CDMA versus TDMA Transfer over Shared Bus

Tatjana R. Nikolić[1] and Mile K. Stojčev[1]

Abstract – **CDMA interconnect is a new interconnect mechanism for SoC design. This paper presents how a multiprocessor system can benefit from the use of concurrent data transfers: CDMA interconnect has been adopted to implement the shared bus of a multiprocessor system. Unlike the conventional TDMA bus-based multiprocessor system that shows degradation in performance as the number of processing cores increases, the proposed CDMA bus-based multiprocessor system shows higher performance up to 188 percent for four processors. The obtained results show that increased data transfer latencies involved by CDMA data transfer are compensated by simultaneous master-slave transfers.**

Keywords – **TDMA, CDMA, Bus, Wrapper, Latency.**

I. Introduction

Current VLSI design trends are shifting toward the System-on-Chip (SoC) incorporating several homogeneous or heterogeneous processing elements. As systems become more complex, the interconnection between subcomponents is becoming increasingly difficult and costly. Therefore, communication architecture has become the bottleneck for improving system-level performance, and it requires special design attention. To design an efficient interconnection network, the system's bandwidth and latency sensitivities should be taken into account together [1].

In general, on-chip communication architectures can be categorized into three main classes: point-to-point interconnects, bus, and network-on-chip [2]. However, there is no standard solution how to establish fast, flexible, efficient, and easy-to-design communication network to connect a large number of IP cores that have heterogeneous requirements.

Until now, most of interconnect networks in the SoC rely on parallel system bus, which apply time division multiple access, TDMA. The popularity and wide acceptance of the shared bus architecture is perhaps due to the fact that it is easy to adopt, well known among the computer industry, and also relatively inexpensive to implement [3]. In those systems, the bus masters perform read and write operations with slave memory or I/O modules [4]. However, bus-based architectures cannot scale up with an increasing number of components.

In this paper, the Code Division Multiple Access, CDMA, is used to implement the memory/peripheral shared bus of multicore systems consisting of multiple processing cores. The motivation of this paper is to explore how the standard memory/peripheral bus could be modified for higher system performance without increasing cost and complexity.

In the proposal, only bus lines that carry address and data signals are CDMA coded. We implemented a pair of master-slave wrapper as a component of CDMA, described in VHDL, using a Xilinx Spartan and Virtex. We also present results which show that benefits of involving CDMA coding relates both to decreasing a number of bus lines, and accomplishing simultaneous multiple master-slave connections. Convenient range indices R_W and R_R to determine data transfer rate for `Write` and `Read` operations in multiprocessor bus systems that use TDMA and CDMA data transfer techniques.

II. IP Cores Interconnecting

Among the various conventional interconnection networks, multipoint shared bus is the most simple and cost-effective solution for small-scale shared-memory multiprocessor systems. A number of different standard busses are already being used in custom SoC designs [5]. Different IP blocks can be used to build a SoC as long as they are designed according to the standard bus specifications. Bus interfaces usually have groups of wires for command, address, and read or write data.

As we have already mentioned, a bus is a resource shared by multiple cores. Therefore, before using bus, cores must go through an arbitration phase. If there is a single arbitration for a request-response pair, the bus is called non-split. Typical shared bus memory/peripheral access scenario in a multiprocessors system is given in Fig. 1. In this case, the bus remains allocated to the master of the transaction until the response is delivered, even when this takes a long time. Alternatively, a split bus is released after the request to allow transactions from other masters to be initiated [1], [5].

The communication between the masters and slaves of bus-based systems is traditionally based on a TDMA. This interconnect is simple but not efficient in dealing with multiple data transactions simultaneously. The TDMA-based buses may have long request latency for concurrent requests due to the bus contention and arbitration delays [1].

The bus allocation in single master-slave connections is determined by an arbitration protocol. However, by using CDMA data transfer technique, which is based on a concept that each master-slave set can use its unique code subset, it is possible, over shared bus, to realize multiple master-slave data transfers, simultaneously. In order to implement this approach, we need to: i) modify the arbiter's hardware (by implementation of logic for solving the contention problem); and ii) involve a wrapper that is used as interface logic between the shared bus and IP connecting to it.

