CDMA CODED WRAPPER-BASED SYSTEM BUS

Tatjana R. Nikolić and Mile K. Stojčev
Faculty of Electronic Engineering, A. Medvedeva 14, 18000 Niš, Serbia, E-mail: {tatjanas,stojcev} @elfak.ni.ac.yu

Abstract: *The recent development of Field Programmable Gate Array (FPGA) System-on-Chip (SoC) architectures, with coarse-grain processors, embedded memories and Intellectual Property (IP) cores, offers high performance for computing power as well as opportunities for rapid system prototyping. These platforms require high-performance on- and off-chip communication architectures for efficient and reliable inter-processor data transfer. By increasing the number of IP cores that are embedded in a SoC design, as well as the number of VLSI circuits that are installed in circuit boards, the problem of interconnection becomes more challenge. In this paper, we propose an efficient technique for realization of on- and off-chip system bus based on wrapper technology and CDMA techniques, in order to achieve efficient data transfer among IP cores in SoC and among chips on circuit boards. The main benefits of using this technique related to decreasing the number of wires on system bus in average for 50 %, while the main disadvantage deals with increasing the latency of Read and Write processor cycles.*

Keywords: *System bus, CDMA technique, SoC, Wrapper logic.*

1. INTRODUCTION

On-chip communications present permanent design challenge for high speed data transfer realization among building blocks in SoC design. Bus architectures (on-chip/off-chip buses), Network-on-Chip (NoCs) and point-to-point connections nowadays are used in order to successful elimination this bottleneck at complex VLSI IC design. On-chip buses can be classified into standard buses and wrapper-based buses. Standard buses are specified and realized for using protocols over wiring connections between IP cores within SoC [1]. Typical on-chip standard buses, which are used in SoC designs, are AMBA, CoreConnect, etc. Standard off-chip buses are VME, Multibus, etc. Wrapper-based approach uses the IP core interface protocol. It is independent of a physical bus protocol, and uses hardware wrappers to handle with core-to-core communications. In contemporary embedded systems, a CDMA technique is proposed as a new way for IP core interconnections. This technology relies on a principle of codeword orthogonality, such that when multiple codewords are summed, they do not interfere completely with each other at every point in time and can be separated without loss of information [2].

In this paper, we consider a realization of a system bus based on wrapper logic and CDMA technique used for data transfer between the CPU, a memory, and input/output subsystem. A corresponding wrapper structure is accompanied to each IP core in a SoC solution, or to memory/input-output module in printed circuit board solutions. By attaching a wrapper hardware the width of data and address buses is decreased, while latency of Read and Write processor cycles is increased. Implementation of a wrapper based bus is illustrated on uni-processor 32-bit system.

2. TAXONOMY OF ON-CHIP COMMUNICATION ARCHITECTURES

Taxonomy of on-chip and off-chip communication architectures is pictured on Figure 1. As can be seen from figure, communication architectures can be divided into three main classes. In this case, the term architecture relates to the structure of interconnection between processing elements, protocols and interface design [1].

In point-to-point interconnect architecture, pairs of processing units communicate directly over dedicated physically wired connections. The interconnections can be performed as custom, referred as ad hoc interconnections, or as uniform. For bus architectures, long wires are grouped together to form a single physical communication channel, which is shared among different logical channels. An arbitration mechanism is used to control sharing of the bus. Typical bus architectures are AMBA, CoreConnect etc. Network-on-Chip (NoC) is an architecture of type data communication networks, such as LANs, with inter-processor communication supported by a packet switched network.

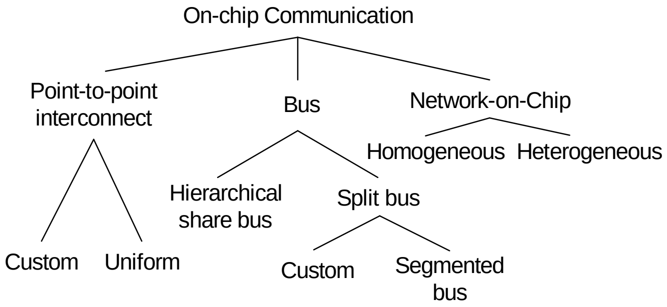

Fig.1: *Taxonomy of on-off-chip communication architectures*

Bus architectures that use the concept of a hierarhical shared bus are in focus of interest for us. Accordingly, in the sequal we will consider the implementation of uni-processor CDMA wrapper based bus.

