Design And Implementation Of USART IP Soft Core Based On DMA Mode

Peddaraju Allam[1]

[1]M.Tech Student, Dept of ECE, Geethanjali College of Engineering & Technology, Hyderabad, A.P, India.

***Abstract*— A Universal synchronous Asynchronous Receiver/Transmitter is a type of "synchronous asynchronous receiver/transmitter", a piece of computer hardware that translates data between parallel and serial forms. The universal designation indicates that the data format and transmission speeds are configurable and that the actual electric signaling levels and typically is handled by a special driver circuit external to the USART. A USART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. USARTs are now commonly included in microcontrollers. A dual USART, or DUART, combines two USARTs into a single chip. Many modern ICs now come with a UART that can also communicate synchronously; these devices are called USARTs (universal synchronous/asynchronous receiver/transmitter).The USART IP hard core is poor at flexibility and transportability while USART IP soft core is only based on poll and interrupt mode at present which consumes so much time of CPU that the performance of embedded system is reduced greatly. USART (with reference of clock) IP soft core based on DMA mode is proposed and well elaborated using the characteristic of DMA. The IP core is AVALON bus-compatible with the control and arithmetic logic of entire IP core completed by a single FPGA chip so that it is very suited to NIOSII embedded system. Five main sub modules are well designed and the whole IP core is tested and verified in a simple NIOSII embedded hardware system. It turns out that USART IP soft core based on DMA mode can reduce elapsed time of CPU greatly in data transmission process so that the performance of NIOSII system can be improved and design requirement can be better met with less resources occupied, high speed, high flexibility and high transportability.**

Keywords- NIOSII; USART; IP; DMA; AVALON bus

I.INTRODUCTION

The Universal Asynchronous Receiver Transmitter (USART) is a popular and widely-used device for data communication in the field of telecommunication. There are different versions of USARTs in the industry. Some of them contain FIFOs for the receiver/transmitter data buffering and some of them have the 9 Data bits mode (Start bit + 9 Data bits + Parity + Stop bits). This application note describes a fully configurable USART optimized for and implemented in a variety of Lattice devices, which have superior performance and architecture compared to existing semiconductor ASSPs (application-specific standard products USART is the most basic and most commonly used method of communication in the embedded system, whose performance will to some extent determine whether the overall system can meet the design requirements. The implementation of USART basically uses the on-chip USART IP hard core, including SCM (such as STM32 [1]) or ARM (such as S3C44B0 [2]) currently. It makes the design of the whole system has great limitations for the parameters of which is solidify already on the chip(that they cannot be changed anymore)and which combine with the other on-chip peripherals that cannot be separated, although the performance is high. Because of the design beyond change, the poor flexibility, the small application, and the poor transportability, it's usually unable to meet the high requirements of the customer.

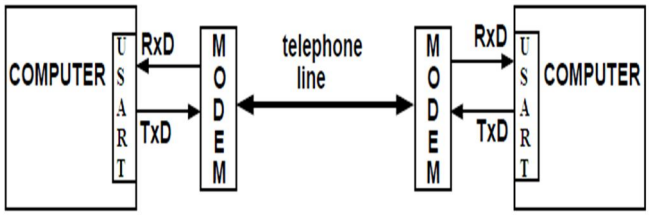

Figure1. The Basic USART Communication

With the rapid development of FPGA (field programmable gate array) and SOPC (system on a programmable chip), soft core plays an increasingly important role in embedded system depending on the high performance, high flexibility, transportability and configuration. Some companies (such as Altera) have provided USART IP (intellectual property) soft core, which only supports the poll and interrupt mode [3] currently. Since the two kinds of transmission will often interrupt the operation of CPU during the data transmission process, especially when transmitting large data, it will occupy a lot of time of CPU, thus greatly reducing the performance of the overall NIOSII system.

