td>Reset value</td> <td colspan="8">%00000000</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Bit</th> <th>23</th> <th>22</th> <th>21</th> <th>20</th> <th>19</th> <th>18</th> <th>17</th> <th>16</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserv ed</td> <td>DVI VSYNC int flag</td> <td>DVI VSYNC int en</td> <td>VGA VSYNC int flag</td> <td>VGA VSYNC int en</td> <td>HW revision</td> <td>DVI protect</td> <td>VGA protect</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>?</td> <td>0</td> <td>0</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Bit</th> <th>15</th> <th>14</th> <th>13</th> <th>12</th> <th>11</th> <th>10</th> <th colspan="2">9:8</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>DVI clk is from clock chip</td> <td>DVI videl clk is 25.175 MHz</td> <td>VGA clk is from clock chip</td> <td>VGA videl clk is 25.175 MHz</td> <td>Videl 25.175 MHz valid</td> <td>Videl 32MHz valid</td> <td colspan="2">Monitor type</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> <td colspan="2">R+W</td> </tr> <tr> <td>Reset value</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td colspan="2">%10</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Bit</th> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>DVI connect ed</td> <td>SV mode</td> <td>DVI Hsync</td> <td>DVI Vsync</td> <td>VGA Hsync</td> <td>VGA Vsync</td> <td>DVI reset</td> <td>VGA reset</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> <td>R</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0</td> <td>1</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p><b>VGA reset:</b> Set to '1' to hold VGA video system logic in reset. No screen is output on the VGA output.</p>	Bit	31:24								Name	Reserved								Read/Write	R								Reset value	%00000000								Bit	23	22	21	20	19	18	17	16	Name	Reserv ed	DVI VSYNC int flag	DVI VSYNC int en	VGA VSYNC int flag	VGA VSYNC int en	HW revision	DVI protect	VGA protect	Read/Write	R	R+W	R+W	R+W	R+W	R	R+W	R+W	Reset value	0	0	0	0	0	?	0	0	Bit	15	14	13	12	11	10	9:8		Name	DVI clk is from clock chip	DVI videl clk is 25.175 MHz	VGA clk is from clock chip	VGA videl clk is 25.175 MHz	Videl 25.175 MHz valid	Videl 32MHz valid	Monitor type		Read/Write	R	R	R	R	R	R	R+W		Reset value	?	?	?	?	?	?	%10		Bit	7	6	5	4	3	2	1	0	Name	DVI connect ed	SV mode	DVI Hsync	DVI Vsync	VGA Hsync	VGA Vsync	DVI reset	VGA reset	Read/Write	R	R	R	R	R	R	R+W	R+W	Reset value	0	1	?	?	?	?	1	1
Bit	31:24																																																																																																																																																		
Name	Reserved																																																																																																																																																		
Read/Write	R																																																																																																																																																		
Reset value	%00000000																																																																																																																																																		
Bit	23	22	21	20	19	18	17	16																																																																																																																																											
Name	Reserv ed	DVI VSYNC int flag	DVI VSYNC int en	VGA VSYNC int flag	VGA VSYNC int en	HW revision	DVI protect	VGA protect																																																																																																																																											
Read/Write	R	R+W	R+W	R+W	R+W	R	R+W	R+W																																																																																																																																											
Reset value	0	0	0	0	0	?	0	0																																																																																																																																											
Bit	15	14	13	12	11	10	9:8																																																																																																																																												
Name	DVI clk is from clock chip	DVI videl clk is 25.175 MHz	VGA clk is from clock chip	VGA videl clk is 25.175 MHz	Videl 25.175 MHz valid	Videl 32MHz valid	Monitor type																																																																																																																																												
Read/Write	R	R	R	R	R	R	R+W																																																																																																																																												
Reset value	?	?	?	?	?	?	%10																																																																																																																																												
Bit	7	6	5	4	3	2	1	0																																																																																																																																											
Name	DVI connect ed	SV mode	DVI Hsync	DVI Vsync	VGA Hsync	VGA Vsync	DVI reset	VGA reset																																																																																																																																											
Read/Write	R	R	R	R	R	R	R+W	R+W																																																																																																																																											
Reset value	0	1	?	?	?	?	1	1																																																																																																																																											

**DVI reset:**

Set to '1' to hold DVI video system logic in reset. No screen is output on the DVI output.

**VGA Vsync active:**

Read-only status bit. Is '1' when a Vsync is in progress on VGA output.

**VGA Hsync active:**

Read-only status bit. Is '1' when a Hsync is in progress on VGA output.

**DVI Vsync active:**

Read-only status bit. Is '1' when a Vsync is in progress on DVI output.

**DVI Hsync active:**

Read-only status bit. Is '1' when a Hsync is in progress on DVI output.

**SV mode:**

Read-only status bit. Is '1' when the video system is currently in SuperVidel register mode, i.e. the SuperVidel registers were the last ones to be written, and the resolution etc is defined by the SuperVidel registers. If the Videll registers were the last ones to be written, the resolution output is defined by the Videll registers, and this bit is '0'. A copy of this bit can be found in the emulated Videll registers (**0xFFFF82C0.w bit 8**). When the user program reads out the Videll registers before changing resolution, this bit will be read too, and when the user program later restores the old Videll register settings, the status of this bit will be restored too. The SuperVidel will then know whether to restore the old SuperVidel resolution or the old Videll resolution.

**DVI connected:**

'1' if a powered-on DVI monitor is connected to the DVI port. '0' otherwise.

**Monitor type:**

Fake monitor type setting used only for the Videll compatibility. The actual connected monitor type is NOT detected by the hardware. The monitor type only applies to the analog (VGA) output. Values:

Monitor type	Value
Monochrome	"00"
RGB - Colormonitor	"01"
VGA - Colormonitor	"10"
TV	"11"

**Videll 32MHz valid:**

Says if the input 32 Mhz clock from the falcon motherboard (or from CT60) is detected as running. But not if it is the correct frequency.

