Power & Area Efficient Router in 2-D Mesh Network-on-Chip Using Low Power Methodology - Clock Gating Techniques

Sudhir N. Shelke[1] and Pramod B. Patil[2]

[1]*Assistant Professor, Department of Electronics & Telecommunication Engineering,*
J.D. College of Engineering, Nagpur, India
[2]*Principal, J.D. College of Engineering, Nagpur, India*
shelkesn@rknec.edu, ppamt07@yahoo.com

Abstract

Network-on-Chip (NoC) is the interconnection platform that answers the requirements of the modern on-Chip design. Small optimizations in NoC router architecture can show a significant improvement in the overall performance of NoC based systems. Power consumption, area overhead and the entire NoC performance is influenced by the router buffers. Resource sharing for on-chip network is critical to reduce the chip area and power consumption.An area efficient implementation of a routing node for a NoC is presented. Of the four components of routing node, the input block (mainly consisting of buffers) and scheduler have been modified to save area requirements. The other two components of the routing node take up negligible area in comparison. The use of custom SRAM in place of synthesizable flip flops in the input block has resulted in a saving of over 26% of the silicon area and power optimization is 65% when operated at 16 ns clock.

Clock gating is an important high-level technique for reducing the power consumption of a design. Clock gating reduces the clock network power dissipation, relaxes the datapath timing, and reduces routing congestion by eliminating feedback multiplexer loops. For designs that have large multi-bit registers, clock gating can save power and reduce the number of gates in the design. In our design case, it has been further observed that the power optimization with clock gating techniques at RTL level saves 67.38%, Gate Level 67.29% & Power Driven around 68.79% of power while 30.38 %, 27.85 % &31.21% silicon area respectively have been saved.

Keywords**:** *Network-on-Chip, Router, SRAM, RTL, Clock Gating*

1. Introduction

Ever-increasing requirements on electronic systems are one of the key factors for evolution of the integrated circuit technology. Multiprocessing is the solution to meet the requirements of upcoming applications. Multiprocessing over heterogeneous functional units require efficient on chip communication [1, 2].

Network-on-Chip (NoC) is a general purpose on-chip communication concept that offers high throughput, which is the basic requirement to deal with complexity of modern systems, as shown in Figure 1.

All links in NoC can be simultaneously used for data transmission, which provides a high level of parallelism and makes it attractive to replace the typical communication architectures like shared buses or point-to-point dedicated wires.

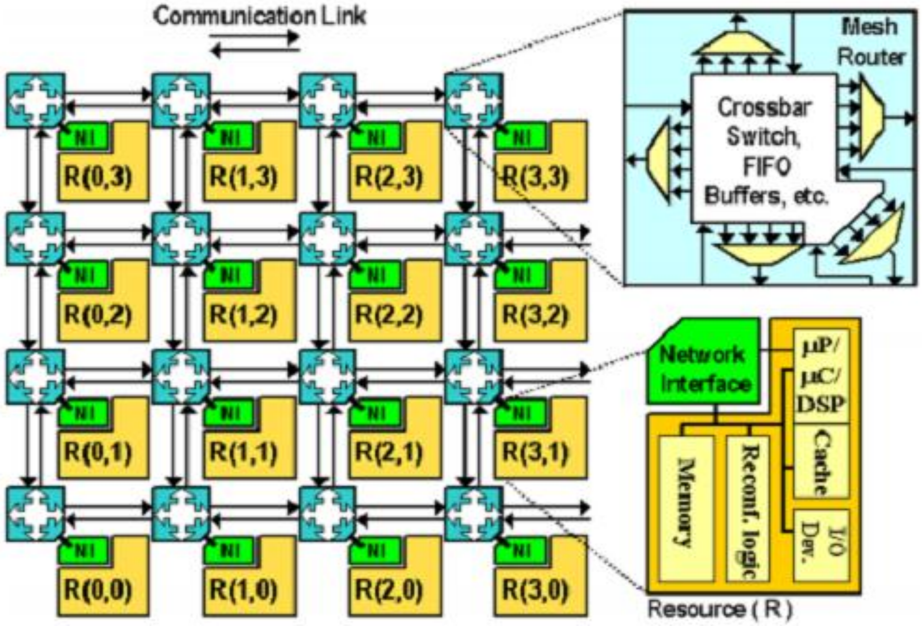


Figure 1. NoC Architecture

Apart from throughput, NoC platform is scalable and has the potential to keep up with the pace of technology advances [3]. But all these enhancements come at the expense of area and power. In the RAW multiprocessor system, interconnection network consumes 36% of the total chip power [4].

A typical NoC system consists of processing elements (PEs), network interfaces (NIs), routers and channels. The router further contains scheduler, switch and buffers. Buffers consume the 64% of the total node (router + link) leakage power for all process technologies, which makes it the largest power consumer in any NoC system [5]. Moreover, buffers are dominant for dynamic energy consumption [6].

2. NoC Architecture

Network-on-Chip has been proposed on various topologies [7-10]. A simple NoC architecture consists of three components: the routing nodes, the links, and network interfaces (or network adapters in some literature), as shown in Figure 2.

Routers direct data over several links (hops). Topology defines their logical lay-out (connections) whereas floorplan defines the physical layout. The function of a network interface (adapter) is to decouple computation (the resources) from communication (the network). Routing decides the path taken from source to the destination whereas switching and flow control policies define the timing of transfers. Task scheduling refers to the order in which the application tasks are executed and task mapping defines which processing element (PE) executes certain task. IP mapping, on the other hand, defines how PEs and other resources are connected to the NoC [11].

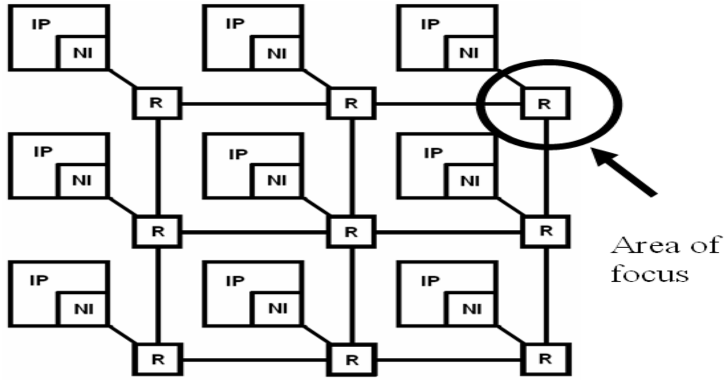


Figure 2. NoC Overview

The major goal of communication-centric design and NoC paradigm is to achieve greater design productivity and performance by handling the increasing parallelism, manufacturing complexity, wiring problems, and reliability. The three critical challenges for NoC are: power, area, latency, and CAD compatibility [12].The key research areas in network-on-chip design [13, 14].are as:

- Communication infrastructure: topology and link optimization, buffer sizing, floorplanning, clock domains, power.
- Communication paradigm: routing, switching, flow control, quality-of-service, network interfaces
- Application mapping: task mapping/scheduling and IP component mapping.

All of these challenges result in area, power, and performance tradeoffs [13].Area and power can be estimated from hardware requirements. Performance is generally estimated using analytical model.

