

## Summary

This document provides detailed reference information with respect to the VGA Controller peripheral device.

The VGA Controller provides a simple, 8-bit interface, between a host microcontroller and any VGA-compatible monitor. This variant of the Controller provides six modes of display, depending on the resolution chosen (640x480 (VGA) or 800x600 (SVGA)) and the color palette (either Black & White, 16 Colors, or 64 Colors).

If your design involves use of a 32-bit processor, there are several 32-bit VGA Controllers available, which can be summarized as follows:

- VGA32** - standard 32-bit VGA Controller with configurable color quality (1,2,4,8bpp), support for screen resolution up to 800x600 and screen refresh rate of up to 75Hz.
- VGA32\_16BPP** - as per the standard 32-bit VGA Controller, but with fixed 16bpp color quality.
- VGA32\_TFT** - this 32-bit VGA Controller is specifically used to interface to a TFT panel. It supports a fixed TFT screen resolution of 240x320 and a fixed refresh rate of 50Hz. Its color quality is fixed at 16bpp.



For more information on the VGA32, refer to the core reference [VGA32 – 32-bit VGA Controller](#).



For more information on the VGA32\_16BPP, refer to the core reference [VGA32\\_16BPP – 32bit VGA Controller with 16bpp Data Support](#).



For more information on the VGA32\_TFT, refer to the core reference [VGA32\\_TFT – 32-bit VGA Controller with TFT Interface](#).

There is also a configurable 32-bit VGA Controller – the WB\_VGA – which allows you to configure which of these three 32-bit Controllers is used after placement on the schematic sheet. For more information, refer to the core reference [WB\\_VGA Configurable Wishbone Display Driver](#).

## Features

- Compatible with any standard VGA- or SVGA-compatible monitor
- Two resolutions supported
  - 640x480 – standard VGA
  - 800x600 – standard SVGA
- Black & White, 16 Color and 64 Color display modes
- Processor-controlled horizontal and vertical display sizing.

## Available Devices

The VGA Controller device can be found in the FPGA Peripherals integrated library (FPGA Peripherals.IntLib), located in the `\Library\Fpga` folder of the installation.

## Functional Description

### Symbol

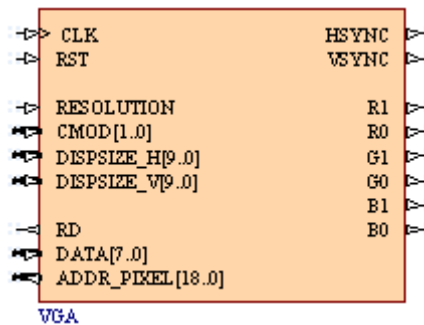


Figure 1. VGA Controller Symbol – 8-bit non-Wishbone variant (VGA)

### Pin Description

Table 1. VGA Pin description

Name	Type	Polarity/ Bus size	Description
<b>Global Control Signals</b>			
CLK	I	Rising	Global system clock. This clock determines the maximum rate at which pixels can be sent to the monitor. The frequency of the clock signal determines the refresh rate as follows: 640x480 CLK = 25MHz, Refresh = 60Hz CLK = 30MHz, Refresh = 72Hz 800x600 CLK = 40MHz, Refresh = 60Hz CLK = 50MHz, Refresh = 72Hz.
RST	I	High	Global system reset
<b>VGA Input Settings Signals</b>			
RESOLUTION	I	High / Low	This input selects the screen resolution to be used. 1 = 640x480 (VGA) 0 = 800x600 (SVGA).
CMOD	I	2	This input selects the color palette to be used: 00 = Black & White 01 = 16 Colors 10 = 64 Colors.
DISPSIZE_H	I	10	This input determines the number of viewable pixels to be displayed in each line of a frame and is therefore used to control the horizontal extents of the visible display area.
DISPSIZE_V	I	10	This input determines the number of lines to be displayed in a frame and is therefore used to control the vertical extents of the visible display area.

