

# Distributed arbitration scheme for on-chip CDMA bus with dynamic codeword assignment

Tatjana R. Nikolic  | Goran S. Nikolic | Goran Lj. Djordjevic

Faculty of Electronic Engineering,  
University of Nis, Nis, Serbia

## Correspondence

Tatjana R. Nikolic, Faculty of Electronic  
Engineering, University of Nis, Nis, Serbia.  
Email: tatjana.nikolic@elfak.ni.ac.rs

## Funding Information

This work was supported by the Serbian  
Ministry of Education, Science and  
Technological Development, Project No.  
TR-32009 – "Low-Power Reconfigurable  
Fault-Tolerant Platforms."

Several code-division multiple access (CDMA)-based interconnect schemes have been recently proposed as alternatives to the conventional time-division multiplexing bus in multicore systems-on-chip. CDMA systems with a dynamic assignment of spreading codewords are particularly attractive because of their potential for higher bandwidth efficiency compared with the systems in which the codewords are statically assigned to processing elements. In this paper, we propose a novel distributed arbitration scheme for dynamic CDMA-bus-based systems, which solves the complexity and scalability issues associated with commonly used centralized arbitration schemes. The proposed arbitration unit is decomposed into multiple simple arbitration elements, which are connected in a ring. The arbitration ring implements a token-passing algorithm, which both resolves destination conflicts and assigns the codewords to processing elements. Simulation results show that the throughput reduction in an optimally configured dynamic CDMA bus due to arbitration-related overheads does not exceed 5%.

## KEYWORDS

CDMA bus, distributed arbitration, dynamic spreading codeword assignment, Multiprocessor SoC

## 1 | INTRODUCTION

Computation and communication are the two most critical aspects affecting the overall system performance of complex integrated circuits. Technology scaling has enabled the integration of a large number of processing elements (PEs), in the form of intellectual property cores, within complex systems-on-chip (SoCs), thus increasing the computation performance of integrated circuits considerably. However, dealing with data communication among the ever-increasing number of PEs in an SoC environment requires faster and more efficient on-chip interconnection architectures [1–4]. Most interconnect networks in embedded SoCs rely on a traditional shared bus employing time-division multiplexing

(TDM) communication protocols to reuse expensive on-chip interconnect resources. A TDM bus is usually accompanied by scheduling and arbitration overheads, which increases the transmission latency [5]. Furthermore, the lack of parallel transmissions makes the scaling of a TDM-bus-based on-chip interconnect difficult, because each added PE decreases the bandwidth available to the other PEs. Network-on-chip (NoC) has been proposed to address the scalability, latency, throughput, and reliability issues of TDM on-chip communication [6]. The PEs in an NoC system communicate data packets through routers and communication links arranged in a specific network topology. The locality of interconnections provides a high level of parallelism, because all the links and routers in the NoC can operate simultaneously on different

data packets [7]. However, the data transfer latency depends on the relative positions of the source and destination PEs in the network, which may make it difficult to fulfil timing constraints in the real-time SoCs. In addition, NoC may incur high power consumption and area overheads and require complex logic [8].

On-chip interconnection based on the code-division multiple access (CDMA) technique has been recently proposed to eliminate the complexity incurred by routing issues [9–11]. Most of these proposals incorporate spread-spectrum technology to encode multiple data streams in parallel onto the same physical interconnect, thus delivering a throughput speedup over a conventional split-transaction bus. In this approach, the data from different transmitting PEs are first independently encoded by using mutually orthogonal spreading codewords and then combined together into a composite data stream. The composite data stream is then transmitted over a shared bus to all receiving PEs, concurrently. Each receiving PE extracts its associated data stream by using the same codeword as the corresponding transmitting PE. This centralized data transfer scheme provides constant data transfer latency between any pair of PEs. In comparison with conventional on-chip communication schemes, the CDMA bus is more suitable for supporting multiple-access communications and it is capable of achieving higher throughput under heavy traffic load conditions. In addition to its use as a multiprocessor bus, the CDMA technique is used as an alternative to a crossbar switch in NoC routers, and a flexible peripheral switch that simplifies interconnection between the CPU core and numerous input/output ports [1,5,12,13].

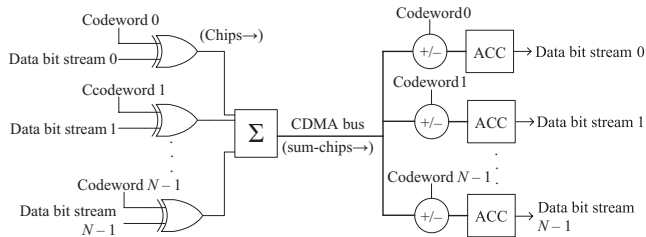
The main design choice in the on-chip CDMA-based interconnect architecture concerns the codeword assignment method. In *static* CDMA systems, each PE is permanently assigned a distinct codeword that it uses to encode its outgoing data [12,14]. With such an assignment scheme, the number of available codewords must match the number of PEs in the system, which is not desirable, especially in large systems. In contrast, in *dynamic* CDMA systems, codewords are assigned to PEs on demand, thus allowing the use of a limited number of codewords in systems with a large number of PEs [15]. The dynamic codeword assignment improves system scalability while providing lower latency compared with the static assignment scheme. However, as the maximum number of concurrent transmissions is determined by the number of available codewords, the dynamic CDMA system may suffer a performance loss under heavy traffic load condition.

Adaptive algorithms for codeword assignment were proposed in [16] and [17] for an efficient usage of spreading codewords. The overloaded CDMA interconnect (OCI) was introduced in [14] to enhance the capacity of CDMA NoC crossbar by increasing the number of usable spreading codewords. The evaluation results demonstrate the superiority of the OCI-based NoCs in terms of area and throughput. A

CDMA protocol for the dynamic assignment of spreading codewords was presented in [15]. It is demonstrated that the parallel CDMA network has a higher throughput and lower latency than the mesh-based NoC and the TDM bus due to the simultaneous medium access nature of CDMA.

Although appealing in many respects, the CDMA concept has not yet found a wider acceptance in the domain of SoC design, primarily because of the difficulties in realizing complex control and arbitration mechanisms in hardware. The on-chip CDMA bus should include an arbiter to resolve conflicts when multiple PEs issue transmission requests to the same destination PE [18–20]. In addition, the arbiter should ensure correct synchronization and provide the distribution of codewords to PEs. The CDMA arbiter design is particularly complex in dynamic CDMA systems, where it should additionally deal with the run-time assignment of spreading codewords [15]. The existing CDMA bus proposals employ centralized arbitration schemes with multiple request and grant lines between different PEs and the arbiter. The PEs send their requests independently and the centralized arbiter decides which of them should be allowed to transmit and which codeword will be used for data encoding/decoding. The centralized arbitration makes CDMA-bus-based systems difficult to scale. In systems with a large number of PEs, the central arbiter can introduce a significant delay and become a performance bottleneck. Another disadvantage is the existence of numerous long arbitration lines. These problems can partially be solved by separating the arbitration phase from the data transmission phase. In this case, while the data bus is used for data transmissions, the next transfer is already being arbitrated by the central arbiter simultaneously. In systems having long data transfers, this scheme could provide satisfactory performance [21]. Nevertheless, the high complexity of the centralized arbiter is a serious obstacle to both the performance and scalability of CDMA-bus-based systems, especially in dynamic CDMA systems, which employ complex codeword assignment algorithms.