**Videll 25.175MHz valid:**

Says if the input 25.175 Mhz clock from the falcon motherboard is detected as running. But not if it is the correct frequency.

**VGA Videll clk is 25.175MHz:**

Says if the Videll 25.175MHz(1) clock is chosen instead of the 32MHz(0) for VGA clock selection MUX.

**VGA clk is from clock chip:**

The VGA clock net is driven by the CDCE925 clock chip(1), not a videll 25.175 or 32MHz clock (0).

**DVI Videll clk is 25.175MHz:**

Says if the Videll 25.175MHz(1) clock is chosen instead of the 32MHz(0) for DVI clock selection MUX.

**DVI clk is from clock chip:**

The DVI clock net is driven by the CDCE925 clock chip(1), not a videll 25.175 or 32MHz clock (0).

			<p><b>VGA protect:</b> If 1 it will protect the VGA output from switching to Videl register settings when Videl registers are written. On reset this is 0, so the Videl resolution will appear on VGA.</p> <p><b>DVI protect:</b> If 1 it will protect the DVI output from switching to Videl register settings when Videl registers are written. On reset this is 0, so the Videl resolution will appear on DVI.</p> <p><b>HW revision:</b> Is 1 if the SV hardware is a revision 1. Is 0 for a revision 2.</p> <p><b>VG/DVI VBL INT enable &amp; Done INT flags</b> Writing VGA or DVI INT enable to 1 makes the vertical sync for VGA or DVI generate an interrupt request, and VGA or DVI Done INT flag will be set to 1. You must clear it in your handler by writing 0 to the right Done INT flag. Make sure you put your ISR pointer in vector number 0xC7 (address 0x31C) (for VGA) or 0xC8 (address 0x320) (for DVI) before enabling either interrupt type.</p>																				
0x80010004	Long	Long Wr only	<p><b>CDCE925 PLL1-reg1 (DDR clock)</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:24</th> <th>23:16</th> <th>15:8</th> <th>7:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reg0x17</td> <td>Reg0x18</td> <td>Reg0x19</td> <td>Reg0x1A</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0x01</td> <td>0x06</td> <td>0x90</td> <td>0x13</td> </tr> </tbody> </table> <p>Use addr 0x17-0x1A (0x17 is MSB) output from CDCE925 ClockFinder program. Write this register first when setting PLL1.</p>	Bit	31:24	23:16	15:8	7:0	Name	Reg0x17	Reg0x18	Reg0x19	Reg0x1A	Read/Write	R+W	R+W	R+W	R+W	Reset value	0x01	0x06	0x90	0x13
Bit	31:24	23:16	15:8	7:0																			
Name	Reg0x17	Reg0x18	Reg0x19	Reg0x1A																			
Read/Write	R+W	R+W	R+W	R+W																			
Reset value	0x01	0x06	0x90	0x13																			
0x80010008	Long	Long Wr only	<p><b>CDCE925 PLL1-reg2 (DDR clock)</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:8</th> <th>7:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Reg0x1B</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0x000000</td> <td>0x47</td> </tr> </tbody> </table> <p>Use addr 0x1B (in LSB) output from the CDCE925 Clockfinder program. Write this register last when setting PLL1, to begin update automatically.</p>	Bit	31:8	7:0	Name	Reserved	Reg0x1B	Read/Write	R	R+W	Reset value	0x000000	0x47								
Bit	31:8	7:0																					
Name	Reserved	Reg0x1B																					
Read/Write	R	R+W																					
Reset value	0x000000	0x47																					
0x8001000C	Long	Long Wr only	<p><b>CDCE925 PLL2-reg1 (Video clock)</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:24</th> <th>23:16</th> <th>15:8</th> <th>7:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reg0x27</td> <td>Reg0x28</td> <td>Reg0x29</td> <td>Reg0x2A</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0x00</td> <td>0x00</td> <td>0x00</td> <td>0x00</td> </tr> </tbody> </table> <p>Use addr 0x27-0x2A (0x27 is MSB) output from CDCE925 ClockFinder program. Write this register first when setting PLL2.</p>	Bit	31:24	23:16	15:8	7:0	Name	Reg0x27	Reg0x28	Reg0x29	Reg0x2A	Read/Write	R+W	R+W	R+W	R+W	Reset value	0x00	0x00	0x00	0x00
Bit	31:24	23:16	15:8	7:0																			
Name	Reg0x27	Reg0x28	Reg0x29	Reg0x2A																			
Read/Write	R+W	R+W	R+W	R+W																			
Reset value	0x00	0x00	0x00	0x00																			
0x80010010	Long	Long Wr only	<p><b>CDCE925 PLL2-reg2 (Video clock)</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:8</th> <th>7:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Reg0x2B</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0x000000</td> <td>0x00</td> </tr> </tbody> </table>	Bit	31:8	7:0	Name	Reserved	Reg0x2B	Read/Write	R	R+W	Reset value	0x000000	0x00								
Bit	31:8	7:0																					
Name	Reserved	Reg0x2B																					
Read/Write	R	R+W																					
Reset value	0x000000	0x00																					

