International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 6 Issue 1 ǁ 1. 2018 ǁ PP. 55-63

IP Cores Interconnection Model Supporting Multiple Fullduplex Communications

Shimin[1],Mojinhui[2],Yiqingming[3]
College Of Information Science And Technology, Jinan Univeristy, China, 510632
Corresponding Author: Shimin

ABSTRACT: Considering the low transmission efficiency between CPU and SPI or UART, and the low execution efficiency of CPU instruction, an IP cores interconnection model supporting data parallel processing is proposed in this paper. Data transmission between CPU and AXI4 can be realized by extending three ARMv4 custom instructions. A conversion interface based on AXI4 protocol is provided to realize the compatibility of high speed bus and low speed device. The model supports the AXI4 out-of-order and overlapped transport mechanisms, and CPU can flexibly perform other instructions while paralleldatasare being transmitted. The entire design uses Verilog to carry on the structure level description, and the Modelsim simulation result shows the high data transmission efficiency. CPU can simultaneously talk with two devices correctly in full duplex way, and high instruction execution efficiency. CPU executes all test instructions with only 36 clock cycles, and has 156 free clock cycles before the data transfer is completed.

Keywords: AXI4protocol;Interconnectionmodel;ARMv4instruction;Conversion interface;Full duplex; Execution efficiency

---

Date of Submission: 17-01-2018 Date of acceptance: 30-01-2018

I. A NEW AXI4 INTERCONNECTION MODEL

With the increasing integration of chips, the capability and technology of SoC (System on Chip) design based on IP (Intellectual Property) have been improved rapidly. SoC design methodology related to IP reuse has become an effective method to solve problems such as high design cost, long development cycle and complexfunction. [1] In the IP reuse process, the SoC system needs the on-chip bus to realize the interconnection between the various IP cores. Different bus technologies will cause differences in the transmission efficiency of the system, thus affecting the overall performance of the system. At present, there are still many on-chip buses like SPI and I2C serial communication on the market. Due to the limitation of low speed transmission, the transmission efficiency of SoC system is low, and the efficiency of CPU instruction execution is low. [2]AXI4 (Advanced eXtensible Interface) bus, a new bus specification introduced by ARM company in 2012, has been widely applied in engineering due to its high transmission efficiency and reliability. Based on the AXI4 bus protocol, we extend the ARMv4 custom instruction and propose the AXI4 conversion interface supporting full duplex communication, and realize an integrated model composed of host CPU, AXI4 bus and slave SPIand UART.The AXI4 bus models [3-5] mostly used APB (Advanced Peripheral Bus) to realize the interconnection of bus and peripherals, but they can notread and write datas at the same time, because of the limitation of APB agreement.The above models also did not detail the way of interconnection between host and bus. In this paper,a new interconnection model adopts the special slave interface which supports the AXI4 protocol, and realizes the full duplex communication between the AXI4 bus and the slave devices. It has high transmission efficiency. In addition, this paper provides a concrete scheme for interconnecting host and bus, which has strong practicability.

As shown in the Figure 1, CPU is interconnectedwithUART(Universal Asynchronous Receiver/Transmitter) and SPI(Serial Peripheral Interface) through the AXI4 bus, which consists of the master interface, the AXI4 interface, the arbitration module and theslave interface.

