Hindawi Publishing Corporation
VLSI Design
Volume 2007, Article ID 18372, 14 pages
doi:10.1155/2007/18372

Research Article

Comparison of a Ring On-Chip Network and a Code-Division Multiple-Access On-Chip Network

Xin Wang and Jari Nurmi

Institute of Digital and Computer Systems, Tampere University of Technology, 33101 Tampere, Finland

Received 5 October 2006; Accepted 6 February 2007

Recommended by Shashi Kumar

Two network-on-chip (NoC) designs are examined and compared in this paper. One design applies a bidirectional ring connection scheme, while the other design applies a code-division multiple-access (CDMA) connection scheme. Both of the designs apply globally asynchronous locally synchronous (GALS) scheme in order to deal with the issue of transferring data in a multiple-clock-domain environment of an on-chip system. The two NoC designs are compared with each other by their network structures, data transfer principles, network node structures, and their asynchronous designs. Both the synchronous and the asynchronous designs of the two on-chip networks are realized using a hardware-description language (HDL) in order to make the entire designs suit the commonly used synchronous design tools and flow. The performance estimation and comparison of the two NoC designs which are based on the HDL realizations are addressed. By comparing the two NoC designs, the advantages and disadvantages of applying direct connection and CDMA connection schemes in an on-chip communication network are discussed.

1. INTRODUCTION

As the technology feature size of integrated circuit fabrication is continuously shrinking down in the deep submicron regime, the number of components which can be integrated into an on-chip system is getting larger and larger. Therefore, the communications among the large number of components in a system-on-chip (SoC) are challenging tasks to deal with. Network-on-chip addresses this communication issue in an on-chip system by separating the concerns of communication from the concerns of computation. It means that the communication issue in an SoC is abstracted and handled by an on-chip communication network which hides the detailed information about how the communications are performed. Therefore, a system designer can pay more attention on the functions of system components and system integration by treating the NoC as a component of an on-chip system.

The NoC structures which have been proposed can be roughly classified into two categories, circuit-switched network and packet-switched network, in terms of the way of using the communication media. PROPHID architecture [1] is an example of a circuit-switched network which connects the terminals in the network by allocating them a set of time or space slices on the communication links. Examples in the packet-switched category are SPIN [2] and Proteo NoC [3]. SPIN network applies fat-tree topology and router blocks to transfer data packets from source node to destination node. In Proteo NoC, the components in the system are connected through network nodes and hubs. The network topology and connections in Proteo NoC can be customized and optimized for a specific application. Since an on-chip system can contain hundreds of functional intellectual property (IP) blocks in the near future, the circuit-switched network will face the problem of scalability and parallelism in that situation. Therefore, a packet-switched scheme is a better choice for future NoC designs because its structure is scalable and its data transfers are performed in parallel by sharing the communication media among multiple network nodes in a time-division manner.

As the number of IP blocks in an SoC is increasing, it is natural that different functional blocks work with different clock frequencies in an SoC. Hence, data transfer among multiple clock domains is another issue that needs to be handled by an on-chip network. A globally asynchronous locally synchronous (GALS) scheme [4] has been proposed to solve this problem of SoC designs. For an NoC, GALS means

that data transfers between each functional IP block and its attached network node are synchronous, whereas data transfers between network nodes are asynchronous.

From the analysis addressed in the previous two paragraphs, we can see that the GALS packet-switched scheme is a promising direction to explore the NoC designs for future on-chip systems. A common and natural way of composing a GALS packet-switched on-chip network is to connect two network nodes with a direct link. This way of connecting network nodes will be referred to as point-to-point (PTP) connection in this paper. Different patterns of the connection links in the network form different network topologies. For example, a mesh topology is applied in the NoC design presented in [5]. In a PTP connection NoC, the routing scheme and router architecture, such as the router presented in Æthereal NoC [6], are very important factors for supplying guaranteed services because the packet transfer latency may vary largely when data packets are transferred to different destinations or to the same destination via different routes in the network.

In order to eliminate the variance of the data transfer latency and complexity incurred by routing in a PTP connection NoC, a connection scheme which applies code-division multiple-access technique has been briefly introduced in [7].

By separating the different data from different users in the code domain, the data transfer latency in the CDMA NoC is stabilized by enabling multiple users to use the communication media parallel in time domain.

In order to examine the advantages and disadvantages of both the PTP connection scheme and the CDMA connection scheme, a bidirectional ring NoC design and a CDMA NoC design developed in our institute will be addressed and compared in terms of the network structure, data transfer principle, network node design, asynchronous design, and performance. For the sake of abbreviation, the bidirectional ring NoC design will be referred to as the PTP NoC in this paper.

The following sections of this paper will be arranged as follows. The network structures of the PTP NoC design and the CDMA NoC design will be presented and compared in Section 2. In Section 3, the data transfer principles of the two NoC designs will be studied and compared. Then the different network node structures in the two NoC designs will be addressed and compared in Section 4. Section 5 will present the asynchronous designs applied in both the PTP NoC and the CDMA NoC designs. In Section 6, two simulation networks of the two NoC designs built up for performance estimation will be presented. Then the performance comparison of the two NoC designs will be addressed upon the simulation results. Finally, the conclusion will be drawn in Section 7.

2. THE NETWORK STRUCTURES

2.1. The network structures of the two NoC designs

The PTP NoC design examined in this paper is based on a network node design [8] proposed for implementing GALS scheme in Proteo NoC. The network structure of the PTP NoC is illustrated in Figure 1. From Figure 1, we can see that

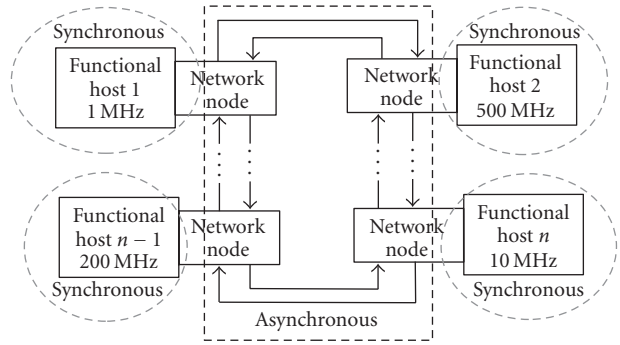

Figure 1: The bidirectional ring NoC structure.

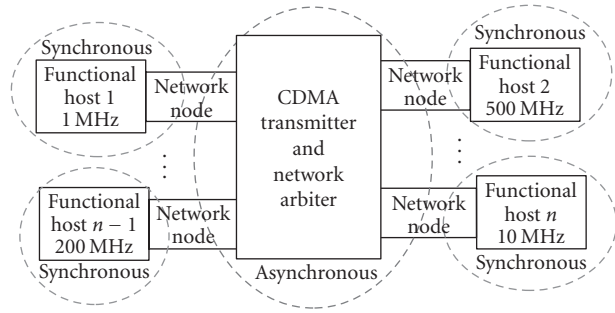

Figure 2: The CDMA NoC structure.

the communication between a "functional host" (functional IP block) and its network node is synchronous, while the data transfers among network nodes are preformed in asynchronous manner. For a large network, it may be necessary to break the entire network into sections and use some bridge nodes or hub nodes as addressed in [3] to connect the network sections together, whereas these bridges or hubs can be seen as one type of network nodes. Therefore, as illustrated in Figure 1, the PTP NoC can be composed simply by connecting the network nodes together with direct links.

The network structure of the CDMA NoC design introduced in [7] is illustrated in Figure 2. In the CDMA NoC, the GALS scheme is applied in the same way as in the PTP NoC; however, the network nodes in the CDMA NoC are no longer connected to each other with direct links. A "CDMA transmitter" and a "network arbiter" blocks are introduced in the CDMA NoC. All the network nodes need to communicate with the "CDMA transmitter" and "network arbiter" blocks directly. The functionality of the "CDMA transmitter" and "network arbiter" blocks will be addressed thoroughly in Section 3. With a direct comparison, we can see that the structure of the CDMA NoC is more complex than the PTP NoC since extra blocks are introduced in the CDMA NoC.

