Design and Implementation of FPGA Based VGA Monitor and PS2 Keyboard Interface Technique

Shivam Sharma¹ & Mr. D.S Gangwar²
¹M.Tech Scholar Dept. of VLSI Design, F.O.T. Uttarakhand Technical University, Dehradun.
²Asst. Prof., Dept. of VLSI Design, F.O.T. Uttarakhand Technical University, Dehradun.

Abstract: This project gives the detailed study about the designing of VGA (Video Graphic Array) Controller and PS-2 keyboard controller, by using combination of three bit input data to Control eight different colors to display text on the screen by using PS2 keyboard as database online for the input. Three color signal collectively referred as R (red), G (green) and B (blue) signal. The VGA monitor screen uses a resolution of 640 by 480 by mode to display colors. The project is developed by using Xilinx ISE 14.7 software and Spartan-3E-500 FPGA board to develop the project into a complete module. The timing diagram must be appropriate, in order to get VGA monitor controller displays properly. The design will be written by using Verilog coding style ensure the VGA controller and PS-2 controller work properly. The behavioral simulation was done by using Xilinx ISE Tool software to verify the working of the design. The Spartan 3E-500 starter Kit board was chosen to implement the design.

Keywords: FPGA (FIELD PROGRAMMABLE GATE ARRAY), VGA (VIDEO GRAPHIC ARRAY), PS2 KEYBOARD, SPARTAN 3E-500

1. INTRODUCTION

As the performance of the monitors has increased, the demand for high-speed VGA controllers, which can provide high resolutions, has also increased. The purpose of this project is to display the monitor with the help of a single chip, like an FPGA working as VGA controller. Since, a single FPGA is used, the power consumption as well as the hardware requirements is less and the efficiency is more. The Spartan 3E-500 Field-Programmable Gate Array family gives users high performance, abundant logic resources, and a rich feature set, all at an exceptionally low price. This project aims at displaying a character on the monitor, which can be read. The size and the location of the characters can be controlled by using the Verilog hardware design language. The character can be displayed on the monitor only by the application of proper timing signals to the H-sync, V-sync, red, green and blue signal inputs to the monitor. As a rule, we saw that any declaration at release board is not appealing such as content static. Besides, many papers are utilized to make a declaration at the notice board which is wastage. To experience this issue, Video Graphics Array (VGA) are utilized as a part of this framework to show any content, picture and figure to the screen to make declaration at notice board, for example, screen or Television which is more appealing, for example, content would move be able to with different development. The VGA are utilized as a part of this framework since it has a top notch determination video standard and the capacity to transmit a sharp point by point picture. This VGA is executed in Xilinx Spartan-3E FPGA starter unit board. By utilizing this board, we can undoubtedly plan an advanced framework which depends on FPGA which to acknowledge controller. As we can get the characters show freed of, VGA is ordinarily utilized as a part of PC screens as a standard mechanical show interface. This standard has characterized numerous parameters of VGA, for example, invigorating rates, synchronization flag timing, flag extremity and RGB signals electrical level. The reason that VGA is called Video Graphics Array is screen shows an edge of picture information at last is an exhibit which is made out of M line and N push pixel spot. M×N is determined to the show determination. VGA Controller and PS-2 Console Controller is a strategy used to make VGA show interface. The development of VGA show framework by this strategy can be connected to many events.

2. VGA CONTROLLER

2.1 INTRODUCTION TO VGA

VGA (video graphics array) is a video display standard protocol which is used in almost all the display devices such as CRTs (cathode ray tubes)
and LCDs (liquid crystal display). It gives a simple procedure to interlink a system with a screen for displaying characters or text. VGA has been broadly used as it is standard display in many displaying devices.

