An NoC architecture for real time SoC applications with modified CDMA scheme

TALLA VAMSHI¹, T.SATYASAVITRI²
¹Department of ECE, Talla Padmavathi College of Engineering, Kvizet, Warangal, Andhra Pradesh, India
²Department of ECE, JNTUH College of Engineering, Kukatpally, Hyderabad, Andhra Pradesh, India
¹vamshi22g@gmail.com, ²tirumalasaty@gmail.com

ABSTRACT
The complexity of System-On-Chip (SoC) design is increasing continuously due to the multidimensional optimization requirements, while integrating complex intellectual property (IP) blocks. The interconnectivity topologies between IPs are playing a vital role in deciding the performance of the SoCs. This paper investigates the existing code division multiple access (CDMA) based network on chip (NoC) architectures. The work presented here explains a variant of CDMA based NoC scheme, which is best suitable at the base band level for dynamic bandwidth management. A six node globally-asynchronous locally-synchronous (GALS) type NoC is realized at RTL level, with two different controller architectures. Both vary in terms of key management and control mechanism. The scheduler-built-in-ring type architecture, with its ease in placement and routing is suitable for complex SoCs. The real time characteristics of proposed architectures are investigated. The architectures are implemented in VHDL and verified at simulation level. It is verified that data transfer latencies for a given pair of nodes remain constant for given bandwidth. The Xilinx FPGA synthesis results promise more than 200 MHz clock speeds resulting in 1.6 Gbps data throughput over 32 bit ring bus on Virtex-6 LX series FPGAs.

Keywords: NoC; CDMA; SoC; Walsh codes; Crossbar; DSSS; IP interconnection; Dynamic bus management; Real time NOC;

[1] INTRODUCTION
The complexity of System-On-Chip (SoC) design is increasing continuously due to the multidimensional optimization requirements, while integrating complex intellectual property (IP) blocks. The continuously evolving multimedia algorithms and wireless protocols demanding silicon level revisions of SoCs during their lifecycle. The present day IP based design paradigm is influencing the total silicon chain. The industry wise thrust to achieve uniformity in IP interfaces and interconnectivity is the fact demonstrating the importance of interface issues among IP blocks.

Even though several bus architectures are evolved to address the interconnectivity aspects, majority of them are only variants of conventional master-slave bus systems. The present day architectures suffer from following disadvantages.

- Not designed to exploit high speed serial bus standards, which are available today.
- To handle simple to complex IPs, a common interface specification is not possible.
- VLSI routing aspects and related issues are not considered by several bus specifications.
- Conventional address and data bus topologies leads lot of wiring in chips.
- Dynamic configuration of bus bandwidths based on run time IP demands is not considered.
- Most of the bus specifications are designed for convenience at the protocol layer, not visualizing the overheads at physical layer level.

The historical improvisation from monolithic [8] bus-based interconnect architectures to hierarchical system integration using multiple smaller buses connected through repeaters or bridges does not yield a general and scalable solution. The global timing improvisation [8] needs complete redesign of the SoC interconnect architectures. In this paradigm the research on NoC architectures with completely new design approaches assumes great importance.

The concept of using CDMA as a multiplexing scheme is one of the major research area for realizing NoC architecture. Among the two broad categories of circuit switched and packet switched types, CDMA based NoC fundamentally falls into packet switched type, also carrying several advantages of circuit switched type networks. This paper presents two architectures to meet requirements of real time NOC, using CDMA technique. The section II elaborates the CDMA based NOC in detail. The section III gives details of proposed two architectures and explains their execution flow. This section also gives qualitative comparison among two architectures with respect to, several key NOC parameters. The section IV has simulation and synthesis results. The section V has details of timing estimation and comparison among two architectures. The section VI concludes the work.
CDMA BASED NOC FOR REAL TIME APPLICATIONS

All NoCs can be broadly divided into two categories: circuit switched type and packet switched type. The SoCBUS architecture [10] is an example of circuit switched type with mesh type on chip network. This architecture faces the problem of scalability and parallelism in the current SoC context, which contains hundreds of IP blocks each having different interfacing and throughput requirements. The Proteo NoC [11] is an example for packet switched type, which connects the IP components through nodes and hubs. The Proteo NoC architecture is scalable and can overcome few drawbacks of circuit switched type by interconnecting several IP components. However the packet switched type networks [11] carry the data in multiple hops and worst case latencies become potential bottleneck for real time SoC applications.