2.2. Distributed traffic versus centralized traffic

After a direct comparison of the two network structures in terms of simplicity, we can take a further analysis of the effects of the two different network structures on the features

of the networks. In the PTP NoC as illustrated in Figure 1, the data traffic load is distributed into the links among the network nodes. This distributed data traffic scheme has the merits of flexibility and scalability, whereas the main disadvantage of the PTP connection is that the data transfer latency between two network nodes can be largely different because the data may be transferred through different routes or because of data traffic congestions in the network. Therefore, the main concern of designing a PTP NoC is to find out the optimal topology and use all kinds of routing and flow-control methods to guarantee a high throughput and low transfer latency.

In the CDMA NoC illustrated in Figure 2, a centralized data transfer scheme is applied since all network nodes communicate with the "CDMA transmitter" and the "network arbiter" blocks directly. This centralized data transfer scheme is different from the conventional bus structures since it can supply parallel data transfers both in time and space domains by applying CDMA technique, whereas a bus structure supplies data transfer service among users in a time-division manner. The advantage of the centralized scheme applied in the CDMA NoC is that the data transfer latency between network nodes is a stable value. This stable transfer latency is contributed by the feature of parallel data transfer in time domain and the universal link distance among all network nodes. With the stable data transfer latency, the communication quality in the CDMA NoC will not vary.

3. THE DATA TRANSFER PRINCIPLES

The fundamental reason of the different network structures between the PTP NoC and the CDMA NoC is the different data transfer principles applied in the two NoC designs. Thus, the data transfer principles of the two NoC designs will be addressed and compared in this section.

3.1. Data transfer principle in the PTP NoC

As presented in Section 2.1, the PTP NoC is built by connecting the network nodes with direct links. The reason of this simplicity of connecting the network nodes is that the data are transferred in their original form in the PTP NoC. The only operation on the data is to encapsulate them into a packet format. This operation is done by a network node after getting the data from its attached functional host block. The packet format used in the PTP NoC is illustrated in Figure 3. After the data packet is formed, the PTP NoC will transfer the data bits with their original values to their destination through the links to the other nodes. Therefore, direct links between the network nodes in the PTP NoC are enough to handle the data transfers.

3.2. Data transfer principle in the CDMA NoC

As indicated by the name, the CDMA NoC applies CDMA technique to perform data transfers in the NoC. The basic principle of CDMA technique is illustrated in Figure 4. At the sending end, the data from different senders are encoded using a set of orthogonal spreading codes. Then the encoded data from different senders are added together for transmission without interfering with each other because of the orthogonal property of spreading codes. The orthogonal property means that the normalized autocorrelation of the spreading codes is 1, while the cross-correlation of the spreading codes is 0. Therefore, at the receiving end, the data can be decoded from the received sum signals by multiplying the received signals with the corresponding spreading code used for encoding. The data packet format applied in the CDMA NoC is the same format as illustrated in Figure 3. The issues related with the data encoding/decoding and transfer principles in the CDMA NoC will be addressed with details in the following subsections.

Destination node ID	Source node ID (optional for CDMA NoC)	(Other fields)
Data cell 1 Data cell 2(optional) Data cell 3(optional)		

Figure 3: Packet format specification.

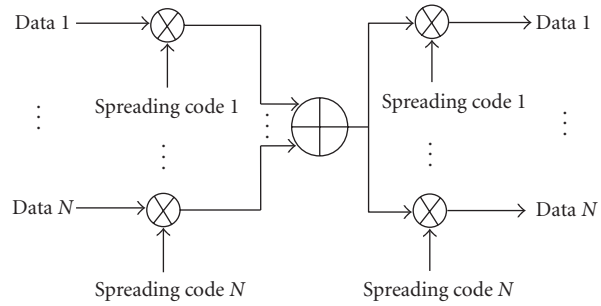

Figure 4: CDMA technique principle.

3.2.1. Data encoding and decoding schemes

Some CDMA encoding and decoding schemes for on-chip communication implemented by analog circuits have been proposed [9–11]. In those schemes, the encoded data are represented by the continuous voltage or capacitance value of the circuits. Therefore, the data transfers in the analog circuits are challenged by the coupling noise, clock skew, and the variations of capacitance and resistance caused by circuit implementation [11]. In order to avoid the challenges faced by the analog circuit implementation, digital encoding and decoding schemes are developed for the CDMA NoC and are illustrated in Figures 5 and 7, respectively. In the presented encoding scheme, data from different senders are fed into the encoding function bit by bit. Each data bit will be spread into S bits by multiplying it with a unique S-bit spreading code. The multiplications are performed by XOR logic gates as illustrated in Figure 5. Each bit of the S-bit encoded data generated by XOR operations is called a data chip. Then the data chips which come from different senders are added

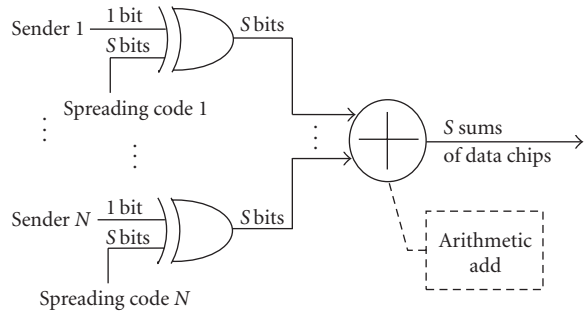

Figure 5: Digital CDMA encoding scheme.

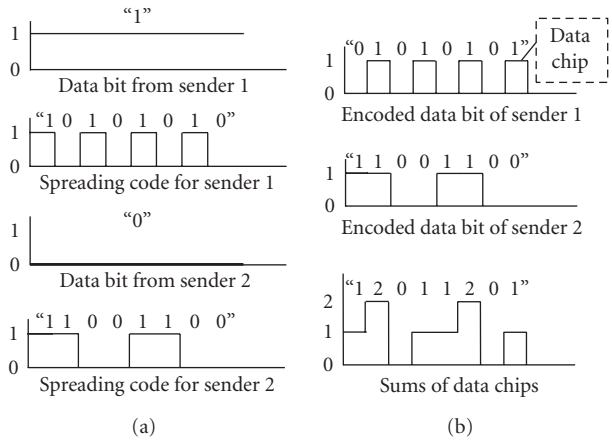

Figure 6: Data encoding example.

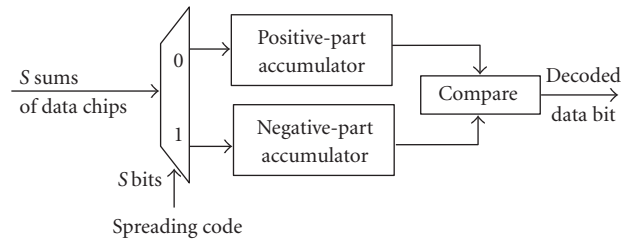

Figure 7: Digital CDMA decoding scheme.

together arithmetically according to their positions in the S-bit sequences. Namely, all the first data chips from different senders are added together, and all the second data chips from different senders are added together, and so on. Therefore, after the add operations, we will get S sum values of S-bit encoded data. Finally, as proposed in [12], binary equivalents of the S sum values are transferred to the receiving end one by one. An example of encoding two data bits from two senders is illustrated in Figure 6 in order to explain the proposed encoding scheme more specifically. Figure 6(a) shows two original data bits from different senders and two 8-bit spreading codes. The top two figures in Figure 6(b) illustrate the results after data encoding (XOR operations) for the original data bits. The bottom figure in Figure 6(b) presents the 8 sum values after adding operations. Then the binary equivalents of each sum value will be transferred to the receiving end. In this case, two binary bits are enough to represent the three possible decimal sum values, "0," "1," and "2." Hence, for example, if a decimal sum value "2" needs to be transferred, we need to transfer two binary digits "10."

The digital decoding scheme used in the CDMA NoC is illustrated in Figure 7. The decoding scheme accumulates the received sum values into two separated parts, a positive part and a negative part, according to the bit value of the spreading code used for decoding. For instance, as illustrated in Figure 7, the received first sum value will be put into the positive accumulator if the first bit of the spreading code for decoding is "0," otherwise, it will be put into the negative accumulator.

The same selection and accumulation operations are also performed on the other received sum values. The principle of this decoding scheme can be explained as follows. If the original data bit to be transferred is "1," after the XOR logic in the encoding scheme illustrated in Figure 5, it can only contribute a nonzero value to the sums of data chips when a bit of spreading code is "0." Similarly, the 0-value original data bit can only contribute a nonzero value to the sums of data chips when a bit of spreading code is "1." Therefore, after accumulating the sum values according to the bit values of the spreading code, either the positive part or negative part is larger than the other if the spreading codes have orthogonal and balance properties. Hence, the original data bit can be decoded by comparing the values between the two accumulators. Namely, if the positive accumulation value is larger than the negative accumulation value, the original data bit is "1"; otherwise, the original data bit is "0."

