A Survey of Three System-on-Chip Buses: AMBA, CoreConnect and Wishbone

Milica Mitić and Mile Stojčev

Abstract - **This article gives an overview of three popular bus organized computer architectures (CAs), called AMBA, CoreConnect and Wishbone. It starts with a brief introduction to on-chip CA, then looks at bus organizations, and concludes with a discussion related to a comparative performance analysis of all three CAs.**

Keywords – **SoC buses, on-Chip bus, AMBA, CoreConnect, Wishbone**

I. INTRODUCTION

Shrinking process technologies and increasing design sizes have led to highly complex billion-transistor integrated circuits (ICs). As a consequence, manufacturers are integrating increasing numbers of components on a chip. A heterogeneous system-on-a-chip (SoC) might include one or more programmable components such as general purpose processors cores, digital signal processor cores, or application-specific intellectual property (IP) cores, as well as an analog front end, on-chip memory, I/O devices, and other application specific circuits [1].

On-chip bus organized CA is among the top challenges in CMOS SoC technology due to rapidly increasing operation frequencies and growing chip size. Usually, IP cores, as constituents of SoCs, are designed with many different interfaces and communication protocols. Integrating such cores in a SoC often requires insertion of suboptimal glue logic. Standards of on-chip bus structures were developed to avoid this problem. Currently there are a few publicly available bus architectures from leading manufacturers, such as CoreConnect from IBM [2], AMBA from ARM [3], SiliconBackplane from Sonics [4], and others. This paper focuses on SoC CAs providing a survey of three popular bus organized CAs, called AMBA, CoreConnect and Wishbone from an industrial and research viewpoint.

II. ON-CHIP COMMUNICATION ARCHITECTURES

A. Background

The design of on-chip CAs addresses the following three issues [5]:

1. **Definition of CA topology** - defines the physical structure of the CA. Numerous topologies exist, ranging from single shared bus to more complex architectures such as bus hierarchies, token ring, crossbar, or custom networks.
2. **Selection and configuration of the communication protocols** - for each channel/bus in the CA, communication protocols specify the exact manner in which communication transaction occur. These protocols include arbitration mechanisms (e.g. round robin access, priority-based selection [2], [3], time division multiplexed access [4], which are implemented in centralized or distributed bus arbiters.
3. **Communication mapping** - refers to the process of associating abstract system-level communications with physical communication paths in the CA topology [5].

B. Topologies

In respect to topology on-chip communication architectures can be classified as:

Shared bus: The system bus is the simplest example of a shared communication architecture topology and is commonly found in many commercial SoCs [6]. Several masters and slaves can be connected to a shared bus. A block, bus arbiter periodically examines accumulated requests from the multiple master interfaces, and grants access to a master using arbitration mechanisms specified by the bus protocol.

Hierarchical bus: this architecture consists of several shared busses interconnected by bridges to form a hierarchy. SoC components are placed at the appropriate level in the hierarchy according to the performance level they require. Low-performance SoC components are placed on lower performance buses, which are bridged to the higher performance buses so as to not burden the higher performance SoC components. Commercial examples of such architectures include the AMBA bus [3], CoreConnect [2]. Transactions across the bridge involve additional overhead, and, during the transfer, both buses remain inaccessible to other SoC components. Hierarchical buses offer large throughput improvements over the shared busses due to: (1) decreased load per bus; (2) the potential for transactions to proceed in parallel on different buses; and (3) multiple ward communications can be preceded across the bridge in a pipelined manner [5].

Ring: in numerous applications, ring based applications are widely used, such as network processors, ATM switches [2], [5]. In a ring, each node component (master/slave) communicates using a ring interface, usually implemented by a token-pass protocol.

C. On-Chip communication protocols

Communication protocols deal with different types of resource management algorithms used for determining access right to shared communication channels. From this point of view, in the rest of this section, we will give a brief comment

Milica Mitić and Mile Stojcev are with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Nis, Serbia and Montenegro, E-mail: [milicam, stojcev]@elfak.ni.ac.yu

related to the main features of the existing communication protocols, which are:

Static-priority: employs an arbitration technique. This protocol is used in shared-bus communication architectures. A centralized arbiter examines accumulated requests from each master and grants access to the requesting master that is of highest priority. Transactions may be of non-preemptive or preemptive type. AMBA and CoreConnect use this protocol [3], [2].

