Design and Implementation of Wishbone Bus Interface Architecture for SoC Integration USING VHDL ON FPGA

Swati R. Mishra[1], Pramod Patil[2], Sudhir Shelke[3]
Research Scholar , Principal, HOD EN/ETC
JDCOEM, Nagpur

Abstract: Wishbone bus is used to communicate variety of devices. While designing SOC, elastic interfacing between varieties of devices is crucial to get extreme performance. Wishbone bus used to interface a variety of devices due to its open structural design and many a free IP core with a Wishbone interface supplied by Open Cores association. Previously, IP cores used non-standard interconnection schemes that made them difficult to integrate. This required the creation of custom stick logic to interface each of the cores together. By adopting a standard interconnection scheme, the cores can be incorporated more quickly and easily by the end user. This pattern can be used for soft core, firm core or hard core IP. Since firm and hard cores are generally conceived as soft cores, the pattern is written from that standpoint.

The WISHBONE interconnect is proposed as a general purpose interface. As such, it defines the standard data exchange between IP core modules. The WISHBONE architects were strongly influenced by three factors. First, there was a need for a good, reliable System-on-Chip integration solution. Second, there was a need for a common interface specification to facilitate structured design methodologies on large project teams. Third, they were impressed by the traditional system integration solutions afforded by microcomputer buses such as PCI bus and VMEbus. A SoC which utilizes ALU master cores and memory slave cores using Wishbone bus interconnection scheme has been designed for this purpose. The final implementations have been done in XILINX FPGA platform. This paper investigates the open core based SOC design stage. Open core uses a standard bus WISHBONE to improve System-On –Chip problem.

***Keywords** – SoC, Wishbone, FPGA, Xilinx, WISHBONE Interface*

*****

I. INTRODUCTION

The Wishbone specification defines the Wishbone bus as the System-on-Chip (SoC) architecture which is a handy edge for use with semiconductor IP cores. It is intended to be used as an internal bus for SoC applications with the aim of alleviating SoC integration problems by nurturing design reuse. It can be concluded that a SoC these days has turned into an IC that implements most or all the functions of a complete electronic system .

The WISHBONE design was strongly influenced by three factors. First, there was a requirement for a good, reliable System-on-Chip integration solution. Second, there was a need for a common interface specification to facilitate structured design methodologies on large project teams. Third, they were impressed by the traditional system integration solutions afforded by microcomputer buses such as PCI bus and VMEbus.

This paper attempts chiefly to present a review of the WISHBONE bus architecture developed by the SILICORE , and making a comparison of its selective features with that of some other popular on-chip standardized bus architectures such as AMBA ,CoreConnect , Avalon , etc. The paper endeavors to evaluate the features like the bus topologies, communication protocols, arbitration methods, bus-widths, and types of data transfers in these SoC organized communication architectures.

II. LITERATURE SURVEY

A few research work have been conducted to explain the concept of wishbone bus Mohandeep Sharma, Dilip Kumar explained the concept of 16–bit Dataflow Interconnection architecture using VHDL language and to make an analytical evaluation with the help of Xilinx 12.3 ISE tool. Bharath.S.V, Ashwini.S.Shivannavar, M.Z.Kurian proposed two level bus linked with two different type of IP cores, with which design-reuse concept can be improved from this architecture and also it gives better reduction of power consumption. The data transfer rate of high speed devices like CPU and memory will increase. Changlei Dongye performed use of two different type of IP core link with the two level bus respectively, the architecture can not only improve the "Design-reuse" it also provide the optimal strategy for reduction of power consumption.. Ahmed H.M. Soliman implemented WISHBONE protocol to the core interface side of network adapters enables designers to use the NoC solution in many SoC designs based on the WISHBONE interface.

