

Delay and Power Reduction in New Routing Fabrics

¹Vijayakumar, S., ²J. Sundararajan, ³P. Kumar and ⁴K. Nithya

¹Department of Electronics and Communication Engineering,
Paavai Engineering College, Namakkal, TN, India

²Paavai College of Technology, Namakkal, TN, India

³Department of Electronics and Communication Engineering,
K.S. Rangasamy College of Technology, Tiruchengode, TN, India

⁴Department of Electronics and Communication Engineering,
Paavai College of Engineering, Namakkal, TN, India

Received 2013-07-15, Revised 2013-09-30; Accepted 2013-10-23

ABSTRACT

In this study we created a new routing fabric for reducing power and delay. The power consumed in a FPGA core consists of both static and dynamic components. Static power contributes only 10% of the total power consumed in a FPGA. On the other hand, dynamic power contributes over 90% of the total power consumed and it is the main source for their power inefficiency. By reducing net length and/or programming overhead the power consumption reduced. Routed net length reduced by using short intersects segments in the routing channels. By decreasing the switch box and/or connection box flexibilities programming overhead reduced. In this study, we concentrated on achieving 1.80 times lower consumption of dynamic power and 1.50 times less significant average net delays by re-architecting the programmable routing fabrics such that both routed net lengths and programming overhead reduced without adversely affecting delay.

Keywords: Routing Fabrics Logic Block, Routed Segment, Glitches, Dynamic Power, Average Net Delay

1. INTRODUCTION

Power consumption is an important factor of designing integrated circuits. FPGA are much less power efficient when compared with cell-based ASIC. This power inefficiency limited application of FPGA in low power area. But FPGA has advantage that well suited to changing need and short design cycles. Hence reducing power consumption is important in FPGA. Dynamic power consumption caused by signal alteration. Higher operating frequencies lead to increased transistor activity which means more dynamic power Dissipation. The largest source of dynamic power Consumption in a FGPA is from charging and discharging capacitor.

Signal transitions which directly determine dynamic power classified into two types they are functional transition and spurious transitions or glitches. Functional transitions occur when there is a transition needed to do the logic function between two successive clock cycles. Glitches is short duration electrical pulse, usually it produces fault result particularly in a digital circuit. In FPGA, glitch power plays a major role of total dynamic power. Hence reducing glitches is important **Fig. 1**.

In this study, we concentrated on reducing glitch power by balancing the path to inputs of look up table. So that signals of the same look up table arrive at the same time and no glitches generated. Here we finding an alternative routes for early arriving signals so that the delays of the new routes causes the signals to arrive at the balanced times.

Corresponding Author: Vijayakumar, S., Department of Electronics and Communication Engineering,
Paavai Engineering College, Namakkal, TN, India

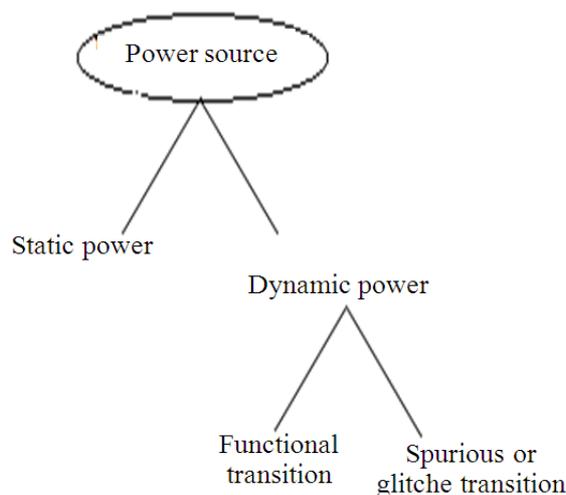


Fig. 1. Power consumption types

2. RELATED WORK

Several techniques proposed to shrink power and Delay which includes.

