Reconfigurable Adaptive Routing Buffer Design for Scalable Power Efficient Network On Chip

G. Selvaraj1* and K. R. Kashwan2

1Department of Electronics and Communication Engineering, Bhaktavatsalam Polytechnic College, Karaipeetai, Kanchipuram - 631552, Tamil Nadu, India; g.selvaraj@rocketmail.com
2Department of Electronics and Communication Engineering-PG, Sona College of Technology, (An Autonomous Institution), TPT Road, Salem - 636005, Tamil Nadu, India; kashwan.kr@gmail.com

Abstract
The layout density of integrated circuits on a single chip has led to the reduced size at sub-micron scales of VLSI design. The System on Chip (SOC) design has few challenges, such as latency, power, area and reliable data transmission among sub-systems interconnected on a single chip. Network on Chip (NOC) is a subset of SOC which accomplishes on-chip communication process. The performance of NOC architecture is significantly affected by power and area. This research work has focused on a new low power reconfigurable NOC architecture with repeaters between the routers. The repeater enables zero buffers between the interlink routers. It works on the principles of store and forward process. The proposed architecture is implemented using mesh network topology. The simulated results of new architecture have shown a reduction in power dissipation by 56% and reduction in on chip area by as much as 60%.

Keywords: Network on Chip (NOC), Reconfigurable Architecture, System on Chip (SOC)

1. Introduction
System on Chip (SOC) mainly consists of many IP cores along with routing techniques. The conventional routing technique of bus topology is no more suitable for SOC architectures. The challenge is overcome by changing the technique to a reconfigurable and scalable NOC architecture. Switched Pockets NOC architecture consists of processing elements, virtual channels, routers and switches. Every processing element is connected to an NOC router for data transmission. Network on chip architecture has crossbar switches with virtual channels for every input port. The router buffers are enabled for each virtual channel. Optimizations of power dissipation and on chip area are the main challenges in an NOC architecture design. To address the challenge, a novel method called NOC for low power and low on chip area is proposed. It is reconfigurable architecture with adaptive storage buffer for data transmission among sub systems on an SOC and NOC. The reconfigurable adaptive buffer performs both the operation of storage and onward transmission. This makes a very effective communication among the sub systems of an SOC. It also facilitates on chip area reduction process by which means a much higher layout density of transistors on chip is achieved.

A new control circuit is introduced for enabling the repeaters along the routers. These perform the function of adaptive link buffer during the data traffic of NOC. The control circuit performs at a less power consumption and operates at variable clock frequency. The proposed architecture is implemented on a 45 nm technology.

*Author for correspondence
The implemented circuit design is simulated for the performance analysis and verification of reliability tests. The simulated tests’ results show an improvement. The elimination of router buffers saves both area and power requirements.

2. Adaptive Link Buffers

The high speed VLSI architectures require repeaters along the data path operators to perform efficiently at much higher data transfer speeds. The existing repeaters are replaced by tri-state repeaters capable of sampling and holding the data on an inter router link.

It gives the flexibility to function in a reconfigurable adaptive routing mode during the flow of network traffic. The controlled circuit mainly consists of a pass transistor, capacitor and clock. The capacitor switched the clock is global. Pass transistors are driven by the clock signals. The Global Clock Signal (GCLK) is subsequently divided into two signals, GCLK1, and GCLK2. The frequencies at sub clocks should be half of global clock. The gate control input signal at the pass transistor takes care of congestion control operation.

The congestion control signal controls the repeaters which are synchronized with global clock. The capacitor is charged and discharged via pass transistor path. Due to the capacitor charging and discharging, the congestion control signal is much delayed for each clock cycle. For the subsequent clock cycles, the corresponding repeaters operate tri-state buffer to hold the data in a position. Meanwhile the congestion signal travels to next control block. The main objective of the control block is to reconfigure the adaptive link buffer. The field programmable gate array circuit is included in order to reduce the area, power and latency of the NOC circuit.

