A Dynamic CDMA Network for Multicore Systems

Basel Halak (bh9@ecs.soton.ac.uk), Teng Ma , and Ximeng Wei

EEE Group, School of Electronic and Computer Engineering, University of Southampton, United Kingdom

***Abstract*- CDMA (code-division multiple-access) is a data transmission method based on the spreading code technology, where in multiple data streams share the same physical medium with no interference. A novel architecture for on-chip communication networks based on this approach is devised. The proposed design allows sharing coding resources among network's users through the use of dynamic assignment of spreading codes. Data transmission latency is reduced by adopting a parallel structure for the coding/decoding circuitry. A 14-node CDMA network based on the proposed architecture is synthesised using 65nm ST technology library. Performance analysis reveals that the proposed approach achieves significantly lower data packet latency compared to both conventional CDMA and packet switched network-on-chip implementations. Large area and power savings compared to existing approaches are also obtained.**

***Index Terms*:** Code-division multiple-access, latency, multicore.

I. INTRODUCTION

Computer performance has been mainly driven by decreasing the size of devices and increasing their number. Recently, power thermal issues- such as dissipating heat from increasingly densely packed transistors- have begun to limit the rate at which processor frequency can be increased.

Chip performance increased 60% per year in the 1990s but slowed to 40 % per year from 2000 to 2004, than performance growth rate dropped to less than 20 % per year [1]. Such incremental increase in performance simply cannot support emerging applications such as data mining across teraflops of data; artificial intelligence (AI) for smarter cars and appliances and virtual reality (VR) for modelling, visualization [2, 3].

In order to meet such performance requirements for these fast growing novel applications; Semiconductor industry needs to develop speed-enhanced systems architectures in coordination with the continuous scaling of silicon manufacturing process towards 18 nm technologies and beyond. In other words, achieving the performance goals for future applications requires innovations in both device technology and system architecture. Multicore architecture is one of the promising solutions, it uses several processing elements (PEs) to concurrently execute programs instructions, which helps improve the overall performance

by handling more work in parallel [2, 4-6]. However, the ability of multicore architecture to increase applications performance critically depends on its interconnection structure (i.e. how the processor cores are physically connected on chip)[3]. In conventional interconnection solutions such as split-transaction buses, only one master at a time can drive the bus. Despite its low complexity, the main drawback of this solution is its lack of scalability, which will result in unacceptable performance degradation (e.g., contention-related delays for bus accesses) when the level of system integration will exceed a dozen of cores. [7]. The use of network-on-chip (NoC) structure has been proposed to address the communication requirements of such large multicore chips. An on-chip network consists of multiple interconnected switches, on which packetized communication takes place. Cores access the network by means of proper interfaces, and have their packets forwarded to destination through a multihop routing path [8-10]. NoCs is an attractive solution as it allows the scalability of communication architecture; however, recent work by Intel has indicated that the implementation cost of networks routers can be prohibitive in terms of power and area [11]. Moreover, in packet switched NoCs that apply multihop point-to -point connection scheme as in [9] the packet transfer latency varies largely depending on the destinations of the packet and on its route through the network. Therefore, the upper bound of the packet transfer latency is determined by the worst case scenario.

One innovative solution to reduce the variance of data packet latency is to employ code-division multiple access (CDMA) technique to reduce communication latency, such technique can also help remove the need for implementing complex costly routing algorithms [12-15].

CDMA is a spread spectrum technique which encodes information prior to transmission onto a communication medium, permitting simultaneous use of the physical medium by separate data streams [16]. The implementation of CDMA techniques for on-chip communication requires the use of orthogonal spreading codes. Each host of the network (e.g. a processing element) is allocated a unique spreading code in order to distinguish traffic for different hosts. The number of hosts that can be connected to a CDMA network depends on the length of the spreading code, for example only seven hosts can be connected to the network if an 8-bit Walsh code is used as the spreading code. One of the earliest work on the use of CDMA for multicores systems is in [12], the authors have illustrated that this technique can be used to eliminate contention and queuing delays on shared buses; however no circuit implementation was presented in this paper. More recent work in [7] has demonstrated that the use of CDMA method improves the performance of multicore systems by reducing bus contention interferences and by supporting higher concurrency in memory accesses, which brings shorter critical word access latency, but the authors of [7] have not estimated the overheads of the CDMA technique, besides they have only considered dual core systems. Wang et al in [14] have shown how the CDMA concept can be realised at the circuit level, they have also demonstrated that a CDMA-based network can achieve a superior performance

over those of a multihop NoC; however their results have been limited to only point-to-point networks, and they did not consider more efficient topologies such as mesh and tree folded torus.

In summary, previous work has shown that the gains achieved using the CDMA technique for on-chip interconnection schemes are high throughput communication, reduced latency and predictable performance. However, there remain a number of issues that need to be addressed *before* the CDMA method can be adopted. First, in all previous implementations of CDMA-based networks, all hosts are assigned fixed spreading codes [12-14], this means the larger the number of processors in a multicore system the longer the spreading code needed to be used. In practice, increasing the length of the spreading code can lead to a dramatic increase in area overheads and network latency, this is due to the fact that longer spreading codes require larger CDMA coding/decoding logic (i.e. more area) and longer data transfer delay (i.e. more latency) [14]. This may render the use of CDMA networks impractical for a system which has tens of processing elements such as those in [2, 4-6].

The second issue that need to be addressed is the delay overheads of the CDMA coding/decoding logic, for example, if 16-bit spreading code is used, it will take 16 clock cycles to encode one data item, this can significantly increase the data transfer latency. The first contribution of this work is the development of a novel CDMA communication protocol that allows for dynamic assignments of spreading codes, which makes it feasible to use short spreading codes for systems with large number of cores. This allows the use of CDMA-based networks in modern multicores systems. This work also devises a new architecture for CDMA digital coding/decoding circuitry, which drastically reduces their delay overheads, hence enhancing overall system performance.

The proposed CDMA scheme has synthesised using ST 65nm technology library. Analysis Results reveal that the proposed design achieves a significant improvement in performance compared to conventional CDMA schemes and packet-switched networks, power and area savings can also be obtained in a number of cases. The organisation of this paper is as follows. Section 2 reviews the CDMA principles and illustrates how a parallel coding/decoding logic can be realised. Section 3 outlines a novel communication protocol for CDMA networks that allows the sharing of coding resources. Section 4 illustrates the proposed structure for implementation a dynamic CDMA network, and explains its operation principles. Section 5 discusses in depth the architecture and functionality of network components. Section 6 explains the metrics used for performance estimation. Section 7 evaluates the latency and throughput of the proposed network in comparison with existing solutions and provide in –depth analysis of area and energy costs. Finally, conclusions are drawn in Section 8.

II. CDMA PRINCIPLES

A. CDMA Transmission Principles

The principle of the CDMA technique is illustrated in figure 1. At the transmitter end, orthogonal spreading codes are used to encode multiple data streams. The orthogonal property of spreading codes allows the encoded data from different senders to be combined together in one data stream without interfering with each other's. The orthogonal property means that the normalized autocorrelation value of spreading codes is "1" and their cross-correlation value is "0". Cross-correlation of spreading codes refers to the sum of the products of two different spreading codes, while autocorrelation refers to the sum of the products of a spreading code with itself.

