Applying CDMA Technique to Network-on-Chip

Xin Wang, Tapani Ahonen, and Jari Nurmi

***Abstract*—The issues of applying the code-division multiple access (CDMA) technique to an on-chip packet switched communication network are discussed in this paper. A packet switched network-on-chip (NoC) that applies the CDMA technique is realized in register-transfer level (RTL) using VHDL. The realized CDMA NoC supports the globally-asynchronous locally-synchronous (GALS) communication scheme by applying both synchronous and asynchronous designs. In a packet switched NoC, which applies a point-to-point connection scheme, e.g., a ring topology NoC, data transfer latency varies largely if the packets are transferred to different destinations or to the same destination through different routes in the network. The CDMA NoC can eliminate the data transfer latency variations by sharing the data communication media among multiple users concurrently. A six-node GALS CDMA on-chip network is modeled and simulated. The characteristics of the CDMA NoC are examined by comparing them with the characteristics of an on-chip bidirectional ring topology network. The simulation results reveal that the data transfer latency in the CDMA NoC is a constant value for a certain length of packet and is equivalent to the best case data transfer latency in the bidirectional ring network when data path width is set to 32 bits.**

***Index Terms*—Code-division multiple access (CDMA), integrated circuit (IC) design, network-on-chip (NoC).**

I. Introduction

As MORE and more components are integrated into an on-chip system, communication issues become complicated. Network-on-chip (NoC) is proposed to solve the on-chip communication problem by separating the concerns of communication from computation. The idea of NoC is to construct an on-chip communication network to perform data transfers among a large number of system components. The NoC structures that have been proposed can be roughly sorted into two categories, circuit switched network and packet switched network, according to their data switching modes. SoCBUS architecture [1], a mesh on-chip network, is an example of a circuit switched network that uses packet connected circuit scheme to allocate time or space slices on the switch links among the terminals in the network. Æthereal NoC [2] and Proteo NoC [3] are examples of the packet switched category. Æthereal NoC applies the combined guaranteed service and best-effort routers to transfer data packets in the network. In Proteo NoC, the components in the system are connected through network nodes and hubs. The network topology and data links in Proteo NoC can be customized and optimized for a specific application. Circuit-switched networks will face the problem of scalability and parallelism if they are applied in a future on-chip system which contains hundreds of functional intellectual property (IP) blocks. The packet switched network can overcome the shortcomings of the circuit switched network by dividing data streams into packets and routing packets to their destinations node by node. However, in a packet switched network that applies multihop point-to-point (PTP) connection scheme as in [2] and [3], the packet transfer latency will vary largely when data packets are transferred to different destinations or to the same destination via different routes in the network. Hence, the upper bound of the packet transfer latency is determined by the worst case scenario.

In order to eliminate variance of data transfer latency and complexity incurred by routing issues in a PTP connected NoC, an on-chip network which applies a code-division multiple access (CDMA) technique is introduced in this paper. As one of the spread-spectrum techniques, the CDMA technique [4] has been widely used in wireless communication systems because it has great bandwidth efficiency and multiple access capability. The CDMA technique applies a set of orthogonal codes to encode the data from different users before transmission in a shared communication media. Therefore, it permits multiple users to use the communication media concurrently by separating data from different users in the code domain. Hence, the CDMA NoC proposed in this paper can transfer data packets from different sources to their destinations directly and concurrently. Consequently, the large variance of data transfer latencies in a PTP connected NoC is eliminated in the CDMA NoC. The constant data transfer latency in the CDMA NoC is helpful for providing a guaranteed communication service for an on-chip system.

The rest of this paper is arranged as follows. In Section II, issues with applying CDMA technique into an on-chip network will be discussed. Section III presents the structure of the CDMA NoC. The realization of the basic components in the CDMA NoC is presented in Section IV. A six-node CDMA NoC is presented in Section V in order to examine characteristics of the CDMA NoC by comparing it with a PTP connected NoC. Finally, conclusions are drawn in Section VI.

II. Applying CDMA Technique to NoC

The principle of the CDMA technique is illustrated in Fig. 1. At the sending end, the data from different senders are encoded using a set of orthogonal spreading codes. 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 value and the cross-correlation value of spreading codes are 1 and 0, respectively. Autocorrelation of spreading codes refers to the sum of the products of a spreading code with itself, while cross-correlation refers to the sum of the products of two different spreading

Manuscript received May 30, 2006; revised April 15, 2007.

The authors are with the Institute of Digital and Computer Systems, Tampere University of Technology, 33101 Tampere, Finland (e-mail: xin.wang@tut.fi; tapani.ahonen@tut.fi; jari.nurmi@tut.fi).

Digital Object Identifier 10.1109/TVLSI.2007.903914

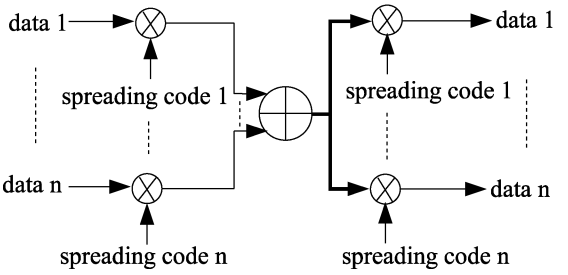

Fig. 1. CDMA technique principle.

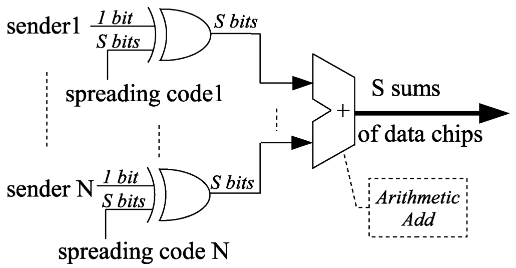

Fig. 2. Digital CDMA encoding scheme.

codes. Because of the orthogonal property, at the receiving end, the data can be decoded from the received sum signals by multiplying the received signals with the spreading code used for encoding. The following three subsections will discuss the issues related to apply the CDMA technique in an NoC.