3.2.2. Spreading code selection

As discussed in Section 3.2.1, the presented encoding/decoding scheme requires the spreading codes used in the CDMA NoC to have both the orthogonal and balance properties. The orthogonal property was explained in the first paragraph of Section 3.2. The balance property means that the number of bit "1" and the number of bit "0" in a spreading code should be equal. Because Walsh code [13] has the required orthogonal and balance properties, it is chosen as the spreading code for the CDMA NoC. In an S-bit ($S = 2^N$, integer $N > 1$) length Walsh code set, there are S-1 sequences which have both the orthogonal and balance properties. Hence, the proposed CDMA NoC can have at most S-1 nodes connecting with one "CDMA transmitter" and one "network arbiter" block as illustrated in Figure 2.

3.2.3. Spreading code protocol

In a CDMA network, if multiple users simultaneously use the same spreading code to encode their data packets for transmission, the data to be transferred will interfere with each other because of the loss of orthogonal property among the spreading codes. This situation is called spreading code conflict, which should be avoided. Spreading code protocol is a

policy used to decide how to assign and use the spreading codes in a CDMA network in order to eliminate or reduce the possible spreading code conflicts. Several spreading code protocols have been proposed for CDMA packet radio network [14, 15]. Among these proposed spreading code protocols, only transmitter-based protocol (T protocol) and transmitter-receiver-based protocol (T-R protocol) are conflict-free if the users in the network send data to each other randomly. The principles of these two spreading code protocols will be shortly introduced in the following two paragraphs.

(1) *Transmitter-based protocol (T protocol)*: the unique spreading code allocated to each user will be used by the user himself to transfer data to others.

(2) *Transmitter-receiver-based protocol (T-R protocol)*: two unique spreading codes will be assigned to each user in the network, and then a user will generate a new spreading code from the assigned two unique codes for its data encoding.

Because the T-R protocol has the drawback of using a large amount of spreading codes and complicated decoding scheme, T protocol is preferred in the CDMA NoC. However, if T protocol is applied in the network, a receiver cannot choose the proper spreading code for decoding because it cannot know who is sending data to it. In order to solve this problem, an arbiter-based T protocol (A-T protocol) is proposed for the CDMA NoC. In a CDMA network which applies A-T protocol, each user is assigned a unique spreading code for data transfer. When a user wants to send data to another user, he will send the destination information of the data packet to the arbiter before starting data transmission. Then the arbiter will inform the requested receiver to prepare the corresponding spreading code for data decoding according to the sender. After the arbiter has got the acknowledge signal from the receiver, it will send an acknowledge signal back to the sender to grant its data transmission. If there are several users who want to send data to the same receiver, the arbiter will grant only one sender to send data at a time. Therefore, by applying the A-T protocol, spreading code conflicts in the CDMA NoC can be eliminated.

3.2.4. Parallel data transfer principle

The parallel data transfer principle of the CDMA NoC is based on the A-T spreading code protocol described in Section 3.2.3. By applying A-T spreading code protocol, every node in the CDMA NoC needs to send the destination address of the packets to the "network arbiter" as illustrated in Figure 2. After getting the grant signal from "network arbiter," the sender node will send data packets to the "CDMA transmitter" block. The data encoding operations and data transfers are performed in the "CDMA Transmitter" block. Finally, the data decoding operation will be carried out by the data receiving network node. Therefore, the data transfer process in the CDMA NoC can be clarified by describing the functions in the "network arbiter" and the "CDMA transmitter" block, respectively.

(1) *Network arbiter*. After receiving a data transfer request from a network node, "network arbiter" will inform the requested receiver node to prepare the proper spreading code for decoding and send a grant signal back to the sender node. In case that there are more than one sender nodes requesting to send data to the same receiver node simultaneously or at different times, the arbiter will apply "round-robin" arbitration scheme or the "first-come first-served" principle to guarantee that there is only one sender sending data to one specific receiver at a time. The reason for this limitation is that the "packet receiver" block in a network node can receive and decode data from only one sender at a time. However, if different sender nodes request to send data to different receiver nodes, these requests will not block each other and will be handled in parallel in the "network arbiter." The "network arbiter" in the CDMA NoC is different from the arbiter used in a conventional bus. This is because the "network arbiter" here is only used to set up spreading codes for receiving and it handles the requests in parallel in the time domain. In contrary, a conventional bus arbiter is used to allocate the usage of the common communication media among the users in the time-division manner.

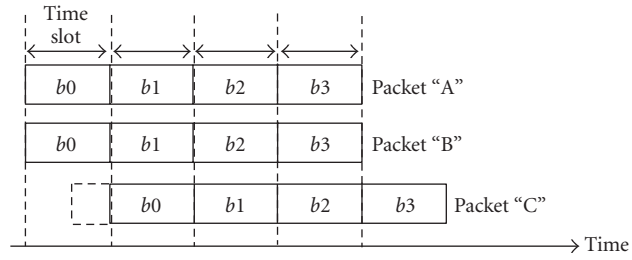

Figure 8: Bit-synchronous transfer scheme.

(2) *CDMA transmitter*. The sender node will start to send data packets to the "CDMA transmitter" after it gets the grant signal from the arbiter. Then the "CDMA transmitter" will encode the data to be transferred with the corresponding unique spreading code of the sender node. Although the "CDMA transmitter" block is implemented by asynchronous circuits, it applies the bit-synchronous transfer scheme. This means that the data from different nodes will be encoded and transmitted synchronously in terms of data bits rather than any clock signals. In Figure 8, the principle of the referred bit-synchronous transfer is illustrated by a situation in which network nodes "A" and "B" send data packets to "CDMA transmitter" simultaneously and node "C" sends a data packet later than "A" and "B." In this situation, the data packet from node "A" will be encoded and transmitted together with the data packet from node "B" synchronously in terms of each data bit. As the data packet from node "C" arrive at a later time point, the transmitter will handle the data bit from "packet C" together with the data bits from packets "A" and "B" at the next start point of the time slot for bit encoding and transmitting processes. The dotted-line frame at the head of the "packet C" in Figure 8 illustrates the waiting duration if the "packet C" arrived in the middle of the time slot for handling the previous data bit. The time slot for handling a data bit is formed by a four-phase

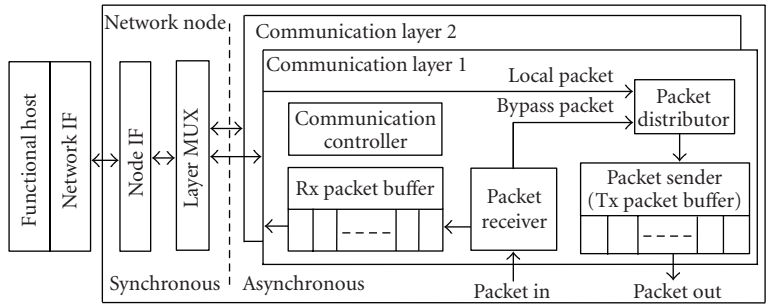

Figure 9: Network node structure of the PTP NoC.

handshake process. The bit-synchronous scheme can avoid the interferences caused by the phase offsets among the orthogonal spreading codes when the data bits from different nodes are encoded and transmitted asynchronously with each other. Because the nodes in the network can request data transfer randomly and independently of each other, "CDMA transmitter" applies the "first-come first-served" mechanism to ensure that the data encoding and transmission are performed as soon as there is a data transfer request.

3.3. Comparison of the data transfer principles

One advantage of the data transfer principle in the CDMA NoC is the feature of parallel data transfer. Although the data transfers in the PTP NoC can also be parallel if they take place in different links among the network nodes, the parallelism in the PTP NoC is largely limited by the possible traffic congestions in a link because the data are transferred through a link between two network nodes in a time-division manner. Another advantage of the CDMA data transfer principle is that no data routing is needed because of the centralized data transfer scheme. This feature can supply stable transfer latency in the CDMA NoC, which in turn facilitates the CDMA NoC to provide a guaranteed service for the on-chip system.

