A New Approach to Realize UART

Yongcheng Wang , Kefei Song
Changchun Institute of Optics, Fine Mechanics and Physics
Chinese Academy of Sciences
Changchun ,China
wyc_dyy@sina.com
songkefei@sina.com

***Abstract*—In order to connect DSP which has synchronous serial ports to the devices implementing asynchronous communications protocol, a method to implement UART communications based on programmable logic device is proposed in the paper. In the proposed method, the core function of UART is integrated in CPLD with VHDL. Firstly, UART data frame format and operational principle of UART were introduced after reviewing some methods to realize UART. The methods to implement UART transmitter, UART receiver and baudrate generator using VHDL were illustrated in detail. Then pre-simulation and synthesize of VHDL program were executed. Finally, the test with bit error rate was carried out on physical system. Experimental results indicate that 75 percent of the GLB are used by UART, and the bit error rate is less than 10^{-9}. The experiment was implemented utilizing the RS-422 protocol and the baudrate is 62.5kb/s. The proposed method can satisfy the system requirements of high integration, stabilization, low bit error rate, strong anti-jamming and low cost.**

Keywords- UART; programmable logic device ; VHDL;DSP ; simulation

I. INTRODUCTION

UART, of which the full name is universal asynchronous receiver and transmitter, is often used in embedded systems. It is also widely used serial data transmission protocol, as it can realize the transmission of the information system control or low-speed data[1]. At present, more and more RS-232, RS-422 and RS-485 serial bus interface standards are applied by the corresponding UART interface chip. However, a lot of the digital signal processors only have a synchronous serial port, such as TMS320C3x, TMS320C54x, ADSP-2106x family of DSP. These digital signal processors could not directly communicate with the equipment which uses asynchronous communication protocol, which brings a lot of inconvenience. There are usually two ways to solve the problem: one is to implement UART using DSP synchronous serial port interface, and the other is using UART controller chips to realize the synchronous serial communication [2-4]. The first method required many software resources, resulting in weak portability. The second method required UART chip and other interface circuits, which will introduce an increase of the complexity of circuit implementation, and control the UART chips, to make the implementation less flexibility.

A new approach to realize the UART with CPLD, FPGA and other programmable logic device widely used is supplied. The transmitter, receiver and other core functions are integrated by VHDL. It can not only avoid the above-mentioned shortcomings of traditional methods, but also make the whole design more compact, stable and reliable. The most important is that the use of programmable logic devices to implement UART can be designed and changed in accordance with the actual needs, which makes the design more flexible, which is a dedicated control chip unmatched[5-7].

In this paper, to, the core components of the method implementing UART with VHDL are realized with state machine, avoiding the emergency of the error state. All modules are simulated by MAXplusII, and simulation results show that the data frame format and protocol compliance is within the design expectations. The program is compiled and downloaded, and the bit error rate test is also done. The results show that the method is flexible, stable, reliable, low error rate, easy to control, and low cost.

II. UART DATA FRAME FORMAT AND ITS WORKING PRINCIPLE

A. UART data transmission protocol

UART transmit and receive data format shown in Figure 1, usually include start bit, data bit, parity bit,stop bit and idle state. Start bit is the beginning of data transmission. When the transmitter sends a character data, a logic "0" signal is firstly send, which is the start bit, and the time width is a baud rate clock cycle. Start bit is followed by data bits, which are usually from 5 to 8 bits. The data bit from the least significant bit (LSB) begin to send. The data bits can be parity bit, which can be odd or even parity, as well as no parity bit. The parity bit or data bit (when no parity bit) is followed by stop bits, which are logic "1" signal containing 1,1.5 or 2 bits. Stop bits are the end of a data. Idle state is a logic "1".This data frame format is adopted by the start bit and stop bit to achieve character synchronization. UART has an internal configuration register, in which user can set the

data bits, whether there is parity bit, as well as the type of parity and stop bits[7].