3. UNI-PROCESSOR SYSTEM BASED ON STANDARD BUS ARCHITECTURE

A typical 32-bit uni-processor system is sketched in Figure 2. The system consists of several IP core/modules. Modules can be of master or slave type. The structure given in Figure 2 is suitable for SoC or PCB implementations. In our case, a system composed of one master and several slaves will be analyzed. The CPU is a master core, while memory blocks, MEMi, i=1, ..., *m*, and peripheral units, PERj, j=1, ..., *k*, are slave cores. All cores are connect via a system bus. The system bus is composed of unidirectional 32-bit address bus, ADR[31:0], bidirectional 32-bit data bus, DATA[31:0], and bidirectional control bus. Constituents of a control bus are: STATUS[2:0] lines - point to a current processor cycle; M/IO line - defines selection of a memory or input-output module; RD line - active when *Read* cycle is in progess; WR line - asserted when *Write* cycle is in progess; INTR - interrupt request initiated by peripheral modules, INTA - interrupt acknowledge; RDY - assertation for data transfer readiness initiated by a slave module; CS_{Mi} (CS_{Pj}) - chip select for memory/peripheral module. The building blocks Memory_Address_Decoding_Logic, MADL, Input/Output_ Address_Decoding_Logic, IOADL, and Wait_State_Logic, WSL, are realised as interface units located between the CPU and memory/peripheral modules. In our case, the MADL and IOADL are implemented as combinational multilevel chip select decoders, while the WSL is implemented as a shift register with possibility to insert variable number of wait states.

4. MOTIVATIONS FOR USING CDMA TECHNIQUE

As a consequence of shrinking transistor dimensions, a complexity of VLSI ICs, from aspect of number of transistors, increases at faster rate than designer's possibilities to use these benefits are. Such a trend in development and research has created in a well-known gap in VLSI production. This gap appears due to limited intelectual designer's capabilities, from one side, and technology possibility to reuse IP cores in a SoC design, from other side. In average, daily, an engineer can design a logic circuit which hardware complexity doesn't exceed forty gates, while the reuse concept enables to build IC blocks with hardware complexity up to 1000000 gates. Having this in mind, but with aim to decrease the productivity gap, nowadays, many designers use extensively the reuse concept, based on pre-designed and pre-verified IP cores. Typically, IP cores are realized as microprocessors, microcontrollers, DSP processors, dedicated functional units, bus interfaces, and numerous others peripheral components. Until recently, the design-space exploration for SoCs has been mainly focused on the computational aspects of the problem, i.e. increasing microprocessor and peripheral chip performance. However, as the number of IP blocks on a single chip and their performance continue to increase, a shift from computation-based to communication-based designs becomes mandatory. As a result, the communication architecture plays a major role in the area, performance, and energy consumption of the overall systems. In order to increase system performance it is necessary to design high speed and high bandwidth data transfer buses. In contrary, further performance increase of computer constituents will be without effect on overall system performance. This is a reason way during the last several years too much research efforts in high speed bus development is devoted. But, during realization of high speed buses we meet with numerous problems. These problems are typical for realization of a interconnect. To solve this problem various techniques are used. In most cases, these techniques include implementation of additional hardware. An alterative solution to increase bus throughput consists of increasing bus data transfer lines. As a consequence, by using this approach, the number of lines, system complexity, occupied PCB area, and PCB tracing increase. In all cases, the buses with corresponding interfaces become very complex system. As a number of bus lines becomes higher the cost of these systems increases. Bearing this in mind, a bandwidth improvement achieved by increasing the number of bus lines, for most design solutions, is not a rational economical solution.