Despite of real time requirements in several embedded applications, little research has been done on providing time predicted latencies for communication among IP nodes. Hard real-time embedded systems impose a strict latency requirement on interconnection of IP subsystems. Hence the NoC architecture must ensure that each packet of a traffic stream has to be delivered within a time interval[12]. To meet real time requirements circuit switching topology with static and deterministic multiplexing is proposed in [13]. However this paper [13] doesn’t propose any special techniques to overcome the scalability issues in complex SoCs. The scheme proposed at [14] uses genetic algorithms to evolve task mappings and to maintain communication flows to meet real time deadlines in different possible scenarios. However the scheme is complex and demands higher chip area for SOC implementation. The work given at [15] develops a framework for predictable communication synthesis in NoCs with hard real-time constraints. The framework models communication at the link level, using traditional task graph based modelling technique and supports various switching techniques. Certain principles from this paper are used here for analysis.

The CDMA based NoC can provide the benefits of packet switching architectures such as scalability, and also ensures deterministic packet latencies between two IP nodes. The CDMA scheme based NoC architecture presented in this paper establishes connection between IP nodes in packet switching style, but once the connection is established has well deterministic characteristics similar to circuit switched type. The capability to establish multiple point to point (PTP) type connectivity links concurrently, while maintaining fixed latencies is the potential advantage of CDMA based NoCs, which makes them suitable for embedded hard real time ASICs. The roots of research on CDMA based NoC topologies are from the principles of DSSS/CDMA communication system. The Code division multiple access (CDMA) technique has advantages such as multiple access, security and robustness against noise etc. The initial experiments using analog buses inside the chip [6][7] are used to carry the continuous voltage levels are similar to CDMA wireless principle. But these techniques are suffered from the basic analog wise limitations as described in [4]. However as a principle the same can be used at the base band level with digital encoding schemes. The principle of CDMA can be applied to NoC problem to result in efficient solution in comparison with crossbar and other conventional architectures. The schemes described in [2] and [4] illustrate this method to globally asynchronous and locally synchronous (GALS) communication schemes. The work demonstrated at [2] is aimed for simpler interconnectivity requirements and suitable for small sized to medium sized SoCs. The NoC based on this core concept is aimed to address the above described limitations of conventional bus based architectures and also carry below mentioned advantages.

- Low latency and less protocol overheads in establishing or disconnecting the communication link
- Single clock routing network, by which aiming for high speed bus realization yielding best out of silicon[8].
- Secure communication among IPs(where ever required), by taking the inherit advantage of CDMA.
- Dynamic bus topology configuration, by effective key (PN codes) management.

The work described in this paper emphasizes on latency and dynamic bandwidth management considering the ease at silicon level design. The work assumes more importance in context of multiprocessor based systems realized on either ASICs or FPGAs.

ARCHITECTURE AND DESIGN ISSUES

3.1 High level architecture

A common observation on architectures evolved so far [2][3][4][5] is that all of them use a central controller or arbitrator for bus management. This approach has merits of effective controlling and very low latencies in establishing or disconnecting the communication links. However when it is visualized at the VLSI realization point of view the central controller demands either a dedicated routing across the chip or managed through segmented bus connections. The former calls for increased routing layers and later effects the performance [8]. Hence avoiding a central controller or arbitrator at an expense of compromised latencies in key allocation [2] could be useful in certain class of SoCs.

Two architectures are presented here to realize the modified CDMA based NoC architectures.

- Ring based NoC architecture with central key scheduler
• Ring based NoC architecture with scheduler being part of ring.[1]

The first architecture is extension of work described in [2] with dynamic bandwidth management with complete base band level implementation. The base band encoding and decoding are similar to the architectures described in [4]. However the routing schemes illustrated in [4] only cover simplex communication type. This paper presents for full duplex communication scheme. The figure 1 shows the high level architecture for this variant.

The second architecture is new in comparison with existing techniques and emphasizes on control being carried out on same ring. Here SoC with six IP nodes is considered for simulation and analysis. In first architecture, in addition to the circular bus connectivity there are dedicated bidirectional control lines to and from the key scheduler. The key scheduler is the central control unit for whole chip. The key scheduler connectivity requirement forms a star topology. Each node consists of two sets communication blocks each consisting one encoder and decoder. This is to maintain bidirectional connectivity in the ring. A controller selects the appropriate set to minimize the communication latency. The figure 3 shows high level block diagram of CDMA encoder and decoder for each IP node.