This paper proposes the area and power efficient design of the router as it is the most redundant component which is equal to the no. of PEs on one kind of NoC, as shown in Figure 2.

3. Problem Statement

The implementation of network-on-chip presents certain challenges. Two of the most critical design metrics for networks-on-chip are area requirements and power consumption. Due to the fact that die area per wafer of silicon is limited, the NoC implementation should be carried out using an approach that minimizes area requirement. Also due to likelihood of most SoCs being implemented in battery powered devices, power consumption of the NoC should also be as low as possible. Usually, reduction in area results in a saving in power requirements due to the fact a smaller area is achieved using fewer components on-chip. Fewer components on-chip will consumeless power compared to architecture requiring more components on-chip.

Standard-cell based ASIC design methodology is the fastest approach available in the design of complex digital circuits. However, the performance of digital systems can be enhanced by making use of custom IP cores to replace some of the standard-cell components. This performance enhancement comes at the price of increased design time and effort, but is preferred for maximizing performance.

4. Design and Implementation of Proposed Task

Given an existing implementation of a routing node for on-chip networks, it is the goal of this work to present a modified implementation of the routing node to minimize the area requirements and as a result lower the power requirement.

The routing node consists of four basic components: the input ports, the output ports, the crossbar switch, and the scheduler. The components arranged in decreasing order of size are the input blocks, the scheduler, the output blocks, and the crossbar switch as shown in Figure 3.

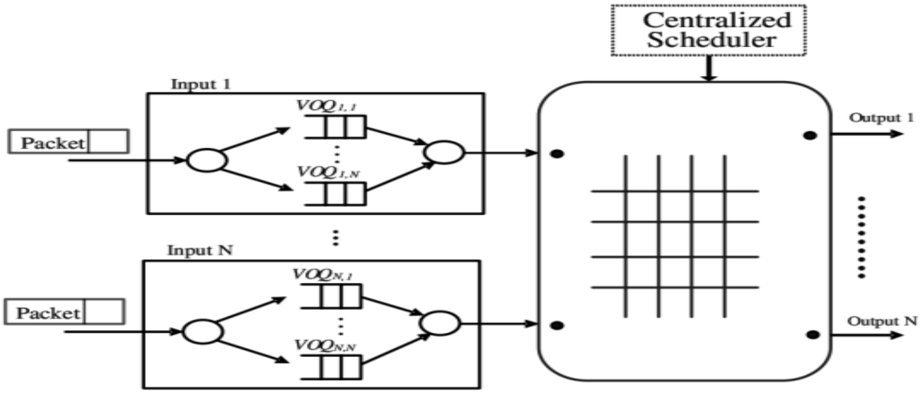


Figure 3. Router Components

The primary function of the input block is to store incoming packets before they can be routed to their respective output ports. Hence, the majority of the area of the input blocks is used by memory elements. The existing design employs DFF (D flip-flop) elements for memory storage. The modified input block will be based on SRAM memory cells. SRAM memory cells provide the fastest and most compact means of on-chip storage. DRAM is much more compact but suffers in terms of speed due to the constant refresh signals required to maintain memory content. In high performance CPU architectures, memory is implemented as on-chip SRAM to achieve the best possible performance.

The function of the scheduler is to arbitrate between conflicting requests for access to the crossbar switch shared medium. The existing scheduler architecture is based on a symmetric implementation of round-robin like algorithm requiring one set of grant arbiters and one set of accept arbiters to perform arbitration. The modified design uses the concept of folding to reduce the area of the scheduler by removing one set of arbiters and using the remaining set of arbiters to perform both grant and accept arbitration in a time multiplexed fashion.

The design of the modified routing node is implemented using standard cell based VLSI flow with provision for custom IP core inclusion. The Synopsis tool chain is used to implement the design from RTL coding to synthesis and place and route.

Design verification is carried out using hierarchical functional simulation at each level of the design flow. Also, static timing analysis is used to verify timing closure in the final design layout.

Area and power are two important parameters which need to be optimized for better NoC performance. The NoC consists of three basic components which are the routing node, the routing links, and network interfaces. Optimization of the routing nodes will lead to improvement in the area and the power requirements of the NoC, as it is the most redundant component which lies in association with every processing element in SOC. Thus, the aim of this work is to present a modified architecture of the routing node to achieve higher area and power efficiency using changes at the RTL architecture level and use of custom IP to boost the performance of standard-cell based ASIC design.

5. The Proposed Router Architecture

The routing node configuration shown in Figure 4 is 4x4. It is based on a 2D mesh NoC topology where each routing node is connected to four other routing nodes. The NOC infrastructure includes components responsible for packetization, transmission, and de-packetization of data. These components, respectively, are the NI, the VC router, and the links. These components are repeated for every grid element in NOC. So, if we consider a NOC with 3×3 mesh network, then it will have nine sets of components of NI, VC router and links. It can be clearly seen that these components will occupy a significant amount of silicon space on the chip and therefore the cost and the power consumption of the chip would increase. However, it must be noted that serial packet-based communication will still remain an optimum solution as compared to a bus-based system in terms of the power consumption and will reduce the cost of system design in the longer run due to the potential for reuse.

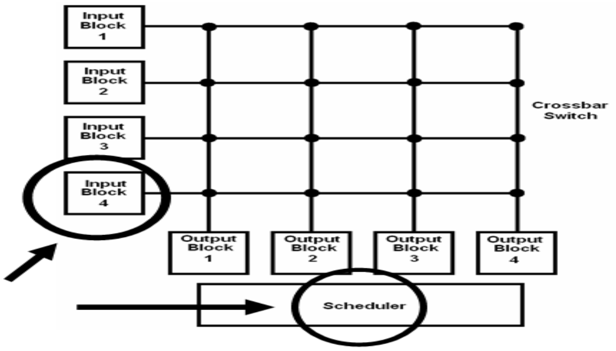


Figure 4. 4x4 Routing Node

6. Area of Focus

The design of the proposed router has been carried out as follows:

6.1 . Proposed Switching Technique

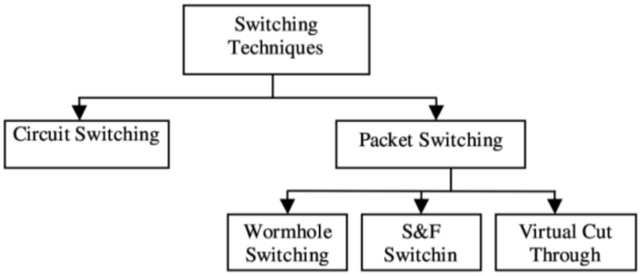


Figure 5. Switching Techniques

Switching techniques can be classified based on network characteristics. Circuit switched networks reserve a physical path before transmitting the data packets, while packet switched networks transmit the packets without reserving the entire path. Packet switched networks can further be classified as Wormhole, Store and Forward (S&F), and Virtual Cut through Switching (VCT) networks as shown in Figure 5. In Wormhole switching networks, only the header flit experiences latency. Other flits belonging to the same packet simply follow the path taken by the header flit. If the header flit is blocked then the entire packet is blocked. It does not require any buffering of the packet. Therefore, the size of the chip drastically reduces. However, the major drawback of this switching technique is a higher latency. Thus, it is not a suitable switching technique for real-time data transfers.

