Aggregated CDMA Crossbar for Network-on-Chip

Khaled E. Ahmed, Mohamed R. Rizk, Mohammed M. Farag
Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt
Email: k.e.elsayed@ieee.org, mrmrizk@ieee.org, mmorsy@alexu.edu.eg

***Abstract*—Code Divison Multiple Access (CDMA) is proposed as the physical layer enabler of Network-On-Chip (NoC) interconnects for its prominent features such as fixed latency, guaranteed service, and reduced system complexity. CDMA interconnects have been adopted by the NoC community as it originates in wireless communications where each bit in a CDMA encoded data word is transmitted on a separate channel to avoid interference. However, the wireless interference problem can be efficiently mitigated in on-chip interconnects eliminating the need for replicating the CDMA channel. Moreover, wireless channels are sequential by nature which is not the case in on-chip interconnects where parallel buses are the default communication means. When CDMA was adopted by the NoC community, the same wireless CDMA scheme has been maintained where each data bit is encoded in a separate CDMA channel and the encoding/decoding logic is replicated for data packets. In this work, we present a novel CDMA encoding/decoding scheme called Aggregated CDMA (ACDMA) for NoC interconnects in which all packet bits are encoded in a single CDMA channel, consequently, eliminating the area and energy overheads resulted from replicating the channel encoding/decoding logic. The ACDMA NoC crossbar is synthesized on a 45-*nm* standard-cell process. Compared to the conventional CDMA NoC crossbars, the presented method achieves 60.5% less area, 55% less power consumption, and 124% more throughput per area ratio.**

***Index Terms*—NoC, On-Chip Interconnect, CDMA Crossbar.**

I. Introduction

Modern Systems-on-chips (SoCs) are becoming massively parallel with many harmoniously interconnected Processing Elements (PEs). Interconnecting the PEs is commonly achieved through buses and Networks-on-Chips (NoCs) [1]. In NoCs, exchanged data is bundled into packets and traverse several network layers passing by the physical layer which defines how packets are actually transmitted between NoC units. The physical layer of a NoC is implemented by routers employing crossbar switches. Code Division Multiple Access (CDMA) is a medium sharing technique that leverages orthogonal codes to enable simultaneous packet routing. Unlike time-shared channels, CDMA leverages the code space to enable channel sharing. CDMA has been proposed as an on-chip interconnect technique for both bus and NoC interconnect architectures [2]. Many advantages of using CDMA for on-chip interconnects include reduced power consumption, fixed communication latency, and reduced system complexity [3].

Utilizing CDMA in NoC interconnects is adopted from the wireless communications literature, where the data is spread by orthogonal codes at the transmitters, the spread data are added on the wireless channel, and the received sum is decoded at the receivers. Classical CDMA systems rely on the Walsh orthogonal code family to enable medium sharing. Many research groups have investigated several aspects of CDMA in NoCs, including our group which presented the Overloaded CDMA for on-chip Interconnects (OCI) [4] [5] [6].

A 14-node CDMA-based network has been developed in [7]. The network utilizes 7 Walsh codes and assignment of the Walsh codes to the network nodes is dynamic based on the request from each node. Two architectures have been introduced in [7]: a serial CDMA network where each data chip in the spreading code is sent in one clock cycle; and a parallel CDMA network where all data chips are sent in the same cycle. The serial and parallel CDMA-based networks have been compared to a conventional CDMA network, a mesh-based NoC, and a Time Division Multiple Access (TDMA) bus. For the same network area, the throughput of the parallel CDMA network is higher than that of the mesh-based NoC and the TDMA bus due to the simultaneous medium access nature of CDMA. Standard-basis codes are proposed as a replacement to Walsh CDMA codes in [8]. Standard-basis codes resemble TDMA signaling because each code consists of only a single chip of one and the remaining chips are zeros. The TDMA codes' orthogonality enables them to replace the Walsh codes as spreading and despreading CDMA codes, which reduces the complexity of the channel adder and decoder as the sum of TDMA codes is limited to zero or one per clock cycle.

The conventional CDMA crossbar employed in the literature is depicted in Figure 1. The crossbar interconnects N transmit ports to N receive ports using N-chip length Walsh spreading codes. The binary data from each transmit port is encoded using an XOR encoder; the data bit is XORed with a unique N-chip spreading code assigned to the transmit-receive pair and transmitted in N clock cycles. Data spread from all encoders are added by the CDMA channel adder and sent to all receive port. The decoder at each receive port extracts the data from the channel sum by correlating the channel sum with the assigned spreading code. The correlation operation is implemented using an accumulator and a multiplexer since the despreading code chips are unipolar ("0" or "1").

