A Literature Review on Wishbone Bus Technique for Network on Chip Architecture

Mr. Arjun M
M.Tech (VLSI Design & EmbeddedSystems)
Dept. Of E C E,
R.V.College Of Engineering,
Bengaluru, India

Mr. Kendaganna Swamy
Asst.Professor,
Dept. Of E C E,
R.V.College Of Engineering,
Bengaluru, India

Dr. Uma.B.V
HOD & Professor ,
Dept. Of E C E,
R.V.College Of Engineering
RIndia
Bengaluru, India

ABSTRACT

The Network-on-Chip has been recognized as a paradigm to solve System-on-Chip (SoC) design challenges. The bus technologies is one of key researches of a NoC design. This is a review paper which gives an overview of different bus computer architectures (CAs) , called AMBA, Coreconnect and Wishbone. It starts with a brief introduction Network On Chip (NoC) , then looks at bus organizations , and concludes with a discussion related to a comparative performance analysis and also gives merits and demerits of all three bus architectures.

Keywords

SoC buses, NoC, AMBA, Coreconnect, Wishbone

1. INTRODUCTION

As chip manufacturing techniques continue to improve, it involves the design of more complex systems. Designing such a complex system cannot be done easily and much research from both academia and industry is required. With the evolution of scaling down of process technologies and increasing design sizes has led to highly complex billion - transistor integrated circuits (ICs).System-on-Chip (SoC) technology is the ability to place multiple function Intellectual property (IP) blocks- in a single silicon chip . As the number of On – Chip components in an SoC increase in number, so do their communication between them also becomes complex, this gives way for a “Network” for the On – Chip components to communicate between themselves and also to the external world known as Network On Chip (NoC).

2. ON - CHIP COMMUNICATION INTERFACE

For driving or receiving data or power from a bus it requires an electronic circuit which is known as Bus interfacing. The On Chip communication interface for Wishbone bus architecture can be classified as : Point to point Interconnection, Shared bus Interconnection , Dataflow Interconnection and Crossbar switch Interconnection.

2.1. Point to Point Interconnection

In Point to point bus interface then communication between cores takes place through some handshake protocol. In this interface only one master can connect to one slave at any given point of time. The traffic in the cores is controlled by the handshaking signals. Since it supports only single master and single slave interface, it is not suitable for SoC interconnection.

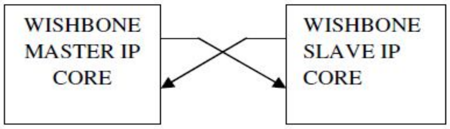


Figure – 1 Point to Point Interconnection

2.2. Shared Bus Interconnection

In Shared bus Interconnection it supports the architecture of many master and slaves. But however at any given point of time only one master can use the bus, while the other masters have to wait for their turn. An arbiter has the control as to which master can use the bus at that moment.

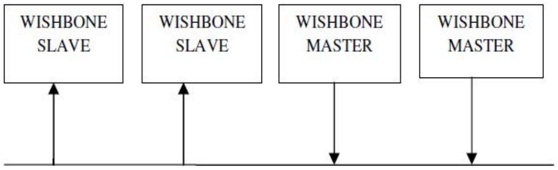


Figure – 2 Shared Bus Interconnection

2.3 Crossbar Switch Interconnection

In Crossbar switch interconnection it has the concept where one master can access simultaneously multiple slaves which can be used in multicore SoCs. There are many methods in crossbar switch interconnection for which data can be transferred between masters and slaves.

It has a higher data rate compared to shared bus. It also has an arbiter to get control over the bus which decides whether which master has to communicate with which slave. Therefore two or more masters can communicate with slaves at the same time, as long as it isn't the same slaves.