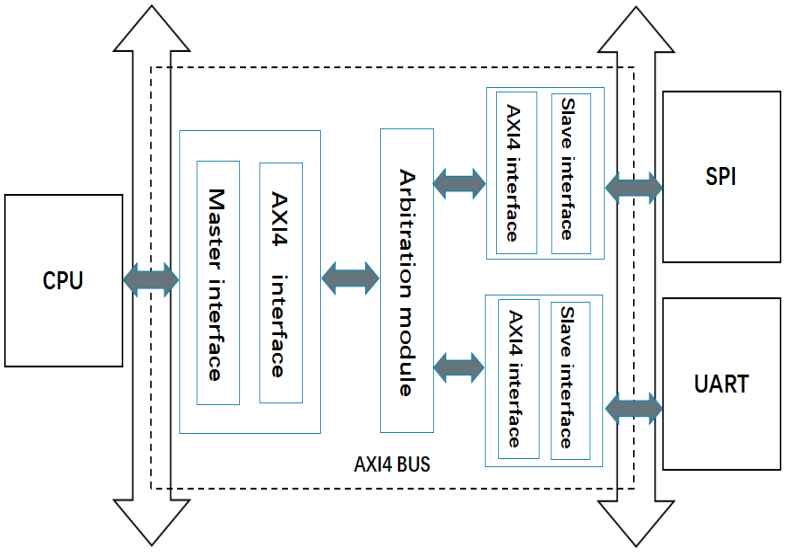

Figure 1: IP cores interconnection model based on AXI4

II. CPU AND AXI4 BUS INTERCONNECTION

2.1 OVERVIEW OF AXI4 BUS:

The AXI4 standard introduced by ARM can meet the requirements of SoCfor its good bandwidth and complex communication. It has become the first choice on chip bus in industry and academic field.[6-11] The AXI4 bus consists of five independent channels: read address, read data, write address, write data, and write response. The advantage of using independent transmission channels is to avoid signal congestion and error redundancy, and to allow both read and write transmissions to be carried out simultaneously, thus improving the transmission efficiency. The AXI4 bus enhances the performance of burst mode and the utilization rate of multiple device interconnections. In order to ensure the efficiency of data transmission, AXI4 proposes a random order operation. It can be used for different accesses in order as shown in Figure 2, and ID is used to mark the attribution of access.

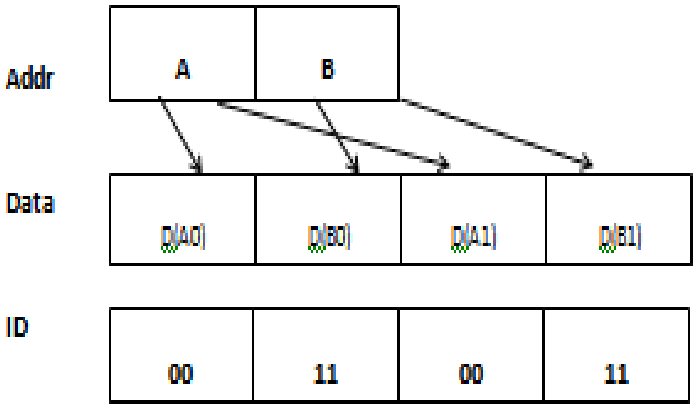

Figure 2AXI4 out-of-order operation

2.2AXI4 Host Interface

Commonly, every IP in the SoC must have a consistent interface to achieve the wanted interconnection. If the timing requirements are inconsistent, the conversion interface shouldbe extendedin the original IP. CPU and AXI4 bus need to be interconnected through the host interface as shown in Figure 3. The function of the host interface is to convert the addresses, datas and to control signals generated by the CPU, lastly providing them to the AXI4 bus by sequential logic.

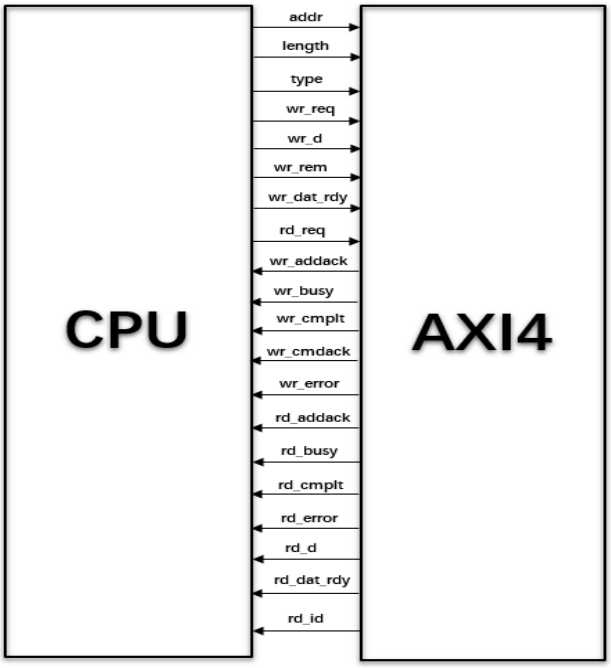

Figure 3 Master interface

As shown in Figure 4 andFigure 5,the host first providesseriesinformations,such as addr, length, type, rd_req, wr_req, to the AXI4 bus for initiating read or write requestoperations. When the slave responds to the host byrd_addack/wr_addracksignal, the host initiates read or write dataoperations separately through independent channels, and the host finally confirms the transmission according to the rd_cmplt/wr_cmplt signal. When the host reads address ADDR1 sent immediately after sending a read address ADDR2, AXI4 bus has two requests to read that transmission, it is overlapping transmission. Overlapping transmission can support the flow transmission, which can reduce the transmission delay and improve the efficiency of data transmission.