In all of the CDMA interconnect related work, each data bit in a data word is encoded and transmitted in a separate CDMA channel and the encoding/decoding logic is replicated W times for data packets of width W which is a direct application of the wireless CDMA principles in NoC interconnects. However, wireless communication channels are sequential by nature due to the interference problem. Multiple access and MIMO techniques can enable concurrent data transmission on the same wireless channel at the expense of increasing the transmitter/receiver complexity. In on-chip interconnects, on

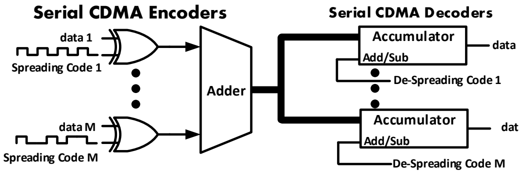


Figure 1. Conventional CDMA crossbar [6].

the other hand, a single channel can be efficiently utilized to enable parallel data transmission on a single CDMA channel as noise and interference effects can be efficiently mitigated [9]. In this work, we present a single channel, multi-bit CDMA crossbar namely Aggregated CDMA (ACDMA) NoC crossbar. The forthcoming discussion is organized as follows; the mathematical foundation of the aggregated CDMA scheme is developed in Section II. Architectural details and a complexity analysis of the ACDMA NoC crossbar are presented in Section III. Implementation results are advanced in Section IV and the conclusions are drawn in Section V

II. ACDMA Mathematical Foundations

Symbols presented in this work are defined in Table I. In the ACDMA crossbar, the encoder multiplies a W-bit width data flit from the transmitting port by a CDMA code. The crossbar transaction takes N clock cycles. At the i^{th} clock cycle, the data flit from the transmitting port j is multiplied by the i^{th} chip from the CDMA code of the port's encoder:

$$E_j^i = d_j \ C_j^i. \tag{1}$$

Since the CDMA code chip value is ±1, then the multiplication can be efficiently computed by negating d_j when $C_j^i = -1$ which is mathematically reduced to:

$$E_j^i = (d_j \ XOR \ U_j^i) + U_j^i. \tag{2}$$

where $U_j^i = (1 - C_j^i)/2$. Data encoded from all encoders are added up, the channel sum at the i^{th} clock cycle can be expressed as:

$$S^i = \sum_{j=0}^{N-1} E_j^i = \sum_{j=0}^{N-1} d_j \ C_j^i. \tag{3}$$

It should be indicated that the random effects such as noise and fading are neglected in the above equation, which is justified and experimentally validated for digital on-chip interconnects [9]. At the receiving port side, the decoder cross-correlates the channel sum with the despreading code by multiplying the channel sum with the CDMA code chips and accumulating over the duration of the N cycles:

$$X_k^l = \sum_{i=0}^{l} S^i \ C_k^i = \sum_{i=0}^{l} (\sum_{j=0}^{N-1} d_j \ C_j^i) \ C_k^i. \tag{4}$$

After N decoding cycles: $l = N - 1$ and the decoder equation can be evaluated by applying the distributive and associative properties of the addition operation:

$$X_k^{N-1} = \sum_{j=0}^{N-1} d_j (\sum_{i=0}^{N-1} \ C_j^i \ C_k^i). \tag{5}$$

Table I
Definition of Notations

Symbol	Description
N	Number of CDMA code chips
W	Port width (the number of bits in a flit)
d_j	Data from port j
C_j^i	The i^{th} chip of the CDMA code from transmitter j
U_j^i	The unipolar representation of C_j^i
E_j^i	CDMA encoded data from transmitter j at the i^{th} clock cycle
S^i	The channel sum of all CDMA encoded data at the i^{th} clock cycle
X_k^l	The output of the k^{th} orthogonal decoder at the l^{th} clock cycle

Since the CDMA code set is orthogonal, the sum $\sum_{i=0}^{N-1} \ C_j^i \ C_k^i$ is equal to N when $j = k$ and is equal to zero otherwise. Thus, the output of the decoder after N decoding cycles is:

$$X_k^{N-1} = N \ d_k. \tag{6}$$

Since the despreading code chips are ±1, the multiply-accumulate process of the decoder can be implemented simply by an up/down accumulator. The data d_j can be extracted from the decoder output X_k^{N-1} by shifting the result $log_2 N$ bits to the right simply by rewiring.