In this paper, we present a novel distributed arbitration scheme for an on-chip CDMA bus, which supports both dynamic and static codeword assignments. The proposed arbitration unit is composed of a set of simple and functionally identical local arbiters, each associated with a PE. The local arbiters are connected in a ring, and they cooperatively implement a two-phase token-passing algorithm. In both phases of the arbitration procedure, the interaction between the PEs is enabled by passing tokens over the arbitration ring. In the first phase, destination conflicts are resolved, ensuring that no more than one source PE sends data to the same destination PE at any time. In the second phase, distinct codewords are assigned to the selected source-destination pairs of PEs. The dynamic codeword assignment algorithm is based on the concept of temporal codeword ownership, according to which only PEs that own the codeword are



**FIGURE 1** General principle of CDMA data transfer

allowed to transmit their data over the CDMA bus. When idle, the codeword owner hands over its codeword ownership to any PE that is ready to transmit but does not own a codeword. The proposed distributed arbiter offers several advantages over the conventional centralized arbitration schemes. First, the distributed and modular arbiter design improves the scalability of the CDMA bus because adding new PEs to the system does not require any modification of relatively simple local arbiters. Second, due to local and short connections between local arbiters, the arbiter can support high-frequency bus operation without any dependency on the system size. Finally, the token-ring approach allows the implementation of complex arbitration algorithms in hardware, which is of crucial importance for an efficient design of the dynamic on-chip CDMA bus. However, due to its multi-cycle operation, the ring arbiter introduces time overheads, which can reduce the overall data throughput of the CDMA bus. However, as demonstrated by the simulation results, the throughput reduction due to arbitration delay is significant only when the number of available codewords is much smaller than the number of PEs. In an optimally configured dynamic CDMA-bus-based system, the reduction in data throughput is less than 5%.

The rest of the paper is organized as follows. In Section 2, we review the basics of on-chip CDMA-based communications, including general organization and two main architectural variants of these systems. Section 3 introduces the proposed token-ring arbitration scheme for a dynamic CDMA-bus-based multiprocessor system. We first present the architecture of the arbitration ring and then describe the distributed arbitration algorithm in detail. Section 4 presents an evaluation of the proposed arbitration scheme in terms of bus throughput (BT) and latency. The concluding remarks are presented in Section 5.

## 2 | CDMA-BUS-BASED ON-CHIP INTERCONNECT INFRASTRUCTURE

The basic principle of an  $N$ -channel CDMA-based interconnect, which permits the simultaneous transmission of up to  $N$  data bit streams from a set of  $N$  transmitters to a set of  $N$

receivers, is illustrated in Figure 1. At the transmitter end, data bit streams are first encoded by using a distinct spreading codeword for each stream. Then, the encoded streams are summed up into a composite CDMA stream, which is delivered to all the receivers. At the receiver end, each receiver can retrieve any data bit stream by correlating the composite CDMA stream with the corresponding codeword.

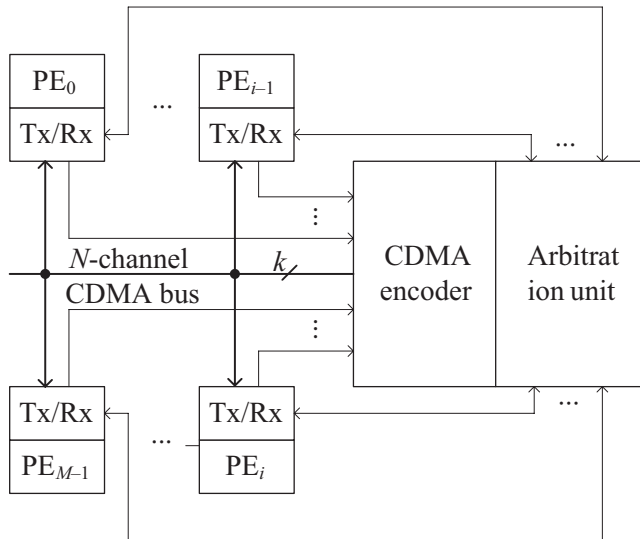
For a perfect separation of data streams, codewords must be mutually orthogonal, that is, each codeword should have a zero correlation with the other codewords and a high correlation with itself [22]. For the encoding/decoding of data streams, CDMA-bus-based systems commonly employ the Walsh code, which satisfies the orthogonality property and is relatively simple to realize. With this code, the codewords are rows (or columns) of Walsh-Hadamard (WH) matrices [23,24]. These matrices are constructed recursively by starting from a single bit, according to the following matrix construction rule:

$$\mathbf{WH}_{2^n} = [w_{i,j}]_{2^n \times 2^n} = \begin{pmatrix} \mathbf{WH}_{2^{n-1}} & \mathbf{WH}_{2^{n-1}} \\ \mathbf{WH}_{2^{n-1}} & \mathbf{WH}_{2^{n-1}} \end{pmatrix}, \quad (1)$$

where  $\mathbf{WH}_1 = [1]$ ,  $n = 1, 2, \dots$ . Therefore,  $N$  distinct codewords, which are required to implement an  $N$ -channel CDMA system, can be obtained from  $\mathbf{WH}_{2^n}$ , where  $n = \lceil \log_2 N \rceil$ . Note that the codewords are of length  $N$ .

To encode a data bit stream by a Walsh codeword, each data bit needs to be XORed with every bit of the codeword, sequentially. Thus, every data bit is spread out into a sequence of  $N$  single-bit values, the so-called chips. The chip values obtained from all the active transmitters in the current cycle are summed arithmetically by the adder unit. The numerical value obtained by chip summation is called the *sum-chip* and can be represented by  $k = (1 + \log_2 N)$  bits [25]. The *CDMA packet* is a sequence of  $N$  sum-chips produced through encoding  $N$  input data bits, one from each transmitter. The CDMA packets are transported to receivers in a sum-chip-serial form over a *binary CDMA bus*, which consists of  $k$  two-level signal wires and allows the transfer of one sum-chip per bus cycle. To decode the desired data bit, the receiver needs to add up all the sum-chips of the incoming CDMA packet by using the bits of the corresponding codeword as signs for the sum-chips. The value of the decoded data bit is decided by the sign of the total summation value of all  $N$  sum-chips. Thus, the capacity of the  $N$ -channel CDMA bus is  $N$  data bits per bus cycle, and it is equally shared among the  $N$  data bit streams.

Figure 2 shows a typical organization of the CDMA-based multiprocessor system. The system is composed of  $M$  PEs, which use an  $N$ -channel CDMA bus for exchanging data streams. In general, the *system size*,  $M$ , and the *CDMA bus size*,  $N$ , can differ, as long as  $N \leq M$ . In this organization, the CDMA encoder, which encodes the bits of all the



**FIGURE 2** Organization of CDMA-bus-based multiprocessor system

data streams and generates the composite CDMA stream, is centralized, whereas the CDMA decoders are distributed and attached to the PEs. Each active transmitting PE sends the bits of the data stream to the CDMA encoder over a separate data line, whereas the composite CDMA stream is delivered back to all the PEs over a common binary CDMA bus and then independently decoded by the local CDMA decoders of the active receiving PEs.