At the receiver end, individual data streams can be regenerated from the received sum signal by multiplying the received signals with the unique spreading code used for encoding each stream. Several types of spreading codes have been proposed for CDMA networks, such as Walsh code, M-sequence, Gold sequence, and Kasami sequence [16]. Walsh codes have been adopted by a number of researcher [7, 13, 14] for their efficient digital implementation of CDMA networks. Such efficiency is attributed to their balance properties. The balance property means that the number of bit '1' and bit '0' in a spreading code should be equal. Walsh codes are also used in this work. In a K-bit ($K = 2^N, integer\ N > 1$) length Walsh code set, there are ($K - 1$) sequences that have both the orthogonal and balance properties. This means conventional CDMA-based networks can connect at most ($K - 1$) hosts as each host is assigned with a fixed code as shown in figure 2. The length increase of the spreading code leads to an increase in the implementation overheads, this renders conventional CDMA-based schemes prohibitively expensive. To differentiate such schemes from the proposed solution we call them *static CDMA architecture* (static means code assignment is fixed). This work proposes a new dynamic *CDMA architecture* which allows sharing of coding resources, this means codes are assigned to the communicating hosts on a temporal basis i.e. only during data transmission. Therefore in the proposed scheme, the spreading code length does not impose any limit on the number of hosts that can be connected to a CDMA network. This will allow the use of CDMA techniques for large networks at a reduced costs.

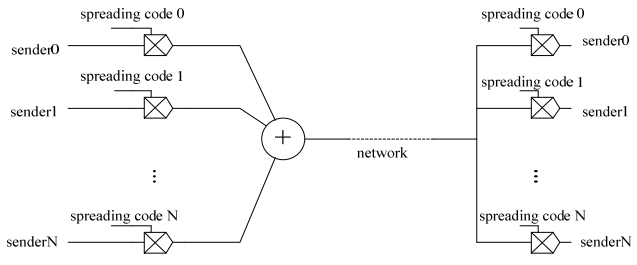


Figure 1: CDMA Transmission Principle [15]

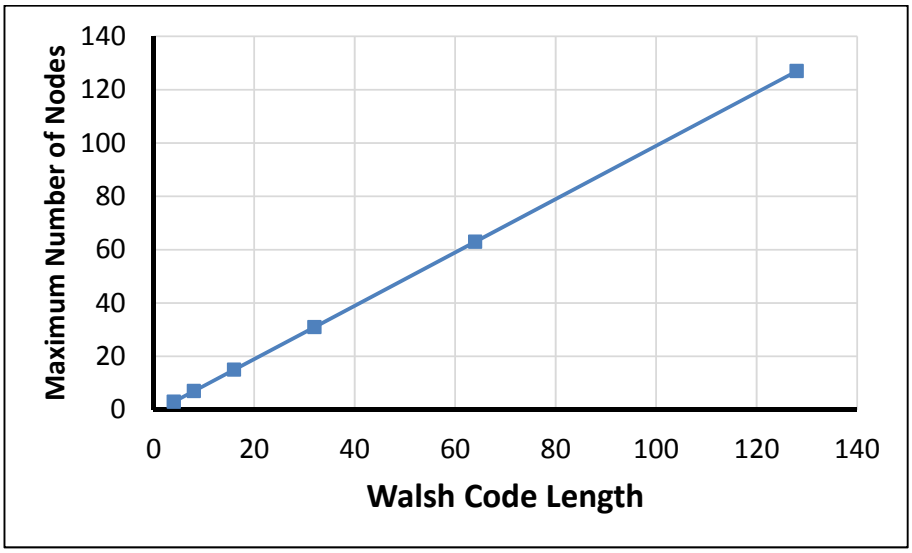


Figure 2: Relationship between the Spreading Code Length and the Maximum Size of Conventional CDMA Network

B. Digital Implementation of Walsh Code based-CDMA

B.1. Serial Implementation

This implementation is based on [12, 14]. In the encoding scheme illustrated in figure 3, data from multiple transmitters are applied to the encoder bit by bit. Each bit is spread into K bits using XOR operation with a unique K-bit spreading code. Each bit of the K-bit encoded data produced by the XOR operations is called a data chip. Then, the data chips coming from all transmitters are summed together arithmetically according to their bit positions in the K-bit sequences. In other words, all the first data chips from different transmitters are added together and all the second data chips from different transmitters are added together, and so on. Consequently, after the add operations, we get K sum values of K-bit encoded data. Finally, binary equivalents of the K sum values are transmitted to the receiver. The decoding process shown in figure 4 calculates the sum of the positive part and the negative part of the received data chips respectively. Each bit value in the spreading code defines the type of the accumulators for each data chip, for example, when spreading code "01010101" is used, the first sum of data chips belongs to the positive part and the second sum of data chip is the negative part and so on. After all the sums of data chips have been classified and added together separately, the comparison will be carried out to generate the decoded value. If the sum of the positive part is larger than that of the negative part, the output data is "1"; otherwise, the output data is "0."

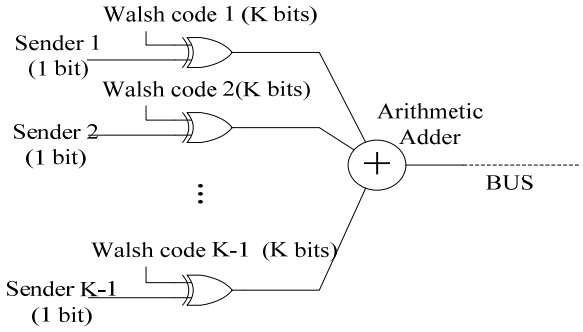


Figure 3: CDMA Digital Encoding [14]

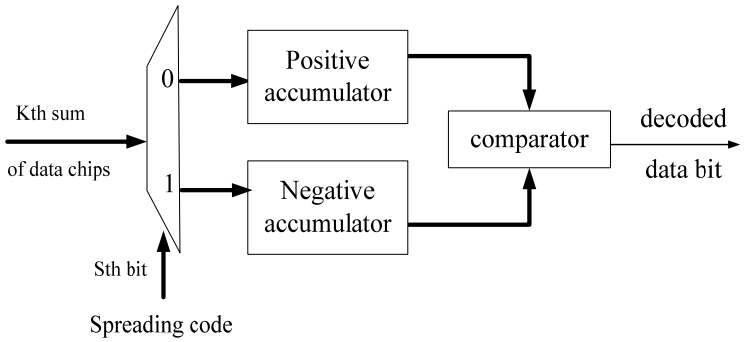


Figure 4: CDMA Digital Decoding [14]

B.2. Proposed Parallel Implementation

The previous serial implementation has a major latency drawback. This is because each data item is spread into K data chips (K is the length of Walsh code), then data chips are transmitted sequentially. So, for an 8-bit Walsh code, 8 clock cycles are needed to encode one bit. This poses a serious performance limitation on the CDMA-based communication architectures which could prevent its use for high speed applications. In order to overcome this problem, we propose a new parallel coding/decoding architecture. Figure 5 shows the implementation of the parallel CDMA coding circuits for an 8-bit Walsh code. As can be seen all data chips are produced and transmitted simultaneously, which means only one clock cycle is needed to transfer one bit data from a sender to a receiver. The decoding is also done in parallel as shown in figure 6.