			Use addr 0x2B (in LSB) output from the CDCE925 Clockfinder program. Write this register last when setting PLL2, to begin update automatically.																								
0x80010014	Long	Long Re+W	<p><b>VGA Screen address</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Video address</td> <td>0</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b000000000000000000000000</td> <td>0</td> </tr> </tbody> </table> <p>Screen address is 27 bits, but bit 0 is always 0, so an odd address is not possible.</p>	Bit	31:27	26:1	0	Name	Reserved	Video address	0	Read/Write	R	R+W	R	Reset value	0b00000	0b000000000000000000000000	0								
Bit	31:27	26:1	0																								
Name	Reserved	Video address	0																								
Read/Write	R	R+W	R																								
Reset value	0b00000	0b000000000000000000000000	0																								
0x80010018	Long	Long Re+W	<p><b>VGA Horizontal data begin/end register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Hor. data begin</td> <td>Reserved</td> <td>Hor. data end</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b00000000000</td> <td>0b00000</td> <td>0b00000000000</td> </tr> </tbody> </table> <p><b>Horizontal data begin:</b> Horizontal position on a scanline where the data window begins. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p> <p><b>Horizontal data end:</b> Horizontal position on a scanline where the data window ends. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p>	Bit	31:27	26:16	15:11	10:0	Name	Reserved	Hor. data begin	Reserved	Hor. data end	Read/Write	R	R+W	R	R+W	Reset value	0b00000	0b00000000000	0b00000	0b00000000000				
Bit	31:27	26:16	15:11	10:0																							
Name	Reserved	Hor. data begin	Reserved	Hor. data end																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00000	0b00000000000	0b00000	0b00000000000																							
0x8001001C	Long	Long Re+W	<p><b>VGA Horizontal sync register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31</th> <th>30:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Polarity</td> <td>Reserved</td> <td>Hor. sync begin</td> <td>Reserved</td> <td>Hor. sync end</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>1</td> <td>0b0000</td> <td>0b00000000000</td> <td>0b00000</td> <td>0b00000000000</td> </tr> </tbody> </table> <p><b>Polarity:</b> Select sync polarity. '1' means sync pulse is active high.</p> <p><b>Horizontal sync begin:</b> Horizontal position on a scanline where the hsync pulse begins. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p> <p><b>Horizontal sync end:</b> Horizontal position on a scanline where the hsync pulse ends. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p>	Bit	31	30:27	26:16	15:11	10:0	Name	Polarity	Reserved	Hor. sync begin	Reserved	Hor. sync end	Read/Write	R+W	R	R+W	R	R+W	Reset value	1	0b0000	0b00000000000	0b00000	0b00000000000
Bit	31	30:27	26:16	15:11	10:0																						
Name	Polarity	Reserved	Hor. sync begin	Reserved	Hor. sync end																						
Read/Write	R+W	R	R+W	R	R+W																						
Reset value	1	0b0000	0b00000000000	0b00000	0b00000000000																						
0x80010020	Long	Long Re+W	<p><b>VGA Vertical data begin/end register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Vert data begin</td> <td>Reserved</td> <td>Vert data end</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b00000000000</td> <td>0b00000</td> <td>0b00000000000</td> </tr> </tbody> </table> <p><b>Vertical data begin:</b> Vertical position (in scanlines) where the data window begins.</p> <p><b>Vertical data end:</b> Vertical position (in scanlines) where the data window ends.</p>	Bit	31:27	26:16	15:11	10:0	Name	Reserved	Vert data begin	Reserved	Vert data end	Read/Write	R	R+W	R	R+W	Reset value	0b00000	0b00000000000	0b00000	0b00000000000				
Bit	31:27	26:16	15:11	10:0																							
Name	Reserved	Vert data begin	Reserved	Vert data end																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00000	0b00000000000	0b00000	0b00000000000																							

