Aggregated CDMA Crossbar with Hybrid ARQ for NoCs

MOHAMMED M. FARAG1,2, (MEMBER, IEEE) KHALED E. AHMED3, (Member, IEEE),
1Electrical Engineering Department, College of Engineering, King Faisal University, Al-Ahsa, Saudi Arabia (e-mail: mfarag@kfuf.edu.sa)
2Electrical Engineering Department, Faculty of Engineering, Alexandria University, Egypt (e-mail: m-morsy@alexu.edu.eg)
3Electrical Engineering Department, Faculty of Engineering, Alexandria University, Egypt (e-mail: k.e.elsayed@ieee.org)
Corresponding author: Mohammed M. Farag (e-mail: mfarag@kfuf.edu.sa).

This work was supported by the Deanship of Scientific Research, King Faisal University, Saudi Arabia [grant number: NA000224]

ABSTRACT Network-on-Chips (NoCs) are the dominant interconnection technique in modern System-on-Chips (SoCs). NoC routers constitute the physical layer of on-chip interconnects while the medium access technique utilized in NoC routers profoundly impacts the performance and footprint of the router. Code Division Multiple Access (CDMA) is one of the prominent access techniques employed in the physical layer crossbar of NoC routers. Adopted from the wireless communications realm, the CDMA-based crossbar relies on spreading the transmitted data bit on the physical channel using orthogonal codes, summing the data spread from various transmitters, then decoding the received sum using correlation decoders to extract the sent binary bit. However, due to multiple access interference and random channel effects notorious in wireless communications and following the same exact approach in on-chip CDMA interconnects, existing CDMA crossbars are bit-wise architectures where binary data bits are spread and serially communicated on an exclusive digital channel while replicating this configuration to communicate multi-bit packets which increases the crossbar area overhead.

We advance an elegant modification to improve the area and power consumption of the classical CDMA crossbars. Aggregated CDMA (ACDMA) relies on high wiring density in modern SoCs in addition to its static nature and relative noise immunity to improve the classical CDMA crossbar architecture by aggregating the transmission of multiple data bits on the same channel which reduces the overhead of ordinary CDMA-based NoCs. In this work, the ACDMA mathematical foundations, crossbar hardware architecture, and full NoC realizations are presented. The implementation results on the 65 nm standard-cell ASIC technology show a significant improvement in the resource utilization and power consumption of the ACDMA crossbar and ACDMA-based NoCs compared to the ordinary CDMA and CONNECT NoC counterparts. Moreover, the ACDMA interconnect reliability in AWGN channels is studied and improved using a hybrid ARQ approach tailored to the ACDMA crossbar.

INDEX TERMS Network-on-Chip, CDMA NoC, Aggregated CDMA Crossbar, NoC Physical Layer, CDMA Interconnect.

I. INTRODUCTION

SYSTEMS on chips (SoCs) are blooming as an efficient design methodology as a consequence of the reduced transistor size, lower production cost, and higher packaging density. A large proportion of the chip area is, in fact, reserved for the inter system communication fabric. The communication layer must be scalable in terms of its overhead and throughput in order to accommodate the increasing number of Processing Elements (PEs) on the chip. Additionally, the communication layer must be optimized for area, power consumption, and throughput. Networks on chip (NoCs) are the scalable communications paradigm that meet modern SoC requirements, it achieves scalability by relaying on a decentralized physical layer; the NoC’s physical layer is spread upon multiple physical routers realizing the NoC layering model.

Consequently, being at the base of the NoC hierarchy, capable realization of the underlying physical layer of NoC routers impacts the overall performance and area of the entire NoC. Time and Space Division Multiple Access (T/SDMA) techniques are used ubiquitously in modern designs of NoC physical crossbar switches owing to the high throughput of
the SDMA crossbar and the low overhead of the TDMA crossbar. Code Division Multiple Access (CDMA) is another multiple access technique that was lately introduced to implement NoC crossbars. Although CDMA crossbars provide lower throughput than SDMA crossbars, they consume less area and power overhead than TDMA crossbars, making it a favorable alternative to the traditional TDMA crossbars [1].

In an ordinary bit-wise CDMA-based crossbar, a single data bit is spread using a unique spreading code of $N$-chip length, drawn from a pool of $N$ orthogonal codes. In a crossbar of $N \times N$ I/O ports, each transmit-receive pair is assigned a unique spreading code and each data bit from all transmit ports is XORed with the preassigned code. A single data bit is spread into $N$ chips which are serially transmitted in $N$ clock cycles; the crossbar communication rate is single chip/cycle. Data spread from each port is then summed up using a binary channel adder of $N$ binary inputs (a single bit from each transmit port) and $\log_2 N + 1$ binary outputs. The channel sum is finally passed to the CDMA decoders at each receive port to get cross-correlated with the preassigned despreading codes to decode the transmitted data bits. The cross-correlation decoder is implemented using two accumulators equivalent to multiplying the bipolar codes by $\pm 1$ and accumulating. The bit-slice architecture of a conventional CDMA crossbar of $N \times N$ ports is illustrated in Figure 1.