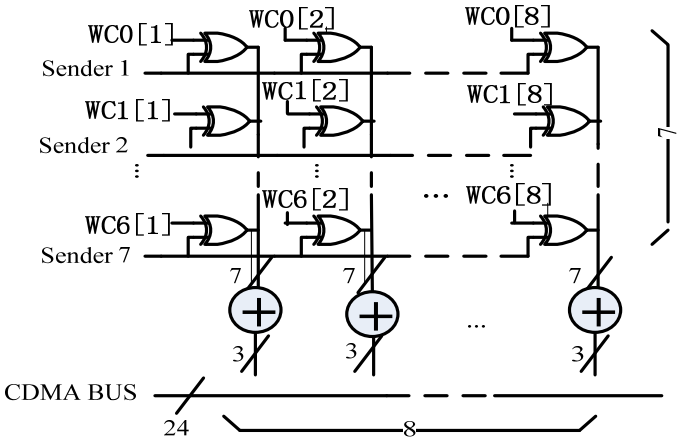


Figure 5: Parallel CDMA Encoding

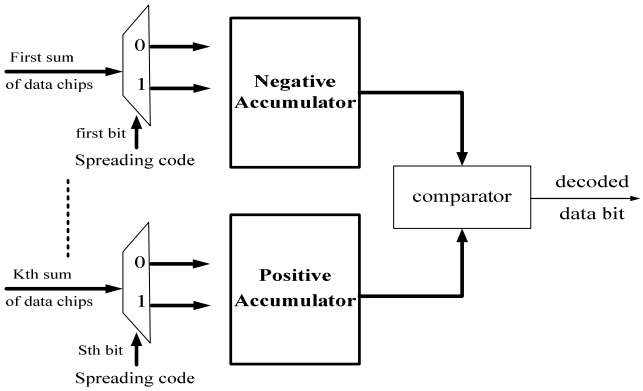


Figure 6: Parallel CDMA Decoding

III. CDMA COMMUNICATION PROTOCOLS

A. Conventional (Static) CDMA Architecture

A conflict in CDMA communication architecture can be caused if multiple transmitters employ the same spreading code to encode their data for simultaneous transmission, which causes data interference because of the loss of orthogonal property among the spreading codes. *Spreading code protocol may be defined as the policy employed to decide how to allocate and utilise the spreading codes resources in order to eradicate or minimise potential conflicts during the communication processes in a CDMA network*. There are a number of CDMA protocols that have been proposed for on-chip based CDMA networks [7, 14]. Some protocols are transmitter based such as *(T protocol)* where in a unique spreading code allocated to each user is used by the user himself to transfer data to others. Other protocols are receiver-based (*R protocol*) where in each user in the network is assigned a

unique spreading code used by the other users who want to send data to that user, other variations also exist. To the best of our knowledge all existing CDMA protocols for on-chip CDMA networks are based on a fixed assignment of spreading codes to the network's users (i.e. static), which means that the maximum number of hosts that can be connected to such networks is determined by the length of the spreading code being used. The larger the network, the longer the spreading code needed. The increase of spreading code length can have serious impact on the area costs and power consumption of a CDMA network.

B. Dynamic CDMA Architecture

This work devises a new CDMA protocol based on the idea of dynamic temporal assignments of spreading codes; it is called (*D protocol*). When a user wants to transfer data in a CDMA network which uses the (*D protocol*), he sends a request to the network arbiter which includes the destination address. After the arbiter establishes the availability of the intended recipient, it assigns a unique spreading code for this data transfer. Then, the arbiter sends a grant signal to the sender along with the allocated spreading code to be used for data encoding. The arbiter also informs the receiver to prepare the corresponding spreading code for data decoding. After the data transmission is completed, the arbiter frees the allocated spreading code so that it can be used by other sender-receiver pairs.

If there are two or more users wanting to transmit data to the same receiver, the arbiter will grant only one sender to send data at a time. Static priority algorithm is used to arbitrate among users in this case. The number of data transmission requests that can be granted at any one time depends on the number of available spreading codes. For an 8-bits Walsh codes, the arbiter can grant up to seven data transmission requests simultaneously provided they are for different destinations. If the number of transmission requests at a particular time is larger than the available spreading codes, then the arbiter will employ a static priority algorithm to choose which requests to grant. Static priority is this work means that each user is assigned a fixed priority level. For a 14 node network there are 14 different levels, with level 0 has the highest priority and 13 the lowest. It should be noted that other arbitration algorithm can also be used in conjunction with the (*D protocol*).

IV. Network Architecture and Operation Principles

A. Network Architecture

The proposed (*D protocol)* is employed to design a 14-node dynamic CDMA network architecture, where in 8-bit Walsh code is used. The network has three main components network interface (NI), CDMA transmitter and an arbiter block.

The overall architecture of a multicore processor based on this network is shown in Figure 7.

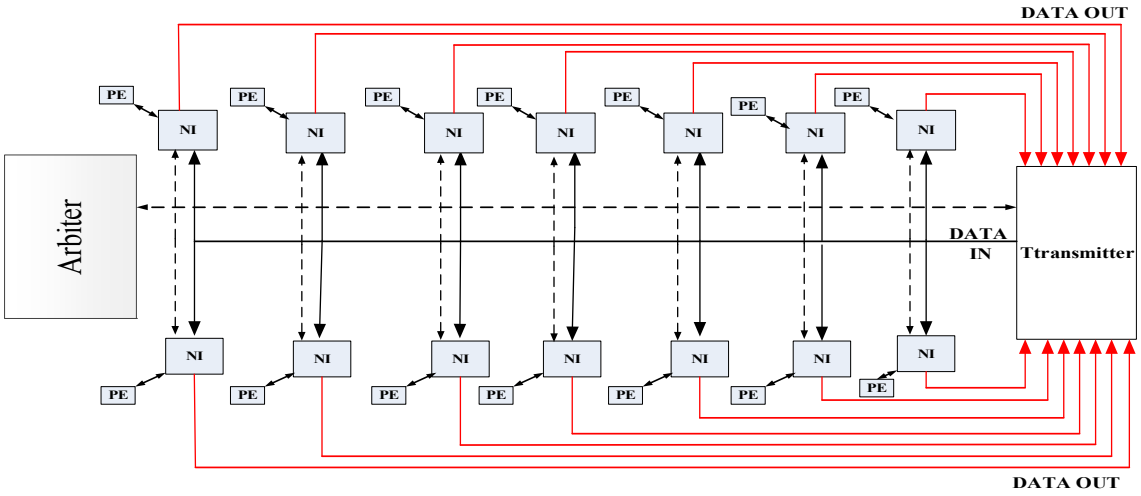


Figure 7: Dynamic Serial CDMA Network

The communication through the network is packet based, each packet consists of a packet header (16 bits) and a payload which can vary from 16 to 240 bits as shown in figure 8 and 9. The packets are formed at the processing elements and transmitted flit by flit to the network interface, each flit is 16 bit long. The header flit (the same as the packet header) includes the source address and the size of the packet in flits. The remaining flits contain only data load. It should be noted here the address of the destination is transmitted separately to the arbiter as will be seen later.

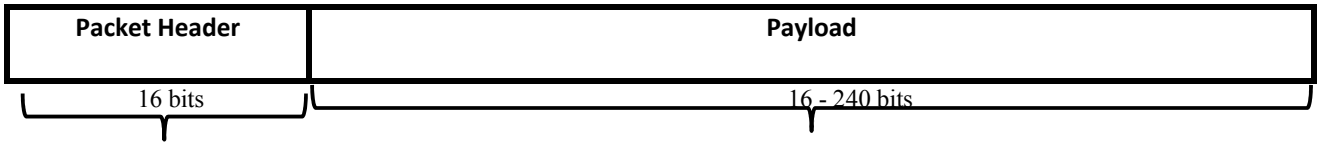


Figure 8: CDMA Network Packet Format

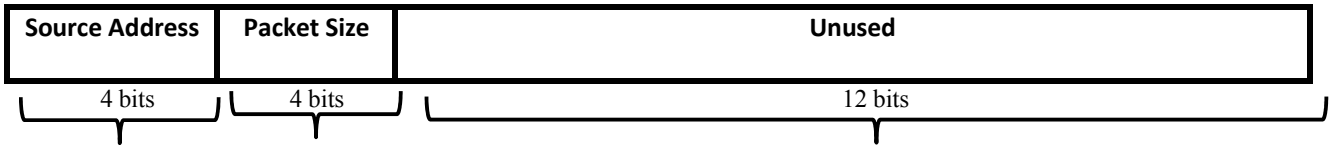


Figure 9: CDMA Network Packet Header Format

B. Operation Principles

The basic operation steps of one transmission cycle are as follows:

a) A processing element (PE's) send a request and a destination address through the network interface to the arbiter asking for a permission to transmit data to a particular node.