Another advantage of the CDMA NoC is that it can easily support multicast data transfers by requesting multiple receiver nodes to use the same spreading code for receiving. In the PTP NoC, the multicast transfer can be realized only by sending multiple copies of a data packet to its multiple destinations, unless extra logic is added in each network node to copy the multicast packet to both the functional host and the output link to the next node. This increases the traffic load in the network, or complicates the network implementation.

By comparing with the data transfer principle in the PTP NoC, one disadvantage of the CDMA data transfer principle is its complexity caused by the data encoding and decoding operations. Another drawback of the CDMA data transfer principle is that the data transfer efficiency obtained by parallel transfers in the time domain is compromised by the latency introduced by the data spreading scheme. As presented in Section 3.2.1, one data bit will be extended to S bits for the CDMA data transfers. The parameter S is the width of the spreading code applied in the CDMA NoC. As the number of nodes in the NoC increases, the width of the applied spreading code will also be increased. Then the data latency caused by the data spreading will be also increased.

4. THE NETWORK NODE STRUCTURES

"Network node" block is a common type of component needed in both of the PTP NoC and the CDMA NoC. However, different data transfer principles in the two NoC designs imply different structures in the network nodes. This section will present the network node structure in each of the two NoCs.

4.1. Network node structure in the PTP NoC

The network node structure of the PTP NoC is illustrated in Figure 9. The network node consists of "node if," "layer MUX," and "communication layer" blocks. The two blocks outside of the network node illustrate how a "functional host" block as presented in Figure 1 is connected with a network node through a network interface ("network if") block. In the PTP NoC, the applied interface standards include VCI [16] and OCP [17]. As illustrated in Figure 9, GALS scheme in the network is implemented by applying both synchronous and asynchronous designs in each network node. The synchronous blocks, "node if" and "layer MUX," are used to communicate with locally synchronous "functional host" in the system, while the asynchronous blocks are used to perform asynchronous communications among the network nodes. The arrows in Figure 9 demonstrate the data packet flow in a network node. The blocks in the network node illustrated in Figure 9 will be introduced in the following three paragraphs.

(1) *Node if*. This block is responsible for assembling the data from "functional host" into data packets or delivering the data packets from network node to "functional host" according to the interface standard applied in "network if" block.

(2) *Layer MUX*. As the name of this block indicates, this block behaves as a multiplexer used to connect "node if" block with a certain "communication layer" block during the data transfers between network node and "functional host" block.

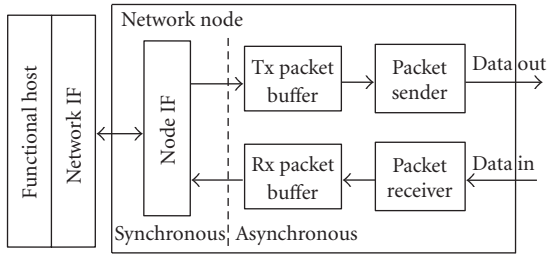

Figure 10: Network node structure of the CDMA NoC.

(3) *Communication layer*. The function of this block is to perform the globally asynchronous communication with other network nodes through a handshake protocol. As illustrated in Figure 9, the two "communication layer" blocks in a network node are used to connect with two other network nodes in the bidirectional ring NoC. More "communication layer" blocks can be used in a network node to implement other types of topology. There are five subblocks in a "communication layer" block. The "packet receiver" subblock is used to receive data packets from another network node. If the destination of the received packet is the current network node, the packet is called "incoming packet," and it will be stored in "Rx packet buffer." Otherwise, the received packet is called "bypass packet," and it will be dispatched into "packet sender" block via "packet distributor" for further transferring. The "communication controller" subblock in Figure 9 represents the controller which takes care of the necessary arbitrations and communication control.

4.2. Network node structure in the CDMA NoC

The block diagram of the network node structure of the CDMA NoC is shown in Figure 10 where the arrows represent the flows of data packets. In Figure 10, the "network if" block which belongs to the functional host is an interface block for connecting a functional host with a "network node." The GALS scheme is applied in "network node" block of the CDMA NoC by using synchronous design in the "node if" subblock and using asynchronous design in the other subblocks. The network interface standards supported in the CDMA NoC also include the VCI and OCP standards. The subblocks in the network node will be addressed in the following four paragraphs.

(1) *Node if*. This block is used to assemble the data from "functional host" into packets and send the packets to "Tx packet buffer" or disassemble the received packet from "Rx packet buffer" and send the extracted data to "functional host."

(2) *Tx/Rx packet buffer*. "Tx packet buffer" is used to store the data packets from "node if," and then deliver the packets to "packet sender." The "Rx packet buffer" stores and delivers the received packets from "packet receiver" to "node if."

(3) *Packet sender*. If "Tx packet buffer" is not empty, "packet sender" will fetch a data packet from the buffer by an asynchronous handshake protocol. Then it will extract the destination information from the fetched packet and send the destination address to "network arbiter." After "packet sender" gets the grant signal from the arbiter, it will start to send data packets to "CDMA transmitter."

(4) *Packet receiver*. After system reset, this subblock will wait for the sender information from "network arbiter" to select the proper spreading code for decoding. After the spreading code for decoding is ready, the receiver will send an acknowledge signal back to "network arbiter" and start to receive data from "CDMA transmitter," and then send the decoded data to "Rx packet buffer" in the packet format.

4.3. Comparison of the network node structures

By comparing with the presented network node in the PTP NoC, the network node in the CDMA NoC has less complexity. The main reason is that the network node of the CDMA NoC does not need to handle any bypass packets or the packet routing issues because of the centralized traffic scheme. Therefore, the "communication controller" block and the "packet distributor" block in the network node for the PTP NoC are not needed in the node for the CDMA NoC. When the data transfer parallelism needs to be increased in the PTP NoC, more "communication layer" blocks in a network node are needed in order to set up more links with other nodes, whereas the network node in the CDMA NoC does not need to change in this situation. One advantage of both of the network nodes is that they are both replicable because each network node structure in the network is the same. This advantage makes both the PTP NoC and the CDMA NoC designs modular.

5. ASYNCHRONOUS DESIGNS

As presented in Section 4, the GALS scheme is applied in the PTP NoC and in the CDMA NoC by implementing the global interconnect fabric with asynchronous designs. However, this is not the only way to implement GALS scheme in an on-chip network. For example, in "islands of synchronicity" (IoS) methodology presented in [18], the GALS scheme is implemented in SoC designs by localizing the clock in each of the functional IP blocks and connecting the isolated clock "islands," the functional IPs, with asynchronous communication links. If the IoS methodology is applied in the presented PTP NoC and the CDMA NoC, it means that all the blocks, including network nodes, "CDMA transmitter," and "network arbiter," in the designs need to be synchronous designs which work with different local clock frequencies. Then, the communications among the blocks in the NoC designs use asynchronous protocols. The advantage of applying the IoS methodology is that all the blocks in the design can be implemented by using standard synchronous design tools and flow. However, two disadvantages addressed in the following two paragraphs need to be noticed.

(1) *Synchronization cost*. The signals need to be synchronized with the local clocks when they cross different clock domains. If the IoS methodology is applied, two synchronization operations are needed when data enter into and leave from the global interconnect fabric during a data transfer process because the interconnect fabric works with its own clock rate. If the interconnect fabric is implemented by asynchronous designs, the synchronization step is not needed when data enter into the global interconnect fabric because a signal from a synchronous domain can enter into an asynchronous domain directly. Therefore, synchronization-related latency and area cost can be reduced 50% if the global interconnect fabric is implemented by asynchronous designs directly.

(2) *Area and power costs*. As presented in Section 4, "Tx/Rx packet buffer" composed by the asynchronous FIFO presented in [19] takes a large portion of the network node structure in both of the NoCs. As addressed in [19], the area and power costs of the asynchronous FIFO are 51.5% and 54.2% less than a synchronous implementation. Hence, if all the blocks related with global interconnection are implemented by synchronous designs, the area and power costs of the "Tx/Rx packet buffer" will be nearly doubled by comparing with the asynchronous implementation.

Therefore, in order to reduce the cost of synchronization, area, and power, asynchronous designs are applied to implement the blocks for global interconnections in the PTP NoC and the CDMA NoC. The asynchronous designs applied in the two NoC designs will be addressed in this section.