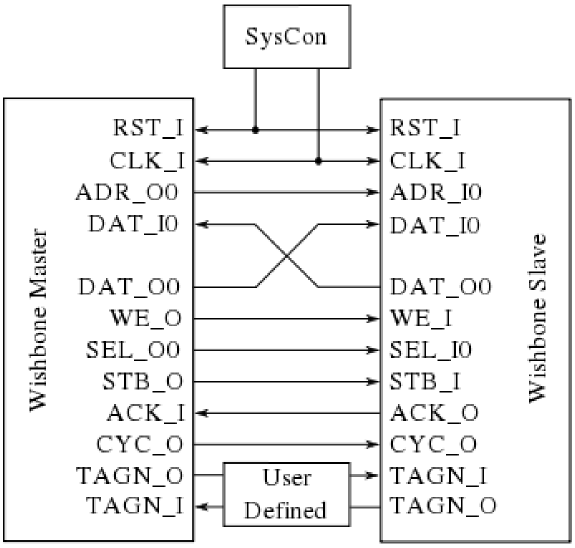
FPG

FPGA means Field Programmable Gate Arrays. It is a semiconductor tool containing programmable logic gears and programmable interconnect. The programmable logic components can be planned to duplicate the functionality of basic logic gates such as AND, OR, XOR, NOT or more complex combinational functions such as decoders or simple mathematical functions. In most FPGAs, these programmable logic components called as IP cores (or logic blocks, in FPGA parlance) also include memory elements, which may be simple flip flops or more complete blocks of memories. A hierarchy of programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer, somewhat like a one-chip programmable breadboard. These logic blocks and interconnects can be programmed after the manufacturing process by the customer/designer (hence the term "field programmable", i.e. programmable in the field) so that the FPGA can perform any logical function is needed. FPGAs are generally slower than their application specific integrated circuit (ASIC) counterparts, as they can't handle as complex a design, and draw more power. However, they have several advantages such as a shorter time to market, ability to re-program in the field to fix bugs, and lower non recurring engineering cost costs. Vendors can sell cheaper, less flexible versions of their FPGAs which cannot be customized after the design is committed. The development of these designs is made on regular FPGAs and then migrated into a fixed version that more resembles an ASIC. Complex programmable logic devices, or CPLDs, are another alternative. FPGA architectures include dedicated blocks such as RAM, hardwired multipliers, multiply-accumulate unit, high-speed clock management circuitry, and serial transceivers, embedded hard processor cores such as PowerPC or ARM, and soft processor cores such as NIOS or Micro blaze.

III. METHODOLOGY

a.DESIGN IMPLEMENTED

Wishbone utilizes master and slave architectures which are connected to each other through an interface called Intercon. Master is an IP core that initiates the data transaction to the slave IP core. Master starts transaction providing an address and control signal to slave. Slave in turn responds to the data transaction with the master with the specified address range. . Intercon is a module that interconnects wishbone MASTER and SLAVE interfaces . Syscon is a wishbone module that drives the system clock [clk_o] and reset [rst_o] signals. The WISHBONE interconnection itself is nothing more than a large, synchronous circuit. It must be designed to logically operate over a nearly infinite frequency range. However, every integrated circuit has physical characteristics that limit the maximum frequency of the circuit.



Transmitter(Master) **Receiver** (Slave)

Fig 1: **Wishbone Bus Interface**

The WISHBONE interconnection itself is nothing more than a large, synchronous circuit. I will be designing it to logically operate over a nearly infinite frequency range. However, every integrated circuit has physical characteristics that limit the maximum frequency of the circuit. This means that a WISHBONE compatible circuit will theoretically function normally at any speed, but that its maximum speed will always be limited by the process technology of the integrated circuit. It can interconnect between soft-core, solid-core and hard-core. Wishbone bus structure is relatively simple, it just defines a high-speed bus and offers four different IP interconnect structure namely point to point, dataflow, shared bus and cross bar switch architectures.

b.*SOFTWARE Implementation*

Figure 1- shows a simple 16-bit WISHBONE Bus . This port has 8-bit granularity, which means that data can be transferred 8 or 16-bits at a time. Bus width can be extended to 16,32 and 64 bits. Its means that the 8-bit register can be accessed with either 8 or 16-bit bus cycles. This is consummate by selecting the high or low byte of data with the select lines [SEL_I(1..0)]. When [SEL_I(0)] is asserted, the low byte is accessed. When [SEL_I(1)] is asserted, the high byte is accessed. When both are asserted, the entire 16-bit word is accessed.

