

## iCE65 as Graphics LCD Interface

### Overview

A Graphic LCD controller is used to control the display and refresh operation of LCDs. This design example illustrates the implementation of a 2.8" TFT LCD controller using iCE65 FPGAs. As a controller, the iCE65 FPGA generates all the required interfacing signals and implements the functions that are required for such a system.

### Description

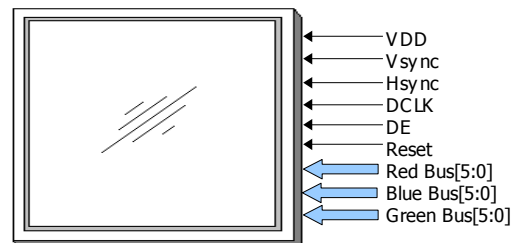
The 2.8" Graphic LCD controller implemented in this project uses the QVGA (Quarter Video Graphics Array) interface for driving a 240x320 pixel sized Low Temperature PolySilicon (LTPS) TFT LCD display using 16-bit RGB colour data.

The LCD display used is a transfective type with 240 x RGB x 320 number of dots. The colour arrangement in each pixel is RGB stripe type. The display is capable of accepting 18-bit RGB data (6 bits for each colour) and can therefore display up to 262 K colours.

Quarter Video Graphics Array (QVGA) is a popular term for a computer display with 320 x 240 resolution. QVGA displays are most often seen in mobile phones, PDAs and some handheld game consoles. Often, the displays are in a "portrait" alignment (as opposed to "landscape") and are referred to as 240 x 320 as the displays are taller than they are wide.

The name is derived from the fact that it offers 1/4 of the 640 x 480 maximum resolution of the original IBM VGA display technology, which became a de facto industry standard in the late 1980s. QVGA implementations are not compatible with, nor directly derived from, standard VGA chipsets or interfaces; the term refers only to the display's resolution and thus the abbreviated term QVGA or Quarter VGA is more appropriate to use.

Figure 1 below shows the interface signals to the LCD panel. These signals are received by an LTPS QVGA LCD Driver ASIC which comes built in with the LCD panel. This ASIC houses a DAC and other control logic for driving the thin film transistor corresponding to each pixel of the LCD display.



**Fig 1: LCD panel**

### Implementation

LCD controller generates the various QVGA signals required for driving the LCD display. This display module is a Topolly TD028STEB2 and the timing requirements are detailed in its specification documents that may be obtained by special request from [www.tpo.biz](http://www.tpo.biz). This module also takes care of the power up and power down sequences of the LCD driver chip. The `lcd_clear` input is used to blank the display (white screen). This input goes high just before power off displaying atleast two full frames of white. Power Sequencer/Debounce Logic Constantly monitors the status of the `on_off` input pin in order to take care of the power down sequence of the LCD display.

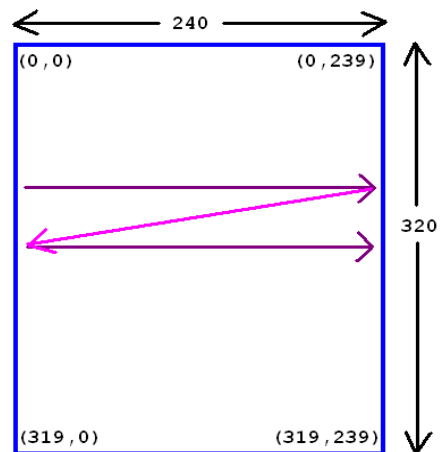
The LCD interface consists of 4 active signals along with the colour bit data buses which control the entire display activity. These are the Vertical Sync (Vsync), Horizontal Sync (Hsync), Data CLock (DCLK) and Data Enable (DE) signals. The Hsync and Vsync signals are used for the synchronization of the video frame with respect to the display. By changing the data on the 18-bit RGB bus, all possible colours (262 K) can be produced. The value of each 6-bit colour data is used to change the intensity of each primary colour (red, green and blue) in a pixel.

The screen refresh process begins in the top left corner and paints one pixel at a time from left to right. This is shown in Figure 2. At the end of the first row, the row increments and the column address is reset to the first column. Once the entire screen has been painted, the refresh process begins again. The video signal must redraw the entire screen at least 60 times per second to provide for motion in the image and to reduce flicker: this period is called the refresh rate.