Direct memory access (**DMA**) is a feature of modern computers that allows certain hardware subsystems within the computer to access system memory independently of the central processing unit (CPU).Without DMA, when the CPU is

using programmed input/output, it is typically fully occupied for the entire duration of the read or write operation, and is thus unavailable to perform other work. With DMA, the CPU initiates the transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller when the operation is done. This feature is useful any time the CPU cannot keep up with the rate of data transfer, or where the CPU needs to perform useful work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards. DMA is also used for intra-chip data transfer in multi-core processors. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without a DMA channel. Similarly, a processing element inside a multi-core processor can transfer data to and from its local memory without occupying its processor time, allowing computation and data transfer to proceed in parallel.

A brief introduction firstly is made about the overall architecture of DMA mode USART IP soft core in this article, and then it focuses all the attention on the design of all the sub- modules of the entire IP core, whose realization using Verilog HDL language is given. Then it makes a hardware validation about the function of the entire IP. At last, comparing the USART IP core of the three different forms of transmission, we make summarization about the design method and characteristics of the USART IP soft core with the DMA mode. The whole system completes the hardware verification on the Altera's FPGA chips Cyclone II EP2C35F672C8 and eventually is applied to a USART communication system.

II.THE OVERALL FRAMEWORK OF USART IP SOFT CORE IN DMA MODE

USART IP soft core is designed using DMA transmission here. Its overall architecture is shown in Figure 1. The entire USART IP soft core in DMA mode mainly includes the following 5 sub-modules: USART send controller, USART Receive controller, Register file with the Interface of Avalon-MM Slave, Master Read type DMA controller with the interface of Avalon-MM Master and Master Write type DMA controller with the interface of Avalon-MM Master. When the NIOSII processor sends data through serial port, firstly, it's necessary to make configuration to the USART sent controller and the Master Read type DMA controller through the register file with the interface of Avalon-MM Slave to set the baud rate of the serial port, the number of bytes of the data to be sent and the base address of the data stored in the memory.

Secondly, write the data to be sent to the specified location in the memory and then start the Master Read type DMA controller, thus the data stored in the memory is sent out one by one through the USART sent controller. When all the data that you want to send has been sent, an interrupt will be generated in the NIOSII processor to inform the processor that the transmission of serial data is completed, so as to start the next data transmission. Since the whole process of data sent is managed by the Master Read type DMA controller, NIOSII processor can concentrate on other things and not be disturbed, thus the utilization ratio of NIOSII CPU increases greatly. When the NIOSII processor need to receive data through serial ports, firstly, it's necessary to fulfill the configuration on the USART receive controller and a Master Write type of DMA controller through the register file with Avalon-MM Slave interface to set baud rate of the serial port, the number of bytes of data which will be received and the base address of the data stored in the memory. Secondly, start the Master Write type of DMA controller, thus the data received through the USART controller can be stored in the specified location in the memory one by one, when all the data is received, an interrupt will be generated in the NIOSII processor to inform the processor that the transmission of serial data is completed, so as to read the data that has been received from the memory for processing and start the next data transmission. Since the whole process of data reception is managed by the Master Write type of DMA controller, NIOSII processor can concentrate on other things and not be disturbed, thus the utilization ratio of NIOSII CPU increased substantially.

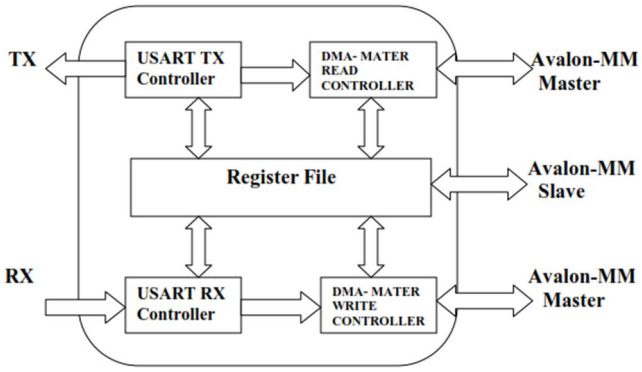

Figure2. The overall framework of USART IP core in DMA mode

III. THE DESIGN AND IMPLEMENTATION OF EACH MODULE

A. *The design of USART receiver controller*

The reception of serial port uses the basic frame format. First of all, detect the start-bit is low-level. Then receive the bytes of data bit by bit under the control of the clock in the baud rate. Finally, receive the high-level of the stop bit.
In this paper, a USART receiver controller is designed using the way of finite state machine in the hardware description language of Verilog HDL, thus completing the timing control of the data reception of serial port. Its state transition diagram is shown in Figure 3.