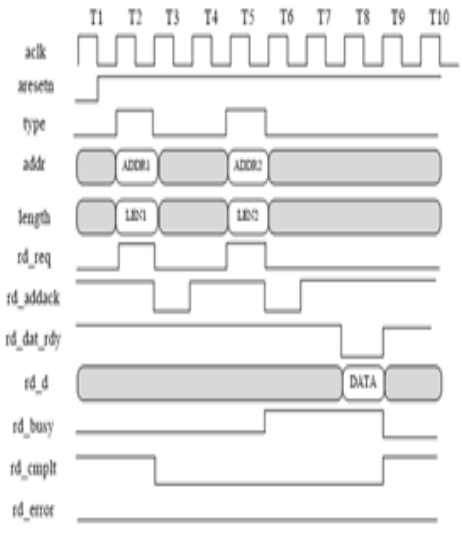

Figure 4 Masterinterface read timing

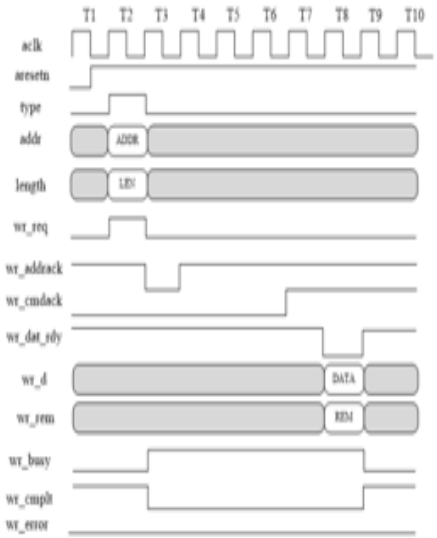

Figure 5 Masterinterface write timing

2.3 Cpu Supporting Axi4 Host Interface:

In this model, The CPU supports the ARMv4 instruction set, and has the advantages of small size, low power consumption and high execution efficiency. As shown in Figure 6, CPU consists ofControl Unit, Data Path Unit, and Arithmetic Logical Unit, in accordance with the diffrernt functions.[12]

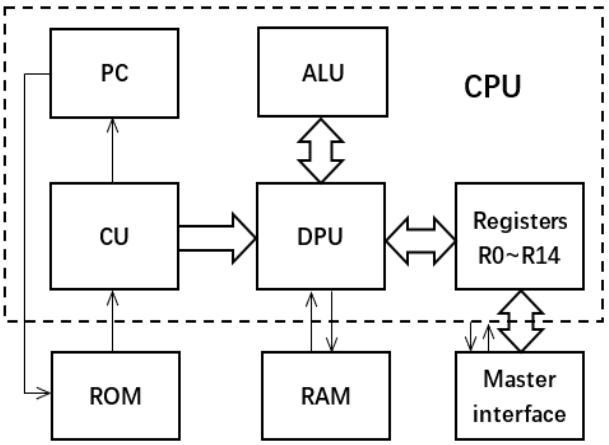

Figure 6 CPU inner structure

The Control Unit is responsible for decoding instructions, analyzing the functions of the current instructions, and sending the signalsrelated to registers or memory selection to the Data Path Unit. The Data Path Unit generates data path signals to read and write the specified registers or memories, and simultaneously performs data operation with the Arithmetic Logical Unit, and lastly writes them back to registers, so the CPU can complete the function of the instructions. The CPU can implement fiveinstruction functions: data processing instruction, multiplication instruction, jump instruction, load/storage instruction and transmission instruction. A typical 32 bit ARM instruction encoding format is shown in Figure 7.**Cond** is a conditional code that determines whether the instruction is executed or not. **Type** is an instruction type code, which is used to distinguish different kinds of instructions. **I** is a symbol to tell the addressing mode of second operand in the Data processing instruction. **Opcode**is the instructions operator encoding, the **S** indicates whether the current instruction operation affect the value of CPSR or not.**Rn** is a register encoding first operands, **Rd** is the destination register encoding, and **shifter_operand** says the second operand.