The traditional bandwidth-focused approach is not enough to improve overall system performance. To compare the `Read` and `Write` operation latencies, the bidirectional shared bus implementations based on a conventional TDMA and a proposed CDMA are discussed. Both busses have the same number of data and address lines with the same aggregate bandwidth per wire line. In a conventional TDMA bus, if two read requests arrive at arbiter at the same time, the second

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

request must stall until the first request finishes using the shared interconnect [6]. As the channel becomes narrower, the address and data latency of TDMA approach increases more rapidly. The goal of this paper is to illustrate that a TDMA based bus has a longer latency than the CDMA based bus.

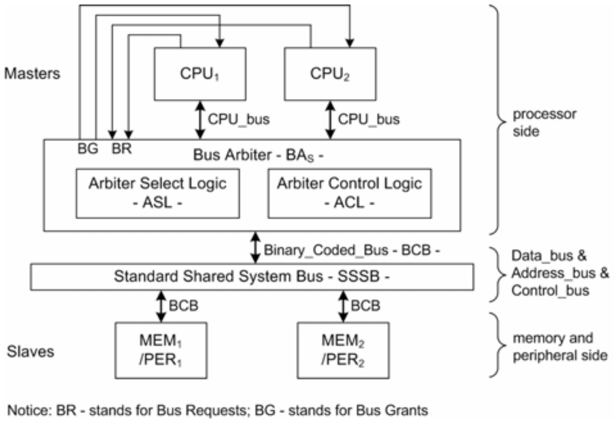
Masters
CPU$_1$
CPU$_2$
CPU_bus
CPU_bus
BG BR
Bus Arbiter - BA$_S$ -
Arbiter Select Logic - ASL -
Arbiter Control Logic - ACL -
processor side
Binary_Coded_Bus - BCB -
Standard Shared System Bus - SSSB -
Data_bus & Address_bus & Control_bus
BCB
BCB
Slaves
MEM$_1$ /PER$_1$
MEM$_2$ /PER$_2$
memory and peripheral side
Notice: BR - stands for Bus Requests; BG - stands for Bus Grants

Fig. 1. A standard bus topology

III. Multiprocessor System Based on CDMA Shared System Bus

In order to eliminate variance of data transfer latency and complexity, an bus based interconnect which applies a CDMA technique is introduced in this paper. As one of the spread-spectrum techniques, the CDMA technique has been widely used in communication systems because it has great bandwidth efficiency and multiple access capability. The CDMA technique applies a set of orthogonal codes to encode the data from different users before transmission in a shared communication media. Therefore, it permits multiple users to use the communication media concurrently by separating data from different users in the code domain [7].

When a CDMA technique is implemented on standard multiprocessor system presented in Fig. 1, we obtain a scheme given in Fig. 2. In order to make the discussion clear, we will assume that the system consists of two local computers and two shared memory or peripheral modules. By comparing the structures sketched in Fig. 1 and 2, we catch sight of the following three basic differences:

1. The Standard Shared System Bus, SSSB (see Fig. 1) is substituted with a CDMA Shared System Bus, CSSB (Fig. 2).

2. The Bus_Arbiter, BA, given in Fig. 2 is realized using the following two building blocks: Arbiter_Switching_Logic, ASL, and Arbiter_Control_Logic, ACL. According to the implemented algorithm for bus priority assignment the ACL's output Switch, defines which CPU bus will drive a corresponding BCB_1 or BCB_2, respectively.

3. Two types of bus wrapper are involved. The master bus wrapper, BW_CPU, converts data and address signals present at BCB_1 and BCB_2 into CDMA coded bus signals and conversely. The slave wrapper, BW_MEM/PER, connects memory or peripheral IP blocks to the CSSB. In all standard solutions based on AMBA bus [8], CoreConnect, STBus, etc., the data transfer protocol over system bus is mainly defined by the timing (signaling) of the Control_bus. In order to improve IP core reusability and be protocol compliant with any standard on-chip bus we involve a CDMA coding for data and address signal lines, while signal lines that belong to the Control_bus remain unchanged.