Name	Type	Polarity/ Bus size	Description
<b>Data Memory Control Signals</b>			
RD	O	High	This is the enable signal when data is required to be read from the memory space. This signal is controlled by, and follows, the internal line enable signal, en, generated by the Synchronization Unit of the Controller.
DATA	I	8	Data input from picture memory space. Data is stored in memory in bytes, the content of which depends on the chosen color palette as follows: Black & White mode : 8, 1-bit pixels 16 Colors mode : 2, 4-bit pixels 64 Colors mode : 1, 6-bit pixel (DATA[5..0]) Note : In 64 Colors mode, bits 7 and 6 of each byte are not used.
ADDR_PIXEL	O	19	Specifies the address of the next pixel in picture memory. Addresses are consecutive – once the end of the current line has been reached, the next address is that of the pixel at the start of the next line down in the frame.
<b>VGA Monitor Control Signals</b>			
HSYNC	O	Falling	Horizontal synchronization signal. This signal is used to control the horizontal deflection circuit in the VGA monitor, so that the start and end of a line of pixels is correctly displayed across the visible display area of the screen. The horizontal size of the display area is controlled by the DISPSIZE_H input to the Controller.
VSYNC	O	Falling	Vertical synchronization signal. This signal is used to control the vertical deflection circuit in the VGA monitor, so that the start and end of a frame (of lines) is correctly displayed between the top and bottom edges of the visible display area of the screen. The vertical size of the display area is controlled by the DISPSIZE_V input to the Controller.
R1 R0	O	High / Low	Provides the 2-bit digital signal for the intensity of red used in composing a pixel's displayed color. These two signals are inputs to a simple 2-bit DAC (external to the Controller) that provides the analog signal required by the VGA monitor.
G1 G0	O	High / Low	Provides the 2-bit digital signal for the intensity of green used in composing a pixel's displayed color. These two signals are inputs to a simple 2-bit DAC (external to the Controller) that provides the analog signal required by the VGA monitor.
B1 B0	O	High / Low	Provides the 2-bit digital signal for the intensity of blue used in composing a pixel's displayed color. These two signals are inputs to a simple 2-bit DAC (external to the Controller) that provides the analog signal required by the VGA monitor.

## Hardware Description

### Block Diagram

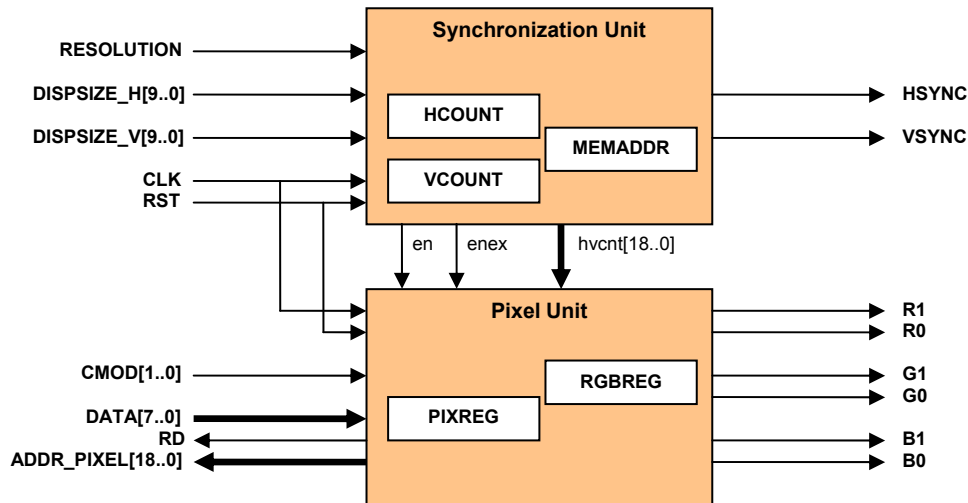


Figure 2. VGA Controller block diagram

### VGA Synchronization Unit

The Synchronization Unit provides the horizontal and vertical synchronization signals – HSYNC and VSYNC – that are required to correctly display a picture frame within the confines of a monitor's display area.

These synchronization signals are used as control inputs by the monitor's horizontal and vertical deflection circuits. These circuits deflect the electrons emitted by the three primary color electron guns (Red, Green, Blue) left to right and from top to bottom, respectively. HSYNC provides the start and stop times for the horizontal deflection circuit, so that a line of pixels is correctly drawn across the screen display. VSYNC provides the start and stop times for the vertical deflection circuit, so that the lines of a frame are correctly drawn from the top to the bottom of the screen display.