Figure 1. UART data frame format

B. *The working principle of UART*

UART widely used in serial data transmission protocol, usually includes the transmitter, receiver and a baud rate generator. The transmitter performs parallel-to-serial conversion, the receiver performs serial-to-parallel conversion, and the baud rate generator generates the required clock signal. The serial transmitter section consists of an 8-bit Transmitter Hold Register (THR) and Transmitter Shift Register (TSR). Transmit operation as shown in Figure 2. Parallel data is stored in THR which received from the CPU, then it is transferred to the shift register and send out in serial data. At the same time parity bit is generated and transmitted by TSR, when the whole character is removed from the TSR, CPU interrupt signal is generated.

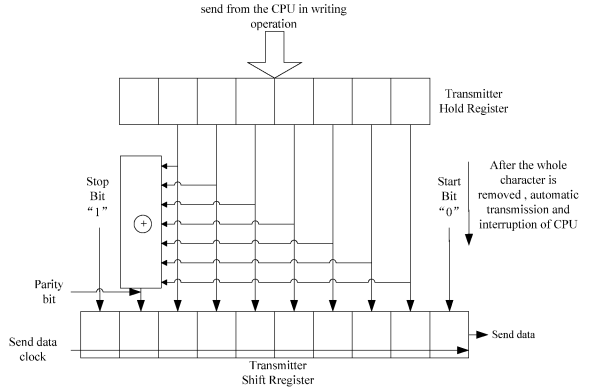

Figure 2. UART transmitter

The serial receiver section also contains an 8-bit Receiver Buffer Register (RBR) and Receiver Shift Register (RSR). Serial data received is stored in the RSR, when the RSR receive the whole character, automatically sent to RBR, status register will be set and generate CPU interrupt signal, parallel data will be transmitted to the CPU in the read command of the CPU, serial-to-parallel conversion is performed. receive operation as shown in Figure 3:

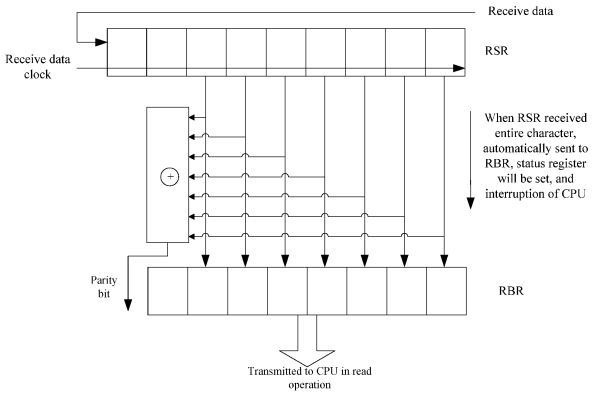

Figure 3. UART receiver

III. USING VHDL TO REALIZE THE UART

According to the working principle of UART in the design, UART is composed of the transmitter, receiver, baud rate generator and data latches four parts, and its block diagram is shown in Figure 4. The data is latched by flip-latch, and the input and output data is controlled through the door control and output enable pins. When the door control signal is effectively, the data is latched into the data device and when the output enable signal is effectively, data is output. The baud rate clock of which the baud rate of 16 times of baud rate clock frequency are generated by the baud rate generator. The transmitter performs parallel-to-serial conversion and the receiver performs serial-to-parallel conversions, which are realized by state machine. In Fig 4, the input clock signal generated by an external crystal, the data bus D0 ~ D7, reset signals and other control signals are connected with DSP, TXD and RXD signals are connected to corresponding pins of RS-422 interface chip. Interrupt signals are connected to external interrupt pin of DSP.