Code division multiple access (CDMA) has been proposed as an alternative way for interconnect of IP cores in a SoC design, or as a solution for interconnecting modules within a system realized in several PCBs. Compared to a conventional TDMA-based bus, a CDMA-based bus has better features concerning channel's isolation and channel's continuity in time domain since channels are divided by the spreading codes [3]. CDMA technology relies on the principle of codeword orthogonality, such that it enables efficient separation of information.

5. SYSTEM BASED ON WRAPPERS AND CDMA TECHNIQUE

A structure of a system is shown in Figure 3. From aspect of functionality, it is identical to the system presented in Figure 2. A main difference relates to the implementation of data transfer technique between master and slave modules. In both cases, for data transfer, a system bus is used. In Figure 2 it corresponds to classical solution, while in Figure 3 a system bus based on CDMA technique is implemented. In order to implement a CDMA technique a corresponding bus wrapper logic is appended to each module.

Wrapper based bus as an innovation technology enables reusing of IP cores in SoC designs, in an efficient way [4, 5]. From logical point of view, by using this approach, the operation of a communication logic and IP core logic can be analyzed separately. This allows us to bridge a connectivity problem, which relates to physical bus protocols. Naimely, a wrapper based approach provides us to use: a) IP core interface protocol independently of a physical bus protocol; and b) hardware wrappers to handle the core-to-core communication. Consequently, IP cores complying with the interface protocol can be easilly integrated into SoC designs that use different physical buses (AMBA, CoreConnect, etc.) as backbones. However, by attaching a simple wrapper hardware we decrease, from one side, the interconnect complexity, but, from the other side, we increase the system latency. In general, an optimal solution represents a compromise between two contradictory reqirements. In order to solve this problem corectly a spetial design attention is needed.

In a concrete proposal sketcked in Figure 3, to the CPU a wrapper BW_CPU called master is appended. A slave wrapper denoted as BW_MEM_i is appended to memory block MEMi, and a slave wrapper BW_PER_j to a peripheral block PERj. In order to achieve a low hardware system complexity, a CDMA data transfer technique is implemented on address and data buses, only. The control bus is identical for both solutions. Let note, that the wait state logic and the logic used for selection of memory blocks or peripheral units (WSL, MADL, and IOADL - see Figure 2) is appended to the wrapper