The resolution for the display is defined by the level on the RESOLUTION input. If High, the 640x480 resolution is used (VGA). If Low, the 800x600 resolution (SVGA) is used.

Although the resolution determines the area of a monitor's screen within which an image can be displayed, the full extents of the chosen resolution do not have to be used. The actual extents of the image display area on the screen can be controlled by the use of the DISPSIZE\_H and DISPSIZE\_V inputs. These inputs determine the total number of pixels to be used in a line and the total number of lines in a frame, respectively.

#### Horizontal (Line) Period

The Horizontal Line Period – the length of time (expressed in cycles of the CLK signal) between starting to send one complete line of pixels and the next – is defined as:

$$T_{LSync} + Pix_{Total} + Blanking_{Left} + Blanking_{Right}$$

- $T_{LSync}$**  - the synchronization time for a line. For 640x480 resolution, this value is 95 clock cycles. For 800x600 resolution, this value is 120 clock cycles.
- $Pix_{Total}$**  - the integer value of the DISPSIZE\_H input. This is the total viewable pixels sent per line to the monitor.
- $Blanking_{Left}$**  - the number of blank pixels that are inserted to the left of the viewable pixel area of the screen display. The number of blank pixels depends on the resolution used:  
 640x480 : No. of blank pixels =  $23 + ((640 - Pix_{Total}) / 2)$   
 800x600 : No. of blank pixels =  $56 + ((800 - Pix_{Total}) / 2)$
- $Blanking_{Right}$**  - the number of blank pixels that are inserted to the right of the viewable pixel area of the screen display. The number of blank pixels depends on the resolution used:

$$640 \times 480 : \text{No. of blank pixels} = 47 + ((640 - \text{Pix}_{\text{Total}}) / 2)$$

$$800 \times 600 : \text{No. of blank pixels} = 63 + ((800 - \text{Pix}_{\text{Total}}) / 2)$$

Note: As a pixel is sent on each rising edge of CLK, the values for **Pix<sub>Total</sub>**, **Blanking<sub>Left</sub>** and **Blanking<sub>Right</sub>** equate to the total number of clock cycles involved for each.

To express the line period in units of time, the result of the above equation must be multiplied by **1/frequency of CLK**.

### Horizontal Counter (HCOUNT)

The horizontal counter (or pixel counter) stores the current horizontal position within a line of pixels. The counter is reset to zero when the VGA Controller receives an external reset signal (RST). The size of the counter depends on the value chosen for DISPSIZE\_H, as the range is simply:

0 to Horizontal Line Period - 1

Substituting the relevant values into the expression for the Horizontal Line Period and taking the maximum number of pixels in a line for each of the supported display resolutions, the maximum ranges for the counter are:

- 640x480: 0 to 804
- 800x600: 0 to 1038.

The counter has 10-bit resolution.

While the value in the counter is less than the Horizontal Line Period, the counter is incremented on the rising edge of the external clock signal (CLK). The counter is cyclic in its operation; on reaching the upper limit of its range, it is rolled over to zero again on the rising edge of the next clock cycle.

The value in the horizontal counter is tested to determine whether or not the address counter (MEMADDR) is incremented. It is also used to determine the activation of the HSYNC signal.

### Generation of the Horizontal Synchronization Signal – HSYNC

The HSYNC signal is High (inactive) after an external reset signal (RST) is received by the VGA Controller. The signal is updated on each rising edge of the external clock signal (CLK).

