

Design of UART (Universal Asynchronous Receiver Transmitter) using VHDL

¹Ananya Chakraborty, ²Surbhi, ³Sukanya Gupta, ⁴Swati Deshkar, ⁵Pradeep Kumar Jaisal

^{1, 2, 3, 4, 5}Dept. of Electronics & Telecommunication, SSIPMT, Raipur, CG, India

Abstract

UART- Universal Asynchronous Receiver Transmitter, generally it is used for better transmission of serial data that is it either transmit or receives data serially with the help of shift register. It consist frame format, one start bit (usually low), 5-8 data bit, one optional parity bit and one stop bit (opposite polarity of start bit). Asynchronous means by using start and stop bit be transmit data, there is no need of sending (PAD) that is ASCII (SYN) for synchronizing transmitter and receiver. It transmit 9600 to 38400 bps for transmitting data bit. Whole process of serial transmission is based upon the principle of shift register.

Keywords

UART, VHDL

I. Introduction

The Universal Asynchronous Receiver Transmitter (UART) is a popular and widely-used device for data communication in the field of telecommunication. There are different versions of UARTs in the industry. (UART) An integrated circuit used for serial communications, containing a transmitter (parallel-to-serial converter) and a receiver (serial-to-parallel converter), each clocked separately. The parallel side of a UART's usually connected to the bus of a computer. When the computer writes a byte to the UART's Transmit Data Register (TDR), the UARTs will start to transmit it on the serial line. The UART's status register contains a flag bit which the computer can read to see if the UART is ready to transmit another byte. Another status register bit says whether the UART has received a byte from the serial line, in which case the computer should read it from the Receive Data Register (RDR). If another byte is received before the previous one is read, the UARTs will signal an "overrun" error via another status bit. The UARTs may be set up to interrupt the computer when data is received or when ready to transmit more data. The UART's serial connections usually go via separate line driver and line receiver integrated circuits which provide the power and voltages required to drive the serial line and give some protection against noise on the line. Data on the serial line is formatted by the UART according to the setting of the UART's control register. This may also determine the transmit and receive baud rates if the UART contains its own clock circuits or "baud rate generators". If incorrectly formatted data is received the UART may signal a "framing error" or "parity error". Often the clock will run at 16 times the baud rate (bits per second) to allow the receiver to do centre sampling - i.e. to read each bit in the middle of its allotted time period. This makes the UART more tolerant to variations in the clock rate ("jitter") of the incoming data. An example of a late 1980s UART was the Intel 8450. In the 1990s, newer UARTs were developed with on-chip buffer space for data. This allowed higher transmission speed without data loss and without requiring such frequent attention from the computer. For example, the Intel 16550 has a 16 byte FIFO. A UART (Universal Asynchronous Receiver/Transmitter) is the microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232C Data

Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with modems and other serial devices. As part of this interface, the UART also converts the bytes it receives from the computer along parallel circuits into a single serial bit stream for outbound transmission. On inbound transmission, it converts the serial bit stream into the bytes that the computer handles. It adds a parity bit (if it's been selected) on outbound transmissions and checks the parity of incoming bytes (if selected) and discards the parity bit ,adds start and stop delineators on outbound and strips them from inbound transmissions handles interrupt from the keyboard and mouse (which are serial devices with special port s),it may handle other kinds of interrupt and device management that require coordinating the computer's speed of operation with device speeds. More advanced UARTs provide some amount of buffering of data so that the computer and serial devices data streams remain coordinated. The most recent UART, the 16550, has a 16-byte buffer that can get filled before the computer's processor needs to handle the data. The original UART was the 8250.

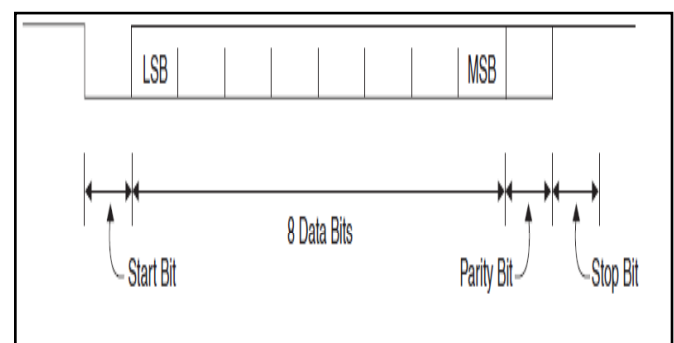


Fig. 1: UART Frame Format: (1 Start Bit, 8 Data Bits, 1 Parity Bit, 1 Stop Bit)

A. Main features of the 16550 include

1. The ability to convert data from serial to parallel, and from parallel to serial, using shift registers.
2. An on-chip bit rate (baud rate) generator to control transmits and receive data rate.
3. Handshake lines for control of an external modem, controllable by software.
4. An interrupt function to the host microprocessor.
5. An on-chip FIFO buffer for both incoming and outgoing data; this gives the host system more time to respond to an interrupt generated by the UART, without loss of data.

