

Enhanced Overloaded CDMA Interconnect (OCI) Bus Architecture for on-Chip Communication

Khaled E. Ahmed, Mohammed M. Farag

Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt

Email: k.e.elsayed@ieee.org, mmorsy@alexu.edu.eg

Abstract—On-chip interconnect is a major building block and a main performance bottleneck in modern complex System-on-Chips (SoCs). The bus topology and its derivatives are the most deployed communication architectures in contemporary SoCs. Space switching exemplified by cross bars and multiplexers, and time sharing are the key enablers of various bus architectures. The cross bar has quadratic complexity while resource sharing significantly degrades the overall system’s performance. In this work we motivate using Code Division Multiple Access (CDMA) as a bus sharing strategy which offers many advantages over other topologies. Our work seeks to complement the conventional CDMA bus features by applying overloaded CDMA practices to increase the bus utilization efficiency.

We propose the Difference-Overloaded CDMA Interconnect (D-OCI) bus that leverages the balancing property of the Walsh codes to increase the number of interconnected elements by 50%. Two implementations of the D-OCI bus optimized for both speed and resource utilization are presented. The bus operation is validated on a Xilinx Artix-7 AC701 FPGA kit and the bus performance is evaluated and compared to other existing bus topologies. We also present the synthesis results for the UMC-0.13 μm design kit to give an idea of the maximum achievable bus frequency on ASIC platforms. Moreover, we advance a proof-of-concept HLS implementation of the D-OCI bus on a Xilinx Zynq-7000 SoC and compare its performance, latency, and resource utilization to the ARM AXI bus. The performance evaluation demonstrates the superiority of the D-OCI bus.

Keywords—SoC, CDMA, Bus Architecture, On-Chip Interconnect, CDMA Bus, Multiple Access Interference, Overloaded CDMA.

I. INTRODUCTION

System-on-Chips (SoCs) are getting more and more complex as the feature size of the building transistors scales down. More IP cores can fit on the same die which causes an exponential increase in the interconnection complexity [1]. The performance of individual IP cores used in SoCs is typically optimized by the vendor leaving the task of implementing the on-chip interconnection architecture to the system designer. The task of implementing on-chip interconnects is not trivial since the wiring density directly impacts the system’s performance, resources, and power consumption. In some applications, on-chip interconnects can be the system’s performance bottleneck which necessitates optimizing the interconnect logical topology. Buses and Networks-on-Chips (NoCs) are the most deployed topologies for on-chip interconnect in SoCs [2].

The straightforward approach to realize on-chip communication is space switching exemplified by crossbar switches where every IP core is physically connected by wires to every other element by a dedicated link providing the better achieved connectivity. The interconnect complexity of the crossbar

scales quadratically with the number of on-chip cores [3] rendering it a feasible solution only for a small number of cores. Another common approach to realize on-chip communication is the bus topology which prevails contemporary SoC designs. In the bus topology, Time Division Multiple Access (TDMA) is adopted, where all cores are interconnected to the same bus and bus access is time shared between interconnected elements according to the bus arbitration rules. As the number of on-chip components increases, the efficiency of the TDMA bus decreases due to the bus contention and increased sharing overheads on the bus [4]. Many SoC designs attempt to overcome this problem by employing hierarchical bus topologies at the expense of increasing the interconnect complexity, overhead, and power consumption [5].

The Code Division Multiple Access (CDMA) bus architecture has been proposed as an alternative to the TDMA-based bus topology to overcome the bus contention problem [6]. Direct sequence CDMA (DS-CDMA) is a well-known approach for medium sharing in wireless communication systems where the channel is shared by assigning orthogonal spreading codes called signatures to all transmit-receive pairs sharing the communication channel. Code orthogonality enables channel sharing and is measured in terms of the cross-correlation between spreading codes which equals zero for orthogonal spreading codes. In a CDMA bus, data from each transmit element is spread by XORing data with a unique spreading code or signature. Data spread by different elements are summed together and sent over the bus. All receiver elements simultaneously access the bus and receive the spread data sum. Despreading is achieved by applying correlation operations to the received sum, where each receiver can extract its data by correlating it with the unique signature assigned for each transmit-receive pair. Other advantages of using CDMA for on-chip interconnect include reduced power consumption, fixed communication latency, and reduced system complexity [7].

Table I shows a brief comparison between the basic crossbar, time-shared, and CDMA buses in terms of the wiring complexity, bus throughput, and arbitration overheads [8] [9] for $M \times M$ interconnected elements. The CDMA bus has less wiring complexity than the crossbar and less arbitration overhead than the TDMA bus, thus provides a good compromise of both. Furthermore, the CDMA bus has the advantage of the possibility of increasing the bus capacity by increasing the number of usable spreading codes, as this work suggests, thus increasing the bus throughput compared to the time-shared bus.

The set of spreading codes used in a CDMA system must be orthogonal to each other and any extra codes added to this set induce Multiple Access Interference (MAI) which

TABLE I. CROSSBAR, TIME-SHARED, AND CDMA BUS COMPARISON FOR $M \times M$ INTERCONNECTED ELEMENTS

Topology	Wiring complexity per M bits interconnection	Throughput	Arbitration overhead
Crossbar	M^2	M	1
TDMA	1	1	M
CDMA	$\log_2(M + 1)$	1	1

arises due to the non-zero cross-correlation between non-orthogonal spreading codes. MAI can also appear due to the auto-correlation between asynchronous orthogonal spreading codes. The MAI problem sets a limit on the maximum number of users in a CDMA communication system. Consequently, the maximum number of IP cores that can simultaneously share the CDMA bus interconnect in SoC is limited by MAI. In most spreading code families, the maximum number of synchronous orthogonal codes of length N equals the code length itself.

State-of-the-art techniques in wireless communications consider deploying non-orthogonal codes for data spreading that can still be separated and identified on the receiver side to increase the CDMA system capacity. These techniques are known as overloaded CDMA and are currently employed in synchronous CDMA wireless communication systems [10]. It was proved that the number of spreading codes can be increased by about 300% in noise-free communication channels at the expense of employing more complex decoders like the Maximum Likelihood (ML) decoder [10], [11]. Therefore, in this work, we attempt to apply the overloaded CDMA practices developed in wireless communications to on-chip interconnects to significantly increase the bus capacity without incurring additional overheads limiting the bus performance.

In our previous work, we presented the MAI-Overloaded CDMA Interconnect (M-OCI) bus topology and presented a systematic approach to generate the non-orthogonal MAI-enabled spreading codes. The number of MAI-enabled codes equals 25% of the orthogonal code set size, thus increasing the bus capacity by 25% [12]. In this work, we propose a different code family that increases the bus capacity by 50%. We present the Difference-Overloaded CDMA Interconnect (D-OCI) bus architecture and compare it to the M-OCI and ordinary CDMA bus topologies. We also provide the implementation results of the reference and pipelined architectures of the D-OCI bus optimized for both resource utilization and speed, respectively.

The remaining of this paper is organized as follows: Related work and a brief background about the conventional CDMA bus architecture are presented in Section II. The solution fundamentals and D-OCI bus architecture are described in Section III. Performance evaluation in terms of resources, maximum bus frequency, power consumption, and bus throughput is presented in Section IV. A high-level synthesis (HLS) implementation of the D-OCI bus and its comparison to the AXI bus on a Zynq SoC is advanced in Section V. Conclusions and future work are portrayed in Section VI.