The control part of the system is represented by the “Arbitration unit” block (AU), which manages the requests from the PEs for establishing/terminating the data streams. The AU performs this function by sampling the requests from the PEs and using an appropriate arbitration algorithm to decide which PE will be the next to gain access to the bus, and which codeword will be used for data encoding. The following two types of resource conflicts should be resolved before a new data stream is established: the destination conflicts and the codeword conflicts. The destination conflicts are a consequence of the constraint that each PE can receive at most one data stream at any given time. When multiple PEs simultaneously request data stream connections targeting the same receiving PE, the AU should allow transmission to one of the requesting PEs and place the rest in the waiting state. The codeword conflicts are a result of the requirement that all concurrent data streams must be encoded by different codewords.

In  $N$ -channel CDMA systems with  $M=N$  PEs, the codeword conflicts can be resolved statically, by fixedly assigning a unique codeword to each PE for reception. In such so-called *static CDMA systems*, the AU need not be involved in resolving codeword conflicts. The CDMA decoders are configured with mutually different codewords during system initialization, whereas the CDMA encoder is automatically configured with the codeword of the receiving PE each time a new

data stream is established. However, the static CDMA systems suffer from a low utilization of the available CDMA-bus capacity. As a CDMA bus offers a constant bandwidth of  $1/N$  bits per cycle to each active data stream, the communication capacity of the static  $N$ -channel CDMA system is fully utilized only when the number of active data transmitters equals  $N$ . This occurs during time periods when each PE is involved in communication either as a data stream transmitter or receiver. When there are  $n < N$  active data streams, the unused fraction of the communication capacity of  $(N-n)/N$  bits per cycle cannot be used to speed up the transmission of active streams. Another disadvantage of the static CDMA scheme is the lack of scalability, because new PEs cannot be added to the system without increasing the CDMA bus capacity, that is, the code length. Increasing the code length leads to an increase in the area overheads and data transfer latency.

In contrast, in the so-called *dynamic CDMA systems*, codewords are assigned to data streams. Each time a new data stream is established, both the CDMA encoder of the transmitting PE and the CDMA decoder of the receiving PE are configured with one of the currently available codewords. After the data stream is terminated, the allocated codeword is made available for use by other transmitter-receiver pairs. The main advantage of the dynamic scheme is that the codeword length does not impose any limit on the number of PEs that can be connected to the CDMA bus. In contrast to the static case, the bandwidth of an  $N$ -channel CDMA bus is not shared equally and statically among  $N$  PEs, but is allocated dynamically, on demand, among  $M > N$  PEs. As long as the number of active data streams is equal to or less than  $N$ , the dynamic CDMA system offers the same level of communication performances as the static one. However, when the CDMA bus is fully utilized, new connection requests should be put on hold until a codeword is freed up. The scalability is achieved at the expense of the increased complexity of the AU, which now must resolve not only destination conflicts but also codeword conflicts.

### 3 | TOKEN-RING ARBITER FOR DYNAMIC CDMA BUS

In this section, we present a new design of the CDMA bus arbitration unit based on a token-ring scheme. The proposed ring-arbiter is suited for dynamic CDMA-bus-based systems and features distributed architecture.

#### 3.1 | Arbitration in dynamic CDMA-bus system

The arbitration in dynamic CDMA-bus-based systems is primarily complicated by the need to assign a distinct codeword

to each concurrently active data stream. In general, the arbitration procedure involves the following four phases. In the first phase, the destination conflicts are resolved. In this phase, the arbitration logic should allow the source PE to test and reserve the destination PE for reception, which might include waiting for the destination PE to finish the reception of an ongoing data stream. In the second phase, the arbitration logic reserves one of the available codewords for the selected pair of source-destination PEs. In contrast to the static CDMA-bus system, in which each PE is the exclusive owner of a codeword, each of  $N$  codewords in the dynamic system can only be temporarily owned by one of  $M$  PEs. The PE that owns a codeword can use it for encoding data bits during data stream transmission. As  $M > N$ , not all PEs can simultaneously be codeword owners. Therefore, a PE that intends to transmit a data stream, but does not currently own a codeword, has to request a codeword from other PEs. In contrast, a PE that owns a codeword, but is currently not using it for transmission, has to hand over its codeword ownership to the requesting PE. Note that the connection might be postponed if all the  $N$  codewords are currently in use. After acquiring the codeword ownership, the source PE should notify the destination PE of the codeword that will be used for data stream encoding. In the third phase, the data stream transmission occurs. In the fourth and final phase, the transmitting PE terminates the transmission of the data stream, which includes notifying the receiving PE about the termination of the data stream, and releasing the receiving PE for a new connection.

### 3.2 | Architecture of the arbitration ring

The ring-arbiter for an  $N$ -channel CDMA bus with  $M$  PEs is composed of  $M$  simple ring-arbiter elements, and each element is attached to a PE, as shown in Figure 3. The ring-arbiter element consists of the token register (TR) and the associated control logic. The TRs of all the ring-arbiter elements are connected to form a system-wide ring. At any time,  $M$  tokens,  $T_i, i = 0, \dots, M-1$ , circulate synchronously around the ring in a parallel form. In the proposed ring-arbiter architecture, each PE is the owner of one token. In particular, we assume that  $PE_i$  owns token  $T_i$ . The TRs of all the ring-arbiter elements are driven by a common clock signal. The period of this clock signal is referred to as a *token interval*. The *ring interval* is defined as the time taken by each token to complete a full circle around the ring. It equals  $M$  token intervals. The beginning of the ring interval corresponds to the token interval during which each PE holds its own token. At the next token interval, token  $T_i$  is transferred to  $PE_{(i+1) \bmod M}$ , whereas  $PE_i$  receives the token  $T_{(i-1) \bmod M}$ . In general, at the  $k$ th token interval,  $PE_i$  holds the token  $T_{(i-k) \bmod M}$ . Thus, for token identification, it is sufficient that the ring-arbiter element implements a modulo- $M$

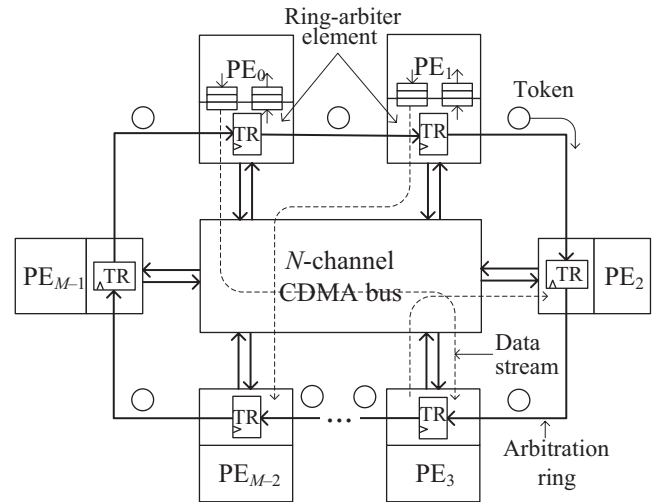


FIGURE 3 Arbitration ring

counter (token counter), which is initialized to the value of  $i$  and is decremented at each token interval. The content of the token counter corresponds to the identifier of the token currently held by the PE. To facilitate interaction between the ring arbiter and the CDMA decoder/encoder logic, we assume that their clock signals are derived from the same oscillator. With this assumption, the token interval lasts the same as the chip interval.