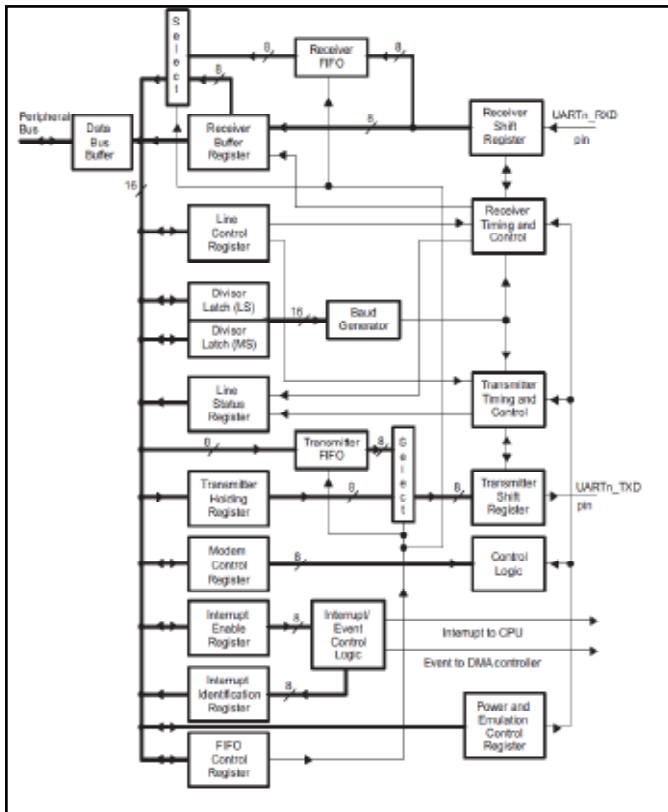


Fig. 2: Detailed Functional Block diagram

B. Features

1. Functionally compatible with the NS16450 UART
2. Faster performance than industry standard hardwired devices
3. Inserts or extracts standard asynchronous communication bits (start, stop and parity) to or from the serial data
4. Holding and shifting registers eliminate the need for precise synchronization between the CPU and serial data
5. Standard CPU Interface
6. Separate interrupt lines for data received (RxRdyn) and data transmitted (TxRdy). A common interrupt line for all internal UART data and error events. Interrupt conditions include: receiver line errors, receiver buffer available, transmit buffer empty and when modem status flag change is detected.
7. Fully-prioritized interrupt system control
8. MODEM interface functions (CTS, RTS, DSR, DTR, RI and DCD)
9. Fully programmable serial interface characteristics – 5, 6, 7 or 8-bit characters
10. Interactive control signaling and status reporting capabilities
11. Separate input and output data buses for use as an embedded module in a larger design
12. Transmitter enabled by new data writes to transmit holding register
13. Receiver synchronizes off the start bit
14. Receiver samples all incoming bits at the center of each bit – Even, odd, or no-parity bit generation and detection – 1, 1.5 or 2-stop bit generation and detection
15. False start bit detection
16. Line breaks generation and detection

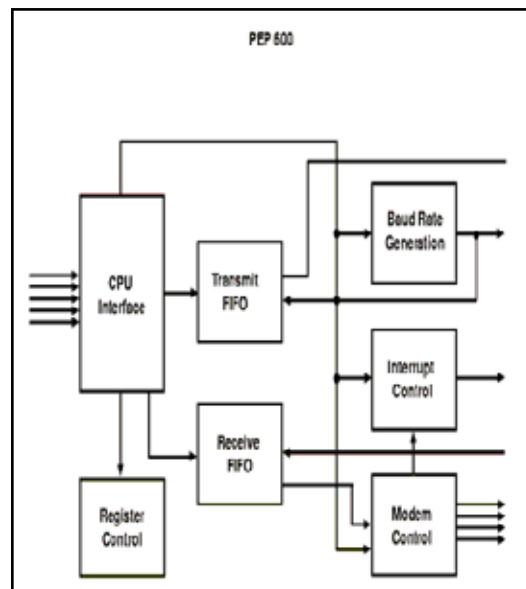
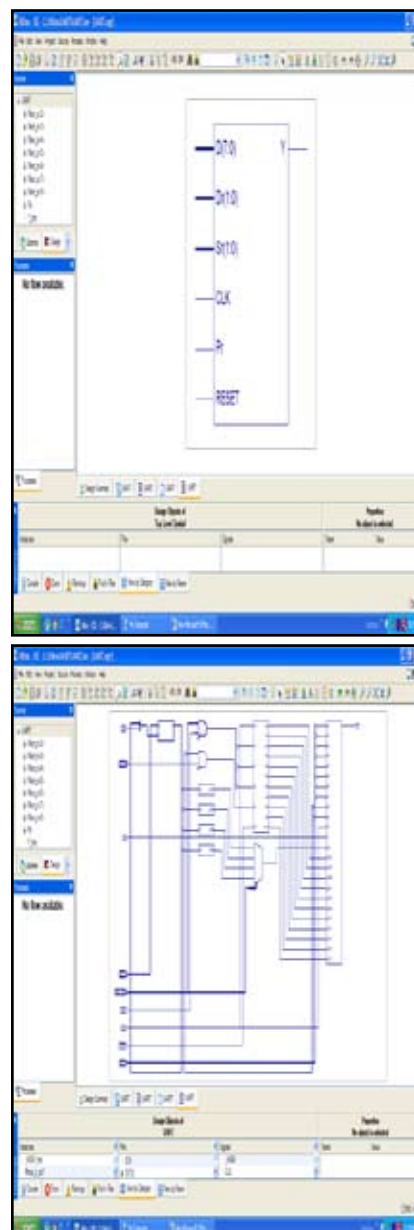