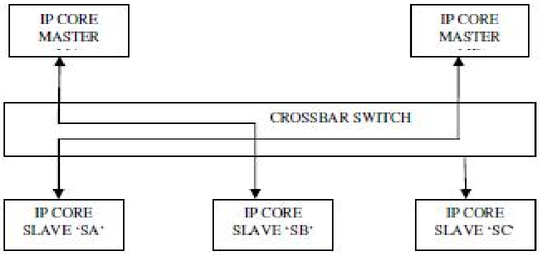


Figure – 3 Crossbar Switch Interconnection

2.4 Dataflow Interconnection

In Dataflow interconnection the data is processed in a sequential manner. Each IP core in the architecture has both MASTER and SLAVE interface. An IP Core acts as a master to the next IP Core in the sequential chain and as a slave to the IP Core prior to it, this is called as pipelining. The execution time is very fast and this architecture also exploits parallelism.

Handshaking signals control the traffic. This is mainly useful in the implementation of DSP algorithms.

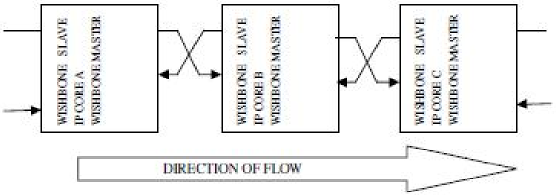


Figure – 4 Data Flow Interconnection

3. SoC BUSES OVERVIEW

In this section the focus is mainly on the three SoC buses namely AMBA , Coreconnect and Wishbone. Because of space crunch only the selective and more distinctive features of every one of these buses is discussed below.

3.1 AMBA

AMBA stands for *Advanced Microcontroller Bus Architecture* and the bus standard has been devised by ARM to support on – chip communications among the processor IP cores which are manufactured by them. AMBA bus systems are used in high performance SoC design. The major issue concerning in SoC nowadays is not only the housing of components or blocks but also the way how they are interconnected. AMBA is hierarchically divided into two bus segments namely the System bus and Peripheral bus which are mutually connected through a bridge which serves to buffer data and operations between them. The three distinct bus types specified in the AMBA bus are ASB, AHB and APB.

3.1.1 Advanced System Bus (ASB)

It is the first generation bus of the AMBA system bus series which is used for simple cost effective designs which supports burst transfer, multiple bus masters and pipelined transfer operation.

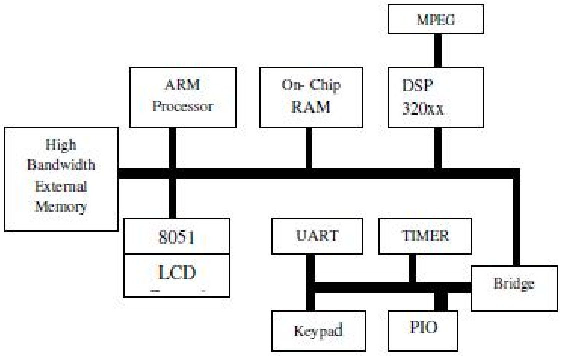


Figure – 5 AMBA based System Architecture

3.1.2 Advanced Peripheral Bus (APB).

This bus is used to connect low speed low power peripheral devices. The bridge is a peripheral bus master, while all the bus devices i.e. timer, UART, PIA etc. are slaves. APB is a static bus which provides simple addressing with latched addresses and control signals for easy interfacing. Recently, two new specifications Multi – layer AHB and AMBA AXI have been added and defined under the AMBA bus system. The Multi-layer AHB provides more flexible interconnect architecture w.r.t AMBA AHB which helps to keep the AHB protocol unchanged. And on the other hand, AMBA AXI is based on the concept of point – to – point interconnection.

3.1.3 Advanced High- Performance Bus (AHB)

It is a later generation of AMBA bus which is intended for high performance high clock synthesizable designs. It supports multiple bus masters operation, peripheral and burst transfers, split transactions, wide data bus configurations, and non tristate implementations. The main constituents of the AHB are Master, slave, arbiter and decoder.

3.2. CoreConnect

CoreConnect is an on-chip bus which was developed by IBM. By the reuse of processor, subsystem and peripheral cores, supplied from different sources, it enables their integration into a single VLSI design. It is basically heirarchically organized architecture. It comprises of three buses which provide a efficient intercnnection of cores, library macros and custom logic within a SoC. The three buses are PLB, OPB and DCR.