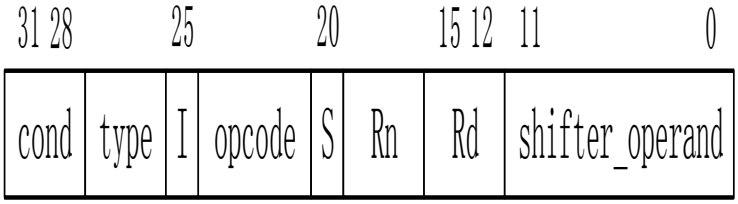

Figure 7 Instruction encoding format

As shown in Table 1, this paper extends three ARMv4 custom instructions for data transmission between CPU and AXI4. The MRB instruction is used for CPU reading datas from AXI4 bus by sendingdatas from host interface to CPU register stack. MBR instruction is used for CPU writing datas to AXI4 bus by sendingdatas from CPU register stack to host interface. The MBI instruction is used to set the operation of address, length, and type of the AXI4 bus burst.

Table 1 ARMv4 new custom command

New Inst	31 28	25	20	15 12	11 0			
	cond	type	I	opcode	S	Rn	Rd	Shifter_operand
MRB	XXXX	00	0	10X0	0	XXXX	XXXX	XXXX1111XXXX
MBR	XXXX	00	0	10X1	0	XXXX	XXXX	XXXX1111XXXX
MBI	XXXX	00	1	10X1	0	XXXX	1111	XXXXXXXXXXXX

The arbitration module of AXI4 bus devides the AXI4interface into twointerface with different addresses . It can separately communicate with SPI and UARTby distinguishing theright address.

III. AXI4 BUS、SPI AND UART INTERCONNECTION

3.1 Axi4 Slave Interface

As shown in Figure 8, the function of the slave interface is to realize the conversion of the addresses, datas and control signals between the AXI4 bus and the SPI and UART.

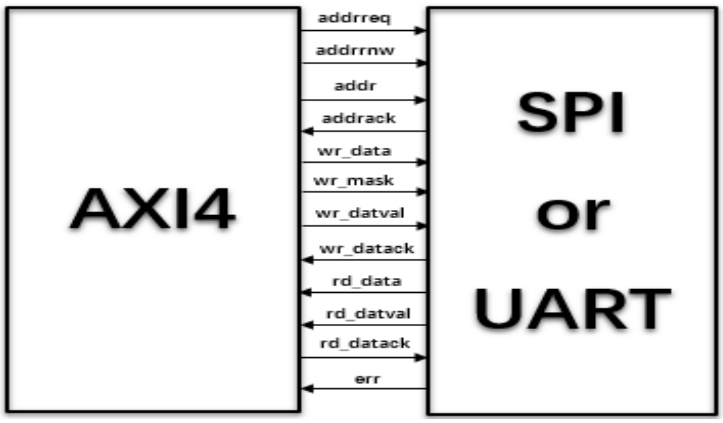

Figure 8 Slave interface

The sequence diagrams of reading and writing operations from the AXI4 bus to the peripheral devices are similar to the operations between CPU andthe AXI4 bus.

3.2 Spi Supporting Axi4 Slave Interface

SPIis a synchronous serial interface that can simultaneously drive the transmission of serial data and receive.[13]It has the synchronous serial clock port, the serial data input port, theserial data output port, and the device selection port.There are four transmission modes can be selected by the configuration of clock polarity and clock phase.SPI has many advantages, such as less pins, simple timing, powerful functions and so on. But its shortcomings are also obvious: there is no transmission demand and response mechanism, and it can not determine whether the data is correct or not. In this paper, SPI supporting AXI4 interface as shown in Figure 9is designed to solve these problems, and it can realize the compatibility of high speed bus with low speed equipment. It mainly includes the following modules:**Transceiver Control Module**, **Clock module**, **Receive FIFO**, **Send FIFO** and **Bus Interface Control Module**.

