Real Time Network on Chip (NOC) Architecture with CDMA Techniques with Audio Decoders

Talla Vamshi[1], Dr. T. Satya Savithri[2]

[1]Talla Padmavathi College of Engineering, Warangal, T.S, India
[2]Jntuh College of Engineering Hyderabad, T.S, India

***Abstract*:** *The 2014 International Conference on Artificial Intelligence & Manufacturing Engineering (ICAIME 2014) is a worldwide, multidisciplinary academic conference concerned with research, education and application into all aspects of Artificial Intelligence & Manufacturing Engineering, etc. (Use "abstract" The current generation System on Chips (SoCs) demand multiple IPs of different nature working seamlessly with real time interfaces. The challenge in realizing such SoCs is providing a generic bus architecture which takes less resources and power . The Network on Chip (NoC) is the research on Network on Chip (NoC) is under study from past few decades evolving various types of architectures catering for different applications .How ever a simple NoC architecture which is suitable for FPGA based applications is not evolved yet .The work presented in this paper utilizes code division multiple access (CDMA) principle based scalable and real time network on chip architecture. The proposed architecture suits both FPGA and ASICs applications in terms of connecting wide category of IP(Intellectual Property) blocks. In this research a generic architecture for implementing such NoC is proposed. The experimental work on Xilinx FPGA Spartan6 carried out with different communication audio encoder and decoder modules. The functional verification of the NoC is verified for different traffic conditions. Mainly the real time communication aspects of the proposed NoC are highlighted. Simulation and Synthesis based thorough verification carried out using Modelsim and Xilinx tool respectively. Results show maximum clock speed for ring architecture up to 64.293MHz and for Central architecture up to 131.214MHz for the proposed architecture. This research find applications in various futuristic audio and multimedia category FPGA/ASICs based applications.*

Keywords: *NOC, Walsh codes, NOC Latencies , Timing aspects of NOC architectures, Central and Ring key scheduler .*

1. Introduction

The SoC s (System- on-Chips) has to maintain many Processing Elements(PEs) ,in traditional on chip communication architectures may avoid required system performance in many applications because of SoCs suffering with a major problem is power efficiency, this problem managed by a new approach is NoC(Network-on-Chip)[3]. NoC has many features like scalability interoperability, When SoC combined with NoC it gives better synchronization issues and feature developments are enhanced. The Power efficiency of complex SoC improved with NoC and NoC provide high operating frequencies. In communication channels for original data recovery we have to use some basic ECC techniques to correct the errors at receiver end which redundant data added by encoder[2] .Similarly if error occurs in network we have classical error correction code that is NEC(Network Error Correction) codes for this concept a author highlighted on NEC [4]. In this we can see the classical error correction codes, when source information is transmitted to a set of receiving nodes on network [5-10]. In proposed architecture defines effective topologies and its timing and device utilities of NoC implementation[11-14].

In this Fig 1 block diagram shows connectivity and data conversion between encoder and decoders of proposed architecture.