III. ACDMA NoC Crossbar Architecture

The ACDMA crossbar implements the psychical layer of the NoC by interconnecting N transmit (TX) ports to N receive ports where the data width of each port is W where $W = log_2 \ max(d_j)$. The high-level architecture of the ACDMA crossbar illustrated in Figure 2(a) is composed of three main parts; encoders, channel adder, and decoders. The encoders spread data from each TX port using W XOR gates as shown in Figure 2(b). Instead of adding the spreading chips of the Walsh orthogonal code to the result in the encoder block as suggested by (2), this operation is postponed to the channel adder block in order to merge the channel adder with the spreading code adders. The output of each encoder is, therefore, limited to W-bit width.

The encoder outputs are then added together to form the sum S^i of (3). To minimize the critical path of the channel adder, the addition is done using a tree adder architecture as depicted by Figure 2(c) where the leafs of the tree are the encoders of each TX port, and the root of the tree is the channel sum output. Because there are N leaves, the height of the tree is $log_2(N)$. The width of the output wires from each adder in the tree is equal to the width of the input wires plus one to prevent overflows. Since the input to the firs level of adders is $(W+1)$-bit wide and the height of the adder tree is $log_2(N)$, then the width of the output wires at the root adder is $W+1+log_2(N)$. Pipeline registers are inserted after each stage in the tree to minimize the critical path of the channel.

The sum S^i is then sent to all the N decoders, a decoder per RX port. The decoders implement the cross-correlation of (4) in a cost efficient manner; the decoder consists of only

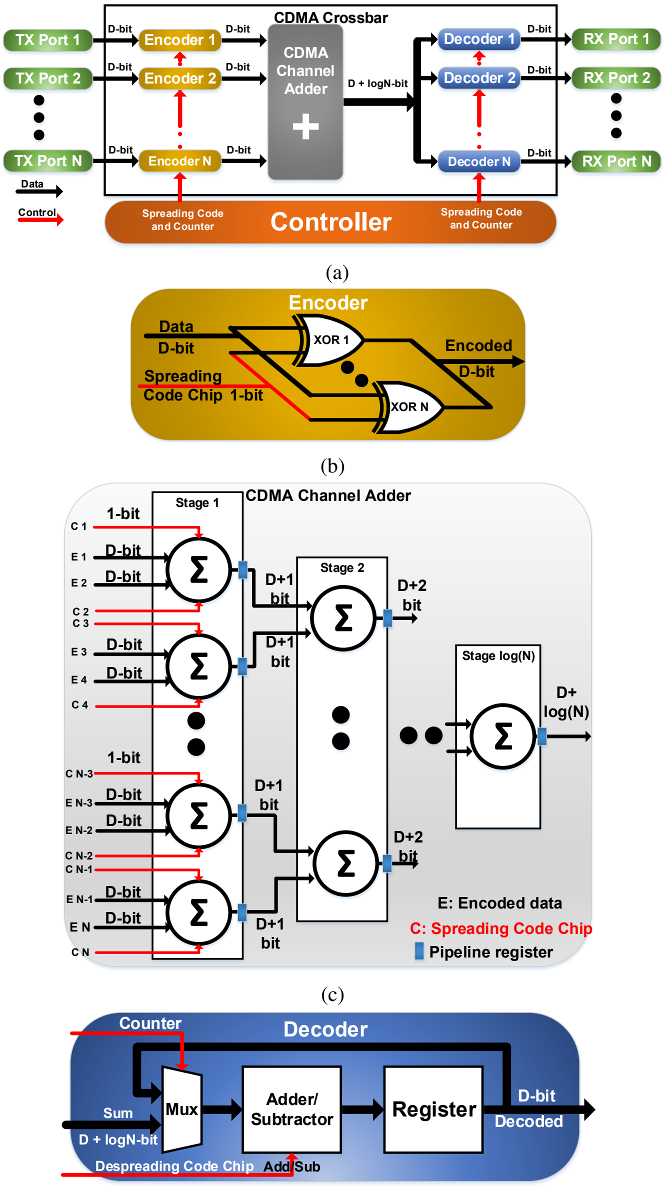


Figure 2. (a) ACDMA crossbar high-level architecture, (b) ACDMA encoder, (c) ACDMA channel adder, (d)ACDMA decoder.

an adder/subtracter and a register configured as an up/down accumulator as shown in Figure 2(d). Since the despreading code C_k consists of ±1 chips, cross correlation is reduced to simple addition and subtraction operations of consequent sums S^i. Therefore, the decoder is implemented as an up/down accumulator; the adder/subtracter adds or subtracts the sum S^i from the result saved in the registers according to the value of despreading chip C_k^i. In particular, when the despreading chip is '1', the adder adds S^i to the contents of the register but subtracts S^i from the contents of the register when the despreading chip is '-1'. At the end of the decoding cycle, the accumulator register holds Nd_k according to (5), and because $N = 2^n$ and n is an integer, data d_k is decoded by shifting the accumulator content by $log_2(N)$ bits.