2.1 Reconfigurable Computing

The reconfigurable adaptive link buffer is simulated using Xilinx platform using Verilog. It is illustrated in Figure 3 with four stages of link buffer. If congestion signal goes high, the output is tri-stated and delayed by one clock cycle ranging 1 to 10 ns. Data are released if a congestion signal goes low at output ranging from 10 to 22 ns. The proposed control circuit acts as an optimum router and congestion controller. The power efficiency and control circuit can be turned OFF if there is no congestion signal. In addition to it, a single control block can drive the control input of the entire adaptive buffer at a given stage. The simple switched capacitor maintains clock accurately at variable frequency. It also provides signal stability at high clock speeds.
2.2 Field Programmable Gate Array

Reconfigurable computing consists of an array of elements that can be interconnected easily by using Field Programmable Gate Array (FPGA) principles. The interconnection amongst elements is user programmable. Block level FPGA representation consists of a two-dimensional array of logic blocks that can be connected by programming. These logic blocks are connected through wire segments of different lengths. Interconnects are programmable switches which connect wire segments to the logic blocks. Firstly, logic circuits in FPGAs are partitioned into logic blocks and subsequently the logic blocks are interconnected by using switches.

FPGAs are capable of implementing digital circuits of various sizes. The size of circuit is also called as the density of the circuit. There can be a trade-off between complexity and flexibility of both the logic blocks and the interconnection of an FPGA.

2.3 Reconfigurable Design using FPGA

FPGA based digital circuit design is based on CAD principles. The digital circuit design can be optimized by using FPGA implementation tools. For this process, the initial logic entry may be syntax based or it may involve the drawing of a schematic using a schematic capture tool. For syntax based programming, entering an HDL description is essential. It can be done by specifying Boolean expression or a state machine entry.

The Boolean functions are optimized by using logical tools for on chip area or data transfer speed or similar other applications.

For simulations, first the Boolean expressions are implemented as digital circuits on FPGA logic blocks. This is called technology mapping. After mapping the digital circuits on logic blocks, placement is optimized by deciding a location for each logic block to be configured in the FPGA. The last step for implementation process is the routing of logic blocks. For this, a number of algorithms are available for different FPGA architectures. After completion of all the steps of simulation tests, the architecture is implemented on FPGA by using a programming module which configures the hardware design on FPGA chip.

2.4 Routing Topology

Routing topology is based on the flow control techniques and a number of available routing algorithms for optimization. The flow control module focuses on buffer size and channel bandwidth requirements. The adaptive reconfigurable NOC can be used to eliminate the buffer unit of architecture. This reduces on chip area and thus less power consumption. Wire delays are normally impacted by channel bandwidth allocation schemes.
Routing is accomplished in two steps. The first step is called global or loose routing. It focuses on wiring channels. The second step is called detailed routing. It ensures precise wire routes for different channel layers.

## 2.5 Area and Channel Routing Algorithm

The reconfigurable routing algorithms such as Maize Routing are mainly used for optimization of area and channel allocations. The efficient paths between source and destination are identified by executing algorithm.

The total on chip area is estimated in order to implement the architecture with reduced chip area, global delays and local delays. Channel routing is used to find the shortest path between the interconnected sub modules of architectures. Both channel and area routing algorithm implementations show an improved performance while testing network data traffic in an NOC. The brief steps of Lee's Maze algorithm for routing are listed here for illustration of execution flow.

```plaintext
Struct grid_point
{Int value;
 /*zero for unused, positive for label, -1 for obstacles*/}
Lee (structgrid_pointS,T)
{Set of structgrid_point wave_front, new_wave_front:
Structgrid_pointneighbor, element, path_element:
Int label;
/*step1:wave propagation*/
New_wave_front {S};
Label=0;
While (T€ new_wave_front)
{label=label+1;
 wave_front=new_wave_front;
 new_wave_front=Ø;
 for each element € wave_front
 for each “neighbor of” element
 /*a neighbor is located above, below, at the left or at the right.*/
 If (neighbor .value = 0)
 {Neighbor.value=label;
 New_wave_front=new_wave_frontU(neighbor);}
 /*step 2:back tracking*/
 Path_element ← T;
 For(i=label;
i≤1; i=i+1) Path_element← “the neighbor of path_element such that neighbor.value=i”;
 /*in case of multiple possibilities use a heuristic to make a choice.*/
 Path_element.value←-1
 /*step3: clean up*/
 For each point “on the grid”
 If (point.value>0)
 Point.value←0 ;}
```