Select lines are used for deciding the granularity of data whether we want to transfer upper 8 bits, lower 8 bits or in all 16 bits. Address lines are used to select the slave. As in my design ,I have used 8 slaves which will be controlled by one Master. So 3 bit address line will select the wanted slave like If in address line we have 001,slave 1 will be selected,for slave 2 address lines will be 010 and so on.

ISSN: 2321-8169
1847 – 1850

Simulating the written VHDL code in ISE simulator and verifying the waveforms generated by the simulator. If the required simulated output is not achieved, then the VHDL code is checked and necessary corrections are made. The required waveforms are noted down as a reference to the synthesis stage or at the final stage. The waveforms are presented subsequently

c. *HARDWARE IMPLEMENTATION*

An altera Cyclone II FPGA device is used for this implementation .The system is tested for transmission of data through wishbone bus.The toggle switches are used as an input to the system.An internal clock pulse of 27MHz is used which is designated as pin D13 in DE2 board. All the VHDL module files are integrated and a top file is selected to run the whole implementation. Implementation compresses of different stages like translate, mapping. A programming file is generated with extention name .SOF which creates a bit format file, which is dumped on the altera Cyclone II FPGA using Xilinx (software) through a USB Blaster. On the hardware, if the design works according requirement, then design is corrected, if any error occurs then reconsider the design.

IV CONCLUSION

Thus I had implemented the design of 8–bit Peer to peer Interconnection architecture using VHDL language and to make an analytical evaluation with the help of Xilinx 9.1 ISE tool. I had implemented one master with 8 slaves communicating via FPGA so as to make a little twist with the standard bus. The synthesis part has been performed on Xilinx ISim Simulator and the Altera D2E Board Cyclon II family FPGA. The results shown below depicts that Peer to peer interconnection which transmits data in a direct manner between Master and Slaves IP cores .The proposed design utilizes only 207 slices for implementation on FPGA. As this Wishbone interface requires a very little logic overhead to implement the entire interface, it gives rise to a highly portable system design that may work with standard logic primitives available in most of the FPGA and ASIC devices. In addition to this, the design also supports an operating frequency of more than 100 MHz. These results project it as a High Speed and Less Area circuit capable of implementation in VLSI Design on account of its Low Latency and High Efficiency [13]. Hence Low cost, portable and Time to market SoC can be designed successfully using Wishbone Bus Dataflow Interface Architecture at the industrial scale.

Following are the device utilization summary:
Number of Slices: 131 out of 6144
Number of Slice Flip Flops:162 out of 12288
Number of 4 input LUTs:120 out of 12288
Number of IOs: 24
Number of bonded IOBs: 24 out of 240
Number of GCLKs:1 out of 32

Thus I had implemented the WISHBONE BUS XILINX ISE 9.1. The RTL VIEW and Simulation waveforms is also shown below

