International Journal of Engineering Technology and Applied Science

(ISSN: 2395 3853), Vol. 3 Issue 7 July 2017

A Comparative Analysis of Design Automation for Application SOCs Architecture

[1]Sandeep Jagre, [2]Nishi Pandey, [3]Dr. Neelesh Gupta
[1]Mtech-Scholar, [2]Assistant Professor, [3]Head of Department
[123]Department of Electronics & Communication Engineering, TCST, Bhopal
(sandeepjagre@gmail.com, pandeynishi738@gmail.com, neelesh.gupta@trubainstitute.ac.in)

Abstract- The majority of recent embedded systems are supported MPSOCs (Multi Processors System on Chip) architectures. This can be explained by the chances that gives this sort of architectures, because it ameliorates performances by duplicating computing units on identical chip. Besides, this tendency is boosted by technological advances permitting a really massive integration scale that is critical to MPSOC fabrication. As a consequence, the challenge for MPSOCs has changed: currently, the calculation capability and therefore the range of processors on identical chip area unit additional and additional increasing and become usually on top of requests. The priority has become then to concentrate on communication and synchronization between theses processors so as to make sure higher performances of the full system. During this survey we have a tendency to propose to create a close study regarding completely different study aspects of existing MPSOCs: initial of all, we are going to touch upon the topologies and therefore the interconnections within multi-processor systems, with comparisons between P to P (Point To Point), buses and NOCs (Networks on Chip) primarily based communications. Then we are going to say GALS (Globally Asynchronous regionally Synchronous Systems). Finally, we are going to finish with introducing memory architectures of MPSOCs. This text provides an outline of 3 widespread bus organized laptop architectures (CAs), referred to as AMBA, Core Connect and wishbone. It starts with a short introduction to on-chip CA, then appearance at bus organizations, and concludes with a discussion associated with a comparative performance analysis of all 3 CAs.

KEYTERMS: MPSOC, NOC, AMBA, Core-Connect, Wishbone, On Chip CA.

I INTRODUCTION

Shrinking method technologies and increasing style sizes have crystal rectifier to extremely advanced billion-transistor integrated circuits (ICs). As a consequence, makers are integration increasing numbers of parts on a chip. A heterogeneous system-on-a-chip (SoC) would possibly embody one or additional programmable parts like general purpose processors cores, digital signal processor cores, or application specific holding (IP) cores, further as associate analog face, on-chip memory, I/O devices, and alternative application specific circuits [1].

On-chip bus organized CA is among the highest challenges in CMOS SoC technology because of quickly increasing operation frequencies and growing chip size. Usually, IP cores, as constituents of SoCs, are designed with many alternative interfaces and communication protocols. Integration such cores in a very SoC usually needs insertion of suboptimal glue logic. Standards of on-chip bus structures were developed to avoid this drawback. Presently there are some publically out there bus architectures from leading makers, like Core Connect from IBM [2], AMBA from ARM [3], Silicon Backplane from Sonics [4], and others. This paper focuses on SoC CAs providing a survey of 3 standard bus organized CAs, referred to as AMBA, Core Connect associated wishing bone from an industrial and analysis viewpoint.

II ON CHIP COMMUNICATION design

A. Background

The design of on-chip CAs addresses the subsequent 3 problems [5]:

1. Definition of CA topology - defines the natural object of the CA. various topologies exist, starting from single shared bus to a lot of complicated architectures like bus hierarchies, token ring, crossbar, or custom networks two.

2. Choice and configuration of the communication protocols - for every channel/bus within the CA, communication protocols specify the precise manner during which communication dealing occur. These protocols embody arbitration mechanisms (e.g. spherical robin access, priority-based choice [2], [3], time division multiplexed access [4], that are enforced in centralized or distributed bus arbiters.

3. Communication mapping - refers to the method of associating abstract system-level communications with physical communication methods within the CA topology [5].