### 2.5.1 Channel Routing

robust_robust_router(structnetlist N)
{Set of int row;
Struct solution S;
Inttotal[channel_width+1], selected_net [channel_width+1];
Inttop, height, c, r, I;
top←1;
height←density(N);
for(r←1;
r≤height; r←r+1)
{for all “nets I in netlist N” wi←compute_weight;(N, top);
total[0]=0;
for(c←1;
c≤channel_width;c←c+1)
{selected_net[c]=0;
total[c]=total[c-1];
if(“some net n has top terminal at position c”)
{Total[c]=wn+total[xnmin-1];
 selected_net[c]=n;}
If (“net n has a bottom terminal at position c”)
{total[c]=wn+total[xmin-1];
 selected_net[c]=n;}
row←Ø;
c←channel_width; While (c>0) if(selected net[c])
{ n←selected_net[c];
 row=rowU{n}; c←xnmin-1;}
else
c←c-1;
solution=solutionU[row];
top=!top;
N€“N without the nets selected in row”}
“Apply maze routing to eliminate possible vertical constraint violations”
```
3. Performance Metrics

3.1 Latency

The main components responsible for the most of the delay are the cascaded drivers at the input and the bus trunks. These contribute for the delay due to inherent nature of architecture. The bus delay can be mathematically modelled as shown in equation 1.

\[ \delta_b = k_1 C \frac{1}{k} N^{1/K} + k_2 N + k_3 N^2 \]  

(1)

Where \( \delta_b \) is bus delay, \( N \) is number of switches, \( K \) is number of nodes and \( C \) is control input. There are a few types of switches, suitable for NOC, such as crossbar switch. A standard crossbar has an \( N \) input and \( N \) output as well as a control input signal. Any input may be connected to any of the outputs or to any combination of outputs as required with the help of control signal. One way to implement a crossbar switch is using MOS transistors shown in Figure 6. The width and length are expressed in micron (\( \mu \)).

A data path from each input connects to \( N \) switches and in the same way each switch is connected to one of the \( N \) output data paths. An alternative design using multiplexer is shown in Figure 7.

A 2-to-1 multiplexer is usually used to select the desired input for each output. The multiplexer based crossbar switch has many advantages such as reduced time delay for selecting the optimum data path operators. Multiplexer tree hierarchy is shown in Figure 8.

The average delay times of a switching circuit normally depends upon the topology of the NOC. For this research work, the reconfigurable adaptive routing is used for delay measurement and complexity analysis for balanced tree technique. The balanced tree technique is illustrated in Figure 9. The proposed NOC routing uses a 2×2 MUX based crossbar switch. This switch is normally used for effective delay minimization for bus and crossbar switch.

Table 1 is listed with asymptotic comparison of network delays for different parameters. As shown in Table 1, \( O (N^3) \) denotes order of \( N^3 \) and \( N \) denotes the number of switches.

3.2 Power and Energy Consumption

There are many ways to put in strategies for building low-power gates. These may include multiple threshold inverter, Variable Threshold CMOS (VTCMOS) gate etc.
Among all these methods, a more aggressive method, called variable threshold CMOS gate is more popular and frequently used for various applications. VTCMOS logic can wake up much faster compared to other logic structures. One such example block diagram for VTCMOS is illustrated in Figure 10.

### 4. Results and Discussion

The reconfigurable adaptive link buffer is designed using 45 nm technology with an operating frequency at 1 GHz. The parameters for simulations are chosen based on the objectives of delay and chip area optimization. The supply voltage at 1.0 v is chosen to source power the repeater interlink router circuit.