Previous works in CDMA-based interconnects focused on utilizing the classical bit-wise CDMA crossbar switch in implementing buses and NoCs for different applications. The classical CDMA encoding/decoding method is employed in [2] to implement a CDMA bus wrapper with lower inter-connection overhead than TDMA buses at the expense of increased latency. In [3], the scalability of CDMA-based NoCs is demonstrated using a star topology, where each $N$ PEs are interconnected to a single CDMA router employing $N$ orthogonal Walsh codes. A centralized CDMA-based NoC is developed in [4] with a dynamic number of spreading code chips $N$ and is compared to an SDMA-based mesh NoC and a centralized TDMA-based NoC; the dynamic CDMA crossbar offered significant savings in the NoC overhead with lower packet latency than its rivals. In [5], TDMA-based Standard Basis (SB) orthogonal codes are used instead of the Walsh Basis (WB) orthogonal codes to build an SB CDMA crossbar with enhanced performance and area characteristics. The parallel CODEC crossbar presented in [6] is the parallel version of the sequential SB switch where the $N$ time slots composing the SB code length are replaced by $N$ dedicated physical links to accomplish communication between the router’s I/O ports in a single clock cycle instead of $N$ clock cycles. This coding technique is called New Parallel CODEC (NPC). Various Overloaded CDMA Interconnect (OCI) crossbar architectures are presented in [1], [7–9].

In OCI, the Walsh spreading code set is extended with a set of non-orthogonal codes of the same size to double the conventional CDMA crossbar capacity, and consequently throughput, while reducing the normalized overhead per port.

In the majority of previous CDMA interconnect works, the CDMA concept was adopted directly from its wireless communications counterpart, the Direct Sequence Spread Spectrum CDMA (DSSS-CDMA), in which the bit-wise CDMA channel and hardware to transmit one bit must be replicated $W$ times to communicate a $W$-bit word. In this work, we propose the Aggregated CDMA (ACDMA) crossbar to aggregate the transmission of a $W$-bit word on a single communication channel as an analog to the $M$-ary modulation in wireless communications. This is made possible due to the improved channel conditions of digital wires in SoC interconnects over those of wireless channels. Baseband digital CDMA channels are employed instead of the wireless channels in DSSS-CDMA, this in turns reduces the noise margins of the CDMA channel; instead of using an analog channel with $M$-ary modulation, a $\log_2(N)$-bit digital channel is employed. This in turns increases the number of crossbar wires at the benefit of increasing the noise margins. However, the wiring density of the presented ACDMA crossbar is much lower than that of the conventional CDMA crossbars as will be proved.

Thereupon, we advance the CDMA crossbar mathematical foundations assuming an ideal noise-free channel. Afterwards, we study the noise effect on the CDMA channel analytically assuming an Additive White Gaussian Noise (AWGN) channel. Moreover, error detection and correction with Automatic Repeat Requests (ARQ) are implemented in the ACDMA crossbar to further improve the reliability of the ACDMA crossbar. Our previous work [10] presented an initial proof of the ACDMA concept that paved the way to complete the work introduced herein.

The rest of this paper is organized as follows, the ACDMA mathematical foundation is presented in Section II. This is followed by introducing the ACDMA crossbar high-level architecture and complexity analysis in Sections III. Implementation results of the ACDMA crossbars on the 65 nm standard-cell ASIC technology for various design parameters are then evaluated in Section IV. A full realization of an ACDMA-based star NoC is demonstrated and compared to other existing NoCs in Section V. The reliability analysis of the ACDMA-based NoC and improvement method using hybrid ARQ are appraised in Section VI. Related work is presented in Section VII. Conclusion and future work are drawn in Section VIII.
II. ACDMA MATHEMATICAL FOUNDATION

Symbols presented in this paper are defined in Table 1. The ACDMA crossbar architecture is identical to the classical CDMA crossbar illustrated in Figure 1 with the major difference of transmitting and receiving \( W \)-bit words from I/O ports instead of the single-bit configuration shown. Each encoder multiplies a \( W \)-bit width data word from the transmitting port by an \( N \)-chip length CDMA spreading code. The crossbar transaction takes place in \( N \) clock cycles. At the \( i \)th clock cycle, the data word from the transmitting port \( j \) is multiplied by the \( i \)th chip from the CDMA code of the port encoder:

\[
E_j^i = d_j C_j^i
\]

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

\[
E_j^i = (d_j \text{ XOR } U_j^i) + U_j^i
\]

where \( U_j^i = (1 - C_j^i)/2 \).

Data encoded from all encoders are summed up using a binary adder of \( N \) inputs each is \( W \)-bit width and an output of \( (W + \log_2(N)) + 1 \)-bit width. The channel sum at the \( i \)th clock cycle can be expressed as:

\[
S_i^j = \sum_{j=1}^{N} E_j^i = \sum_{j=1}^{N} d_j C_j^i
\]

It should be indicated that random effects such as noise, interference, and fading are neglected in the above equation and these effects will be studied later in this work.

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 \( N \)-cycle duration:

\[
X_k^N = \sum_{i=1}^{N} S_i^j C_k^i = \sum_{i=1}^{N} (\sum_{j=1}^{N} d_j C_j^i) C_k^i
\]

After \( N \) decoding cycles, \( i = N \) and the decoder equation can be evaluated by applying the distributive and associative properties of the addition operation:

\[
X_k^N = \sum_{j=1}^{N} d_j (\sum_{i=1}^{N} C_j^i C_k^i)
\]

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

\[
X_k^N = N d_k
\]

Since the Walsh orthogonal codes are bipolar of \( \pm 1 \) values, 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 \( X_k^N \) by shifting the result \( \log_2 N \) bits to the right simply by rewiring.

