Research Journal of Applied Sciences, Engineering and Technology 4(2): 116-119, 2012
ISSN: 2040-7467

Submitted: September 23, 2011 Accepted: November 02 , 2011 Published: January 15, 2012

Research on Communication Protocol Based on NOC Resources

Zhang Bo and Zhang Gang
College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024

Abstract: SoC design is based on Bus Structure. It can resolve the complex system and multi-processor system effectively. NoC applies the computer network technology on system structure, and it will deal with the complex SoC design. The paper presents a new Chip-Only-Writing Bus (COWB) for the communication among the NoC resources. We define the communication protocol and data frame format for COWB. Utilizing MicroBlaze processor as the main resource, DDR2 controller and RS232 controller as the slave resources, the reliability of the COWB is verified with the Xilinx XUPV5 FPGA platform. The testing results show that COWB is easy to expand the number of slave resources and the main resource will access two slave resources simultaneously.

Key words: FPGA, microblaze, network on chip, soc on chip, topology

INTRODUCTION

With the development of semiconductor technology, chip integration has been developed according to Moore law until now. The design method of SoC adopted in the nineties in the last century has been the mainstream technology in IC design. Bus structure technology is adopted in SoC (Steve, 2005), and all functional modules are integrated on the silicon chip. However, with the expansion of modules integrated on chips, especially for the multi-processor system, SoC faces the greatest challenges (Ming-lun and Gao-ming, 2006; Benini and De-Micheli, 2002), First, the address space of SoC is limited so that the system cannot be expanded unlimitedly; Second, two or more functional modules cannot use the bus simultaneously with bus architecture; Third, Globally clock Synchronization raised power consumption of the system enormously. SoC design based on bus structure in the multi-processor system is in a dilemma. To resolve the problem, some research institutes put forward the design method of IC based on NoC. The core idea of NoC is consulting the computer network technology. Bus structure is replaced by router and packet switching, so that the bottleneck of multi-processor communication could be resolved thoroughly (Cesar *et al*., 2002; Jingcao and Radu, 2005). Compared to SoC structure, the qualities of NoC are as follows: The address space is easier to be extended, and functional modules could be integrated infinitely; It owns the capacity of parallel communication so that different functional modules can transmit data simultaneously; It adopts the clock strategy of Globally Asynchronous Locally Synchronous (GALS) so that each switch can work at different time, and asynchronous communication between different switches can reduce the power consumption effectively. NoC structure will be the mainstream technology of the multi-processor system in the future (Hemani *et al*., 2000).

This study presents the composition and principle of NoC structure, and puts forward a communication protocol for Chip-Only-Writing Bus (COWB) which is fit for NoC Resources Communication, then defines the data frame format. The whole design is described in VHDL, and verified on Xilinx XUPV5 development platform.

Noc architecture: The communication among resources could be realized by topology, which is the key technology of NoC design. Common topologies are as follows: Honeycomb cellular structure, 2D-Mesh, 3D-Mesh, Torus, Fat-tree, octagon structure, and Proteo, etc (Xiao-Qiang and Gao, 2008). The structure of 2D-Mesh is simple, and it is also easy to realized by hard logic.

Taking 2D-Mesh as an instance, this paper presents the NoC architecture (Hong-Zhi, 2009; Benini and De-Micheli, 2002), as illustrated in Fig. 1.

In Fig. 1, S represents Switch. Each switch connects to four switches nearby and one local resource. R represents Resource, which sends or receives data. It may be a processor, ASIC, memory, special functional module IP. RNI represents Resource Network Interface, which connects Resource with Switch, and it packs or unpacks data.When resource sends data, the data will be packed to network transmission frame format through RNI, then it will be output to the local switch. According to destination switches which receive data, some routing algorithms will be adopted. Deterministic XY routing algorithm and adaptive DyAD routing algorithm are all easy to be realized, and they can effectively avoid the deadlock phenomenon (Xiao-Qiang and Gao, 2008).

Corresponding Author: Zhang Bo, College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024

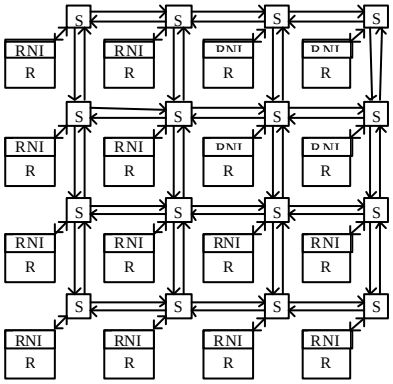

Fig. 1: 2D-Mesh network structure diagram

From Fig. 1, we can see that more than one resources can transmit data in 2D-Mesh network simultaneously, which enhances parallel processing capacity of the system. However, with the expansion of system scale, if each resource contains only one processing unit, then the network topology will be huge, and data transmitting latency will be raised. If each resource contains one processing system which is composed of one main resource and some slave resources, then the function of each resource will be enhanced. The communication between MR and SR may adopt the traditional bus structure, such as: AMBA, Wishbone, CoreConnect, etc. However, with the increase of the number of slave resources, the communication efficiency will be the challenge for the application of the bus structure above. To improve the processing performance of resource system in NoC, this paper presents a new communication protocol for Chip-Only-Writing Bus (COWB), which can make the parallel communication realized between MR and SR.