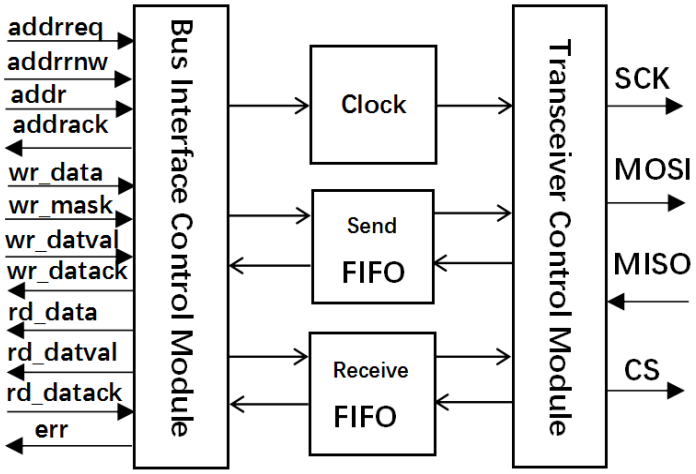

Figure 9 SPI structure with AXI4 slave interface

The function of the **Transceiver Control Module** is to detect the storage status of the**Send FIFO** and the **Receive FIFO** at all times, and to control the sending and reception of the datas. **Receive FIFO** is used to store

datas received by SPI interface, and **Send FIFO** is usedto store datas sent by SPI interface. Both of them will send state signal to the Transceiver Control Module and Bus Interface Control Module simultaneously. **Bus Interface Control Module** is to achieve the function of reading or writting operations between AXI4 bus and SPI. When the AXI4 bus applies to transmit datas, this module will write signalsto **Send FIFO**fisrt and datas will be in the**Send FIFO**, waiting the SPI sending datas to the **Send FIFO**lastly; When the AXI4 bus receivesdatas ,the **Bus Interface Control Module** will send signals to the **Receive FIFO** and read the datasfrom **Receive FIFO**.

3.3 Uart Supporting Axi4 Slave Interface

UART is an asynchronous communication interface that supports full duplex communicationon the serial link.[14] The sender and receiver of the communication have their own independent clocks, and the rate of transmission is agreed by both parties. Its function is to convert the received serial datas into parallel datas,andto convert parallel datas into serial datas that will be sent out.UART has two signal ports: the sending signal **TXD** and the receiving signal **RXD**. Normally, a frame of data consists of a starting bit, a data bit, a parity bit, and a stop bit. The data transmission format of UART takes low level as the starting bit, and transfers 8 bits from low to high order, 1 bit parity bit, and finally uses high level as the stop bit.As the protocol of UART uses serial communication, only eight useful bitscan be transmitted in one frame, and the other bits cause the transmission efficiency low. This paper designsa new UART supporting AXI4 interface, As shown in Figure 10, to reduce transmission delays by making good use of the advantages of AXI4 bus. The improved UART consisits of sendingunit, receiving unit,baudrate generator and bus interface control module.