A. Digital Encoding and Decoding Scheme

Several on-chip bus schemes that apply the CDMA technique have been presented in [5]–[8]. Those schemes are implemented by analog circuits, namely, the encoded data are represented by the continuous voltage or capacitance value of the circuits. Therefore, the data transfers in the analog bus are challenged by the coupling noise, clock skew, and the variations of capacitance and resistance caused by circuit implementation [8]. In order to avoid the challenges faced by the analog circuit implementation, digital encoding and decoding schemes developed for the CDMA NoC are illustrated in Figs. 2 and 4, respectively. In the encoding scheme illustrated in Fig. 2, data from different senders fed into the encoder bit by bit. Each data bit will be spread into S bits by XOR logic operations with a unique S-bit spreading code as illustrated in Fig. 2. 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 together arithmetically according to their bit 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 [9], binary equivalents of the S sum values are transferred to the receiving end. An example of encoding two data bits from two senders is illustrated in Fig. 3 in order to illustrate the proposed encoding scheme in more detail. Fig. 3(a) illustrates two original data bits from different senders and two 8-bit spreading codes. The top two figures in Fig. 3(b) illustrate the results after data encoding (XOR operations) for the original data bits. The bottom figure in Fig. 3(b) presents the eight sum values after add 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 different decimal sum values, “0,” “1,” and “2.” For example, if a decimal sum value “2” needs to be transferred, we need to transfer two binary digits “10.”

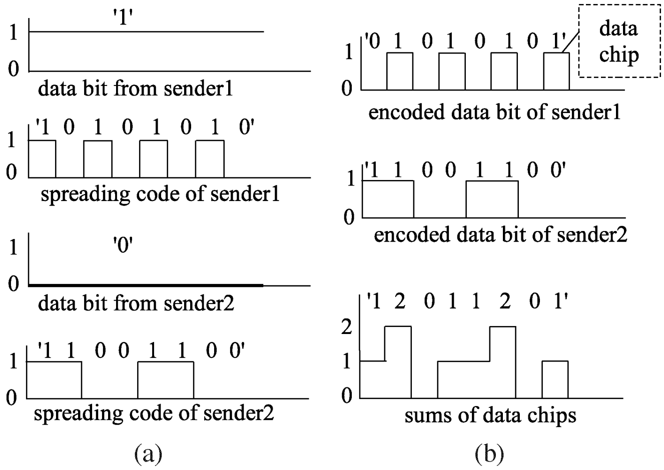

Fig. 3. Data encoding example.

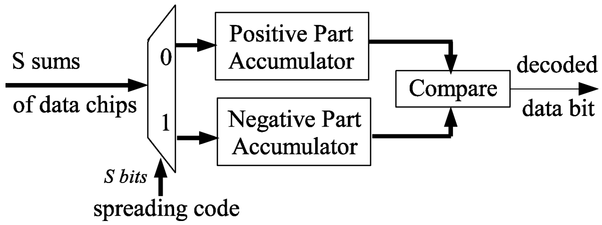

Fig. 4. Digital CDMA decoding scheme.

The digital decoding scheme applied in the CDMA NoC is depicted in Fig. 4. The decoding scheme accumulates the received sum values into two separate 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 Fig. 4, 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 operations in the encoding scheme illustrated in Fig. 2, it can only contribute 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 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 are orthogonal and balance. Hence, the original data bit can be decoded by comparing the values between the two accumulators. Namely, if the value of the positive accumulator is larger than the value in the negative accumulator, the original data bit is “1”; otherwise, the original data bit is “0.”

B. Spreading Code Selection

As discussed in Section II-A, the proposed decoding scheme requires the spreading codes used in the CDMA NoC to have both the orthogonal and balance properties. The orthogonal property has been explained in the first paragraph of Section II. The balance property means that the number of bit "1" and bit "0" in a spreading code should be equal. Several types of spreading codes have been proposed for CDMA communication, such as Walsh code, M-sequence, Gold sequence, and Kasami sequence, etc. [10]. However, only Walsh code [10] has the required orthogonal and balance properties. Therefore, Walsh code family is chosen as the spreading code library for the CDMA NoC. In an S-bit ($S = 2^N$, integer $N > 1$) length Walsh code set, there are $S-1$ sequences that have both the orthogonal and balance properties. Hence, the proposed CDMA NoC can have at most $S - 1$ network nodes. The length of applied Walsh code set should be kept as small as possible according to the number network nodes. The purpose is to reduce the number of data chips generated during data encoding operations as illustrated in Fig. 2. For example, if there are six nodes in the CDMA NoC, the 8-bit Walsh code set should be used instead of a longer Walsh code set.

C. Spreading Code Protocol

In a CDMA network, if multiple users use the same spreading code to encode their data packets for transmission simultaneously, 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 during the communication processes. Several spreading code protocols have been presented for CDMA packet radio network [11], [12] and will be shortly introduced in the following six paragraphs.

1) *Common Code Protocol (C protocol)*: All users in the network use the same spreading code to encode their data packets to be transferred.
2) *Receiver-Based Protocol (R protocol)*: Each user in the network is assigned a unique spreading code used by the other users who want to send data to that user.
3) *Transmitter-Based Protocol (T protocol)*: The unique spreading code allocated to each user is used by the user himself to transfer data to others.
4) *Common-Transmitter-Based Protocol (C-T protocol)*: The destination address portion of a data packet is encoded using C protocol, whereas, the data portion of a packet is encoded using T protocol.
5) *Receiver-Transmitter-Based Protocol (R-T protocol)*: It is the same as the C-T protocol except that the destination address portion of a data packet is encoded using R protocol.
6) *Transmitter-Receiver-Based Protocol (T-R protocol)*: Two unique spreading codes are 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.