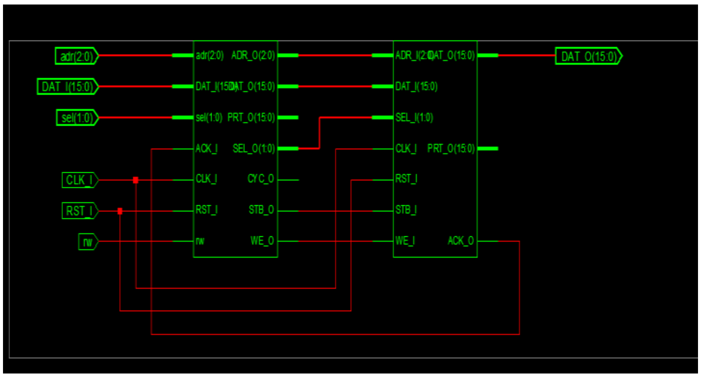


Fig 2 RTL ***View Of Complete System***

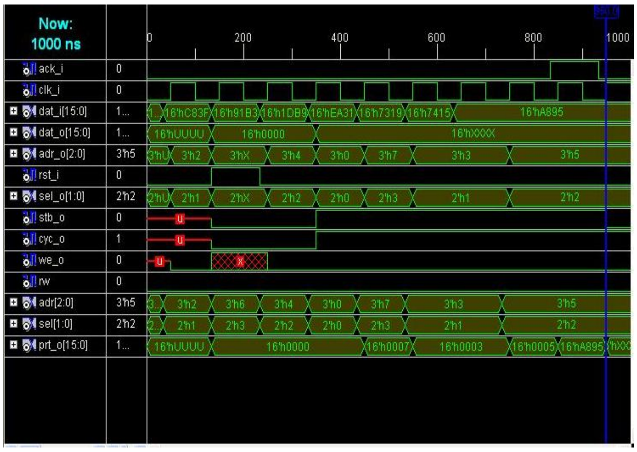


Fig:3 ***MASTER Simulation result***

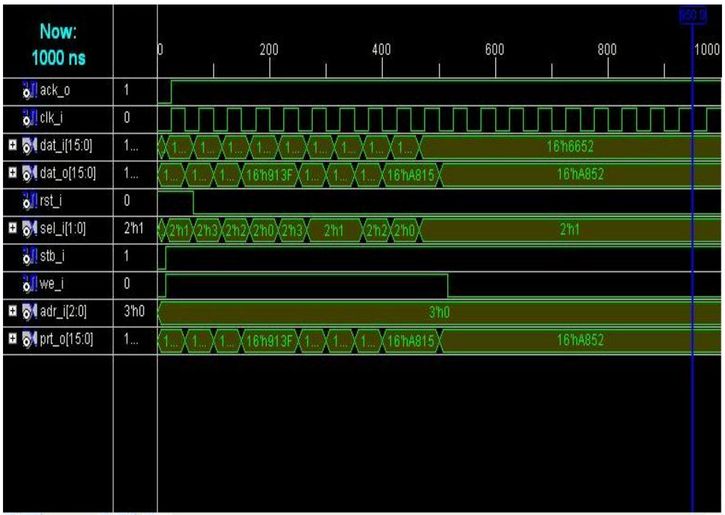


Fig:4 ***SLAVE SIMULATION RESULT***

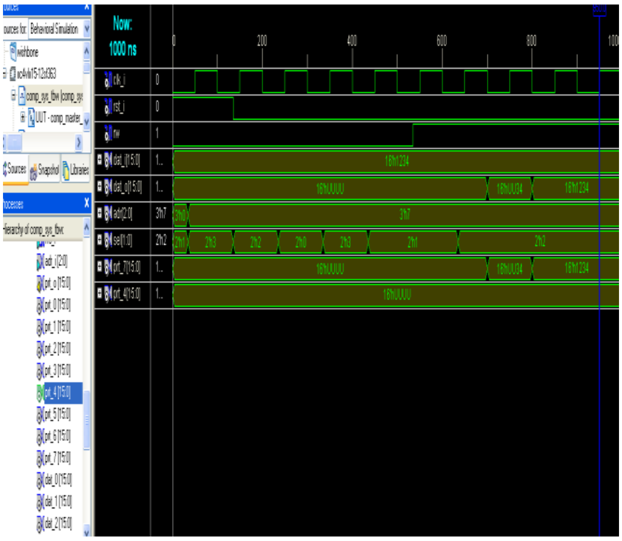


*Fig 5: **Complete simulation result***

V REFERENCES

[1] Mohandeep Sharma, Dilip Kumar "Design and Synthesis of Wishbone Bus Dataflow Interface Architecture for SoC Integration"IEEE 2012.

[2] Richard Herveille, WISHBONE System-on- Chip (SoC) Interconnection Architecture for Portable IP Cores, rev. version: B4, 2010. By Open Cores Organization, p.7, 2010.