5.1. Asynchronous design in the PTP NoC

The basic component of the PTP NoC is the network node presented in Section 4.1. As illustrated in Figure 9, the blocks which apply asynchronous designs in the network node are the "communication controller," "packet receiver," "packet distributor/sender," and "packet Rx/Tx buffer" blocks. The four-phase dual-rail protocol is applied in the asynchronous designs in order to make the data transfers delay-insensitive. The control logic used in the asynchronous blocks of the PTP NoC will be presented in this subsection.

5.1.1. Control logic in the "communication controller"

The "communication controller" block is the main control block which takes care of data packet receiving, sending, and storing processes in the network node. Each of the mentioned packet handling processes is controlled by a control pipeline which can be seen as a finite state machine (FSM) in the "communication controller" block. The control processes of the FSMs are illustrated in Figure 11 and are explained in the following three paragraphs.

(1) *Packet receiving FSM*. As illustrated in Figure 11(a), there are six states in this FSM. The machine will move from its initial "rx_idle" state to "rx_pkt" state when "packet receiver" starts to receive a packet. After the packet receiving is completed, the FSM will move to "chk_addr" state to check the destination of the received packet. If the received packet is "incoming packet," the FSM will move to "chk_buf" state to check the status of "Rx packet buffer." If the buffer is not full, the "incoming packet" will be stored in the buffer during "st_pkt" state, otherwise, it will be held by "packet receiver" until there is room available in the buffer. If the received packet is "bypass packet," it will be sent to the network node connected to the output port of current node in "bp_pkt" state.

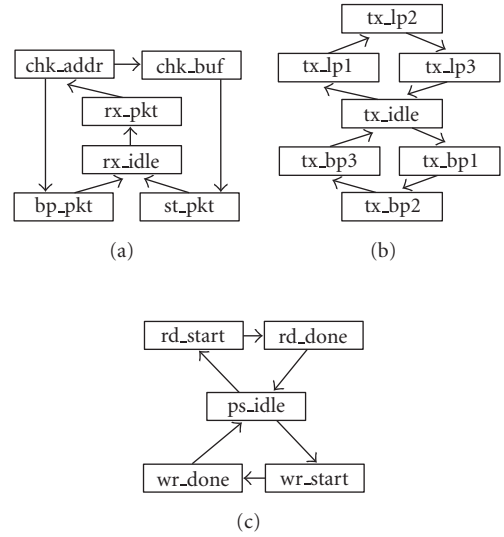

Figure 11: State transfer graphs of the FSMs.

(2) *Packet sending FSM*. This FSM illustrated in Figure 11(b) is responsible for sending two types of data packets into the "Tx packet buffer" in "packet sender" via "packet distributor" block. One type of packets is "local packet" which refers to the packet which comes from the "functional host" connected with the network node. Another type of packets is the "bypass packet" explained in Section 4.1. For sending "local packet," the FSM will be triggered by the signal from the "node if" block after a "local packet" is ready to be transferred. Then the FSM will go through "tx_lp1," "tx_lp2," and "tx_lp3" states for checking the status of the "Tx packet buffer," sending the packet into the buffer, and going back to "tx_idle" state, respectively. The process of sending a "bypass packet" into the transfer buffer is similar to the process of sending "local packet" except that the FSM will go through "tx_bp1," "tx_bp2," and "tx_bp3" states.

(3) *Packet storing FSM*. This FSM presented in Figure 11(c) is used to store or fetch an "incoming packet" to or from "Rx packet buffer" block. The process of storing an "incoming packet" will be triggered by packet receiving FSM during the "st_pkt" state as illustrated in Figure 11(a). The packet storing FSM will go through the "wr_start" and "wr_done" states to complete the storing task. The "rd_start" and "rd_done" states are for the process of fetching a stored "incoming packet" from "Rx packet buffer." The fetching process will be triggered by the "node if" block after getting flag signal from"communication controller" block.

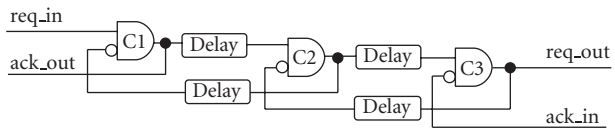

Figure 12: Micropipeline control logic.

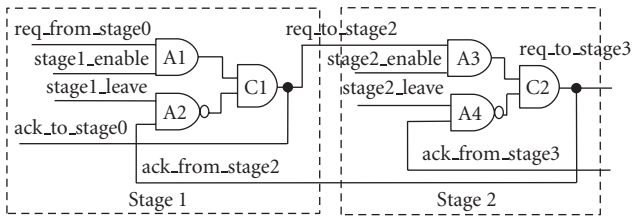

Figure 13: Control pipeline structure in the FSMs.

The presented control FSMs in the "communication controller" block are realized by applying the delay-insensitive micropipeline control logic presented in [20]. The structure of the micropipeline control logic is illustrated in Figure 12. The principle of micropipeline control logic is to use the output from the current stage to enable or disable the input of previous stage. Two stages of the control pipeline used in "communication controller" block for building the FSMs are illustrated in Figure 13. Each stage of the pipeline represents a state element of an FSM. In Figure 13, we can see that the FSM uses micropipeline control logic as the backbone and few AND gates as the delay components illustrated in Figure 12, hence it is also delay-insensitive. The state information of the FSM is passed through each stage in the pipeline by a four-phase handshake protocol. If we take the "stage 1" illustrated in Figure 13 as an example, when both the "req_from_stage0" and "stag1_enable" signals are "1," the output of "C1" will be set to logic "1" which indicates that the current state of the FSM is in "stage 1." Then the output of "C1" can be used as a request signal to trigger the control logic in the corresponding function blocks for a certain communication process.

5.1.2. Control logic in other blocks

Besides the FSMs in the "communication controller" block, control pipelines exist also in other asynchronous blocks used to perform the concrete task of moving the data packets in or out of the individual blocks through a four-phase dual-rail protocol. The FSMs in the "communication controller" block control the processes of receiving, sending, or storing data packets by triggering the control pipeline in the corresponding function blocks. The control pipeline structure used in the "packet receiver," "packet distributor," and "packet sender" blocks is illustrated in Figure 14. This structure is derived from the micropipeline control logic and is used to perform data transfers by interacting with the control signals coming from the "communication controller" block. For example, when the pipeline structure is used in "packet receiver," it will receive the packet receiving enable signal from the "communication controller" block as the enable signal for the first stage of the pipeline structure. Then, a request signal will be generated by gate "A2" in the pipeline stage as illustrated in Figure 14. The generated request signal will start a handshake process to receive and store a packet. When the acknowledge signal appears at the input of "A1," it will turn the output of "C1" to "1," which will clear the request signal via "A2" and enable the "C2" to capture the falling edge of the acknowledge signal through "A3." The falling edge of the "ack" signal means that the required tasks of the current step have been done and the current handshake process is completed. Hence, when it appears at the input of "A3," the output of "C2" will be triggered to "1" to enable the next stage to take over the control process of the next operation, for example, receiving next data packet cell in the "packet receive" block.

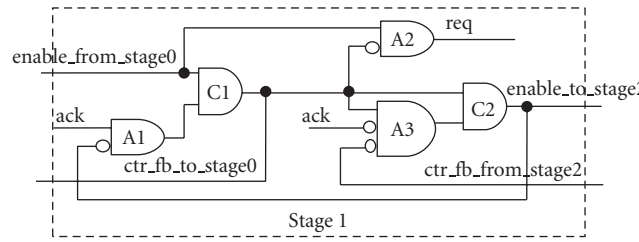

Figure 14: Block control pipeline structure.

The presented block control pipeline structure can only meet quasidelay-independent (QDI) model because the input "ack" signal is branched to "A1" and "A3," however, the timing requirement for distributing the "ack" input signal along the isochronic wire forks is quite loose since the logic delays in "A1" and "C1" are usually much larger than the logic delay of the inverter at the input of "A3."

The control pipeline structure illustrated in Figure 14 is also used in the "Tx/Rx packet buffer" blocks to control the process of accessing the asynchronous FIFOs presented in [19].

5.2. Asynchronous design in the CDMA NoC

As illustrated in Figure 2 and addressed in Section 4.2, the asynchronous blocks in the CDMA NoC include the "CDMA transmitter," "network arbiter," "Tx/Rx packet buffer," and "packet receiver/sender" blocks. Since the "CDMA transmitter" and "network arbiter" blocks are data-path centric blocks, the control logic used in these blocks can be composed by a straightforward C-element pipeline as illustrated in Figure 15. Each stage in the C-element pipeline is enabled by the enable signals generated by the data completion detection circuits. The control token will be passed from one stage to the next one through each C-element in the pipeline.