2.1. Torch (Lin and El Gamal, 2008)

Lin and El Gamal (2008) shown that TORCH based on simulated annealing procedure to find an optimized segmentation based on an average delay-power product. In each iteration, segmentation is incrementally changed, for new segmentation the benchmark designs routed into the FPGA using Versatile place route, the performance metric restructured and the new segmentation is either accepted or rejected. Because of infrequent placements, Run time is much condensed. TORCH outputs have an optimized mix of track segment lengths and an ordering of the segmented tracks in the channel.

2.2. Timing-Driven Routing Algorithm (Li *et al.*, 2003)

Li *et al.* (2003) revealed that every net in the circuit are repeatedly rips-up and re-routed and gradually resolves routing congestion by gradually increasing the cost of overused routing resources. The slack of each connection used to decide the congestion avoidance delay minimization trade-off to use for that connection.

2.3. Versatile Place Route Algorithm (Betz and Rose, 1997)

Betz and Rose (1997) exposed that the inputs to versatile place route consist of a technology routed net

list and a text file describing the FPGA architecture. Versatile place route can place a pre-existing placement, or the circuit. Versatile place route can then do either a global route or a detailed route of the placement. Versatile place route output consists of the placement and routing, as well as statistics details useful in assessing utility of a FPGA architecture, which includes routed wire length, track count and largest net length.

2.4. Voltage Scaling Algorithm (Li *et al.*, 2004)

Li *et al.* (2004) publicized that there are two types of routing tracks based on consumption of voltage: High tracks and Low tracks. Both tracks are differs from using switches. High tracks get high supply voltage and faster than the low tracks. The paths having zero values can use the faster High tracks and the other paths can use the slower. Low paths to save power.

2.5. Congestion/Delay Algorithm (Lin and El Gamal, 2009)

Lin and El Gamal (2009) revealed that initially, nets routed one at a time using the shortest path by considering interconnect segment or logic block pin overuse. Each iteration of the router consists of sequential net rip-up and reroute according to the lowest cost path available. The cost of using a routing resource is a current overuse and any overuse that occurred in preceding routing iterations. By gradually increasing the cost of an oversubscribed routing resource, the algorithm forces nets with alternative routes to avoid using that resource, leaving it to the net that most needs it.

2.6. Power-Delay Product (PDP) (Tuan *et al.*, 2006)

Tuan *et al.* (2006) shown that PDP persist to fall even below 0.8 V, where performance degradation becomes prohibitively large as well as reliability turn out to be a concern. therefore, considering performance, energy efficiency and reliability, they choose 1.0 V as core operating voltage. This leads to power reduction in every core blocks apart from for the configuration memory, which are disqualified since they can be more effectively addressed.

2.7. Power Optimization Techniques (Wang *et al.*, 2006)

This environment allows the enlargement and experimentation of power models, tracking dynamic power consumption during simulation and power estimation at the synthesis level, whereas providing an infrastructure to rapidly design and execute new power optimization algorithms. By using area minimization constraints a design is packed down more tightly in a agreed area of a chip. Net lengths are shortened and thus power is saved.

2.8. GlitchLess: An Active Glitch Minimization Technique (Lamoureux *et al.*, 2007)

Lamoureux *et al.* (2007) avoided the glitches by adding programmable delay elements within the logic blocks of the FPGA such that early arriving signals can be delayed so as to align the edges on each LUT input, thereby reducing number of glitches on the output of each LUT. By delaying the input signal, the output glitches can be abolished since only the early arriving signals are delayed, the overall critical path of the circuit is not increased.

2.9. Monolithically Stacked 3D FPGA (Lin *et al.*, 2006; Lin and El Gamal, 2007)

Lin *et al.* (2006) they believed that logic-density enhancement can be accomplished with the addition of only a small number of mask layers on peak of a standard CMOS technology, a monolithically stacked FPGA is expected to comprise lower manufacturing cost than an FPGA by way of the same logic capacity fabricated using only the standard CMOS technology. It is also anticipated that additional performance developments can be accomplished by re architecting the 3-D FPGA to receive full benefit of the extra layers.