b) The arbiter checks the availability of the intended recipient, assign a unique CDMA code for each granted request and inform both the transmitter and the network interfaces of the communicating processing elements with these codes. (The arbiter can grant up to seven transmission requests simultaneously in this case).

c) After receiving a grant signal, the processing element proceed with the transmission,

d) The network interface of the sender receives data packets from the processing element and transforms these into 16 bits flits, which are released to (DATA OUT) bus (as shown in figure 7) to be encoded by the transmitter.

e) The transmitter uses the allocated spreading codes to combine up to seven stream of information into one encoded data stream.

f) The encoded data stream is then transmitted on the DATA IN bus.

g) Each intended recipient uses its allocated spreading code to extract their information from the encoded data stream.

h) Each allocated spreading code remain reserve till the corresponding sender completes its transmission

i) Finally the arbiter frees coding resources, this conclude one transmission cycle.

Details on the operation of each block are provided in the following section.

V. Operation Principles of Network Components

A. Network Interface

The network interface consists of several components, namely: interface control, decoder and FIFO buffers, as shown in Figure 10. Details on the functionality of these blocks are provided below

The FIFO blocks provide extra space for data storage of the incoming and outgoing flits, each element can store up to four flits.

The interface control manages the interaction between the network hosts and the network arbiter and transmitter. It also controls the storage elements

The decoder block decodes incoming CDMA data; Figure 11 illustrates the architecture of the decoder for a CDMA network with dynamic arbitration and serial coding/decoding circuits.

The code pool is designed as a controller of the decoder. It receives the decoding enable signal "EN" from the interface block and the codeword number from the arbiter directly. For decoding the serial data, it outputs the spreading code bit by bit to the relevant

components every clock cycle and gives a final "judge" signal to the comparator for producing the decoded data. The network interface block allows simultaneous transmission and reception from and by its functional host.

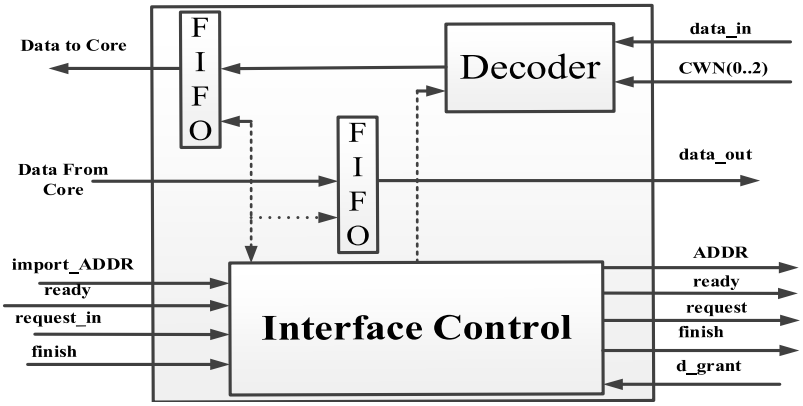


Figure 10: Dynamic CDMA Network Interface

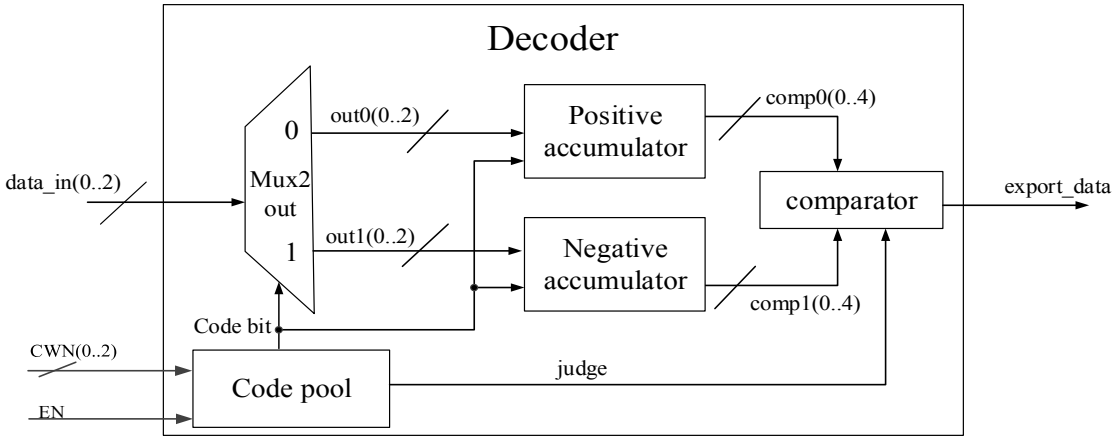


Figure 11: Decoder Architecture

B. Arbiter design

The arbiter is the core of the dynamic CDMA network, it has two main functions:

1. It controls dynamic assignments of spreading codes.
2. It resolves any potential conflict arising when two nodes send transmission requests to the same destination.

The block diagram of the arbiter for a 14-node CDMA network based on 8-bits Walsh Code is shown in figure 12. The "request" signals are received from their corresponding network nodes; they indicate a request for data transmission. Each request signal is accompanied with a 4-bit destination address (signal AADR). The arbiter grants sender access to the bus by sending them an encoding-enable signal (en_chan) along with their allocated spreading code (signal CWN). Each spreading code is assigned a

code number as outlined in Table I, this helps reduce wiring overheads as each code needs only three wires to be represented as opposed to 8 wires needed for 8-bit Walsh codes.

Table I: Walsh Codes Numbers

Code number	0	1	2	3	4	5	6
Spreading code	01010101	00110011	01100110	00001111	01011010	00111100	01101001

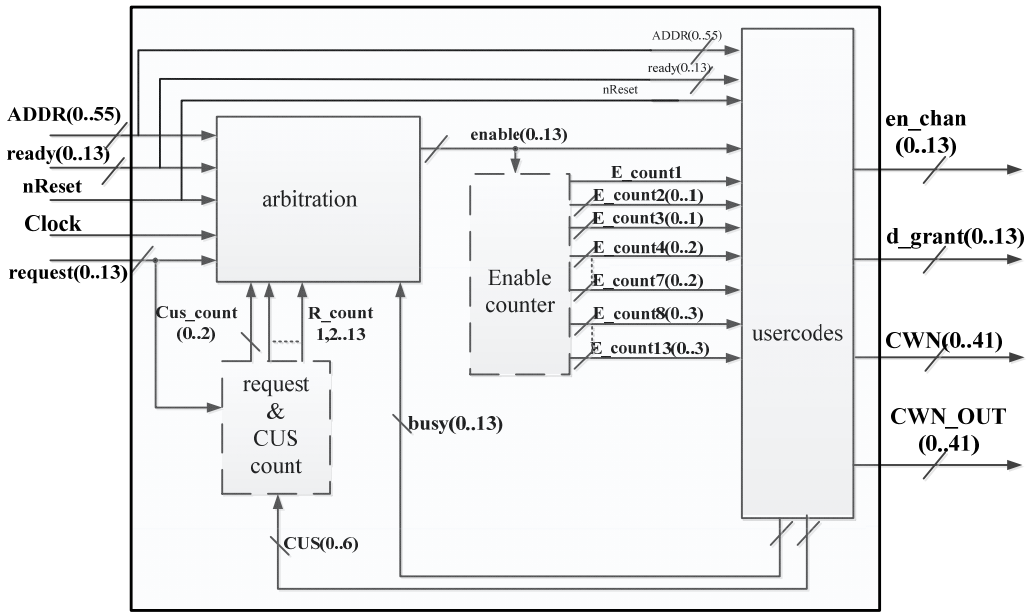


Figure 12: Dynamic CDMA Network Arbiter

After a transmission request is made, the arbiter first confirms the availability of the intended recipient to receive data through checking the corresponding flag "busy". Then, the arbiter sends a decoding grant signal "d_grant" to the intended recipient along with the allocated spreading code "CWN_OUT" to allow CDMA decoding.