B. Topologies

In regard to topology on-chip communication architectures may be classified as:

Shared bus: The system bus is that the simplest example of a shared communication design topology and is usually found in several industrial SoCs [6]. Many masters and slaves may be connected to a shared bus. A block, bus arbiter sporadically examines accumulated requests from the multiple master interfaces, and grants access to a master mistreatment arbitration mechanisms mere by the bus protocol.

Hierarchical bus: This design consists of many shared busses interconnected by bridges to make a hierarchy. SoC parts are placed at the suitable level within the hierarchy consistent with the performance level they need.

Low-performance SoC parts are placed on lower performance buses, that are bridged to the upper performance buses thus on not burden the upper performance SoC parts. Industrial samples of such architectures embody the AMBA bus [3], Core Connect [2]. Transactions across the bridge involve further overhead, and, throughout the transfer, each buses stay inaccessible to alternative SoC parts. Gradable buses supply giant turnout enhancements over the shared busses due to: (1) weakened load per bus; (2) the potential for transactions to proceed in parallel on completely different buses; and (3) multiple ward communications may be preceded across the bridge in an exceedingly pipelined manner [5].

Ring: In various applications, ring primarily based applications are wide used, like network processors, ATM switches [2], [5]. In a ring, every node element (master/slave) communicates employing a ring interface, typically enforced by a token-pass protocol.

C. On-Chip communication protocols

Communication protocols agitate differing types of resource management algorithms used for determinative access right to shared communication channels. From this time of read, within the remainder of this section, we'll provides a transient comment associated with the most options of the present communication protocols, which are:

Static-priority: employs associate arbitration technique. This protocol is employed in shared-bus communication architectures. A centralized arbiter examines accumulated requests from every master and grants access to the requesting master that's of highest priority. Transactions could also be of non-preemptive or preventive sort. AMBA and Core Connect use this protocol [3], [2].

Time Division Multiple Access (TDMA): the arbitration mechanism is predicated on a temporal order wheel with every slot statically reserved for distinctive master. Special techniques area unit accustomed alleviate the matter of wasted slots. Sonics uses this protocol [4].

Lottery: a centralized lottery manager accumulates request for possession of shared communication resources from one or a lot of masters, every of that is, statically or dynamically, assigned variety of "lottery tickets" [7].

Token passing: this protocol is employed in ring primarily based architectures. A special information word, referred to as token, circulates on the ring. Associate degree interface that receives a token is allowed to initiate a dealing. Once the dealing completes, the interface releases the token and sends it to the neighboring interface. For instance, VCI uses this protocol [8]

Code Division Multiple Access (CDMA): This protocol has been projected for sharing on-chip communicating. in an exceedingly sharing medium, it provides higher resilience to noise/interference and has a capability to support at the same time transfer of knowledge streams. However this protocol needs implementation of complicated special direct sequence unfold spectrum secret writing schemes, and energy/battery inefficient systems like pseudorandom code generators, modulation and reception circuits at the element bus interfaces, and differential sign [9].

III SOC BUSES summary

In the sequel an outline of the lot of relevant SoCCAs (AMBA, Core Connect and Wishbone) are given. As a result of area limitation the discussion are centered on describing the lot of distinctive options of every of them.

A. AMBA

AMBA (Advanced Microcontroller Bus Architecture) [3], [10], could be a bus commonplace devised by ARM with aim to support economical on-chip communications among ARM processor cores. Nowadays, AMBA is one in every of the leading on-chip busing systems utilized in high performance SoC style. AMBA (see Fig. 1) is hierarchically organized into 2 bus segments, system- and peripheral-bus, reciprocally connected via bridge that buffers information and operations between them. Common place bus protocols for connecting on-chip parts generalized for various SoC structures, freelance of the processor sort, area unit outlined by AMBA specifications. AMBA doesn't outline methodology of arbitration. Instead it permits the arbiter to be designed to best suit the applications desires. The 3 distinct buses mere among the AMBA bus are:

ASB (Advanced System Bus) - initial generation of AMBA system bus used for easy efficient styles that

support burst transfer, pipelined transfer operation, and multiple bus masters.