0x80010024	Long	Long Re+W	<p><b>VGA Vertical sync register</b></p> <table border="1" data-bbox="635 232 1437 414"> <thead> <tr> <th>Bit</th> <th>31</th> <th>30:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Polarity</td> <td>Reserved</td> <td>Vert. sync begin</td> <td>Reserved</td> <td>Vert. sync end</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>1</td> <td>0b0000</td> <td>0b000000000000</td> <td>0b000000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Polarity:</b> Select sync polarity. '1' means sync pulse is active high.</p> <p><b>Vertical sync begin:</b> Vertical position (in scanlines) where the vsync pulse begins.</p> <p><b>Vertical sync end:</b> Vertical position (in scanlines) where the vsync pulse ends.</p>	Bit	31	30:27	26:16	15:11	10:0	Name	Polarity	Reserved	Vert. sync begin	Reserved	Vert. sync end	Read/Write	R+W	R	R+W	R	R+W	Reset value	1	0b0000	0b000000000000	0b000000	0b000000000000
Bit	31	30:27	26:16	15:11	10:0																						
Name	Polarity	Reserved	Vert. sync begin	Reserved	Vert. sync end																						
Read/Write	R+W	R	R+W	R	R+W																						
Reset value	1	0b0000	0b000000000000	0b000000	0b000000000000																						
0x80010028	Long	Long Re+W	<p><b>VGA Last horizontal/vertical position register</b></p> <table border="1" data-bbox="635 719 1437 878"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Last hor. pos.</td> <td>Reserved</td> <td>Last vertical pos.</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b000000</td> <td>0b000000000000</td> <td>0b000000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Last horizontal position:</b> The last horizontal position (in pixels), i.e. the last value that the horizontal counter reaches before returning to 0. <b>Divide the value by 2 after calculation.</b></p> <p><b>Last vertical position:</b> The last vertical position (in lines), i.e. the last value that the vertical counter reaches before returning to 0.</p>	Bit	31:27	26:16	15:11	10:0	Name	Reserved	Last hor. pos.	Reserved	Last vertical pos.	Read/Write	R	R+W	R	R+W	Reset value	0b000000	0b000000000000	0b000000	0b000000000000				
Bit	31:27	26:16	15:11	10:0																							
Name	Reserved	Last hor. pos.	Reserved	Last vertical pos.																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b000000	0b000000000000	0b000000	0b000000000000																							
0x8001002C	Long	Long Re+W	<p><b>VGA Data amount register</b></p> <table border="1" data-bbox="635 1211 1437 1370"> <thead> <tr> <th>Bit</th> <th>31:30</th> <th>29:16</th> <th>15:14</th> <th>13:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Virtual screen data</td> <td>Reserved</td> <td>Visible screen data</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00</td> <td>0b0000000000000000</td> <td>0b00</td> <td>0b0000000000000000</td> </tr> </tbody> </table> <p><b>Virtual screen data:</b> The number of bytes on a virtual screen row (the visible data + data to be skipped on each row). Note that bit 0 (at position 16 in the longword register) is always 0, so only even bytes are possible, i.e. whole 16-bit words.</p> <p><b>Visible screen data:</b> The number of bytes on a visible screen row. Note that bits 3:0 are always 0, so only even 16 bytes are possible. Always round up to the nearest 16 bytes, since the extra read data will be discarded.</p>	Bit	31:30	29:16	15:14	13:0	Name	Reserved	Virtual screen data	Reserved	Visible screen data	Read/Write	R	R+W	R	R+W	Reset value	0b00	0b0000000000000000	0b00	0b0000000000000000				
Bit	31:30	29:16	15:14	13:0																							
Name	Reserved	Virtual screen data	Reserved	Visible screen data																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00	0b0000000000000000	0b00	0b0000000000000000																							
0x80010030	Long	Long Re+W	<p><b>VGA Video mode settings</b></p> <table border="1" data-bbox="635 1771 1437 2002"> <thead> <tr> <th>Bit</th> <th>31:6</th> <th>5</th> <th>4</th> <th>3</th> <th>2:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Line doubling</td> <td>Double pixel width</td> <td>ST comp.</td> <td>Video mode</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b000000000000000000000000</td> <td>0b0</td> <td>0b0</td> <td>0b0</td> <td>0b000</td> </tr> </tbody> </table>	Bit	31:6	5	4	3	2:0	Name	Reserved	Line doubling	Double pixel width	ST comp.	Video mode	Read/Write	R	R+W	R+W	R+W	R+W	Reset value	0b000000000000000000000000	0b0	0b0	0b0	0b000
Bit	31:6	5	4	3	2:0																						
Name	Reserved	Line doubling	Double pixel width	ST comp.	Video mode																						
Read/Write	R	R+W	R+W	R+W	R+W																						
Reset value	0b000000000000000000000000	0b0	0b0	0b0	0b000																						

			<table border="1"> <thead> <tr> <th>Bits 2:0</th> <th>Video mode</th> </tr> </thead> <tbody> <tr> <td>0b000</td> <td>1-bit bitplane (Falcon030 compatible)</td> </tr> <tr> <td>0b001</td> <td>2-bit bitplane (Falcon030 compatible)</td> </tr> <tr> <td>0b010</td> <td>4-bit bitplane (Falcon030 compatible)</td> </tr> <tr> <td>0b011</td> <td>8-bit bitplane (Falcon030 compatible)</td> </tr> <tr> <td>0b100</td> <td>8-bit chunky</td> </tr> <tr> <td>0b101</td> <td>16-bit highcolor (Falcon030 compatible)</td> </tr> <tr> <td>0b110</td> <td>reserved</td> </tr> <tr> <td>0b111</td> <td>32-bit truecolor (24-bit RGB + alpha byte)</td> </tr> </tbody> </table> <p><b>ST compatibility mode:</b> Set to '1' to turn on ST compatibility mode. (No effect right now).</p> <p><b>Double pixel width:</b> Set to '1' to double the pixel width, i.e. halve the amount of pixels horizontally. This is needed to get a 320 wide resolution with VGA timing (640x480).</p> <p><b>Line doubling:</b> Set to '1' to make each visible line be drawn twice, i.e. halve the visible vertical resolution. This is needed to get a 240 high resolution with VGA timing (640x480).</p>	Bits 2:0	Video mode	0b000	1-bit bitplane (Falcon030 compatible)	0b001	2-bit bitplane (Falcon030 compatible)	0b010	4-bit bitplane (Falcon030 compatible)	0b011	8-bit bitplane (Falcon030 compatible)	0b100	8-bit chunky	0b101	16-bit highcolor (Falcon030 compatible)	0b110	reserved	0b111	32-bit truecolor (24-bit RGB + alpha byte)						
Bits 2:0	Video mode																										
0b000	1-bit bitplane (Falcon030 compatible)																										
0b001	2-bit bitplane (Falcon030 compatible)																										
0b010	4-bit bitplane (Falcon030 compatible)																										
0b011	8-bit bitplane (Falcon030 compatible)																										
0b100	8-bit chunky																										
0b101	16-bit highcolor (Falcon030 compatible)																										
0b110	reserved																										
0b111	32-bit truecolor (24-bit RGB + alpha byte)																										
0x80010034	Long	Long Re+W	<p><b>DVI Screen address</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Video address</td> <td>0</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b000000000000000000000000</td> <td>0</td> </tr> </tbody> </table> <p>Screen address is 27 bits, but bit 0 is always 0, so an odd address is not possible.</p>	Bit	31:27	26:1	0	Name	Reserved	Video address	0	Read/Write	R	R+W	R	Reset value	0b00000	0b000000000000000000000000	0								
Bit	31:27	26:1	0																								
Name	Reserved	Video address	0																								
Read/Write	R	R+W	R																								
Reset value	0b00000	0b000000000000000000000000	0																								
0x80010038	Long	Long Re+W	<p><b>DVI Horizontal data begin/end register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Hor. data begin</td> <td>Reserved</td> <td>Hor. data end</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b000000000000</td> <td>0b00000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Horizontal data begin:</b> Horizontal position on a scanline where the data window begins. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p> <p><b>Horizontal data end:</b> Horizontal position on a scanline where the data window ends. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p>	Bit	31:27	26:16	15:11	10:0	Name	Reserved	Hor. data begin	Reserved	Hor. data end	Read/Write	R	R+W	R	R+W	Reset value	0b00000	0b000000000000	0b00000	0b000000000000				
Bit	31:27	26:16	15:11	10:0																							
Name	Reserved	Hor. data begin	Reserved	Hor. data end																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00000	0b000000000000	0b00000	0b000000000000																							
0x8001003C	Long	Long Re+W	<p><b>DVI Horizontal sync register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31</th> <th>30:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Polarity</td> <td>Reserved</td> <td>Hor. sync begin</td> <td>Reserved</td> <td>Hor. sync end</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>1</td> <td>0b0000</td> <td>0b000000000000</td> <td>0b00000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Polarity:</b></p>	Bit	31	30:27	26:16	15:11	10:0	Name	Polarity	Reserved	Hor. sync begin	Reserved	Hor. sync end	Read/Write	R+W	R	R+W	R	R+W	Reset value	1	0b0000	0b000000000000	0b00000	0b000000000000
Bit	31	30:27	26:16	15:11	10:0																						
Name	Polarity	Reserved	Hor. sync begin	Reserved	Hor. sync end																						
Read/Write	R+W	R	R+W	R	R+W																						
Reset value	1	0b0000	0b000000000000	0b00000	0b000000000000																						