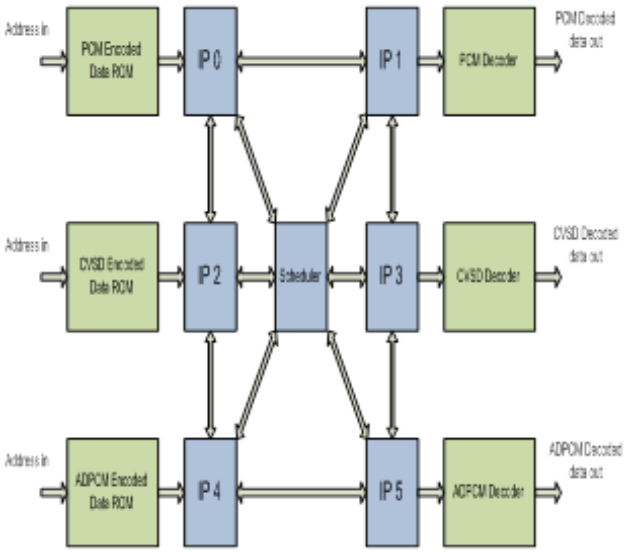

Fig1. Block diagram of NoC with audio decoders

1.1. Connectivity of individual Encoded Data ROMs and IP_Nodes :

The above figure 1 shows NoC with audio decoders. There are three audio decoders, namely PCM, CVSD and ADPCM. Out of 6 IP nodes, 3 nodes are transmitters and the other 3 nodes are receivers. IP_Node_0 transmits PCM encoded data to IP_Node_1, IP_Node_1 receives PCM encoded data and decodes it. In the same way, IP_Node_2 transmits CVSD encoded data to IP_Node_3. IP_Node_3 receives CVSD encoded data and decodes it. IP_Node_4 transmits ADPCM encoded data to IP_Node_5. IP_Node_5 receives ADPCM encoded data and decodes it. The transmitter requests and node addresses are set such that node_0 transmits data to node_1, node_2 transmits data to node_3 and node_4 transmits data to node_5.

1.2. Data Conversion between Encoder, IP_Nodes & Decoder Blocks:

The above discussed connectivity of PCM, CVSD and ADPCM of encoders and decoders at IP_Nodes of NoC, the encoded data samples are loaded to the individual ROMs and those are read using address counter. At every ROM output, the data is parallel. The parallel data is converted to serial and then given to IP_Nodes. The date received at the decoders IP nodes is serial which is then converted to parallel.

2. Walsh Codes

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[1].ts.

W1 +1 +1 +1 +1 +1 +1 +1 +1

W2 +1 +1 -1 -1 -1 -1 +1 +1

W3 +1 +1 +1 +1 -1 -1 -1 -1

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.

W4 +1 +1 -1 -1 +1 +1 -1 -1

S X W4 +3 +3 -1 -1 -1 -1 -1 -1

$\sum$ S X W4 = 0

Whereas 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 X W3 +3 +3 +1 +1 +1 +1 -1 -1

$\sum$ S X W4 = 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. Real Time Aspects

In this proposed architectures time aspects are involved for data transfer with are explained in this section. The local clock period of IP node considered as TCLK_IP. The ring bus and scheduler are considered to be working with same clk, which is referred as TRING_BUS [15].

From figure 2 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 TACKRN clock cycles or TRNBUSY clock cycles to respond.

$$TEC = TITRQ + TIRRQ + \max(TACKRN, TRNBUSY) + \max(TRSTN, TKEYTR)$$

T_{EC}	It is time period to Establish Communication
T_{ITRQ}	It is time period to Initiate Transmit Request
T_{IRRQ}	It is time period to Initiate Receiver about the transmit Request
T_{ACKRN}	Time taken by receiving node to Acknowledge for its readiness in Receiving Data
T_{RNBUSY}	Time taken by Receiving Node to inform that it is Busy.
T_{RSTN}	Response from Scheduler to Transmitter Node
T_{KEYTR}	Allocation time for Keys to Transmitter and Receiver