Note that CWN_OUT uses the same code numbers as shown in Table I. "ready" signals indicate that data transmission is completed. The detailed structure of the arbiter block is shown in figure 12; it consists of two main sub-blocks: arbitration and usercodes.

1. Arbitration block

The function of this module is to control the dynamic assignment of spreading codes; and to resolve conflicts arising from simultaneous requests to the same destination. This block has external input signal shown in Figure 12, in addition it takes as inputs three types of internal control signals. The first is a 14-bit "busy" flag, where in each bit represents the status of a particular receiver node in the systems. The second type of internal control signals is "Cus_{count}" which represents the number of spreading

codes being used at any moment in time. The "Cus_{count}" is a three bit signal equal to the sum of all the bits of the state flag CUS[0:6] which indicates the status of each spreading codes, so if there are 4 spreading codes being used, Cus_{count} will be (100). The third type of internal control signals is the "R_{count}" flags. There are 14 "R_{count}" flags, each corresponds to a particular network host. Each flag shows the current number of network access requests from the nodes which hold higher priority than the request by the host under consideration. This flag is constantly updated by the arbiter according to the following equation:

$$\boldsymbol{R_{COUNT}[i] = \sum_{j=0}^{i-1} request[j]} \quad \textbf{(1)}$$

Where i: is the priority level of the requesting host.

For example for a user with priority level 3, the value of its corresponding flag ($R_{count}[3]$) is between 0 and 3. Similarly $R_{count}[7]$ can have a value between 0 and 7.

The arbiter grants a request if all the following conditions are satisfied:

1) The intended recipient is free to receive data
2) There are no requests with higher priority wanting to transmit to the same destination
3) There are sufficient spreading codes.

The requests are assigned spreading codes based on their static priority level.

The number of spreading codes (M) available for a particular request in an 8-bit Walsh code dynamic CDMA network is calculated as follows.

$$\boldsymbol{M[i] = 7 - (Cus_{COUNT} + R_{COUNT}[i])} \quad \textbf{(2)}$$

Where:

i: the priority level of the requesting node.

Cus_{COUNT}: the number of spreading codes already in use.

$R_{COUNT}[i]$: the number of higher priority requests (see equation 1).

The operation principles of this module can be explained as follows. Upon receiving one or more "request" signals, the arbitration module first checks the status of the intended recipients through the "busy" signal. If the destination node is free, the arbiter checks for any conflicting requests, if more than one users want to send data to the same destination, only the-highest-priority request can be allowed. Thirdly, the arbiter checks whether or not it has sufficient spreading codes, there should be at least one available

spreading code ($M[i] \geq 1$).When all arbitration conditions are met for a particular request, the arbitration module triggers the corresponding "enable" signal. This causes the "usercodes" module to assign a spreading code to the granted request.

2. The "usercodes" module

This module has three functions

1. It is responsible for assigning spreading codes for granted requests as indicated by the corresponding "enable" signals.
2. It updates the "CUS" flag which indicates which spreading code is being used.
3. It updates the "busy" flag which indicates which network nodes currently un-able to receive data.

The operation principles are as follows. Assuming user 3 was granted by the arbitration block to send data to user 7. The *enable* signal corresponding to user 3 (i.e. with a priority level 3) will be triggered. Then the "usercodes" module will check the content of the flag "$E_{COUNT}[3]$", the latter flag indicates the number of spreading codes to be assigned to higher priority requests. The E_{COUNT} flags are calculated from the enable signals as follows:

$$\boldsymbol{E_{COUNT}[i] = \sum_{j=0}^{i-1} enable[j]} \quad \textbf{(3)}$$

So $E_{COUNT}[3]$ could have a vlaue between 0 and 3. Assuming $E_{COUNT}[3]$ was found to be 2, the "usercodes" will assign a spreading code number 3 according to Table I. Then, the "usercodes" will send both user 3 and user 7 the number of the allocated spreading code (i.e. 00001111). In addition, the "usercodes" will set the bit corresponding to user 3 in the "en_chan" and the bit corresponding to user 7 in the "d_grant" to logic 1. The "usercodes" will also set the bit in the "CUS" flag which corresponds to the allocated spreading code to value "1" (i.e. CUS[3]="1"). In addition, The "usercodes" will set the value of the busy flag for user 7 to 1 (i.e. busy[7]= "1"). When data transmission is completed, the sender node activates its ready signal which allows the arbiter to release all resources allocated to this transmission and to set the all flags related to this transmission to a zero value.

C. Dynamic CDMA Transmitter

This block receives data packets from network nodes and encodes them with the corresponding unique spreading code of the sender according to the principles explained in section 2. The data encoding and transfer processes for different network nodes are performed in parallel and independently. The transmitter can accept data from up-to seven nodes simultaneously. In the dynamic CDMA network, the number of connected nodes is larger than the number of available spreading codes, which may cause interference on the CDMA bus if careful measures are not taken. To prevent such scenario, only the nodes which are granted access by the arbiter are allowed to send their data to the arithmetic summer. To achieve this, multiplexers are used between the

xor encoders and the arithmetic summer as shown in Figure 13. These multiplexers are controlled by the *en_chan* signal produced by the arbiter. The block diagram of the dynamic serial transmitter is shown in Figure 13.

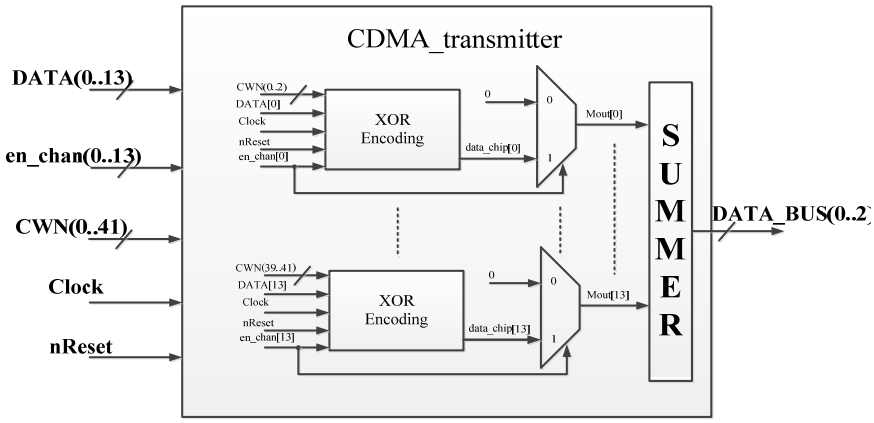


Figure 13: Dynamic CDMA Network Transmitter

VI. PERFORMANCE METRICS

To compare the proposed dynamic CDMA architectures with existing communication, a set of standard performance metrics are employed [17]. It is vital for a multi-core interconnect architecture to have high throughput, low latency, energy efficiency, and low area overhead. The following sub-sections describe these metric in more details.

A. Data packet Latency (DPL)

For a fair comparison with packet-based networks on chips, data packet latency is going to be employed. Latency is defined as the time that elapses between the moment a request is made to the network at the source core and the arrival of the last flit to the destination core.