2.2 WORKING OF VGA CONTROLLER

Five signals red, green, blue, horizontal synchronization, and vertical synchronization are generally used to control the VGA monitor, the color of a pixel at a fixed location on the screen is controlled by the three RGB color signals. These three color signals are analog signals with voltages ranging from 0.7 to 1.0 volt. By changing the voltage different color intensities are obtained. These three-color signals that is RGB are manipulated as digital signals, so we can just switch each one on or off. The horizontal and vertical synchronization signals are used to handle the timing of the scan rates. The h-sync and v-sync signals are the digital signals as compared to the three color signals. In other words, they take on either at logic high or logic low value. The horizontal deflection circuit in the VGA monitor is controlled by the horizontal synchronization signal which produces different color intensities, due to this the start and end of a line of pixels is correctly displayed on the monitor. The vertical deflection circuit in the VGA monitor is controlled by the vertical synchronization signal, due to this the starting and ending of a frame (of lines) is correctly visible between the top and bottom edges of the displaying space on the monitor screen. In other words, the horizontal synchronization signal gives the value of the time needed to scan a row, while the vertical synchronization signal gives the value of the time needed to scan the entire monitor screen. By manipulating these two sync signals and the three RGB signals, characters are generated on the monitor screen.

In the diagram given below, for section B of the horizontal synchronization signal, 3.77 microsecond is required, which is approximately 95 clock cycles (3.77/0.0397). For section C, 1.79 microsecond is required, which is around 45 clock cycles. Similarly, for section D 640 clock cycles for columns of pixels and 20 clock cycles for section E. The exact number of clock cycles required for each row scanning is around 800 clock cycles. With a 25.175 MHz clock, section D needed exactly 640 cycles, generating the 640 columns per row. When a varying clock speed is used, the resolution obtained will be different. As the vertical sync signal is equivalent to the horizontal sync signal, the same process can be executed as with the horizontal sync regions to obtain the number of cycles needed for each vertical region. However, instead of using the number of periods of a 25.175 MHz clock, the times for each vertical region are multiples of the horizontal cycles. For example, the time for a horizontal cycle is 31.77 microseconds, and section P requires 64 microseconds, which is appropriately two horizontal cycles. Section Q needed 1020 microseconds, which is equal to 32 horizontal cycles. The number of cycles for section R is 480 horizontal cycles.

In the VGA standard specifications, each line in the 640*480 VGA video, 40 cycles of the pixel clock are needed for the back porch, 8 cycles are required for the blank left border of the screen, then comes the 640 cycles for the actual pixels of the video, followed by 8 cycles for the blank right border, 8 cycles for the back porch, and at last 96 cycles are needed in which the horizontal sync signal is active. For one line of 640*480 video a total of 800 cycles of the pixel clock is needed.
2.3 TIMING SPECIFICATION

- Screen refresh rate - 60HZ
- Vertical Refresh - 31.469 kHz
- Pixel frequency - 25.175 MHZ
2.4 CHARACTER GENERATION IN VGA

A character is produced when the electron beam from a cathode ray tube strikes the fluorescent display. The display will be scanned by the electron beam in order to generate a character. Progressive scan and interlaced scan are the two types of the scan modes. The progressive scan starts from the top left corner of the display system and it scans point by point from left to the right of one line. The electron beam moves back to the left start point of the next line and the beam is coordinated by horizontal synchronization signal after one line is scanned. During this process the cathode ray tube blanks the electron beam. The scanning process reaches to the bottom right corner of the display system after scanning the entire screen line by line.

After scanning the whole screen, the beam moves back to the top left corner of the screen and the beam is synchronized by horizontal synchronization signal the cathode ray tube blanks the electron beam. At this point, a character is generated. The interlaced scanning process scans every other line of the monitor screen. Once the screen is scanned the electron beam returns to rest of the lines.

3. PS2 KEYBOARD INTERFACE

This details a PS/2 keyboard interface component for use in CPLDs and FPGAs. The component receives data transactions from a PS/2 keyboard and provides the keyboard make and break codes to user logic over a parallel interface.
The figure conceptually illustrates the PS/2 keyboard interface component’s architecture. The clock and data signals from the keyboard are first synchronized and then debounced. The resultant internal PS/2 data signal is then serially loaded into a shift register on falling edges of the PS/2 clock. An idle counter determines when the transaction is finished, defined by the PS/2 clock remaining at a high logic level for more than 55us, i.e. longer than half of the worst-case PS/2 clock period. Combinational error checking logic verifies the start bit, stop bit, and parity bit with the data. When the PS/2 port is idle and the data is valid, the component outputs the received PS/2 code and sets the new code flag high to indicate that a new code is available on the bus. The code remains available on the bus until another code is received. The new code flag remains high until another PS/2 transaction begins i.e. when a low PS/2 clock signal clears the idle counter.