In addition to token and counter registers, the implementation of the dynamic codeword allocation scheme requires the inclusion of a third register in each ring-arbiter element for storing the identifier of the codeword owned by the local PE. This so-called *codeword register* ( $R_{CW}$ ) is also associated with two single-bit flags:  $B$  (*busy*) and  $V$  (*valid*). The  $V$  flag, if set, indicates that the  $R_{CW}$  contains a valid codeword, that is, the local PE is the owner of the codeword whose identifier is stored in  $R_{CW}$ . The  $B$  flag, if set, indicates that the PE is currently using its codeword for encoding the outgoing data bit stream. If  $B$  is reset, and  $V$  is set, then the PE is the owner of the codeword that it does not use, and hence, it can transfer the codeword ownership to another PE if it receives such a request. After handing over its codeword ownership, the PE resets the  $V$  flag. If the PE needs a codeword later, it will have to request it from other PEs. As the number of PEs is equal to or greater than the number of codewords, only  $N$  out of  $M$  PEs are initialized with valid codewords.

In the proposed ring-arbiter architecture, tokens are used as a means for exchanging status/control information among the PEs during the arbitration process. The token itself is a simple data structure composed of several bit-fields. In particular, there are four single bit fields,  $R$ ,  $L$ ,  $C$ , and  $S$ , and two multiple-bit fields,  $CW$  and  $ID$ . The  $R$  (*Reserved*) and  $L$  (*Last*) bits are used for controlling the establishment and termination of the data stream connection, whereas the rest are used during the codeword allocation process. The  $R$  bit

of token  $T_i$ , if set, indicates that the  $PE_i$  is currently reserved for data stream reception. The bit  $L$  of token  $T_i$ , if set, serves as an indication to  $PE_i$  that the source PE has initiated the termination of the data stream. The field  $ID$  of token  $T_i$  holds the identifier of the PE that has initiated the data stream connection with  $PE_i$ . The length of the  $ID$  field is  $\lceil \log_2 M \rceil$  bits. The  $S$  (*Search*) bit is used as the codeword request indicator. The  $CW$  field is used

for exchanging codewords between PEs. This field can hold a codeword identifier, and its length is  $\lceil \log_2 N \rceil$  bits. The  $C$  (*Codeword*) bit, if set, indicates that the  $CW$  field contains a valid codeword identifier.

### 3.3 | Token-ring arbitration algorithm

The control path of the ring-arbiter element is divided into two concurrent processes: the transmitter (TX) and receiver (RX) processes. The RX process is always active, and it has the following two responsibilities. First, it responds to codeword requests, and second, it manages data stream reception. The TX process regulates the data stream transmission. The transmission is started each time the local PE issues a request for transmitting a new data stream. After the data stream is transmitted, the TX process returns to the non-active state. Both processes can access the codeword and token registers. The detailed description of both processes, in pseudocode form, is given in Listing 1 and 2. In the pseudocode, we use the “dot” notation to refer to the individual fields of registers and tokens.

#### 3.3.1 | TX process

Suppose that  $PE_i$  wants to establish a data stream connection with  $PE_j$ . Before issuing the connection request, the  $PE_i$  writes the content of the data stream into the transmit FIFO and passes the identifier of the destination PE and the length of the data stream to the TX process. Once started, the TX process should first reserve the destination PE for reception (lines 1 and 2 of Listing 1). Therefore, it waits for token  $T_j$  and tests the  $R$  bit. Waiting for token  $T_j$  indicates that the TX process continuously checks the token counter until its content becomes equal to  $j$ . If  $PE_j$  is available for reception, then the  $R$  bit of token  $T_j$  will be reset (0). Otherwise, if  $T_j.R = 1$ , the TX process must wait for the full ring-interval to receive  $T_j$  again and repeat the test. Once the TX process receives the token  $T_j$  with the  $R$  bit reset, it sets this bit to 1. Thus,  $PE_i$  informs  $PE_j$  about its connection request, and simultaneously prevents other PEs from initiating connection with  $PE_j$ . At the same token interval, the TX process checks whether it is the codeword owner by testing the  $V$  bit of the codeword register (line 3). If  $PE_i$  owns a codeword, then it copies the content of the codeword register into the  $CW$  field of token  $T_j$  to

inform the  $PE_j$  about the codeword that will be used for data stream encoding (line 4). Both  $S$  and  $C$  bits of this token must be reset for  $PE_j$  to accept the content of the  $CW$  field as the new codeword (line 11). Furthermore, the TX process writes the identifier of the local PE (that is,  $i$ ) into the  $ID$  field of token  $T_j$  (line 12). Then, the TX process prepares data stream transmission by configuring its CDMA encoder with the codeword present in the codeword register. The  $B$  bit is also set to indicate that the codeword is in use henceforth (line 13). However,  $PE_i$  does not start the data transmission immediately, because it has to wait for  $PE_j$  to receive the token  $T_j$ , which occurs at the beginning of the next ring interval (line 15). In contrast, if  $PE_i$  does not own a codeword, then the TX process should issue a codeword request by setting the  $S$  bit in token  $T_j$  (line 6). Then, it repetitively checks the  $C$  bit of token  $T_j$  until the value of this bit is 1, which indicates that the codeword is determined and its identifier is present in the  $CW$  field of the same token. Once  $PE_i$  becomes the codeword owner, the TX process proceeds as already described (lines 11–16).

**LISTING 1** TX process (of  $PE_i$ )

```

1.  wait until ( $T_j.R=0$ )
2.   $T_j.R = 1$ 
3.  if( $R_{CW}.V = 1$ ) then
4.     $T_j.CW = R_{CW}$ 
5.  else
6.     $T_j.S = 1$ 
7.  wait until ( $T_j.C = 1$ )
8.     $R_{CW} = T_j.CW$ ;
9.     $R_{CW}.V = 1$ 
10. end if
11.  $T_j.C = T_j.S = 0$ 
12.  $T_j.ID = i$ 
13.  $R_{CW}.B = 1$ 
14. configure CDMA encoder with codeword  $R_{CW}$ 
15. wait until next ring interval
16. transmit data stream
17. wait for  $T_j$ 
18.  $T_j.L = 1$ 
19. wait for  $T_j$ 
20.  $T_j.R = T_j.L = 0$ 
21. Stop

```

**LISTING 2** RX process (of  $PE_j$ )

```

1.  wait until (token,  $T_k$ , is received)
2.  if( $T_k.S = 1 \ \&\& \ T_k.C = 0 \ \&\& \ R_{CW}.V = 1 \ \&\& \ R_{CW}.B = 0$ ) then
3.     $T_k.CW = R_{CW}$ 
4.     $T_k.C = 1$ 

```

```

5.       $T_k.S = 0$ 
6.       $R_{CW}.V = 0$ 
7.      end if
8.      if( $k = j$ ) then
9.          if( $T_j.R = 1 \ \&\& \ T_j.S = 0 \ \&\& \ T_j.C = 0$ )then
11.         configure CDMA decoder with  $T_j.CW$ ; start reception
13.         else if( $T_j.L = 1$ ) then
14.             stop data stream reception
15.         end if
16.     end if
17.     goto 1

```

After the connection is established, the data bit stream is transferred continuously. The length of the data stream may be arbitrary with the only constraint being that the total number of data bits must be a multiple of  $p$  bytes, where  $p$  is an integer greater than or equal to  $\lceil M/(8N) \rceil$ . This constraint is necessary to ensure correct synchronization between the CDMA bus encoder/decoder logic and the arbitration ring. The termination of the data stream transmission is a two-step procedure. First, the TX process of  $PE_i$  notifies the RX process of  $PE_j$  that the transmission ends by setting the  $L$  bit of token  $T_j$  (lines 17 and 18). Then, in the next ring interval, it releases  $PE_j$  for other connections by resetting the bits  $R$  and  $L$  of the same token (lines 19 and 20). The two bits,  $R$  and  $L$ , are necessary for controlling the connection to avoid joining two data streams addressed to the same destination. When only the  $R$  bit exists, it is possible that  $PE_i$  first resets the  $R$  bit of token  $T_j$ , and then another PE on the path to  $PE_j$  again sets the  $R$  bit and begins the transmission of its data stream. As the change of the  $R$  bit value is not visible to  $PE_j$ , it continues to receive the data stream without noticing that the source PE has been changed. The  $L$  bit resolves this situation.