The state of the HSYNC signal depends on the value stored in the horizontal counter and is driven low when:

$$\text{HCOUNT} \geq (\text{Pix}_{\text{Total}} + \text{Blanking}_{\text{Left}})$$

and remains low while:

$$\text{HCOUNT} < (\text{Pix}_{\text{Total}} + \text{Blanking}_{\text{Left}} + T_{\text{LSync}})$$

### Vertical (Frame) Period

The Vertical Frame Period – the length of time (expressed in cycles of the HSYNC signal) between starting to send one complete frame and the next – is defined as:

$$T_{\text{FSync}} + \text{Lin}_{\text{Total}} + \text{Blanking}_{\text{Left}} + \text{Blanking}_{\text{Right}}$$

- T<sub>FSync</sub>** - the synchronization time for a frame. For 640x480 resolution, this value is 2 HSYNC cycles. For 800x600 resolution, this value is 6 HSYNC cycles.
- Lin<sub>Total</sub>** - the integer value of the DISPSIZE\_V input. This is the total number of viewable lines that will be sent per frame to the monitor.
- Blanking<sub>Left</sub>** - the number of blank lines that are inserted above the viewable pixel area of the screen display. The number of blank lines depends on the resolution used:  
 640x480 : No. of blank lines = 14 + ((480 – Lin<sub>Total</sub>) / 2)  
 800x600 : No. of blank lines = 37 + ((600 – Lin<sub>Total</sub>) / 2)
- Blanking<sub>Right</sub>** - the number of blank lines that are inserted below the viewable pixel area of the screen display. The number of blank lines depends on the resolution used:  
 640x480 : No. of blank lines = 32 + ((480 – Lin<sub>Total</sub>) / 2)  
 800x600 : No. of blank lines = 23 + ((600 – Lin<sub>Total</sub>) / 2)

Note: As a line is sent on each rising edge of HSYNC, the values for **Lin<sub>Total</sub>**, **Blanking<sub>Left</sub>** and **Blanking<sub>Right</sub>** equate to the total number of HSYNC cycles involved for each.

## Vertical Counter (VCOUNT)

The vertical counter (or line counter) stores the current vertical position within a frame of lines. The counter is reset to zero when the VGA Controller receives an external reset signal (RST). The size of the counter depends on the value chosen for DISPSIZE\_V, as the range is simply:

0 to Vertical Frame Period - 1

Substituting the relevant values into the expression for the Vertical Frame Period and taking the maximum number of lines in a frame for each of the supported display resolutions, the maximum ranges for the counter are:

- 640x480: 0 to 527
- 800x600: 0 to 665.

The counter has 10-bit resolution.

While the value in the counter is less than the Vertical Frame Period, the counter is incremented on the rising edge of the horizontal synchronization signal (HSYNC). The counter is cyclic in its operation; on reaching the upper limit of its range, it is rolled over to zero again on the next rising edge of HSYNC.

The value in the vertical counter is tested to determine whether the address counter (MEMADDR) is rolled over to zero or not. It is also used to determine the activation of the VSYNC signal.

## Generation of the Vertical Synchronization Signal - VSYNC

The VSYNC signal is High (inactive) after an external reset signal (RST) is received by the VGA Controller. The signal is updated after every line of pixels is completed (i.e. on each rising edge of the HSYNC signal).

The state of the VSYNC signal depends on the value stored in the vertical counter and is driven low when:

$$VCOUNT \geq (Lin_{Total} + Blanking_{Left})$$

and remains low while:

$$VCOUNT < (Lin_{Total} + Blanking_{Left} + T_{FSync})$$

## Address Counter (MEMADDR)

The address counter is used to store the position of the next consecutive pixel in the frame. Its value is passed to the Pixel Unit on the internal bus signal hvcnt, which is then used to provide the ADDR\_PIXEL signal, to obtain the next pixel from picture memory.

The counter is reset to zero when the VGA Controller receives an external reset signal (RST). The size of the counter depends on the values chosen for DISPSIZE\_H and DISPSIZE\_V, as the range is simply:

$$0 \text{ to } (Pix_{Total} \times Lin_{Total}) - 1$$

Taking the maximum number of pixels in a line and lines in a frame, for each of the supported display resolutions, the maximum ranges for the counter are:

- 640x480: 0 to 307199
- 800x600: 0 to 479999.

The counter has 19-bit resolution.

While the value in the horizontal counter (HCOUNT) is less than the total number of viewable pixels in a line (**Pix<sub>Total</sub>**, the integer value of DISPSIZE\_H), the counter is incremented on the rising edge of the external clock signal (CLK). Pixel addressing within the frame is consecutive. When the counter reaches the last pixel in a line, its incremented value is the first pixel in the next line down.