In Table II, analysis of the ACDMA versus conventional CDMA crossbar complexity is presented. The number of two input XOR gates is the same for both circuits. The improvement of the ACDMA crossbar over the conventional CDMA crossbar is evident in the number of channel adder wires; in the conventional CDMA crossbar, the number of the adder wires for the single-bit channel is increased by one in each stage due to the additional carry bit. Therefore, the number of adder wires in stage i is equal to $1+log_2(N)-i$. For a W-bit word, the number of adder wires is increased to $W+W(log_2(N)-i)$, and since there are 2^i adders at each stage, then the total number of wires is equal to $\sum_{i=0}^{log2N-1} 2^i(W+W(log_2N-i))$. In the ACDMA crossbar, conversely, the number of adder wires for a W-bit word is $W+log_2(N)-i$, which makes the total number of wires equals to $\sum_{i=0}^{log2N-1} 2^i(W+log_2N-i)$ which is a factor of W less than that of the conventional CDMA crossbar. The reduced number of carry bits of the ACDMA crossbar is the prime reason for its superiority. The number of wires for the decoder accumulator and the number of flip-flops in the decoder registers is proportional to the number of channel wires—the last stage of the adder. This follows that the complexity of the ACDMA crossbar is in an order of W less than that of the conventional CDMA crossbar.

Table II
COMPLEXITY ANALYSIS OF THE CONVENTIONAL AND ACDMA CROSSBARS.

Component	Aggregated CDMA	Conventional CDMA
Encoders	$N\ W$ XOR gates	$N\ W$ XOR gates
Channel adder wires	$\sum_{i=0}^{log_2N-1}$ $2^i(W+log_2N-i)$	$\sum_{i=0}^{log_2N-1}$ $2^i(W+W(log_2N-i))$
Decoder accumulators	$N(W+log_2N)$ $=N\ W+N\ log_2N$	$N\ W(1+log_2N)$ $=N\ W+N\ W\ log_2N$

IV. IMPLEMENTATION RESULTS

In this section, the implementation results of the ACDMA crossbar are presented. The crossbar is synthesized using a $45-nm$ standard-cell process; the synthesis results are compared to the conventional Walsh-based (WB) CDMA crossbar and the standard-basis (SB) NoC crossbar presented in [8]. However, to neutralize any discrepancy between the implementation of the ACDMA crossbar and the implementations presented in [8] that may arise due to different synthesis strategies, the WB and SB crossbar implementation results are reproduced for the same technology. The results in [8] are derived from a single bit interconnect where the data flit of width W is serialized by a parallel to serial converter and transmitted bit by bit using the single bit CDMA interconnect. Therefore, to transmit W bits in parallel, the single bit interconnect of [8] is replicated W times in the reproduced implementation compared herein with the ACDMA crossbar.

The area is estimated using Synopsys Design Compiler, activity factors are estimated by ModelSim to aid the Synopsys Design Compiler to accurately estimate the power. Table III compares the area A in μm and total – leakage and dynamic– power dissipation of the ACDMA crossbar against the WB and SB crossbars. The IoWB% and IoSB% columns indicate

Table III
AREA IN μm^2 AND POWER DISSIPATION IN mW OF THE WB, SB, AND ACDMA CROSSBARS FOR $N = 8$ NODES (CODE CHIPS) AND $W = 4$ BITS.

Modules		Area (μm^2)					Power (mW)				
		WB [8]	SB [8]	ACDMA	IoWB%	IoSB%	WB [8]	SB [8]	ACDMA	IoWB%	IoSB%
8 Nodes	Encoder and adder	1806.2	01511.9	1082.1	40.1	28.4	0.422	0.29	0.227	46.2	21.7
	Decoder	3756.3	1696	1522.8	59.5	10.2	0.764	0.286	0.304	60.2	-6.3
	Total	5562.5	3207.9	2604.9	53.2	18.8	1.186	0.576	0.531	55.2	7.8
16 Nodes	Encoder and adder	7294.3	6290.3	2849.9	60.9	54.7	1.367	0.801	0.597	56.3	25.5
	Decoder	12448.2	6703.2	4939.3	60.3	26.3	2.275	0.985	1.042	54.2	-5.8
	Total	19742.5	12993.5	7789.2	60.5	40.1	3.642	1.786	1.639	55	8.2

the percentage improvement of the ACDMA crossbar over WB and SB crossbars, respectively. The clock frequency of the architectures presented in Table III is fixed at 2.5 GHz to facilitate the comparison with the results in [8]. To arrive at this clock frequency, the number of bits W is set to four, because as W increases, the carry chain in the adders reduces the clock frequency which does not occur in conventional CDMA. However, as illustrated by Table III, the savings in the area compared to the WB and SB crossbars are up to 60.5% and 40.1%, respectively. Furthermore, the ACDMA crossbar achieves up to 55.2% and 8.2% reduction in power dissipation compared to that of the WB and SB crossbars, respectively.