The control logic takes the input circular bus value and processes it and places it on the output bus. The processing at each node can be data reading, or data writing or both. The entire communication between nodes is similar to DSSS communication scheme followed in CDMA networks. Each node’s data is spreaded with unique Pseudo Noise (PN) code and the same code shall be used by the destination node to receive the data. The keys (PN codes) are selected from a set of fully orthogonal codes.

Let the node N1 needs to send the data to N3, then N1 sends request to key scheduler, key scheduler first checks the readiness of the N3 and then allocates a key from a pool of keys. The same key is passed to N1 and N3. In case if the Destination node N3 cannot receive data, then the key scheduler informs the same to N1.

Once the confirmation is issued from key scheduler the N1 transmits the data bits after spreading with key bits. The destination node N3 performs decoding with the key hence it will be able to receive the information bits. For all the remaining nodes that are also reading or writing values from the bus, shall not get effected, because their keys will be different and they are orthogonal to the key used by N1 & N3.

The key allocation and management is possible by both static and dynamic methods. A detailed description of such possibilities is given at [4]. As the proposed architecture supposed to have dynamic bus management the transmitter-receiver (TR) protocol is selected among others. This TR key allocation scheme can also ensure data security beyond the advantages described in [4] at an additional cost of increased decoding circuit complexity. Hence the TR key allocation scheme is used against the A-T protocol scheme which is used in [4].

The architecture of each IP node is same for both the ways of implementing key scheduler block. The bus management protocol and key allocation mechanism is different for both. Towards speed optimization the ring bus is routed in both directions connecting all the IP blocks. This results in double routing resources at the benefit of reducing the maximum latency to half.

The operation of multiply and accumulate (CDMA correlation) and further data decoding is discussed through an example in section C.

The operation of key scheduler in central control architecture is mainly to allocate the keys and provide the bandwidth based on the preset priorities and traffic on ring bus. The operation of key scheduler can be described through below steps.

• The key scheduler loads the controls and priorities for various IPs at the start up from pre designated registers.
• The key scheduler assigns allocation bit zero to all keys
• A request form the node (N1) for communication to node (N3) is raised by the making the request signal ‘1’ and providing the destination address.
• The node N1 also sends the bandwidth request to scheduler.
• The scheduler checks the availability of N3 for reception and the number of keys as per the requested bandwidth.
• If N3 is busy then receiver busy signal is set.
• If bandwidth cannot be allocated no bandwidth signal will be set
• If both N3 is ready and bandwidth is available then keys will be transmitted to both N1 and N3. The allocation bits of keys will be made ‘1’.
• The transmitter enable for N1 and receiver enable for N3 will be set.
• Once the data transfer is completed the node N1 removes the request.
• The key scheduler makes both the enable signals to ‘0’ and allocation bits will be turned ‘0’ for the keys.
Figure 1. Ring based NoC with central key scheduler

Figure 2. Ring based NoC with key scheduler being part of ring

Figure 3. Architecture of IP node interface

Figure 4. Ports for IP node interface module.

Figure 5. Time slots used for command and control information
The scheduler holds one key permanently for sending the control words. Also each IP node permanently assigned with one key for communicating with scheduler. However these keys are not blocked they are also used for data communication. The process of handshaking and key assignment happens as per the below described steps.

- For every N time slots one time slot will be used by scheduler for issuing control packet. The value of N can be configured by user register. The dedicated key will be used for this purpose. The control packet contains a command which is encoded in specific number of bits.
- All the IP nodes are continuously listening for the control packet. When control packet is issued the next time slot is assigned for sending the responses or requests.
- Each IP node sends the messages to controller with its pre-assigned key.
- All the responses are collected in one time slot by the key scheduler and it decodes the information.
- Every time scheduler dispatches control packets it does in two iterations. So that in the second iteration is useful for collection of responses from packets for which, the requests were initiated by the IP nodes which are forward in ring bus in the first control slot.
- In the first control packet case the requests are issued and in the second control packet case IP nodes give responses.

Based on this handshaking any command assignment, acknowledgement or key assignment must happen. At the message level different bit sequences are encoded for various type of commands/messages. Note that each time slot consists of several ring bus clock cycles depending on the bit length of command packet.

The figure 5, shows the time slots during which the command information and responses will be communicated. Two subsequent time slots are used for this purpose. During the remaining time the data communication happens as per the established communication links.