Summary of Our Contribution:

- We propose the new routing fabric for reducing overall power and delay with the help of short segment
- Dynamic power reduced by reducing glitches through path balancing
- We describe algorithm to find a shortest path between source nodes and sink node with desired delay

The paper organized as follows section 3 provides new routing architecture. In section 4 we described the method of reducing the dynamic power. Section 5 we described about routing algorithm for finding shortest path and we compared and concluded the paper in section 6 and section 7 respectively.

3. ROUTING FABRICS FOR 3D FPGA

In new routing fabrics logic block merged and arranged in an array format with horizontal and vertical routing channel overlay. Routed net lengths reduced by using only short interconnect segment in the routing channels **Fig. 2**. Lin and El Gamal (2007) the routing block provides connectivity for logic block inputs and outputs as well as that it integrate the functions of connection and switch boxes. The routing points used to:

- Form local connections between neighboring logic blocks without going to channels
- Connect routing block inputs and outputs to channel segments and Chain channel segments together to form longer segments without entering routing blocks

3.1. Connection between Logical Block and Routing Block

Routing block perform function of connection and switch boxes. Logical block comprises of look up tables, flip-flops and programming overhead. Every routing block can connect to n_i . LB input such that each LB and routing block connected to bypass transistor switches. By choosing a value of n_i such that each LB input connect to the same number of routing block inputs. The loading on a routing block segment is lower than on a routing block input segment in the baseline fabric **Fig. 3**.

In addition to the connection through switch points, routing block architecture allows for extended switching width. In which a signal in a routing block looped back twice into it and exit to a perpendicular direction if it cannot do so directly. This extended switching much improves the efficiency of routing.

