VGA Port and PS/2 Mouse Port

Task:

Using UP3 Education Kit board implement a cursor controlled with a PS/2 mouse moving on monitor screen. The color of the cursor should change whenever any button on the mouse is pressed. The background color should not be black. The cursor is not allowed to leave the screen. Refer to UP3 manual for pin numbers.

VGA port

UP3 Education Kit board includes a standard VGA connector. It may be connected directly to most PC monitors or flat-panel LCD displays using a standard monitor cable. The FPGA device controls five VGA signals: Red, Green, Blue, Horizontal Sync and Vertical Sync. Eight possible colors can be generated (Table 1).

Table 1: Color codes

Red	Green	Blue	Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

The major component inside VGA monitor is color Cathode Ray Tube (CRT). The electron beam must be scanned over the viewing screen in a sequence of horizontal lines to generate an image. Light is generated when the beam is turned on by a video signal and it strikes a color phosphor dot on the face of the CRT. The video signal must redraw the entire screen at least 60 times per second to provide motion in the image and to reduce flicker.

The beam moves on the display surface in a “raster” pattern, horizontally from left to right and vertically from top to bottom. Information is displayed only when it moves in forward direction. Thus, much of the time is lost in blank periods, when beam is reset and stabilized to begin a new pass. VGA controller generates two synchronizing signals – Horizontal Sync (HS) and Vertical Sync (VS) to control the raster pattern and video data delivery.

The VS signal defines the refresh frequency of the display (the frequency at which all information is redrawn). The HS signal defines the number of horizontal lines displayed at a given refresh rate. Both signals have the same waveform (Figure 1), but their timing is different. The pixel clock defines the time available to display one pixel of information. Note that information can be displayed only during Display Time interval. Color signals must be set to Low for other intervals.

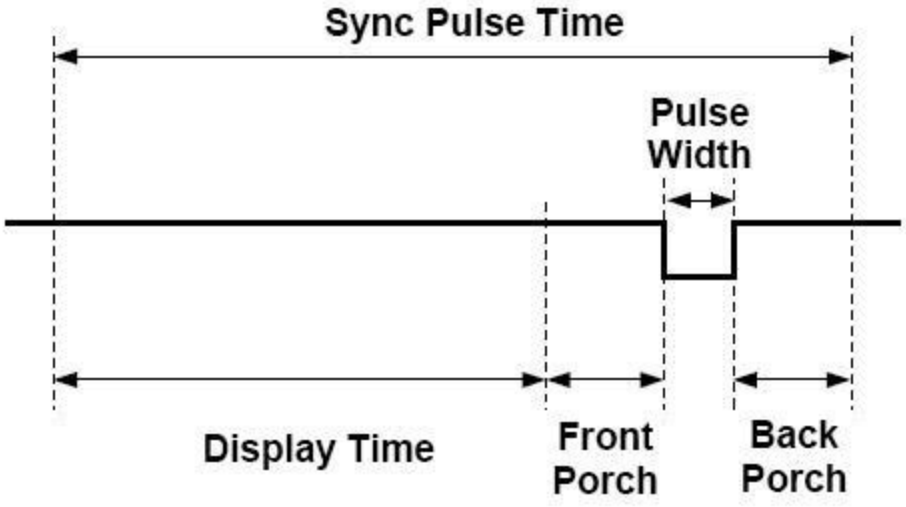

Figure 1: Sync signal waveform

The timing information in Table 2 provides as an example of how the FPGA device might drive a monitor in 800-pixels by 600-rows mode with 60Hz refresh rate. The pixel clock frequency for this mode should equal 40 MHz. However, master clock chip on the board provides 48 MHz. In order to get the right pixel clock frequency Phase-Locked Loop (PLL) megafunction should be used.

From the **Tools** drop-down menu select **MegaWizard Plug-In Manager**. Select **Create a new megafunction** option and click **Next**. Expand **IO** folder and select **ALTPLL** component. Set output file type to **VHDL**, name it and click **Next** to proceed. On the following page set **Speed grade** to **8** and input clock frequency to **48** MHz. On the next page uncheck all boxes to remove excess inputs and outputs, proceed further. Set either the **Output clock frequency** to **40** MHz, or **Multiplication factor** to **5** and **Division factor** to **6** (both options give equal result). There would be nothing to configure on the remaining pages, so just click **Finish** to move to the **Summary**. Be sure to check the *megafunction_name.cmp* file and click **Finish** to generate design files for PLL component. To find the PLL component declaration in VHDL go to **Insert Template** and search under **Megafunctions** category.

