International Journal of Sciences & Applied Research

www.ijsar.in

FPGA implementation and verification of UART (RS232) to VGA interface with AMBA AHB as system bus

Balasubramanya G. and Manjula R.*

Department of ECE, GITAM, Bangalore, Karnataka, India.
Corresponding author: *Manjula R., Department of ECE, GITAM, Bangalore, Karnataka, India.

Abstract
UART (universal asynchronous receiver and transmitter) is always an 8bit or one byte. So in reality there is no use of one byte data, it should be stream of data. The proposed scheme contains AMBA–AHB (advanced microcontroller bus architecture-advanced high-performance bus) bus protocol it contains on chip communications standard for designing high-performance system.
Now a day's system on chip (SoC) architecture is very famous because with the help of SoC architecture any peripherals can be integrated together very easily, UART contains of three main components namely transmitter, receiver and baud rate generator which are nothing but the frequency divider. UART design will help to reduce area; hence more complex design could be added on the same chip. This type of design architecture will give a great advantage of reusability of modules.
In this project UART is used as an interface between computer and the chip means FPGA 9 field programmable gate array 0, using AMBA-AHB protocol, VGA 9 video graphics array driver is connected to display image in the LCD via UART and verified by simulating the design using modelsim PE student edition 10.2a, also synthesized on FPGA kits such as spartan3E using Xilinx ISE Design suit 14.3. Output is observed and cross verified for different trails.

Keywords: UART, VGA, AMBA-AHB, System Bus

Introduction

A universal asynchronous receiver and transmitter (UART) is an integrated circuit which plays an important role in serial communication. A UART is microchip with programming that controls a computer's interface to its attached serial device. Specifically, it provides the computer with the RS-232C data terminal equipment (DTE) interface so that exchange data with modem and other serial devices.

However, by using a UART controller data can be translated between serial and parallel forms. The UART takes all the bytes of data and sequentially transmit it one bit it a time. At the destination, an additional UART controller will receive the individual bits and finally reassemble the transmission such as the LM3S8962 by taxes instruments have multiple native UART on a board .the UART is a standard communication component that is provided by most of the available microprocessors. IN general, the

number of the UART we will apply the synchronized clock signal to both transmitter and the receiver. The function of the transmitter receiver module is to convert the 8 bit serial data into the single bit data. As we are using the state machines for transmitter and receiver due to which our design become less complex and the proposed UART becomes more stable , reliable and compact for serial data communication.

Objectives

- Design and implementation of UART based on FIFO based module for data transfer.
- Design and implementation of configurable pipelining of UART algorithm which permits both transmitter and receiver.
- Functional verification using verilog and XILINX ISE simulator.
- Implementation using verilog and XILINX ISE simulator.
- Implementation using FPGA kit.
- The major blocks are UART module, interface, and register and VGA driver.
- UART module.

Why UART?

In this project it contains low cost, high speed and wide application. The main application are data transmission system, optical wire communication, notice suppression application, on-chip design etc.

Methodology

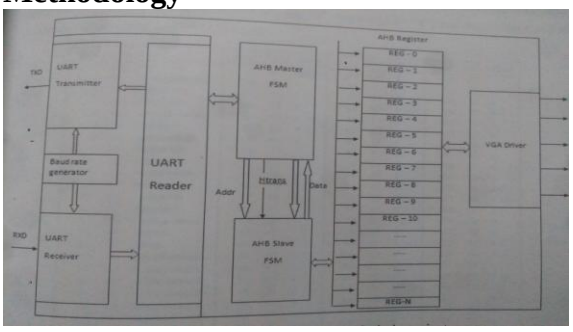

Figure 1 Block diagram of the project.

The major block is UART module, interface, register and VGA Driver. The universal asynchronous receive transmitter (UART) is popular and widely used device for data communication in the field of telecommunication. Mostly used for short-distance, low speed, low-cost data exchange between computer and peripherals. UART is the heart of serial communication. Synchronous communication requires a common clock which should have same phase and frequency for the entire of communication. But it is very difficult to maintain same clock frequency and phase for a longer time to all devices. in asynchronous communication which operates on different clocks where to check the order of communication extra synchronization bits are added.

UARTs are used for asynchronous serial data communication by converting data from parallel to serial at transmitter with some overhead bits shift register as shown in figure 3.1 and vice versa at receiver. All UART design contains a receiver and a transmitter. The receiver performs serial-to-parallel conversion on the asynchronous data frame received from the serial data input. The transmitter performs parallel-to-serial conversion on the 8-bit data receive from the CPU. In order to synchronize the asynchronous serial data and to insure the data integrity. Start parity and bits are added to the serial data.

It has three main components i.e. transmitter, receiver and baud rate ,due to which our design become less complex and the proposed UART becomes more stable, reliable and compact for serial data communication. Due to which, the consumption of LUTs, flip flops or in short the area consumption of the chip becomes less.

It contains

- UART receiver
- UART transmitter
- UART reader

- Interfacing AHB system bus
- VGA driver

UART receiver

This module is responsible for the interface between the computer and the FPGA kit, the hex character pressed on the computer key board will be transmitted to FPGA in ASCII format for example if key pressed is "4" then the equivalent ASCII code is sent as '34"; notice that the pressed key value could be represented in 4 bits foe "3" and 4 bites for "4"; and the protocol will be as shown below along with ASCII chart.

The above figure 3.1 shows the 8 bytes of data is encapsulated with start bit and stop bit; Start bit represented by making TXD signal from computer high to low, like wise stop bit is represented by making TXD signal from computer low to high.