Note also that the  $R$  and  $L$  bits are reset by the source PE, rather than by the destination PE. This is because of the intention to implement the round-robin arbitration policy. With the  $R$  and  $L$  bits reset by the source PE, the highest priority for establishing a new connection with the same destination PE is given to the immediately next PE after the PE that just terminated the connection. The round-robin policy is a desirable feature, as it provides fairness in arbitration.

### 3.3.2 | RX process

The RX process wakes up at every token interval to examine the token it just received over the arbitration ring (line 1 of Listing 2). If the token carries the codeword request, and the local PE owns a codeword that it does not use, the RX process takes over the codeword and adjusts the flag

bits accordingly (lines 3–6). Additionally, if the local PE is the owner of the received token (line 8), the RX process checks whether the token carries the data stream connection/termination request (lines 9 and 13). In the case of a connection request, the RX process configures the CDMA decoder with the codeword whose identifier is contained in the CW field of the received token and starts data reception immediately (line 11). The RX process obtains the identifier of the transmitting PE from the ID field of the same token. The RX process stops data reception and switches off the CDMA decoder once it finds the  $L$  bit set to 1 in its own token (line 14). Because of the delay between the token interval at which the source PE ends the data stream transmission, and the token interval at which the destination PE receives its own token, the RX process must also discard any incompletely received  $p$ -byte data word.

### 3.3.3 | Arbitration delay

The arbitration delay is defined as the time span between the issue of the connection request by the source PE and the beginning of the data stream transmission. The arbitration delay is not constant but it varies depending on the current traffic conditions, the distance between the source and destination PEs, and the position of the token of the destination PE when the source PE has issued the request. Assuming that the destination PE is available for reception and there are no other connection requests targeting the same PE, and that the source PE owns a codeword, the average arbitration delay is equal to a ring interval, that is,  $M$  token intervals. This is because the source PE must first wait for the token of the destination PE, and then the token must reach the destination PE. If the source PE does not own a codeword, and there is some other PE willing to hand over its codeword, the delay is increased for  $M$  additional token intervals. This is because the token that carries the codeword search request needs a full ring-interval to return to the source PE. The arbitration delay might increase for an integer number of ring intervals in the case of unavailability of the destination PE and/or a temporary absence of codewords.

## 4 | EVALUATION RESULTS

In this section, we will present and discuss the evaluation results on the effectiveness of the proposed token-ring-based arbitration unit for an on-chip CDMA-bus with dynamic codeword assignment. Based on the arbitration algorithm presented in Listings 1 and 2, we developed both a cycle-accurate simulator in SystemC, for assessing the communication performances of the CDMA bus, and a synthesizable RTL

description in VHDL, for estimating the hardware complexity of the ring arbiter. By using VHDL description, we implemented a ring arbiter with 16 ring elements in a Xilinx Artix 7 FPGA device. In this implementation, each ring element comprises arbitration-related logic only. The implementation report shows the usage of only 26 look-up tables (LUTs) and 23 flip-flops per ring element, and the maximum operating frequency of 200 MHz. These results indicate that the proposed distributed arbiter can be incorporated into a typical CDMA bus without significantly increasing the overall hardware complexity of the CDMA-related logic [11,16,17].

Communication performances are estimated under saturated and variable traffic loads. The variable traffic load is emulated with the PEs generating data bit streams independently at a rate  $\lambda$  following the Poisson distribution. The traffic rate,  $\lambda$ , is expressed in data bits per chip-interval, and it is the same for all the PEs. For saturated traffic load, every PE generates a new data stream immediately after successfully sending the previous one. In simulations, we use two traffic patterns: uniform and hotspot. Under the uniform traffic pattern, the generated data streams are destined randomly to other PEs with an equal probability. Under the hotspot traffic pattern, a PE is chosen as the hotspot receiving an extra partition of the traffic in addition to the regular uniform traffic. In particular, given a hotspot percentage of  $h$ , a newly generated data stream is directed to the hotspot PE with an additional  $h$  percent probability.

The design parameters varied between simulations are as follows: (a) the system size ( $M$ ), that is, the number of PEs and (b) the bus width ( $N$ ), that is, the maximum number of concurrent data stream transmissions supported by the CDMA bus. The  $N$ -channel CDMA system supports up to  $N$  parallel data-stream transmissions at the rate of  $1/N$  bits per chip-interval per channel. Therefore, the capacity of the CDMA bus equals 1 bit per chip-interval, independently of the bus width. For example, in the case of  $N=1$  (corresponding to the standard shared bus), a data stream can be transmitted over the bus at a time at the rate of 1 bit per chip-interval. In contrast, in an  $M$ -channel CDMA bus (corresponding to the static CDMA system), all  $M$  PEs can simultaneously send their data streams, but at the rate of  $1/M$  bits per chip-interval only.

The performance metrics used for the evaluations are as follows:

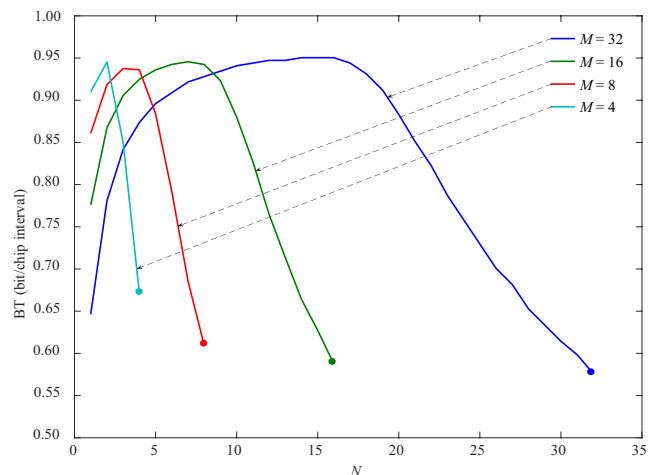
- *Data stream latency* (DSL), which is defined as the time period spanning from the time a data stream transmission request is issued to the arbiter unit at the source PE to the time the data stream is delivered to the destination PE.
- *Node throughput* (NT), which is defined as the average number of data bits transmitted by a PE per chip-interval under specific traffic load conditions.
- *Bus throughput* (BT), which is defined as the sum of node

throughputs of all the PEs. Note that, as the BT never exceeds the bus capacity of 1 bit per chip-interval, it effectively represents the fraction of utilized bus capacity.

#### 4.1 | Evaluation with saturated traffic load

Figure 4 shows the BT under saturated traffic load with a uniform traffic pattern as a function of the bus width for four different system sizes,  $M = 8, 16,$  and  $32$ . For each system size, the bus width changes from  $N=1$  to  $N=M$ . In these simulations, the length of data streams is fixed to 64 bits.