DPL consists of three portions:

1. **Arbiter Setup (A_{SETUP})**: is the time needed for the arbiter to setup a transmission link between a sender and a receiver after receiving a request assuming all resources are available. A_{SETUP} is in all cases one clock cycle per one packet transmission.
2. **Network transfer delay (*ND*)** is the time needed to transfer one data packet through the network after the link has been setup, including coding/decoding overheads. *ND* increases with the length of the spreading codes and the number of transmitted data items. It can be reduced by increasing the width of the data bus and using parallel coding/decoding logic.

3. **Queuing delay (*Q*)**: is the additional time the data packet has to wait before the transmission can start after a transmission request is made to the arbiter. This delay may be incurred if there are simultaneous requests to the same destination or if there are not enough spreading codes in the case of dynamic CDMA. The average value of the queuing delay depends on the network traffic which is in turn application dependent.

B. Network Throughput (NT)

Throughput can be defined in a variety of different ways depending on the specifics of the implementation. For message passing systems, we can define message throughput, as follows:

$$\boldsymbol{NT = \frac{Total\ Number\ of\ Messages * message\ Length}{Total\ Time}} \quad \textbf{(4)}$$

Where *Total Number of Messages* refers to the number of data packets that successfully arrive at their destination cores, *Message length* is measured in bits, and *Total Time* is the time (in clock cycles) that elapses between the occurrence of the first message generation and the last message reception. Throughput is measured in bits/clock cycle, for example a throughput NT = 256, corresponds to 256 bit per clock cycle.

VII. Implementation and Simulations Results

A. Implementation

In order to compare the overheads of proposed network with existing communication schemes, a 14 node system is considered the following architectures have been designed:

1. Parallel Dynamic network: this implements dynamic assignment of spreading codes and uses parallel coding/decoding circuitry as explained in sections IV and V. It employs 8 bit Walsh codes
2. Serial Dynamic network: This implements dynamic assignment of spreading codes, but it uses serial coding/decoding circuitry. It employs 8 bit Walsh codes
3. Conventional CDMA network: This implements conventional fixed assignment of spreading codes such as the network in [14] and uses serial coding/decoding circuitry, therefore this scheme must use 16 bits Walsh codes to be able to connect 14 nodes [16]. This leads to increased latency and overheads as will be seen later. The overall architecture of this network is the same as shown in figurer 7, however, the arbiter design is simpler as it does not need to handle dynamic assignment of spreading code, in addition, the transmitter block does not need to use multiplexer as the number of spreading code is equal to the number of network nodes

4. Mesh _based network on-chip which implements wormhole switching and source routing algorithm based on [15], each node in this network is connected to a router through an interface. The communication is packet based. The length of the data packet is up to 256 bits as in the CDMA proposed architecture. Data packets are generated at the network hosts and transformed into flits at the interface blocks, the flits are then transmitted to their destination through the networks routers.

   (Details on the architecture of these blocks are provided in appendix 1)

5. Split Transection Bus: this is the conventional crossbar architecture where in only one node have access to the shared interconnection at any moment in time. The bus has two main components, namely the arbiter block and the network interface. The arbiter employs static priority arbitration algorithm, the network interface is similar to that shown in section IV, but it does not have a decoder block.

All above architecture have been realized in *System Verilog* and synthesised using ST 65nm technology library. Functional verifications have been conducted through digital simulation in Modelsim, they include transmission and reception of data packets between different hosts.

B. Analysis of Network Latency Results

Data packet Latency has been estimated through digital simulations in Modelsim by injecting a data packet on a sender node and measuring the time it takes for the data to appear on the receiving end. A number of experiments have been conducted to compare the performance of different architectures.

The purpose of the first experiment was to estimate the overall network latency as a function of injection load, the latter represents the number of packets injected into the network simultaneously per IP, an injection load of 1 means all IP blocks are injecting data packet simultaneously, one packet each.

Network latency is estimated as the total time it takes for all data packets to arrive at their respective destinations. The results from figure 14 show that dynamic parallel CDMA scheme (DP) achieves the lowest latency, followed by the Mesh-based architecture (NOC). This is mainly due to the fact that both networks allow for concurrent access to the communication medium. Other CDMA architecture (dynamic serial (DS) and conventional (SS)) have very large latencies even compared to a conventional crossbar architecture (BUS), this due to the serial nature f their respective CDMA encoding circuitry.

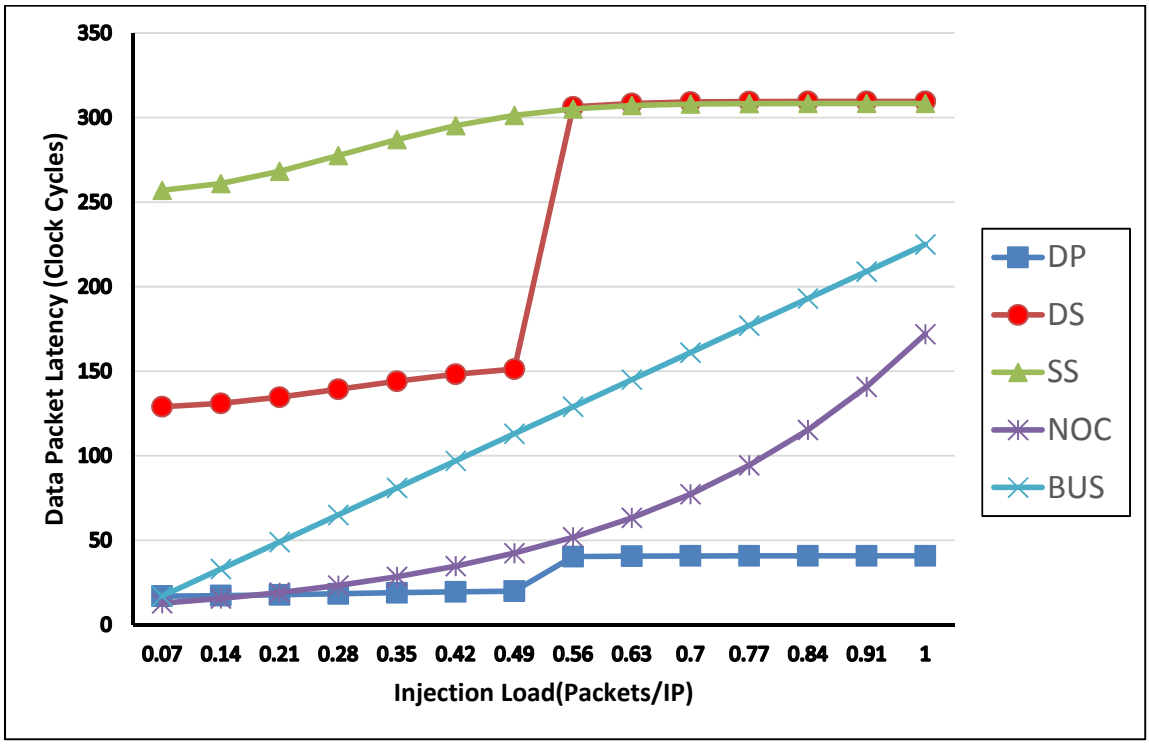


Figure 14: Network Latency vs. Injection Load

The purpose of the second experiment is to estimate the average data packet latency for each architecture under generic traffic. A generic case has been considered, where in; all network nodes have the same probability of transmission, and each node can send data to all other nodes with equal probabilities. Based on these assumptions, a random traffic stimulus have been generated and applied to each network in Modelsim simulator. The mean and standard deviation of the data packet latency (DPL) have been estimated in each case. In addition another metric called DPL variation is used for comparison, it is calculated as percentage of standard deviation to the mean latency value, this metric is used to estimate the performance predictability of each architecture. A number of lessons can be learned from the results summarised in Table II.