II. BACKGROUND

The classical CDMA bus topology relies on orthogonal Walsh codes to enable bus sharing. Nikolic *et. al.* propose a full CDMA-based bus system in [13] to decrease the number of parallel transfer lines of TDMA buses. Multilevel 2-bit

CDMA in [4] was used as an I/O reconfiguration scheme which also demonstrated a reduction in the bus contention over TDMA. CDMA and TDMA are combined in the CT-Bus where data is communicated over both the time and code domains [7]. CDMA also has been utilized to enable intra-chip communication in NoC topologies. In [14] a CDMA based NoC is compared to a PTP bidirectional ring based NoC. The simulation results show that the CDMA NoC's fixed data transfer latency is equal to the best case latency of the PTP of the same channel width. The fixed data transfer latency of the CDMA NoC is attributed to the concurrent sharing of the communication channel by network nodes. A hierarchical CDMA star NoC topology is presented in [15], it is compared to a pure mesh and a Fat tree topology, the CDMA star NoC has fewer resources and routing complexity than its rivals. In [16], a wireless CDMA NoC architecture was demonstrated to have significantly lower energy dissipation and higher bandwidth than a TDMA NoC.

Most related work addressing CDMA on-chip interconnect investigate architectural and topological enhancements and performance evaluation for the conventional DS-CDMA communication scheme. In this work, we address a different aspect of the CDMA technology for on-chip interconnects. We investigate increasing the bus capacity by applying overloaded CDMA to the existing on-chip CDMA bus topology. We make use of the ordinary CDMA bus architecture presented by Nikolic *et. al.* in [17] with some modifications to develop the overloaded CDMA bus. Therefore, we present a brief overview of the ordinary CDMA bus topology in this section.

Figure 1 shows the block diagram of the conventional CDMA bus. The system is composed of a number of XOR encoders and accumulator-based decoders. In the encoder, an N -chip length binary orthogonal code, generated from the Walsh spreading code family, is XORed with the data bit and sent out serially, indicating that a single bit is spread in a duration of N clock cycles. The number of transmit-receive IP core pairs sharing the bus equals to M where $M \geq N$. For the ordinary CDMA bus topology using Walsh spreading codes $M = N$. Serial streams from all transmitting cores sharing the bus are added together and the sum is represented in binary and sent to a decoding circuit feeding the receiving IP cores.

The decoder is implemented as a wrapper that cross correlates the serialized channel sum with the signature code assigned for the transmit-receive pair. As the spreading codes are generated from the bipolar Walsh code family, decorrelation (despreading) mainly involves two operations: sum multiplication by ± 1 and accumulation. The bus data is passed to the zero accumulator when the current chip value equals to "0" and to the one accumulator when the chip value equals to "1". The one and zero accumulator circuits accumulate their inputs during the decoding cycle and are reset to zero

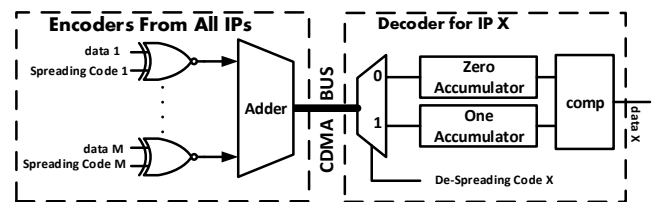


Fig. 1. SoC CDMA XOR encoder and accumulator decoder

at the beginning of each decoding cycle. Consequently, each accumulator adds $N/2$ different inputs during the decoding cycle because the spreading signature codes are balanced. At the end of the decoding cycle, if the zero accumulator content is greater than the one accumulator content, the original data bit is “1”; otherwise, the original data bit is “0”. The choice of Walsh spreading codes is of a particular importance for the design of the overloaded CDMA codes presented in this paper due to two properties: the balancing property which causes a constant difference between the two accumulators at the end of the decoding cycles; and the property of the even difference between the decoding pair to be discussed in the next section.

In our previous work [12], we established a non-orthogonal spreading code family with an AND gate encoder that exploit the steady difference of $\pm N/2$ between the two accumulators to encode extra data and increase the bus capacity. The codes were built such that their effects are mutually exclusive, thus enables errorless detection of the spreading codes. The MAI code family mimics MAI in wireless communications with the main difference is that MAI is controllable, measurable, and encoding data. Unfortunately, the MAI-code design limited the number of the overloaded non-orthogonal codes to only 25% of the spreading code length N .

III. DIFFERENCE-OVERLOADED CDMA INTERCONNECT (D-OCI) CODE DESIGN AND BUS ARCHITECTURE

Our main objective is increasing the number of IP cores sharing the ordinary CDMA bus while keeping the system complexity unchanged by using simple encoding and decoding circuitry. To achieve this goal, we propose slight modifications to the ordinary CDMA bus. Figure 4 shows the overloaded CDMA bus architecture for a single bit interconnect. The same architecture is replicated for multi-bit CDMA bus. M transmit-receive IP core pairs share the CDMA bus, spread data from transmit IP cores are summed together using an arithmetic adder circuit having M bit binary inputs and an output of m -bit width, where $m = \lceil \log_2(M) \rceil$. Each transmit and receive IP core is interfaced to an encoder or a decoder wrapper for data spreading and despreading. The CDMA bus is only used by the data signals while control signals are not interfaced by the CDMA architecture. The destination address of data sent by any transmit IP core is embedded in the signature code which can eliminate the need for an address bus. The bus controller is responsible for assigning spreading and despreading codes and handshaking with the transmit and receive IP cores.

There is an interesting property of the Walsh code family used in the conventional CDMA bus system. The difference between any two consecutive channel sums on the bus produced by any combination of data spread is always even. For the used accumulator decoder, this property forces the difference between the zero accumulator input and the consecutive one accumulator input to be always even. If all data sent is “0”, the bus data at any cycle is either “0” or $\frac{N}{2}$, so if a code is flipped (its encoded data = 1), then the bus data can be either “1” or $\frac{N}{2} - 1$. If the flipped code is used as a despreading code, then the difference between the bus values when the despreading code is “0” and “1” is even ($\pm (1 - (\frac{N}{2} - 1)) = \pm (2 - \frac{N}{2})$). Flipping any other code will not affect the even difference since the codes are orthogonal, any other flipped code will add either “1” or “0” to both accumulator inputs so the Pair

Difference (PD) will remain even. In Figure 2, only three codes are sufficient to illustrate how flipping an orthogonal code does not affect the even difference in a decoding pair.

One can exploit this unique property to design a set of extra non-orthogonal spreading codes and, consequently, increase the bus capacity. We develop a set of non-orthogonal spreading codes that alters the channel sum to produce the odd difference between consecutive bus values at specific time slots. The two cycles where the bus difference is computed are called the decoding pair, and the proposed non-orthogonal codes are called the Pair Difference Spreading (PD_S) codes. The new codes cause MAI to appear on the bus, but it does not invalidate the decoding operation as long as the added MAI does not deviate the accumulator’s difference by more than $N/2$.

Unlike orthogonal spreading codes which are XORed with the binary data bit, we utilize an AND gate to encode the PD_S codes with the binary data bit. The AND gate encoder works as follows: if sent data is “0” it sends a stream of zeros that does not deviate the even bus difference, and if sent data is “1” it sends one of the PD_S codes. Therefore, the additional PD_S code will either make the bus difference between two cycles in a decoding pair even or odd. The XOR encoder of the ordinary CDMA bus cannot be used to encode the PD_S codes because it only complements the spreading code chips, so an XOR gate will cause the difference to be odd whether the data is “0” or “1”. A hybrid encoder is developed for both orthogonal and non-orthogonal spreading with an XOR gate, an AND gate, and a multiplexer unit as shown by Figure 4.