The BT is the most important system-level communication performance indicator, which indicates the absolute limit reached by the throughput of a CDMA-bus-based system under saturation traffic. As shown in Figure 4, the CDMA bus capacity cannot be fully utilized, although there are always pending requests for data stream transmissions under the saturated traffic condition. A high BT of more than 0.95 is achieved for a relatively wide range of middle-valued bus widths. However, there is a significant drop in BT outside of this range. The reduction in BT for small and large bus widths is a consequence of the arbitration-related overheads and the destination conflicts, respectively. Consider the case of  $N=1$ . Such a single-channel bus-based system has only one codeword, and consequently, only one data stream can be transferred at a time. As the resolution of destination conflicts by using the arbitration ring can proceed concurrently with the data stream transmission, there are potentially multiple PEs waiting for a codeword. Once the data stream transmission is completed, the codeword is released and then reassigned to one of the awaiting PEs. However, it takes some time (that is, chip intervals) until the codeword is transferred from the current

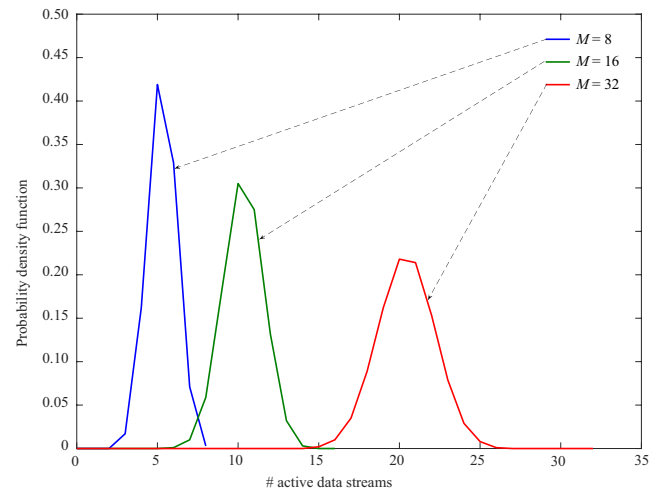


**FIGURE 4** BT under saturated traffic load with a uniform traffic pattern as a function of the bus width ( $N$ ) for four system sizes,  $M=4, 8, 16,$  and  $32$ . The ending points of the throughput curves correspond to static CDMA bus

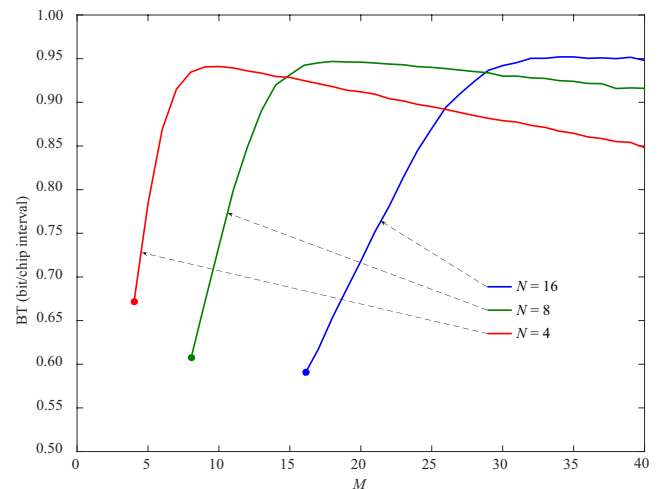


codeword owner to the requesting PE over the arbitration ring. During that time, the CDMA bus is idle, which effectively reduces the throughput. The codeword delivery time depends on the system size, because the token carrying the codeword needs to make more ring hops on average to reach the requesting PE in larger arbitration rings than in smaller ones. Hence, the system throughput of a single-channel CDMA system with  $M=32$  PEs is smaller than that of the systems with  $M=16$  and  $M=8$  PEs. In the case when  $N>1$ , the codeword delivery time influences only the channel associated with the specific codeword, whereas the data stream transmissions over other channels continue without interruption. Because only  $1/N$  fraction of the bus capacity is wasted during the codeword delivery time, the BT increases as the bus width increases.

Let us consider now how the BT changes when the bus width approaches the system size. In a CDMA-bus-based system, PEs are limited to receiving one data stream at a time. Under the saturated traffic condition, this constraint leads to frequent destination conflicts, when multiple source PEs need to contend for the same destination PE. While waiting for the destination PE, the source PE cannot start another transmission but it remains idle until the previous one is completed. Thus, the destination conflicts reduce the number of actively transmitting PEs, which effectively reduces the load of the CDMA bus. When the number of active transmitters is less than the number of available CDMA channels, a fraction of the bus capacity remains unused, and the BT drops. Figure 5 shows the probability density functions of the number of active transmitters in a static CDMA system under saturated traffic load for systems with  $M=8, 16$ , and  $32$  PEs. As can be observed, the probability that all  $M$  PEs intend to transmit to the same receiver PE is almost zero, and the probability that all  $M$  PEs intend to transmit to different receiver PEs is also zero. For example, in the system with  $M=32$  PEs, the expected number of active transmitters is 20, and this number varies for no more than 5 transmitters in 99% cases. With such a distribution, the destination conflicts do not influence the BT of systems with a bus width less than 15, because the number of active transmitters is never larger than the number of available channels. However, when  $N>15$ , some channels occasionally remain idle due to the lack of sufficient number of data streams, which lowers the BT (Figure 4). As the bus width further increases, the BT rapidly reduces and drops to 0.6 for a static CDMA system ( $M=32$ ). Similar observations hold for other system sizes. This result indicates that the bus width of approximately  $M/2$  represents the optimal configuration of a dynamic CDMA system in terms of BT. In other words, in CDMA systems with a constant bus width of  $N$ , the BT is maximized with  $M=2N$  PEs. This observation is confirmed by Figure 6, which shows the BT under saturated traffic as a function of the number of PEs for three different bus widths,  $N=4, 8$ , and  $16$ . For each bus width,  $N$ , the number of PEs is changed from  $N$  to 32.



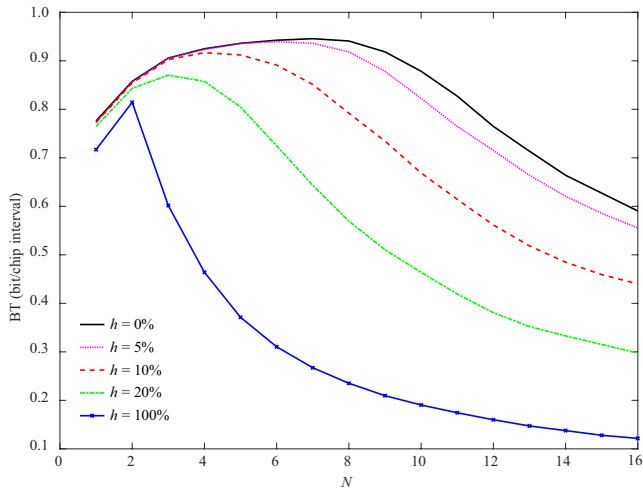
**FIGURE 5** Probability density functions of the number of active transmitters in a static CDMA system under saturated traffic load for three system sizes,  $M=8, 16$ , and  $32$



**FIGURE 6** BT under saturated traffic load as a function of the system size ( $M$ ) for three bus widths,  $N=4, 8$ , and  $16$

In all the cases, the maximum BT is achieved when the number of PEs is approximately  $2N$ . When  $M$  is less than  $2N$ , the BT is reduced due to the destination conflicts; when  $M$  is greater than  $2N$ , the throughput is reduced due to the arbitration overheads.