AHB (Advanced superior Bus) – as a later generation of AMBA bus is meant for top performance high-clock synthesizable styles. It provides high information measure communicating 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 information bus configurations, and non-tristate implementations. Constituents of AHB are: AHB-master, slave-, arbiter-, and –decoder.

APB (Advanced Peripheral Bus) – is employed to attach general purpose low-speed low-power peripheral devices. The bridge is peripheral bus master, whereas all buses devices (Timer, UART, PIA, etc) are slaves. APB is static bus that has an easy addressing with barred addresses and management signals for simple interfacing.

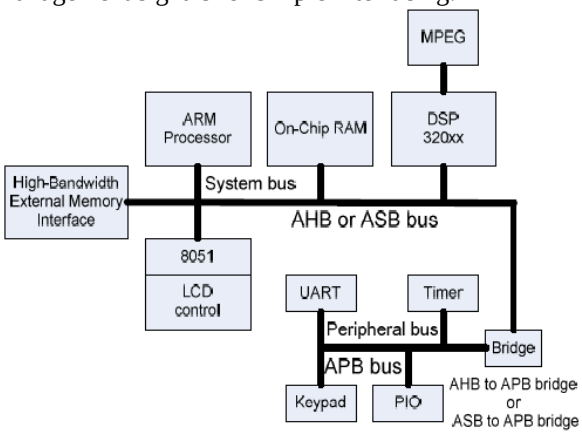

Figure 1 AMBA based system architecture (Wikipedia)

Recently, 2 new specifications for AMBA bus, Multi-Layer AHB and AMBA AXI, are outlined. [11], [12]. Multi-layer AHB provides additional versatile interconnect design (matrix that allows parallel access methods between multiple masters and slaves) with reference to AMBA AHB, and keeps the AHB protocol unchanged. AMBA AXI is predicated on the idea point-to-point association. Goog summary papers associated with AMBA specifications are references [11], [12] and [13].

B. Core-Connect

Core-Connect [2] is an IBM-developed on-chip bus. By reusing of processor, scheme and peripheral cores, equipped from totally different sources, it allows their integration into one VLSI style. Core-Connect is hierarchically organized design. It's comprised of 3 buses that give an economical interconnection of cores, library macros, and custom logic at intervals a SoC (see Fig. 2).

PLB (Processor native Bus) – is that the main system bus. It's synchronous, multi-master, central arbitrated bus that permits achieving superior and low-latency on-chip communication. Separated address, and knowledge buses support synchronal browse and write transfers. PLB macro, as glue logic, is employed to interconnect varied master and slave macros. Every PLB master is hooked up to the PLB through separate addresses, read-data and write-data buses, and different management signals. PLB slaves square measure hooked up to PLB through shared, however decoupled, address, read data, and write knowledge buses. Up to sixteen masters are often supported by the arbitration unit, whereas there are not any restrictions within the variety of slave devices [10].

OPB (On-chip Peripheral Bus) - is optimized to attach lower speed, low outturn peripherals, like serial and port, UART, etc. Crucial options of OPB are: absolutely operation, dynamic bus filler, separate address and knowledge buses, multiple OPB bus masters, single cycle transfer of knowledge between bus masters, single cycle transfer of knowledge between OPB bus master and OPB slaves, etc. OPB is enforced as multi-master, arbitrated buses. Rather than tristate drivers OPB uses distributed electronic device. 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 management Register bus) – may be a single master bus principally used as another comparatively low speed knowledge path to the system for: (a) passing standing and setting configuration info into the individual device-control registers between the Processor Core et al. SoC constituents like Auxiliary Processors, On-Chip Memory, System Cores, Peripheral Cores, etc; and (b) style for testability functions. DCR is synchronous bus supported a hoop topology enforced as distributed electronic device across the chip. It consists of 10-bit address bus and 32-bit knowledge bus. Core Connect implements arbitration supported a static priority, with programmable priority fairness.