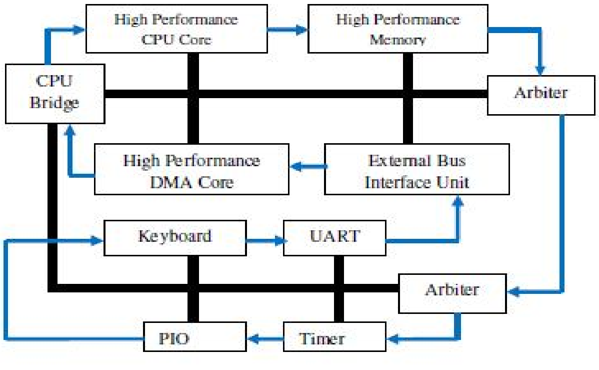

High Performance CPU Core
High Performance Memory
CPU Bridge
Arbiter
High Performance DMA Core
External Bus Interface Unit
Keyboard
UART
Arbiter
PIO
Timer

Figure – 6 CoreConnect Bus Architecture

3.2.1 Processor Local Bus (PLB)

This is the main system bus. It is synchronous, multi-master, centrally arbitrated bus capable of achieving high-performance and low-latency on-chip communication. Separate address bus and data bus is supported by concurrent read and write transfers. A glue logic, PLB macro is used to interconnect various master and slave macros. Arbitration unit supports upto 16 masters, while there are no restrictions in the number of slave devices.

3.2.2 On – Chip Peripheral Bus (OPB)

This is designed to connect low speed, low throughput peripherals i.e. serial and parallel port, UART, etc. The important features of OPB are fully synchronous operation, dynamic bus sizing, separate address and data buses, multiple OPB bus masters, single cycle transfer of data between bus masters, single cycle transfer of data between OPB bus master and OPB slaves, etc. OPB is implemented as multi-master, arbitrated buses. Instead of tristate drivers OPB uses distributed multiplexer.

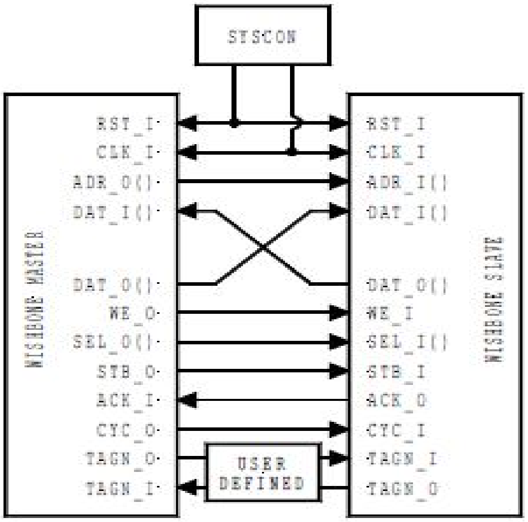
3.2.3 Device Control Register Bus (DCR)

This is a single master bus which is used as an alternative for two reasons. First, passing status and setting configuration information into the individual device-control registers between the Processor Core and others SoC constituents such as Auxilliary Processors, On-Chip Memory, System Cores, Perhipheral Cores, etc. Second, it is used as a design for testability purposes. The arbitration is on the basis of static priority, and with programmable priority fairness.

3.3. Wishbone

Wishbone architecture was developed by Silicore Corporation. OpenCores organization are the promoters of open IP cores development. In August 2002, it put Wishbone architecture into public domain which means that Wishbone is not copyrighted and can be freely copied and distributed. Wishbone establishes common interface standards for data exchange between modules within an integrated circuit chip.

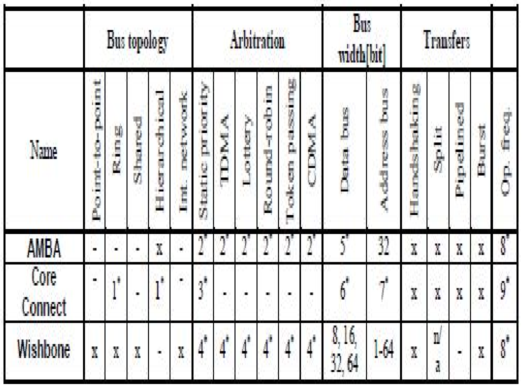
The Wishbone comprises of two types of interfaces known as the Master and Slave. Master interfaces are IPs which initiate the bus transfer, while slave interfaces are capable of accepting bus cycles.