III. ACDMA CROSSBAR ARCHITECTURE AND COMPLEXITY ANALYSIS

In a CDMA NoC router, each PE is connected to two Network Interfaces (NIs), a transmit and a receive NI module. During packet transmission from a PE, the packet is divided into flits to be stored in the transmit NI First-In-First-Output (FIFO) buffer. The router arbiter, then, selects at most \( N \) winning flits from the top of the NI FIFO buffers to be transmitted during the current transaction. The selected flits must all have an exclusive destination address to prevent conflicts, a winner from two conflicting flits is selected according to the router’s priority scheme. In this work, the employed priority scheme is fixed winner takes all priority scheme, only one of the transmitters is given a spreading code and is acknowledged to start encoding. Once done, the router assigns CDMA codes to each transmit and receive NI. NIs with empty FIFO buffers or conflicting destinations are assigned all-zero codes such that they do not contribute to the CDMA channel sum. Afterward, flits from each NI are spread by the CDMA codes in the encoder module.

The ACDMA crossbar implements the physical layer of the NoC router by interconnecting \( N \times N \) transmit-receive ports of width \( W \) each. The ACDMA communication model is to transfer \( W \) bits between each transmit-receive pair every crossbar transaction where the \( W \) bits from each transmit port are handled collectively rather than the bit-wise approach of the classical CDMA crossbar. Figure 2 illustrates the high-level architecture of an ACDMA crossbar. A data word from a transmit port is spread into \( N \) chips, where \( N \) is the CDMA code size which is the same as the number of clock cycles in a single crossbar transaction. These data spread from all encoders are summed up by the CDMA crossbar adder illustrated in Figure 2(d) and the sum is sent out serially to all decoders. At each decoder, the assigned code is cross-correlated with the received sum to decode the data from the summed chips. The decoded flits are stored in the receive NI FIFO buffers until they are read by the PEs. The encoding is realized using XOR gates depicted in Figure 2(b) while the cross-correlator decoder is realized using the up/down
(add/sub) accumulator circuit shown in Figure 2(c). The process of data encoding and decoding lasts for $N$ clock cycles synchronized via a counter.

Two various ACDMA crossbar architectures are presented in this work, the ACDMA crossbar with serial encoding/decoding and the ACDMA crossbar with parallel encoding/decoding or shortly S-ACDMA and P-ACDMA crossbars, respectively. The communication model of the P-ACDMA is the same as that of the S-ACDMA, however, spread data is transmitted in a single cycle using $N$ channel replicas, a channel for each code chip, instead of spending $N$ clock cycles to transmit the $N$ chips. Therefore, the discussion herein applies for both the S-ACDMA and P-ACDMA unless a distinction is stated. The P-ACDMA decoder is shown in Figure 2(c) no correlator is needed, instead a tree adder is used, because all the $N$ chips are available in the same clock cycle. The channel adder shown by Figure 2(d) is replicated $N$ times in the P-ACDMA crossbar.

The ACDMA crossbar depicted in Figure 2 is composed of four main parts; crossbar controller, encoders, channel adder, and decoders. In the following, detailed functionality, implementation details, and complexity analysis of the crossbar components are elaborated:

- **Crossbar Controller**: The controller assigns spreading codes to different encoders at the beginning of each crossbar transaction. The assignment of orthogonal despreading codes to receive ports is fixed, i.e. does not change between crossbar transactions. Therefore, for a router port to initiate the com-
unication with the receive port it addresses, the controller assigns the same code to the encoder in the transmit port and the decoder in the receiver port. This controller employs a Receiver-Based Protocol code assignment scheme; if two different ports request to the same decoder, the controller allows one access and suspends the other. In this work, a static allocation scheme that allocates fixed spreading codes to all encoders is used.

- **Encoder:** As shown in Figure each encoder spreads out data from each transmitting port using W XOR gates. Instead of adding the spreading chips of the Walsh code to the result in the encoder block as suggested by , the addition process is postponed to the channel adder block to merge both adders in a single block. Therefore, the output of each encoder is W-bit width. In the P-ACDMA crossbar, the W-bit encoder is replicated N times.

- **Crossbar Adder:** The encoder outputs are then added together to form the sum of the leaves of the tree adder architecture as depicted by where the height of the tree is log(N). The width of the output wires from each encoder in the tree is equal to the width of the input wires plus one to prevent overflows. Since the input to the first level of adders is (W+1)-bit wide and the height of the adder tree is log2(N), then the width of the output wires at the root adder is W + log2(N) + 1. Pipeline registers are inserted after each stage in the tree to minimize the critical path of the channel. The crossbar adder is replicated N times for the P-ACDMA crossbar.

- **Decoder:** The sum of all decoder, a decoder per receiver port. The decoders implement the cross-correlation in a cost-efficient manner; the decoder consists of only an adder/subtractor and a register configured as an up/down accumulator as shown in Figure. Since the despreading code Ck consists of ±1 chips, cross correlation is reduced to simple addition and subtraction operations of consequent sums S'. Therefore, the decoder is implemented as an up/down accumulator; the adder/subtractor adds or subtracts the sum S' from the result saved in the registers according to the value of despreading chip Ck. In particular, when the despreading chip is ‘1’, the adder adds S' to the contents of the register but subtracts S' from the contents of the register when the despreading chip is ‘-1’. At the end of the decoding cycle, the accumulator register holds Ndk according to . Since N = 2n and n is an integer, data dk is decoded by shifting the accumulator content by log2(N) bits. The P-ACDMA decoder shown in Figure is similar to the serial decoder, but all the channel sums are received in parallel not sequentially, therefore, the accumulator loop is unrolled into a parallel adder.