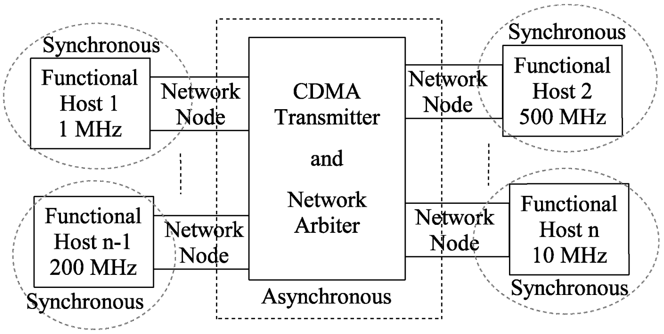

Fig. 5. Proposed CDMA NoC structure.

Among the introduced spreading code protocols, only T protocol and T-R protocol are conflict-free if the users in the network send data to each other randomly. 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 developed for the CDMA NoC. In a CDMA NoC which applies A-T protocol, each user is assigned with 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 is more than one user who wants to send data to the same receiver, the arbiter will grant only one sender to send data at a time. Therefore, by applying the proposed A-T protocol, spreading code conflicts in the CDMA NoC can be eliminated.

III. CDMA NoC Structure

The proposed CDMA NoC is a packet switched network that consists of "Network Node," "CDMA Transmitter," and "Network Arbiter" blocks as illustrated in Fig. 5. The functional IP blocks (functional hosts) are connected to the CDMA NoC through individual "Network Node" blocks. The CDMA communications in the network are performed by "CDMA Transmitter" and "Network Arbiter" blocks. Because the different functional hosts may work at different clock frequencies as illustrated in Fig. 5, coordinating the data transfers among different clock domains would be a problem. A globally-asynchronous locally-synchronous (GALS) scheme [13] has been proposed as a solution for this problem. Applying the GALS scheme to the CDMA NoC means that the communications between each functional host and its network node use local clock frequency, while the communications between network nodes through the CDMA network are asynchronous. In order to support the GALS scheme, both synchronous and asynchronous circuits are applied in the design. The three types of components in the CDMA NoC will be presented in the following three subsections.

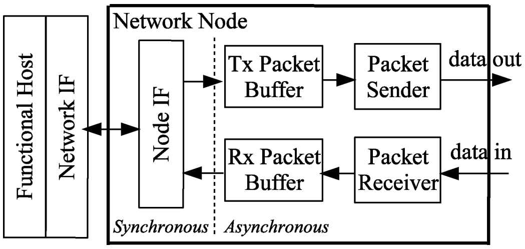

Fig. 6. Block diagram of the network node in CDMA NoC.

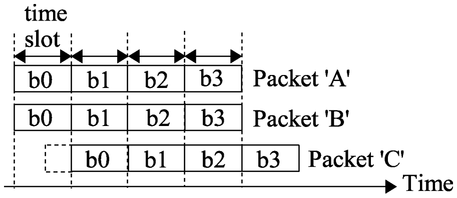

Fig. 7. Bit-synchronous transfer scheme.

A. Network Node

The block diagram of the "Network Node" in the CDMA NoC is illustrated in Fig. 6, where the arrows represent the flows of data packets. In Fig. 6, the "Network IF" block, which belongs to the functional host, is an interface block for connecting a functional host with a "Network Node" through VCI [14] or OCP interface standard [15]. GALS scheme is realized in "Network Node" block by using synchronous design in the "Node IF" subblock and using asynchronous design in the other subblocks. The function of the subblocks in a "Network Node" will be described in the following four paragraphs.

1) *Node IF*: This block is used to receive data from the "Network IF" block of a functional host through the applied VCI or OCP standard. Then it will assemble the received data into packet format and send the packet to "Tx Packet Buffer," or disassemble the received packet from "Rx Packet Buffer" and send the extracted data to the functional host.
2) *Tx/Rx Packet Buffer*: These two blocks are buffers that consist of the asynchronous first-input–first-output (FIFO) presented in [16]. "Tx Packet Buffer" is used to store the data packets from "Node IF" block, and then deliver the packets to "Packet Sender" block. 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 the data packet to "CDMA Transmitter."
4) *Packet Receiver*: After system reset, this block 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 wait to receive and decode the data from "CDMA Transmitter," and then send the decoded data to "Rx Packet Buffer" in packet format.

B. Network Arbiter

"Network Arbiter" block is the core component to implement the A-T spreading code protocol presented in Section II-C. By applying A-T spreading code protocol, every sender node cannot start to send data packets to "CDMA Transmitter" until it gets the grant signal from "Network Arbiter." "Network Arbiter" takes charge of informing the requested receiver node to prepare the proper spreading code for decoding and sending a grant signal back to the sender node. In the 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 a "round-robin" arbitration scheme or the "first-come first-served" principle, respectively, to guarantee that there is only one sender sending data to one specific receiver at a time. However, if different sender nodes request to send data to different receiver nodes, these requests would 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. The reason is that 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. However, a conventional bus arbiter is used to allocate the usage of the common communication media among the users in the time-division manner.

C. CDMA Transmitter

The "CDMA Transmitter" block takes care of receiving data packets from network nodes and encoding the data to be transferred with the corresponding unique spreading code of the sender node. Although this block is realized using asynchronous circuits, it applies a bit-synchronous transfer scheme. It means that the data from different nodes will be encoded and transmitted synchronously in terms of data bits rather than any clock signals. In Fig. 7, the principle of the referred bit-synchronous transfer scheme is illustrated by a situation that 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. When the data packet from node "C" arrives at a later time point, the transmitter will handle the data bit of "Packet C" together with the data bits of packet "A" and "B" at the next start point of the time slot for bit encoding and transmitting processes. The dot-line frame at the head of the "Packet C" in Fig. 7 is used to illustrate 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 handshake process. The bit-synchronous transfer scheme can avoid the interferences caused by the phase offsets among the orthogonal spreading codes if 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

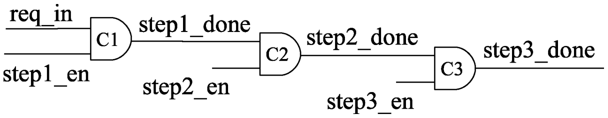

Fig. 8. C-element control pipeline.