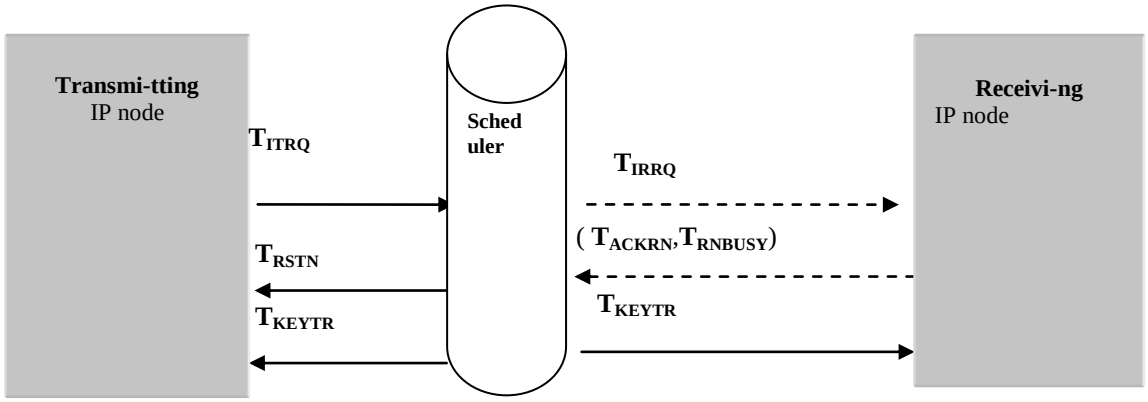

Fig 2. Timing aspects in establishing communication

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 (TCTRL_PACK) 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 I 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 signaling 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 I. Some Timing Aspects for Two Noc Architectures

	NoC architectures Comparison		
	Parameter	*Central key scheduler*	*scheduler-built-in-ring*
1	T_{ITRQ}	1 . T_{CLK_IP}	1. T_{CLK_IP} + 7 . T_{RING_BUS} + T_{CTRL_PACK}
2	T_{IRRQ}	1 . T_{RING_BUS}	7 . T_{RING_BUS}
3	T_{RSTN}	1 . T_{RING_BUS}	7 . T_{RING_BUS}
4	T_{RNBUSY}	Depends on IP node	Depends on IP node
5	T_{KEYTR}	1 . T_{RING_BUS}	7 . T_{RING_BUS}
6	T_{ACKRN}	Depends on IP node	Depends on IP node

The response times of receiver node can make the response times unpredictable. A specific feature is implemented to overcome this limitation whenever possible. The IP nodes can select the option of "dedicated port" at the scheduler, when they are ready for receiving data all the time. This will enable the scheduler to directly reply the transmitter about the readiness of receiver if its receiver port is not engaged by any other transmitter. This makes the proposed NOC completely time deterministic making it suitable for real time SOC applications. The following table lists the latencies for all operations for both the architectures.

TABLE II. Noc Architecture's Latencies And Its Comparison

	NoC architectures Comparison		
	Parameter	***Central key scheduler***	***scheduler-built-in-ring***
1	Establishing communication	1 . T_{CLK_IP} + 2 . T_{RING_BUS}	T_{CTRL_PACK} +1. T_{CLK_IP} + 14 . T_{RING_IPS}
2	Terminating the connection	1 . T_{RING_BUS} +1 . T_{CLK_IP}	T_{CTRL_PACK} +1. T_{CLK_IP} + 14 . T_{RING_BUS}
3	Sending one packet from transmitter to receiver	1 to 6 clock cycles depending on the position of Tx and Rx IP nodes in ring bus	1 to 7 clock cycles depending on the position of Tx and Rx IP nodes in ring bus

4. Results

In this section ModelSim Results are used for explaining the functional simulation and Chipscope results used for on chip verification

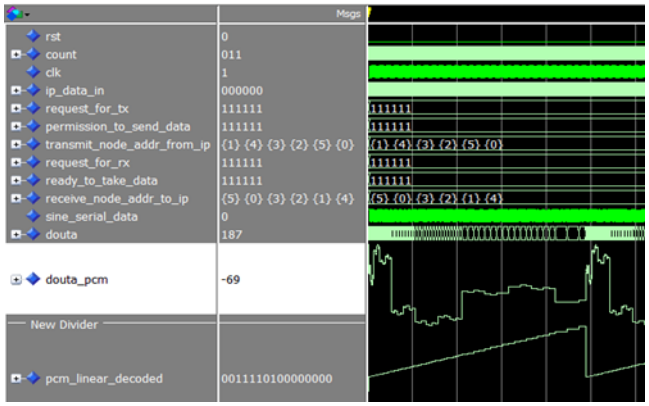

Fig 3. Simulation results of PCM decoding RAMP samples signal

The PCM encoded data (ramp samples) is given to input of IP node 0 serially. In above figure 3 node 0 is transmitting data to node 1. So, the PCM encoded serial data can be received at the node 1. As the PCM decoder

is available at the node 1, decoding is done once the data is received at node 1. The receive node addr to ip shows address value 0 in place of the receiver node 1, it means that node 0 is sending data to node 1. The decoded data can be observed from the last signal which is ramp.