A. Pair Difference Spreading and Despreading Code Design

Before proceeding to the bus architecture, we will discuss how to design the PD_S codes for an arbitrary balanced orthogonal code family of length N . Figure 3 shows the encoding and decoding of four PD_S codes overloaded to the set of 3 codes shown in Figure 2. Let us consider a non-orthogonal PD_S code composed of a first single chip of “1” and all the remaining chips are “0” in the N clock cycles—data encoding and decoding cycle. Assume this code is assigned to an extra IP core sharing the bus. When this core accesses the bus and sends “1”, this code is sent and the single chip of “1” is the input to either the one or zero accumulators in the orthogonal decoders based on the despreading code. This code contributes an MAI value corresponding to only one chip, and the difference D between the accumulators at an orthogonal despreading code accumulator decoder is:

$$D = \pm \frac{N}{2} + 1 \quad (1)$$

The difference between the bus values in the decoding pair is:

$$PD'(k) = PD(k) + 1 \quad (2)$$

where k is the number of decoding pairs, $PD(k)$ is the original even pair difference and $PD'(k)$ is the pair difference after adding the non-orthogonal PD_S code. If $PD'(k)$ is even, then the sent bit is “0”, if $PD'(k)$ is odd then the sent bit is “1”. Thus, the decoded bit at the PD decoder k is the modulo 2 of $PD'(k)$, which can be implemented by XORing the LSBs of the two bus values in the decoding pair. Since the orthogonal codes are balanced, then the number of ones and zeroes in the despreading code is equal and equals to $N/2$. Therefore, the

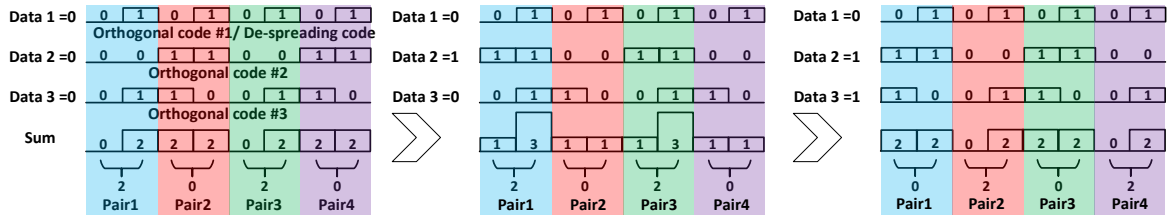


Fig. 2. The balancing property of Walsh codes: flipping any of the orthogonal codes does not affect the even difference in a decoding pair.

number of decoding pairs is $N/2$ which is also the maximum number of non-orthogonal PD_S codes that can be added to the bus because the accumulator difference D sign might be changed if the number of the added chips exceeds $N/2$ invalidating detection of orthogonal spreading codes. Since k is the number of decoding pairs, then it ranges from 1 to $N/2$. A shift register is needed to hold the first value of the bus pair till the second value arrives in order to XOR the two values.

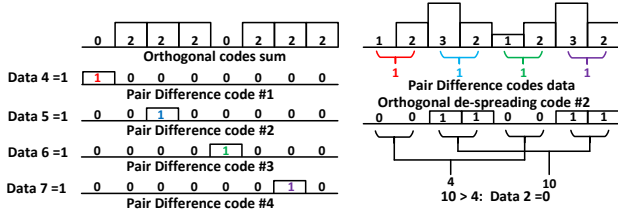


Fig. 3. Encoding and decoding of four PD_S codes overloaded to three orthogonal codes.

To simplify designing the decoder circuit, we can select the Pair Difference Despreading (PD_D) code to be $\{0, 1, 0, 1, 0, 1, \dots\}$. Thus the first decoding pair is $Bus(1)$ and $Bus(2)$, the second is $Bus(2)$ and $Bus(3)$ and so on. This results in a simple shift register structure because the required difference is between two successive decoder inputs. The first PD decoder requires an N -bit shift register since the two values to be subtracted arrive first on the bus and should be held till the N^{th} decoding cycle. The second requires an $(N - 2)$ -bit shift register, and so forth. The last PD decoder requires only a 2-bit shift register. Hence the total number of the needed 1-bit shift registers is $\frac{N^2}{4} + \frac{N}{2}$. Dividing this number by the total number of PD decoders of $N/2$ yields $\frac{N}{2} + 1$ registers per PD decoder. The PD_D code can be any one of the codes in the orthogonal code set since the despreading code must be both orthogonal and balanced in order to yield the even difference in a decoding pair. To minimize the width of registers per PD decoder, the chips in every decoding pair must be adjacent to eliminate the need to store bits between the two chips.

For the orthogonal signature decoders, the difference between the two accumulators is no longer $\pm N/2$ because of MAI caused by the non-orthogonal PD_S codes. However, a comparator circuit can still detect data encoded by orthogonal spreading codes by comparing the one and zero accumulator contents, as long as the total MAI value contributed by non-orthogonal codes is less than $N/2$ to preserve the sign of the difference and consequently facilitate orthogonal code despreading. To clarify this we present this example for a code length $N = 8$, the selected PD_D code is:

$$PD_D = \{0, 1, 0, 1, 0, 1, 0, 1\} \quad (3)$$

which is the concatenation of four consecutive decoding pairs. We can generate the PD_S codes using the designed despreading

code. Generally, $PD_S(k) = 2^l$ where l is the location of the next "0" chip in the despreading code counted from the LSB upwards. Therefore, for $k = \{1, 2, 3, 4\}$, $l = \{7, 5, 3, 1\}$, and the PD_S codes are:

$$\begin{aligned} PD_S[1] &= 2^7 = \{1, 0, 0, 0, 0, 0, 0, 0\} \\ PD_S[2] &= 2^5 = \{0, 0, 1, 0, 0, 0, 0, 0\} \\ PD_S[3] &= 2^3 = \{0, 0, 0, 0, 1, 0, 0, 0\} \\ PD_S[4] &= 2^1 = \{0, 0, 0, 0, 0, 0, 1, 0\} \end{aligned} \quad (4)$$

Thus, each PD_S code either adds only a single chip to a decoding pair or does not, according to the data to be sent. Generally speaking, we can say that there are $N/2$ cycles to encode PD_S bits and $N/2$ free cycles.

B. Basic and Optimized Difference-Overloaded CDMA Interconnect (D-OCI) Decoder Architectures

The non-orthogonal PD decoder is only required to find the difference between the bus values at two different bus cycles inside a decoding pair. As illustrated by Figure 4, the transmit IP cores are interfaced to the encoder wrappers, and the receiving memory/peripheral units (MPUs) are connected to the decoder wrappers. We apply a static code allocation scheme where each transmit-receive pair has a fixed signature code, the added $N/2$ decoders are connected to $N/2$ MPUs. There are $1.5N$ encoders and $1.5N$ decoders, the decoders are decomposed into N orthogonal decoders and $N/2$ PD decoders that decode data for the $N/2$ PUs as explained previously. Encoders are configured by applying specific spreading codes, according to the intended communication link. If the intended link uses an orthogonal spreading code, the XOR encoder is selected; otherwise, the AND encoder is selected.

We implemented two variants of the bus, a reference architecture, and a pipelined architecture. The reference architecture is a direct implementation of the spreading and despreading circuitry without adding any non-functional registers. The pipelined architecture is implemented to increase the bus operating frequency and, consequently, throughput by adding non-functional registers to reduce the bus critical path. Two pipelining registers are inserted around the bus adder circuit as shown in Figure 4. The encoded data register holds data encoded by the orthogonal and PD encoders while the sum register holds the adder output to be passed to the decoding circuitry. Thus, the critical path inside the CDMA bus circuitry is reduced to include the longest path in the three parts, which is usually the adder circuit. This architecture can be pipelined further by breaking the critical path in the adder circuit, but at the expense of adding more pipelining registers. The bus register is m -bit wide where $(m = \lceil \log_2 1.5N \rceil)$ for the orthogonal decoders, but only 1-bit wide for PD decoders since only the LSB is required for the PD decoding.