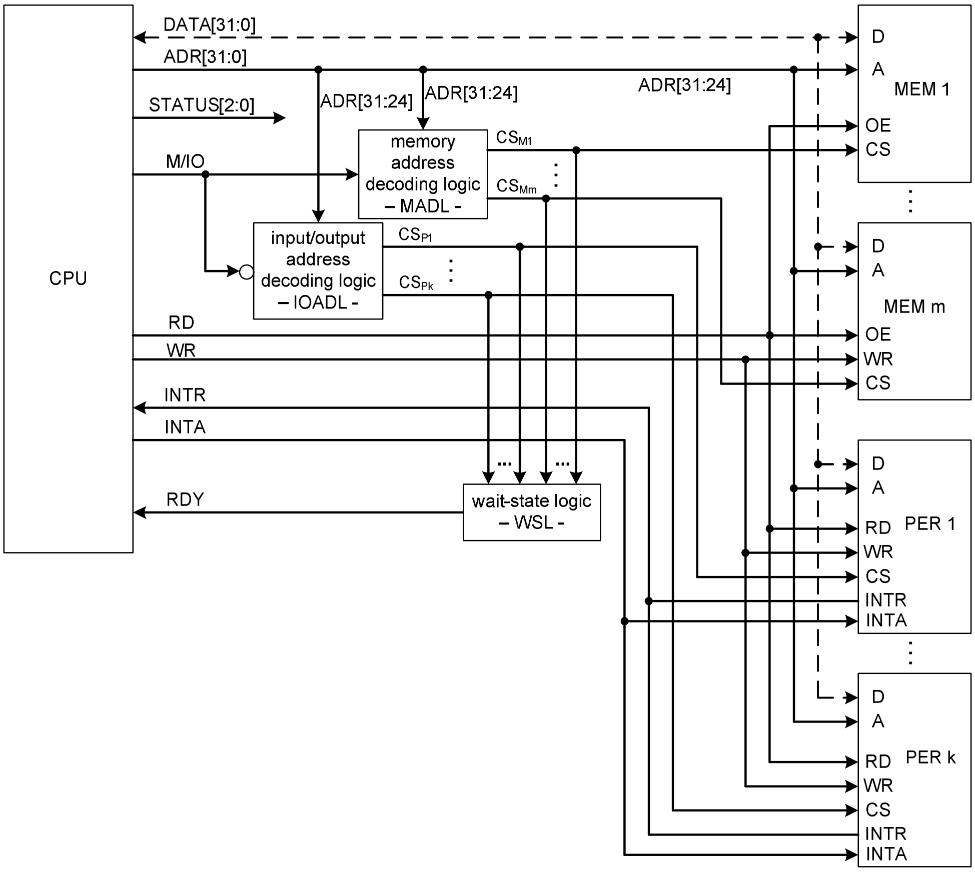

Fig.2: *Uni-processor 32-bit system based on standard bus architecture*

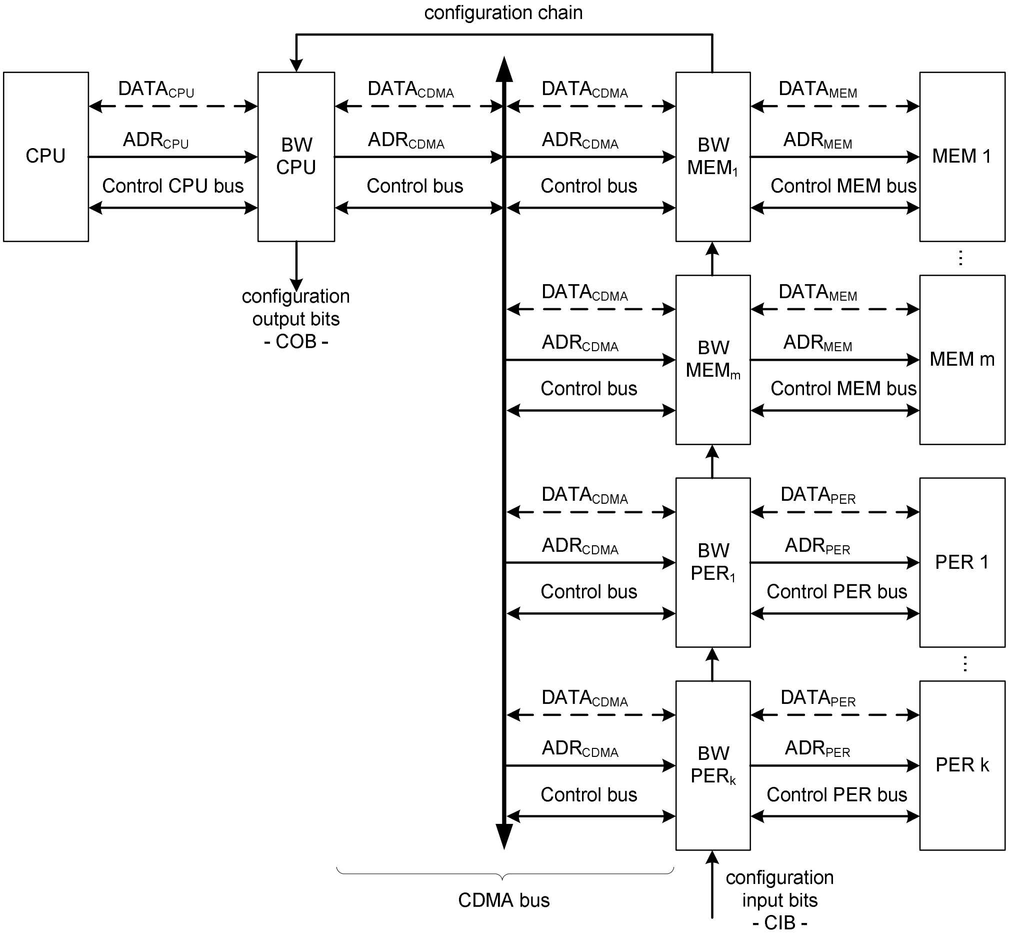

Fig.3: *Uni-processor 32-bit system organized around system bus which uses wrappers as interface logic and CDMA transmission technique for data transfer*