			<p>Select sync polarity. '1' means sync pulse is active high.</p> <p><b>Horizontal sync begin:</b> Horizontal position on a scanline where the hsync pulse begins. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p> <p><b>Horizontal sync end:</b> Horizontal position on a scanline where the hsync pulse ends. Counted in video clock pulses (pixels). <b>Divide the value by 2 after calculation.</b></p>																								
0x80010040	Long	Long Re+W	<p><b>DVI Vertical data begin/end register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Vert data begin</td> <td>Reserved</td> <td>Vert data end</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b000000000000</td> <td>0b00000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Vertical data begin:</b> Vertical position (in scanlines) where the data window begins.</p> <p><b>Vertical data end:</b> Vertical position (in scanlines) where the data window ends.</p>	Bit	31:27	26:16	15:11	10:0	Name	Reserved	Vert data begin	Reserved	Vert data end	Read/Write	R	R+W	R	R+W	Reset value	0b00000	0b000000000000	0b00000	0b000000000000				
Bit	31:27	26:16	15:11	10:0																							
Name	Reserved	Vert data begin	Reserved	Vert data end																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00000	0b000000000000	0b00000	0b000000000000																							
0x80010044	Long	Long Re+W	<p><b>DVI Vertical sync register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31</th> <th>30:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Polarity</td> <td>Reserved</td> <td>Hor. sync begin</td> <td>Reserved</td> <td>Hor. sync end</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>1</td> <td>0b0000</td> <td>0b000000000000</td> <td>0b00000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Polarity:</b> Select sync polarity. '1' means sync pulse is active high.</p> <p><b>Vertical sync begin:</b> Vertical position (in scanlines) where the vsync pulse begins.</p> <p><b>Vertical sync end:</b> Vertical position (in scanlines) where the vsync pulse ends.</p>	Bit	31	30:27	26:16	15:11	10:0	Name	Polarity	Reserved	Hor. sync begin	Reserved	Hor. sync end	Read/Write	R+W	R	R+W	R	R+W	Reset value	1	0b0000	0b000000000000	0b00000	0b000000000000
Bit	31	30:27	26:16	15:11	10:0																						
Name	Polarity	Reserved	Hor. sync begin	Reserved	Hor. sync end																						
Read/Write	R+W	R	R+W	R	R+W																						
Reset value	1	0b0000	0b000000000000	0b00000	0b000000000000																						
0x80010048	Long	Long Re+W	<p><b>DVI Last horizontal/vertical position register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:16</th> <th>15:11</th> <th>10:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Last hor. pos.</td> <td>Reserved</td> <td>Last vertical pos.</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b000000000000</td> <td>0b00000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Last horizontal position:</b> The last horizontal position (in pixels), i.e. the last value that the horizontal counter reaches before returning to 0. <b>Divide the value by 2 after calculation.</b></p> <p><b>Last vertical position:</b> The last vertical position (in lines), i.e. the last value that the vertical counter reaches before returning to 0.</p>	Bit	31:27	26:16	15:11	10:0	Name	Reserved	Last hor. pos.	Reserved	Last vertical pos.	Read/Write	R	R+W	R	R+W	Reset value	0b00000	0b000000000000	0b00000	0b000000000000				
Bit	31:27	26:16	15:11	10:0																							
Name	Reserved	Last hor. pos.	Reserved	Last vertical pos.																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00000	0b000000000000	0b00000	0b000000000000																							
0x8001004C	Long	Long Re+W	<p><b>DVI Data amount register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:30</th> <th>29:16</th> <th>15:14</th> <th>13:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Virtual screen data</td> <td>Reserved</td> <td>Visible screen data</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00</td> <td>0b00000000000000</td> <td>0b00</td> <td>0b00000000000000</td> </tr> </tbody> </table>	Bit	31:30	29:16	15:14	13:0	Name	Reserved	Virtual screen data	Reserved	Visible screen data	Read/Write	R	R+W	R	R+W	Reset value	0b00	0b00000000000000	0b00	0b00000000000000				
Bit	31:30	29:16	15:14	13:0																							
Name	Reserved	Virtual screen data	Reserved	Visible screen data																							
Read/Write	R	R+W	R	R+W																							
Reset value	0b00	0b00000000000000	0b00	0b00000000000000																							