In Table the S-ACDMA crossbar and the classical bit-wise crossbar complexity is analyzed. The number of two input XOR gates is the same for both circuits. The improvement of the S-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 + log2(N) − i. For a W-bit word, the number of adder wires is increased to W + W(log2(N) − i), and since there are 2i adders at each stage, then the total number of wires is equal to log2 N−1 2i(W + W(log2 N − i)). In the S-ACDMA crossbar, conversely, the number of adder wires for a W-bit word is W + log2(N) − i, which makes the total number of wires equals to log2 N−1 2i log2 N − i which is a factor of W less than that of the conventional CDMA crossbar. The reduced number of carry bits of the S-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 S-ACDMA crossbar complexity is in an order of W less than that of the conventional CDMA crossbar. The complexity of the P-ACDMA crossbar is N times that of the S-ACDMA crossbar due to the encoder and adder replication.

### IV. ACDMA CROSSBAR IMPLEMENTATION RESULTS

In this section, synthesis results of the ACDMA crossbars on a 65 nm standard-cell process are presented. The synthesis results are compared to the conventional WB CDMA crossbar, the high-throughput OCI crossbar presented in , and the low-overhead SB NoC crossbar presented in . However, to neutralize any discrepancy between various crossbar implementations that may arise due to different synthesis strategies and implementation technologies, all results are reproduced for the 65 nm technology. The 65 nm standard-cell is a mature technology that combines improved performance and reduced power consumption with increased design possibilities, cost efficiency, and chip yield compared to the recent nano technologies. Nevertheless, the proposed work can be readily resynthesized in a more recent technology since the HDL code and the design procedures will be very similar with minor changes related to the updated technology. The crossbar circuits presented in , are bit-wise architectures—multi-bit data words are communicated via either resource replication or data serialization. Since the ACDMA crossbar is word-wise, the bit-wise interconnect circuits of , are replicated W times while reproducing

### Table 2. 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 | \(\sum_{i=0}^{\log_2 N-1} 2^i(W + \log_2 N - i)\) | \(\sum_{i=0}^{\log_2 N-1} 2^i(W + W(\log_2 N - i))\) |
| Decoder accumulators | \(N(W + log_2 N)\) | \(N(1 + log_2 N)\) |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
the implementation results to enable comparison with the ACDMA crossbars. Area is estimated using Synopsis Design Compiler, activity factors are estimated by ModelSim to aid the Design Compiler to accurately estimate power dissipation. Implementation and performance results are displayed for various $N$ and $W$ parameters. As a reminder, $N$ is the CDMA spreading code length which equals the number of crossbar I/O ports while $W$ is the port width in bits.

Figure 3 depicts the area and power consumption results of the compared crossbars, the area and power consumption results are normalized to the number of crossbar I/O ports to conduct a fair comparison between different crossbar architectures. As illustrated by Figure 3(a) as $W$ increases for fixed $N$ the area saving of the S-ACDMA over the WB and SB also increases. However, due to resource replication in the P-ACDMA, the P-ACDMA area is higher than that of the SB and WB by at most 38.8%. Additionally, the area per port in $\mu m^2$ of the S-ACDMA crossbar for variable $N$ shown by Figure 3(b) is lower than that of the SB crossbar by at least 19% while the area per port of the P-ACDMA crossbar is higher than that of the SB crossbar by up to 115%. The power per port results depicted in Figures 3(c) and 3(d) follow its counterparts of the area per port for all crossbars since the load capacitance increases with the area. Moreover, as the data width $W$ increases, the power dissipation improves; at $W = 32$, the power dissipation reduction of the S-ACDMA over the WB and SB is 71.3% and 46%, respectively, and the power dissipation reduction of the P-ACDMA over the WB is 33.9% while it is higher than that of the SB by only 22.5%.

Figure 3 also shows the latency, throughput, and throughput per area (TPA) for various $N$ and $W$ parameters. As depicted by Figure 3(e) due to the increase in the ACDMA carry chain length in the channel adder with the increase in the input data width $W$, the ACDMA crossbar latency $L$ in $ns$ increases with $W$ for fixed $N$. Conversely, the carry chain length of the WB, SB, and OCI crossbars is invariant in $W$ since the channel adders are replicated $W$ times instead of increasing the adder input width $W$ as in the ACDMA crossbars. For $N = 8$ and compared to the WB and SB crossbars, the S-ACDMA crossbar latency increases from 2% at $W = 4$ to 63% at $W = 32$ while the P-ACDMA crossbar latency increases from 2% at $W = 4$ to 38% at $W = 32$. On the other hand, for fixed $W$, latency of all crossbars is almost invariant for varying $N$ as shown by Figure 3(f) due to pipelining and fixing the carry chain length; $N$ defines the number of channel adder inputs and consequently the number of tree adder stages while pipelining reduces the crossbar delay to only a single-stage delay. For different code lengths $N$, latency of the ACDMA crossbars shown in Figure 3(g) is up to 25% higher than that of their classical CDMA counterparts due to the increase in channel adder width. The OCI crossbars have the highest latency compared to the classical and ACDMA crossbar counterparts due to doubling the number of crossbar ports defining the number of channel adder inputs.

The increase in the ACDMA crossbar latency causes throughput to degrade as $W$ increases, the throughput reduction of the S-ACDMA crossbar raises from 2% at $W = 4$ to 39% at $W = 32$ compared to the WB and SB crossbars. On the other hand, due to the parallel transmission in the P-ACDMA crossbar, the P-ACDMA throughput increases from 678% at $W = 4$ to 476% at $W = 32$ compared to the WB and SB crossbars. The throughput of the S-ACDMA crossbar shown in Figure 3(h) is at most 20% lower than that of the SB crossbar due to its higher latency while the P-ACDMA provides at least 678% higher throughput due to the parallel transmission. The serial and parallel OCI crossbars outperform the classical CDMA and ACDMA rivals because the OCI main design objective is maximizing throughput.