Source of Figure 2-
www.ibm.com/chips/products/coreconnect

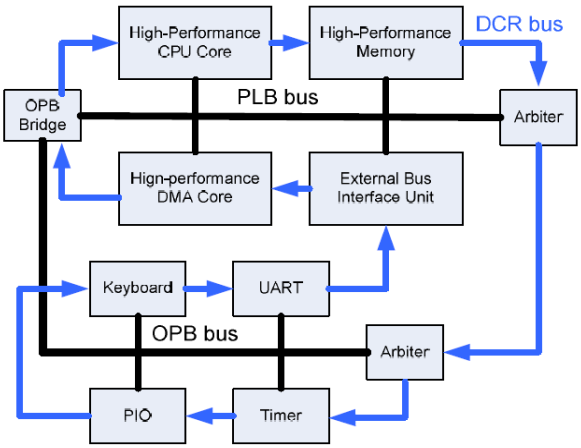

Figure 2 Core Connect bus based system [2]

C. Wishbone

Wishbone [13] bus design was developed by Silicore Corporation. In August 2002, Open Cores (organization that promotes open science cores development) place it into the general public domain. This implies that wishbone isn't proprietary and might be freely traced and distributed.

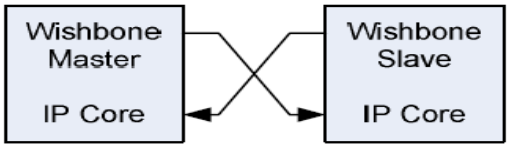

Figure 3 Point to Point Interconnection [13]

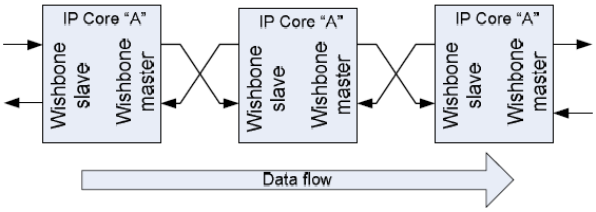

Figure 4 Data Flow Interconnection [13]

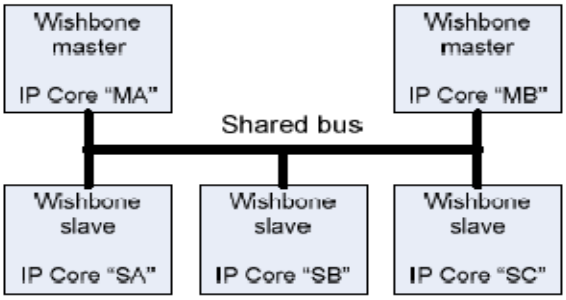

Figure 5 Shared Bus [13]

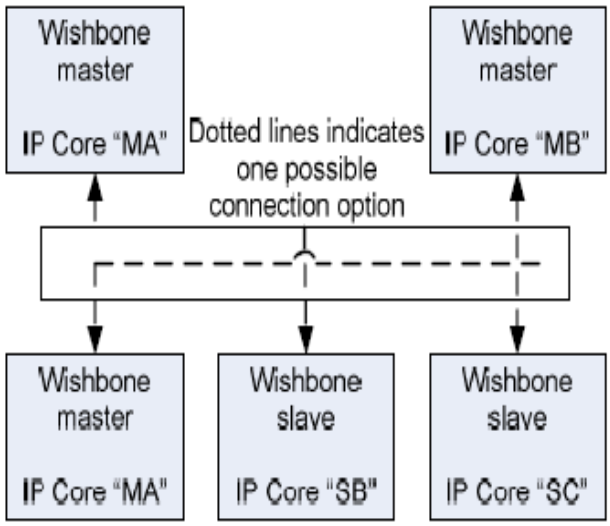

Figure 6 Cross Bar Switch Interconnection [13]

Source of Figure 3, 4, 5 and 6- www.opencores.org/projects.cgi/web/wishbone/wishbone.