			<p><b>Virtual screen data:</b> The number of bytes on a virtual screen row (the visible data + data to be skipped on each row). Note that bit 0 (at position 16 in the longword register) is always 0, so only even bytes are possible, i.e. whole 16-bit words.</p> <p><b>Visible screen data:</b> The number of bytes on a visible screen row. Note that bits 3:0 are always 0, so only even 16 bytes are possible. Always round up to the nearest 16 bytes, since the extra read data will be discarded.</p>																																										
0x80010050	Long	Long Re+W	<p><b>DVI Video mode settings</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:6</th> <th>5</th> <th>4</th> <th>3</th> <th>2:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Line doubling</td> <td>Double pixel width</td> <td>ST comp.</td> <td>Video mode</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b0000000000000000 0000000000</td> <td>0b0</td> <td>0b0</td> <td>0b0</td> <td>0b000</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Bits 2:0</th> <th>Video mode</th> </tr> </thead> <tbody> <tr> <td>0b000</td> <td>1-bit bitplane Falcon030 mode</td> </tr> <tr> <td>0b001</td> <td>2-bit bitplane Falcon030 mode</td> </tr> <tr> <td>0b010</td> <td>4-bit bitplane Falcon030 mode</td> </tr> <tr> <td>0b011</td> <td>8-bit bitplane Falcon030 mode</td> </tr> <tr> <td>0b100</td> <td>8-bit chunky</td> </tr> <tr> <td>0b101</td> <td>16-bit highcolor Falcon030 mode</td> </tr> <tr> <td>0b110</td> <td>reserved</td> </tr> <tr> <td>0b111</td> <td>32-bit truecolor (24-bit RGB + alpha byte)</td> </tr> </tbody> </table> <p><b>ST compatibility mode:</b> Set to '1' to turn on ST compatibility mode. (No effect right now).</p> <p><b>Double pixel width:</b> Set to '1' to double the pixel width, i.e. halve the amount of pixels horizontally. This is needed to get a 320 wide resolution with VGA timing (640x480).</p> <p><b>Line doubling:</b> Set to '1' to make each visible line be drawn twice, i.e. halve the visible vertical resolution. This is needed to get a 240 high resolution with VGA timing (640x480).</p>	Bit	31:6	5	4	3	2:0	Name	Reserved	Line doubling	Double pixel width	ST comp.	Video mode	Read/Write	R	R+W	R+W	R+W	R+W	Reset value	0b0000000000000000 0000000000	0b0	0b0	0b0	0b000	Bits 2:0	Video mode	0b000	1-bit bitplane Falcon030 mode	0b001	2-bit bitplane Falcon030 mode	0b010	4-bit bitplane Falcon030 mode	0b011	8-bit bitplane Falcon030 mode	0b100	8-bit chunky	0b101	16-bit highcolor Falcon030 mode	0b110	reserved	0b111	32-bit truecolor (24-bit RGB + alpha byte)
Bit	31:6	5	4	3	2:0																																								
Name	Reserved	Line doubling	Double pixel width	ST comp.	Video mode																																								
Read/Write	R	R+W	R+W	R+W	R+W																																								
Reset value	0b0000000000000000 0000000000	0b0	0b0	0b0	0b000																																								
Bits 2:0	Video mode																																												
0b000	1-bit bitplane Falcon030 mode																																												
0b001	2-bit bitplane Falcon030 mode																																												
0b010	4-bit bitplane Falcon030 mode																																												
0b011	8-bit bitplane Falcon030 mode																																												
0b100	8-bit chunky																																												
0b101	16-bit highcolor Falcon030 mode																																												
0b110	reserved																																												
0b111	32-bit truecolor (24-bit RGB + alpha byte)																																												
0x80010054	Long	Long Re+W	<p><b>JTAG control register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31</th> <th>30:16</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>JTAG enable</td> <td>Magic key</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>W</td> </tr> <tr> <td>Reset value</td> <td>0</td> <td>%0000000000000000</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Bit</th> <th>15:4</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> </tr> <tr> <td>Read/Write</td> <td>R</td> </tr> </tbody> </table>	Bit	31	30:16	Name	JTAG enable	Magic key	Read/Write	R+W	W	Reset value	0	%0000000000000000	Bit	15:4	Name	Reserved	Read/Write	R																								
Bit	31	30:16																																											
Name	JTAG enable	Magic key																																											
Read/Write	R+W	W																																											
Reset value	0	%0000000000000000																																											
Bit	15:4																																												
Name	Reserved																																												
Read/Write	R																																												