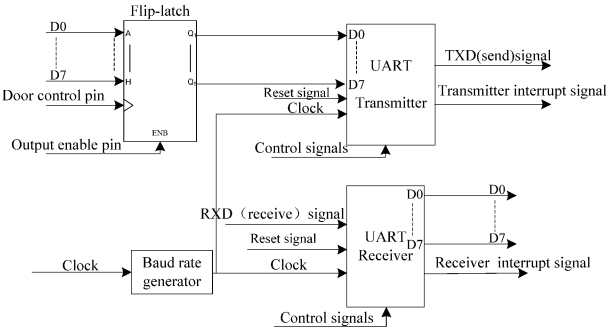

Figure 4. The structure diagram of UART

A. *Transmitter*

The transmitter is realized by state machine as shown in figure 5. There are 6 states in the state machine. When the UART is reset by reset pin, the transmitter will be reset to start state, in which the transmitter will wait untill the Start bit is asserted. A Start bit will be asserted as soon as “THR” is not empty. Once a low SOUT (Start bit) is asserted, the FSM will switch to shift state. When the FSM is in shift state,

it's waiting for the last (most significant) Data bit to be shifted out. After the last Data bit is shifted out, the FSM will switch to parity state if parity is enabled. Otherwise, it will switch to stop_1bit state. When the FSM is in this state, the last Data bit is still in transmission. When the transmission is completed, the FSM will assert the Parity bit. Once the Parity bit is asserted, the FSM switch to the stop_1bit state. No matter if the Stop bit is configured to be 1, 1.5 or 2 bits long, the FSM will always switch to this state, wait for a baud clock cycle, and then assert the Stop bit(s). For 1 Stop bit, the FSM switches back to start state and waits to assert the Start bit of another frame. For 1.5 Stop bit, it switches to stop_half bit state and stays there for just half baud clock cycle before switching to start state. For 2 Stop bits, it switches to stop_2bit state then switches back to start state. Note that the Stop bit(s) is asserted at the time when the FSM is leaving the stop_1_bit state. Stop_halfbit state is for 5-bit Data bits with 1.5 Stop bit. The FSM will stay in this state for only half baud clock cycle and then switch to start state. When the FSM is in stop_2bit state, the first Stop bit is in transmission. It waits for a baud clock cycle, then asserts the second Stop bit and switches to the start state. The stop bit is the ending of a data, when the stop bit is asserted, an interrupt signal will be sent, and the pin will be low for one baud rate clock cycle.

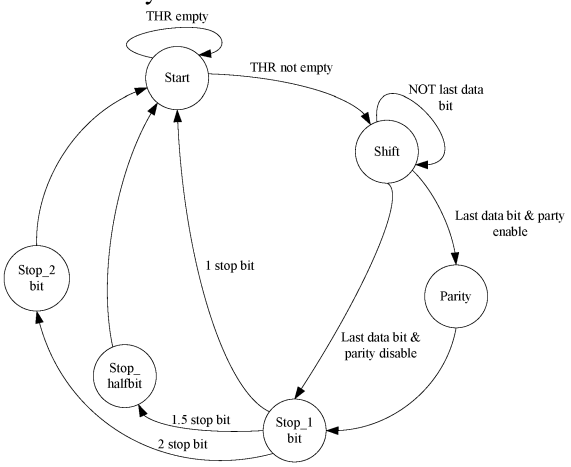

Figure 5. Transmitter state machine

The simulation waveform of the transmitter in MAXplusII is shown in figure 6. When system is powered on Reset will go high active for some times and then go low. “ThrWRn_re” is the baud rate clock, of which the frequency is 1/16 times of the system clock “CLK16”. Parity is disabled, and DataBit0 and DataBit1 are configured as “1” indicating that the data is 8-bit data. “Cs” is chip select signal asserted as low active. “Tint” is transmitter interrupt signal, “Sout” is the serial output signal and “THR” is Transmitter Hold Register. The data are 47H, 48H and 49H in simulation. “TxCNT_r” is transmit count register, “TSR” is Transmitter Shift Register and “Tx_State” is Transmitter state signal. As can be seen from the simulation waveform, when THR is 47H, the binary data "0111000101" will be appeared in SOUT. According to the UART data frame format decoding the data , which is 47H when stop bit is asserted, “Tint” will go low for one CLX16 cycle. When THR is 48H, the binary data "0000100101" will be appeared in SOUT. According to the UART data frame format the data is decoded, to be 48H. From the simulation results it can see that the serial output data is in full compliance with the UART data frame format, and the Output data and input data are exactly the same.