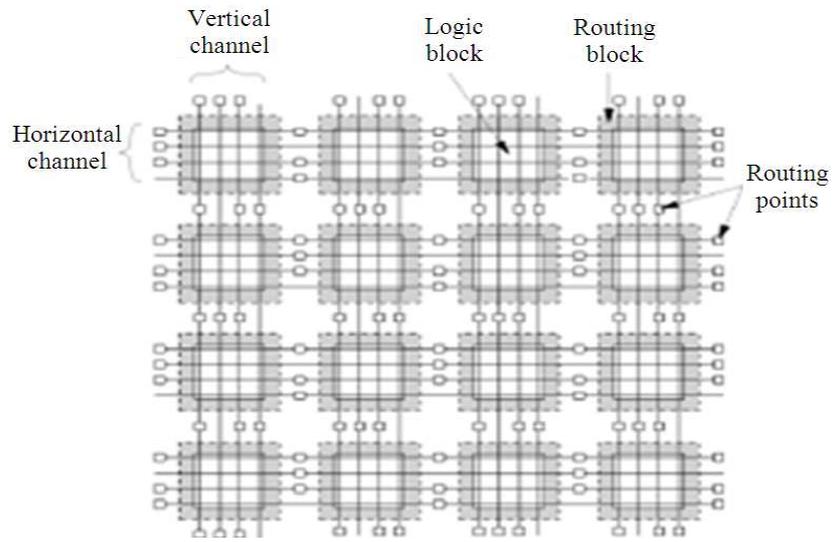


Fig. 2. Modified 2D FPGA

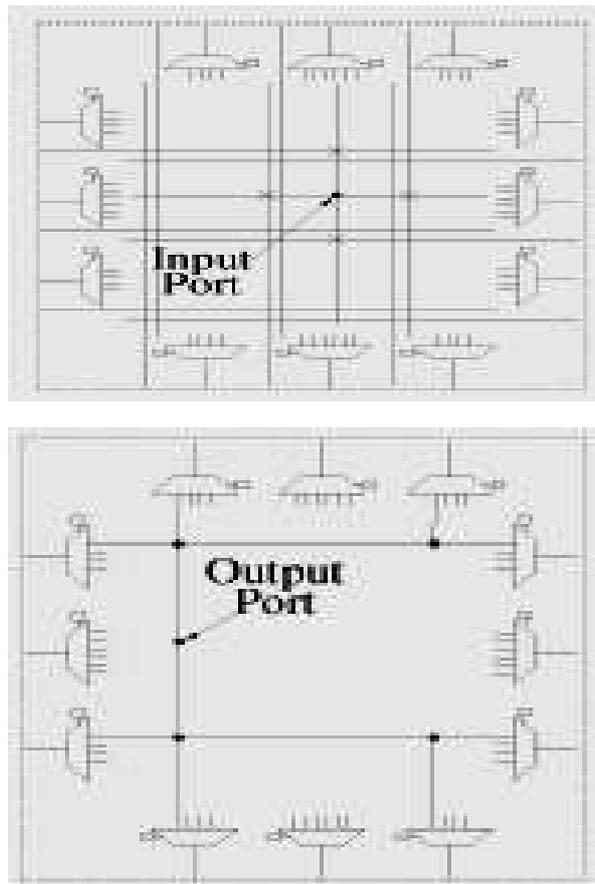


Fig. 3. Logical block input and output connected to routing block

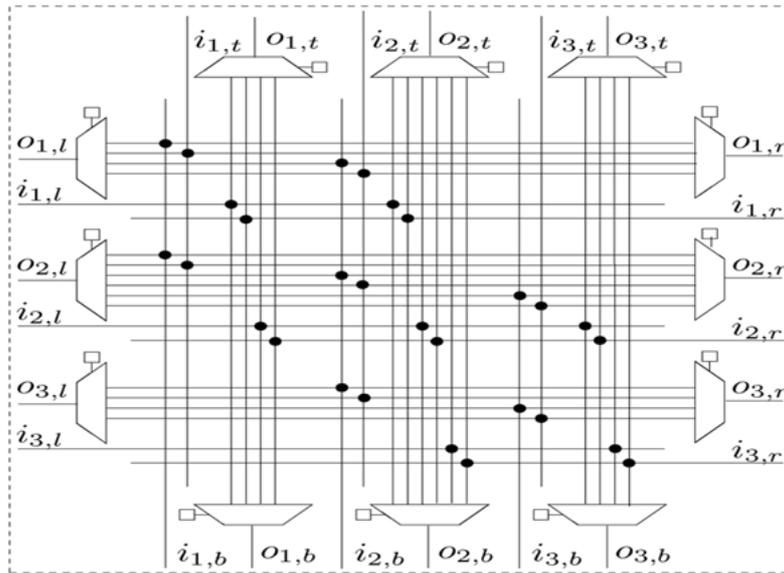


Fig. 4. Routing block

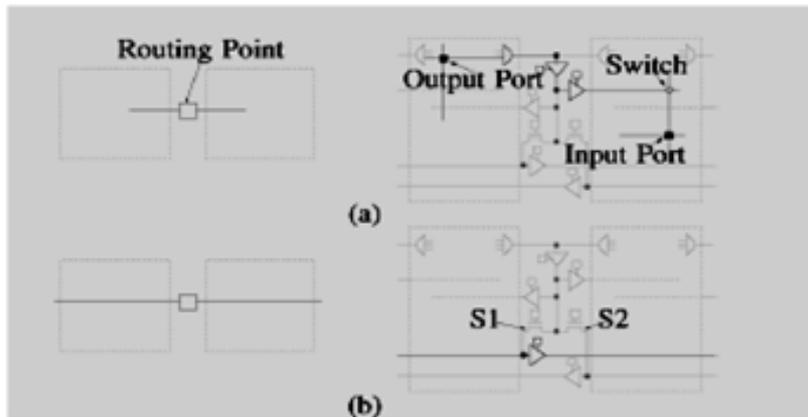


Fig. 5. (a) Local interconnect and (b) By-pass interconnect

3.2. Connection between Routing Block and Routing Channel Overlay

Every routing channel comprises of single and double segmented tracks. This segment has two unidirectional wires. The input and output connected by channel segment using the routing points. Segments joined together to form a longer segments, which called bypass interconnect shown in Fig. 4. The segments can also be connected via routing points to routing blocks to connect to LB inputs and outputs, make bends, or fan-out.