The address counter will continue to be incremented until the value in the vertical counter (VCOUNT) is greater than or equal to the total number of viewable lines in a frame (**Lin<sub>Total</sub>**, the integer value of DISPSIZE\_V). At this point, it will be rolled over to zero.

## Blank Pixel Generation

The total number of viewable pixels in a line and viewable lines in a frame is determined by the display resolution chosen through the RESOLUTION input (1= 640x480; 0 = 800x600) and the values received on the DISPSIZE\_H and DISPSIZE\_V buses. Whether the full extent of the chosen display resolution is used or not, the areas of the monitor screen to the top, bottom, left and right of the viewable frame area are blanked, by putting black pixels at the required line-pixel positions. This has the effect of centering the image on the screen.

The color generated for a pixel in the Pixel Unit depends on whether the particular pixel requires to be blanked or not. The Synchronization Unit provides a signal to the Pixel Unit for this very reason. This is the line display enable signal - en. The signal is checked on each rising edge of the external clock signal (CLK) and is set as follows:

If ( $HCOUNT \geq Pix_{Total}$ ) or ( $VCOUNT \geq Lin_{Total}$ ) then

en = 0 (pixel requires to be blanked – set color to be black)

Else

en = 1 (pixel is a viewable pixel – generate RGB color accordingly).

### VGA Signal Timing

Figure 3 summarizes the signal timing involved in sending a line of pixels and a frame of lines. The actual time values differ according to the resolution selected (640x480 or 800x600), the processor-defined values for DISPSIZE\_H and DISPSIZE\_V and the frequency of the external clock signal (CLK).