The wishing bone defines 2 forms of interfaces, known as master and slave. Master interfaces are IPs that are capable of initiating bus cycles, whereas slave interfaces are capable of accepted bus cycles [10]. The hardware implementations support numerous forms of interconnection topologies such as: point-to-point affiliation (Figure 3) - used for direct affiliation of 2 participants that transfer knowledge per some shake protocol

a) Knowledge flow interconnection (Figure 4) - utilized in linear pulsation 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) - sometimes utilized in MPSoCs once quite one masters will at the same time access many completely different slaves. The master requests a channel on the switch, once this can be established, knowledge is transferred in an exceedingly point-to-point manner.
The wishing bone supports differing types of bus transactions, like read/write, implementing blocking/unblocking access.
A Read-Modify-Write transfer is additionally supported. Wishing bone doesn't outline hierarchical buses. In applications wherever 2 buses ought to exist, one slow and one quick, 2 separated wishing bone interfaces may well be created. Designer may opt for arbitration mechanism and implements it to best work the applying desires.

Table No. 1 An analysis of Different Topologies

Ref. No.	Topologies	Evaluation Type	Objective function	Constraints	Other Features
14	Heter. Arch	Analytical	Length	Area	-
15	Heter Arch	Analytical	Length	Area	Joint Scheduling
16	Cross Bar	N/A	Power, Area	Length	Dynamic Configuration
17	Cascaded Cross	Hybrid	Energy, Area	Length	-
18	Cascaded Cross	Analytical	Power, Area	Length, Bandwidth	Floor-Planning

Some common options for presented SoC buses, like topology, arbitration, transfers, and bus dimension. All given buses are synchronous. AMBA and Core-Connect are stratified buses. wishing bone doesn't defines stratified bus interconnection, however permits numerous different attainable interconnections, like point-to-point, ring, uni-level shared bus, crossbar switch interconnection, etc. Arbitration technique for AMBA and wishing bone is application specific, which suggests that arbiter may be designed concerning to the appliance necessities. Core-Connect defines static priority. Given SoC buses support numerous transfer varieties. All support acknowledgment, split transfer and burst transfer, whereas pipelined transfer support AMBA and Core-Connect, but not Wishbone. Address and information bus dimension are configurable. For AMBA and Core-Connect information bus dimension depends on sort of the bus (for AHB and ASB bus dimension is thirty two, 64, 128 or 256 computer memory unit, for APB 8, sixteen or thirty two computer memory unit and for PLB bus dimension is thirty two, 64, 128 or 256 computer memory unit, for OPB 8, 16 or 32 computer memory unit and for DCR 32 byte). In operation frequency is for all buses user outlined. Core-Connect defines most frequency looking on the PLB dimension (for thirty two b PLB dimension top frequency is 256 MB/s, for sixty four b PLB dimension 800 MB/s and for 128 b PLB dimension, 2.9 GB/s).

IV PROBLEM STATEMENT

In this brief, we have done the survey of different on-chip protocols along with their features and architectures. A descriptive comparison between various on-chip protocols is needed. So we have to find out the efficient protocol as it can efficiently transfers block of data thereby reducing the hardware resources and minimal power consumption. This can be verified by implementing the our projected protocol at RTL in HDL and comparing the same with other protocols by considering various parameters such as transfer time consumption, wire efficiency, valid data bandwidth, dynamic energy efficiency and power consumption.