S&F switching forwards a packet only when there is enough space available in the receiving buffer to hold the entire packet. Thus, there is no need for dividing a packet into flits. This reduces the overhead, as it does not require circuits such as a flit builder, a flit decoder, a flit stripper and a flit sequencer. Store and forward is the easiest policy in terms of implementation complexity. So this implementation is based on store and forward switching.

6.2. Proposed Flow Control Mechanism

Flow control determines how network resources, such as channel bandwidth, buffer capacity, and control state, are allocated to a packet traversing the network. The flow control may be buffered or buffer less as shown in Figure 6. The Buffer less Flow Control has more latency and fewer throughputs than the Buffered Flow Control. The Buffered Flow Control can be classified further as:

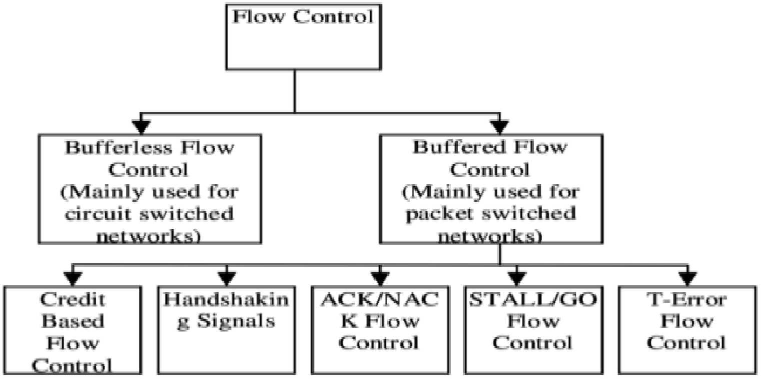

Flow Control
Bufferless Flow Control (Mainly used for circuit switched networks)
Buffered Flow Control (Mainly used for packet switched networks)
Credit Based Flow Control
Handshaking Signals
ACK/NACK Flow Control
STALL/GO Flow Control
T-Error Flow Control

Figure 6. Flow Control Techniques

In Credit Based Flow Control, an upstream node keeps count of data transfers, and thus the available free slots are termed as credits. Once the transmitted data packet is either consumed or further transmitted, a credit is sent back and used [15, 16].

In Handshaking Signal Based Flow Control, a VALID signal is sent whenever a sender transmits any flit. The receiver acknowledges by asserting a VALID signal after consuming the data flit and used in SoCIN NOC implementation [17].

To minimize the chances of dropped packets at the receiving end, the credit based flow control mechanism has been incorporated wherein only those output IP blocks take part in the scheduling that has some credit. In addition to this, every input block maintains packet array and the linked list array to maintain the proper flow so as to avoid the out of order delivery.

6.3. Proposed Buffer Implementation in the Design of Router

A higher buffer capacity and a larger number of virtual channels in the buffer will reduce network contention, thereby reducing latency. However, buffers are area hungry, and their use needs to be carefully directed [18, 19] therefore proposed a simple implementation of a buffer architecture for NOC buffers using 0.18 μm technology to estimate the cost and area of buffers needed for NOC. Also proposed the trade-off between buffer size and channel bandwidth to secure constant latency and concluded that increasing the channel bandwidth is preferable to reducing the latency in NOC.

The input block consists of six major components: the packet array, the linked list array, the destination head array, the destination tail array, the free-list FIFO, and a shift register. Four of these six components are conventional memory elements. In a standard cell based design, memory elements are realized using D flip flops in the standard SYNOPSYS Library. If we consider a NAND gate implementation of a D flip flop with no RESET or SET inputs it requires 28 MOS transistors to realize one D flip flop [20]. A more area efficient implementation of memory is through the use of SRAM cells. Each SRAM cell is implemented using 6 transistors. Therefore, memory realization using SRAM is more efficient compared to D flip flops. However, standard cell based approach to ASIC design does provide SRAM standard cells because of the many possible configurations of width and depth. SRAM design is carried out using full custom approach to ASIC design as shown in Figure 7. By combining standard cell based and full custom ASIC design, D flip flops can be replaced by SRAM, improving the area efficiency of the input block. Full custom design of SRAMs has been carried out by MILKYWAY of SYNOPSY, while physical implementation of the input module with SRAMs has been carried out by IC Compiler of SYNOPSY.

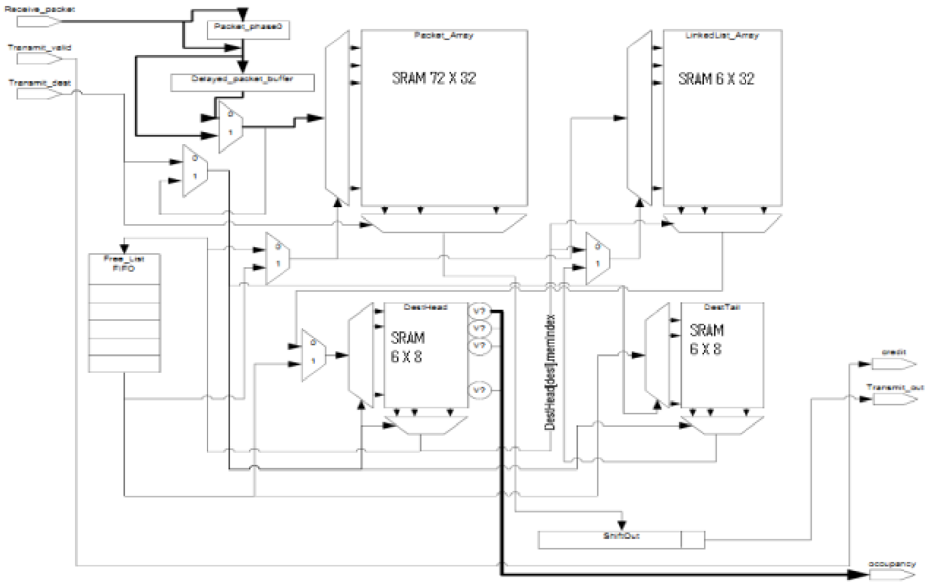


Figure 7. Input Module with SRAM based Arrays

6.4. Proposed Scheduler in the Design

The scheduler was modified using a folding approach due to the regular structure and placement of the arbiters. The modified scheduler is as shown in Figure 8. Each arbiter in the modified scheduler now has to generate both grant and accept signals in a time multiplexed fashion. The arbiter is modified to hold both grant and accept pointers for successive time slots.

The proposed scheduler belongs to a Router in 2D Mesh NOC design. So here the value of N is 4.