In general, the hardware structures and principles of operation of the BW_CPU and BW_MEM/PER are similar. The following six modules, presented in [9], are parts of the bus wrapper logic: 1) Bus Wrapper Control Unit, BWCU; 2) Control Protocol Transfer Block, CPTB; 3) CDMA Data Encoder/Decoder, DED; 4) CDMA Address Encoder, AE; 5) Configuration Register, CR; and 6) Clock Generator, CG. The specifity of the wrapper structure, sketched in Fig. 2, relates to the implementation of CDMA encoder and decoder blocks.

IV. CDMA Coded Bus Transfer Operation

The operation of a CDMA coded wrapper-based bus we will explain on execution of CPU Read and Write cycles. Figure 3 shows a case when the CPU IP core initiates a Read cycle. The BA logic acts as a transparent block, i.e. it maps standard CPU bus signals into binary coded bus signals.

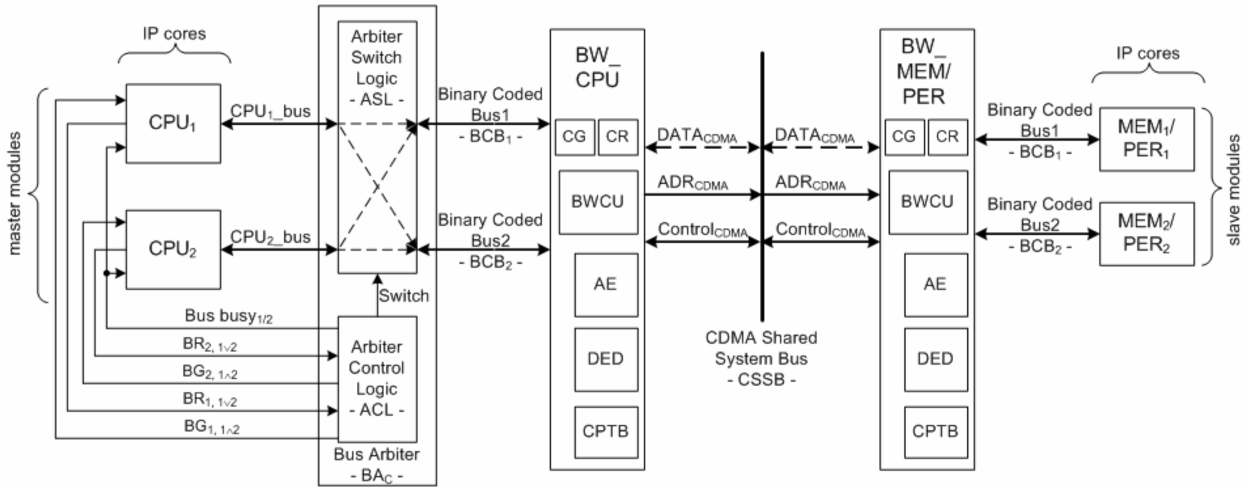

Fig. 2. Multiprocessor system based on CDMA transmission technique

The Read cycle begins at instant t_0. At t_1 the CPU IP core sets its address and status lines at valid states and, after passing through BA logic, the BW_CPU accepts them and asserts a signal RDY_activate. It signals to the CPU IP core to insert wait states. In addition, the BW_CPU converts address from binary to CDMA form and sends the address via CSSB. The BW_MEM/PER wrapper decodes CDMA coded address and at instant t_2 drives the MEM/PER IP core with a binary coded address via BCB_ADR bus. After the access time, t_{ACC}, has expired, at instant t_3, a MEM/PER IP core sets its data lines at valid states. The BW_MEM/PER wrapper encodes a binary coded data into a CDMA coded and forwards them via a CSSB back to BW_CPU. The BW_CPU decodes them and at t_4 passes CPU_DATA signals to the CPU IP core. At instant t_4 the BW_MEM/PER generates a signal RDY_deactivate and signals to the CPU IP core to deassert the wait state period, t_w. At t_5 the CPU IP core terminates its Read cycle.