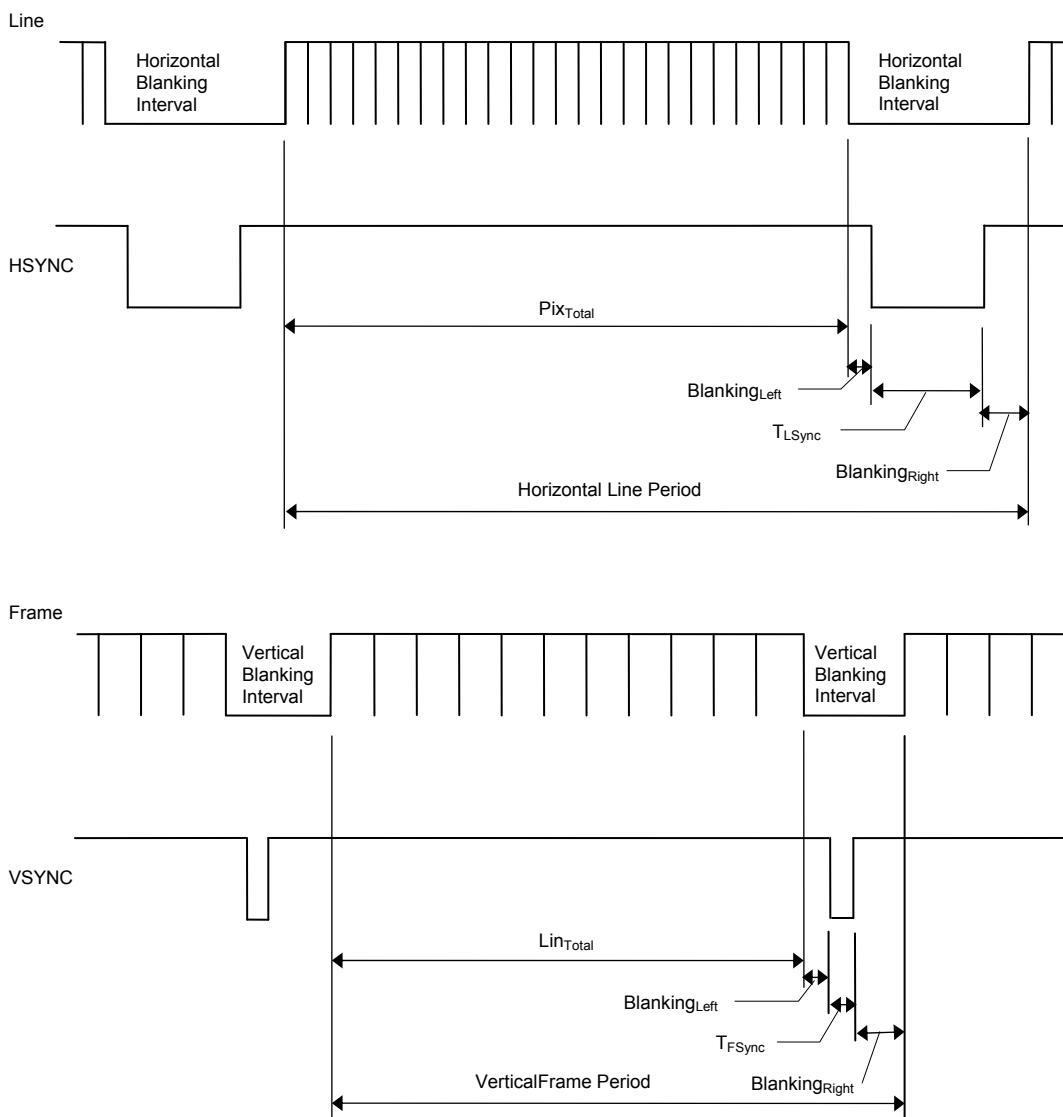


Figure 3. Horizontal (line) and vertical (frame) timing signals

### VGA Pixel Unit

The Pixel Unit provides access to the pixilated image stored in external picture memory, reading in data a byte at a time and formatting each pixel to be displayed. For each pixel, the 6-bit RGB value required for the control of the monitor's three, primary color electron guns is generated, so that the pixel is displayed on the screen with the correct color.

## Accessing the Stored Image

The image to be displayed on the monitor screen is written, by the host microcontroller, into external memory space (RAM). This memory space can be located anywhere (e.g. using a block of RAM within the FPGA design, or using the SRAM on the NanoBoard itself).

Pixel data is stored in the memory space in bytes. The number of pixels in a byte depends on which color palette is being used for the image (selected by the CMOD input):

- in Black & White mode – one byte of data in RAM contains 8, 1-bit pixels
- in 16 Colors mode – one byte of data in RAM contains 2, 4-bit pixels
- in 64 Colors mode – one byte of data in RAM contains 1, 6-bit pixel (DATA[7..6] are not used)

The size of memory required to store a picture is determined by the total number of viewable pixels in a line (determined by DISPSIZE\_H), the total number of viewable lines in a frame (determined by DISPSIZE\_V) and the number of pixels stored in each byte in memory space:

Memory required for picture =  $(\text{Pix}_{\text{Total}} \times \text{Lin}_{\text{Total}}) / \text{number of pixels per byte}$

The address in RAM where the next pixel is stored is determined using an internal signal provided by the Synchronization Unit – hvcnt – which reflects the current contents of the MEMADDR register. The exact addressing is described below.

### Black & White Mode

The picture memory address – the byte of data containing the next 8 pixels – is determined by using bits 18..3 of hvcnt and right shifting the contents by three:

$\text{ADDR\_PIXEL} = "00" \& \text{hvcnt}[18..3]$

### 16 Colors Mode

The picture memory address – the byte of data containing the next 2 pixels – is determined by using bits 18..1 of hvcnt and right shifting the contents by one:

$\text{ADDR\_PIXEL} = '0' \& \text{hvcnt}[18..1]$

### 64 Colors Mode

The picture memory address – the byte of data containing the next pixel – is determined by using the full value of hvcnt:

$\text{ADDR\_PIXEL} = \text{hvcnt}[18..0]$

## The Pixel Register (PIXREG)

The Pixel register is used to receive the byte of pixel data read from the current address in memory space. The register, PIXREG, is reset to zero when the VGA Controller receives an external reset signal (RST).