The Vsync signal tells the monitor to start displaying a new image or frame, and the monitor starts in the upper left corner with pixel (0,0). The horizontal sync signal tells the monitor to refresh another row of 240 pixels. After 320 rows of pixels are refreshed with 320 Hsync signal cycles, a vertical sync signal resets the monitor to the upper left corner and the process continues. Figure 3 shows the Hsync and Vsync signal cycles in more detail.

During the time when pixel data is not being displayed and the beam is returning to the left column to start another horizontal scan, the DE signal should be kept low. A dedicated memory buffer can be used to store the color value of every pixel in the display. This memory can be read out as the pixels are scanned across the screen to produce the RGB signals. It is important to source the pixel data from this memory buffer at the rate required by the LCD while jumping from one pixel to another. Hence, extremely fast memories like SRAM or DRAM chips are best suited for realizing this buffer.

Figure 5 shows the high level view of the Graphics LCD Controller. Interface details of this design example is listed in Table 1. The resource utilization summary is listed in Table 2.



**Fig 2: LCD Refresh Sequence**

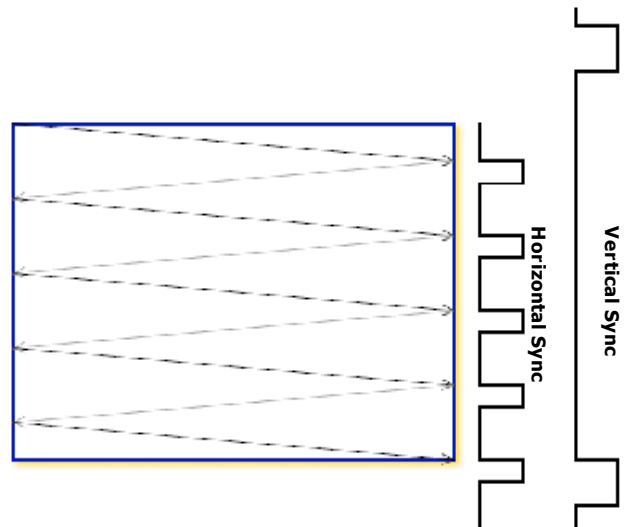
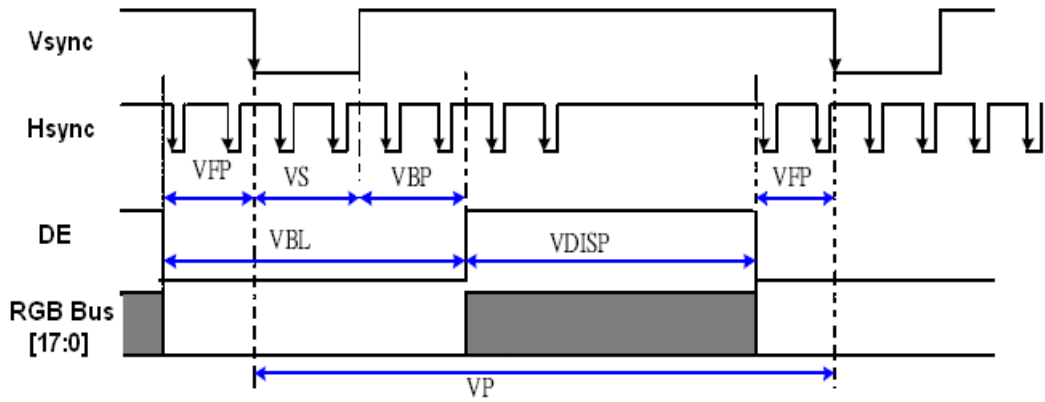