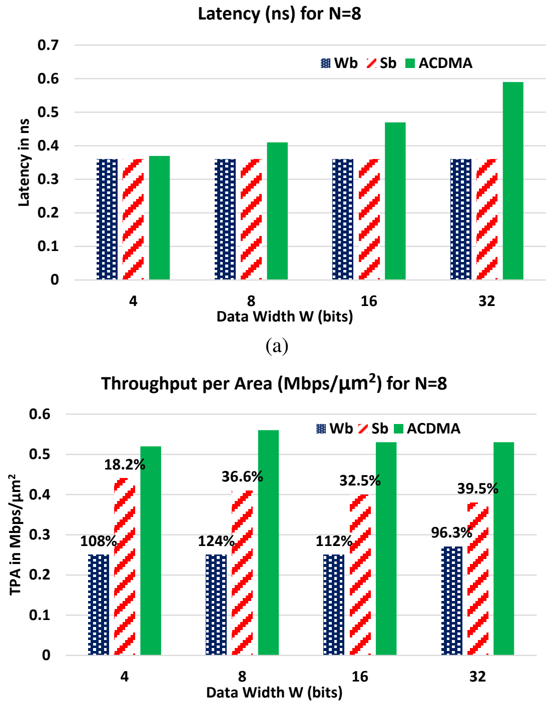
The latency L in *ns* of the ACDMA crossbar is compared to that of the SB and WB crossbars in Figure 3(a). Due to the increase in the carry chain length in the channel adder with the increase in the data width W, the latency of the channel adder and the ACDMA crossbar latency, consequently, increases. Conversely, the carry chains of the WB and SB crossbars are invariant with the data width W since the adders are replicated W times instead of increasing the adder input width as in the ACDMA crossbar. However, due to the substantial reduction in total area as illustrated in Table III, the Throughput per Area (TPA), calculated through (7), of the ACDMA crossbar is still higher than that of the SB and WB crossbar counterparts.

$$TPA = \frac{W}{L\ N\ A} \tag{7}$$

The TPA of the WB, SB, and ACDMA crossbars in *Mbps per* μm^2 are juxtaposed in Figure 3(b) and the increase in TPA of the ACDMA crossbar over that of the WB and SB crossbars are up to 124% and 39.5%, respectively.

V. CONCLUSION

In this work, we presented the ACDMA NoC crossbar to enable parallel transmission of multi-bit data packets on a single CDMA channel. The overhead of channel replication is mitigated which results in up to 60.5% area and 55% power savings with 124% improvement in throughput per area compared to the conventional CDMA crossbar. As a future work, we plan to build and evaluate a full ACDMA-based NoC under different workloads and routing protocols.



(a)

(b)

Figure 3. Latency (a) in *ns* and Throughput per Area (b) in *Mbps/*μm^2 for the WB, SB, and ACDMA crossbars.

REFERENCES

[1] L. Wang, J. Hao, and F. 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.

[2] R. H. Bell, 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.

[3] 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.

[4] K. E. Ahmed and M. M. Farag. Overloaded CDMA bus topology for MPSoC interconnect. In *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*, pages 1–7, Dec 2014.

[5] K. E. Ahmed and M. M. Farag. Enhanced overloaded CDMA interconnect (OCI) bus architecture for on-chip communication. In *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, pages 78–87, Aug 2015.

[6] K. E. Ahmed and M. M. Farag. Parallel overloaded CDMA interconnect (OCI) bus architecture for on-chip communications. In *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 621–624, Dec 2015.

[7] Basel Halak, Teng Ma, and Ximeng Wei. A dynamic CDMA network for multicore systems. *Microelectronics Journal*, 45(4):424 – 434, 2014.

[8] J. Wang, Z. Lu, and Y. Li. A new CDMA encoding/decoding method for on-chip communication network. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(4):1607–1611, April 2016.

[9] Jacob Postman and Patrick Chiang. A survey addressing on-chip interconnect: energy and reliability considerations. *ISRN Electronics*, 2012, 2012.