The timing of a Write cycle is simpler in respect to the Read cycle. The main difference is that, during a Write cycle, at instant t_1 the CPU IP core generates an address, and after that, it generates a valid data.

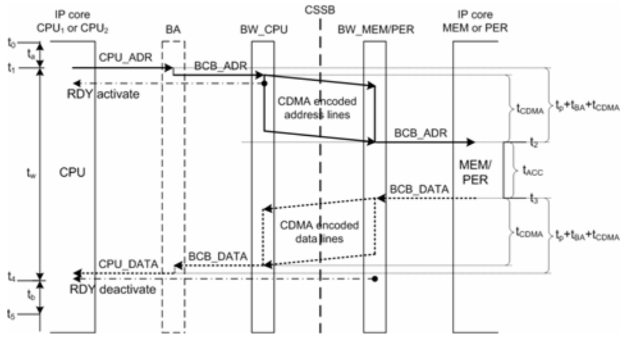

Fig. 3. Signaling scenario of a Read processor cycle

V. Performance Metrics

In order to evaluate latencies of a system based on standard Binary Coded Bus, BCB (Fig. 1), in respect to CDMA Shared System Bus, CSSB (Fig. 2), we will assume the following:

a) Multiprocessor systems given in Fig. 1 and 2 are composed of k processors (masters), and master-slave data transfer rates, for both kinds of systems, will be considered.

b) Binary Coded Bus (BCB_1 or BCB_2) consists of 32 bit address bus, 32 bit data bus, and Control bus (see Fig. 2).

c) The velocity of signal propagation over bus wires is $2*10^8$ m/s, and the master-slave distance is, in average, 30 cm. Accordingly, the signal propagation delay is $t_p = 1{,}5$ ns.

d) Time delays involved by both types of arbiter logic, given in Fig. 1 and 2, are identical, $t_{BAS} = t_{BAC} = t_{BA} = 10$ ns;

e) Access times to all slave modules are identical, $t_{ACCMEM} = t_{ACCI/O} = t_{ACC} = 30$ ns;

f) An address decoder is installed in each slave module of Fig. 1, only, and involves time delay t_D. In our case $t_D = 3$ ns.

g) The total time delay involved by a CDMA coding and decoding process (t_{CDMA} - see Fig. 3), for different spreading code sizes is generated by a CAD tool.

h) According to a bus allocation policy, bus requests in a multiprocessor system given in Fig. 1 will be served in a sequential manner. Contrary, for a multiprocessor system pictured in Fig. 2 all requests will be served simultaneously.

In order to evaluate data transfer latency for both multiprocessor systems (Fig. 1 and 2) we will consider first a timing related to Write, and Read operation. We will assume that $t_{BA} \cong 7 \cdot t_P$, $t_D = 2 \cdot t_P$, and $t_{ACC} = 20 \cdot t_P$ is valid.

A. Standard system (Fig. 1): When there is single issue for master-slave data transfer, Write and Read operation will be served for

$$T_{TS1W} = t_P + t_{BA} + t_D + t_{ACC} = 30 \cdot t_P \quad (1)$$

$$T_{TS1R} = 2*t_P + 2*t_{BA} + t_D + t_{ACC} = 38 \cdot t_P \quad (2)$$

When all k master modules issue requests for data transfer, the access times for Write and Read operation will be

$$T_{TSW} = k * T_{TS1W} = k * 30 \cdot t_P \quad (3)$$

$$T_{TSR} = k * T_{TS1R} = k * 38 \cdot t_P \quad (4)$$

B. CDMA based system (Fig. 2): In this case all k master-slave connections, during Write and Read operation, are performed simultaneously for

$$T_{TCW} = t_P + t_{BA} + t_{CDMA} + t_{ACC} = 28 \cdot t_P + t_{CDMA} \quad (5)$$

$$T_{TCR} = 2*t_P + 2*t_{BA} + 2*t_{CDMA} + t_{ACC} = 36 \cdot t_P + 2*t_{CDMA} \quad (6)$$

where t_{CDMA} is a time interval generated by a CAD tool.