### 3.2 Walsh codes
A set of fully orthogonal codes are required for the described application. The Walsh codes are selected for this purpose here[3]. The Walsh codes are defined as a set of N codes, denoted by $W_i$, for $i = 0, 1, \ldots, N - 1$, exhibit the following properties:

- $W_i$ takes on the values $+1$ and $-1$.
- $W_i[0] = 1$ for all $i$.
- $W_i$ has exactly $i$ zero crossings, for $i = 0, 1, \ldots, N - 1$.
- Each code $W_i$ can be even or odd with respect to its midpoint.

Walsh codes are defined using a Hadamard matrix of order N. The logic described in [8] is used for generating the Walsh codes of 8 bit length, which are used in the present simulation.

$$W_i = \{ 255,195,240,153,170,150,165,204 \}$$

#### 3.3 [XOR, OR] Vs Arithmetic sum
The aggregate signal which is result of multiple data elements from various IP nodes is required to be formed by the arithmetic sum of individual spreaded codes. The other operations such as XOR or OR [5] shall not be useful for utilizing the full bandwidth of the system. The following example explains the same.

Let the node N1, N2 and N3 are transmitting the data with Walsh codes W1, W2 and W3 as shown below.

Considering that all the information bits are 1s the same codes are transmitted on ring bus.

<table>
<thead>
<tr>
<th>Node</th>
<th>Walsh Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>W1</td>
</tr>
<tr>
<td>N2</td>
<td>W2</td>
</tr>
<tr>
<td>N3</td>
<td>W3</td>
</tr>
</tbody>
</table>

The resultant values on the ring bus with arithmetic sum shall be $S: [+3 +3 +1 +1 -1 -1 -1 +1]$. When this sum is correlated with other key W4 (from the same set of Walsh codes) results perfectly zero.

<table>
<thead>
<tr>
<th>$W_4$</th>
<th>+1</th>
<th>+1</th>
<th>-1</th>
<th>-1</th>
<th>+1</th>
<th>+1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \times W_4$</td>
<td>+3</td>
<td>+3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>$\sum S \times W_4$</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where as the when the S is when correlated with any one of the W1, W2 and W3 results 8, which is length of the code.

| $S \times W_3$ | +3 | +3 | +1 | +1 | -1 | -1 |
| $\sum S \times W_4$ | 8  |    |    |    |    |    |

Note that the decoding scheme is unique and has perfect decision boundaries 0 or 8 or -8, which is not possible with OR or XOR operations. Hence the penalty of higher bandwidth usage (using 4 bits instead of 1 bit) is unavoidable in CDMA based NoC systems.

#### 3.4 Comparison
The following table compares both the architectures in various aspects.
TABLE I. COMPARISON OF NOC ARCHITECTURES

<table>
<thead>
<tr>
<th>Comparison of two NoC architectures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>1 Communication between scheduler and IP nodes</td>
</tr>
<tr>
<td>2 Data communication between two IP nodes</td>
</tr>
<tr>
<td>3 Maximum number of keys that can be used for data communication</td>
</tr>
<tr>
<td>4 Routing complexity in SoCs</td>
</tr>
<tr>
<td>5 Throughput limiting factors</td>
</tr>
<tr>
<td>6 Establishment and termination of communication</td>
</tr>
</tbody>
</table>

[4] SIMULATION AND SYNTHESIS

The Modelsim simulation results obtained for key scheduler is shown in figure 6. Several state conditions are simulated. When node1 rises the req_1 with destination address 01, the scheduler allocated keys to it and sets tx_en1. As soon as req_1 is turned low, it means that node1 completed data transfer. It can be seen that rx_en2 is made high so that the data communication is established from node1 to node2. The node2 rises req_2 with destination address of 11. Since initially node4 is free the communication is established. Later when node 3 rises request for communication with node4, busy line is set. Once the node 2 releases link with node4, then node 3 to node 4 communication is established.

![Figure 6: Simulation of key scheduler when 4 IP nodes are active](image)

![Figure 7: Simulation results of IP node interface](image)

The Xilinx synthesis tool is used to perform area and speed analysis of both architectures. The results are compared in table II. The scheduler-built-in-ring type architecture can run slightly faster than conventional architecture with slight increase in area overhead. However the advantage of relaxed routing of scheduler-built-in-ring type in SOC realization makes it more suitable for real time complex SOCs in comparison with conventional architecture.