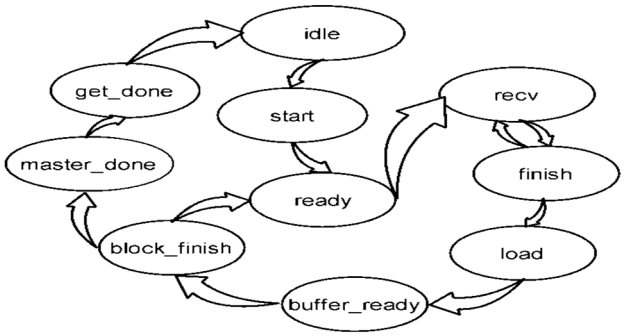

Figure 3. The state transition diagram of the USART receiver controller

As can be seen from figure 3, the state machine is in the idle state at the beginning. When the DMA Master Write controller is started to conduct a reception of the data, the state machine enter into the start state. Start & ready, these two states are mainly used to clear the receiver shift register and the bit counter and prepare for receiving a byte of data. When it's ready, the state machine enters into the receiver state. Recv & finish, these two states are mainly used to receive the byte of data bit by bit under the control of serial port baud rate clock and store it in the reception shift register. When the reception of a byte of data is completed, the state machine enters into the load state. Load, buffer_ready, these two states are mainly used to move the byte data to the Master Write type of DMA controller in order to complete write operation from the bytes of data to the memory. Then the state machine enters into the block_finish state. In this state, the state machine makes a judgment of the number of bytes of the data that has been received. If the number is less than the number of bytes of data that should be received, it shows that all the data has not been received, so the count plus 1 and the state machine enters into the ready state to receive the next data byte and send it to the Master Write type of DMA controller. The state machine doesn't enter into the state of master_done, until all the data bytes are received. Master done and get_done, these two states detect whether this DMA Transfer is completed. If it's done, the state machine will generate an interrupt signal and enter into the idle state. At this point, a full serial data reception in the DMA transfer mode.

B. *The design of USART sending controller*

The transmission of serial port uses the basic frame format. First of all, send low to the start-bit, and then under the control of the clock in the baud rate, send 8-bit data from D0 to D7, finally sent high to the stop-bit. In this paper, a USART sending controller is designed using the way of finite state machine in the hardware description language of Verilog HDL, thus completing the timing control of the data transmission of serial port. Its state transition diagram is shown in Figure 4.

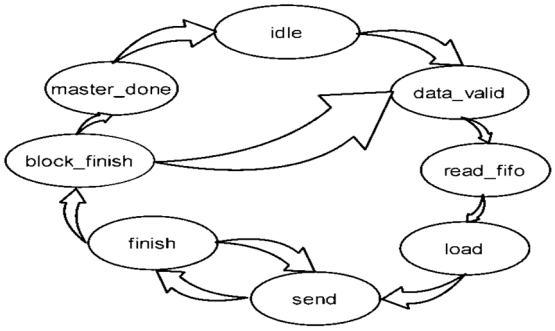

Figure 4. The state transition diagram of the USART sending controller

As can be seen from Figure 2, the state machine firstly is in idle state. When the Master Read type of DMA controller starts to conduct a data transmission, the state machine moves into the data_valid state. Data_valid, read_fifo and the load, these three states are mainly used to access a byte which is read by the Master Read type of DMA controller from the memory. Splice the byte with the start bit and stop bit together and send it to the shift register. After that, the state machine enters into the send state. Send and finish, these two states are mainly used to send the data in the transmit shift register bit by bit under the control of serial port baud clock. When the process of sending the data in the transmit shift register is completed; the state machine will enter into the state of block_finish.