Design of chip-only-writing bus: The structure of Chip-only-writing Bus (COWB) is illustrated in Fig. 2. In Figure 2, MR represents Main Resource. SR represents Slave Resource. MRNI represents Main Resource Network Interface. SRNI represents Slave Resource Network Interface. Blue real lines are MR exclusive buses, and MR can send data to each SR through the bus. Red short lines are SR shared buses, and each SR can send data to MR through the bus. Black long broken lines connect SRNI with the arbitration module.

Principle of chip-only-writing bus: The system of COWB is composed of one main resource, some slave resources, one arbitration module, the main resource network interface, and slave resource network interfaces. Data sent by MR is packed to data frame through MRNI, and it will be sent to each SRNI through MR exclusive bus. Then each SRNI will receive the data frame and

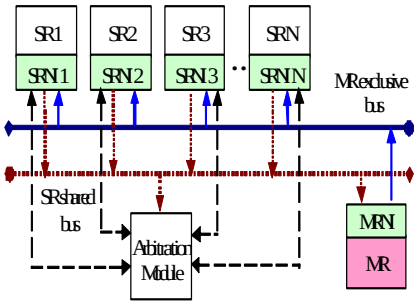

Fig. 2: Chip-only-writing bus structure diagram

detect its destination address. If the destination address is equal to the address which is distributed by the local SR, then the SRNI will receive the data frame and send it to the local SR after unpacking it. If not, SRNI will abandon the data frame. The bus is a one-way writing-only bus, and MR sends data to every SR. When SR sends data to MR, each SR sends a request to the arbitration module through request signal wires which are independent separately. Then the arbitration module will send a response to SR according to the principle that who arrives first who will be serviced first. The request response signal wire is the black long broken line as illustrated in Fig. 2. After SRNI received the response signal, the SR shared bus will be occupied, and the data will be packed by SRNI and then sent to MR.

Data frame format of COWB: The communication data between MR and SR must be transferred according to the data frame format defined, and the data frame format is illustrated in Fig. 3.

DA (Destination address) is the resource address which receives data packets, which occupies 2 bytes and supports 216 resources mostly. SA (Source address) which occupies 2 bytes is the resource address. Type which occupies 1 byte represents that the data frame is a writing data, reading command, or request for sending again. Length field which occupies 2 bytes is the number of bytes of the data. Checking field which adopts CRC checking occupies 4 bytes.

System design: RNI is the communication bridge between the resource and Chip-Only-Writing Bus (COWB). RNI receives the data frame of COWB, and sends it to resources after unpacking it. RNI receives the data sent by resources, and sends it to COWB after packing it to the data frame format. The inner structure of MR is similar to SR, and their difference is that the bus which sends or receives data is different (Fig. 4).

When MR sends data, the data frame is sent to the receiving control module through the COWB interface occupied by MR exclusively. The receiving control module firstly judges the destination address field of the

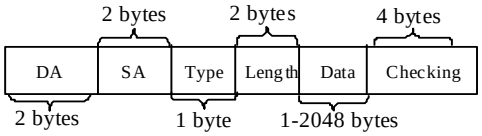

Fig. 3: Data frame structure

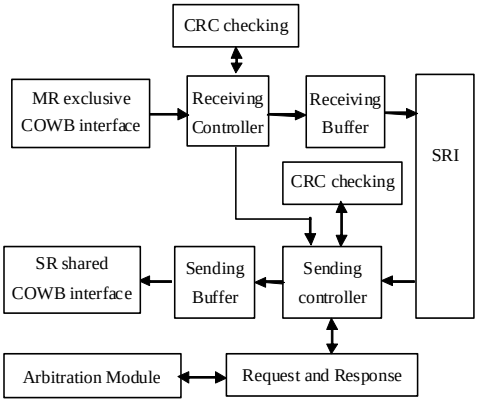

Fig. 4: Resource network interface structure

data frame received. If the destination address is not equal to the local resource address, then the data frame will be abandoned. If two addresses are equal, the receiving control module will receive data continuously, and the effective data of data field will be stored to the reception buffer module. At the same time, the received data will be output to the CRC checking module for checking. After one frame data was all received, the checking value computed by CRC checking module will be compared to the checking data of the data frame. If they are the same, then the data transfer is right. If not, the sending control module will be activated, and a request for sending again will be generated.

When SR sends data, the SRI module transmits the data to the receiving control module, then the receiving control module will pack the data to data frame and send it to the sending buffer module. CRC checking module checks and computes the sending data, then it will transport the checking value to the sending control module. After one data frame packed by sending control module was stored into sending buffer module, the request response module will be activated by the sending control module and will send a request signal to the arbitration module. When it is replied, the SR shared bus could be used. The sending control module activates the sending buffer module, and the data in the buffer module will be sent to COWB through COWB interface module shared by SR. Then the data in the writing-only bus will be read by MRNI, after being unpacked, it will be sent to MR.