Time Division Multiple Access (TDMA): the arbitration mechanism is based on a timing wheel with each slot statically reserved for unique master. Special techniques are used to alleviate the problem of wasted slots. Sonics uses this protocol [4].

Lottery: a centralized lottery manager accumulates request for ownership of shared communication resources from one ore more masters, each of which is, statically or dynamically, assigned a number of "lottery tickets" [7].

Token passing: this protocol is used in ring based architectures. A special data word, called token, circulates on the ring. An interface that receives a token is allowed to initiate a transaction. When the transaction completes, the interface releases the token and sends it to the neighboring interface. For example, VCI uses this protocol [8]

Code Division Multiple Access (CDMA): this protocol has been proposed for sharing on-chip communication channel. In a sharing medium, it provides better resilience to noise/interference and has an ability to support simultaneously transfer of data streams. But this protocol requires implementation of complex special direct sequence spread spectrum coding schemes, and energy/battery inefficient systems such as pseudorandom code generators, modulation and demodulation circuits at the component bus interfaces, and differential signaling [9].

III. SoC BUSES OVERVIEW

In the sequel an overview of the more relevant SoC CAs (AMBA, CoreConnect and Wishbone) will be given. Due to space limitation the discussion will be focused on describing the more distinctive features of each of them.

A. AMBA

AMBA (*Advanced Microcontroller Bus Architecture*) [3], [10], is a bus standard devised by ARM with aim to support efficient on-chip communications among ARM processor cores. Nowadays, AMBA is one of the leading on-chip busing systems used in high performance SoC design. AMBA (see Fig. 1) is hierarchically organized into two bus segments, system- and peripheral-bus, mutually connected via bridge that buffers data and operations between them. Standard bus protocols for connecting on-chip components generalized for different SoC structures, independent of the processor type, are defined by AMBA specifications. AMBA does not define method of arbitration. Instead it allows the arbiter to be designed to best suit the applications needs. The three distinct buses specified within the AMBA bus are:

- **ASB (*Advanced System Bus*) -** first generation of AMBA system bus used for simple cost-effective designs that support burst transfer, pipelined transfer operation, and multiple bus masters.
- **AHB (*Advanced High-performance Bus*)** – as a later generation of AMBA bus is intended for high performance high-clock synthesizable designs. It provides high-bandwidth communication channel between embedded processor (ARM, MIPS, AVR, DSP 320xx, 8051, etc.) and high performance peripherals/ hardware accelerators (ASICs MPEG, color LCD, etc), on-chip SRAM, on-chip external memory interface, and APB bridge. AHB supports multiple bus masters operation, peripheral and burst transfer, split transactions, wide data bus configurations, and non tristate implementations. Constituents of AHB are: AHB-master, slave-, arbiter-, and –decoder.
- **APB (*Advanced Peripheral Bus*** – is used to connect general purpose low-speed low-power peripheral devices. The bridge is peripheral bus master, while all buses devices (Timer, UART, PIA, etc) are slaves. APB is static bus that provides a simple addressing with latched addresses and control signals for easy interfacing.

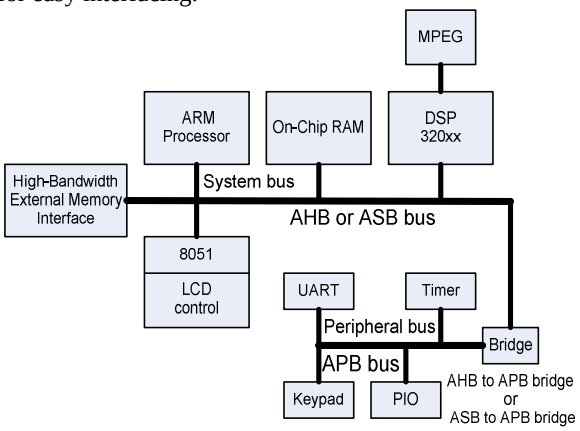

Figure 1 AMBA based system architecture

Recently, two new specifications for AMBA bus, Multi-Layer AHB and AMBA AXI, are defined. [11], [12]. Multi-layer AHB provides more flexible interconnect architecture (matrix which enables parallel access paths between multiple masters and slaves) with respect to AMBA AHB, and keeps the AHB protocol unchanged. AMBA AXI is based on the concept point-to-point connection.

Goog overview papers related to AMBA specifications are references [11], [12] and [13].