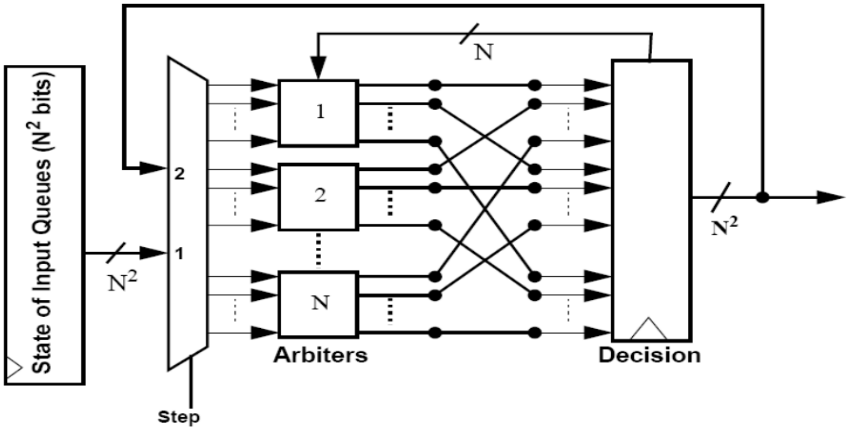


Figure 8. Modified Scheduler

7. Introduction to Clock Gating

Clock gating [21-24] applies to synchronous load-enable registers, which are groups of flip-flops that share the same clock and synchronous control signals and that are inferred from the same HDL variable. Synchronous control signals include synchronous load-enable, synchronous set, synchronous reset, and synchronous toggle. The registers are implemented by Design Compiler by use of feedback loops. However, these registers maintain the same logic value through multiple cycles and unnecessarily use power. Clock gating saves power by eliminating the unnecessary activity associated with reloading register banks. Designs that benefit most from clock gating are those with low-throughput data paths. Designs that benefit less from RTL clock gating include designs with finite state machines or designs with throughput-of-one data paths.

Power Compiler allows performing clock gating with the following techniques [21]:

1. RTL-based clock gate insertion on unmapped registers. Clock gating occurs when the register bank size meets certain minimum width constraints.
2. Gate-level clock gate insertion on both unmapped and previously mapped registers. In this case, clock gating is also applied to objects such as IP cores that are already mapped.
3. Power-driven gate-level clock gate insertion, which allows for further power optimizations because all aspects of power savings, such as switching activity and the flip-flop types to which the registers are mapped, are considered.

Without clock gating, Design Compiler implements register banks by using a feedback loop and a multiplexer. When such registers maintain the same value through multiple cycles, they use power unnecessarily.

Figure 9 shows a simple register bank implementation using a multiplexer and a feedback loop.

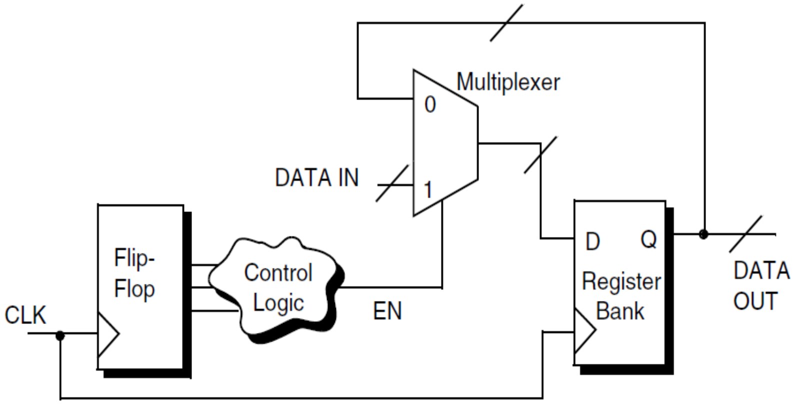


Figure 9. Synchronous Load-Enable Register With Multiplexer

The multiplexer also consumes power. Clock gating eliminates the feedback net and multiplexer shown in Figure 9 by inserting a 2-input gate in the clock net of the register. Clock gating can insert inverters or buffers to satisfy timing or clock waveform polarity requirements. The 2-input clock gate selectively prevents clock edges, thus preventing the gated-clock signal from clocking the gated register.

Figure 10 shows a latch-based clock-gating style using a 2-input AND gate, however, depending on the type of register and the gating style, gating can use NAND, OR, and NOR gates instead.

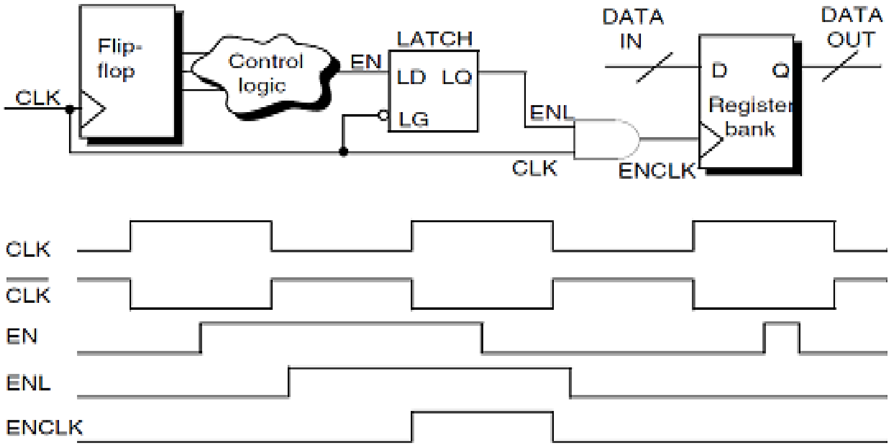


Figure 10. Latch-Based Clock Gating

Clock gating reduces the clock network power dissipation, relaxes the datapath timing, and reduces routing congestion by eliminating feedback multiplexer loops. For designs that have large multi-bit registers, clock gating can save power and reduce the number of gates in the design. However, for smaller register banks, the overhead of adding logic to the clock tree might not compare favorably to the power saved by eliminating a few feedback nets and multiplexers.

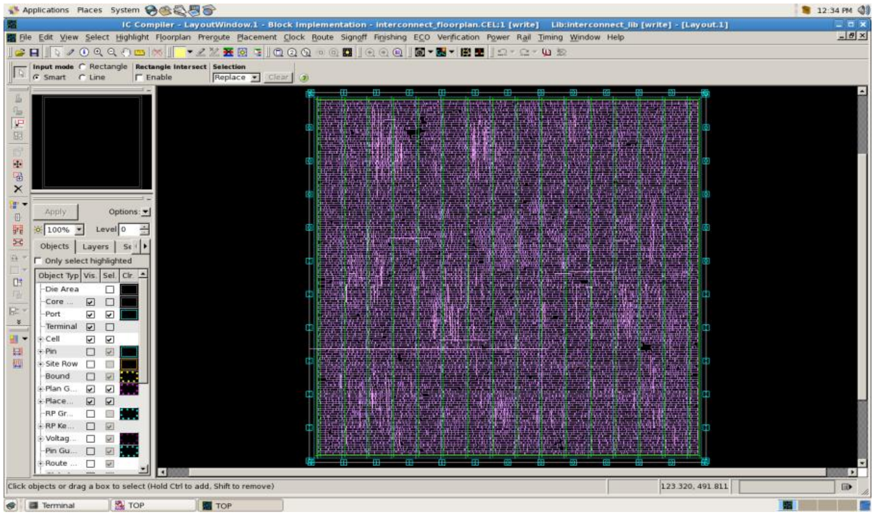
Table 1. Comparative Results of Proposed Router with Existing Router Design