That means that the TXD signal will be always at one when nothing is driving on it .So receiver will wait for only RXD signal which is TXD from computer to go low and then next 8 bites are treated as data from computer.

UART transmitter

This module is responsible for the interface between the FPGA kit and computer, all the data content stored in FPGA will be in the form of hex, where the computer's UART can't understand, so all the hex data will be converted back from hex to ASCII, and also UART can handle only 8 bit of data at any time so each character in hex is treated as 8 bit; so 8bit of data to be transmitted from FPGA is treated as 16 bit due to addition of ASCII code. So will make use of parallel in and serial out module, in which the parallel input will be 8 bit and then it will be passed to HEX_ASCII converter where it will be converted to 16 bit and data is sent on the TXD line of FPGA.

UART reader

UART is will only transmit or receive 8 bit data in which only one hex charter is receive and also there will be no difference between address and data, as a reason a data reader is built using verilog called as UART reader; the major task is to just separate the address and data according to the syntax.

Syntax to write a data to a particular address is; "w,a=0,d=a2," here the 'w" indicates write mode and "a" address and "0" address content, 'd" data '2a" is the content of data. Likewise for read; "r,a=0"here"r" indicates read mode, "a" address and "0' address content. Then the UART controller will generate address and data that to be written to locations of registers through AHB system bus. Any register on SoC should be molded as shown above. It should contain a clock and reset as global signals, chip select (c_s) which is the higest priority signals after rest signal, address bus (addr), write data bus (w data), read data bus (rdata), write/read signal (RW) which will diced weather to write or read.

The register may have n-number of internal register (reg0, reg1 , reg2, reg3) which may be made output or input, also the width of the internal register need not to be same size as in memory module. the internal register are bit addressable but in memory it is always byte and multiples of byte addressable. This provides the flexibility to design.Once design is built; connecting it to the controller or processor is one of the major challenges; so to overcome this system on chip architecture is implemented, AHB buses may be used; which are defined by ARM buses.The AMBA specification defines as on-chip communication standard fo designing high- performance embedded microcontrollers.

AMBA specification

Conceptual view of the AMBA specification

The advanced microcontroller bus architecture (AMBA) specification defines

an on chip communications standard for designing high-performance embedded microcontrollers.

Three distinct buses are defined within the AMBA specification:

- The advanced high-performance bus (AHB)
- The advanced system bus (ASB)
- The advanced peripheral bus (APB)

A test methodology is included with the AMBA specification which provides an infrastructure for modular macro cell test and diagnostic access.

VGA driver

The VGA consists of seven subs -system, including: graphics controller, display memory, serialize, attribute controller, sequencer and CRT controller. Basically, the CUP performs most of the work, feeding pixel land text information to the VGA.

- Graphics controller: can perform logical function on data being written to display memory.
- Display memory: a bank of 256K DRAM divided into 4 64k color planes.
- Serialize; and convert it to a serial bit stream which is sent to the attribute controller.
- Attribute controller: what color will be displayed for a given pixel value in display memory.
- Sequencer: controls timing of the board and enables/disables color planes. CRT controller: generates syncing and blanking signals to control the monitor display.

Hardware requirements

Hardware: SPARTAN 3 FPGA

The Spartan 3E starter kit board highlights the unique features of the Spartan 3E FPGA family and provides a convenient development board for embedded processing applications.

Software requirements

Simulation Tool: ModelSim 6.4c

ModelSim SE is our UNIX, Linux and Windows-based simulation and debug environment, combining high performance with the most powerful and intuitive GUI in the industry.

Synthesis tool: XILINX ISE

Domain specific platform targets one of the three primary Xilinx FPGA user profiles (domains).

- The embedded processing developer.
- The digital signal processing (DSP) developer.
- Or the logic/connectivity developer.

Every element in this platform is tested, targeted, and supported by Xilinx. The tools then apply specific optimizations to help meet the design goal. In addition, the ISE design suite boasts substantially faster place-and-route and simulation run times, providing users with 2 xs faster compile times. Finally, Xilinx has adopted the FLEX net licensing strategy that provides a floating license to track and monitor usage.

Implementation and simulation

Verilog HDL is a hardware description language (HDL). A hardware description language is a language used to describe a digital system. For example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an integrated circuit(IC) chip, and the switch level. Or, it might higher level describes and the transfers of vector of information between them.

Conclusion

The UART protocol was successfully designed is verilog and implication in connecting devices. as a reason it has only two major pins as TXD and RXD which are responsible for transmitting and receiving

serial data respectably. It is very important to communicate with internal circuitry. AMBA-AHB which is standard SoC bus protocol interfaced with the UART because high-performance, high clock frequency system modules as a reason proposed UART architecture can be smoothly connected to any SoC modules. For displaying interfaced VGA with UART via AHB. The image in the form of hexadecimal was successfully transmitted on UART and displayed on the VGA.

References

1. L.K.Hu and Q.CH wang,"UART –Based reliable Communication and performance analysis", Computer Engineering Vol 32 No.10,May 2006,pp5-21.
2. Norhuzaimin, J., Maimun, H.H. The Design of High Speed UART, Asia Pacific Conference on Applied Electromagnetics (APACE 2005), Dec. 2005
3. Parul Sharma and Ashutosh Gupta, Design Implementation of Highly efficient UART (January 10, 2010), Journal of Science and Technology, Vol.5, No.4, pp.21-30, December 2009.