The last 2 signals are the main input and output for one of the PCM decoder. Here, the ramp wave samples are encoded and loaded to the ROM. The signal data pcm is output of ROM, means encoded data samples can be observed by this signal. The last signal is the PCM output which is named as pcm_linear_decoded, decoded samples can be clearly observed from this signal.

The PCM encoded data (ramp wave samples) are given to input of IP node 0 serially, in which node 0 is transmitting data to node 1. So, the encoded serial data can be received at the node 1. As the PCM decoder is available at the node 1, the decoding is done once the data is received at node 1. In receive node addr to ip, it is showing address value 0 in the place of the receiver node 1, it means that node 0 is sending data to node 1. The decoded data can be observed from the last signal which is ramp data.

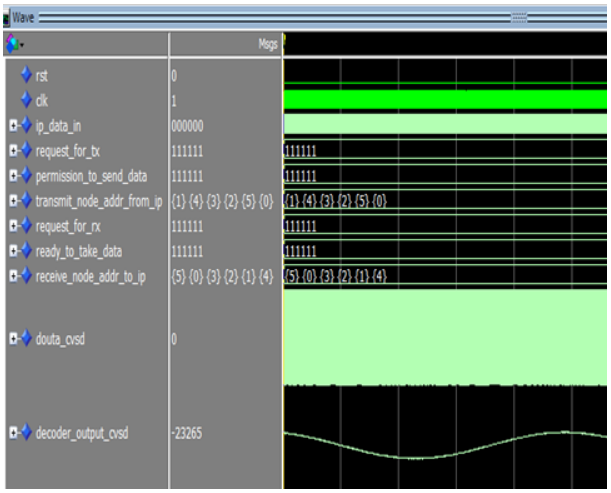

Fig 4. Simulation results of CVSD

The CVSD encoded data (sine wave samples) is given to input of IP node 2 serially. In this Figure 4, node 2 is transmitting data to node 3. So, the CVSD encoded serial data can be received at the node 3. As the CVSD decoder is available at the node 3, the decoding is done once the data is received at node 3. In receive node addr to ip, it is showing address value 2 in the place of the receiver node 3, it means that node 2 is sending data to node 3. The decoded data can be observed from the last signal which is sine wave.

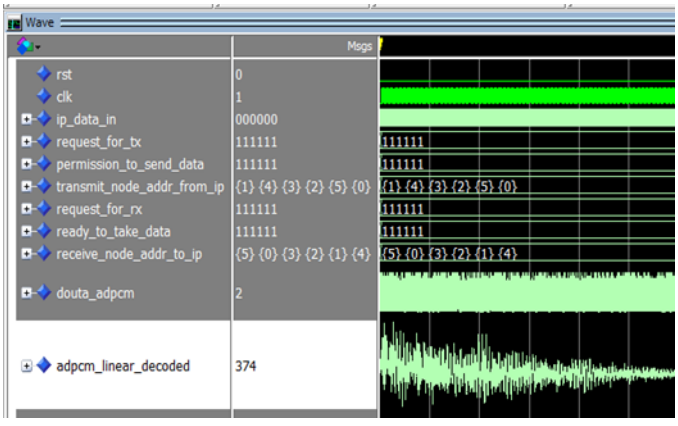

Fig:5 Simulation results of ADPCM

The above figure 5 shows ADPCM Decoder simulation results. The first signal is asynchronous reset named as rst, second is the clock named as clk. Next signal is request_for_tx, using this input transmitter can put its request. In this figure, all nodes are requesting to transmit data. The next signal is used to give the permission for transmitter to send data, permission is given based on the readiness of the receiver node. In this example, all nodes got permission to send data because all nodes are ready to take data, receiver node readiness can be observed from the signal ready_to_take_data. The signal named as receive_node_addr_to_ip indicates the transmitter address from which it is receiving data.