In this state, the state machine makes a judgment of the number of bytes of the data which has been sent out. If the number is less than the number of bytes of data that should be sent, it shows that all the data has not been sent out, so the count plus 1 and the state machine enters into the state of data_valid to read and send the next data byte. The state machine doesn't enter into the state of master_done, until all the data bytes are sent out. In this state, the state machine makes detection whether this DMA transfer is completed. If it's done, the state machine will generate an interrupt signal and enter into the idle state. At this point, a full serial data transmission in the DMA transfer mode.

C. The design of the register file with the interface of Avalon-MM Slave

AVALON bus, an open interconnect bus; can be used to connect the main peripherals and the minor peripherals. The main peripheral can initiate bus transfers on the AVALON bus. While the minor peripheral can only respond to the bus transfers. The main peripheral connects with the AVALON bus using the Avalon-MM Master interface, while the minor peripheral using the Avalon-MM Slave interface.

The register file with the Avalon-MM Slave interface designed in this passage is a peripheral with the Avalon-MM Salve interface. There are a total of four 32-bit registers in it, whose specific structure and function is shown in Table 1.The NIOSII processor accesses these 4 registers by the way of base address plus address offset, which can control the configuration of the USART IP soft-core in the DMA mode as well as the reception and the transmission of the serial data.

D. The design of the Master Read type of DMA controller with the interface of Avalon-MM Master

The Master Read type DMA controller with the Avalon-MM Master interface designed in this passage is the peripheral with the Avalon-MM Master main ports. It finishes the basic reading transport through the switching fabric between Avalon- MM Master main ports and the AVALON, so that it can read the specified length of data from the memory whose starting address is specified and send it out one by one to the USART send controller. Functional simulation is carried through in this passage using the QUSARTus II software. The simulation waveform is shown in figure 5.
As can been seen from figure 5, the reading transfer of the main port starts at the first rising edge of clk. In the first clock cycle, the primary port makes the address and the read_n effective. If the wait request signal is invalid, valid data which is read will appear in the read data signal line in the second clock cycle. The primary port only captures read data in the second rising edge of the clock cycle to complete a basic reading transmission.

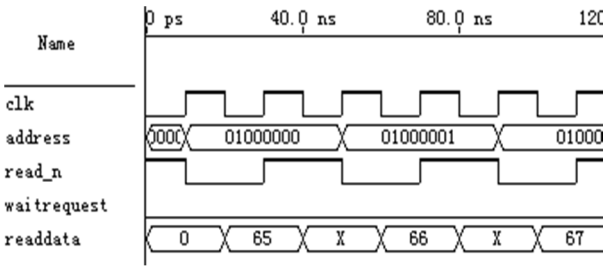

Figure 5. The waveform of the basic reading transfer of the Avalon-MM Master interface

E. The design of the Master Write type of DMA controller with the interface of Avalon-MM Master

The Master Write type DMA controller with the Avalon-MM Master interface designed in this passage is the peripheral with the Avalon-MM Master main ports. It finishes the basic writing transport through the switching fabric between Avalon-MM Master main ports and the AVALON, so that it can continuously store the specified length of data received from the USART receiving controller in the memory whose starting address is specified. Functional simulation is carried through in this passage using the QUSARTus II software. The simulation as can been from figure 5, the writing transfer of the main port starts at the first rising edge of clk. In the first clock cycle, the primary port makes the address, writedata and the write_n effective. If the waitrequest signal is invalid, the valid writing data writedata, will be captured in the second rising edge of the clock cycle to complete a basic writing transmission.

IV. THE HARDWARE TEST

A top-level file named dma USART ip is created in verilog HDL language in this design. It completes the design of the USART IP soft core in the DMA mode eventually through the instantiation and interconnection of the above 5 sub-modules including USART sending controller, USART receiver controller, the register file with Avalon-MM Slave interface, the Master Read type of DMA controller with the Avalon-MM Master interface, the Master Write type of DMA controller with the Avalon-MM Master interface. The RTL generated after synthesizing is shown in figure 6.