Network	Topo-logy	Flit Size in bits	Ports	Buf Size in flits	Tech in nm	L in Clk	A in Sq.mm	F in MHz
Teraflops	Mesh	32	4	16	65	5	0.34	4270
Xpipes	Custom	32	4	--	100	7	--	--
Dally	Torus	256	5	4	100	3	--	200 - 2000
HIBI	Bus	32	2	2,8	130	4	0.03 -0.05	435
Octagon	Ext. Ring	32	4	2,8	130	4	0.04 - 0.09	435
SPIN	Fat- T	16	8	8	130		0.24	200
Aethereal	Mesh	96	5	8	120		0.26	500
ANoC	Mesh	32			130		0.25	500
Mango	Mesh	32	5	1			0.19	795
Hermes	Mesh	32	5	2,8	130	10	0.05-0.11	435
SoCBus	Custom	16	3	1	180			
ASoC	Mesh	32	4	2	180		0.04-0.08	400
Avg.		**50.1**	**4.8**	**6.4**	**170**	**5.2**	**0.14-0.22**	**328-596**
Present Work	**Mesh**	**32**	**4**	**8**	**90**	**4**	**0.15**	**500**

Legends used in above Table:-, Buf.-Buffer, Tech-Technology, L-Latency, A-Area, F-Frequency, Ext-Extended, R-Ring, , T-Tree Cus- Custom, Avg-Average,Pre-Present.

8. Experimental Results 1: Physical Implementation

8.1. 4x4 Routing Nodes D Flip Flop (DFF) _Physical Implementation



IC Compiler - LayoutWindow.1 - Block Implementation - interconnect_floorplan.CEL;1 [write]   Lib:interconnect_lib [write] - [Layout.1]

Input mode ● Smart ○ Rectangle ○ Line

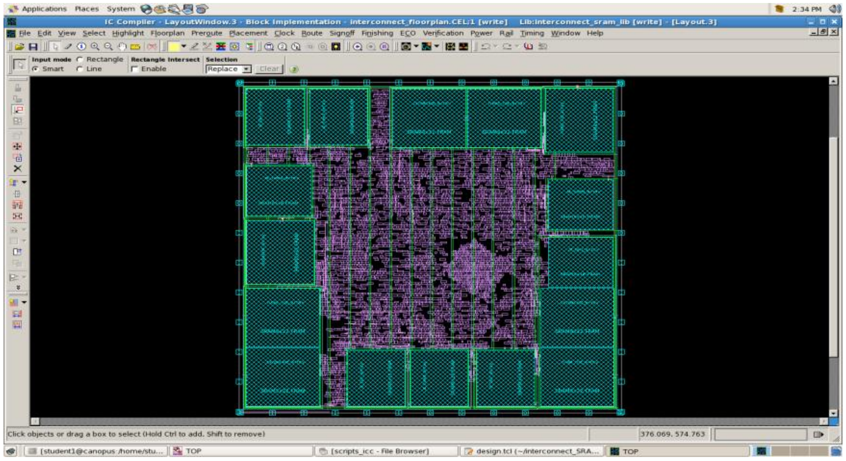
Rectangle Intersect □ Enable

Selection: Replace

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

123.320, 491.811

8.2. 4x4 Routing Node (SRAM)_Physical Implementation


Applications Places System
2:34 PM
IC Compiler - LayoutWindow.3 - Block Implementation - interconnect_floorplan.CEL;1 [write] Lib:interconnect_sram_lib [write] - [Layout.3]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Input mode ○ Rectangle ● Smart ○ Line
Rectangle Intersect □ Enable
Selection Replace Clear
376.069, 574.763
[student1@canopus:/home/stu... TOP [scripts_icc - File Browser] design.tcl (~/interconnect_SRA... TOP

Click objects or drag a box to select (Hold Ctrl to add, Shift to remove)

Table 2. Comparisons Showing Result of Area, Power of 4x4 Routing Node at Post Synthesis

# Clock Period: 16ns.

Post Synthesis	DFF based	SRAM based	% Reduction in SRAM based Method
Area	202704 sq um	150314 sq um	25.85%
Power	10.7 mW	3.9 mW	63.55%

Table 3. Comparisons Showing Results of Area, Power of 4x4 Routing Node at Post Layout

# Clock Period: 16ns.

Post Layout	DFF based	SRAM based	% Reduction in SRAM based Method
Area	205298 sq um	150756 sq um	26.567%
Power	14.29 mW	4.98 mW	65.15%

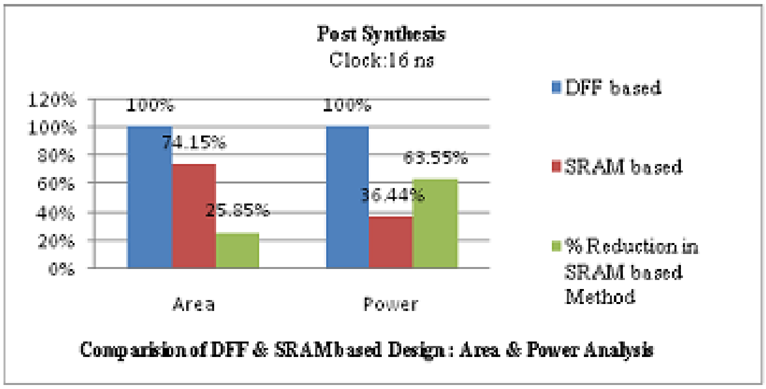


Figure 11. Comparison of DFF & SRAM Based Designs at Post Synthesis

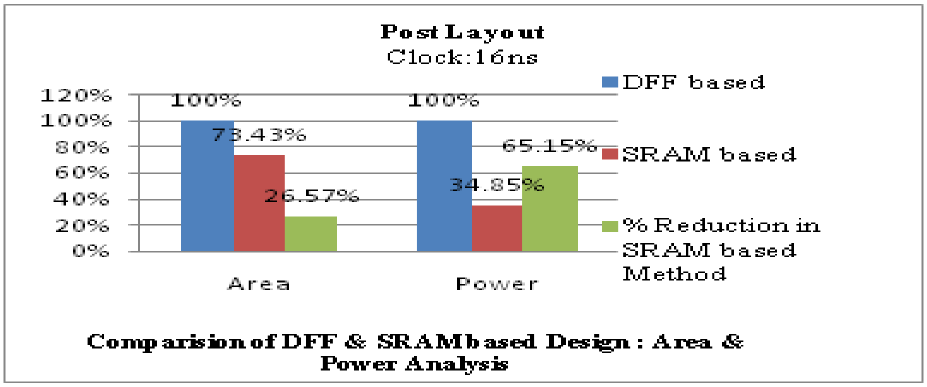


Figure 12. Comparison of DFF & SRAM Based Designs at Post Layout

9. Insertion of Clock Gating Techniques

9.1. Inserting Clock Gates in the RTL Design

To insert clock gating logic in RTL design and to synthesize the design with the clock-gating logic:

1. Read the RTL design.
2. Use the compile_ultra -gate_clock command to compile design.

To apply this method, the compiler executes the following tcl commands:

# Optional setting

```
read_verilog interconnect.v
create_clock -period 16 -name CLK
compile_ultra -gate_clock -scan
insert_dft
report_clock_gating
report_power
```

9.2. Inserting Clock Gates in Gate-Level Design

To insert clock gating logic in gate-level netlist and to re-synthesize the design with the clock gating logic at Gate Level Netlist:

1. Read the gate-level netlist.
2. Use the compile_ultra -gate_clock command to compile your design.

To apply this method, the compiler executes the following tcl commands:

# Optional setting

read_ddc interconnect.ddc
compile_ultra -incremental -gate_clock -scan
insert_dft
report_clock_gating
report_power

9.3. Power-Driven Clock Gating

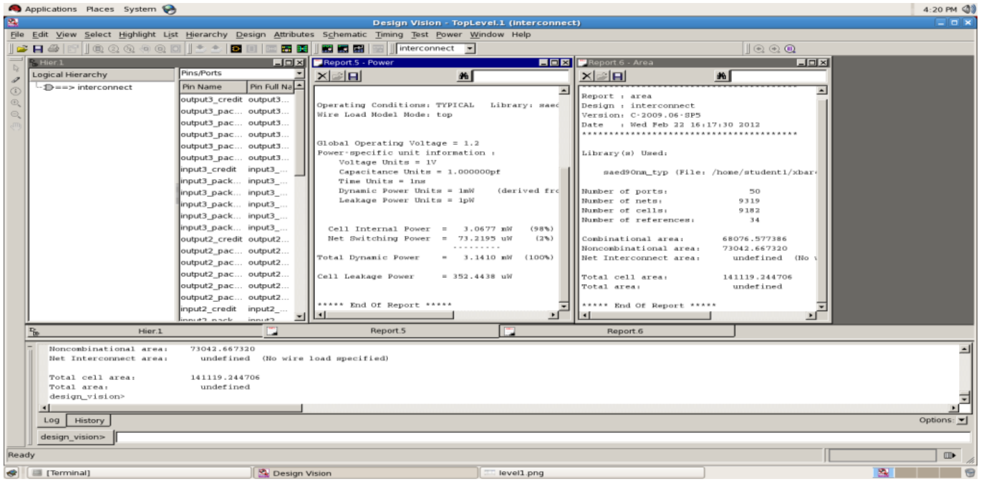
To apply this method, the compiler executes the following tcl commands:

# Optional setting

set_clock_gating_style -positive_edge_logic integrated
read_verilog {scheduler.v input_block.v interconnect.v}
Current_design interconnect link
create_clock -period 16 clk -name CLK
compile_ultra -gate_clock -incremental
insert_dft
report_clock_gating
report_power

10. Experimental Results 2: Using Clock Gating Techniques

10.1. Power Analysis at RTL Level

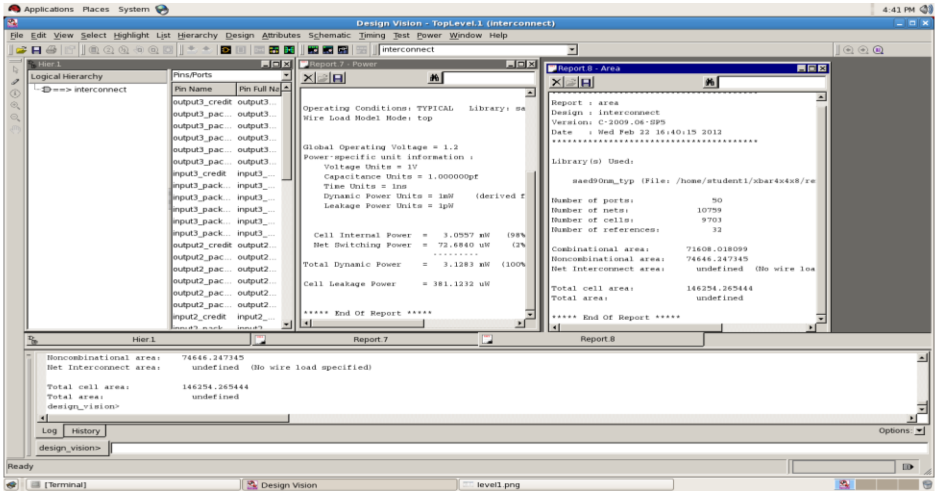


```
Operating Conditions: TYPICAL   Library: saed
Wire Load Model Mode: top


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived fro
    Leakage Power Units = 1pW


  Cell Internal Power  =   3.0677 mW   (98%)
  Net Switching Power  =  73.2195 uW    (2%)
                         ---------
Total Dynamic Power    =   3.1410 mW  (100%)

Cell Leakage Power     = 352.4438 uW


***** End Of Report *****
```

```
Report : area
Design : interconnect
Version: C-2009.06-SP5
Date   : Wed Feb 22 16:17:30 2012
****************************************


Library(s) Used:

    saed90nm_typ (File: /home/student1/xbar

Number of ports:                   50
Number of nets:                  9319
Number of cells:                 9182
Number of references:              34

Combinational area:       68076.577386
Noncombinational area:    73042.667320
Net Interconnect area:      undefined  (No w

Total cell area:          141119.244706
Total area:                 undefined

***** End Of Report *****
```

```
Noncombinational area:    73042.667320
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          141119.244706
Total area:                 undefined
design_vision>
```

10.2. Power Analysis at Gate Level

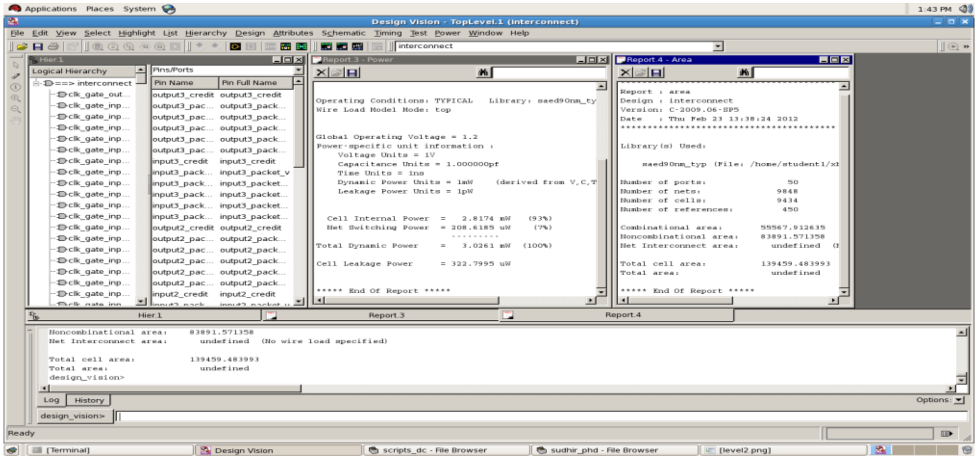

Design Vision - TopLevel.1 (interconnect)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power Window Help
interconnect
Report.7 - Power
Report.8 - Area
Hier.1
Report.7
Report.8
Log
History
Options:
design_vision>
Ready
[Terminal]
Design Vision
level1.png

```
Operating Conditions: TYPICAL   Library: sa
Wire Load Model Mode: top


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived f
    Leakage Power Units = 1pW


  Cell Internal Power  =   3.0557 mW   (98%
  Net Switching Power  =  72.6840 uW    (2%
                         ---------
Total Dynamic Power    =   3.1283 mW  (100%

Cell Leakage Power     = 381.1232 uW


***** End Of Report *****
```

```
Report : area
Design : interconnect
Version: C-2009.06-SP5
Date   : Wed Feb 22 16:40:15 2012
****************************************

Library(s) Used:

    saed90nm_typ (File: /home/student1/xbar4x4x8/re

Number of ports:                 50
Number of nets:               10759
Number of cells:               9703
Number of references:            32

Combinational area:       71608.018099
Noncombinational area:    74646.247345
Net Interconnect area:      undefined  (No wire loa

Total cell area:          146254.265444
Total area:                 undefined

***** End Of Report *****
```

```
Noncombinational area:    74646.247345
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          146254.265444
Total area:                 undefined
design_vision>
```