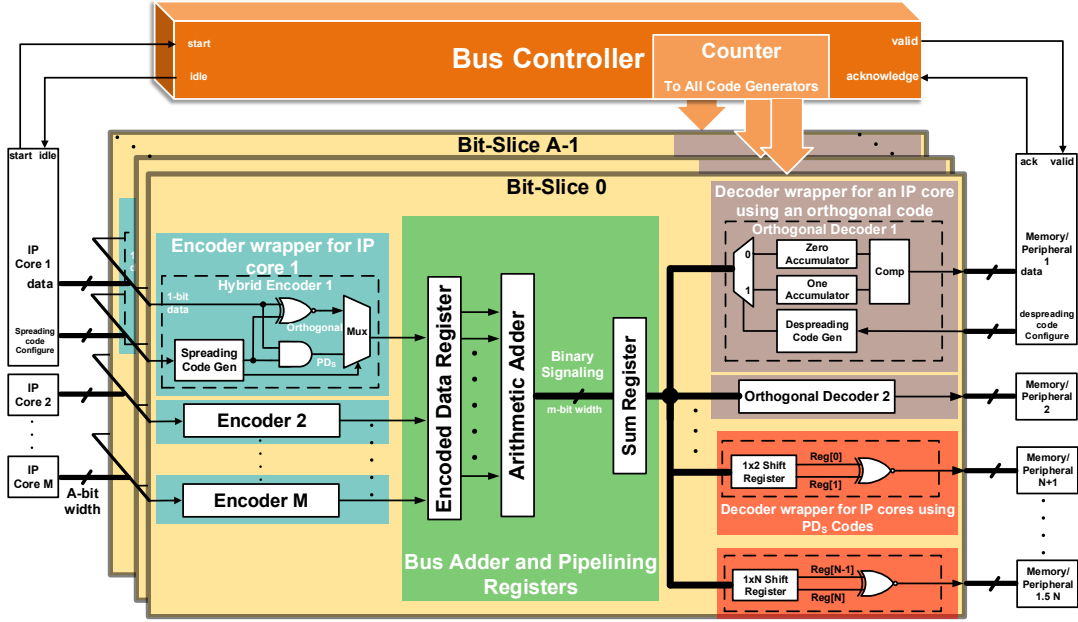


Fig. 4. Pipelined Difference Overloaded CDMA bus system containing the hybrid encoder, and both the orthogonal and the PD overloaded codes decoders.

IV. PERFORMANCE EVALUATION

A. Overloaded CDMA Interconnect (OCI) Bus Evaluation

In this section, we present the evaluation results of the overloaded CDMA bus. A system containing a number of IP cores and peripheral devices was built with full capacity, i.e. the number of IP cores is the maximum number offered by the bus. All CDMA bus variants are implemented and validated on an Artix-7 AC701 evaluation kit. Specifically, we compare between the conventional CDMA bus, M-OCI bus, basic and pipelined D-OCI bus variants for different spreading code lengths (number of chips) $N = \{8, 16, 32, 64\}$. To establish a fair comparison between different bus architectures connecting a number of elements, all performance metrics are normalized to the number of interconnected elements, i.e. all performance metrics for a bus interconnecting M IP cores are divided by M to evaluate bus performance per IP core. Evaluation results, including resource utilization expressed in LUTs and Flip-Flops per IP core, maximum bus frequency, dynamic power consumption per IP core, and the bus bandwidth are shown in Figure 5. To give an idea about ASIC implementation of the CDMA bus, initial synthesis results of the bus using UMC-0.13 μm ASIC cell library are illustrated in Figure 5.

As depicted by Figure 5(a), 5(b), for a fixed spreading code length N , resource utilization per IP core of the M-OCI and D-OCI buses is less than the ordinary CDMA bus by 25% and 50% for M-OCI and D-OCI, respectively. This resource reduction per IP core is due to the significant increase in bus capacity compared to the marginal overhead added by the OCI circuitry. Also, for a fixed spreading code length N , the D-OCI has further fewer resources per IP core compared to the M-OCI due to the increase in the overloading percentage. Increasing the spreading code length N increases the resource utilization per IP core due to the increase in the bus complexity. Specifically, with increasing N , the size of the bus adder and accumulator decoder circuitry increases. Another note worth mentioning in Figure 5(a), 5(b) is that the resource utilization

of the pipelined D-OCI bus is always larger than the basic D-OCI bus due to the added non-architectural pipelining registers.

For all CDMA bus variants, the operating frequency is limited by the critical path length, including the spreading circuit, channel adder, and accumulator decoder components. For various CDMA buses of the same spreading code length N , orthogonal spreading and despreading circuits are identical, non-orthogonal data encoders and decoders are running parallel to the orthogonal spreading circuitry with a shorter critical path length, and the input size of the adder circuit is equal to the number of transmit IP cores M which varies with the CDMA bus type. Figure 5(c) illustrates that for a fixed spreading code length N , the bus frequency of the overloaded CDMA buses is less than the basic CDMA bus frequency due to the increase in the adder circuit size. The pipelined design isolates the critical path at the CDMA bus adder tree which improves the maximum bus frequency at the expense of the extra non-architectural registers and output latency. The bus frequency decreases with increasing N for both overloaded and ordinary CDMA buses due to increasing the computational complexity of the adders as shown by Figure 5(c). The operating frequency of the UMC-0.13 μm implementation of the CDMA bus is about $10x$ greater than the FPGA implementation counterparts.

With increasing N , the drop in frequency is compensated by the increase in the bus bandwidth due to the capacity enhancement offered by the overloaded buses as shown by Figure 5(d). The bus bandwidth is plotted for only a single bit per IP interconnected via the CDMA bus. For fixed N , we can clearly see the enhancement of the bus bandwidth for the D-OCI bus over the M-OCI and conventional CDMA buses, and the enhancement of the pipelined D-OCI bus bandwidth over the basic D-OCI bus. Generally, the CDMA bus bandwidth BW is given by the following equation:

$$BW = \frac{N_{bits} * f_b * M}{N} \quad (5)$$

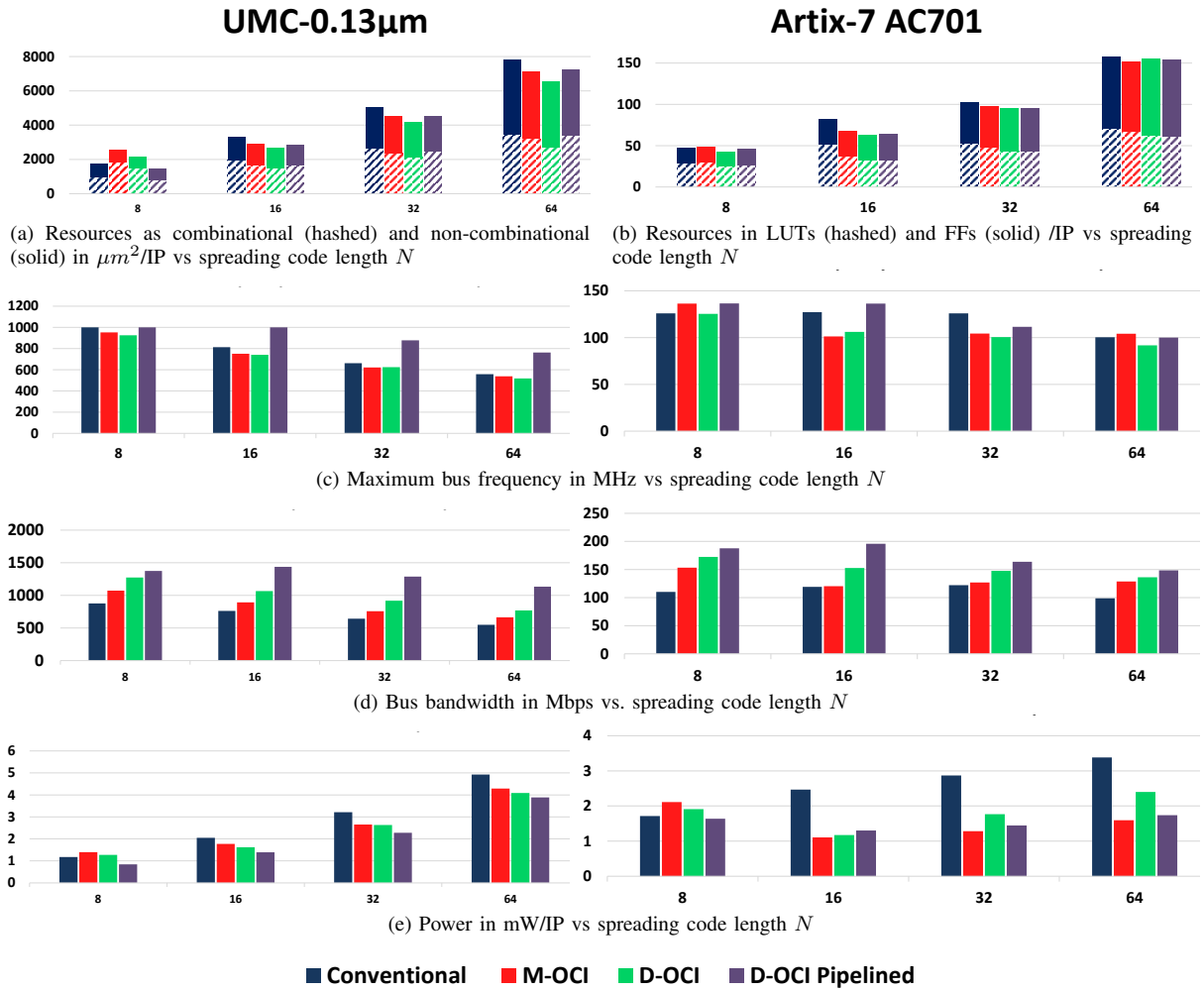


Fig. 5. Synthesis and implementation results of the overload CDMA bus for code length $N = \{8, 16, 32, 64\}$.

where N_{bits} is the number of interconnected bits per IP core (data bus width), f_b is the bus frequency, M is the number of transmit-receive core pairs sharing the bus, and N is the spreading code length. The M-OCI and D-OCI bandwidth has significant improvement over the ordinary CDMA bus as they have an overloading ratio of $\frac{M}{N} = 1.25, 1.5$, respectively, compared to the basic CDMA bus ratio of $\frac{M}{N} = 1$.

As illustrated by Figure 5(e), for a fixed spreading code length N , power dissipation per IP core is decreased for the M-OCI and D-OCI buses due to the offered capacity enhancement. For increasing N , power dissipation per IP core increases for all CDMA buses due to the increased size and complexity of the bus components. The aforementioned conclusions apply for both the ASIC and FPGA implementations of the bus. However, the routing overhead in the D-OCI increases the dynamic power consumption over the M-OCI in the FPGA platform. The ASIC synthesis (pre-place and route) results do not include routing information, so the D-OCI appears to have less power consumption than the M-OCI.

B. OCI Bus Comparison to Other Interconnect Topologies

In order to evaluate the CDMA interconnect performance relative to TDMA and SDMA, we implemented the basic architecture for both TDMA and SDMA buses illustrated in

Figures 6 and 7, respectively. The TDMA bus is basically composed of multiplexer and demultiplexer circuits back-to-back connected as shown in Figure 6. An arbiter module is responsible for selecting modules to be connected in specific time slots according to specific access priorities and arbitration rules. Access time is divided between the elements sharing the bus and the bus utilization cannot be increased beyond 1 because only M transmit-receive pairs can access the bus in M time slots. Though the arbitration overhead in TDMA buses is significantly large, we only implement the switching elements without the arbiter in order to assess the basic concept without details. The SDMA bus depicted in Figure 7 is mainly composed of M multiplexers each has M inputs to facilitate connecting $M \times M$ elements without blocking communication for any element. The SDMA bus dedicates a physical link between every pair of interconnected elements which provides uninterrupted communication, but at the expense of increasing resource utilization. A new multiplexer is needed for every additional transmit-receive pair and the complexity of existing multiplexers increases due to the additional input/output pair.

The SDMA and TDMA buses of Figures 6 and 7 and the basic CDMA bus of Figure 1 are implemented on the Xilinx Artix-7 AC701 kit and the synthesis results are illustrated in Figure 8. The resource utilization is expressed as the number of LUTs and FFs. As depicted in Figure 8(a), the resource utiliza-

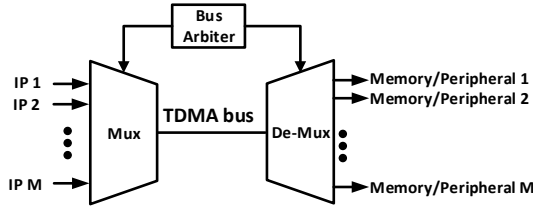


Fig. 6. Basic TDMA bus topology

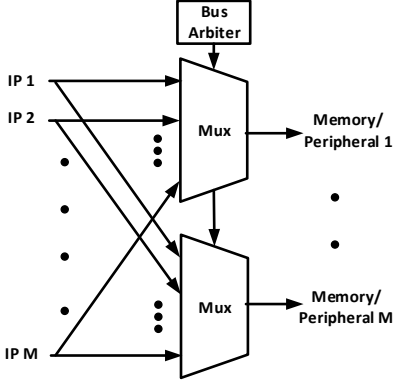


Fig. 7. Basic SDMA bus topology

tion in the case of the TDMA bus is constant $\approx M/M \approx 1$. For the SDMA bus, the resource utilization $\approx M^2/M = M$ which follows the linear trend shown in Figure 8(a). The CDMA bus resource utilization $\approx M \log_2(M)/M = \log_2(M)$ which results in a logarithmic utilization trend. The bandwidth of the SDMA bus, on the other hand, is M -folds the constant bandwidth of the SDMA and CDMA buses which is depicted by the log scaled bandwidth comparison of Figure 8(c). Increasing wiring complexity reduces the bus frequency, thus the TDMA bus can achieve a higher bus frequency than the CDMA bus, which in turns has a higher bus frequency than the SDMA bus as shown by Figure 8(b). Dynamic power consumption of the CDMA bus shown in Figure 8(d) is significantly higher than that of the S/TDMA buses due to the larger number of deployed registers. Nevertheless, the power consumption of the CDMA bus approaches the SDMA bus as M increases due the quadratic increase in wiring complexity of the SDMA bus.