<table>
<thead>
<tr>
<th>Comparison of two NoC architectures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>1 Slices</td>
</tr>
<tr>
<td>2 BRAMs</td>
</tr>
<tr>
<td>3 DSP48s</td>
</tr>
<tr>
<td>4 Maximum clock frequency (MHz)</td>
</tr>
</tbody>
</table>
[5] REAL TIME ASPECTS

The time latencies involved in data transfer with proposed architectures are explained in this section. The IP node’s local clock period is considered as $T_{IP_CLK}$. The ring bus and scheduler are considered to be working with same clk, which is referred as $T_{RING_BUS}$.

The time required to establish communication among IP nodes consists of five components as given in below equation. When a transmit request is raised by an IP node to a specific IP node, then the receiving IP node either can be free state or busy state. Depending on these states it takes either $T_{ACK}$ clock cycles or $T_{BUSY}$ clock cycles to respond.

$$ T_{EST} = T_{TREQ} + T_{RREQ} + \max (T_{ACK}, T_{BUSY}) + \max (T_{RES}, T_{KEY}) $$

- $T_{TREQ}$: Time required to initiate transmit request
- $T_{RREQ}$: Time required to initiate receiver about the transmit request
- $T_{ACK}$: Time taken by receiving node to acknowledge for its readiness in receiving data
- $T_{BUSY}$: Time taken by receiving node to inform that it is busy.
- $T_{RES}$: Response from scheduler to transmitter node
- $T_{KEY}$: Allocation time for keys to transmitter and receiver

![Figure 8. Timing aspects in establishing communication](image)

In addition to this the nodes have to wait for arrival of control packet in scheduler-built-in-ring type architecture. The worst case waiting time could be the period ($T_{CNTL_PAK}$) at which control packets are issued. Once the communication is established the data transfer latencies are fully deterministic and depend number of nodes and pipeline stages. The table III has comparison details for both architectures. The timing aspect to terminate the connection also well deterministic and are not presented here as their implementation is achieved by falling edge of same signalling lines, which are used for establishing the communication. The timing parameters described above are tabulated in below table as per the implemented six node architectures. Here one pipeline stage is considered between two IP nodes in ring bus.

### TABLE III. TIMING PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comparison of two NoC architectures</th>
<th>scheduler-built-in-ring</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{TREQ}$</td>
<td>$T_{IP_CLK}$</td>
<td>$T_{IP_CLK} + T_{RING_BUS} + T_{CNTL_PAK}$</td>
</tr>
<tr>
<td>$T_{RREQ}$</td>
<td>$T_{RING_BUS}$</td>
<td>$T_{RING_BUS}$</td>
</tr>
<tr>
<td>$T_{ACK}$</td>
<td>Depends on IP node</td>
<td>Depends on IP node</td>
</tr>
<tr>
<td>$T_{BUSY}$</td>
<td>Depends on IP node</td>
<td>Depends on IP node</td>
</tr>
<tr>
<td>$T_{RES}$</td>
<td>$T_{RING_BUS}$</td>
<td>$T_{RING_BUS}$</td>
</tr>
<tr>
<td>$T_{KEY}$</td>
<td>$T_{RING_BUS}$</td>
<td>$T_{RING_BUS}$</td>
</tr>
</tbody>
</table>

### TABLE IV. LATENCIES COMPARISON OF NOC ARCHITECTURES

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comparison of two NoC architectures</th>
<th>scheduler-built-in-ring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establishing communication</td>
<td>$T_{IP_CLK} + 2 \cdot T_{RING_BUS}$</td>
<td>$T_{IP_CLK} + 14 \cdot T_{RING_BUS} + T_{CNTL_PAK}$</td>
</tr>
<tr>
<td>Sending one packet from transmitter to receiver</td>
<td>$1 \text{ to } 6\text{ clock cycles depending on the position of Tx and Rx IP nodes in ring bus}$</td>
<td>$1 \text{ to } 7\text{ clock cycles depending on the position of Tx and Rx IP nodes in ring bus}$</td>
</tr>
<tr>
<td>Terminating the connection</td>
<td>$T_{IP_CLK}$</td>
<td>$T_{IP_CLK} + 14 \cdot T_{RING_BUS} + T_{CNTL_PAK}$</td>
</tr>
</tbody>
</table>
CONCLUSIONS
The presented work illustrates the capabilities of CDMA based NoC architectures with two different control mechanisms. The key allocation process, throughput and latency analysis are given. The work illustrates detailed key management mechanism with fully orthogonal Walsh codes. The central key scheduler and ring inserted key scheduler are verified for functionality with VHDL simulation using Modelsim tool. The work aimed to be continued in complete SoC system implementation on FPGA.

REFERENCES