Two types of net connections: Local and bypass connection. In local connection output of LB are already routed inputs of its neighboring LB without using routing channel segment. In order to route a longer net without entering intermediate routing block.

3.3. Dynamic Power Consumption

In FPGA, glitches generated at the output of a LUT when signals transition takes place at different times. The pulse width of these glitches depends on how uneven the input signal arrival times are. Due to the limited connectivity of FPGA routing resources FPGA glitches are wider than ASIC glitches.

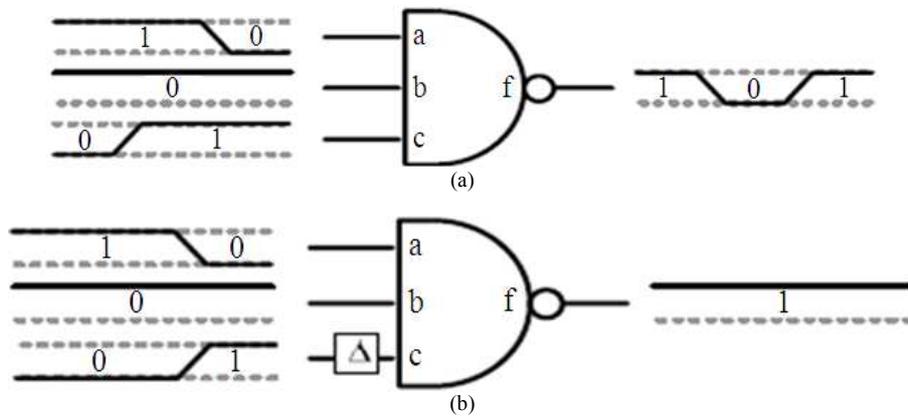


Fig. 6. Removing Glitches by delaying early arriving signal (a) Original circuit with glitch (b) Glitch removed by delaying input c

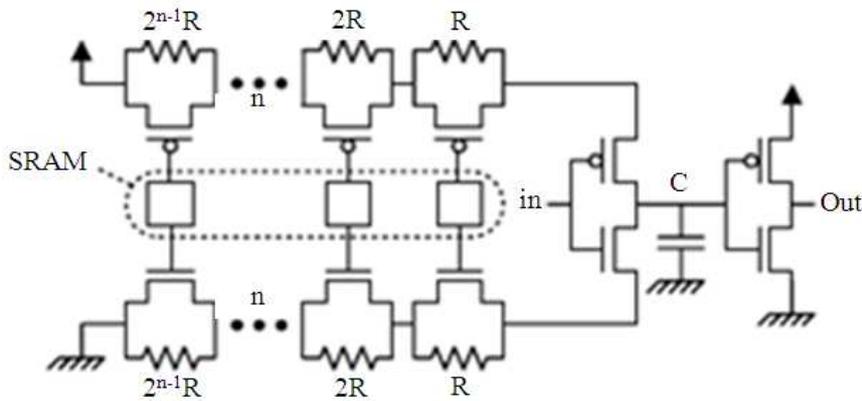


Fig. 7. Programmable delay elements

We avoided the glitches by adding programmable delay elements within the logic blocks of the FPGA such that we delay the early arriving signals to align the edges on each LUT input signals, thereby reducing some glitches on the output of each LUT.

The method demonstrated in Fig. 6 by delaying the input signal of c, the output glitches eliminated since only the early arriving signals delayed, the overall critical path of the circuit is not increased.

3.4. Programmable Delay Element

The delay element circuit consists of two inverters shown in Fig. 7. The first inverter composed of pull-up and pull-down resistor for controlling the delay of the circuit. The second inverter has large channel lengths to decrease short-circuit power. Both pull-up and pull-down resistor have n stages with a resistor and a bypass transistor which controlled by an SRAM

bit. Control bits used to double the value of resistor in later stages.

The control bit planned to produce any delay.

Figure 5a and b shows the by-pass connection implementation. The resource sharing increases ability of routing. In addition to that it will also decrease the reloading on by-pass interconnect note that buffers alone turned on. This again reduces the loading on the connection, thus power consumption and its delay reduced.