The register is updated on each rising edge of the CLK signal. Data can be read from the memory space as long as the RD signal is active (High). The RD signal is itself controlled by the external line display enable signal, enex. This internally generated signal is defined as follows:

If  $((\text{HCOUNT} > (\text{Pix}_{\text{Total}}+1)) \text{ and } (\text{HCOUNT} < \text{Line Period}))$  or  $((\text{VCOUNT} > (\text{Lin}_{\text{Total}}+1)) \text{ and } (\text{VCOUNT} < \text{Frame Period}))$  then

$\text{enex} = 0$

Else

$\text{enex} = 1$

When enex is Low, read access from memory is disabled (RD = 0).

The point at which data is loaded from memory into PIXREG depends on the particular color palette that is chosen – Black & White, 16 Colors, or 64 Colors.

### Black & White Mode

The next byte of data will be loaded into the register whenever the lowest three bits of the hvcnt signal – received from the Synchronization Unit - are "000".

For the currently loaded byte, the active pixel is always in the lowest bit position of the Pixel register. Each pixel in the data byte is moved into this active pixel position by shifting the contents of the register right by one bit, on each rising edge of CLK.

### 16 Colors Mode

The next byte of data will be loaded into the register whenever the lowest bit of the hvcnt signal – received from the Synchronization Unit - is a '0'.



For the currently loaded byte, the active pixel is always in the low order nibble of the Pixel register. Remember that in this mode, each byte of data contains two pixels. The second pixel is moved into this active pixel position by shifting the contents of the register right by four bits, on the rising edge of CLK.

#### 64 Colors Mode

The next byte of data will be loaded into the register on the rising edge of the external system clock signal (CLK). In this mode, the read of pixel data does not depend on the status of the hvcnt signal received from the Synchronization Unit.

#### The RGB Register (RGBREG)

The RGB register is used to store the six bits that are required for driving the red, green and blue color guns of the target monitor. When the chosen color palette is either Black & White or 16 Colors, these six bits are obtained by mapping the value of the active pixel to a predefined RGB code. When the chosen color palette is 64 Colors, the actual pixel value is used directly.

The register, RGBREG, is reset to zero (000000) when the VGA Controller receives an external reset signal (RST). This RGB code represents black.

The register is updated on each rising edge of the CLK signal and the value loaded is dependent on the state of the line display enable signal, en. When en is Low, blanking is required and RGBREG is loaded with the code for black (000000).

Table 2 illustrates the mapping of a 1-bit pixel (Black & White mode) and a 4-bit pixel (16 Colors mode) into the required RGB color code.

Table 2. Mapping of pixel data to RGB color

Color Palette Mode	State of line display enable signal (en)	Active pixel value	RGB Color Code (loaded into RGBREG)	
B & W	1	0	000000	Black
		1	111111	White
	0	0	000000	Black
		1	000000	Black
16 Colors	1	0000	000000	Black
		0001	100000	
		0010	001000	
		0011	101000	
		0100	000010	
		0101	100010	
		0110	001010	
		0111	010101	
		1000	101010	
		1001	110000	Red
		1010	001100	Green
		1011	111100	Yellow
		1100	000011	Blue
		1101	110011	Magenta
1110	001111	Cyan		
1111	111111	White		
	0	0000	000000	Black
		0001	000000	Black

Color Palette Mode	State of line display enable signal (en)	Active pixel value	RGB Color Code (loaded into RGBREG)	
		0010	000000	Black
		0011	000000	Black
		0100	000000	Black
		0101	000000	Black
		0110	000000	Black
		0111	000000	Black
		1000	000000	Black
		1001	000000	Black
		1010	000000	Black
		1011	000000	Black
		1100	000000	Black
		1101	000000	Black
		1110	000000	Black
		1111	000000	Black

The RGB color code stored in the RGB register is output from the VGA Controller as separate 2-bit R, G and B values (outputs R0, R1, G0, G1, B0 and B1).

The monitor itself expects analog signals as inputs to its electron gun control circuits. This is achieved by using 2-bit digital to analog converter circuitry, located on the NanoBoard-NB1 itself, as shown in Figure 4.