However, due to the substantial reduction in the total area of the ACDMA crossbars as illustrated by Figure 3(a) and (b), TPA of the ACDMA crossbars defined by (7) is significantly higher than that of the SB, WB, and OCI crossbars.

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

The TPA results of the WB, SB, OCI, and ACDMA crossbars in Mbps/$\mu m^2$ are juxtaposed in Figure 3(i) and the increase in the S-ACDMA crossbar TPA over that of the WB, SB, and T-OCI crossbars is at least 96.3%, 18.2%, and 118.6% respectively, while the TPA enhancement of the P-ACDMA crossbar over that of the WB, SB, and P-OCI crossbars is at least 400%, 255.3%, and 184.2%, respectively. Furthermore, despite the lower throughput of the S-ACDMA crossbar in comparison with the conventional crossbar, the S-ACDMA TPA depicted in Figure 3(j) is at least 20% higher than that of the conventional crossbar due to the large reduction in area while the P-ACDMA provides at least 382% higher TPA despite its larger area due to the improved throughput.

V. ACDMA-BASED NOC

The presented ACDMA crossbars are employed in a full NoC configuration, a 65 node star topology is built using five ACDMA routers, each 15 PEs are connected to a 16-port ACDMA router where $N = 16$, while the sixteenth port of each ACDMA router is connected to an SDMA central router in a star topology. Two ACDMA-based NoC architectures are implemented: the serial S-ACDMA and parallel P-ACDMA architectures. Both S-ACDMA- and P-ACDMA-based NoCs are compared to the same 64-node, 16-bit flit, 8-ary 2-cube torus SDMA-based CONNECT NoC [11]. Each topology is selected to optimize the performance of the associated router architecture [1]. Figure 4 depicts the topology of both ACDMA and CONNECT NoCs. Table 2 summarizes the Star-ACDMA and CONNECT NoC design parameters.

The ACDMA-based router architecture is shown in Figure 5. In the S-ACDMA architecture, if the number of flits to be transmitted by the arbiter is no greater than four, then there is no need to employ $N = 16$ Walsh spreading codes; only $N = 4$ spreading codes are sufficient since they are capable of transmitting $M = N = 4$ flits incurring a four-cycle latency only instead of sixteen. Similarly, if the number of flits...
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3158331, IEEE Access

Mohammed Farag et al.: Aggregated CDMA Crossbar with Hybrid ARQ for NoCs

FIGURE 3. (a, b) Area in $\mu$m$^2$; (c, d) Total power dissipation in mW; (e, f) Latency in ns; (g, h) Throughput in Gbps; and (i, j) TPA in Mbps/$\mu$m$^2$ of the ACDMA, WB, SB, and OCI crossbars for variable $W$ and $N$ parameters.
TABLE 3. Parameters of the star ACDMA and torus CONNECT NoCs [11].

| Topology     | Star-ACDMA | CONNECT |
|--------------|------------|---------|
| Routing Algorithm | Deterministic | Deterministic |
| Router Type   | Simple Input-Queued | Simple Input-Queued |
| Flow Control  | Peck       | Peck    |
| Number of buffer bits | $70 \times 71 = 4970$ | $64 \times 5 \times 16 = 5120$ |

FIGURE 4. CONNECT torus topology versus the ACDMA-based NoC star topology.

FIGURE 5. ACDMA-based router architecture.

to be transmitted is no less than five and no more than eight, $N = 8$ spreading codes are sufficient which incurs a latency of eight clock cycles. Consequently, the S-ACDMA crossbar utilizes spreading codes of variable length to minimize the hop latency according to the traffic conditions. Code lengths $N = 4$ and $N = 8$ are employed when the number of flits to be sent at the next crossbar transaction is less than five, or between five and eight, respectively, and utilizes $N = 16$ code length otherwise. However, only the $N = 16$ Walsh code set shown in Figure 6 is stored in the spreading codes ROM since the $N = 4$ and $N = 8$ Walsh code sets are partial Walsh spreading code sets [12].

The presented NoCs are tested by subjecting the synthesized NoC models to various synthetic traffic patterns for 500 clock cycles each, the latency per packet is then computed by dividing the total number of clock cycles (500) by the total number of packets that arrived successfully to their target PEs in each traffic pattern. Additionally, the throughput $\Theta$ is calculated as follows:

$$\Theta = \frac{N_c \times N_b \times p}{t_c} \quad (8)$$

where $N_c$ is the number of simulation clock cycles (500), $N_b$ is the number of bits per packet (256), $p$ is the number of packets received successfully by their target PEs, and $t_c$ is the clock period.

The performance results of the ACDMA, OCI, and CONNECT NoCs are illustrated in Figure 7 for six synthetic traffic patterns and for the same packet width of 256 bits. The uniform, hotspot, and tornado traffic patterns are employed with two variants: local and global traffic. In the global traffic, the traffic pattern is applied to the entire network, while in the local traffic the traffic pattern is applied to separate clusters. For the ACDMA network, there are five clusters corresponding to the five ACDMA routers. On the other hand, the 64 nodes of the torus network are divided in the network layer into five clusters according to the proximity of the routers.