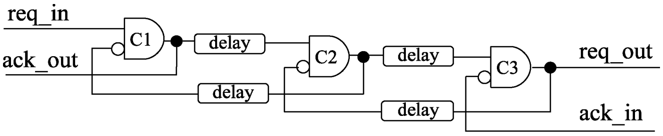

Fig. 9. Micropipeline control logic.

Transmitter" applies the "first come, first served" mechanism to ensure that the data encoding and transmission are performed as soon as there is data transfer request.

IV. Realization

Two issues related with realizing the CDMA NoC are addressed in this section. One issue is about asynchronous design realization. Another is the configuration of the data path in the CDMA NoC.

A. Asynchronous Design

As illustrated in Fig. 5 and addressed in Section III-A, the asynchronous blocks in the CDMA NoC include the "CDMA Transmitter," "Network Arbiter," "Tx/Rx Packet Buffer," and "Packet Receiver/Sender" blocks. The important part of the asynchronous design of these blocks is the control logic. Since the "CDMA Transmitter" and "Network Arbiter" blocks are data-path centric blocks, the control logic used in these blocks is composed by a straightforward C-element pipeline as illustrated in Fig. 8. Each stage in the C-element pipeline is enabled by the enable signals generated from 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 bases on the micropipeline control logic presented in [17] and illustrated in Fig. 9. The principle of micropipeline control logic is to use the output from the current stage to enable or disable the input of previous stage. The "delay" components illustrated in Fig. 9 are realized by logic gates of generating or receiving four-phase handshake signals for control tasks in the asynchronous blocks in the CDMA NoC. An example with more details about applying micropipeline control logics to asynchronous designs can be found in [18].

In order to suit the conventional synchronous design tools and other synchronous designs in the CDMA NoC, all the asynchronous blocks of the CDMA NoC are realized in RTL using VHDL together with the synchronous blocks. The basic principle is to model the basic components, C-element, latches, and combinational logic gates, in RTL using VHDL, and then build the asynchronous circuits using these RTL component models in a hierarchical way.

TABLE I
Area Cost of CDMA NoC Components

Data Path Width ***Block Name***		**Area (K equivalent gates)**			
		1-bit	**8-bit**	**16-bit**	**32-bit**
Node IF	Target	1.600			
	Initiator	3.882			
Tx/Rx Packet Buffer		6.101			
Packet Sender		1.505	1.509	1.513	0.962
Packet Receiver		1.977 ***(100.0%)***	7.331 ***(370.9%)***	13.718 ***(694.1%)***	26.488 ***(1340%)***
CDMA Transmitter		0.879 ***(100.0%)***	3.970 ***(451.8%)***	7.730 ***(879.6%)***	15.161 ***(1725%)***
Network Arbiter		0.936			

B. Data Path Configuration

Figs. 2 and 4 illustrate the principle of data encoding and decoding schemes used in the CDMA NoC by an example of processing and delivering one data chip of encoded data from the sender to the receiver at one time. Since one original data bit will be spread into S bits after encoding, the degree of data transfer parallelism between the "CDMA Transmitter" and "Packet Sender/Receiver" blocks affects the data transfer latency in the CDMA NoC largely. Namely, increasing the number of data bit encoded and delivered via "CDMA Transmitter" at one time can reduce the data transfer latency in the CDMA NoC and vice versa. However, increasing the data processing and delivering parallelism will incur larger area cost. Hence, in order to figure the tradeoff character between the parallelism and the area cost, the "Packet Sender," "CDMA Transmitter," and "Packet Receiver" blocks have been realized with four different data path configurations. According to the number of data bit transferred from a "Packet Sender" to a "Packet Receiver" through "CDMA Transmitter," the configurations are named as 1-, 8-, 16-, and 32-bit schemes.

C. Synthesis Results

The components of the CDMA NoC are synthesized using a 0.18-μm standard cell library. The Basic VCI (BVCI) interface standard [14] is applied in the realization of "Node IF" block. The data width and buffer depth in the "Tx/Rx Packet Buffer" blocks are set to 32 bits and 4 packets, respectively. In order to facilitate the simulation work later on, six network node and 8-bit Walsh codes are applied for synthesizing the "CDMA Transmitter," "Network Arbiter," and "Packet Sender/Receiver" blocks. The area cost of the components of the CDMA NoC under different data path configurations are listed in Table I. The area cost figures in Table I are presented as the number of equivalent gates. 85 K gates/mm^2 is used to calculate the number of equivalent gates for the 0.18-μm standard cell library.

From Table I we can see that when the data path width is increased from 1 to 32 bits, the area cost of "Packet Receiver" and "CDMA Transmitter" becomes 13 and 17 times larger. The area increase is due to the duplications of the encoding and decoding logic in the "CDMA Transmitter" and "Packet Receiver" blocks for increasing the data path width. By comparing the ratio of increased data path width, the increased area cost of the components is reasonable. To be noticed in Table I is that the area cost of the 32-bit version of "Packet Sender" block is smaller

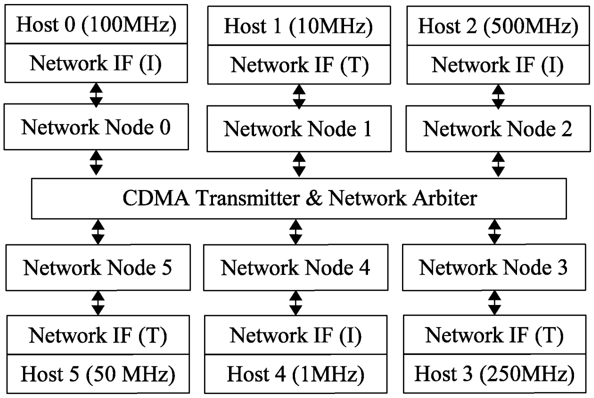

Fig. 10. Six-node CDMA NoC simulation network.