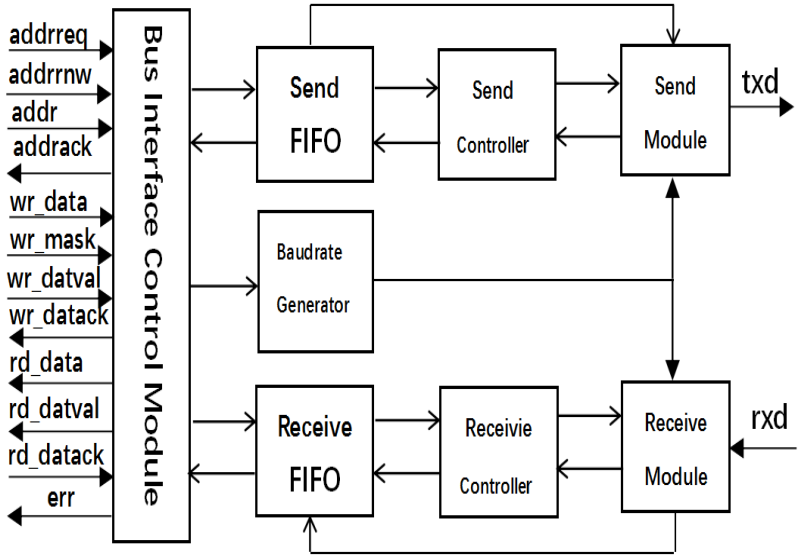

Figure 10 UART structure with AXI4 slave interface

The sending unit of the UART contains**SendModule**, **SendController** and**Send FIFO**. The main function of the **Send Module** is to receive effective signals from**Send FIFO**, transform parallel datas into serial datas, add initial bits, parity bits and stop bits, and finanlly send them out. The function of the **SendControl Module** is to detect the storage state of **Send FIFO** and control **SendModule**carringdatas into the **Send FIFO**. Similarly, the receiving unit of the UART is compost of**ReceivieModule**, **ReceivieController**, and **Receive FIFO**. The main function of the **ReceiveModule** is to detect the received signals, and convert the received serial effective datas into parallel datas,which will be written to**Receive FIFO**. **ReceiveController** is used to control the**ReceiveModule**to writedatas into the**Receive FIFO** according to the storage situation of the**Receive FIFO**. The function of **Bus Interface Control Module** is to realize the readingor writing operationsbetween the AXI4 bus and the SPI.

IV. SIMULATIONRESULTS

This paper compiles the CPU test instructions shown in Table 2, and the entiremodel is verified by Modelsim software -Altera 10.1a. Modelsim needs to provide an incentive source, which includes providing datas:**8fh, 89h, b3h** to the **RXT**,input of the UART,and the datas:**35h, 69h, 92h** to the**MISO**,input of the SPI. It can be finally verified the correctness of the model by judging the correctionof data transmission from CPU to AXI4 bus and to SPI, UART, and the comparisons between the instruction testing resultsand the expected values.

Table 2 Test instructions from ROM

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	E3A00037	E3A01045	E0050190	E0803001	E320FB01	E3201003	E3202001	E12FF0F0
08	E12FF0F5	E12FF0F3	E320FB01	E3201003	E3202001	E10F30F7	E3A020EF	E0040291
10	E3A06181	E5864000	E5967000	E320F000	E3201003	E3202001	E12FF0F2	E12FF0F6
18	E12FF0F7	E8A6003F	E320F000	E3201003	E3202001	E10F20F7	00000000	00000000

The main test contents are as below: CPU applies for a burst write with the address **00000400h** , length **3**, and sends datas from R0, R5, R3 to the AXI4 bus; CPU applies for a burst read with the address **00000400h**,length **3**,and stores the datas in the R3; CPU applies for a burst write with the address **00000000h** , length **3**, and sends datas from R2, R6, R7 to the AXI4 bus; Load the datas from R0~R5 in batch to the RAM with starting address **0x40000020** ;CPU applies for a burst read with the address **00000000h**, length **3**,and stores the datas in the R2.