Form now a performance metric, Q, as a product of the following three parameters: a) a number of bus lines, L; b) total access time, T; and c) communication bandwidth, B:

$$Q = L \cdot T \cdot B \quad (7)$$

For the standard system (Fig. 1) and CDMA bus based system (Fig. 2), respectively, we have:

$$Q_{1X} = L_1 \cdot T_{TSX} \cdot B_1 \quad (8)$$

$$Q_{2X} = L_2 \cdot T_{TCX} \cdot B_2 \quad (9)$$

where a subscript X is W(R) for Write(Read) operation.

Let define now a data transfer ratio for Write and Read operation, R_W, and R_R, respectively, if B_1=B_2 (i.e. an equal amount of information is transferred by both systems), as

$$R_W = \frac{Q_{1W}}{Q_{2W}} = \frac{L_1 \cdot T_{TSW}}{L_2 \cdot T_{TCW}} \quad (10)$$

$$R_R = \frac{Q_{1R}}{Q_{2R}} = \frac{L_1 \cdot T_{TSR}}{L_2 \cdot T_{TCR}} \quad (11)$$

We use R_W and R_R as convenient range indices by which we will evaluate the latency of the simultaneous CDMA bus transfer in respect to a sequential TDMA bus transfer.

VI. Results

An on-chip interconnect scheme based on CDMA technique is proposed here and its performance related to data transfer latency are evaluated. The wrapper logic was described at RTL using VHDL. For synthesis, routing, and mapping a Xilinx development CAD tool ISE 9.1i was used. The wrapper was implemented on FPGA devices from Spartan2, Virtex4, and Virtex5 series. The results generated by a CAD tool relate to a signal propagation time which corresponds to the total latency of a communication channel (the time interval t_{CDMA} in Fig. 3).

TABLE I
DATA TRANSFER RATIO FOR WRITE AND READ OPERATIONS

Target device from series	Number of masters	TDMA based system			CDMA based system			Data transfer ratio, R_W	Data transfer ratio, R_R
		Number of lines	Access time, T_{TSW}	Access time, T_{TSR}	Number of lines	Total access time, T_{TCW}	Total access time, T_{TCR}		
Spartan3E	2	64	60 t_p	76 t_p	16	169 t_p	318 t_p	1.42	0.95
	4	64	120 t_p	152 t_p	18	311 t_p	602 t_p	1.37	0.90
	8	64	240 t_p	304 t_p	20	593 t_p	1166 t_p	1.29	0.83
	16	64	480 t_p	608 t_p	22	1159 t_p	2298 t_p	1.20	0.77
VirtexE	2	64	60 t_p	76 t_p	16	164 t_p	308 t_p	1.46	0.99
	4	64	120 t_p	152 t_p	18	300 t_p	580 t_p	1.42	0.93
	8	64	240 t_p	304 t_p	20	572 t_p	1124 t_p	1.34	0.86
	16	64	480 t_p	608 t_p	22	1116 t_p	2212 t_p	1.25	0.80
Virtex5	2	64	60 t_p	76 t_p	16	88 t_p	156 t_p	2.72	1.94
	4	64	120 t_p	152 t_p	18	148 t_p	276 t_p	2.88	1.96
	8	64	240 t_p	304 t_p	20	268 t_p	516 t_p	2.86	1.88
	16	64	480 t_p	608 t_p	22	508 t_p	996 t_p	2.74	1.78

Notice: t_p = 1.5 ns; Column referred as Number of lines includes address and data bus lines.

Table I reports the results which relate to data transfer latencies for standard TDMA and CDMA bus based systems. Multiprocessor systems composed of 2, 4, 8, and 16 master and slave modules were considered. We assumed that all master modules simultaneously access to different slave modules, i.e. a contention problem was omitted.

