International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Design of a Configurable Transmission Device for Interface of a High Speed SOC with Low Speed Components

B. Harika Goud
M. Tech .Student, CVSR College of Engineering
bimagani.harika@gmail.com

K. Ch Prathap Kumar m
Asst. Prof., ECE Deptt., CVSR College of Engineering
chaithu58@gmail.com

***Abstract* — UART is the key component of serial communications subsystem of a computer. Here UART is a configurable transmission device and high speed soc is AHB. AHB is used to connect High speed components.DMA acts as a master and memory acts as a slave. The data is transmitting from DMA to memory through AHB bus. its acts like interfacing. Five main sub modules are well designed and verified. Compared to other system buses it turns out that the performance of the SOC can be improved and processing time of SOC is also improved by using AHB Bus interfacing.**

***Keywords* — AHB, DMA, UART.**

I. Introduction

UART is the most basic and most commonly used method of communication in the embedded system, whose performance will to some extent determine where the overall system can meet the design requirements. The implementation of UART is basically uses the on chip. It makes the design of hole system has great limitations for the parameters of which is solidify already on the chip 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. Design of a Configurable transmission device for interfacing of a high speed soc here.

II. The Overall Framework

The entire UART DMA mode mainly includes the following 5 sub-modules: UART send controller, UART Receive controller, Register file with the Interface of AHB-MM Slave , Master Read type DMA controller with the interface of AHB-MM Master and Master Write type DMA controller with the interface of AHB-MM Master.

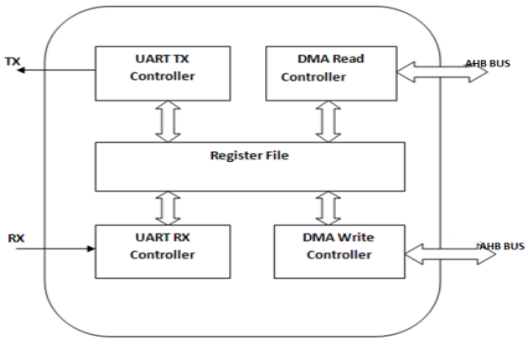


Fig.1. Block diagram

III. The Design and Implementation of Each Module

A. The design of UART 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 UART 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. As can be seen from Figure 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.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

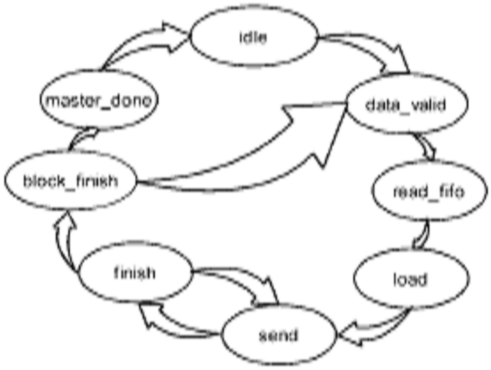


Fig.2. The state transmission diagram of the UART sending controller

B. The design of UART 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 UART 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. As can be seen from figure, the state machine is in the idle state at the beginning. When the Master Write type of DMA 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 recv state. Racv, 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.

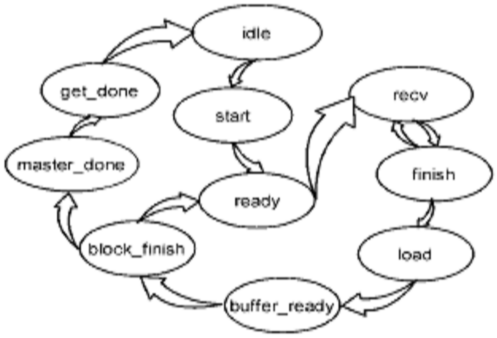


Fig.3. The state transmission diagram of the UART receiver controller

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

AHB 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 AHB bus. While the minor peripheral can only respond to the bus transfers. The main peripheral connects with the AHB bus using the AHB-Master interface, while the minor peripheral using the AHB- Slave interface. The register file with the AHB-Slave interface designed in this passage is a peripheral with the AHB-Salve interface. There are a total of four 32-bit registers in it.

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

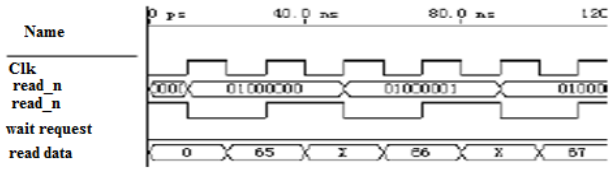
The Master Read type DMA controller with the AHB_Master interface designed in this passage is the peripheral with the AHB-Master main ports. It finishes the basic reading transport through the switching fabric between AHB- Master main ports and the AHB, so that it can read the specified length of data from the memory whose starting address is specified and send it out one by one to the UART send controller. Functional simulation is carried through in this passage using the Modelsim software The simulation waveform is shown in figure 4.As can been seen from figure 4, 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.

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 AHB-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 AHB- Master main ports and the AHB, so that it can continuously store the specified length of data received from the UART receiving controller in the memory whose starting address is specified. Functional simulation is carried through in this passage using the Modelsim software. The simulation as can been from fig., the writing transfer of the main port starts at the first rising edge of clk.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 6, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

IV. Simulation Results



V. Software Test

Top-level file named dma_uart is created in verilog HDL language in this design. It completes the design of the UART IP soft core in the DMA mode eventually through the instantiation and interconnection of the above 5 sub modules including UART sending controller, UART receiver controller, the register file with AHB- Slave interface, the Master Read type of DMA controller with the AHB-Master.

VI. Conclusion

By employing DMA controller along with UART, the process of data transmission, the processing time is shorter by using the interface of AHB bus than that using other system buses. Received data is handled by the DMA and it has written the data into the specified location of memory. And the DMA has given the same data from the memory to the transmitter and the transmitter sent the serial data successfully which is nothing but the same data received serially.

References

[1] J. Liang, Swaminathan, S, ?ASOC: a scalable single chip communications architecture?, tessier, R. parallel architectures and compilation.
[2] AMBA Bus DMA controller specification,draft 1.03,gareth Morrise.
[3] C8237 programmable DMA controller Core,http://www.cast-inc.com.
[4] Flynn, D.?AMBA:enabling reusable on-chip designs?, IEEE micro,17(4),july/August 1997,pp 20-27.