[3] Bharath.S.V, Ashwini.S.Shivannavar, M.Z.Kurian "Design of Efficient SOC Bus Based on WISHBONE" (IJITEE) ISSN: 2278-3075, Volume-3, Issue-1, June 2013

[4] [4] Changlei Dongye "Design of the On-chip Bus Based on Wishbone" 2011 IEEE

[5] Ahmed H.M. Soliman, E.M. Saad, M. El-Bably and Hesham M. A. M. Keshk, " Designing a WISHBONE Protocol Network Adapter for an synchronous Network-on-Chip" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011

[6] Qian Min, Zhu Jing, Cao Yunpeng, Yang Cuijun, "Design and FPGA Verification of Dual- speed Adaptive Ethernet Controller" IEEE 2011

[7] Cao Fan*, Chen Lan, and Yi Bo, "Designing WISHBONE to AMBA Wrapper" IEEE 2009

[8] "A simple design with Wishbone bus" http://en.wikipedia.org/wiki/Wishbone_(computer_bus)

[9] Vani.R.M and M.Roopa "Design of AMBA Based HB2APB Bridge" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.11, Nov 2010

[10] Jaehoon Song, Student member, IEEE, Hyunbean Yi,Member,IEEE, Juhee Han, and Sungju Park, Member, IEEE,"An Efficient SOC Test Technique by Reusing On/Off-Chip Bus Bridge"IEEE Transcactions on Circuits and Systems-I: Regular Papers, Vol,56,No.3,March2009

[11] Andreas Ehliar, Dake Liu "An Fpga Based Open Source Network-On-Chip Architecture" IEEE 2007

[12] W. Dally and B. Towles, "Principles and Practices of Interconnection Networks." Morgan Kaufmann, 2004.

[13] K. Goossens, J. Dielissen, and A. Radulescu, "Aethereal network on chip: concepts, architectures, and implementations," Design & Test of Computers, IEEE, vol. 22.

[14] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," Circuits and Systems Magazine, IEEE, vol. 4, 2004.

[15] Balal Ahmad, "Communication Centric Platforms for Future High Data Intensive Applications,"Ph.D. Thesis: The University of Edinburgh, p.25, March 2009.

[16] Anurag Shrivastava, G.S. Tomar, and Ashutosh Kumar Singh, "Performance Comparison of AMBA Bus-Based System-On-Chip Communication Protocol," International Conference on Communication Systems and Network Technologies, IEEE Computer Society, p. 449, 2011.

[17] Milica Mitic and Mile Stojcv, "A Survey of Three System-on-Chip Buses: AMBA, CoreConnect and Wishbone," ICEST 2006.

[18] Ho W.H. and Pinkston T.M., "A Design Methodology for Efficient Application-specific on-chip interconnects," IEEE Trans. Parallel Distributed System, 17(2), pp. 174-179, February 2006.

[19] Kyeong Keol Ryu, Shin, and Vincent J. Mooney, "A Comparison of Five Different Multi processor SoC Bus Architectures," Digital Systems Design: Proceedings. Euro micro Symposium, Warsaw, Poland, p.209, Sept. 2001.

[20] D. A. K. K. Sylvester, "A global wiring paradigm for deep submicron design", IEEE Trans., Vol. 19, No. 2, 2000, pp. 242 – 252.

[21] H. Scott, B. Steven, B. Gaetano, and E. Carl, "An FPGA for Implementing synchronous Circuits", IEEE Des. Test, Vol. 11, No. 3, 1994, pp. 60-69.

[22] J. N. Lassen, "FPGA prototyping of asynchronous Networks-on-Chip," M.Sc. thesis, Technical University of Denmark, DTU, 2008.

[23] D. Park, "Design of High-performance, Energy Efficient, and Reliable NoC Architectures", Ph.D. thesis, Pennsylvania State University, 2008.

[24] T. Bartic, J.-Y. Mignolet, V. Nollet, T. Marescaux, D. Verkest, S. Vernalde, and R. Lauwereins, "Highly scalable network on chip for reconfigurable systemsin," System-on-Chip, 2003. Proceedings. International Symposium on, 2003.

[25] http://www.opencores.org/, 2002