As illustrated by Figure 7(a), the latency in cycles per packet of the S-ACDMA NoC is higher than that of the CONNECT NoC in most traffic patterns due to the serial spreading of packets. However, the S-ACDMA latency is lower in the hotspot traffic pattern due to the smaller number of hops needed to reach the hotspot node. Additionally, the P-ACDMA NoC offers lower packet latency than the CONNECT NoC for all traffic patterns except for the uniform pattern since torus NoCs are better at balancing the injected load than star NoCs. Consequently, the P-ACDMA NoC throughput shown in Figure 7(b) is higher than that of the CONNECT NoC for all traffic patterns. Moreover, the improvement in throughput and area of the S-ACDMA and P-ACDMA NoCs over that of the CONNECT NoC saliently appears in the TPA comparison in Figure 7(c). However, as illustrated by Figure 7(d), the dynamic (hashed bars) and static (solid bars) power consumption of the ACDMA NoCs for all traffic patterns are larger than that of the CONNECT NoC.
NoC due to the large fanout of the crossbar adder. Therefore, the improvement in the TPA of the S-ACDMA and P-ACDMA NoCs comes at the expense of the increased power consumption. Resource replication and lowering the clock speed can be employed to reduce the power consumption at the expense of reducing TPA.

In Table 4, the ACDMA NoC router implementation results including the area and maximum clock cycle are compared to the CDMA NoCs presented in [1], [11], [13]. The comparison is limited to full NoC routers implemented in the ASIC technology. The compared NoCs are: S-ACDMA, P-ACDMA, T-OCI, P-OCI, CONNECT, and NPC-based NoCs [6]. All compared NoC routers feature 64 or 65 I/O ports which are implemented in the 65 nm standard cell technology except the NPC-NoC which is implemented in a 40 nm technology. The S-ACDMA NoC router has the lowest area among all routers even the NPC-NoC implemented in 40 nm. The area of the S-ACDMA NoC is 66.4% less than that of the CONNECT NoC while the area overhead of the P-ACDMA NoC is 31.6% higher than that of the CONNECT NoC. The P-ACDMA router has the largest area but as shown by Figure 7 it has higher throughput and lower latency compared to other NoC routers. In conclusion, the S-ACDMA and P-ACDMA NoC routers provide SoC designers with various area-speed trade-offs making them suitable for a wide range of applications.

VI. ACDMA COMMUNICATION RELIABILITY

Wireless channels are purely analog exposing them to all random effects such as noise, fading, and interference. On the other hand, the ACDMA crossbars adopt parallel binary signaling to carry the crossbar sum instead of multilevel or analog signaling which enhances their robustness. According to [14], while full-swing digital implementations have typically been able to assume BER values less than $10^{-15}$ over the operating range of voltages and frequencies, this assumption does not hold true for custom low-swing interconnects and modern deep sub-micron circuits.

Because the ACDMA scheme relies on aggregating multiple binary channels in a single M-ary channel, the robustness of the ACDMA crossbar against noise may be raised as a series concern since random errors may affect the entire M-ary symbol carrying the sum of all transmit words. Compared to the classical bit-wise CDMA, the noise effect in ACDMA channels is aggravated by accumulating binary bits in symbols which is analogous to the M-ary wireless communications due to the distance reduction between the constellation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
ACDMA Arbitration output ports Data from N Ports W-bit Port 1 Port 2 Port N Crossbar Hamming Generator Hamming Decoder Hamming Decoder Hamming Decoder CRC Mismatch Wrong Parity bit ARQ

ACDMA Crossbar

FIGURE 8. ACDMA crossbar with Hamming codes, CRC, and ARQ.

points. In other words, errors that occur in classical bit-wise CDMA only affect the channel sum corresponding to a single bit from each transmit port while in ACDMA errors affect the channel sum corresponding to a multi-bit word sent from each port.

The digital nature of the ACDMA interconnect enables enhancing its robustness by employing error detection and correction techniques to mitigate such random effects. To enhance the ACDMA crossbar reliability, a hybrid error detection and correction with Automatic Repeat Request (ARQ) block is implemented in the physical layer of the ACDMA routers as shown in Figure 8. The WCDMA CRC-8 [15] is first computed for each packet and from each NI and appended to it, then the packet is split into multiple parts then encoded using Hamming code with an additional parity bit to detect two-bit errors and correct a single bit error. The Hamming-encoded packet is encoded in ACDMA and transmitted to the destination NIs. After the encoded packet arrives at the destination decoder, the received packet is Hamming decoded. In case no two bit errors were detected, the CRC-8 of the decoded packet is computed and compared to the CRC-8 field of the decoded packet. If the CRC-8 fields do not match, the destination NI signals an ARQ to the arbiter and drops the packet, the arbiter then requests the transmit NI to resend the packet. Table 5 depicts the overhead percentages to the original ACDMA crossbar area and clock period of Table 4, both time and area overheads are a result of the extra encoding and decoding steps and the additional redundant and CRC bits which increases the ACDMA channel width.

To analytically study the ACDMA crossbar reliability in the presence of error sources such as noise, the BER of the CDMA, ACDMA, and overloaded CDMA links presented in [1] subject to Additive White Gaussian Noise (AWGN) with a variable Signal to Noise Ratio (SNR) was computed using Matlab simulation for the following test scenario: The CDMA sum $S$ can be expressed as a bit vector $S = [s_1 \ s_2 \ \ldots \ s_N]$. An AWGN vector of size $N$ is added to the sum $S$ to generate the corrupted sum $\tilde{S}$ such that $\tilde{S} = S + [N_1 \ N_2 \ \ldots \ N_N]$ where each $N_i$ is the AWGN with zero mean and variance $\sigma^2 = P_s/SNR$ where $P_s$ is the signal power. A total of $10^7$ test vectors are randomly generated per SNR value which changes from 0 to 20. The BER of the ACDMA and classical CDMA encoded with [7,4] and [15, 11] Hamming codes is demonstrated in Figure 9, the number of the packet parts to be Hamming-encoded (P) is varied between one and two. To compare the ACDMA crossbars to the overloaded CDMA (OCDMA) crossbar, the BER of the ODMA and classical CDMAs crossbars with no ARQ presented in [1] is also illustrated in Figure 9.