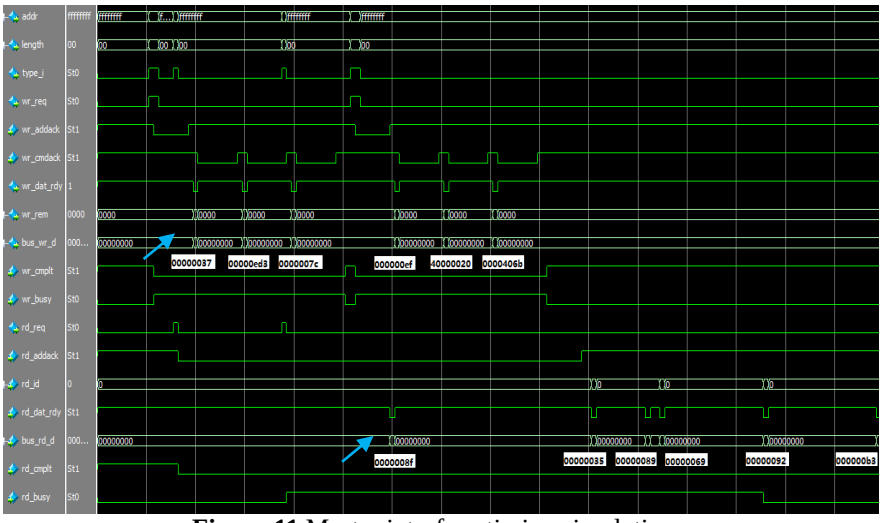

Figure 11 Master interface timing simulation

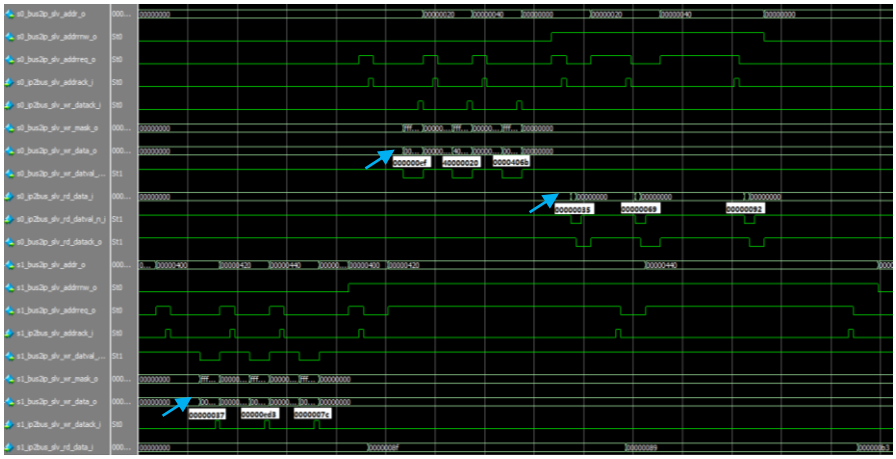

Figure11 and Figure12 are the timing diagrams of the host interface and slave interface.When CPU initiates the read and write operations, and the related transmission datas can be observed from the two diagrams.

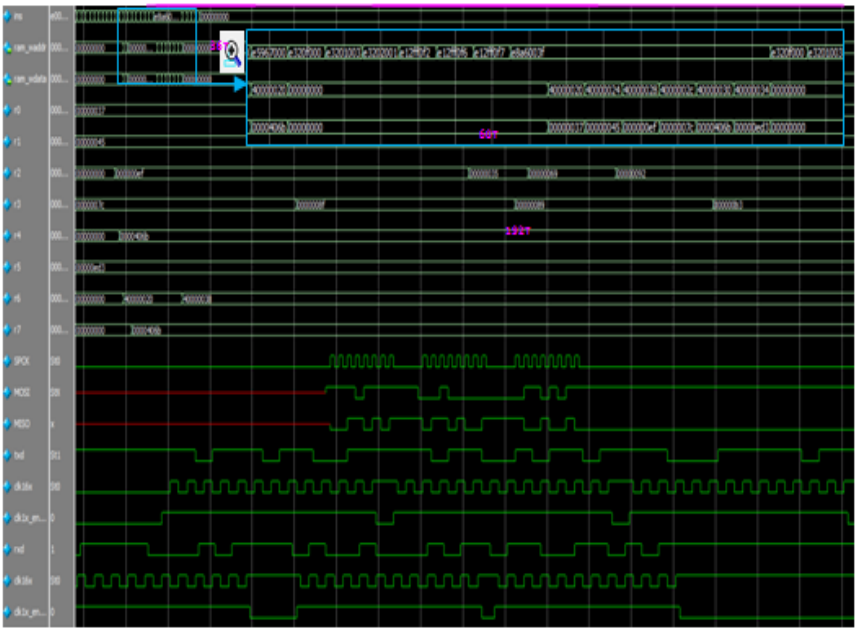

Figure 13 Overall model simulation diagram

CPU performs data processing and store\loadinstructions, then R0~R7 turn to be **00000037, 00000045, 000000ef, 0000007c, 0000406b, 00000ed3, 40000020, 00000406**, coinciding with the expected value; When CPU executes all instructions, the R2 appeares to be **00000035, 00000069, and 00000092** in turn,the R3 appeares to be **0000008f, 00000089, 000000b3** in turn, and the above six datas are presented in the form of overlapping, it suggests that CPU can read the SPI and UART datas at the same time;The UART txd signaloutputs **00000037, 00000ed3, 0000007c**, and the SPI MOSIsignaloutputs**000000ef, 40000020, 0000406b**.The above datas indicate that both UART and SPI can correctly communicate with CPU.