Fig 3: Vsync and Hsync Signal Cycle

Vertical Timing Chart



Horizontal Timing Chart

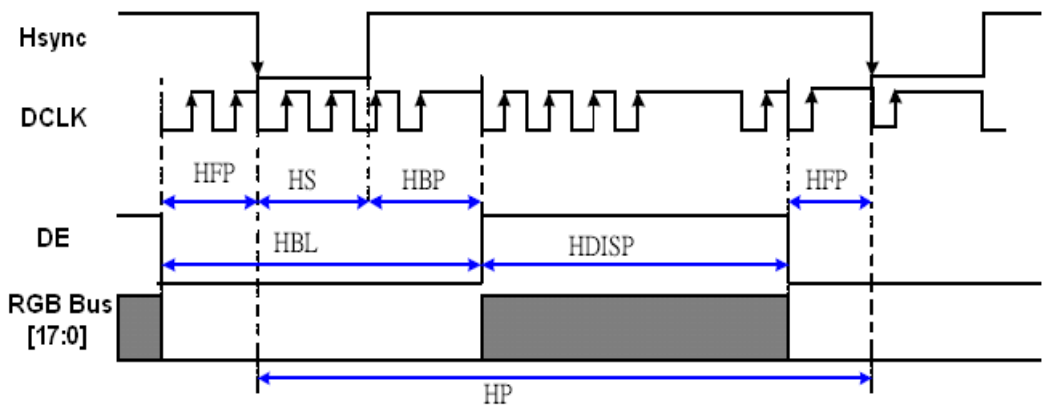
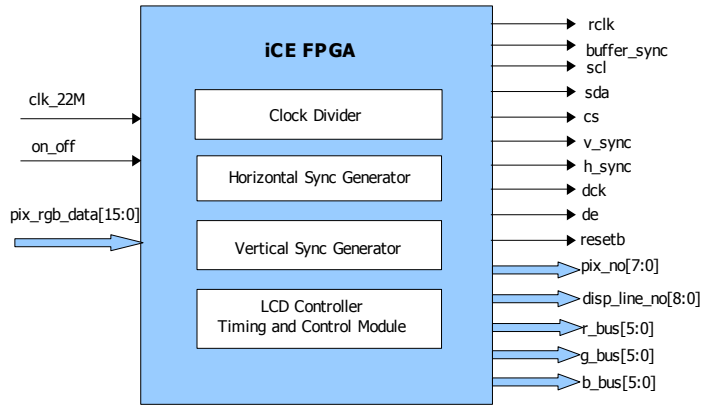


Fig 4: Graphics LCD Timing Diagrams



**Fig 5: Block diagram of Graphics LCD Controller**

Pin	Direction	Description
clk_22M	Input	System clock
on_off	Input	Toggle switch used to turn on/off the LCD display
pix_rgb_data[15:0]	Input	RGB pixel data from the display memory
rclk	output	Clock control signal to the display memory
pix_no[7:0]	output	Location of the pixel within a line to display RAM
disp_line_no[8:0]	output	Line number of pixel whose RGB data is required
buffer_sync	output	Synchronizes the display memory's fill operation
scl	output	Clock signal to LCD driver chip
sda	output	Data signal to LCD driver chip
cs	output	Chip select signal to LCD driver chip
v_sync	output	Vertical sync signal
h_sync	output	Horizontal sync signal
dck	output	Data clock to LCD display
de	output	Data enable to LCD display
resetb	output	Reset signal to LCD display
r_bus[5:0]	output	QVGA(Red) pixel data to LCD display
g_bus[5:0]	output	QVGA(Green) pixel data to LCD display
b_bus[5:0]	output	QVGA(Blue) pixel data to LCD display

**Table 1: Pin Description**

Device	Logic Cells	IO Cells
iCE65L04-UCB284	657	63

**Table 2: Resource Utilization**

## Conclusion

This design example demonstrates the implementation of a Graphics LCD controller using iCE FPGAs. iCE FPGA's very low power capabilities makes iCE FPGAs an obvious choice for implementing a Graphics LCD controller for battery operated compact/handheld devices PDAs, cellular phones etc..

## Disclaimer

SiliconBlue is providing this document, design example, or information "as is." SiliconBlue does not ensure that this implementation or information is void of any claims of infringement. As a reader / implementer, you are responsible for obtaining any rights you may require for your implementation. SiliconBlue also does not warranty the fitness of this design example to be readily implemented in any specific context. It is your sole responsibility to validate this design example and its correctness while implementing it for your requirement.