B. CoreConnect

CoreConnect [2] is an IBM-developed on-chip bus. By reusing of processor, subsystem and peripheral cores, supplied from different sources, it enables their integration into a single VLSI design. CoreConnect is hierarchically organized architecture. It is comprised of three buses that provide an efficient interconnection of cores, library macros, and custom logic within a SoC (see Fig. 2).

- **PLB (*Processor Local Bus*) –** is the main system bus. It is synchronous, multi-master, central arbitrated bus that allows achieving high-performance and low-latency on-chip

communication. Separated address, and data buses support concurrent read and write transfers. PLB macro, as glue logic, is used to interconnect various master and slave macros. Each PLB master is attached to the PLB through separate addresses, read-data and write-data buses, and other control signals. PLB slaves are attached to PLB through shared, but decoupled, address, read data, and write data buses. Up to 16 masters can be supported by the arbitration unit, while there are no restrictions in the number of slave devices [10].

- **OPB (*On-chip Peripheral Bus*) -** is optimized to connect lower speed, low throughput peripherals, such as serial and parallel port, UART, etc. Crucial 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. PLB masters gain access to the peripherals on the OPB bus through the OPB bridge macro. The OPB bridge acts as a slave device on the PLB and a master on the OPB.
- **DCR bus (*Device Control Register bus*)** – is a single master bus mainly used as an alternative relatively low speed datapath to the system for: (a) 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; and (b) design for testability purposes. DCR is synchronous bus based on a ring topology implemented as distributed multiplexer across the chip. It consists of 10-bit address bus and 32-bit data bus. CoreConnect implements arbitration based on a static priority, with programmable priority fairness.

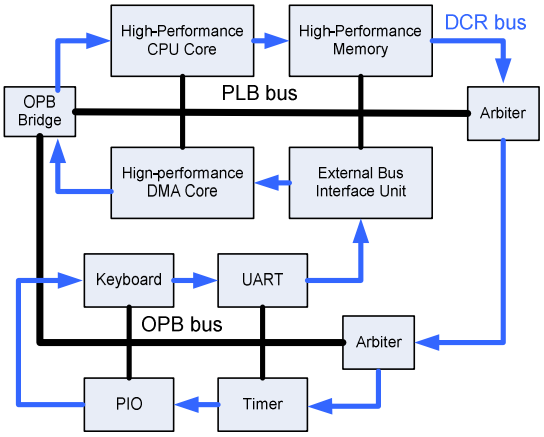

Figure 2 CoreConnect bus based system

C. Wishbone

Wishbone [14] bus architecture was developed by Silicore Corporation. In August 2002, OpenCores (organization that promotes open IP cores development) put it into the public domain. This means that Wishbone is not copyrighted and can be freely copied and distributed.

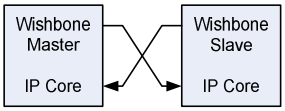

Figure 3 Point to point interconnection

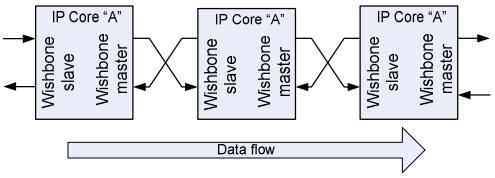

Figure 4 Dataflow interconnection

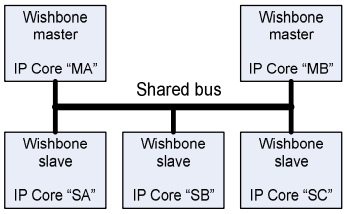

Figure 5 Shared bus

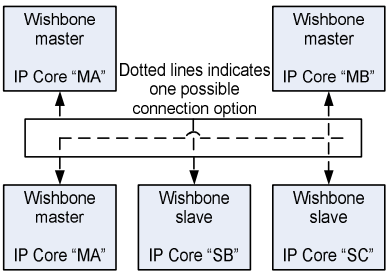

Figure 6 Crossbar switch interconnection

The Wishbone defines two types of interfaces, called master and slave. Master interfaces are IPs which are capable of initiating bus cycles, while slave interfaces are capable of accepting bus cycles [10]. The hardware implementations support various types of interconnection topologies such as:
point-to-point connection (Figure 3) - used for direct connection of two participants that transfer data according to some handshake protocol

a) dataflow interconnection (Figure 4) - used in linear systolic array architectures for implementation of DSP algorithms
b) shared bus (Figure 5)- typical for MPSoCs organized around single system bus
c) crossbar switch interconnection (Figure 6) - usually used in MPSoCs when more than one masters can simultaneously access several different slaves. The master requests a channel on the switch, once this is established, data is transferred in a point-to-point manner.