Figure 7 shows the BT under saturated hotspot traffic with different hotspot percentages ( $h$ ) as a function of the bus width in a CDMA-bus-based system comprising  $M=16$  PEs. Note that the curve for  $h=0\%$  corresponds to the uniform traffic pattern. As might be expected, the hotspot traffic has a significant effect on the BT, especially when  $h\geq 10\%$ . The occurrence of hotspot not only reduces the BT, but also changes the value of the bus width for which the BT is maximized. For example, in the system with  $M=16$

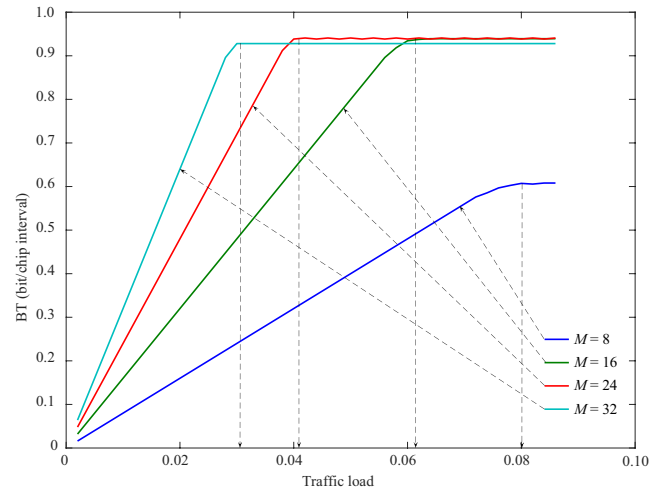


**FIGURE 7** BT under saturated hotspot traffic of 8-channel CDMA-bus-based system as a function of the bus width ( $N$ ) for five different hotspot percentages ( $h$ )

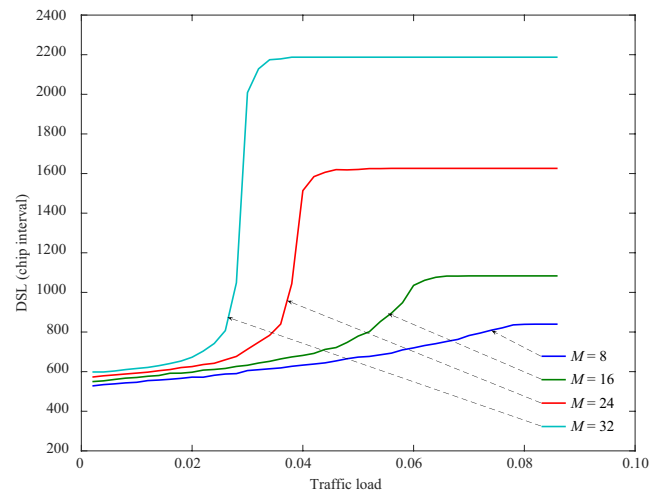
PEs with  $h=5\%$ , the optimum bus width is reduced to  $N=6$ , although the maximum BT is only slightly lower than that under the uniform traffic. When the hotspot percentage rises to  $h=20\%$ , the optimum bus width drops to  $N=3$ . These results suggest that, if the hotspot traffic is expected, the bus width should be set to a value less than  $M/2$ . Note also that the starting points of the curves in Figure 7 correspond to the conventional TDM bus ( $N=1$ ). Based on the shape of the curves in Figure 7, the optimally configured dynamic CDMA bus is less vulnerable to hotspot traffic than the TDM bus. This performance advantage of a CDMA bus can contribute to its possibility of parallel data transmissions.

## 4.2 | Evaluation with variable traffic load

The performance results for variable traffic load are shown in Figures 8 and 9. Figure 8 shows the BT as a function of traffic rate for CDMA systems with the bus width of  $N=8$ , and four different system sizes,  $M=8, 16$ , and 32. For low traffic rates, that is, when the total load of the CDMA bus is well below its capacity, all the generated data streams are transferred without any added delay. Thus, with the increase in traffic rate, the BT increases linearly up to the maximum. A higher system throughput in systems with a larger number of PEs is a consequence of a larger total load. Note that the traffic rate at which the BT saturates corresponds to the maximum node throughput (NT). As the bus width in this simulation study is maintained constant, and the bus capacity is shared equally among all PEs, an increase in the number of PEs inevitably leads to a decrease in the maximum NT. For example, the increase in the number of PEs from  $M=8$  to  $M=16$  increases the maximum BT by 54.5%. However, despite the higher BT, the maximum NT is decreased by 22.8%.



**FIGURE 8** BT of 8-channel CDMA-bus-based system as a function of traffic load ( $\lambda$ ) for four system sizes,  $M=8, 16, 24$ , and 32



**FIGURE 9** DSL of 8-channel CDMA-bus-based system as a function of traffic load ( $\lambda$ ) for four system sizes,  $M=8, 16, 24$ , and 32

The reduction in NT is due to a larger number of PEs, as the NT and BT are related as  $NT=BT/M$ . A further increase in the system size does not increase the maximum BT, although the maximum NT continues to decrease. Note that these results are in accordance with the analysis of BT under saturated traffic load, which indicates that the optimum size of a system with the bus width of  $N$  is approximately  $M=2N$ .

Figure 9 shows the DSL as a function of traffic rate for the CDMA system with the bus width of  $N=8$ , and four different system sizes,  $M=8, 16, 24$ , and 32. At low traffic rates, the receiver conflicts are rare, and the DSL is almost constant. A larger DSL in systems with a larger number of PEs is due to an increased arbitration delay. For example, at low traffic rates, the DSL in the system with  $M=16$  PEs is only 4% larger than that in the system with  $M=8$ . In the static CDMA system ( $M=8$ ), the number of available codewords (that is, bus channels) is always larger than the number of requested data stream

transmissions. Therefore, the gradual increase in the DSL is due to destination conflicts, which appear more frequently at higher traffic rates. In dynamic CDMA systems ( $M > 8$ ), the DSL rapidly increases as the traffic rate approaches the maximum NT. This is because of an additional transmission delay due to the lack of sufficient number of available codewords.

In conclusion, according to the simulation results, the main advantage of a dynamic CDMA system over a static one is a higher BT, that is, the possibility of full utilization of the available CDMA bus capacity. However, the higher BT of dynamic CDMA systems is accompanied by an increased DSL, especially at higher traffic rates. For example, when the traffic rate varies from 0 to the maximum NT, the 8-channel CDMA system with  $M = 16$  PEs provides up to 54.5% higher BT at the expense of 4%–30% higher DSL compared with the system with  $M = 8$  PEs. With a further increase in system size, the BT increases marginally, whereas the DSL increases substantially. Therefore, the system configuration with  $M = 2N$  PEs provides the best trade-off between the BT and DSL.