By analyzing the results given in Table I we can conclude the following:

1) Spartan3E and VirtexE series: transfer rates for Write operations are faster for CDMA based bus architecture in respect to TDMA bus architecture. Namely, for Spartan3E series Write operations are from 42% (for 2 processors) to 20% (for 16 processors) faster, while for VirtexE series are from 46% to 25% faster. Contrary, transfer rates for Read operations TDMA bus architecture is faster from 5.2% to 30% for Spartan3E series, and from 1% to 25% for VirtexE series.

2) Virtex5 series: for both Write and Read operations performance related to data transfer rates for CDMA based bus in respect to standard bus architecture are superior. Namely, Write operations are from 188% to 172% and Read operations are from 96% to 78% faster.

Let note that the results given in Table I are illustrative only. Other factors such as bus allocation policy, physical bus wiring limitations, propagation delay involved by the bus arbiter, etc. have to be considered in real applications.

In general, the results given in Table I show that: a) involvement of CDMA bus based system is a trade-off problem between decreased number of bus lines and communication time and it may appeal to applications where bus size (wiring) reduction is imperative; and b) increased data transfer latencies involved by CDMA data transfer are compensated by simultaneous master-slave transfers.

VII. CONCLUSION

With aim to combine the positive attributes of smaller address and data buses as well as to achieve control bus compliance with existing bus conversion protocols (such as AMBA to BVCI) we have proposed a CDMA encoding technique both for address and data buses, but not for a control bus. At behavioral level a wrapper structure was described using VHDL code. For synthesis, routing, and technology mapping a Xilinx development CAD tool ISE 9.1 was used. The wrapper was implemented on FPGA devices from Spartan and Virtex series. We involved convenient range indices called R_W and R_R to determine data transfer rate for Write and Read operations in multiprocessor bus systems that use TDMA and CDMA data transfer techniques. In addition, let note that increased data transfer latencies involved by CDMA data transfer are compensated by simultaneous master-slave transfers.

REFERENCES

[1] J. Kim, et. all, "A Cost-Effective Latency-Aware Memory Bus for Symmetric Multiprocessor Systems" *IEEE Transactions on Computers*, vol. 57, no. 12, pp. 1714-1719, May 2008

[2] T.S.T. Mak, et. all, "On-FPGA Communication Architectures and Design Factors", *In: Proc. Internat. Conference on Field Programmable Logic and Applications*, Madrid; 2006, pp. 1-8

[3] S. Dey, et. all, "Design of Communication Architectures for High-Performance and Energy-Efficient Systems-on-Chips", *In: Jerraya AA, Wolf W, editors. Multiprocessor Systems-on-Chips*, San Francisco: Morgan Kaufman; 2005, pp. 187-222

[4] T. D. Richardson, et. all, "A Hybrid SoC Interconnect with Dynamic TDMA-Based Transaction-Less Buses and On-Chip Networks", *19th International Conference on VLSI Design*, January 2006, pp. 657-664

[5] L. Benini, G. De Micheli, "Networks on chips: A new paradigm for component based MPSoC design", In: Jerraya AA, Wolf W, editors. Multiprocessor Systems-on-Chips, San Francisco: Morgan Kaufmann, 2004, p. 187-222

[6] J. Kim, et. all, "Design of an Interconnect Architecture and Signaling Technology for Parallelism in Communication", *IEEE Trans on VLSI Systems* 2007, 15(8), pp. 881-94

[7] X. Wang, et. all, "Applying CDMA Technique to Network-on-Chip", *IEEE Transactions on VLSI Systems*, Vol. 15, No. 10, October 2007, pp. 1091-1100

[8] ARM AMBA Specification and multi-layer AHB specification (rev 2.0), [Online], available at http://www.arm.com, July 2008

[9] T. Nikolić, M. Stojčev, "CDMA Coded Wrapper-Based SoC Interconnect", *ICEST 2008*, Vol. 2, No. 1, June 2008, Nis, pp. 399-402