SRAM cell is designed to store data contents of adaptive link buffer. The test case is denoted by $V_{c_n}R_{r_n}B_{b_n}$ where $c_n$ is number of virtual channels per input port, $r_n$ is the number of router buffer and $b_n$ is number of adaptive storage buffer. The base line is denoted as $V4_R4_B0$. The power is consumed per running of a data cycle of on-chip network under uniform random traffic. The buffer is distributed by adaptive storage buffer at network power of 0.3 v. It eliminates router buffer compare to base line $V4-R0-B4$. This process saves power and area as much as 70% and 80% are respectively. The throughput is degraded by a 3% margin, due to elimination of buffer. This can be overcome by the reconfiguration using FPGA. A better network performance is achieved without any other parameter degradation.

The reconfigurable adaptive routing power minimization implementation is based on VTCMOS method. This method is used for reduction of leakage power. It also avoids power consumption for inactive switches and routers for longtime in the NOC. The proposed low power gating scheme describes multimedia like Dream mode, Snore mode and sleep modes. Power gating scheme is designed and analyzed by using CADENCE simulator environment. Results are shown in Figure 12, Figure 13 and Figure 14.

The dream mode is most suitable for reconfigurable routing. Dream mode has less sleep time and wake up time compared to other the modes.

The proposed NOC routing is configured on FPGA blocks by using hardware description language. These blocks are used for storing packets as well as configuring the virtual channels for NOC router. These virtual channels are effectively used if there occurs any network congestion.
5. Conclusion

The reconfigurable adaptive buffer is far more efficient technique for integrated circuits on chip. The buffer mainly focuses on the efficient channel allocation while data traffic occurs along the data path operators. This architectural technique saves power, area and decreases the latency by a significant margin. The power consumption is minimized due to a low power gating technique adopted. The power consumption is further reduced by eliminating the long time inactivity of interconnected chips by variable threshold CMOS technique. The global delay and local delays are minimized by replacing normal crossbar switch with multiplexer based crossbar switch. The area is minimized.
due to zero buffer technique. The implementation of technique and its simulated test results show that the overall network performance of the proposed technique is better compared to the conventional method. The improvement in performance is significant. The power consumption is improved by 70% and on chip area utilization is improved by 80%. The throughput, however, is degraded marginally at 3% due to elimination of buffer, which is tolerable and insignificant for most of the applications. The power and area improvement is huge and thus overall good improvement has been achieved.

6. Reference

1. Altman E, Boulogue T, Azouzi RE, Jimenez T. A survey on networking games. Telecommunication Systems. 2000 Nov.
2. Andersen DG, Balakrishnan H, Kaashoek MF, Morris R. Resilient overlay networks. Proceedings of the 18th Annual ACM Symposium on Operating Systems Principles; Banff, Canada. 2001 Oct.
3. Mizuno M, Dally WJ, Onishi H. Elastic interconnects: repeater inserted long wiring capable of compressing and decompressing data. Proceedings of the IEEE Int
8. Blake G, Dreslinski RG, Mudge T. A survey of multicore processors. IEEE Signal Processing Magazine. 2009 Nov. 26(6):26–37.
9. Karam L, AlKamal I, Gatherer A, Frantz G, Anderson D, Evans B. Trends in multicore DSP platforms. IEEE Signal Processing Magazine. 2009 Nov; 26(6):38–49.
10. Jerger NE, Peh L. On chip networks (synthesis lectures on computer architecture). Morgan and Claypool Publishers; 2009.
11. Hübner M. Multiprocessor system-on-chip: hardware design and tool integration. Springer; 2011.
12. Kamal R, Yadav N. NoC and Bus architecture: A comparison. International Journal of Engineering Science and Technology. 2012 Apr; 4(4):1438–42.