## 5 | CONCLUSIONS

The arbitration unit is a key module influencing the performance and scalability of CDMA-bus-based on-chip interconnects, especially in systems where it should provide a dynamic assignment of the available spreading codewords. Existing centralized arbiters for dynamic CDMA systems suffer from a complex logic design, which negatively influences their scalability and flexibility. In this paper, we proposed a novel design of a CDMA bus arbitration unit based on a token-ring algorithm. 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. Because of distributed organization, the ring arbiter can easily scale to large systems without a significant performance loss. Indeed, the simulations showed that the communication performances, in terms of bus utilization and latency, are influenced by the multi-cycle operation of the ring arbiter only under high traffic load conditions, and only in systems where the number of PEs is much larger than the number of available codewords. In the presented arbiter design, the number of codewords can be configured by software, which is an important characteristic allowing run-time adaptation to varying communication demands of the application. Another important feature of the proposed technique is that it can be implemented without adding long on-chip wires for request/grant signals, but by only relying on local connectivity between adjacent PEs. The analysis conducted in the study shows the advantages of the distributed arbitration over the centralized one in terms of scalability, flexibility, reusability, and conceptual simplicity. Overall, the proposed arbitration scheme represents

a practical approach of arbiter design for high-throughput CDMA-bus-based multiprocessor SoCs optimized for data-stream-oriented applications.

## ACKNOWLEDGMENTS

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

## ORCID

Tatjana R. Nikolic  <https://orcid.org/0000-0003-2649-4478>

## REFERENCES

1. S. Pasricha and N. Dutt, *On-chip communication architectures: System on chip interconnect*, Morgan Kaufmann, Burlington, USA, 2008.
2. A. Karkar et al., *A Survey of emerging interconnects for on-chip efficient multicast and broadcast in many-cores*, *IEEE Circ. Syst. Mag.* **16** (2016), 58–72.
3. S. J. Hollis et al., *Exploiting emergence in on-chip interconnects*, *IEEE Trans. Comput.* **63** (2014), 570–582.
4. J. Kim, I. Verbauwhede, and M.-C. F. Chang, *Design of an interconnect architecture and signaling technology for parallelism in communication*, *IEEE Trans. VLSI Syst.* **15** (2007), 881–894.
5. J. Kim et al., *A cost-effective latency-aware memory bus for symmetric multiprocessor systems*, *IEEE Trans. Comput.* **57** (2008), 1714–1719.
6. A. Assad et al., *A survey on energy-efficient methodologies and architectures of network-on-chip*, *Comput. Elect. Eng.* **40** (2014), 333–347.
7. D. Sigüenza-Tortosa, T. Ahoonen, and J. Nurmi, *Issues in the development of a practical NoC: The Proteo concept*, *Integr. VLSI J.* **38** (2004), 95–105.
8. D. Sanchez, G. Micheliogiannakis, and C. Kozyrakis, *An Analysis of on-chip interconnection networks for large-scale chip multiprocessors*, *ACM Trans. Archit. Code Optimization* **7** (2010), 1–28.
9. R. H. Bell et al., *CDMA as a multiprocessor interconnect strategy*, in *Proc. Conf. record Asilomar Conf. Signals, Syst. Comput.* (Pacific Grove, CA, USA), Nov. 2001, pp. 1246–1250.
10. D. Kim, M. Kim, and G. E. Sobelman, *CDMA-based network-on-chip architecture*, in *Proc. IEEE Asia-Pacific Conf. Circuits Syst.* (Tainan, Taiwan), Dec. 2004, pp. 137–140.
11. T. Nikolic, M. Stojcev, and G. Djordjevic, *CDMA bus-based on-chip interconnect infrastructure*, *Microel. Reliabil.* **49** (2009), 448–459.
12. X. Wang, T. Ahoonen, and J. Nurmi, *Applying CDMA technique to network-on-chip*, *IEEE Trans. VLSI Syst.* **15** (2007), 1091–1100.
13. J. Wang et al., *A new parallel CODEC technique for CDMA NoCs*, *IEEE Trans. Indust. Elect.* **65** (2018), 6527–6537.
14. K. E. Ahmed, R. Rizkand, and M. M. Farag, *Overloaded CDMA crossbar for network on chip*, *IEEE Trans. VLSI Syst.* **25** (2017), 1842–1855.
15. B. Halak, T. Ma, and X. Wei, *A dynamic CDMA network for multi-core systems*, *Microelectron. J.* **45** (2014), 424–434.
16. M. Kim, D. Kim, and G. E. Sobelman, *Adaptive scheduling for CDMA-based networks-on-chip*, in *Proc. Int. IEEE-NEWCAS Conf.* (Quebec, Canada), June, 2005, pp. 357–360. <https://doi.org/10.1109/NEWCAS.2005.1496682>

17. W. Lee and G. E. Sobelman, *Semi-distributed scheduling for flexible codeword assignment in a CDMA network-on-chip*, in Proc. IEEE 8th Int. Conf. ASIC (Changsha, China), 2009, pp. 431–434. <https://doi.org/10.1109/ASICON.2009.5351263>
18. C. Hamacher et al., *Computer Organization and Embedded Systems* 6th ed, McGraw-Hill Education, New York, 2012.
19. M. B. Slimane, I. B. Hafaiedh, and R. Robbana, *Formal-based design and verification of SoC arbitration protocols: A comparative analysis of TDMA and Round-Robin*, IEEE Des. Test **34** (2017), 54–62.
20. A. Kulmala, E. Salminen, and T. D. Hamalainen, *Distributed bus arbitration algorithm comparison on FPGA-based MPEG-4 multi-processor system on chip*, IET Comput. Digital Techniq. **2** (2008), 314–325.
21. J. Nurmi (ed.), *Interconnect-Centric Design for Advanced SoC and NoC*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.
22. J. Wang, Z. Lu, and Y. Li, *A new CDMA encoding/decoding method for on-chip communication network*, IEEE Trans. VLSI Syst. **24** (2016), 1607–1611.
23. B. Golubov, A. Efimov, and V. Skvortsov, *Walsh Series and Transforms: Theory and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
24. A. M. Chan et al., *Low-Complexity Localized Walsh Decoding for CDMA Systems*, in Proc. IEEE Military Comm. Conf. MILCOM (Washington, DC, USA), 2006, pp. 1–6.
25. T. R. Nikolic et al., *Improving fault-tolerance capability of on-chip binary CDMA bus*, J. Supercomput. **72** (2016), 275–294.

## AUTHOR BIOGRAPHIES



**Tatjana R. Nikolic** received her BS degree in communication engineering, and her MS and PhD degrees in electronic engineering from the Faculty of Electronic Engineering, University of Nis, Serbia, in 2000, 2005, and 2010, respectively. She is currently an associate professor with the Department of

Electrical Engineering at the Faculty of Electronic Engineering, University of Nis, Serbia. Her research interests include fault-tolerant on-chip communication, low-power system-on-chip design, and reconfigurable hardware architectures for application-specific acceleration.



**Goran S. Nikolic** received his BS degree in communication engineering, and his MS and PhD degrees in electronic engineering from the Faculty of Electronic Engineering, University of Nis, Serbia, in 2003, 2010, and 2019, respectively. He is a teaching assistant with the Department of Electrical Engineering at the Faculty of Electronic Engineering, University of Nis, Serbia. His research interests include fault-tolerant and low-power embedded system design, and wireless sensor networks.



**Goran Lj. Djordjevic** received his BS degree in computer science, and his MS and PhD degrees in electronic engineering from the Faculty of Electronic Engineering, University of Nis, Serbia, in 1989, 1994, and 1998, respectively. He is a full professor with the Department of Electrical Engineering at the Faculty of Electronic Engineering, University of Nis, Serbia. He has been involved in several research projects, focusing on parallel and distributed computing, reconfigurable computing, and wireless sensor networks. His current research interests include wireless sensor networks, embedded systems, networks-on-chip, and reconfigurable system-on-chip design.