The control logic used in the "Tx/Rx packet buffer" and "packet receiver/sender" blocks of the network node for the CDMA NoC is similar to the control logic illustrated in Figure 14 except that the enable conditions of the C-element are different.

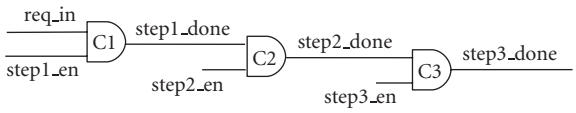

Figure 15: C-element control pipeline.

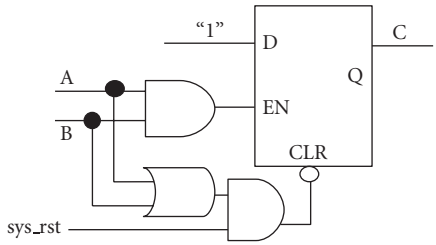

Figure 16: C-element structure.

5.3. Asynchronous design implementation

Since the synchronous designs in the presented PTP NoC and the CDMA NoC are done in register-transfer level (RTL) by using VHDL, it would be convenient for the implementation if the asynchronous designs apply the same design format. From the control logic structures described, we can see that the C-element is a basic component widely used in the asynchronous designs and a C-element is normally implemented in transistor level. Therefore, modeling the C-element in RTL is an important task for modeling the asynchronous designs using VHDL. Hence, an RTL two-input C-element structure is proposed in Figure 16. The proposed C-element structure is based on a D-flipflop (D-FF) which uses "A AND B" as the enable ("EN") signal and "A OR B" as the reset signal ("CLR"). The data input port ("D") of the D-FF is attached to logic "1" constantly. The idea of using a flipflop to build a C-element has been presented in [21] where an RS-FF is suggested to be used; whereas, the D-FF C-element structure in Figure 16 is more stable because it avoids data switching at the data input port.

The C-element structure illustrated in Figure 16 is hazard-free under one-input change assumption by applying the "AND" gate and "OR" gate at the "EN" port and "CLR" port, respectively. For certain two-input switch patterns, 00→11 and 11→00, the structure in Figure 16 is also hazard-free; whereas, the input switch patterns, 01→10 and 10→01, are not allowed because they may produce a logic error which depends on the wire delay. Because all the C-elements in the PTP and the CDMA NoC designs are used to follow a four-phase handshake protocol, there are no 01→10 or 10→01 input switch patterns for the C-elements in the designs. Thus, the proposed C-element structure can be safely used in the NoC designs.

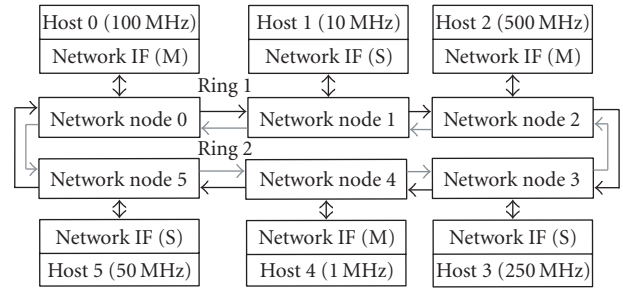

Figure 17: Six-node PTP NoC simulation network.

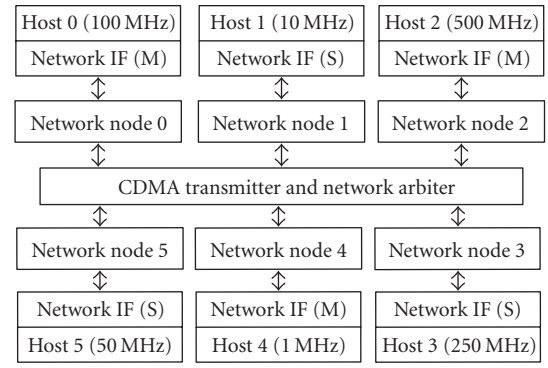

Figure 18: Six-node CDMA NoC simulation network.

By using the proposed RTL C-element structure, the asynchronous designs of the two NoCs are modeled in RTL using VHDL. Since the entire designs are in a uniform VHDL format, the commonly used synchronous design tools and flow can be used for implementing the NoC designs.

6. PERFORMANCE ESTIMATION

After the structures and designs of the PTP NoC and the CDMA NoC have been discussed, the performance of the two NoC designs will be addressed in this section. The two six-node simulation networks used for performance estimations and the estimation results will be presented and discussed in the following subsections.

6.1. The simulation network setup

In order to estimate the performance, two six-node networks have been set up for simulation purpose. The simulation network which applies the PTP NoC structure is illustrated in Figure 17, while the network which applies the CDMA NoC structure is illustrated in Figure 18. In each of the two simulation networks, six "functional host" blocks are connected into the network through six network nodes. The network nodes are connected to each other through the two different NoC structures, respectively, for the purpose of comparison. The interface standard applied in the simulation networks is BVCI standard [16]. Three hosts act as masters and the other three act as slaves, as denoted by the labels "M" and "S,"

Table 1: Area cost of the PTP NoC components.

Blocks of network node	Area (μm^2)
Node IF (BVCI slave type)	13430.8
Layer MUX	18346.0
Communication controller	7823.4
Packet distributor	6783.0
Packet sender (include Tx packet buffer)	44740.6
Packet receiver	6955.0
Rx packet buffer	40255.5
Total area of a network node (includes 2 "communication layer blocks")	**244891.8**

Table 2: Area cost of the CDMA NoC components.

Block name		Area (μm^2)
Network node	Node IF	18825.2
	Tx/Rx packet buffer	71778.3
	Packet sender	17707.0
	Packet receiver	23253.0
	Total area of a network node	**131563.5**
	CDMA transmitter	10338.3
	Network arbiter	17686.5

respectively, in the "network if" blocks. The master hosts can generate requests to any slave hosts, while the slave hosts can generate responses only for the received requests passively. The functional hosts in the two simulation networks are not implemented as any concrete designs. They are simulated by the stimulus which comes from the "network if" block to the network nodes. Hence, the configurations of the two simulation networks are the same except for the connection structures.

6.2. The area costs

By using the scheme described in Section 5.3, both the synchronous and asynchronous designs in the simulation networks are realized in RTL using VHDL. A 0.18 μm standard cell library is used for synthesizing the two simulation networks. The area costs of the two simulation networks are listed in Tables 1 and 2, respectively. As listed in Table 2, the area cost of the "network node" in the CDMA NoC is 53% smaller than the area of the "network node" block in the PTP NoC. The reason of this big difference of the area costs is that there are two "communication layer" blocks contained in each network node of the PTP NoC in order to set up bidirectional ring links. After including the area costs of "CDMA transmitter" and "network arbiter" blocks, the total area cost of the six-node CDMA NoC is 55% smaller than the area cost of the six-node PTP NoC.

6.3. The data transfer latencies

After the networks were synthesized, gate-level simulations of the two networks were performed using an event-driven simulator. Because the GALS scheme is applied in the two simulation networks, the data transfer latency of the networks can be separated into two parts which include synchronous transfer latency (STL) and asynchronous transfer latency (ATL). The STL refers to the data transfer latency between a functional host and the network node attached to it. STL depends on the local clock and the type of interface. Since the same "node if" block is applied in both networks, the STL values for the two simulation networks are the same. The measured STL values are listed in Table 3. The constant values in Table 3 are caused by the handshakes in the asynchronous domain. They are independent of the local clock rate but belong to the synchronous transfer processes. Therefore they are counted as a part of STL.

Table 3: Synchronous transfer latency.

Interface type	Latency of sending data to "network node"	Latency of receiving data from "network node"
BVCI master	8 local clock cycles + 2.5 ns	8 local clock cycles + 3.2 ns
BVCI slave	4 local clock cycles + 2.5 ns	4 local clock cycles + 3.1 ns

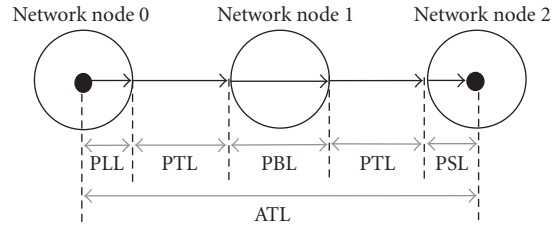

Figure 19: ATL parameters of the PTP NoC.