10.3. Power Analysis using Power Driven


Design Vision - TopLevel.1 (interconnect)
File Edit View Select Highlight List Hierarchy Design Attributes Schematic Timing Test Power Window Help
interconnect
Report.3 - Power
Report.4 - Area
Hier.1
Report.3
Report.4
Log
History
Options:
design_vision>
Ready
[Terminal]
Design Vision
scripts_dc - File Browser
sudhir_phd - File Browser
[level2.png]

```
Operating Conditions: TYPICAL   Library: saed90nm_ty
Wire Load Model Mode: top


Global Operating Voltage = 1.2
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T
    Leakage Power Units = 1pW


  Cell Internal Power  =   2.8174 mW   (93%)
  Net Switching Power  = 208.6185 uW    (7%)
                         ---------
Total Dynamic Power    =   3.0261 mW  (100%)

Cell Leakage Power     = 322.7995 uW


***** End Of Report *****
```

```
Report : area
Design : interconnect
Version: C-2009.06-SP5
Date   : Thu Feb 23 13:38:24 2012
****************************************

Library(s) Used:

    saed90nm_typ (File: /home/student1/xb

Number of ports:                 50
Number of nets:                9848
Number of cells:               9434
Number of references:           450

Combinational area:       55567.912635
Noncombinational area:    83891.571358
Net Interconnect area:      undefined  (N

Total cell area:          139459.483993
Total area:                 undefined

***** End Of Report *****
```

```
Noncombinational area:    83891.571358
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:          139459.483993
Total area:                 undefined
design_vision>
```

10.4. Comparisons Chart Showing Result of Area & Power

# Clock Period: 16 ns

	DFF based	**RTL C_G**	**Gate-Level C_G**	**Power driven C_G**
Area	202704 Sq um	141119 Sq um	146254 Sq um	139459 Sq um
Power	10.7 mW	3.49 mW	3.50 mW	3.34 mW

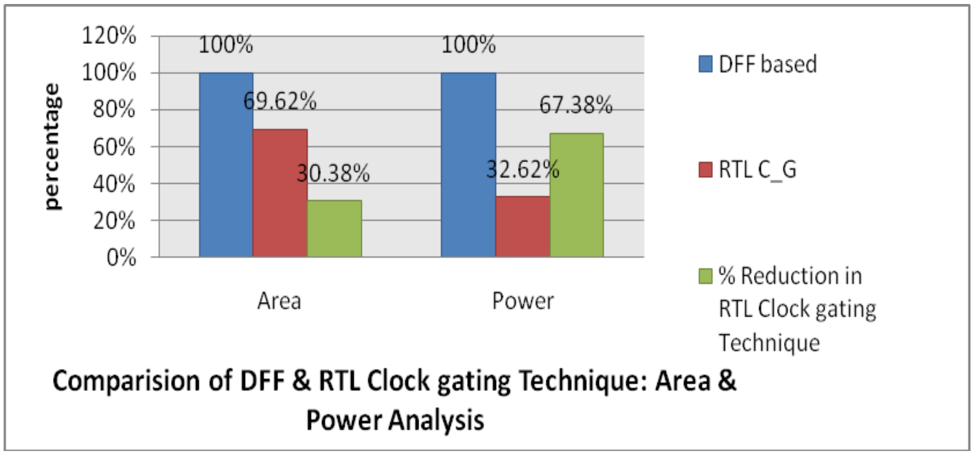


Figure 13. Comparison of DFF & RTL Clock Gating

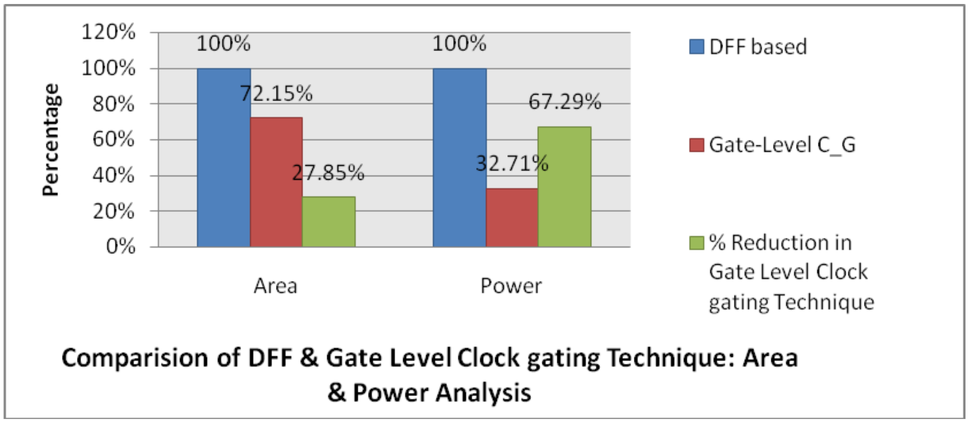


Figure 14. Comparison of DFF & Gate Level Clock Gating

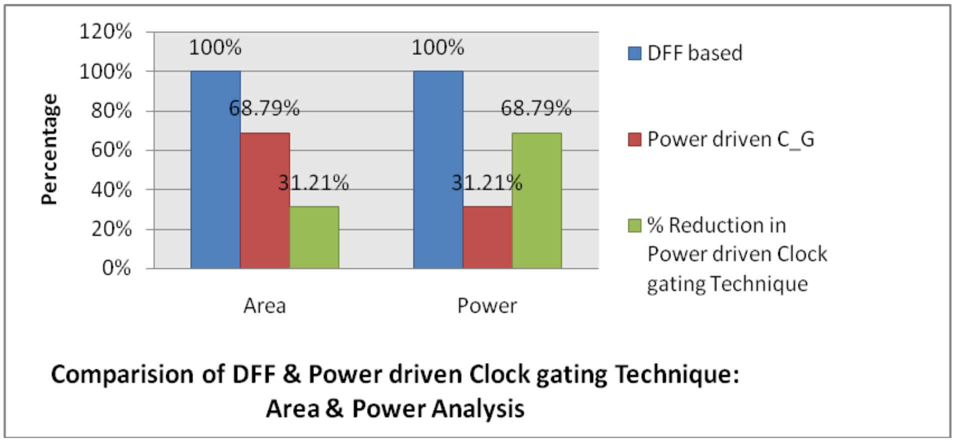


Figure 15. Comparison of DFF & Power Driven Clock Gating

11. Conclusion

2D-Mesh has been an area efficient implementation of a routing node for an NOC is demonstrated. Of the four components of routing node, the input block (mainly consisting of buffers) and scheduler have been modified to save area requirements. The other two components of the routing node take up negligible area in comparison. The use of custom SRAM in place of synthesizable flip flops in the input block has resulted in a saving of over 26% of the silicon area and power optimization is 65% when operated at 16 ns clock.