Fig. 3: UART Block Diagram

C. Result Obtained



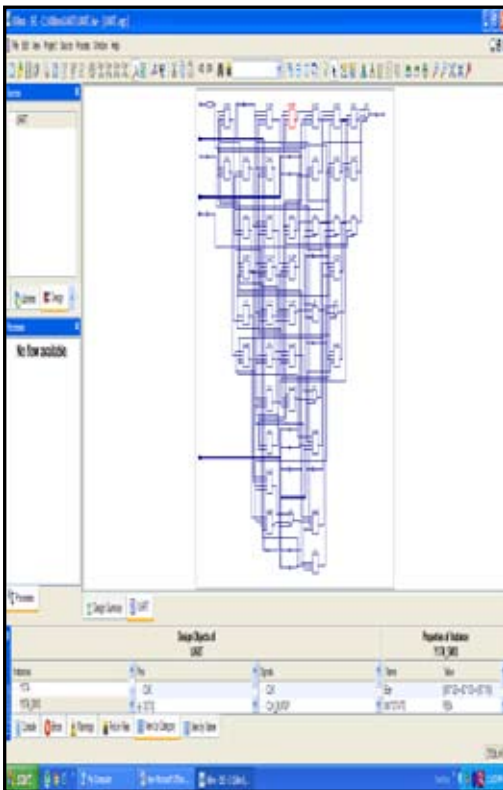


Fig. 4: Results of UART

REFERENCES

- [1] S. Wang, "Generation of low power dissipation and high fault coverage patterns for scan-based BIST", in Proceedings of International Test Conference, 2002, pp. 834–843.
- [2] M. Ibrahim Abu-Bakr, "A Built in Self Testable Bit-Slice Processor", Faculty of Computer Science & Information Technology, University of Malaya, May 1995.
- [3] J. Turino, "RTL DFT Rule Checking—The Circuit Designer's Secret Weapon", Integrated System Design Magazine, 2000.
- [4] A. P. Stroele, H. J. Wunderlich, "Hardware-Optimal Test Register Insertion", in IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems, June 1998, Vol. 17, No. 6, pp. 531-539.
- [5] Z. Navabi, "VHDL Analysis and Modeling of Digital Systems", McGraw-Hill Inc., 1991.
- [6] M. S. Michael, "A Comparison of the INS8250, NS16450 and NS16550AF Series of UARTs", National Semiconductor Application Note 493, 1989.
- [6] "PC16550D Universal Asynchronous Receiver/Transmitter with FIFOs", National Semiconductor Application Note, 1995.
- [7] M. S. Harvey, "Generic UART Manual", SiliconValley, 1999.
- [8] O. A. Petlin, S. B. Furber, "Built-In-Self-Testing of Micro pipelines", in Advanced Research in Asynchronous Circuits and Systems, IEEE, 1997, pp. 22-29.
- [9] J. Turino, "RTL DFT Rule Checking—The Circuit Designer's Secret Weapon", Integrated System Design Magazine, 2000.
- [10] C. H. Roth, "Digital System Design Using VHDL", PWS Publishing Company, 1998.
- [11] [Online] Available: <http://www.xess.com>



Ananya Chakraborty, student of final year of Electronics and Telecommunication branch in Shri Shankaracharya Institute of Professional Management and Technology



Sukanya Gupta, student of final year of Electronics and Telecommunication branch in Shri Shankaracharya Institute of Professional Management and Technology



Surbhi, student of final year of Electronics and Telecommunication branch in Shri Shankaracharya Institute of Professional Management and Technology



Swati Deshkar, student of final year of Electronics and Telecommunication branch in Shri Shankaracharya Institute of Professional Management and Technology



Pradeep Kumar Jaisal, working as Assistant Professor, Dept of ETC at Shri Sankaracharya Institute of Professional Management & Technology (SSIPMT), & he is a Research Scholar, Department of Electronics & Telecommunication, Rungta College of Engineering & Technology, Bhilai, CSVTU University Bhilai, Chhattisgarh, India. He has received his M.Tech Degree in Digital Electronics from ETC Dept, Rungta College of Engineering & Technology, Bhilai, CSVTU University, Bhilai, Chhattisgarh, India. His main research includes. Design of Time Triggered Shared Bus Architecture Using VHDL, Design and Simulation of 8255 PPI Adapter Using VHDL. He has published 04 papers in reputed national & international Journals. He has participated in 1 conferences.