The above analysis illustrates that the conventional CDMA bus has a higher area setback when compared to the TDMA bus but offers an equal bandwidth. Also, the conventional CDMA bus has lower bandwidth setback against the SDMA bus, but a much smaller area. Therefore, the OCI bus architecture helps in enhancing the two setbacks by increasing the bus bandwidth and reducing the resource utilization per IP core. In other words, the OCI buses can increase the bus bandwidth by overloading the channel, which cannot be achieved with TDMA buses, at the expense of increasing the bus complexity to be in order of $\log_2 M$, which is much less than the SDMA bus. Thus, the OCI bus is a good compromise for SoCs requiring higher bandwidth than that achieved using TDMA bus topologies but significantly less area than SDMA bus architectures. The implementation results reinforce the theoretical comparison provided in Table I.

The CDMA interconnect has a larger bandwidth per area ratio of $1/\log_2(N+1)$ compared to the SDMA interconnect ratio of $1/N$. This ratio is significantly increased by M -

OCI and D-OCI to $1.25/\log_2(N+1)$ and $1.5/\log_2(N+1)$, respectively. The OCI buses have less bandwidth to resources ratio than the TDMA bus because the added complexity is significantly larger than the increased bandwidth. An exclusive advantage of the OCI bus over T/SDMA buses is the capability of over utilizing the communication channel. In the TDMA bus, each time slot can carry a maximum of 1-bit while, in the SDMA bus, each PTP link can also carry no more than 1-bit. Conventional CDMA buses can also carry a maximum of N bits per N time slots or one bit per time slot. Fortunately, the OCI bus has the ability to increase the channel utilization beyond one via channel overloading.

Table II provides a comparison between the OCI bus and other interconnect architectures presented in the literature in terms of resources, capacity, frequency, bandwidth and bandwidth to resources ratio. It should be noted that in all the compared interconnects, a full system was implemented including bus arbiters. As shown by Table II, the D-OCI bus has the highest bandwidth to resources ratio compared to other interconnect topologies. We should indicate that OCI is still in its initial development phase where various architectural and functional optimizations can be still applied to compete with the state-of-the-art HOT interconnects. For instance, in this work, we only presented the OCI bus architecture to illustrate the bus overloading idea. However, we can also use the same concept to build the OCI NoC architecture which can significantly increase the interconnect bandwidth.

V. HLS AND AUTOMATION OF THE D-OCI BUS IP CORE

In the last section, we only presented a simple implementation of the D-OCI bus interconnecting a set of elements generating test data and we compared the D-OCI bus to other CDMA bus variants. However, such an evaluation is not sufficient to prove the bus competency with the deployed long-established SoC buses. We are currently automating the generation of the D-OCI bus IP core to facilitate bus deployment in modern SoCs. In this section, we present a direct comparison between an initial prototype of the D-OCI core and the ARM AXI bus deployed in Xilinx Zynq-7000 SoC. The D-OCI IP core is implemented using the Xilinx Vivado HLS design flow while the AXI bus IP core is provided by Xilinx. The HLS C to RTL flow allows automated quick implementation and verification of the core, Vivado HLS tool also offers compiler directives for RTL optimization. The AXI bus is chosen for this comparison due its widespread deployment in modern SoCs, the availability of a number of bus variants for different performance needs, and its extensive support by different vendors and CAD tools. Moreover, the AXI bus does not require a fixed number of power of 2 of connected elements, which facilitates its comparison to the D-OCI bus.

A SoC bus testbed shown in Figure 9 is built on a Zynq-7000 SoC to evaluate the D-OCI bus. The testbed comprises the Bus Under Test (BUT), M master and M slave IP cores, an ARM processor, and a controller described as follows:

- The BUT:** The D-OCI bus is compared against the AXI crossbar switch, namely the Shared Address Multiple Data (SAMD) mode, and against the AXI TDMA switch, namely the Shared Address Shared Data (SASD) mode.
- Master IP cores:** The M master IPs in this testbed are capable generating one data beat to be written on one slave.

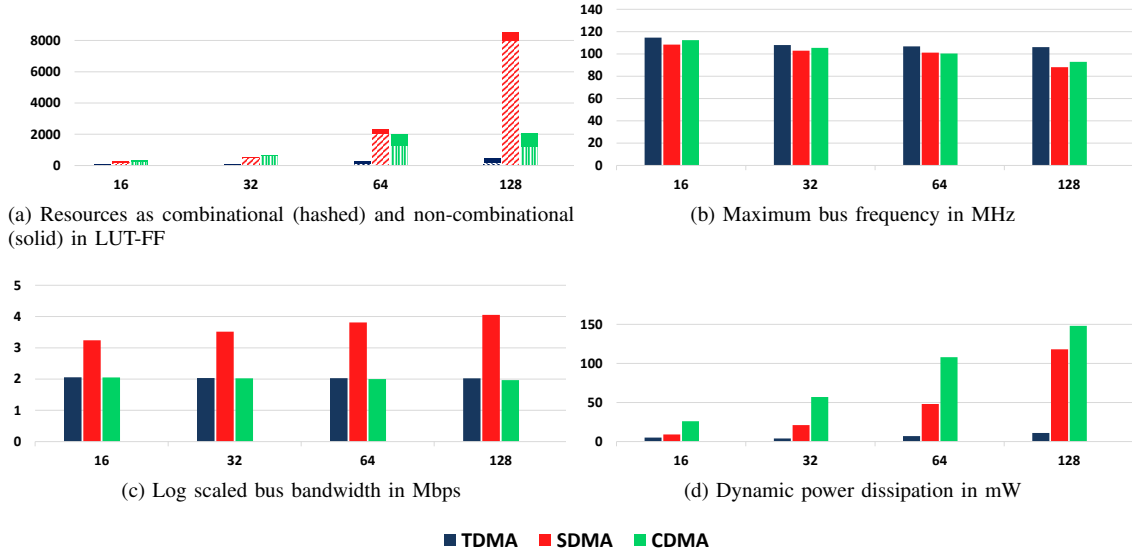


Fig. 8. Synthesis and implementation results of TDMA, SDMA and CDMA bus topologies of $M \times M$ size for $M = \{16, 32, 64, 128\}$.

TABLE II. AREA, FREQUENCY, BANDWIDTH AND BANDWIDTH TO RESOURCES RATIO OF THE D-OCI BUS VERSUS OTHER INTERCONNECTS

Bus Topology	Implementation Technology	Bus Capacity Masters \times Slaves	Resources (Gate Count)	Frequency (MHz)	Bandwidth (Mbps)	Bandwidth to resources ratio (Mbps/Gate Count)
D-OCI	ASIC 0.13 μm	11 \times 11	1,268	1000	1,375	1.08
CDMA router [15]	ASIC 0.18 μm	7 \times 7	47,754	94.2	21,100	0.44
CDMA NoC [14]	ASIC 0.18 μm	6 \times 6	272,806	Asynch	7,410	0.027
PTP NoC [14]	ASIC 0.18 μm	6 \times 6	177,007	Asynch	6,756	0.038
D-OCI	Artix7	11 \times 11	503	146	187	0.37
CDMA wrapper [18]	Virtex5	4 \times 4	2,064	-	587.6	0.28
PTP MPEG [19]	Virtex2	7 \times 7	43,248	-	4714	0.108
TDMA MPEG [19]	Virtex2	7 \times 7	40,048	-	3,669	0.09
NoC MPEG [19]	Virtex2	7 \times 7	41,768	-	4,622	0.11