The BER-SNR curves depicts that the classical CDMA crossbar with ARQ outperforms the ACDMA and OCDMA crossbars in a digital communication channel subject to AWGN. This is expected due to the bit-wise nature of the CDMA crossbar in which the error only invalidates a single bit rather than a whole packet as in the ACDMA crossbars. Compared to the classical CDMA and overloaded CDMA crossbars with no ARQ, CDMA crossbars with ARQ have lower BER which motivates using the ARQ mechanism to enhance the communication quality in NoCs and justifies the overhead added by the ARQ circuitry. Varying the number of the packet parts to be Hamming-encoded P from 1 to 2 slightly enhances the BER performance of the classical CDMA crossbars with ARQ but does not enhance the ACDMA crossbar BER performance. To enhance the BER of the ACDMA crossbars, the Hamming code redundancy has been increased from [15,11] to [7, 4]. However, as shown by the BER curves, increasing the Hamming code redundancy does not result in a significant impact on the BER (BER curves are almost identical). On the other hand, using [15,11] Hamming codes adds 36% redundancy bits while using [7,4] Hamming codes adds 75% redundancy bits which suggests using [15,11] Hamming codes for the ARQ mechanism.

TABLE 5. Overhead Percentage of the Error Detection and Correction with ARQ Circuitry for $N = 16$ NoC

|               | S-ACDMA | P-ACDMA |
|---------------|---------|---------|
| Area (mm²)    | 1.22 (82%) | 4.76 (81%) |
| Clock period (ns) | 3.07 (45%) | 2.51 (156%) |
The number of packets dropped due to bit errors is measured by counting the number of times the ARQ crossbar signal is asserted. This metric is measurable only for routers with the ARQ mechanism. The percentage of packets dropped due to bit errors with respect to packets injected in ACDMA NoCs is illustrated in Figure 10. The S-ACDMA has a slightly better packet dropping percentage compared to the P-ACDMA. For both routers, the packet dropping action nearly vanishes at an SNR of 7 dB or higher.

VII. RELATED WORK

Recently, many NoC architectures have been inspired by our previous works of the overloaded and aggregated CDMA crossbars presented in [1], [9], [10]. An experimental study on the effect of noise on CDMA crossbars is presented in [19]. An 8-node WB-CDMA crossbar was developed and synthesized in a Xilinx FPGA. The BER performance of the crossbar is studied under AWGN channel conditions using the mixed-signal analysis tool provided by the Xilinx system generator tool. The simulation reveals that the 8-bit data can be transferred to each node without the loss of information. From the simulation results, the WCDMA NoC at each node achieves a constant data transfer latency of 20 cycles, irrespective of the communication media in the code domain. Mixed-signal analysis reveals a 20 dB SNR is sufficient for the simultaneous recovery of data in the receiving nodes under AWGN noise environment. Results provided in this work support the claims of our previous and current works.

Enhanced overloaded CDMA for NoCs is presented in [17]. In this work, the base architecture of the crossbar is the T-OCI architecture and the main difference is extending the set of the spreading codes to increase the code overloading ratio and consequently increases the channel utilization. The Walsh code and its inverse form a set of $2^N$ spreading codes, where $N$ is the spreading code length. The data encoded by the inverted Walsh codes are shifted one bit to the left and then added to the data encoded by the Walsh code. For the TDMA coding section, structural changes have been made in the transmitter and receiver to enable using the two least significant bits instead of one bit as in the T-OCI crossbar. No mathematical evidence was presented in this work to show how the data spread by the proposed method is decoded from the channel sum, only RTL simulation of a single specific example is provided to support the paper claims. At the implementation level, the crossbar is implemented in a Xilinx FPGA, and the area, power, maximum clock frequency, and maximum throughput synthesis results are presented for several values of $N$. The results show an increase in the crossbar throughput compared to the classical CDMA and T-OCI crossbars at the expense of increasing the crossbar area to support the newly added codes. Unfortunately, the ASIC implementation results and NoC router performance analysis under different traffic patterns are not presented to compare this work to the S-ACDMA and P-ACDMA crossbars proposed herein.