From the waveform of **MOSI, MISO, TXD** and **RXD**, it can be seen that CPU can perform full duplex communication with SPI and UART simultaneously. In addition, when CPU transfers data with SPI and UART, CPU has already executed time-consuminginstruction **E8A6003F** ahead of time, which means that the high-speed instruction execution of CPU is independent from the low speed data transmission of the system. The simulation shows that CPU needs 36 clock cycles to execute all the test instructions, SPI data transmission needs 60 clock cycles, and UART data transmission needs 192 clock cycles. Thus, CPU can make full use of the 156 delayclock cycles of the system data transmissionto deal with the other transactions of the system, which has a high efficiency of instruction execution.The reference [2] model adopts the custom SPI with fixed frame length of 32 bits,but the effective data part accounts for 16 bits, bus data transmission utilization rate is only 50%. The reference [3] model uses the APB to connect the AXI4 bus with the SPI, but they can not read and write at the same time. The model in this paper is compared with the above two models in the same condition: CPU clock frequency is 25MHz, 24 byte data is read and written between CPU and the bus, the result is shown in Table 3It can be seen in table 4,Compared with the bandwidth of reference[2], the interconnection model designed in this paper has increased by 63%, and the transmission rate is increased by 8.6 times, and 1.2 times higher than that of the reference[3].Therefore, it indicates that the interconnection model of bandwidth and transmission rate have beenoptimized and proposed significantly.

Table 3 Performance comparison of different models

Model	Bandwidth / bps	24 Bytes transmission time / μ s
reference［2］	25	15.36
reference［3］	1600	3.5
This paper	1600	1.6

V. SUMMARY

By extending ARMv4 custom instructions and proposing a conversion interface based on AXI4 bus protocol, an new IP cores interconnection modelis designed and implemented,which has the featurethat multiple slave devicescan support full duplex communication at the same time. This model can solve thelong waitsand single duplex problems in the data transmission. The experimental result shows that the model can greatly improve the efficiency of data transmission and that CPU has high efficiency of instruction execution.

REFERENCE

[1]. Dylan Stow, Itir Akgun, Russell Barnes, et al. Cost analysis and cost-driven IP reuse methodology for SoC design based on 2.5D/3D integration[C]. IEEE/ACM International Conference on Computer-Aided Design, 2016, pp.1-6

[2]. 雷 红.一种自定义 SPI 总线协议的设计与实现[J].信息通信,2017,173(5):84-85

[3]. 肖潇.基于 AXI 的 SoC 互联结构的设计与验证[D].硕士,国防科学技术大学,2015

[4]. 魏晓川,潘明.支持多协议的 AXI 互联的设计[J].价值工程,2016,35(18):59-61

[5]. 胡景华.基于 AXI 总线的 SoC 架构设计与分析[D].硕士,上海交通大学,2013

[6]. Ckristian Duran, D. Luis Rueda, Giovanny Castillo, et al. A 32-bit RISC-V AXI4-lite bus-based Microcontroller with 10-bit SAR ADC [C]. 2016 IEEE 7th Latin American Symposium on Circuits & Systems, 2016, pp.315 - 318

[7]. AMBA™ Specification (Rev 2.0), ARM Co.Ltd,1999, http://www.arm.com

[8]. Shaila S Math, Manjula R. B, S.S. Manvi, et al. Data Transactions on System-on-Chip Bus Using AXI4 Protocol[C]. 2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering, 2011, pp.423-427

[9]. Priyanka Gandhani, Charu Patel. Moving from AMBA AHB to AXI Bus in SoC[C]. Int. J Comp Sci. Emerging Tech, Vol-2, No 4 August, 2011, pp.476-479

[10]. 杨舜琪. AMBA AXI4 总线的研究与实现[D]. 哈尔滨工业大学, 2011

[11]. 马飞,刘琦,包斌.基于 FPGA 的 AXI4 总线时序设计与实现[J].电子技术应用,2015,41(6):13-15

[12]. 陈明敏,易清明,石敏.ARMv4 指令集嵌入式微处理器设计[J].电子技术应用,2014,40(12)：23-26

[13]. 秦宇.基于 APB 总线 SPI 接口 IP 核的设计与验证[D]. 硕士,贵州大学, 2015

[14]. 余龙,肖明,王建雄.基于 AMBA 总线的 UART 模块设计与验证[J].微处理机,2013,34(2):73-77

Shimin. “IP Cores Interconnection Model Supporting Multiple Fullduplex Communications.” International Journal of Research in Engineering and Science (IJRES), vol. 06, no. 01, 2018, pp. 55–63.