			<table border="1"> <tr> <td><b>Reset value</b></td> <td colspan="4">%000000000000</td> </tr> <tr> <td><b>Bit</b></td> <td><b>3</b></td> <td><b>2</b></td> <td><b>1</b></td> <td><b>0</b></td> </tr> <tr> <td><b>Name</b></td> <td>JTAG TCK</td> <td>JTAG TMS</td> <td>JTAG TDI</td> <td>JTAG TDO</td> </tr> <tr> <td><b>Read/Write</b></td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R</td> </tr> <tr> <td><b>Reset value</b></td> <td>0</td> <td>1</td> <td>1</td> <td>?</td> </tr> </table> <p><b>Magic key:</b> Must equal %1010101010101 to allow any changes to the JTAG control bits below, i.e. every write to this register that wishes to modify the JTAG bits must have this value at bits 30-16.</p> <p><b>JTAG enable:</b> 0=JTAG interface disabled and JTAG pins in high impedance. 1=JTAG enabled and FPGA is driving JTAG pins towards PROM memories.</p> <p><b>JTAG TDO:</b> When read, returns the level of the JTAG-TDO signal going from the PROMs to the FPGA.</p> <p><b>JTAG TDI:</b> When written, controls the output level of the JTAG-TDI signal going from the FPGA to the PROMs. When read, returns the level of the internal JTAG TDI signal in the FPGA and not the level of the external signal between the FPGA and the PROMs. So this cannot be used to detect a possible external JTAG master trying to drive the signals.</p> <p><b>JTAG TMS:</b> When written, controls the output level of the JTAG-TMS signal going from the FPGA to the PROMs. When read, returns the level of the internal JTAG TMS signal in the FPGA and not the level of the external signal between the FPGA and the PROMs. So this cannot be used to detect a possible external JTAG master trying to drive the signals.</p> <p><b>JTAG TCK:</b> When written, controls the output level of the JTAG-TCK signal going from the FPGA to the PROMs. When read, returns the level of the internal JTAG TCK signal in the FPGA and not the level of the external signal between the FPGA and the PROMs. So this cannot be used to detect a possible external JTAG master trying to drive the signals.</p>	<b>Reset value</b>	%000000000000				<b>Bit</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>	<b>Name</b>	JTAG TCK	JTAG TMS	JTAG TDI	JTAG TDO	<b>Read/Write</b>	R+W	R+W	R+W	R	<b>Reset value</b>	0	1	1	?
<b>Reset value</b>	%000000000000																											
<b>Bit</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>																								
<b>Name</b>	JTAG TCK	JTAG TMS	JTAG TDI	JTAG TDO																								
<b>Read/Write</b>	R+W	R+W	R+W	R																								
<b>Reset value</b>	0	1	1	?																								
0x80010058	Long	Long Re+W	<p><b>SuperBlitter source1 address register</b></p> <table border="1"> <tr> <td><b>Bit</b></td> <td><b>31:27</b></td> <td><b>26:0</b></td> </tr> <tr> <td><b>Name</b></td> <td>Reserved</td> <td>Source address 1</td> </tr> <tr> <td><b>Read/Write</b></td> <td>R</td> <td>R+W</td> </tr> <tr> <td><b>Reset value</b></td> <td>0b00000</td> <td>0b00000000000000000000000000000000</td> </tr> </table> <p>The address of the main data source, addressing bytes.</p>	<b>Bit</b>	<b>31:27</b>	<b>26:0</b>	<b>Name</b>	Reserved	Source address 1	<b>Read/Write</b>	R	R+W	<b>Reset value</b>	0b00000	0b00000000000000000000000000000000													
<b>Bit</b>	<b>31:27</b>	<b>26:0</b>																										
<b>Name</b>	Reserved	Source address 1																										
<b>Read/Write</b>	R	R+W																										
<b>Reset value</b>	0b00000	0b00000000000000000000000000000000																										
0x8001005C	Long	Long Re+W	<p><b>SuperBlitter source2 address register</b></p> <table border="1"> <tr> <td><b>Bit</b></td> <td><b>31:27</b></td> <td><b>26:0</b></td> </tr> <tr> <td><b>Name</b></td> <td>Reserved</td> <td>Source address 2</td> </tr> <tr> <td><b>Read/Write</b></td> <td>R</td> <td>R+W</td> </tr> </table>	<b>Bit</b>	<b>31:27</b>	<b>26:0</b>	<b>Name</b>	Reserved	Source address 2	<b>Read/Write</b>	R	R+W																
<b>Bit</b>	<b>31:27</b>	<b>26:0</b>																										
<b>Name</b>	Reserved	Source address 2																										
<b>Read/Write</b>	R	R+W																										



			<p>The amount of bytes that are to be added to the line start address after a line has been copied, in source2, in order to reach the next line start address. Negative amounts (address decrements) can be realised by using numbers starting with '1' here (in bit 26).</p>																								
0x80010070	Long	Long Re+W	<p><b>SuperBlitter destination line offset register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:27</th> <th>26:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Line offset</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000</td> <td>0b00000000000000000000000000000000</td> </tr> </tbody> </table> <p>The amount of bytes that are to be added to the line start address after a line has been copied, in destination, in order to reach the next line start address. Negative amounts (address decrements) can be realised by using numbers starting with '1' here (in bit 26).</p>	Bit	31:27	26:0	Name	Reserved	Line offset	Read/Write	R	R+W	Reset value	0b00000	0b00000000000000000000000000000000												
Bit	31:27	26:0																									
Name	Reserved	Line offset																									
Read/Write	R	R+W																									
Reset value	0b00000	0b00000000000000000000000000000000																									
0x80010074	Long	Long Re+W	<p><b>SuperBlitter address mask and number of lines register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:24</th> <th>23:16</th> <th>15:12</th> <th>11:0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Src1 address count modulo</td> <td>Src2 address count modulo</td> <td>Reserved</td> <td>Number of lines</td> </tr> <tr> <td>Read/Write</td> <td>R+W</td> <td>R+W</td> <td>R</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000000</td> <td>0b00000000</td> <td>0b0000</td> <td>0b000000000000</td> </tr> </tbody> </table> <p><b>Src1 address count mask:</b> A mask to apply to the source1 address counter while incrementing it during the write phase. The value set will make the address wrap at that value. Only 2<sup>n</sup> values are possible: 0,1,2,4,8...128. Note that we're counting 64-bit words here, not bytes, since the blitter works with 64-bit words internally. Also note that it is only the counter, that is added to the src1 base address, that is affected by this mask, not the src1 base address itself. Setting the mask to 0 will turn off masking of the address counter (but it can only count to 255 anyway).</p> <p><b>Src2 address count mask:</b> The same as the src1 address mask above, but for src2.</p> <p><b>Number of lines:</b> The amount of horizontal lines to do.</p>	Bit	31:24	23:16	15:12	11:0	Name	Src1 address count modulo	Src2 address count modulo	Reserved	Number of lines	Read/Write	R+W	R+W	R	R+W	Reset value	0b00000000	0b00000000	0b0000	0b000000000000				
Bit	31:24	23:16	15:12	11:0																							
Name	Src1 address count modulo	Src2 address count modulo	Reserved	Number of lines																							
Read/Write	R+W	R+W	R	R+W																							
Reset value	0b00000000	0b00000000	0b0000	0b000000000000																							
0x80010078	Long	Long Re+W	<p><b>SuperBlitter control and status register</b></p> <table border="1"> <thead> <tr> <th>Bit</th> <th>31:7</th> <th>6</th> <th>5</th> <th>4:1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Reserved</td> <td>Done INT flag</td> <td>INT enable</td> <td>Blit mode</td> <td>Busy (read) Start (write)</td> </tr> <tr> <td>Read/Write</td> <td>R</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> <td>R+W</td> </tr> <tr> <td>Reset value</td> <td>0b00000000000000000000000000000000</td> <td>0b0</td> <td>0b0</td> <td>0b0000</td> <td>0 (read), 0 (write)</td> </tr> </tbody> </table>	Bit	31:7	6	5	4:1	0	Name	Reserved	Done INT flag	INT enable	Blit mode	Busy (read) Start (write)	Read/Write	R	R+W	R+W	R+W	R+W	Reset value	0b00000000000000000000000000000000	0b0	0b0	0b0000	0 (read), 0 (write)
Bit	31:7	6	5	4:1	0																						
Name	Reserved	Done INT flag	INT enable	Blit mode	Busy (read) Start (write)																						
Read/Write	R	R+W	R+W	R+W	R+W																						
Reset value	0b00000000000000000000000000000000	0b0	0b0	0b0000	0 (read), 0 (write)																						