The ATL refers to the data transfer latency of transferring data packets from one network node to the other node through an NoC structure using asynchronous handshake protocols. The ATL values in the PTP and CDMA simulation networks consist of different parameters which will be discussed in the following subsections.

6.3.1. ATL in the PTP NoC

The ATL in the PTP NoC consists of four parameters: packet loading latency (PLL), packet transfer latency (PTL), packet bypass latency (PBL), and packet storing latency (PSL). These latency parameters are measured in a noncongested situation which means that no conflicts between "bypass packet" transfer and the "local packet" transfer are included in the simulation. The concept of the four latency parameters is illustrated in Figure 19 with an example that "network node 0" sends one packet to "network node 2" via "network node 1." The black arrows in Figure 19 represent the packet transfer direction. The portions of the transfer used to measure the different parameters of latency are marked by grey arrows in Figure 19 and are explained in the next four

Table 4: Measured ATL values of the PTP NoC.

Packet length	PLL (ns)	PTL (ns)	PBL (ns)	PSL (ns)
2 data cells	11.7	9.7	10.7	3.3
3 data cells	15.2	13.1	14.2	3.3
4 data cells	18.6	16.5	17.6	3.3

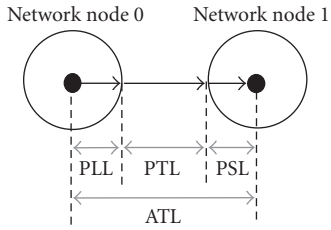

Figure 20: ATL parameters of the CDMA NoC.

paragraphs,

$$\text{ATL} = \text{PLL} + \text{PTL} \times (N + 1) + \text{PBL} \times N + \text{PSL}. \quad (1)$$

(1) *Packet load latency (PLL)*. It is the time used to load one "local packet" into "Tx packet buffer."

(2) *Packet transfer latency (PTL)*. This latency refers to the time used to transfer one data packet from the "packet sender" of a network node to the "packet receiver" of an adjacent node using a handshake protocol.

(3) *Packet bypass latency (PBL)*. After a network node receives a packet from another node, it will check its destination address. If it is a "bypass packet," it will be delivered into "Tx packet buffer." The time spent on this process is called PBL.

(4) *Packet storing latency (PSL)*. It is the time spent on storing one "incoming packet" into "Rx packet buffer."

The formula of calculating the ATL of transferring one packet in the PTP NoC is given in (1). It represents the situation in which the packet traverses several network nodes before reaching its destination. N refers to the number of intermediate nodes between the source node and destination node of a packet. If a packet is transferred between two adjacent network nodes, then N is 0. The values of ATL parameters measured in the simulation are listed in Table 4. The listed latency values only include the logic gate delay of the circuits, no wire delay is considered. More accurate latency values could be obtained by including the wire delay after layout.

6.3.2. ATL in the CDMA NoC

The ATL in the CDMA NoC consists of three parameters: packet loading latency (PLL), packet transfer latency (PTL), and packet storing latency (PSL). The concept of those ATL parameters is illustrated in Figure 20 with an example where "network node 0" sends one data packet to "network node 1." The black arrows in Figure 20 represent the packet transfer direction. The portions of the transfer used to measure the different parameters of ATL are marked by grey arrows in Figure 20 and are explained in the following three paragraphs.

Table 5: Measured ATL values of the CDMA NoC.

Packet length	PLL (ns)	PTL (ns)	PSL (ns)
2 data cells	5.7	384.6	5.5
3 data cells	5.7	768.9	5.5
4 data cells	5.7	1153.7	5.5

(1) *Packet load latency (PLL)*. This is the time used by the "packet sender" block in a "network node" to fetch one data packet from "Tx packet buffer" and prepare to send the data packet to "CDMA transmitter."

(2) *Packet transfer latency (PTL)*. This latency refers to the time used to transfer one data packet from the "packet sender" of the sender node to the "packet receiver" of the receiver node through the CDMA channel using a handshake protocol.

(3) *Packet storing latency (PSL)*. After the receiver node receives a data packet, it will spend a certain amount of time to store the received data packet into "Rx packet buffer." This time duration is measured as PSL.

The measured values of ATL parameters of the CDMA NoC are listed in Table 5. The listed latency values only include the logic gate delay of the circuits, no wire delay is considered. In Table 5, we can see that PTL increases as the packet length increases. This is because the data cells in a packet are sent in a serial manner in the CDMA NoC. Thus, more data cells need more transmission time. The PLL and PSL values are not affected by the packet length. The reason is that the data cells in a packet are loaded or stored in a parallel manner.

6.3.3. Comparing the ATL values

From the simulation results presented in Sections 6.3.1 and 6.3.2, we can see that the ATL value in the six-node CDMA NoC is a stable value for a certain data packet length, whereas the ATL value in the PTP NoC is a variable depending on the packet traffic route. The ATL parameter "PBL" of the PTP NoC does not exist in the ATL of the CDMA NoC because the data packets in the CDMA NoC are transferred directly from their source nodes to their destination nodes. The stable ATL value is an advantage of the CDMA NoC since it is very helpful for supplying guaranteed transfer latency service in the network. However, by comparing with the ATL value of the PTP NoC, the ATL value of the CDMA NoC is much larger. For example, according to values listed in Table 5, ATL of transferring a two-cell packet in the CDMA NoC is 395.8 nanoseconds. This value equals the ATL value of transferring the same size packet through 17 intermediate nodes in the PTP NoC according to (1) and Table 4. The reason for the large ATL value in the CDMA NoC is that each original data bit is extended into S bits by an S-bit spreading code during the obligatory data spreading process for CDMA transmission, and the encoded data are transferred bit by bit in the current realization of the CDMA NoC; whereas, in the

PTP NoC, the data bits are transferred cell by cell without any encoding, namely 32 original data bits are transferred at one time. Therefore, the ATL value of CDMA NoC can be reduced by transferring the encoded bits in parallel.

6.4. SystemC modeling for further estimation

The data transfer latency estimations made in Section 6.3 are based on two six-node simulation networks. However, in different applications, the number of nodes in an NoC can be different. Therefore, data transfer latency estimations under different numbers of network nodes of the two NoC designs would be helpful for further evaluation.

As discussed in Section 6.3.1, the data transfer latency of the PTP NoC is mainly affected by the number of intermediate network nodes which a packet passes through during the transfer. Therefore, by using the transfer latency values extracted from the six-node RTL simulation network, the ATL values of the PTP NoC with different numbers of network nodes can be estimated by using (1). For the CDMA NoC, the ATL values with different network node numbers are difficult to get from the six-node simulation network presented in Section 6.3 due to the lack of scalability in the CDMA NoC. Since the data transfer latency estimation presented in Section 6.3 is based on the RTL simulation network realized by using VHDL, any changes in the simulation network will incur a time-consuming synthesis and simulation design cycle. Therefore, a flexible and fast simulation model of the CDMA NoC is preferred for further ATL performance estimations of the CDMA NoC.

SystemC [22] is a C++ class library which can be used to model system-level designs. Since a SystemC model is totally described by a software programming language, the abstraction level of the system model can be very flexible and the simulation can run at a faster speed than an RTL model. Thus, a SystemC model of the CDMA NoC is built for the flexible and fast simulation purpose.

The SystemC model of the CDMA NoC is built in transaction level by modelling each block of the CDMA NoC as a channel [22]. The asynchronous communications among the blocks are modelled by calling each others' channel interface functions in the SystemC model. In order to estimate the ATL values of the CDMA NoC via the transaction-level SystemC model, a set of latency values listed in Table 6 is extracted from the gate-level simulation of the RTL six-node CDMA network presented in Section 6.3. By back-annotating the transfer latency values to the corresponding channels in the SystemC model, the ATL estimations of the CDMA NoC with different numbers of network nodes can be obtained through simulating the SystemC model in transaction level. The obtained ATL estimation values from the SystemC model simulations are listed in Table 7. From Table 7, we can see that the transfer latency of the CDMA NoC increases as the number of network nodes increases. The main reason of the latency increasing is that the data encoding latency in "CDMA transmitter" block is getting larger when the number of network nodes increased. Another reason is that the width of the orthogonal codes used for encoding increases as the number of nodes increased in the network. Thus, the spreading code loading latency in "packet receiver" block would be increased as a consequence. Because the back-annotated SystemC model of the CDMA NoC only uses a limited number of extracted latency values presented in Table 6, the ATL values listed in Table 7 only can give a quick glimpse on the latency situations when the number of network nodes in the CDMA NoC is changed. The accurate latency information needs to be obtained through real circuit implementations.