First, the application of dynamic assignment (SD) can lead in average to a reduction of 50% in data packet latency compared to conventional the CDMA network , which is mainly due to the use of shorter Walsh spreading code (8 bits instead of 16 bits).

Second, the use of parallel coding in combination with dynamic code assignment achieves the lowest data packet latency compared to both to conventional the CDMA network and packet-switched Mesh-based NoC topologies under consideration.

Third, by observing DPL variation results, it can be seen that the CDMA architectures have the most predictable performance compared to NOC and BUS, This is a very important advantage over existing methods for applications which require a guaranteed service level. It is also noted that the proposed dynamic CDMA architectures (PD and SD) have more variations than conventional CDMA scheme (SS) due to the sharing of spreading codes among different nodes.

Table II: Data Packet Latency under Generic Traffic

Design	DPL mean(μ)	DPL Standard Deviations(σ)	DPL Variation (σ/ μ)
Serial Dynamic CDMA (SD)	162	21	12%
Parallel Dynamic CDMA (PD)	22	2.7	12%
Serial Conventional CDMA [14](SS)	298	11	3%
Mesh-Based (NOC)[15]	64	20	30%
Spilt Transection Bus(BUS)	113	42	37%

C. *Analysis of Network Throughput Results*

Another important metric of a communication schemes is the throughput, this has been estimated according to equation (4) figure 15 illustrated the network throughput as a function of injection load, the latter represents the number of packet injected into the network simultaneously, an injection load of 1 means all IP block are injecting data packet simultaneously, one packet each. A number of observations can be made from the results depicted in figure 15.

First it can be seen that the highest throughput can be achieved by the proposed parallel dynamic CDMA for injection load more than 20%. This is due to the inherent parallel architecture of this scheme.

Second, there is a drop of the achievable throughput for the parallel dynamic architecture when the injection load exceeds 50%, this due to the fact that this scheme only allows 7 simultaneous transmission, therefore, when there 8 nodes requesting transmissions, than one of the nodes has to wait till the transmission of the first seven packet is completed, this queuing delay adds to the overall network latency.

Third, For Mesh-based network, throughput seems to reach a maximum point of around 0.35 packets/clock cycle then it drops afterwards, this is due to the fact that injection load directly affects the average message latency as shown in figure 14. As the injection load increases there will be more message contention and congestions, hence packet latency will increase, which lead to a drop of the overall network throughput.

Finally, for the remaining architectures under consideration, poor network throughput is observed due to their inherent serial nature.

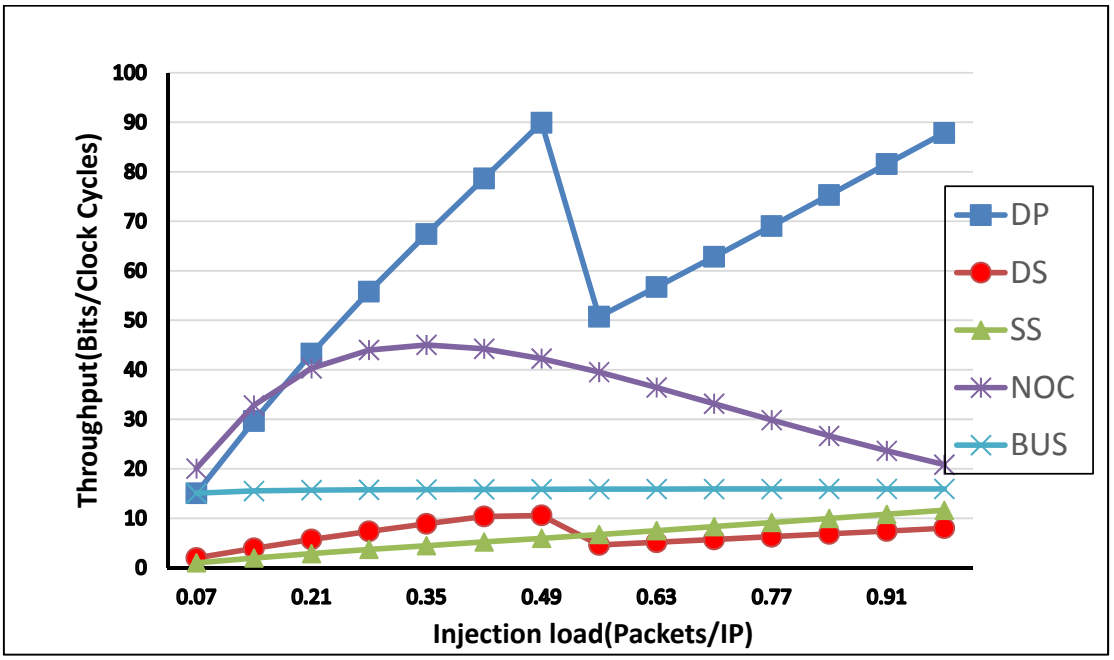


Figure 15: Network Throughput vs. Injection Load

D. Area and Energy Cost Analysis

To estimate area and energy costs, the architectures under consideration are synthesized using a ST 65nm technology library, a clock frequency of 1 GHz has been used in all cases, in addition individual components for all architectures have been synthesised separately to estimate their areas. Area results are shown in Table III

Energy estimation methodology involved the generation of switching activity information for each block through digital simulation in Modelsim, these data have then been fed to Synopsys Design Compiler to obtain energy figures.

First the energy dissipations of the individual components of all architectures have been estimated, second, the energy incurred by transferring one data packet through each communication architecture is measured. Energy results are shown in Table IV.

First the area overhead results are examined from table III,

It can be noted that the use of serial dynamic architecture (SD) lead to a 22% reduction in area overheads compared to conventional serial static CDMA networks (SS), this is mainly due to the fact that dynamic assignment of spreading codes allow the use of 8 bit length spreading code to connect 14 nodes as opposed to the 16 bits length code needed by the conventional scheme. The use of a shorter spreading code decreases the number of CDMA coding/decoding circuitry needed, hence area savings. The second observations is that the use of parallel CDMA coding/decoding circuitry as in the parallel dynamic CDMA network (PD) leads to a significant increase in area overheads compared to serial CDMA schemes (SS and SD), however , the parallel

architecture (PD) has less network latency and higher throughput compared to the serial schemes as discusses above. Therefore , a trade-off should be made in this case between area costs and performance.

The results also show that the Mesh-based architecture (NoC) incurs a large area overheads, which is mainly attributed to the complex architecture of the router block(as shown in appendix 1), which require several space consuming data buffers, in the case under consideration , each router has five input buffers , and five output buffers. In general, the more the data buffers the less the congestion and the better the performance for networks on chips, however this enhanced performance comes at a heavy area price.

In comparison with CDMA architectures, NoC has comparable area overheads to that of parallel dynamic scheme (PD).

Finally, the results have shown the conventional crossbar scheme (BUS) consume the least overhead, this is expected due to the simplicity of its architectures.

Second the energy costs results are examined from table IV,

First, it can be noted that although individual components of both the (SS) and the (SD) architectures have comparable energy costs, the (SD) architecture have a 25% lower packet transfer energy cost compared to conventional CDMA network (SS), this is mainly due to the fact that dynamic CDMA scheme use a shorter spreading code, which means it incurs less data transections to transfer the same number of bits. Second, a comparison between packet energy cost of (SD) and (PD) schemes reveals that parallelizing the coding/decoding circuits has a relatively small impact on packet energy costs (around 10%), this is because increasing the parallelism of the CDMA architecture does not change the number of coding/decoding operarions which have to be done per each packet transfer. The small increase in energy cost in this case is attributed to the additional buffering requirements of parallel dynamic scheme (PD).