V GENERAL PROPOSED METHODOLGY

The main difficulty facet by the design or layout design engineer to design the successful SOC with well-structured and synthesizable RTL code are power consumption and clock skew. The aim of this paper is to formulate the problem in design of successful SOC and propose a general mechanism steps to solve this problem of power consumption. Mainly in digital sequence circuit clock is major concern. To minimize the clock skew problem we use flip-flop master slave approach and reduce the total power consumption in SOC. General Steps or proposed methodology steps are given below:

1. Identify the clock signal.
2. Transition of clock signal.
3. Examined each and every clock precisely.
4. Find interrelation and unused clock
5. Design state diagram.
6. Behavioral description.
7. RTL synthesis.
8. Simulation
9. Power analysis.

These proposed steps will be implemented on Xilinx 14.1 and Model-Sim using Verilog language.

IV CONCLUSION

Complex VLSI IC style is being revolutionized by the widespread adoption of the SoC paradigm. The advantages of the SoC approaches are various, together with enhancements in system performance, cost, size, power dissipation, and style turn-around time. So as to use these blessings to the fullest, system style methodology should optimize CA needs. Throughout this, we've outlined the on-chip CA as a cloth that integrates the assorted SoC parts that gives them with a mechanism for the exchange information. This paper offers an outline of 3 standard on chip CAs, referred to as AMBA, Core-Connect and Wishbone. At the beginning a background material regarding typical topologies and communication protocols is bestowed. Within the central half an outline of most generally used on-chip CAs is provided. Finally, a brief analysis associated with the chances of all 3 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 http://www.sonicsinc.com
[4] Lahiri K., Raghunathan A., Dey S., Design Space Exploration for Optimizing On-Chip Communication Architectures, IEEETrans. on CAD of ICs and Systems, Vol. 23, No. 6, June 2004,pp. 952 – 961.
[5] OMI 324 PI Bus, Rev. 0.3d, OMI standards Draft, 1994 Siemens.
[6] Dally W. J., Towel B., Principles and Practices of Interconnection Networks, Elsevier, Amsterdam, 2004.
[7] Virtual Component Interface Standard version (OCB2.2.0) VCI Alliance, April 2001, available at January 2006.
[8] Shandhag N.R., Reliable and Efficient System-on-Chip Design, IEEE Computer, Vol. 37, No. 3, pp. 42-50, March 2004.
[9] Ayala J., Lopez-Vellejo M., Bertozzi D., Benini L., State-of-the- Art SoC Communication Architectures, pp. 20.1- 20.22, inEmbedded Systems Handbook, ed by Zurawski R., CRC Press,Boca Raton, 2006.
[10] ARM.AMBA Specifications v2.0, 1999.
[11] ARM.AMBA Multi-Layer AHB Overview, 2001.
[12] ARM.AMBA AXI Protocol Specifications, 2003.
[13] http://www.opencores.org/projects.cgi/web/wishbone/wishbone.
[14] Cilardo, E. Fusella, L. Gallo, A. Mazzeo, Automated synthesis off Fpga-based heterogeneous interconnect topologies, in: 2013 23rd International Conference on Field Programmable Logic and Applications (FPL), IEEE, 2013, pp.1–8.
[15] Cilardo, E. Fusella, L. Gallo, A. Mazzeo, Exploiting concurrency for the automated synthesis of mpsoc interconnects, ACM Trans. Embed. Comput. Syst. (TECS) 14(3) (2015)57.
[16] J. Y. Hur, T. Stefanov, S. Wong, K. Goossens, Customisation of on-chip network interconnects and experiments in field-programmable gate arrays, IET Comput. Digit. Tech. 6(1) (2012) 59–68.
[17] Y. -P. Joo, S. Kim, S. Ha, Efficient hierarchical bus-matrix architecture exploration of process or pool-based mpsoc, Des. Autom. Embed. Syst.16 (4) (2012) 293–317.
[18] M. Jun, D. Woo, E.- Y. Chung, Partial connection aware topology synthesis for on-chip cascaded crossbar network, IEEE Trans. Comput. 61 (1) (2012) 73–86.