TABLE I. THE SPECIFIC STRUCTURE AND FUNCTION OF EACH REGISTER

Property / Register	offset	Access properties	The number of effective bits	Function
START DMA start register	0	Write-only	2	start DMA controller
BAUD serial port baud rate register	1	Read& write	16	set the baud rate of the serial port
Register of BASE base address	2	Read& write	32	set base address of DMA
LENGTH data length register	3	Read& write	32	set data length

The FPGA chips used in this design is Altera's Cyclone II EP2C35F672C8 as the hardware verification platform.After the compilation, adaptation and integration of the QUSARTusII

software as well as the analysis to the compile report, the IP core uses 465 LE (logic gates), which account for 1.4% of the total LE, 352 registers, 256 bytes of memory (2048 memory bits) which account for 0.4% of the total on-chip memory, what's more, the highest frequency goes up to 118.314Mhz.

As the DMA mode USART IP soft core is AVALON bus-based IP core, a simple system NIOSII system is built to the test validation. In this paper, a top-level application test program is made based on NIOSII processor in C++ language to test the time of the NIOSII CPU occupied by the DMA mode USART IP soft core when transmitting different sizes of data blocks, comparing with query mode USART IP soft core and interrupt mode USART IP soft core. The results are shown in table 2.

TABLE II.

COMPARISON OF CPU TIME CONSUMED∧CLOCK CYCLES∨

CPU Time-consumed / Data Block / IP Core	64 Bytes	512 Bytes	4096 Bytes
USART IP core in poll mode	277761	2222063	17776676
USART IP core in interrupt mode	11312	90624	725024
USART IP core in DMA Mode	177	177	177

As can been seen from table 2, in the process of data transmission, the time of CPU consumed by DMA mode USART IP soft core is shorter than that consumed by query mode USART IP soft core or interrupt mode USART IP soft core. What's more, with the increase of the size of the data block which need to be transmitted, the advantages of the shorter occupied CPU time becomes increasingly evident. In this way, it improves the performance of the NIOSII processor greatly.

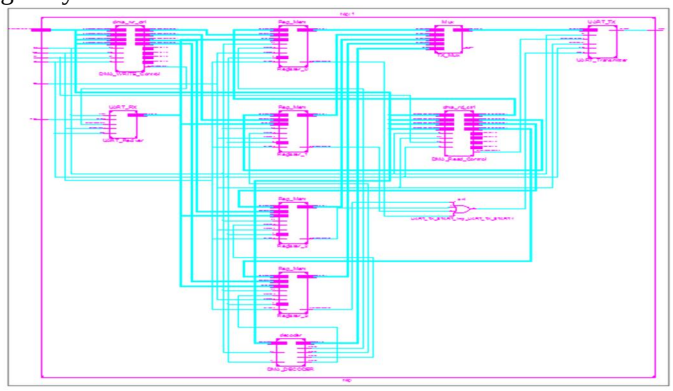

Figure 6. Design of USART IP soft core based on DMA mode

Advantages and Applications

1. CPU will be free from data transmission process
2. GPRS,GPS and other communication devices

Conclusion

1. IP sot-core of USART is a better choice for faster communication devices
2. As the design uses less logic system performance will be improved greatly

REFERENCES

[1] Altera Corp, Avalon-MM Slave reference manual.

[2] Altera corp, USART Core User's Guide.

[3] Sicong Wu, Weiwe Zhao Yanhong CHhen,"Application research of Hilbert transform in reactive energy measurement",Hunan normal university polytechnic college, Changsha 410081, China

[4] Hua Liu,"Reactive Power Mearsurement and Multi-function Meter Data Management", Henan University.

[6] Zhou ligong. SOPC-based Embedded System Tutorial [M]. Beijing: Beijing University of Aeronautics 2006

[7]

[8] Lingge Jiang,"Theories and Methods For Reactive Energy Measurement",Zhangjiajie Power Bureau, Zhangjiajie.

[9] Wei Wang, Xiaoru Wang, Xiaoqing Huang, Da-peng Xie "The research on Hilbert digital filter of reactive power measurement" Southwest Jiaotong University, Chengdu 610031, China

[10] Yang fuguang. Efficient UART communication and its applications based on DMA in ARM.Chinese Academic Journal Web Publishing General Library 2008. [5] Altera Corp, Avalon-MM Maste reference manual.