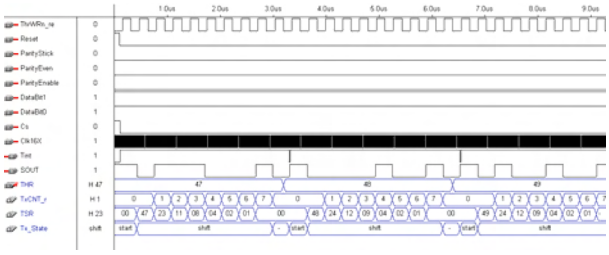

Figure 6. Simulation waveform of the transmitter

B. *Receiver*

The receiver is realized by state machine which has 4 states is shown in figure 7. When the start bit is not detected, the FSM is in idle state. In this state, it's waiting for SIN to change from high to low and stay low for 8 Clk16X clocks, which is considered as a valid Start bit. Once a valid Start bit is detected, the FSM will switch to shift state. When the FSM is in shift state, it waits 16 Clk16X clocks for each Data bit to be shifted into RSR. After the last Data bit is shifted in, the FSM will switch to parity state if parity is enabled. Otherwise, it will switch to stop state. When the FSM is in parity state, it will wait for 16 Clk16X clocks and then sample the Parity bit. Once the Parity bit is sampled, the FSM will switch to the stop state. No matter the Stop bit length is configured to be 1, 1.5 or 2 bits long, the FSM will always wait for 16 Clk16X clocks and then sample the Stop bit. As long as logic “1” is sampled at the Stop bit, the Framing error flag (FE) in LSR will not be set. The receiver does not check whether the Stop bit is in the right length as configured. The FSM will switch back to idle state after the Stop bit sampled.

Since the serial frame is asynchronous in the receiving clock, a high to low transition of “SIN” pin will be treated as the Start bit of a frame. However, in order to avoid receiving a incorrect data due to “SIN” signal noise, the False Start Bit Detection feature is implemented in the design, which requires the Start bit to be low at least 50% of the receiving baud rate clock cycle. Since the internal clock Clk16X is as 16 times as the receiving/transmitting baud rate clock frequency, the Start bit needs to be low at least 8 Clk16X clocks, which can be considered as a valid Start bit. Once a valid 8 Clk16X clocks Start bit is received, the Data bits and Parity bit will be sampled every 16 Clk16X clocks (the receiving baud rate). If the Start bit is exactly 16 Clk16X clocks long, each of the following bits will be sampled at the center of the bit itself. When the data is available in RBR, the Receiver Data available flag will be set to logic 1 to inform the CPU that the data is ready to be read.

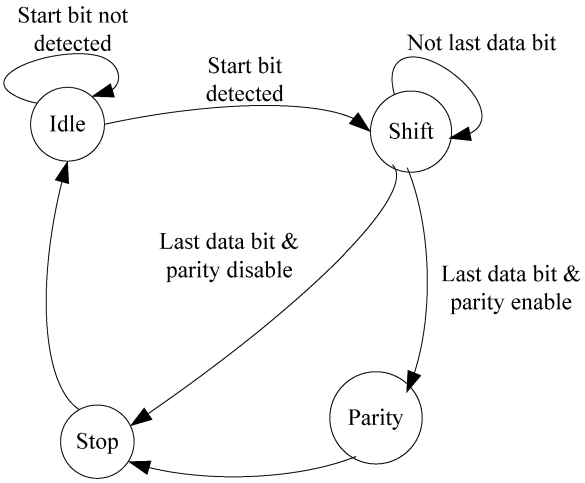

Figure 7. Receiver state machine

The simulation waveform of the receiver in MAXplusII is shown in figure 8. After system is powered on, "Reset" will go high active for some times and then go low. "SIN" is serial input signal, parity is disabled, DataBit0 and DataBit1 are configured 1 indicating that the data is 8-bit data. RBR is Receiver Buffer Register, Rint is receiver interrupt signal. Oe is output enable signal, low active, Cs is chip select signal, low active. CNT_r is a 4 bits counter. In simulation, according to the format of the UART, "SIN" is configured as "0101010101", which can be expressed by hexadecimal number 55H. From the simulation waveform we can see that the signal "RBR" is 55H, and then "SIN" is configured as "0001011101", which is expressed by a hexadecimal number 74H. Then signal "RBR" becomes 74H. When the stop bit is asserted, an interrupt signal will be asserted and the pin will be low for one baud rate clock cycle. From the results we can see the data in "RBR" is correct.