The Wishbone supports different types of bus transactions, such as read/write, implementing blocking/unblocking access. A Read-Modify-Write transfer is also supported.
Wishbone doesn't define hierarchical buses. In applications where two buses should exist, one slow and one fast, two separated Wishbone interfaces could be created.
Designer can also choose arbitration mechanism and implements it to best fit the application needs.

TABLE 1: SOC BUSES FEATURES OVERVIEW

	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

IV. COMPARISON OF SOC BUS

In Table 1 are given some common features for presented SoC buses, such as topology, arbitration, transfers, and bus width. All presented buses are synchronous.

AMBA and CoreConnect are hierarchical buses. Wishbone does not defines hierarchical bus interconnection, but allows various other possible interconnections, such as point-to-point, ring, unilevel shared bus, crossbar switch interconnection, etc.

Arbitration method for AMBA and Wishbone is application specific, which means that arbiter can be designed regarding to the application requirements. CoreConnect defines static priority.

Presented SoC buses support various transfer types. All support handshaking, split transfer and burst transfer, while pipelined transfer support AMBA and CoreConnect, but not Wishbone.

Address and data bus width are configurable. For AMBA and CoreConnect data bus width depends on type of the bus (for AHB and ASB bus width is 32, 64, 128 or 256 byte, for APB 8, 16 or 32 byte and for PLB bus width is 32, 64, 128 or 256 byte, for OPB 8, 16 or 32 byte and for DCR 32 byte).

Operating frequency is for all buses user defined. CoreConnect defines maximum frequency depending on the PLB width (for 32 b PLB width maximal frequency is 256 MB/s, for 64 b PLB width 800 MB/s and for 128 b PLB width, 2.9 GB/s)

V. CONCLUSION

Complex VLSI IC design is being revolutionized by the widespread adoption of the SoC paradigm. The benefits of the SoC approaches are numerous, including improvements in system performance, cost, size, power dissipation, and design turn-around time. In order to exploit these advantages to the fullest, system design methodology must optimize CA requirements. During this, we have defined the on-chip CA as a fabric that integrates the various SoC components that provides them with a mechanism for the exchange data. This paper gives an overview of three popular on chip CAs, called AMBA, CoreConnect and Wishbone. At the start a background material concerning typical topologies and communication protocols is presented. In the central part an overview of most widely used on-chip CAs is provided. Finally, a short analysis related to the possibilities of all three buses is given.

REFERENCES

1. Bennini L., DeMicheli G., *Networks on Chips: A New SoC Paradigm*, IEEE Computer, Vol. 35, No. 1, January 2002, pp. 70- 78.
2. http://www.ibm.com/chips/products/coreconnect
3. http://www.arm.com
4. http://www.sonicsinc.com
5. Lahiri K., Raghunathan A., Dey S., *Design Space Exploration for Optimizing On-Chip Communication Architectures*, IEEE Trans. on CAD of ICs and Systems, Vol. 23, No. 6, June 2004, pp. 952 – 961.
6. OMI 324 PI Bus, Rev. 0.3d, OMI standards Draft, 1994 Siemens.
7. Dally W. J., Towel B., *Principles and Practices of Interconnection Networks*, Elsevier, Amsterdam, 2004.
8. Virtual Component Interface Standard version (OCB 2.2.0) VCI Alliance, April 2001, http://www.vsi.org, available at January 2006.
9. Shandhag N.R., Reliable *and Efficient System-on-Chip Design*, IEEE Computer, Vol. 37, No. 3, pp. 42-50, March 2004.
10. Ayala J., Lopez-Vellejo M., Bertozzi D., Benini L., *State-of-the-Art SoC Communication Architectures*, pp. 20.1- 20.22, in Embedded Systems Handbook, ed by Zurawski R., CRC Press, Boca Raton, 2006.
11. ARM.AMBA Specifications v2.0, 1999.
12. ARM.AMBA Multi-Layer AHB Overview, 2001.
13. ARM.AMBA AXI Protocol Specifications, 2003.
14. http://www.opencores.org/projects.cgi/web/wishbone/wishbone