5. Chipscope Results

Chipscope Results at Decoders of Proposed Implementation:

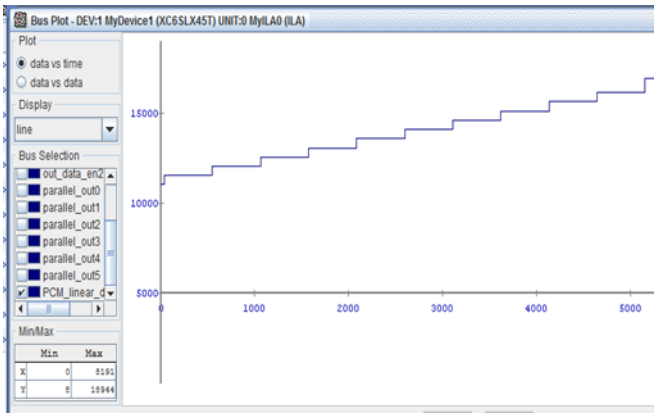

Fig 6. Chipscope verified results of PCM Decoder results received at node_1

The above figure 6 shows the results of PCM Decoder. At the transmitter ramp encoded data is given. So, the same kind of data is received after decoding. The PCM encoded data of ramp wave is stored in ROM and transmitted form node_0. The node_0 transmitted data is received at node_1 and PCM Decoding is performed. The chipscope result showing ramp which is decoder output of PCM Decoder.

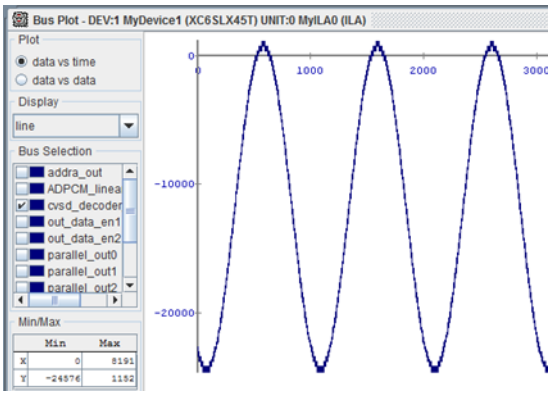

Fig 7. Chipscope verified results of CVSD Decoder results received at node_3

The above figure 7 shows the results of CVSD Decoder. At the transmitter ramp encoded data is given. So, the same kind of data is received after decoding. The CVSD encoded data of sine wave is stored in ROM and transmitted form node_2. The node_2 transmitted data is received at node_3 and CVSD Decoding is performed. In the chipscope results it can be observed that decoded signal is sine wave.

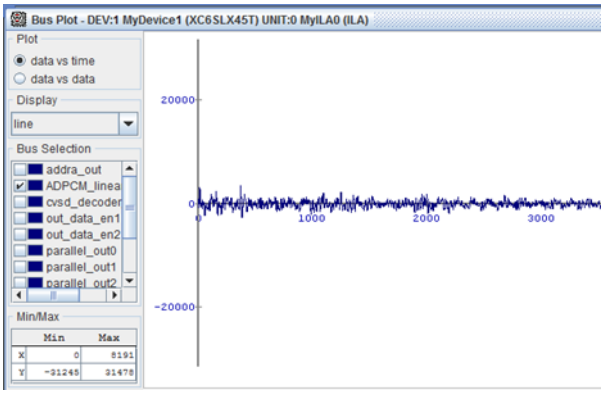

Fig 8. ADPCM Decoder results received at node_5

The above figure 8 shows the results of ADPCM Decoder. At the transmitter ramp encoded data is given. So, the same kind of data is received after decoding. The ADPCM encoded data of audio samples is stored in ROM and transmitted form node_4. The node_4 transmitted data is received at node_5 and ADPCM Decoding is performed.