Wishbone supports three types of bus cycles. They are Single Read/Write cycles, Block Read/Write cycles and Read-Modify-Write cycles.

Table – 1 WISHBONE Signals

Signal	Name	Optional	Description
Acknowledged Out	ACK_O	No	Acknowledged signal from the Master to Slave
Address Out/In	ADR_O/I()	No	Address Array
Clock In	CLK_I	No	System Clock for Wishbone Interface.
Error In/Out	ERR_I/O	Yes	Indicates an abnormal cycle termination occurred.
Data In/Out	DAT_I/O	No	Data input/output array, used to send/receive data.
Write Enable In	WE_I	No	Read or Write Signals. If asserted it is Write signal otherwise Read Signal.
Strobe In	STB_I	No	Indicates that the slave is selected The slave asserts either ACK_O, ERR_O
Strobe Out	STB_O	No	Handshaking Signal.
Address tag Out	TGA_O()	Yes	Contains Information about the address array.
Cycle tag type Out	TGC_O()	Yes	Contains Information about the transfer cycle.
Write Enable Out	WE_O	No	Shows if the transfer cycle is a Read or Write cycle.
Reset	RST_I	No	Reset signal



	Bus topology					Arbitration						Bus width[bit]		Transfers				
Name	Point-to-point	Ring	Shared	Hierarchical	Int. network	Static priority	TDMA	Lottery	Round-robin	Token passing	CDMA	Data bus	Address bus	Handshaking	Split	Pipelined	Burst	Op. freq.
AMBA	-	-	-	x	-	2*	2*	2*	2*	2*	2*	5*	32	x	x	x	x	8*
Core Connect	-	1*	-	1*	-	3*	-	-	-	-	-	6*	7*	x	x	x	x	9*
Wishbone	x	x	x	-	x	4*	4*	4*	4*	4*	4*	8, 16, 32, 64	1-64	x	n/a	-	x	8*

Exceptions for Table 1: 1* Data lines shared, control lines point-to-point ring; 2* Application specific except for APB which requires no arbitration; 3* Programmable priority fairness; 4* Application specific, arbiter can be designed regarding to the application requirements; 5* For AHB and ASB bus width is 32, 64, 128 or 256 byte, for APB 8, 16 or 32 byte; 6* For PLB bus width is 32, 64, 128 or 256 byte, for OPB 8, 16 or 32 byte and for DCR 32 byte; 7* For PLB and OPB bus width is 32 byte, and for DCR 10 byte; 8* User defined operating frequency; 9* Operating frequency depending on PLB width

4. CONCLUSION

A survey of the Wishbone bus and its comparison with two other buses AMBA from the ARM, CoreConnect from the IBM reveals that in terms of compared performance parameters, the WISHBONE bus tends to gain an upper edge over the other three types because it provides for connecting circuit functions together in a way that is simple, flexible and portable due to its synchronous design. The Wishbone bus differs from other buses over the issues of registration before use, in offering Support and Development Tools, in terms of designing one's own Libraries for plug-and-play logic utilization, in terms of Bus Architecture and Transfer Cycles because Wishbone offers Read-Modify-Write (RMW) transfer that none of the other bus architectures does. At the end, this paper endorses the view held by Rudolf Usselmann that it would be a wise choice to adopt Wishbone as a primary interface to our cores because its signaling appears to be very intuitive and should be easily adopted to the other interfaces when needed.

REFERENCES

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

[2] Bernini L. and De Micheli G., Networks on Chips. A New Paradigm for Component-based MpSoC Design, chapter 3 in Jerraya A.A. and Wolf W., (Eds.), **Multiprocessor Systems-on-Chips**, Elsevier, Amsterdam, pp. 49–80, 2005.

[3] 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.

[4] Richard Herveille, WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores, Open Cores Organi zation, rev. version: B4,2010.

[5] IBM Corp., USA," CoreConnect bus Architecture," 1999.

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

[7] 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.