Table 6: Extracted transfer latency values.

Blocks	Processes	Latency
Tx/Rx packet buffer	Read	10.9 ns
	Write	11.5 ns
Packet sender	Send a 2-cell packet to "CDMA transmitter"	99.2 ns
Packet receiver	Load decoding PN code	1.2 ns
	Receive a 2-cell packet from "CDMA transmitter"	192.0 ns
Network arbiter	Arbitration	4.3 ns
CDMA transmitter	Data encoding	2.9 ns

Table 7: ATL estimation values of the CDMA NoC with different numbers of nodes.

Number of nodes	Spreading code length (bits)	ATL of sending a 2-cell packet
3	4	362.7 ns
6	8	411.8 ns
12	16	510.0 ns
24	32	706.4 ns

7. CONCLUSION

A PTP connection NoC and a CDMA connection NoC were examined and compared in this paper. Both of the presented NoC designs are packet-switched networks and support the GALS communication scheme. The two NoC designs are compared in terms of NoC structures, data transfer principles, network node designs, asynchronous designs, and the performances. The features of the two NoC structures are summarized in the following five paragraphs.

(1) *NoC structures.* The PTP NoC applies direct links among the network nodes and a distributed traffic scheme for the data communication. The CDMA NoC applies a centralized connection scheme and provides parallel data transfers in time domain.

(2) *Data transfer principles.* The PTP NoC transfers data packets in their original form among the links in the network. The CDMA NoC applies CDMA technique to share the centralized communication channel among all the network nodes both in time and space domains. The data streams from different network nodes in the CDMA NoC are separated from each other by encoding them with a set of orthogonal codes.

(3) *Network node designs.* The network node structure in the PTP NoC is more complex than the structure of the network node in the CDMA NoC. The communication control tasks in the CDMA NoC network node are less than the tasks in the PTP NoC network node. The complexity caused by packet routing processes in the network node of the PTP NoC is avoided in the network node of the CDMA NoC.

(4) *Asynchronous designs.* The asynchronous designs applied in the PTP and CDMA NoCs are similar to each other. The four-phase dual-rail protocol is applied in both NoC designs. The control logic used in the asynchronous designs of the two NoC designs is based on the micropipeline control logic. Both the synchronous and asynchronous designs in the two NoC designs are realized in RTL using VHDL.

(5) *Performance estimations.* Two simulation networks which apply the PTP NoC structure and the CDMA NoC structure, respectively, have been synthesized using a 0.18 μm standard cell library. The area cost of the CDMA simulation network is 55% smaller than the PTP simulation network. When the number of network nodes is certain, the ATL value of the CDMA NoC is a stable value for the same-size packets. However, the ATL value of the PTP NoC is smaller than the value of the CDMA NoC when no data transfer congestions are considered in the PTP NoC. One reason of the large ATL in the CDMA NoC is that the applied data spreading technique produces a large amount of encoded data bits for transmission. Another reason is that the encoded data are delivered bit by bit in the CDMA NoC, whereas the PTP NoC transfers 32 data bits at one time. Therefore, the ATL value of the CDMA NoC can be improved largely by increasing the number of data bits delivered at one time.

REFERENCES

[1] J. A. J. Leijten, J. L. van Meerbergen, A. H. Timmer, and J. A. G. Jess, "PROPHID: a data-driven multi-processor architecture for high performance DSP," in *Proceedings of European Design and Test Conference*, p. 611, Paris, France, March 1997.

[2] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE '00)*, pp. 250–256, Paris, France, March 2000.

[3] D. Sigüenza-Tortosa, T. Ahonen, and J. Nurmi, "Issues in the development of a practical NoC: the Proteo concept," *Integration, the VLSI Journal*, vol. 38, no. 1, pp. 95–105, 2004.

[4] J. Muttersbach, T. Villiger, H. Kaeslin, N. Felber, and W. Fichtner, "Globally-asynchronous locally-synchronous architectures to simplify the design of on-chip systems," in *Proceedings of the 12th Annual IEEE International ASIC/SOC Conference*, pp. 317–321, Washington, DC, USA, September 1999.

[5] C. A. Zeferino, F. G. M. E. Santo, and A. A. Susin, "ParlS: a parameterizable interconnect switch for networks-on-chip," in *Proceedings of the 17th Symposium on Integrated Cicuits and Systems Design (SBCCI '04)*, pp. 204–209, Pernambuco, Brazil, September 2004.

[6] K. Goossens, J. Dielissen, and A. Rădulescu, "Æthereal network on chip: concepts, architectures, and implementations," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 414–421, 2005.

[7] X. Wang and J. Nurmi, "An on-chip CDMA communication network," in *Proceedings of International Symposium on System-on-Chip*, pp. 155–160, Tampere, Finland, November 2005.

[8] X. Wang, D. Sigüenza-Tortosa, T. Ahonen, and J. Nurmi, "Asynchronous network node design for Network-on-Chip," in *Proceedings of International Symposium on Signals, Circuits and Systems (ISSCS '05)*, vol. 1, pp. 55–58, Iasi, Romania, July 2005.

[9] B.-K. Tan, R. Yoshimura, T. Matsuoka, and K. Taniguchi, "A novel dynamically programmable arithmetic array using code division multiple access bus," in *Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems (ICECS '01)*, vol. 2, pp. 913–916, Malta, September 2001.

[10] S. Shimizu, T. Matsuoka, and K. Taniguchi, "Parallel bus systems using code-division multiple access technique," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 240–243, Bangkok, Thailand, May 2003.

[11] M. Takahashi, B.-K. Tan, H. Iwamura, T. Matsuoka, and K. Taniguchi, "A study of robustness and coupling-noise immunity on simultaneous data transfer CDMA bus interface," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 611–614, Phoenix, Ariz, USA, May 2002.

[12] R. H. Bell Jr., C. Y. Kang, L. John, and E. E. Swartzlander Jr., "CDMA as a multiprocessor interconnect strategy," in *Proceedings of the 35th Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1246–1250, Pacific Grove, Calif, USA, November 2001.

[13] E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence CDMA and wideband CDMA cellular networks," *IEEE Communications Magazine*, vol. 36, no. 9, pp. 48–54, 1998.

[14] E. S. Sousa and J. A. Silvester, "Spreading code protocols for distributed spread-spectrum packet radio networks," *IEEE Transactions on Communications*, vol. 36, no. 3, pp. 272–281, 1988.

[15] D. D. Lin and T. J. Lim, "Subspace-based active user identification for a collision-free slotted ad hoc network," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 612–621, 2004.

[16] VSI Alliance, "Virtual Component Interface Standard version 2," April 2001, http://www.vsi.org/.

[17] OCP-IP Association, "Open Core Protocol Specification," 2001, http://www.ocpip.org/.

[18] A. P. Niranjan and P. Wiscombe, "Islands of synchronicity, a design methodology for SoC design," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE '04)*, vol. 3, pp. 64–69, Paris, France, February 2004.

[19] X. Wang and J. Nurmi, "A RTL asynchronous FIFO design using modified micropipeline," in *Proceedings of the 10th Biennial Baltic Electronics Conference (BEC '06)*, Tallinn, Estonia, October 2006.

[20] I. E. Sutherland, "Micropipelines," *Communications of the ACM*, vol. 32, no. 6, pp. 720–738, 1989.

[21] Q. T. Ho, J. B. Rigaud, L. Fesquet, M. Renaudin, and R. Rolland, "Implementing asynchronous circuits on LUT based FPGAs," in *Proceedings of the 12th International Conference on Field-Programmable Logic and Applications (FPL '02)*, vol. 2438 of *Lecture Notes in Computer Science*, pp. 36–46, Montpellier, France, September 2002.

[22] *IEEE Standard SystemC Language Reference Manual*, http://www.systemc.org/.