The parallel overloaded CDMA (OCDMA) crossbar is presented in [18]. In this work, the overloaded CDMA crossbar is modified with a parallel encoding/decoding circuit using Gold codes for increasing the channel utilization and improving the crossbar throughput. The crossbar presented in this work is identical to the P-OCI crossbar presented in [1] with the main difference is the used spreading code family. Gold codes are not orthogonal but have low cross-correlation at an arbitrary delay. Gold codes are generated by XORing two maximal length sequences (m-sequence) generated by two linear feedback shift registers (LFSRs) of the same length. A set of Gold code sequences consists of $2^n + 1$ sequences each of length $2^n - 1$, where $n$ is the number of registers in the LFSRs. A parallel encoder/decoder is used to transfer the channel sum in a single clock cycle similar to the parallel channel used in the P-OCI and P-ACDMA crossbars. The parallel OCDMA crossbar is implemented in a Xilinx FPGA; the area, power, maximum clock frequency, and max-
The proposed crossbar is implemented in a Xilinx FPGA and increases the area and critical path delay of the CDMA channel which adds an overhead to the crossbar presented in [16]. In this work, a Parallel Compare and Compress (PaCC) encoder module observes the channel sum for consecutive zeros or ones, and compresses the sum to reduce the amount of data transferred through the CDMA channel. An encoder/decoder circuitry is added to the T-OCI crossbar for compressing/decompressing the channel sum which adds an overhead to the crossbar and increases the area and critical path delay of the CDMA crossbar. The proposed crossbar is implemented in a Xilinx FPGA for \( N = 8 \) and the results show improvement in the area and power consumption. However, the presented results are not sufficient to support the paper claims. This work neither presents thorough analysis of the crossbar for different parameters and traffic patterns nor provides detailed implementation and performance results to compare with the base T-OCI crossbar.

On the other hand, there are some recent works which can be investigated for enhancing the ACDMA crossbar and NoC router architectures presented in this work. Improving the fault-tolerance capability of the CDMA-based bus using WB spreading codes is studied in [22] through involving a bus encoding scheme adapted to specific properties of CDMA-encoded data streams. The proposed technique relies on the information redundancy inherently involved in CDMA transmission and does not add extra wires to the crossbar interconnect. A line encoding scheme is adopted in which the weighted binary code is replaced with a code in which the error magnitude of any error pattern will be smaller than \( N \). At the hardware level, a line encoder block and \( N \) line decoder blocks are inserted. The Line encoder converts the sum value at the output of the CDMA channel adder into a line codeword which is distributed to the CDMA decoders at the receive ports. At each destination node, the line decoder converts the line codeword back into the original sum-chip value for further processing by the CDMA decoder. The proposed scheme tolerates detection and correction of a single-bit error in the CDMA packet without adding extra wires to the bus. However, it incurs an overhead of the encoder/decoder blocks which will increase the area and critical path delay of the CDMA crossbar. Simulation results show that this fault-tolerance scheme improves post-decoding BER performance of the WB-CDMA bus. The proposed fault-tolerance scheme can be investigated for the aggregated CDMA crossbars which use the WB-CDMA spreading codes to improve their BER performance in AWGN channels.

A distributed arbitration scheme based on a token-ring algorithm is proposed for CDMA-based NoCs with dynamic code assignment in [20], [21] to solve the complexity and scalability issues associated with commonly used centralized arbitration schemes. The arbitration unit is organized as a ring of relatively simple and functionally identical arbitration elements, which cooperatively resolve destination conflicts and assign codewords to PEs. Distributed arbitration enables scaling the ring arbiter can to large systems without a significant performance loss. The synthesis results show that the overheads introduced by the arbitration unit do not significantly influence the throughput and latency of the packet-oriented CDMA bus. The analysis conducted in this work shows the advantages of distributed arbitration over centralized arbitration in terms of performance, scalability, and flexibility. The distributed arbitration mechanism can be investigated for the proposed aggregated and overloaded CDMA NoC routers to enhance their performance.

VIII. CONCLUSIONS

In this work, the Aggregated CDMA crossbar was presented. The ACDMA crossbar leverages the enhanced channel conditions of digital wires relative to wireless channels in order to implement on-chip M-ary data transmission. The aggregation of multiple data bits into M-ary symbols reduces the area and power dissipation overheads of the ACDMA crossbar compared to the conventional CDMA crossbar. Additionally, a star-connected ACDMA-based NoC is presented and compared to the ubiquitous torus SDMA-based NoC under multiple synthetic workloads. The comparison indicates that serial-encoding ACDMA-based NoCs exhibit better area and power consumption profiles while parallel-encoding ACDMA-based NoCs provide higher throughput per area than SDMA-based NoCs. Moreover, the ACDMA-based NoC is augmented with error detection and correction circuitry using CRC-8 and (15,11) Hamming codes in addition to an automatic repeat request circuitry in order to enhance the ACDMA crossbar reliability in the presence of error sources. The results call for adopting more simple techniques like the M-ary encoding for CDMA-based crossbars. Such techniques are capable of pushing the CDMA-based crossbar performance and efficiency beyond what is provided by current implementations. Consequently, future direction inspired by this research includes experiencing more encoding techniques for CDMA crossbars. In addition, optimization techniques for the parallel-encoding based CDMA crossbars can be explored since the parallel transmission provides higher throughput than serial transmission but at the expense of large area and power consumption.

IX. ACKNOWLEDGMENT

Mohammed M. Farag thanks the Deanship of Scientific Research, King Faisal University, Saudi Arabia for financially supporting this research under the Grant number NA000224.

REFERENCES

[1] K. E. Ahmed, M. R. Rizk, and M. M. Farag. Overloaded CDMA crossbar for network-on-chip. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(6):1842–1855, June 2017.
Network-on-Chip”. In our previous paper, we briefly introduced the Aggregated CDMA crossbar. Our novel contributions in this manuscript are:

- Comparing with the recent OCI crossbar in Section IV
- Employing the Aggregated CDMA crossbar in a star-connected NoC and comparing it to a CONNECT torus NoC in Section VI
- Adding reliability analysis and error detection and correction in Section VI

X. DIFFERENCES FROM PREVIOUS WORK

In this manuscript, we build on our previous work published in the 28th “International Conference on Microelectronics (ICM)” in 2016 entitled “Aggregated CDMA crossbar for...