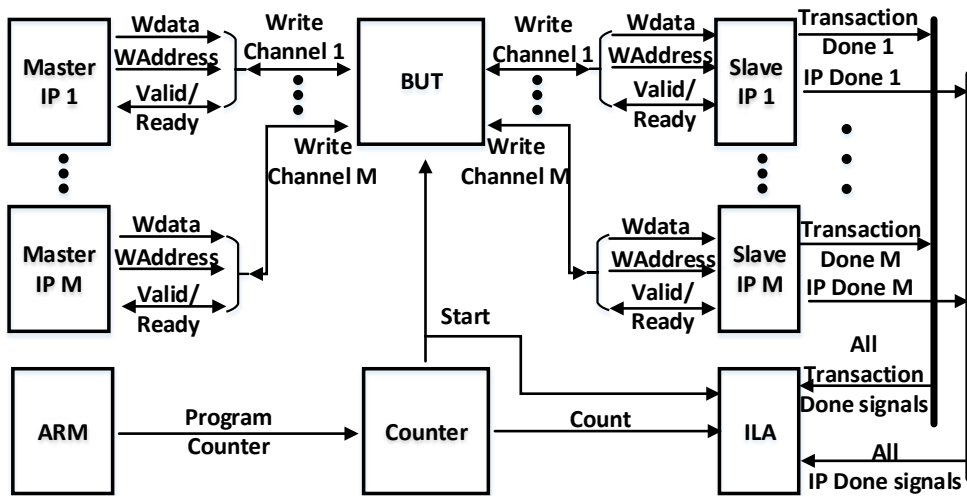


Fig. 9. SoC testbed in for the D-OCI and AXI bus architectures

The data beat and the address of the slave is specified at the compile time. When the BUT is the D-OCI, the master IPs are connected to the D-OCI via a data bus and address bus with valid/ready handshake signals. When the AXI is used as the BUT, the master IPs are connected to the AXI by an AXI bus wrapper provided by Xilinx.

c) **Slave IP cores:** the S slave IPs polls the bus for the data beat and asserts a “transaction done” signal to validate that data is received correctly. The correct data that each slave should receive is known at the compile time. The slave IPs are connected to the D-OCI by a data bus and valid/ready handshake signals. The AXI bus wrapper is also used to

connect the slaves to the AXI bus.

- d) **An Integrated Logic Analyser (ILA) and a counter:** The counter is used to as a measure of the clock latency of BUT. The counter also has a one hot “start” output that starts the entire system at a fixed count. The ILA is used to probe the “transaction done” signals indicating the correctness of the received data at each slave, the ILA also probes a “done” signal for each slave indicating that the slave is no longer probing for data. The bus clock latency can be measured by the number of counts of the counter between issuing the “start” signal when all slaves assert the “done” signal.
- e) **An ARM processor:** the processor is used to validate the bus operation, trigger the ILA probing, and start/stop or clear the counter to monitor the testbed performance.

A comparison between the D-OCI and AXI buses in terms of resource utilization and write latency in clock cycles for different bus sizes is illustrated in Table III. The testbed runs at a 100 MHz clock frequency, the D-OCI IP core has only the write channel implemented while the AXI bus has the read, write, and write response channels implemented. Two different implementations of the D-OCI bus with $N = 8$ and 16 are tested, which results in 11×11 and 23×23 bus size, respectively. The crossbar and TDMA implementations of the 11×11 and 16×16 AXI bus are tested, while the 23×23 AXI bus is not implemented due to limitations of the Xilinx CAD tools which can only generate up to 16×16 AXI bus. The timing diagram of the six tested buses shown in Figure 10 is obtained by the ILA probing of the “start”, “done” and “transaction done” signals of the six tested buses. The implementation and simulation results can be analyzed as follows:

- a) The 11×11 D-OCI bus utilizes 97% and 90% fewer resources than the 11×11 AXI crossbar and TDMA switch, respectively, while the 23×23 D-OCI utilizes 94% and 80% fewer resources than the AXI crossbar and TDMA switch, respectively. The resource utilization ratio is calculated between the combined number of LUTs and FFs for the compared buses. The huge increase in the resource utilization of the AXI crossbar can be attributed to three factors. Firstly, crossbars are space switching elements using a dedicated physical link for each interconnect. Secondly, the AXI bus has 3 working channels while the D-OCI bus has only one. Finally, the master and slave IP cores contain AXI bus wrappers, which increases the utilization of the masters and slaves and congests the design and causes the placement and routing tool to duplicate AXI resources in order to meet timing constraints. The increase in the AXI TDMA switch resource utilization is mainly due to the decoding, arbitration, and control overheads.
- b) The 11×11 D-OCI bus latency is 13 clock cycles while the 11×11 AXI TDMA bus latency is 122 clock cycles (89% reduction). The AXI TDMA bus latency is significantly larger than the D-OCI bus because it serves only one write request at a time without pipelining the requests [20].
- c) The 11×11 D-OCI bus latency of 13 cycles is also better than the 11×11 AXI crossbar bus latency of 42 clock cycles, about 70% reduction. The 23×23 D-OCI bus latency of 22 cycles is less than the 16×16 AXI crossbar bus latency of 61 cycles, about 64% reduction. This improvement in latency is because addressing in the D-OCI bus is performed once at the beginning of

the transaction by assigning different spreading codes to different masters, while in the AXI bus addresses are sent in a TDMA fashion due to the shared address channel [20].

- d) Finally, the implementation results show that the achievable bandwidth of the D-OCI CDMA bus is significantly greater than that of both AXI crossbar and TDMA bus configurations. This can be attributed to that the arbitration and control units of the D-OCI bus are not fully implemented yet. It should be indicated that in case of a burst transfer mode, which is not implemented yet in the D-OCI, the AXI crossbar should be about N times faster the D-OCI bus, where N is the spreading code length. This speedup is due to that in burst transfers the addressing phase is only performed once, then in every N clock cycles the AXI crossbar can send N data beats while the D-OCI bus can send only 1 data beat from each IP core. This implies that the D-OCI IP can be better utilized for application requiring single data beat transfers like those with a substantial number of random reads and write requests.

VI. CONCLUSIONS

In this work, we presented the enhanced OCI bus architecture, namely D-OCI, which improves the conventional CDMA bus capacity by 50%. The D-OCI topology can replace the TDMA topology to implement on-chip interconnects in either a bus or a NoC router. We exploited the balancing property of the spreading code family employed in the classical CDMA bus to increase the number of IP cores sharing the bus without altering the simple accumulator decoder of the conventional CDMA bus. A systematic generation procedure of the non-orthogonal spreading and despreading codes is presented along with two optimized, reference and pipelined, implementations of the bus architecture. The D-OCI bus topology was implemented and validated on an Artix-7 AC701 FPGA development kit and the UMC $0.13\mu\text{m}$ ASIC technology.

We compared the D-OCI bus performance with the conventional CDMA bus and M-OCI bus presented in our previous work. The reference D-OCI bus achieves 21% higher bandwidth over the conventional bus on the FPGA platform, while the pipelined D-OCI bus achieves 34% more bandwidth. The dynamic power consumption on the FPGA is reduced by 29% and 48% for the reference and pipelined D-OCI, respectively. Initial ASIC synthesis results show that the D-OCI bus utilizes less cell area, consumes less power, and can achieve a bus frequency of up to 1 GHz, which promotes the deployment of the D-OCI bus in ASIC platforms. We also compared between the basic CDMA, SDMA, and TDMA bus implementations and proved that the CDMA bus is the only topology that can increase the bus utilization efficiency beyond one. We presented a proof-of-concept prototype of the D-OCI bus and compared it to the ARM AXI bus in its two modes of operation, the crossbar and TDMA. The resource utilization and clock latency comparisons between the D-OCI and AXI buses demonstrate the significant improvement of the CDMA bus architecture over the TDMA and SDMA bus architectures.