Destination Node ID	Source Node ID (optional for CDMA NoC)	other fields (optional)
Data Cell 1		
Data Cell 2 (optional)		
Data Cell 3 (optional)		

Fig. 11. Data packet format specification.

than others. The reason is that the data width of the output of "Tx Packet Buffer" block is 32 bits, thus the "Packet Sender" block need some control logic to adjust the fetched packet cells to be sent out according to the data path width if it is smaller than 32 bits. However, when the data path width is increased to 32 bits, the output data width adjusting logic is not needed in the "Packet Sender" block. The initiator type "Node IF" has larger area than the target type because it needs a buffer to store the header cell of received packets for supporting split-transaction feature in the BVCI standard.

V. Comparing With a PTP NoC

In order to examine the characteristics and performance of the CDMA NoC thoroughly, a simulation network that applies the CDMA NoC scheme is built and compared with a PTP NoC presented in [19].

A. Simulation Network Setup

The simulation network that applies the CDMA NoC is illustrated in Fig. 10. It contains six network nodes which work in different clock frequencies as illustrated in Fig. 10. The BVCI interface standard is applied in the network. Three hosts act as initiators and the other three act as targets, as denoted by the labels "I" and "T," respectively, in the "Network IF" blocks. The initiator hosts can generate requests to any target hosts, while the target hosts can generate responses only for the received requests passively. The network nodes are connected to each other through "CDMA Transmitter" and "Network Arbiter" blocks. The spreading codes used in the network are six 8-bit Walsh codes. The basic data unit transferred in the network is data packets composed by one header cell and several data cells as illustrated in Fig. 11. The number of data cells in a packet varies from one to three, while the width of each packet cell is fixed at 32 bits. The "functional host" blocks and their "Network IF" blocks are not realized with any real IP blocks; they are simulated by adding stimulus signals on each "Network Node" block according to the BVCI standard. A four-phase dual-rail handshake protocol is applied in the CDMA network to transfer data between network nodes. The PTP network illustrated in Fig. 12 has the same mentioned network configurations as the CDMA network except that the network nodes in the PTP network are connected with each other through bidirectional ring topology. Therefore, the characteristics of the CDMA NoC can be examined more clearly by comparing the two networks in different aspects in the following four subsections.

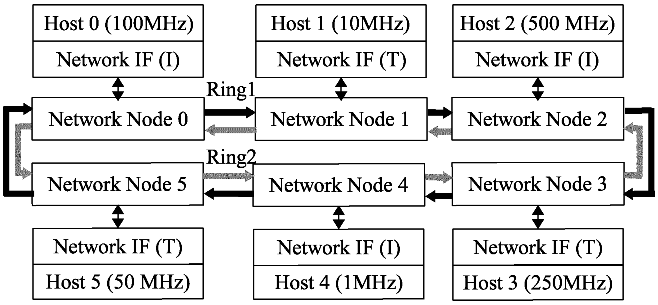

Fig. 12. Six-node PTP NoC simulation network.

B. Comparison of Data Transfer Principles

In the PTP connected network illustrated in Fig. 12, the data traffic load is distributed into the links among the network nodes. This distributed traffic scheme has the benefits of flexibility and scalability, whereas the main disadvantage is that the data transfer latency between two network nodes can be largely different when data are transferred to different destinations or to the same destination via different routes.

Although data transfers in the PTP network can be parallel if they take place in different links among the network nodes, concurrent data transfers over a single link is impossible in the PTP NoC because a link between two network nodes is shared in a time-division manner. Therefore, by applying CDMA technique, the main advantage of the CDMA NoC is the feature of concurrent data transfers. Hence, the data transfer latency in the CDMA NoC is a constant value which in turn helps 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 would increase the traffic load in the PTP network, or complicate the network implementation. One more benefit of applying the CDMA NoC is that the header cell in a packet needs not to be transferred in the network after a sending node gets the grant signal from the "Network Arbiter" since the receiving node already knew the sender information through the A-T protocol presented in Section II-C. However, in the PTP NoC, the header cell in a packet needs to be transferred in the network for packet routing.

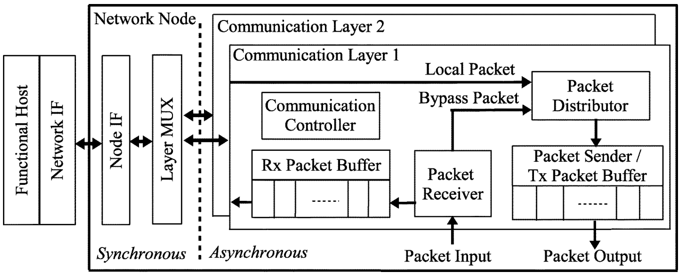

Fig. 13. Network node structure of the PTP NoC.