logic (see Figure 3). After a system reset the CPU initializes all wrappers connected in a chain. Initialization is performed by inputing a corresponding configuration file through CIB pin.

Master and slave wrappers are of similar hardware structure. From operating point of view the main difference between them is the following:

a) At its inputs the master wrapper accepts signals compliant with VCI 2.0 standard defined by VSIA [6], and at its outputs generates signals that are used for CDMA bus transfer.

b) At its inputs a slave wrapper accepts signals from CDMA bus, and at its outputs generates signals compliant with VCI 2.0 standard.

Since the structures of a master and slave wrapper are similar, in the sequel, we will explain the structure of a master wrapper, only (see Figure 4).

Constituents of a BW_CPU are: the functional units BWCU (*Bus Wrapper Control Unit*), AE (*Address Encoder*), DED (*Data Encoder/Decoder*) and CD (*Command Decoder*), and system control units CG (*Clock Generator*) i CR (*Configuration Register*). The CR unit accepts and stores a configuration file [7]. CG acts as a PLL system. It generates clock signals for all functional units. A BWCU is realized as a finite state machine (FSM). It generates control signals for driving BW_CPU functional units. The AE converts a binary coded address into a CDMA address. A DED is bidirectional converter. It operates in half-duplex mode using time division multiplexing. In a direction CPU → CDMA bus a DED converts CPU output data into CDMA data. In opposite direction, it converts CDMA data into binary coded CPU input data. A CD unit passes through the CPU control signal lines to CDMA control bus.