6. Central And Ring Type Nocs Timing Reports For Various Fpgas

TABLE III: Timings Comparison of Both Architecture

FPGA TYPE	CENTRAL	RING
Spartan_3E	15.846ns(64.576Mhz)	2.825ns(43.81Mhz)
Spartan_6	9.393ns (106.465MHz)	15.554ns(64.293Mhz)
Virtex_6	4.883ns(204.792Mhz)	8.497ns(117.688Mhz)
Kintex_7	4.702ns(212.675Mhz)	8.437ns(118.523Mhz)

7. Timing Latencies for Both Architectures\

Noc_Central:

Time to get permission = 3 clocks.

Total time to get output (in worst case, from 1st node to 6th node communication) = 10 clocks (3 for permission and 7 for communication).

Noc_Ring:

Total time for 1 time interval = 255 clocks.

command mode time in time interval = 18 clocks. (6 for requests, 6 for responses, and 6 more for proper communication). All permissions will be given with in command mode and output will also come with in command mode.

8. Device Utilization Summary and Timing Summary for Cental and Ring Architectures

From figure 9 shows device utilization summary of Central architecture .

Noc_Central:

Selected Device : 6slx45tfgg484-3

Slice Logic Utilization:

Number of Slice Registers:	407 out of 54576	0%
Number of Slice LUTs:	1177 out of 27288	4%

Number used as Logic: 1123 out of 27288 4%
Number used as Memory: 54 out of 6408 0%
Number used as SRL: 54

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 1232
Number with an unused Flip Flop: 825 out of 1232 66%
Number with an unused LUT: 55 out of 1232 4%
Number of fully used LUT-FF pairs: 352 out of 1232 28%
Number of unique control sets: 10

IO Utilization:

Number of IOs: 55
Number of bonded IOBs: 50 out of 296 16%

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs: 1 out of 16 6%

Timing Summary:

Speed Grade: -3

Minimum period: 7.621ns (Maximum Frequency: 131.214MHz)
Minimum input arrival time before clock: 6.408ns
Maximum output required time after clock: 3.597ns
Maximum combinational path delay: No path found

==========

Process "Synthesize - XST" completed successfully

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	407	54,576	1%	
Number used as Flip Flops	407			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	1,039	27,288	3%	
Number used as logic	1,011	27,288	3%	
Number using O6 output only	899			
Number using O5 output only	0			
Number using O5 and O6	112			
Number used as ROM	0			
Number used as Memory	28	6,408	1%	
Number used as Dual Port RAM	0			
Number used as Single Port RAM	0			
Number used as Shift Register	28			
Number using O6 output only	2			
Number using O5 output only	0			
Number using O5 and O6	26			
Number of occupied Slices	349	6,822	5%	
Number of LUT Flip Flop pairs used	1,058			

Fig 9: Device Utilization for Central architecture

Noc_Ring:

From figure 9 shows device utilization summary of Central architecture .

Selected Device : 6slx45tfgg484-3

Slice Logic Utilization:

Number of Slice Registers: 692 out of 54576 1%

Number of Slice LUTs: 1501 out of 27288 7%
Number used as Logic: 1471 out of 27288 6%
Number used as Memory: 28 out of 6408 0%
Number used as SRL: 54

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 2150
Number with an unused Flip Flop: 1458 out of 2150 67%
Number with an unused LUT: 225 out of 2150 10%
Number of fully used LUT-FF pairs: 467 out of 2150 21%
Number of unique control sets: 25

IO Utilization:

Number of IOs: 74
Number of bonded IOBs: 74 out of 296 25%
IOB Flip Flops/Latches: 25

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs: 8 out of 16 50%
Timing Summary:

Speed Grade: -3

Minimum period: 15.554ns (Maximum Frequency: 64.293MHz)
Minimum input arrival time before clock: 15.016ns
Maximum output required time after clock: 3.752ns
Maximum combinational path delay: No path found

==========

Process "Synthesize - XST" completed successfully

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	692	54,576	1%	
Number used as Flip Flops	462			
Number used as Latches	230			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	1,501	27,288	5%	
Number used as logic	1,471	27,288	5%	
Number using O6 output only	987			
Number using O5 output only	37			
Number using O5 and O6	447			
Number used as ROM	0			
Number used as Memory	28	6,408	1%	
Number used as Dual Port RAM	0			
Number used as Single Port RAM	0			
Number used as Shift Register	28			
Number using O6 output only	2			
Number using O5 output only	0			
Number using O5 and O6	26			
Number used exclusively as route-thrus	2			

Fig 10: Device Utilization for Ring architecture

9. References

[1] Talla Vamshi and Dr. T. Satya Savithri, "NoC architecture with Direct Sequence Spread Spectrum coding techniques, India Conference (INDICON) 2011 Annual IEEE conference, pp.1-5

[2] T. Etzion and A. Vardy, "Error-correcting codes in projective space," Information Theory, IEEE Transactions on, vol. 57, pp. 1165-1173, 2011.

[3] Ahmed A. El Badry and Mohamed A. Abd El Ghany, "A CDMA Based Scalable Hierarchical Architecture for Network-On-Chip", IJCSI International Journal of Computer Sciences Issues, Vol 9, Issue 5, No 2, 2012.

[4] Wael M.El-Medany,"FPGA-Based MWD for Network Error Correction", in Int. J. Com. Net. Teach. 3, No. 2, 59-68 (May 2015).

[5] N. Cai and R. W. Yeung, "Network error correction, II: Lower bounds," Communications in Information & Systems, vol. 6, pp. 37-54, 2006.

[6] I. Martin, B. Honary, and P. Farrell, "Modified minimum- weight decoding for Reed-Solomon codes," Electronics Letters, vol. 31, pp. 713-714, 1995.

[7] N. Cai and R. W. Yeung, "Network coding and error correction," in Proc. 2002 IEEE Inform. Theory Workshop, 2002, pp. 119-122.

[8] Z. Zhang, "Linear network error correction codes in packet networks," Information Theory, IEEE Transactions on, vol. 54, pp. 209-218, 2008

[9] Z. Zhang, "Theory and applications of network error correction coding," Proceedings of the IEEE, vol. 99, pp. 406-420, 2011.

[10] C. Peters, "Information-set decoding for linear codes over Fq," in Post-Quantum Cryptography, ed: Springer, 2010, pp.81-94.

[11] Talla Vamshi and Dr. T. Satya Savithri, ―A modified CDMA scheme based NoC architecture for complex SOC applications‖, Micro Electronics and Electronics (Prime Asia), 2013 IEEE Asia Pacific Conference on Post Graduate Research , pp.137-142

[12] Ahmed A. El Badry and Mohamed A. Abd El Ghany, "A CDMA Based Scalable Hierarchical Architecture for Network-On-Chip", IJCSI International Journal of Computer Sciences Issues, Vol 9, Issue 5, No 2, 2012.

[13] Daewook Kim,Manho Kim and Gerald E.Sobelman FPGA- based CDMA Switch for Networks-on-Chip‖, (FCCM'05), IEEE computer SoCiety.

[14] Xin Wang, Tapani Ahonen ,and Jari Nurmi, "Applying CDMA Technique to Network-on-Chip"., IEEE TRANSACTIONS ON VLSI SYSTEMS, VOL.15, NO.10, OCTOBER2007.

[15] Cristian Grecu, Partha Pratim Pande, Andre´ Ivanov and Res Saleh ―Timing analysis of network on chip architectures for MP-SoC platforms‖, ELSEVIER Microelectronics Journal 36 (2005) 833–845.