TABLE II
DATA TRANSACTION SPECIFICATION

Initiator Node	Target Node	Number of Transactions	Packet Length	
			Request Packet	Response Packet
Node 0	Node 1	2	4, 3	2, 3
	Node 3	2	3, 4	3, 2
	Node 5	1	4	2
Node 2	Node 1	1	4	2
	Node 3	1	3	3
	Node 5	2	4, 3	2, 3
Node 4	Node 1	2	3, 4	3, 2
	Node 3	1	3	3
	Node 5	1	4	2

C. Comparison of Network Node Structures

The network node structure of the PTP NoC presented in [19] is illustrated in Fig. 13. It contains two same "Communication Layer" blocks for supporting the bidirectional ring topology. By comparing with the network node illustrated in Fig. 6, the network node of 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 its one-hop data transfer scheme. Therefore, the "Communication Controller" and "Packet Distributor" blocks illustrated in Fig. 13 are not needed in the node of the CDMA NoC. Since the CDMA NoC applies centralized traffic scheme, its network node does not need multiple "Communication Layer" blocks and "Layer MUX" block in the node of the PTP NoC illustrated in Fig. 13. 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 structure in the CDMA NoC does not need to change in this situation because of the parallel data transfer scheme.

D. Comparison of Data Transfer Latencies

The CDMA network illustrated in Fig. 10 and the PTP NoC illustrated in Fig. 12 are both synthesized using the same 0.18-μm technology library. Gate-level simulations are performed on both simulation networks. The data transactions performed during the simulations are listed in Table II. Each data transaction consists of one request packet from an initiator host to a target host and one corresponding response packet from the target host to the initiator host.

Because the GALS scheme is applied both in the CDMA network and the PTP network, the data transfer latency in the two simulation networks can be separated into two parts, 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. The measured STL values of the CDMA network are listed in Table III. The constant values in Table III 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. From Table III, we can see that an initiator type of network node takes more clock cycles for local data transfers. The reason is that the initiator node needs to store or read the header cell to or from a buffer as mentioned in Section IV-C. Since the same "Node IF" block design is applied in both simulation networks, the STL of the PTP network has the same value as listed in Table III.

TABLE III
SYNCHRONOUS TRANSFER LATENCY

Node Type	Latency of sending data to 'Network Node'	Latency of receiving data from 'Network Node'
BVCI Initiator	8 local clock cycles + 2.5 ns	8 local clock cycles + 3.2 ns
BVCI Target	4 local clock cycles + 2.5 ns	4 local clock cycles + 3.1 ns

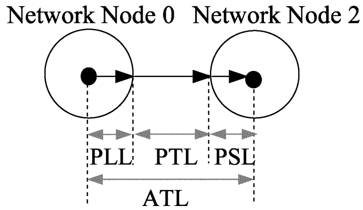

Fig. 14. ATL portions of the CDMA NoC.

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

1) ATL in the CDMA NoC: The ATL of the CDMA network consists of three portions: packet loading latency (PLL), packet transfer latency (PTL), and packet storing latency (PSL). The concept of those ATL portions is illustrated in Fig. 14 with an example where "Network Node 0" sends one data packet to "Network Node 2." The black arrows in Fig. 14 represent the packet transfer direction. The different portions of ATL are marked by grey arrows in Fig. 14 and explained in the following three paragraphs.

a) *PLL*: This is the time used by the "Packet Sender" block to fetch a data packet from "Tx Packet Buffer" and prepare to send the packet to "CDMA Transmitter."

b) *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 Transmitter" and "Network Arbiter" blocks using a handshake protocol.

c) *PSL*: After the receiver node receives a data packet, it will spend a certain amount of time to store the received

TABLE IV
ATL PORTION VALUES OF THE CDMA NOC

		1 data cell	2 data cells	3 data cells
PLL (ns)		5.7	5.7	5.7
PTL (ns)	*1-bit*	384.6	768.9	1153.7
	8-bit	45.9	88.4	130.9
	16-bit	26.2	49.0	71.8
	32-bit	14.7	26.0	37.8
PSL (ns)		5.5	5.5	5.5

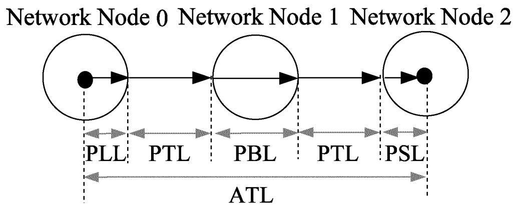

Fig. 15. ATL portions of the PTP NoC.

TABLE V
ATL PORTION VALUES OF THE PTP NOC

Packet Length	PLL (ns)	PTL (ns)	PBL (ns)	PSL (ns)
1 data cell	11.7	13.4	10.7	3.3
2 data cells	15.2	18.7	14.2	3.3
3 data cells	18.6	24.0	17.6	3.3

data packet into "Rx Packet Buffer." This time duration is measured as PSL.

The measured values of ATL portions of the CDMA NoC under different data path configurations are listed in Table IV. The ATL value of the CDMA NoC can be calculated by directly adding the three portions under the same configuration.

2) ATL in the PTP NoC: The concept of the ATL portions of the PTP NoC is illustrated in Fig. 15 with an example that "Network Node 0" sends one packet to "Network Node 2" via "Network Node 1." The black and grey arrows in Fig. 15 represent the same meanings as the arrows in Fig. 14. The meaning of ATL portions will be explained briefly in the following four paragraphs.

a) *PLL*: It is the time used to load one "local packet" into "Tx packet buffer" in the "Packet Sender" block as illustrated in Fig. 13.

b) *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.

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

c) *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.

d) *PSL*: It is the time spent on storing one "incoming packet" into "Rx Packet Buffer" block.

The ATL portion values of the PTP NoC are listed in Table V. The formula of calculating the ATL of transferring one packet in the PTP NoC is given in (1). 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.

TABLE VI
EQUIVALENT NUMBER OF INTERMEDIATE NODES IN THE PTP NOC

	1-bit	8-bit	16-bit	32-bit
1 data cell	N=15.2	N=1.2	N=0.4	N= - 0.1
2 data cells	N= 22.6	N=1.9	N=0.7	N=0.0
3 data cells	N= 26.9	N=2.3	N=0.9	N=0.1

3) Comparing the ATL Values: In Tables IV and V, we can see that PTL values of the CDMA NoC and the PTP NoC increases as the packet length increases. This is because the data cells in a packet are sent in a serial manner in the two networks. Thus, more data cells need more transmission time. Whereas, the PLL and PSL values of the CDMA NoC and the PTP NoC are nearly not affected by the packet length. The reason is that the data cells in a packet are loaded or stored in a parallel manner in both networks.