Finally, it is noted that all CDMA architectures (SS,SD and PD) incur higher packet transfer energy cost than Mesh network (NoC) and shared bus(BUS), this is attributed to the inherent requirement of a CDMA transmission which requires each data item to be spread (i.e. transformed into multiple bits) before it can be transmitted, therefore, more data transections, hence more energy dissipation. For application with high speed requirement, this rise in energy costs can be justified by the large performance improvement that can be achieved as evident from figures 14 & 15.

Table III: Area Costs of the Communication Architectures

Design	Area Overheads (µm2)				
	Router	Arbiter	Transmitter	Network Interface	Network Area
Serial Dynamic CDMA(SD)		19989	39088	6850	157038
Parallel Dynamic CDMA(PD)		19989	70920	20770	377916
Serial Static CDMA (SS)[14]		14210	22385	11722	200703
Mesh-Based Network on Chips (NOC)	23168	n/a	n/a	3037	365464
Spilt Transection (BUS)	n/a	7420	n/a	2305	39690

Table IV: Energy Costs of the Communication Architectures

Design	Energy Dissipation (pJ)				
	Router	Arbiter	Transmitter	Interface	Cost of Transferring One Data Packet (256 bit)
Serial Dynamic CDMA(SD)		3.01	15.8	3.5	24.01
Parallel Dynamic CDMA (PD)		3.01	28.1	8.3	27.1
Serial Static CDMA(SS) [14]		3.5	14.6	7.5	32.5
Mesh Network on Chip(NoC)	11.4	n/a	n/a	3.2	23.4
Spilt Transection (BUS)		1.1	n/a	1.5	3.9

VIII. CONCLUSIONS

The proliferation of high speed multicore systems has led to growing demands for innovative communication architectures that can satisfy the performance requirements of such systems. The use of *code-division multiple-access* (CDMA) networks allows the transmission of multiple data stream on the same physical interconnects, this is a promising solution for connecting high speed systems. The potential of this technique has already been demonstrated for system with small number of nodes (e.g. dual core processor systems, 6 nodes system-on-chip). To reap the benefits of CDMA technique in systems with large number of nodes (such as many core systems and/or systems-on-chips); efficient digital implementation must be found that reduces area, power and latency costs.

This work has developed a novel design for on chip CDMA networks, which allows the sharing of coding resources among network's users through the use of dynamic assignment of spreading codes, the proposed approach also reduces transmission latency by enhancing coding/decoding parallelism. Several versions of a 14 node network have been implemented using ST 65nm technology. The analysis results have indicated a reduction in the data packet transmission latency and network throughput can be achieved compared to conventional CDMA and packet switched architectures. Overheads estimation has illustrated that the use of dynamic codes assignment can lead to large reduction in area cost and power consumption 22% and 25% respectively compared to conventional CDMA networks at the price of increased delay variations. In Comparison with a packet-switched NoC, the proposed parallel dynamic CDMA scheme has similar area and energy costs, but better throughput and performance predictability. Overall, the proposed scheme will make it feasible to develop high throughput low latency CDMA-based networks for many-core systems.

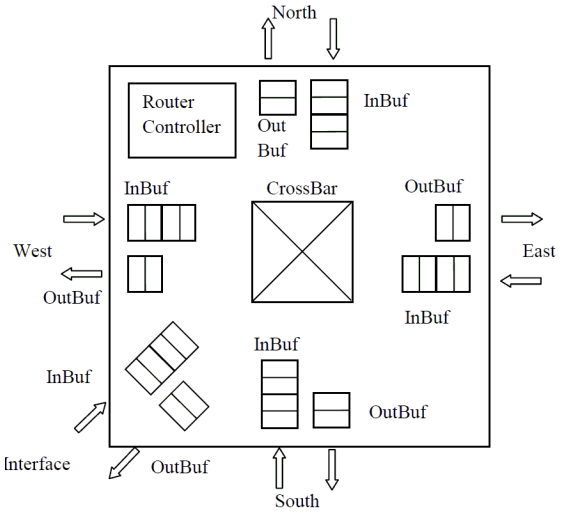
IX. REFERENCES

1. International Technology Roadmap for Semiconductors (www.itrs.net).
2. Geer, D., Chip makers turn to multicore processors. Computer, 2005. 38(5): p. 11-13.
3. J. Held, J.B., and S. Koehl, From a few cores to many: A tera-scale computing research overview. Intel White Paper, 2006.
4. Kalla, R., S. Balaram, and J.M. Tendler, IBM Power5 chip: a dual-core multithreaded processor. Micro, IEEE, 2004.
5. Howard, J., et al., A 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS for Performance and Power Scaling. Solid-State Circuits, IEEE Journal of, 2011. 46(1): p. 173-183.
6. Vangal, S.R., et al., An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS. Solid-State Circuits, IEEE Journal of, 2008. 43(1): p. 29-41.
7. Jongsun, K., et al., A Cost-Effective Latency-Aware Memory Bus for Symmetric Multiprocessor Systems. Computers, IEEE Transactions on, 2008. 57(12).
8. Bertozzi, D. and L. Benini, Xpipes: a network-on-chip architecture for gigascale systems-on-chip. Circuits and Systems Magazine, IEEE, 2004. 4(2): p. 18-31.
9. Goossens, K., J. Dielissen, and A. Radulescu, AEthereal network on chip: concepts, architectures, and implementations. Design & Test of Computers, IEEE, 2005. 22(5): p. 414-421.
10. Partha Pratim, P., et al., Performance evaluation and design trade-offs for network-on-chip interconnect architectures. Computers, IEEE Transactions on, 2005. 54(8).
11. Shekhar Borkar, I.C., Future of Interconnect Fabric - A Contrarian View. System Level Interconnect Prediction(SLIP) Workshop, 2010.
12. Bell, R.H., Jr., et al. CDMA as a multiprocessor interconnect strategy. in Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on. 2001.
13. Woojoon, L. and G.E. Sobelman. Mesh-star Hybrid NoC architecture with CDMA switch. in IEEE International Symposium on Circuits and Systems 2009.

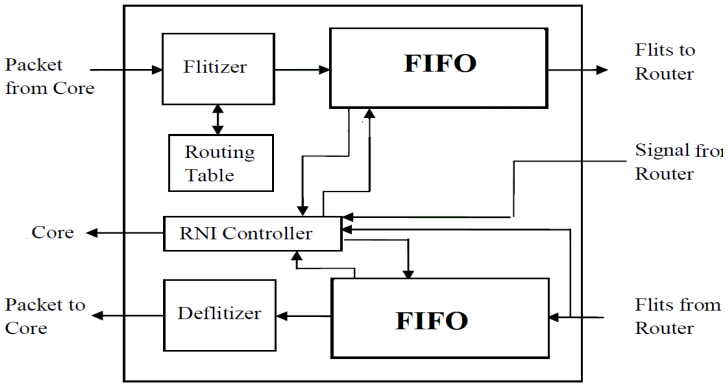
14. Xin, W., A. Tapani, and N. Jari, Applying CDMA Technique to Network-on-Chip. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2007. 15(10): p. 1091-1100.
15. Mubeen, S. and S. Kumar. Designing Efficient Source Routing for Mesh Topology Network on Chip Platforms. in Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on. 2010.
16. Dinan, E.H. and B. Jabbari, Spreading codes for direct sequence CDMA and wideband CDMA cellular networks. Communications Magazine, IEEE, 1998. 36(9): p. 48-54.
17. Patterson, J.H.a.D., Computer Architecture: A Quantitative Approach. , ed. M. Kaufmann. 2003.

X. APPENDIX 1

The figures below show the architectures of the router and the interface of a network on chip which implement source routing algorithm and has a Mesh topology. More details on the operation details of these block can be found in [15]



Figurer 16: Router Architecture



Figurer 17: Network Interface Architecture