Many directions for future work are inspired by this research. We aim to develop a fully-functional prototype of the OCI-bus IP core. Functional and architectural optimizations will be investigated to improve the OCI bus performance. Also, we will investigate increasing the OCI bus capacity by

TABLE III. UTILIZATION AND WRITE LATENCY OF THE D-OCI IP VS AXI BUS

Bus Topology	Bus Capacity $M \times M$	LUTs	FFs	Latency clock cycles	Frequency MHz	Bandwidth Gbps
D-OCI N = 8	11 × 11	177	222	13	109	2.951
D-OCI N = 16	23 × 23	487	567	22	113	3.78
AXI SAMD-Crossbar	11 × 11	8,229	5,651	42	104	0.871
AXI SAMD-Crossbar	16 × 16	11,299	7,833	61	93	0.78
AXI SASD-TDMA	11 × 11	2,123	1,761	122	107	0.309
AXI SASD-TDMA	16 × 16	2,919	2,532	177	105	0.304

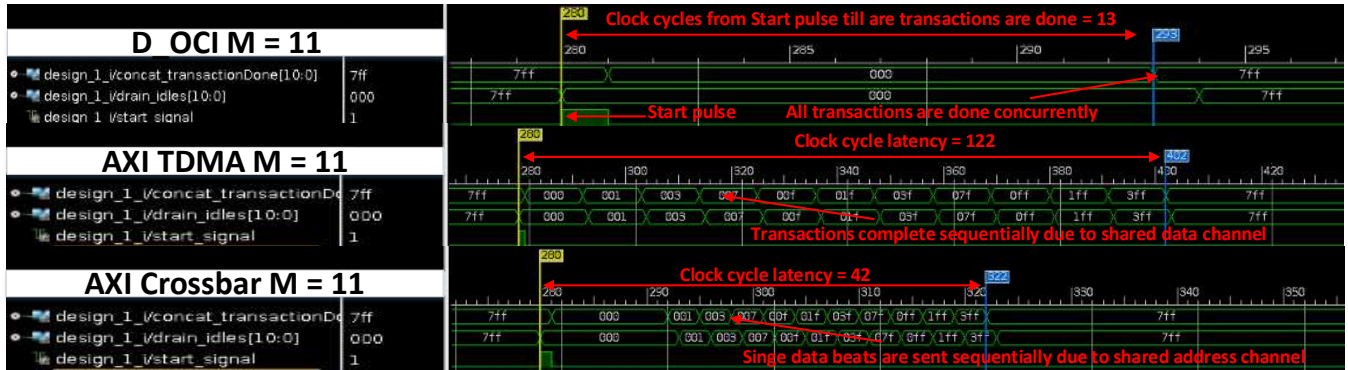


Fig. 10. Write latency of D-OCI bus vs AXI crossbar and AXI TDMA, The clock latency is measured from the instance when the bus starts till all slave IPs receive the data beats.

expanding the spreading code set and using low-complexity orthogonal decoders other than the accumulator decoder such as the ML decoder presented in [21]. We will investigate applying the OCI concepts to the NoC CDMA architecture to enhance the interconnect bandwidth and power consumption.

REFERENCES

- [1] R. Ho, K.W. Mai, and M.A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, Apr 2001.
- [2] Ling Wang, Jianye Hao, and Feixuan Wang. Bus-based and NoC infrastructure performance emulation and comparison. In *Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on*, pages 855–858, April 2009.
- [3] G. Passas, M. Katevenis, and D. Pnevmatikatos. The combined input-output queued (CIOQ) crossbar architecture for high-radix on-chip switches. *Micro, IEEE*, PP(99):1–1, 2014.
- [4] Jongsun Kim, I. Verbauwhede, and M.-C.F. Chang. Design of an interconnect architecture and signaling technology for parallelism in communication. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 15(8):881–894, Aug 2007.
- [5] M Mitic, M Stojcev, and Z Stamenkovic. An overview of SoC buses. In Vojin G Oklobdzija, editor, *Digital Systems and Applications*. CRC Press, 2007.
- [6] Jr. Bell, R.H., Chang Yong Kang, L. John, and E.E. Swartzlander. CDMA as a multiprocessor interconnect strategy. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, volume 2, pages 1246–1250 vol.2, Nov 2001.
- [7] B.-C.C. Lai, P. Schaumont, and I Verbauwhede. CT-bus: a heterogeneous CDMA/TDMA bus for future SoC. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, volume 2, pages 1868–1872 Vol.2, Nov 2004.
- [8] Sudeep Pasricha and Nikil Dutt. Chapter 2 - basic concepts of bus-based communication architectures. In Sudeep Pasricha and Nikil Dutt, editors, *On-Chip Communication Architectures*, Systems on Silicon, pages 17 – 41. Morgan Kaufmann, Burlington, 2008.
- [9] Sudeep Pasricha and Nikil Dutt. Chapter 3 - on-chip communication architecture standards. In Sudeep Pasricha and Nikil Dutt, editors, *On-Chip Communication Architectures*, Systems on Silicon, pages 43 – 100. Morgan Kaufmann, Burlington, 2008.
- [10] Seyed Amirhossein Hosseini, Omid Javidbakht, Pedram Pad, and Farrokh Marvasti. A review on synchronous CDMA systems: optimum overloaded codes, channel capacity, and power control. *EURASIP Journal on Wireless Communications and Networking*, (1):1–22, 2011.
- [11] Kasra Alishahi, Shayan Dashmiz, Pedram Pad, Farrokh Marvasti, M. H. Shafinia, and M. S. Mansouri. The enigma of CDMA revisited. *CoRR*, abs/1005.0677, 2010.
- [12] K.E. Ahmed and M.M. Farag. Overloaded CDMA bus topology for MPSoC interconnect. In *ReConfigurable Computing and FPGAs (ReConFig), 2014 International Conference on*, pages 1–7, Dec 2014.
- [13] Tatjana Nikolic, Mile Stojcev, and Goran Djordjevic. CDMA bus-based on-chip interconnect infrastructure. *Microelectronics Reliability*, 49(4):448 – 459, 2009.
- [14] Xin Wang, T. Ahonen, and J. Nurmi. Applying CDMA technique to network-on-chip. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 15(10):1091–1100, Oct 2007.
- [15] Daewook Kim, Manho Kim, and Gerald E Sobelman. Design of a high-performance scalable CDMA router for on-chip switched networks. *Memory*, 8:01100110, 2005.
- [16] A. Vidapalapati, V. Vijayakumar, A. Ganguly, and A. Kwasinski. NoC architectures with adaptive code division multiple access based wireless links. In *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pages 636–639, May 2012.
- [17] T. Nikolic, G. Djordjevic, and M. Stojcev. Simultaneous data transfers over peripheral bus using CDMA technique. In *Microelectronics, 2008. MIEL 2008. 26th International Conference on*, pages 437–440, 2008.
- [18] T. Nikolic, M. Stojcev, and Z. Stamenkovic. Wrapper design for a CDMA bus in SOC. In *Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010 IEEE 13th International Symposium on*, pages 243–248, April 2010.
- [19] Hyung Gyu Lee, Naehyuck Chang, Umit Y. Ogras, and Radu Marculescu. On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches. *ACM Trans. Des. Autom. Electron. Syst.*, 12(3):23:1–23:20, May 2008.
- [20] Xilinx. *UG761-AXI Reference Guide*. 2012.
- [21] M. Li. Fast code design for overloaded code-division multiplexing systems. *Vehicular Technology, IEEE Transactions on*, PP(99):1–1, 2015.