The main difference between the ATL values of the two NoCs is that the ATL value of the CDMA NoC is a constant 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 portion 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 providing guaranteed service in the network.

The PTL values listed in Table IV show that the data path width configuration affects the ATL of the CDMA NoC in a linear manner. For instance, the PTL value of transferring a three-data-cell packet is reduced around 30 times when the data path width is increased from 1 to 32 bits. Since the data path width in the PTP network illustrated in Fig. 12 is realized as 32 bits, only the ATL value of the CDMA NoC with 32-bit data path width is comparable with the ATL value of the PTP NoC. However, in order to compare the data transfer latency characteristics of the two NoCs thoroughly, Table VI lists the equivalent number of intermediate network nodes which would be gone through by a data packet in the PTP NoC when the same size packet is transferred in the CDMA NoC under different data path configurations. From Table VI, we can see that when the data path widths in the CDMA NoC and the PTP NoC are both 32 bits, the ATL of delivering a two-data-cell packet in the CDMA NoC is equivalent to transferring the same packet between two adjacent network nodes in the PTP NoC, which means that the ATL of the CDMA NoC equals to the best case ATL value in the PTP NoC. When transferring a one-data-cell packet, the ATL in the CDMA NoC is even smaller than the best case ATL in the PTP NoC as denoted by the negative value of N in Table VI. The latency caused by the data encoding and decoding scheme in the CDMA NoC is compensated by its one-hop data transfer scheme. Hence, the CDMA NoC can transfer data packets with the equivalent best case ATL of the PTP NoC when the data path width is set to 32 bits.

TABLE VII
AREA AND POWER COSTS OF THE TWO NETWORKS

NoC Type	Data Path Width	Six-Node Simulation Network		
		Area (K equivalent gates)	Dynamic Power (mw)	Energy Cost of Delivering 32 bits (pJ)
CDMA NoC	*1-bit*	113.145	19.340	12.5168
	8-bit	148.369	6.563	3.7428
	16-bit	191.037	7.331	4.0868
	32-bit	272.806	7.332	4.0873
PTP NoC	*32-bit*	177.007	7.324	4.7401

E. Comparison of Area and Power Costs

The two simulation networks illustrated in Figs. 10 and 12 are synthesized using a 0.18-μm technology library. The area costs of the two simulation networks with different data path widths are listed in Table VII for comparison purpose. According to the data transactions performed in the gate level simulations and listed in Table II, the dynamic power costs during simulations and the energy costs of transferring 32 data bits in the CDMA NoC and the PTP NoC are also listed in Table VII.

From the figures in Table VII, we can see that when the data path width is increased from 1 to 32 bits in the CDMA NoC, the area cost of the CDMA network becomes 2.4 times larger because more logic are used to perform parallel data encoding and decoding. With 16- and 32-bit data path widths, the CDMA NoC loses its area cost advantage by comparing with the PTP NoC.

In terms of the dynamic power costs listed in Table VII, a 1-bit CDMA NoC should not be applied due to the much larger power consumption by comparing with the CDMA NoCs under other data path width configurations and the PTP NoC. The reason of the large power cost of the 1-bit CDMA NoC is that the 1-bit CDMA NoC needs much more switching activities than the other versions of the CDMA NoC due to the over-serialized data transfer scheme. To be noticed in Table VII is that the 16- and 32-bit CDMA NoCs have almost the same power consumptions. The reason is that the power consumption increase caused by the data path width increasing is compensated by reducing the control logic for data output adjust operations in each "Packet Sender" in the 32-bit CDMA NoC as explained in Section IV-C. By comparing the dynamic power costs and the energy costs of transferring 32 bits in Table VII, we can see that the PTP NoC has similar dynamic power cost with the 16- and 32-bit CDMA NoCs, while the energy figures are slightly larger than the figures in the two CDMA NoCs. This is because the PTP NoC takes more time to perform the data transactions listed in Table II due to its multiple hop data routing scheme. However, the CDMA NoC can perform the same data transactions with shorter time since its one-hop concurrent data transfer scheme. Therefore, the average energy spent on transferring 32 data bits in the CDMA NoC, except the 1-bit CDMA NoC, is smaller than the energy cost in the PTP NoC.

VI. CONCLUSION

An on-chip packet switched communication network that applies the CDMA technique and supports the GALS communication scheme was presented. The presented CDMA NoC uses an asynchronous scheme to perform the global data transfers between network nodes, and uses synchronous scheme to deal with the local data transfers between a functional host and the network node attached to it. A CDMA encoding and decoding scheme which suits digital-circuit implementation was presented. The main advantage of the presented CDMA NoC is that it can perform data transfer concurrently by applying CDMA technique in the network. Therefore, the large data transfer latency variance caused by the packet routing in a PTP NoC is eliminated in the CDMA NoC. The constant data transfer latency in the CDMA NoC is helpful for providing guaranteed communication services to an on-chip system. Another advantage of the CDMA NoC is that it can perform multicast data transfers easily by utilizing the multiple access feature of CDMA technique.

Both the asynchronous and synchronous circuits of the CDMA NoC with different data path widths are realized in RTL using VHDL in order to suit the conventional synchronous design flow and tools. Two six-node on-chip networks were constructed to compare the CDMA NoC with a PTP NoC. One network applies the CDMA NoC, while the other applies a bidirectional ring PTP NoC. The two networks were simulated and compared against each other. The simulation results reveal that when the data path width of the two simulation networks is set to 32 bits, the asynchronous transfer latency in the CDMA NoC is equivalent to the best case data transfer latency in the PTP NoC. The best case data transfer in the PTP NoC means that packets are transferred between two adjacent nodes. It indicates that the data transfers between any network nodes in the CDMA NoC can be performed as quickly as transferring the same data packets between two adjacent nodes in the PTP NoC.