Table 2: 800x600 60Hz mode timing

Parameter	Vertical Sync (lines)	Horizontal Sync (pixel clocks)
Sync Pulse Time	628	1056
Display time	600	800
Pulse Width	4	128
Front Porch	1	40
Back Porch	23	88

Generally, a counter clocked by the pixel clock could control the horizontal timing. From its value current pixel display location on a given row can be easily tracked, as well as the correct time for HS signal transitions. A separate counter can do the same to control the vertical timing. It increments with each HS pulse, tracks the current display row and can be used to control the VS signal transitions. Values of these two constantly running counters may be used to form a Video Memory address for the currently displayed pixel.

PS/2 mouse interfaces

Mouse uses the bi-directional two-wire PS/2 serial bus, consisting of clock line and data line, to communicate with host device. Clock line is normally controlled by the mouse, but it can also be driven by host device when the data transmission from the mouse must be stopped.

Both lines are open-collector with pull-up resistors to supply voltage. It means that line can either be driven Low or left floating. To implement a bi-directional transmission a tri-state buffer must be used. It can be instantiated using the following VHDL code:
with output_enable select data_out <= data_in when '1', 'Z' when others;

Mouse should be attached to UP3 Education Kit board prior to turning it on. FPGA device outputs are tri-stated and float High forming an idle state. At power-up, the mouse performs a self-test called BAT (Basic Assurance Test) and sends AAh (successful) of FCh (error). The device ID (00h) is sent afterwards. These messages should not be considered in the interface, as no acknowledge is required.

After the device ID is sent, mouse enters Stream Mode. It is a default operating mode, when mouse sends data packets whenever movement is detected or any button is pressed. However, data reporting is disabled at the moment. In order to make the mouse start sending data, an Enable Data Reporting command (F4h) must be issued by host device.

To send data to the mouse the host device should inhibit any data transmissions first by driving the clock line Low for at least 60 us. On the other hand, the mouse is not supposed to send anything at the moment. Thus, the host device may drive the data line Low right away to signal it has a command to send. As this is the only time when the host device sends data to the mouse, the clock line may be controlled by mouse only and do not have to be declared as bi-directional.

When the mouse detects Ready to Send condition, it starts generating clock pulses and reading data. The command should be transmitted serially starting from the least significant bit. Data is followed by an odd parity bit and a stop bit. When the packet is received, the mouse respond with Acknowledge (FAh) message. Data reporting is now enabled.

The mouse employs a relative coordinate system. The movement is registered in two dimensions: horizontal (X axis) and vertical (Y axis). Instead of the absolute coordinates,

mouse sends the offset values when moved. These values are relative to the position the mouse was in prior to sending the data packet. Offsets are 9-bit wide and can be positive (up or right movement) or negative (down or left movement).

The standard PS/2 mouse sends data using three consecutive packets. Each packet begins with the Start bit (always Low), followed by a data byte (starting with least significant bit), the Odd Parity bit and the Stop bit (always High). Data bytes contain mouse movement and button status information (Table 3). Byte 2 and Byte 3 contain the motion values stored in two's complement format. Byte 1 contains the overflow bits (Yo and Xo), the sign bits (Ys and Xs) and button status bits (M for middle, R for right and L for left).

Table 3: PS/2 mouse data packet format

Bit	7	6	5	4	3	2	1	0
Byte 1	Yo	Xo	Ys	Xs	'1'	M	R	L
Byte 2	X7	X6	X5	X4	X3	X2	X1	X0
Byte 3	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Mouse interface would be more reliable if a clock filter is applied. Whenever an electrical pulse is transmitted, electromagnetic properties of the wire cause the pulse to be distorted and some portions of it may be reflected from the end of the wire. This reflected pulse can be strong enough to cause additional clocks to appear on the clock line. To solve this problem the PS/2 clock signal must be fed into 16-bit shift register, which is clocked at 48MHz. When it is fully loaded High, the filtered clock value goes also High. Similarly, filtered clock value changes to Low when 16-bit shift register is fully loaded Low. Of course, some keyboards work fine without the clock filter, but some do not.