4. RS-232/UART
RS-232 is an asynchronous serial communication protocol broadly used in electronic systems, by this protocol data is transferred bit by bit from UART transmitter to the UART receiver. This process is asynchronous in nature because there is no other separate synchronizing clock signal as compared to other serial protocols like SPI and I2C. Synchronization is automatically done itself in such type of serial communication protocols. RS-232 cable cab be used to connect the FPGA based projects and the computer systems. A UART can be used to send the serial data between the computer systems and several kinds of peripheral devices such as printers, modems, etc.), which are linked by the RS-232 cable.

RS-232 basically consists of two data wires, one for the transmission process (TX) and other for the reception process (RX). TX is the data line in which data is sent from the transmitting device. RX is the data wire in which other side receiving device put the data on it in order to need to send it back again to the transmitting device.

5. SIMULATIONS AND RESULTS
5.1 RTL SCHEMATIC OF INTERFACING BETWEEN PS2 KEYBOARD AND THE FPGA BOARD

The PS2 interfacing module gets the data and checks the bits. If the trigger is activated, at that point check if the PS2_CLK changed its state. If the state changed and if the clock is on falling edge do add up the data bit that was right now gotten to the previous bits, mark down that one more piece was received. If 11 bits were gotten, trigger out another flag, telling that it completed the process of perusing. Check the equality bit if they got data is correct. If 11 bits were not gotten check in the event that it took more than 4000 times for count reading to number up after it got the past piece and reset everything in the event that it took more. Else skirt this. Another little module extricates the data: Wait for the trigger, which checks if the full pack of 11 bits was received. If there was a blunder in the got bundle, dispose of everything. However remove the DATA bits if the data was correct.

![RTL schematic of interfacing between PS2 keyboard and the FPGA board](image1)

5.2 RTL SCHEMATIC OF INTERFACING BETWEEN VGA CONTROLLER AND THE FPGA BOARD

The Character generation circuitry is designed that took the character from inside of ROM to display the characters on a screen. To show message VGA screen, we sorted out the 640x480 show region into "tiles" where each tile is linked to a character area. In this venture, the text dimension of each character is 16x8 (tallness and Width). This textual style will show 80 content characters in

![Block structure of PS2 keyboard](image2)
each line Mapped onto a 640x480 show (i.e., 640 pixels isolated by 8 segments for every character) and 30 lines (480/16). Each of the 640x480 pixels are related with one of the 80x30 character areas.

Fig: RTL schematic of interfacing between the VGA controller and the FPGA board

Fig: Block structure of interfacing between the VGA controller and the FPGA board

Fig: Simulation waveform of VGA controller interfacing
5.3 RTL SCHEMATIC OF INTERFACING BETWEEN THE UART TRANSMITTER AND THE FPGA BOARD

The UART design is tried and confirmed. This interface utilizes an asynchronous convention. That implies that no clock signal is transmitted along the information. The collector needs to have an approach to time itself to the approaching information bits. On account of RS-232, both side of the link concur ahead of time on the correspondence parameters (speed, format...). That is done physically before correspondence begins. The transmitter sends "idle" ("1") when and the length of the line is idle. The transmitter sends "begin" ("0") before every byte transmitted, so that the collector would figure be able to out that a byte is coming. The 8 bits of the byte information are sent. The transmitter sends "stop" (="1") after every byte.

Fig: RTL schematic UART transmitter

Fig: Simulation waveform of UART TRANSMITTER
CONCLUSION

In this project interfacing of VGA port available on FPGA board to generate the character(s) from ASCII text characters which are given as input through the PS2 keyboard, a different set of protocol has been designed for interfacing the PS2 keyboard with the FPGA board. To display characters on video controller is an important function, to transfer the characters from the FPGA board to the VGA controller screen UART protocol has been designed. The Character generation circuit that took the character from ROM to facilitate the display of text characters on a screen. To display characters on my VGA display, organization of 640*480 display area into tiles where each tile represents a character location. In this project, the font size of each character is 16*8(height & Width). This font will display 80 text characters in each line Mapped onto a 640*480 display (i.e., 640 pixels divided by 8 columns per character) and 30 lines (480/16). Each of the 640*480 pixels in the display are associated with one of the 80*30 character locations. VGA is an important output section for displaying the results. This can be move beyond for interfacing for more interesting real
world implementation to a complete gaming device.

REFERENCES