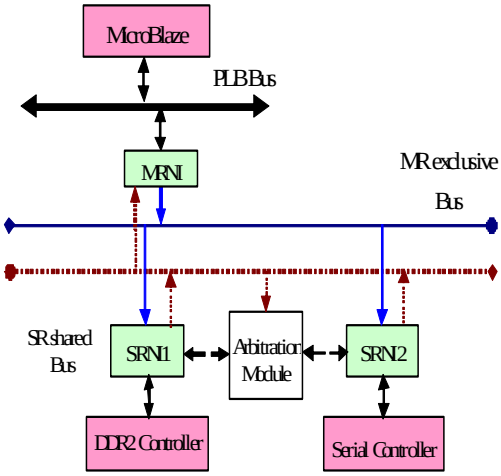

Fig. 5: COWB verify system

Implementation and verification of COWB: According to the module design principle used by the modern EDA, program designs of MRNI module, SRNI module, and the arbitration module are realized in VHDL. The system verification environment makes use of the Micro Blaze processor as MR, and DDR2 SDRA memory controller and RS232 controller as SR. The structure of the verification system is illustrated in Fig. 5.

The Micro Blaze processor of MR visits MRNI through PLB bus. Taking the Micro Blaze processor reading from or writing to DDR2 memory as an instance, we will verify the system. Micro Blaze processor writes data to MRNI through PLB bus, then MRNI will pack the data. The destination address is DDR2 controller address of resource. The data frame will be wrritten to SRNI1 of DDR2 controller through MR exclusive bus, then SRNI1 will unpack the data frame. If CRC checking is correct, the data will be sent to DDR2 controller module of SR. After the data was wrote over, Micro Blaze processor visits MRNI through PLB bus and makes a request for reading data from DDR2 memory. Then MRNI sends the reading request data frame to SRNI1. When SRNI1 receives the request frame, it sends request to the arbitration module. After being replied, the SR shared bus will be occupied, and DDR2 controller reads data from DDR2 memory outside and sends it to SRNI. The data will be packed and be sent to MRNI for unpacking through the bus shared by SR. After being unpacked, the data will be transported to Micro Blaze processor through PLB bus.

MicroBlaze processor can also verify the function of system by visiting to RS232 controller. MicroBlaze processor sends data to MRNI through PLB bus, then the data will be packed by MRNI, then it will be sent to SRNI2 through the MR exclusive bus. Then the data will be unpacked by SRNI2, and it will be sent to the serial

controller of SR. Next, the serial controller will send the data to PC through RS232. Then PC will transport the data sent by the serial module by super terminal print. By comparing, We will know that the data received by super terminal is the data sent by MicroBlaze, so that we can verify the system works right or wrong.

CONCLUSION

This study presented a new communication protocol for resources in NoC structure. We adopted two Chip-only-writing Buses to realize the communication between MR and SR. We defined the data frame format for COWB. We designed MicroBlaze processor as MR, and DDR2 controller and RS232 controller as SR, so that we could verify the system works right or wrong. Compared to the traditional bus on chip, COWB designed by this paper owns the following traits: the structure is simple; the resource is easy to be expanded; two resources can occupy two Chip-only-writing Buses simultaneously, so that two resources could be read or wrote simultaneously. So that the transport speed of data is improved, and the performance of the system get enhanced.

ACKNOWLEDGMENT

This study is supported by the Chinese national natural science fund project (60772101).

REFERENCES

Benini, L. and G. De-Micheli, 2002. Networks on chip: A new SOC paradigm. IEEE Comp., 35(1): 70-80.

Cesar, A.Z., E.K. Marcio and C. Luigi, A. Postula, J. Obey, M. Millberg and D. Lindqvist, 2002. A study on communication issues for system-on-chip. Proceeding of the 15 Symposium on Integrated circuits and Systems design.

Hemani, A., A. Jantsch, S. Kumar, , 2000. Network on a chip: An architecture for billion transistor era. Proceeding of IEEE NorChip Conference, pp: 166-173.

Hong-Zhi, Z., 2009. Study of the impace of switch service performance on 2D mesh network on Chip and Its improved topology. Acta Electronica Sinica, 37(2): 294-298.

Jingcao, H. and M. Radu, 2005. Energy-and performance-aware mapping for regular NOC architectures. IEEE Trans. Computer-Aided Design Integrated Circuits Syst., 24(4): 551-562.

Ming-lun, G. and D. Gao-ming, 2006. Next generation mainstream architecture for intergrated circuits. Microelectronics, 36(4): 461-466.

Steve, F., 2005. Future trends in SOC Interconnect IEEE VLSI-TSA international symposium on VLSI design. Automation Test, 4: 295-298.

Xiao-Qiang, Y. and Y.Gao, 2008. A study of design for network on chip. Microelectronics Comp., 25(7): 176-179.