**Start/Busy:**

When the Busy bit reads as '0' the Start bit may be set to '1' to start the SuperBlitter again. But make sure that the other SuperBlitter registers are properly set up first! While this bit is set to '1' the SuperBlitter is busy performing earlier copy operations.

**Blit mode (Yes, the bits are not enumerated in the expected order.):**

Code	Blit mode
0b0000	<i>dest = src1</i>
0b0010	Reserved
0b0100	Reserved
0b0110	Reserved
0b1000	Reserved
0b1010	Reserved
0b1100	Reserved
0b1110	Reserved
0b0001	<i>dest = src1 if src2 != 0, else dest = dest</i> Each <b>byte</b> of src1 is written to dest if the corresponding byte of src2 is not 0. For 8-bit chunky mode: src1=src2=sprite data For 16-bit sprites: src1=sprite data, src2=mask data
0b0011	<i>dest = src1 * src1.alpha + src2 * (256-src1.alpha)</i> 32-bit pixel alpha blending. Each pixel is R,G,B,A. 8 bits per channel. Only the alpha channel of src1 is used.
0b0101	<i>dest = src1 if src1.bit5 == 1, else dest = dest</i> Each 16bit word of src1 is written to dest if its bit 5 is 1. This is bit 0 of the green channel in a 16bit mode.
0b0111	<i>dest = src1 * alpha_reg + src2 * (256-alpha_reg)</i> 32-bit pixel alpha blending. Each pixel is R,G,B,A. 8 bits per channel. The alpha value is from the register at 0x80010064.
0b1001	Reserved
0b1011	Reserved
0b1101	Reserved
0b1111	Reserved

**INT enable & Done INT flag**

Writing INT enable to 1 makes the blitter generate an interrupt request when it completes a blitting job, and Done INT flag will be set to 1. You must clear it in your handler by writing 0 to Done INT flag. Make sure you put your ISR pointer in vector number 0xC6 (address 0x318) before starting a blitter job with interrupts enabled.

## ***Ethernet MAC core registers***

<b>Address</b>	<b>Size</b>	<b>Access types</b>	<b>Description</b>
0x80012000-0x80012053	84 bytes	Long Re+Wr	<b>Ethernet MAC core control registers</b>  See eth_speci.pdf, section 3.
0x80012400-0x800127FF	1024 bytes	Long Re+Wr	<b>Ethernet MAC core buffer descriptors</b>  See eth_design_document.pdf: section 2.3.7.3, and eth_speci.pdf: section 4.2.2.
0x80012800-0x80013FFF	6144 bytes	Long Re+Wr	<b>Ethernet MAC core data buffers</b>  3 blocks of 2048 bytes form 4 packet buffers of 1536 bytes each. Note that writing this using other sizes than Long will corrupt data.

## 256 color palette

Address	Size	Access types	Description																				
0xFFFF9800	1024 bytes	Byte Word Long Re+Wr	<b>256 color palette</b>  Each longword entry looks like this: <table border="1"><thead><tr><th>Bit</th><th>31:24</th><th>23:16</th><th>15:8</th><th>7:0</th></tr></thead><tbody><tr><th>Name</th><td>Red</td><td>Green</td><td>Reserved</td><td>Blue</td></tr><tr><th>Read/Write</th><td>R+W</td><td>R+W</td><td>R</td><td>R+W</td></tr><tr><th>Reset value</th><td>0x00</td><td>0x00</td><td>0x00</td><td>0x00</td></tr></tbody></table>	Bit	31:24	23:16	15:8	7:0	Name	Red	Green	Reserved	Blue	Read/Write	R+W	R+W	R	R+W	Reset value	0x00	0x00	0x00	0x00
Bit	31:24	23:16	15:8	7:0																			
Name	Red	Green	Reserved	Blue																			
Read/Write	R+W	R+W	R	R+W																			
Reset value	0x00	0x00	0x00	0x00																			