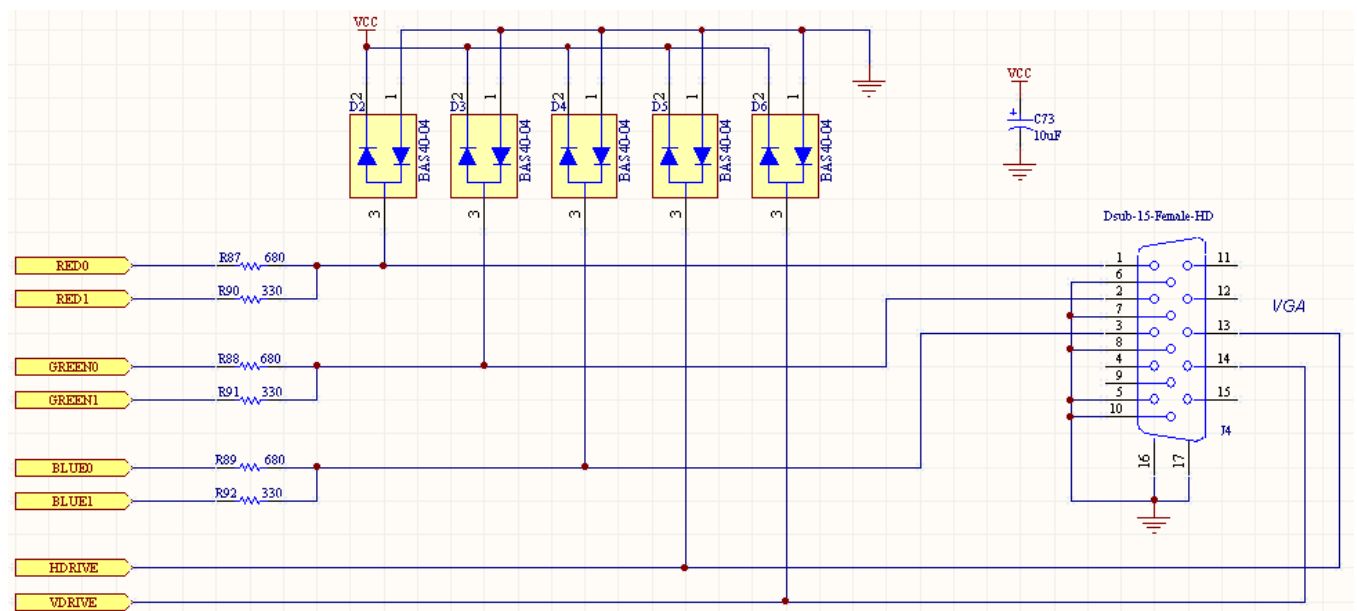


Figure 4. Digital to Analog RGB conversion

For each color, the 2-bit digital signal from the VGA Controller can be converted into 4 distinct analog levels. These levels specify the intensity of each of the three primary colors to use when displaying the pixel on the monitor's screen. The levels range from 0V (total darkness) to 0.7V (maximum brightness).

With each analog input being one of four possible levels, the monitor can display each pixel on the screen with one of 64 different color permutations.

## Revision History

Date	Version No.	Revision
30-Dec-2003	1.0	New product release
01-Dec-2004	1.1	Schematic symbol update
04-Mar-2005	1.2	Addition of 32-bit Wishbone variant of the Controller – VGA32.
27-May-2005	1.3	Updated for Altium Designer SP4
30-Sep-2005	1.4	Correction to definition of RESOLUTION pin in table 17. When High, the resolution is 640x480 and not 640x400 as previously stated.
12-Dec-2005	1.5	Path references updated for Altium Designer 6
06-Aug-2006	2.0	Document renamed (from CR0113 VGA Controller) and updated for Altium Designer 6.4. 32-bit variant information extracted to a new document – CR0169 VGA32 – 32-bit VGA Controller).
13-Mar-2008	3.0	Updated for Altium Designer Summer 08
30-Aug-2011	-	Updated template.

Software, hardware, documentation and related materials:

Copyright © 2011 Altium Limited.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of Altium Limited. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment.

Altium, Altium Designer, Board Insight, DXP, Innovation Station, LiveDesign, NanoBoard, NanoTalk, OpenBus, P-CAD, SimCode, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.