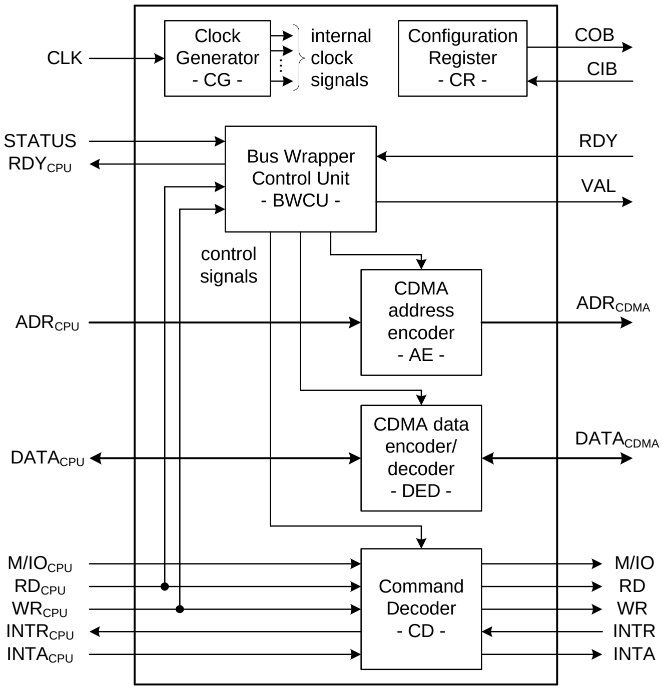

Fig.4: *Wrapper structure*

The operation of a CDMA coded wrapper-based bus we will explain on execution of processor Read (see Figure 5) and Write cycles (see Figure 6). The Read cycle begins at instant t_0. After a time period t_a the CPU sets its address and status output lines at valid states and the MW accepts them. As a response, the MW activates a signal RDY. It signals CPU to insert wait states. In addition, the MW converts address from binary to CDMA code and sends it via a CDMA bus to slave wrappers. A total time period, t_{12}, used for address transfer in a direction CPU → MEM/PER module (see Figure 5), is defined as $t_{12} = t_{MW} + t_{cc} + t_p + t_{SW}$; where: $t_{MW}(t_{SW})$ is a needed time to latch address in a master (slave) wrapper; t_{cc} is a transfer time of CDMA coded information; and t_p corresponds to signal propagation time over CDMA bus. t_{cc} is proportional to number of bits in a spreading code and is equal to $t_{cc} = s * t_{br}$; where s is number of bits in a spreading code, and t_{br} is time for processing and transfer of single bit through CDMA bus. Since, $t_{cc} >> t_{MW} + t_{SW} + t_p$ than $t_{12} \approx t_{cc}$ is valid. Let note that in Figure 5 time intervals t_{MW} i t_{SW} are not presented. After t_2, an access to memory or peripheral module is performed, for time period t_{acc}. At instant t_3, the addressed module has ready data and send them to SW. A module SW codes data and delivers them to MW. MW block decodes data and sends them to CPU-u. After that, SW deactivates a signal RDY which passes through MW and drives CPU. In this moment, the wait state is terminated. Next, the CPU accepts data present on data lines and after t_b, in t_5 it terminates a Read cycle.

In Figure 6 a Write cycle is sketched. The scenario of this cycle is simpler in respect to the Read cycle. The main difference is the following: During the first part of a Write cycle, at instant t_1, the CPU generates an address, while at t_1' it generates a binary coded valid data. Both address and data bus signals drive the BW_CPU which converts them into CDMA coded signals. During this $\tau = t_1' - t_1 \approx 0$ is valid. In order to make visual presentations (Figures 5 and 6) more ilustrative, the address transfer is pictured by a full line, while data transfer is pictured by a broken line.