Clock gating is an important high-level technique for reducing the power consumption of a design. Clock gating reduces the clock network power dissipation, relaxes the datapath timing, and reduces routing congestion by eliminating feedback multiplexer loops. For designs that have large multi-bit registers, clock gating can save power and reduce the number of gates in the design. In our design case, it has been further observed that the power optimization with clock gating techniques at RTL level saves 67.38%, Gate Level 67.29% & Power Driven around 68.79 of power while 30.38 %, 27.85 % &31.21% silicon area respectively have been saved.

12. Future Scope

The existing design has been synthesized using MUX as a crossbar switch more Area efficient crossbar like tristate and sense amps can further make the design efficient.

We foresee that Area & Power of existing design can be further optimized by Low Power Design Methodology Likes:

1. Multi VDD - Since dynamic power is proportional to VDD^2 lowering VDD on selected blocks helps reduce power significantly. Unfortunately, lowering the voltage also increases the delay of the gates in the design.

2. Multi Threshold Logic- As geometries have shrunk to 130nm, 90nm, and below, using libraries with multiple V_T has become a more efficient way of reducing leakage voltage.

References

[1] K. Latif, T. Seceleanu and H. Tenhunen, “Power and Area Efficient Design of Network-on-Chip Router through Utilization of Ideal Buffer”, Proc.17th IEEE International Conference and workshop on Engineering of Computer based System**, (2010).**

[2] L. Benini and G. De Micheli, “Networks on Chips, Morgan”, Kaufmann Publishers, **(2006).**

[3] W. Hangsheng, L. S. Peh and S. Malik, “Power- driven design of router micro architectures in on-chip networks”, Proc. of the 36th Annual IEEE/ACM International Symposium on Micro architecture (MI-CRO), **(2003)**, pp. 105-116**.**

[4] X. Chen and L. S. Peh, “Leakage power modeling and optimization of interconnection networks”, Proc. of International Symposium on Low Power Electronics and Design, **(2003)**, pp. 90-95**.**

[5] N. Banerjee, P. Vellanki and K. S. Chatha, “A Power and Performance Model for Network-on-Chip Architectures”, Proc. of the conference on Design, automation and test in Europe (DATE), vol. 2, **(2004)**, pp. 1250-1255.

[6] ARTISAN, “A comparison of network-on-chip and busses”, white paper, **(2005).**

[7] W. J. Dally and B. Towles, “Principles and practices of interconnection networks”, Morgan Kaufmann Publishers, **(2004).**

[8] T. Bartic, “Topology adaptive network-on-chip design and implementation”, IEEE Proc. Comput. Digit Tech., vol. 152, no. 4, **(2005)** July, pp. 467–472.

[9] E. Beigne, “An asynchronous NOC architecture is providing low latency service and its multi-level design framework”, in ASYNC, **(2005)** March, pp. 54–63.

[10] L. Benini and G. de Micheli, “Networks on chips: A new SoC paradigm”, IEEE Computer, vol. 35, no. 1, **(2002)** January, pp. 70–78.

[11] J. Owens, et. al., “Research challenges for on-chip interconnection networks”, IEEE Micro, vol. 27, no. 5, **(2007)** September-October, pp. 96–108.

[12] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of network-on-chip”, ACM Computing Surveys, vol. 38, no. 1, **(2006)** June, Article no. 1.

[13] T. Bjerregaard and J Sparose, “A router architecture for connection oriented service guarantees in the MANGO clockless network-on-chip”, *in DATE*, vol. 2, **(2005)** March, pp. 1226-1231.

[14] C. Wang, et. al., “Area and power efficient innovative NoC Architecture”, Proc. of 18thEuroMicro International Conference, PDP 2010, Pisa, Italy, **(2010).**

[15] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, “QNoC: QoS architecture and design process for network on chip”, Journal of Systems Architecture, vol. 50, Issue 2-3 (Special Issue on Network on Chip), **(2004)** February, pp. 105-128.

[16] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, “Cost considerations in Network on Chip Integration”, The VLSI Journal, no. 38, **(2004)**, pp. 19-42.

[17] C. A. Zeferino and A. A. Susin, “SoCIN: A parametric and scalable network-on- chip”, Proc. 16th Symposium on Integrated Circuits and Systems Design, **(2003)**, pp. 169-175**.**

[18] H. Zimmer, S. Zink, T. Hollstein and M. Glesner, “Buffer-architecture exploration for routers in a hierarchical network-on-chip”, Proc. 19th IEEE International Symposium on Parallel and Distributed Processing, **(2005)** April, pp. 1-4.

[19] E. Bolotin, A. Morgenshtein, I. Cidon, R. Ginosar and A. Kolodny, “Automatic hardware-efficient SoC integration by QoS Network-on-Chip”, Proc. 11th International IEEE Conference on Electronics, Circuits and Systems, **(2004)**, pp. 479-482**.**

[20] S. M. Alho and J. Nurmi, “Buffer implementation for Proteome network-on-chip”, International IEEE Proceeding on Circuits and Systems, vol. 2, **(2003)** May, pp. 113-116.

[21] N. S. Shelke and P. B. Patil, “Impact of SoC Low Power Design, Power Management Techniques & Physical Challenges”, International Journal of Electrical, Electronics &Computing Technology (IJEECT), vol. 1, Issue 2, **(2011)** January-April.

[22] Power Compiler User Guide, SYNOPSIS Version F 2011.9, **(2011)** December**.**

[23] M. Keating, “The Future of Low Power”, SNUG, **(2007),**

[24] M. Keating, D. Flynn, R. Aitken, A. Gibbons and K. Shi, “Low Power Methodology Manual: For System-on-Chip Design”, Springer, **(2007).**

Authors

Sudhir Shelke was born in Nagpur, India in Jun 1976. He has received Degree of Engineering in 2000 in Electronics Engineering from SRTM University, Nanded, India and He has obtained Master of Engineering in 2007 in Digital Electronics from SGB Amravati University, India. He is currently working as an Assistant Professor in Department of Electronics & Telecommunication Engineering, J D College of Engineering, Nagpur, His principal research area includes VLSI Chip Design, Embedded System Design.etc. His undergoing Research is on Low Power VLSI Designs Techniques & Physical Challenge for his doctorate Program. He has Published Twelve papers in International journals, International Conferences & National Conferences so far. His one of the paper is awarded as the best paper of the session of International Conference ICSCI-2008 Hyderabad, India. He had been worked on PCI Express 3rd Generation I/O Interconnect.

Pramod B. Patil was born in Amravati, India in May 1968. He has received Degree of Engineering in 1989 in Electronics Engineering from SGB Amravati University, India and he has obtained Master and Doctorate in Engineering in 1997 & 2007 respectively. Currently working as a Principal & Professor in J.D. College of Engineering, Nagpur. He has More than 20 years of Teaching & Research experience. His principal research area includes VLSI Chip Design, Digital Image Processing, & Signal Processing etc. He is reviewer of IEEE transaction of Signal, Speech & Audio processing & associate editor of two international Journals, He has more than 37 research papers in International Journals &International conferences on his credit so far and many research scholars are working under his esteemed guidance.