By considering the tradeoff between transfer latency performance listed in Table VI and the costs listed in Table VII, a 16-bit CDMA NoC is a good option for replacing the PTP NoC in an on-chip system where universal data transfer latency is a desired requirement. With a 16-bit data path width, the data transfer latency of the CDMA NoC is close to the best case transfer latency in the PTP NoC while the area and dynamic power costs remain similar. If the area and power costs have higher priority, the 8-bit CDMA NoC can be applied because its area is 16.2% smaller than the PTP NoC while its energy cost of transferring 32 bits is 21.0% smaller than the cost in the PTP NoC.

REFERENCES

[1] D. Wiklund and D. Liu, "SoCBUS: Switched network on chip for hard real time systems," in *Proc. Int. Parallel Distrib. Process. Symp. (IPDPS)*, 2003, p. 8.

[2] K. Goossens, J. Dielissen, and A. Radulescu, "Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Des. Test Comput.*, vol. 22, no. 5, pp. 414–421, Sep./Oct. 2005.

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

[4] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communications*. Reading, MA: Addison-Wesley, 1995.

[5] R. Yoshimura, T. B. Keat, T. Ogawa, S. Hatanaka, T. Matsuoka, and K. Taniguchi, "DS-CDMA wired bus with simple interconnection topology for parallel processing system LSIs," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, 2000, pp. 370–371.

[6] T. B. Keat, R. Yoshimura, T. Matsuoka, and K. Taniguchi, "A novel dynamically programmable arithmetic array using code division multiple access bus," in *Proc. 8th IEEE Int. Conf. Electron., Circuits Syst.*, 2001, pp. 913–916.

[7] S. Shimizu, T. Matsuoka, and K. Taniguchi, "Parallel bus systems using code-division multiple access technique," in *Proc. Int. Symp. Circuits Syst.*, 2003, pp. 240–243.

[8] M. Takahashi, T. B. Keat, H. Iwamura, T. Matsuoka, and K. Taniguchi, "A study of robustness and coupling-noise immunity on simultaneous data transfer CDMA bus interface," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2002, pp. 611–614.

[9] R. H. Bell, Jr., K. Y. Chang, L. John, and E. E. Swartzlander, Jr., "CDMA as a multiprocessor interconnect strategy," in *Conf. Record 35th Asilomar Conf. Signals, Syst. Comput.*, 2001, pp. 1246–1250.

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

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

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

[13] D. M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, Dept. Comput. Sci., Stanford University, Stanford, CA, 1984.

[14] VSI Alliance, Wakefield, MA, "Virtual component interface standard version 2," 2001. [Online]. Available: http://www.vsi.org/

[15] OCP-International Partnership. Beaverton, OR, "Open core protocol specification," 2001. [Online]. Available: http://www.ocpip.org/

[16] X. Wang and J. Nurmi, "A RTL asynchronous FIFO design using modified micropipeline," in *Proc. 10th Biennial Baltic Electron. Conf. (BEC)*, 2006, pp. 1–4.

[17] I. E. Sutherland, "Micropipelines," *Commun. ACM*, vol. 32, no. 6, pp. 720–738, 1989.

[18] X. Wang, T. Ahonen, and J. Nurmi, "Prototyping a globally asynchronous locally synchronous network-on-chip on a conventional FPGA device using synchronous design tools," in *Proc. Int. Conf. Field Program. Logic Appl.*, 2006, pp. 1–6.

[19] X. Wang, D. Sigüenza-Tortosa, T. Ahonen, and J. Nurmi, "Asynchronous network node design for network-on-chip," in *Proc. Int. Symp. Signal, Circuits, Syst.*, 2005, pp. 55–58.

Xin Wang received the M.Sc. degree in electronic circuits and systems from Northwestern Polytechnical University, Xi'an, China in 2002.

Currently, he is a Researcher with the Institute of Digital and Computer Systems, Tampere University of Technology, Tampere, Finland. His research interests are focused on on-chip communication networks and asynchronous circuits design.

Tapani Ahonen received the M.Sc. degree in electrical engineering and the Ph.D. degree in information technology from Tampere University of Technology, Tampere, Finland.

He is a Senior Research Scientist with Tampere University of Technology. His research interests are focused on varying aspects of system-on-chip design.

Jari Nurmi is received the Ph.D. degree from Tampere University of Technology (TUT), Tampere, Finland, in 1994.

He is a Professor of digital and computer systems with TUT. He has held various research, education, and management positions at TUT and in the industry since 1987. His current research interests include system-on-chip integration, on-chip communication, embedded and application-specific processor architectures, and circuit implementations of digital communication, positioning, and DSP systems. He is leading a group of about 25 researchers at TUT. He is the author or coauthor of about 160 international papers, the editor of *Processor Design: System-on-Chip Computing for ASICs and FPGAs* (Springer, 2007), coeditor of *Interconnect-Centric Design For Advanced SoC and NoC* (Kluwer, 2004), and has supervised more than 90 M.Sc., Licentiate, and Doctoral theses.

Dr. Nurmi is currently the general chairman of the annual International Symposium on System-on-Chip (SoC) and of its predecessor SoC Seminar in Tampere since 1999 and a board member of SoC, FPL, and NORCHIP conference series. He was the head of the national TELESOC graduate school 2001–2005. He is a senior member in the IEEE Signal Processing Society, the Circuits and Systems Society, the Computer Society, the Solid-State Circuits Society, and the Communications Society. In 2004, he was a corecipient of the Nokia Educational Award, the recipient of the Tampere Congress Award in 2005, and the Academy of Finland Senior Scientist research grant for the academic year 2007–2008.