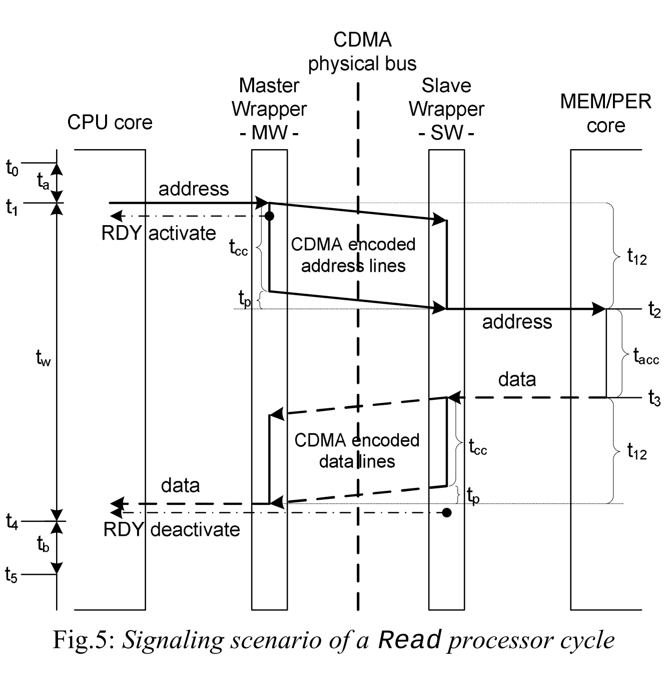

Fig.5: *Signaling scenario of a* Read *processor cycle*

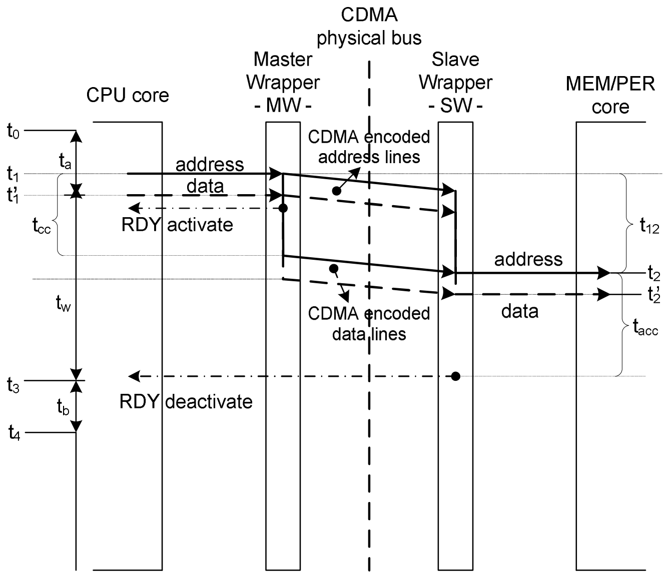

Fig.6: *Signaling scenario of a* `Write` *processor cycle*

6. RESULTS

The proposed method can be implemented to any address and data bus as constituent of a system bus. In a concrete case, we use S orthogonal codes for CDMA coding. As a result, if unencoded buses are of n bits width then the CDMA coded equivalent buses will be reduced to $p = \frac{n}{S}\lceil \log_2 S + 1 \rceil$ bits. For 8-bit spreading code a bus reduction is 50%. Results concerning an average bus reduction, denoted as R_{BL}, are given in Table I. As a consequence of implementation a CDMA data transfer the system latency increases. It is expressed as a number of clock pulse periods in term of a spreading code width. For Read cycle a latency is denoted as T_{RCP}, and for Write cycle as T_{WCP}. The results which relate to T_{RCP}, and T_{WCP} are given in Table 1.

Table 1. *Performance concerning bit reduction and latency*

n	8	16	32	64	128	256	R_{BL} (%)	T_{RCP} (clk)	T_{WCP} (clk)
S									
4	6	12	24	48	96	192	25	8	4
8	4	8	16	32	64	128	50	16	8
16	-	5	10	20	40	80	68.75	32	16
32	-	-	6	12	24	48	81.25	64	32

7. CONCLUSION

An efficient technique for realization of on- and off-chip system bus based on CDMA techniques and wrapper technology is proposed in this paper. It is intended to achieve an efficient communication between IP cores in on-chip bus, and CPU and memory/peripheral modules in off-chip system bus. The benefits of the proposal relate to decreasing data and address bus width. The drawback deals with increasing system latency. The proposed solution is not implemented on a control bus in order to be compliant with already well know wrapper based solutions for standard buses such as AMBA, CoreConnect etc. The wrapper logic, realized on FPGA platform, provides a possibility to implement a reconfigurable solution. The reconfiguration is possible to achieve by modifying the configuration file during FPGA/system initialization phase.

8. REFERENCES

[1] Terrence S. T. Mak, N. Pete Sedcole, Peter Y. K. Cheung, Wayne Luk , “On-FPGA Communication Architectures and Design Factors”, *in International Conference on Field Programmable Logic and Applications*, 2006, pp. 1-8

[2] X. Wang, T. Ahonen, and J. Nurmi, “Applying CDMA Technique to Network-on-Chip”, *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, Vol. 15, No. 10, October 2007, pp. 1091-1100

[3] B. C. Lai, P. Schaumont, and I. Verbauwhede, “CT-bus: A Heterogeneous CDMA/TDMA Bus for Future SOC” in *Proc. 38th Annu. AsilomarConf. Signals, Systems, and Computers*., Vol. 2, Issue 7-10, Nov. 2004, pp. 1868 - 1872

[4] K. Anjo, A. Okamura, M. Motomura, “Wrapper-based bus implementation techniques for performance improvement and cost reduction”, *IEEE Journal of Solid-State Circuits,* Vol. 39, Issue 5, May 2004, pp. 804 - 817

[5] R. Lysecky, F. Vahid., “Pre-fetching for Improved Bus Wrapper Performance in Cores” *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, Vol. 7, No. 1, January 2002, pp. 1-33

[6] On-chip Bus Development Working Group, “Virtual Component Interface Standard Version 2”, April 2001.

[7] A. Athavale, C. Christensen, *High-Speed Serial I/O Made Simple*, Xilinx Connectivity Solutions, San Jose, April 2005