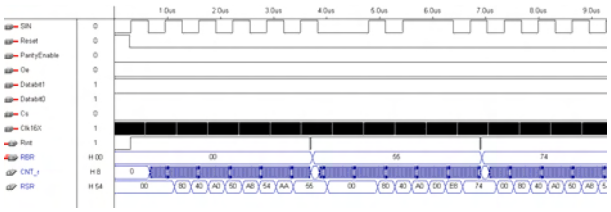

Figure 8. Simulation waveform of the receiver

C. *Baud rate generator*

Baud rate generator generates the baud rate clock and "clkx" is 16 times the baud rate clock frequency.

IV. SYNTHESIZE AND TEST

After the UART is designed, the program is synthesized using the software ispDesignEXPERT8.4, and then downloaded to ispLSI1032-60LG for hardware emulation. The program used 24 Generic Logic Blocks, which was 75% of the ispLSI1032-60LG all GLB, showing that the method occupying very few resources. Then test was made by experiments. Figure 9 shows the Structure of test system, in which the address bus of the DSP was decoded to select UART, and control bus includes read/write and strb signals. RS-422 communication terminal equipment is designed to communicate with CPLD, display and analyze data. In this system, baud rate is 62.5kb/s, data is configured as 8-bit data format, parity disable and stop bit length is 1 bit. RS-422 communication cables using twisted-pair shielded is 3m long. Test results show that the bit error rate less than 10^{-9}.

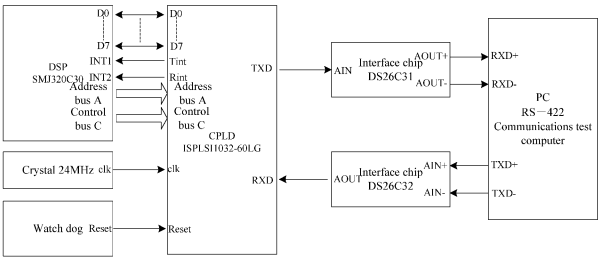

Figure 9. The structure diagram of test system

V. CONCLUSION

In this paper, a new approach to realize UART using programmable logic device is introduced. First of all, the working principle of the UART is described. Then all modules of UART are designed and simulated using VHDL. Finally, the UART is synthesized and downloaded to CPLD, and bit error rate test is made. The test results show that the bit error rate is less than 10^{-9}. The method using programmable logic device to realize UART is flexible, stable, low bit error rate and low cost.

REFERENCES

[1] Andreas Dannenberg, "UART Implementation Using the High-End Timer ," Texas Instruments,2006.

[2] Analog Devices DSP Applications group, " Analog Devices Serial Port Development and Troubleshooting Guide ," Analog Device Inc,1999.

[3] Lawrence Wong, "Hardware UART for the TMS320C3x ," Texas Instruments,1993.

[4] "ADSP-2106x SHARC® Processor User ' s Manual ," Analog Device Inc, 2004.

[5] XU L Y, "Realization of UART Communication Based on FPGA ," Microcomputer Information,vol. 23(35), 2007, pp. 218-219.

[6] QIAO SH J,HU Y P,GAO Y, " An Image Processing System and Its FPGA Implementation ," Chinese Journal of Electron Devices Instrument, vol. 29(3) ,2006, pp. 825-828.

[7] TANG SH W, "Design and Test of the Multiple Serial Ports Extension System Based on CPLD ," Chinese Journal of Electron Devices Instrument, vol. 29(3), 2006, pp. 981-984.