4. DYNAMIC POWER CONSUMPTION

In FPGA, glitches generated at the output of a LUT when signals transition takes place at different times. The pulse width of these glitches depends on how uneven the input signal arrival times are. Due to the limited connectivity of FPGA routing resources FPGA glitches are wider than ASIC glitches.

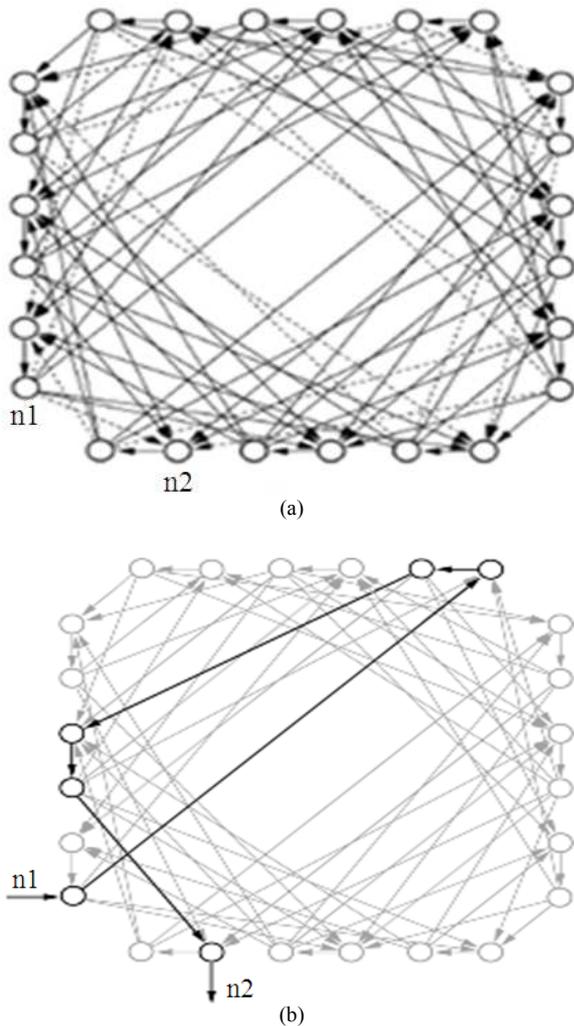


Fig. 8. (a) Routing graph for routing block (b) Shortest path between n1&n2

We avoided the glitches by adding programmable delay elements within the logic blocks of the FPGA such that we delay the early arriving signals to align the edges on each LUT input signals, thereby reducing some glitches on the output of each LUT.

The method demonstrated in **Fig. 6** by delaying the input signal of c, the output glitches eliminated since only the early arriving signals delayed, the overall critical path of the circuit is not increased.

4.1. Programmable Delay Element

The delay element circuit consists of two inverters shown in **Fig. 7**. The first inverter composed of pull-up

and pull-down resistor for controlling the delay of the circuit. The second inverter has large channel lengths to decrease short-circuit power. Both pull-up and pull-down resistor have n stages with a resistor and a bypass transistor which controlled by an SRAM bit. Control bits used to double the value of resistor in later stages:

$$\Delta \in \{k, \tau + k, 2\tau + k, 3\tau + k, \dots, (2n - 1)\tau + k\}$$

where, τ is the delay produced by a resistor R to charge or discharge the capacitor C and k is the delay produced by the delay produced by the bypass resistances and inverters.

4.2. Routing Algorithm for Reducing Delay

Initially without considering interconnect segment or logic block pins, nets routed one at a time using the shortest path. Several iterations carried out for finding shortest path. According to the lowest cost path nets are ripping up and rerouted at each iterations. [1]The cost of mapping resource is function of its current overuse and any overuse that occurred in preceding mapping iteration. If the resources overused then algorithm forces nets with alternative routes to avoid using that resource.

Figure 8a shown below is the placing route graph for routing block. Here each routing block input signal and output represented by node. When routing algorithm applied to the routing graph shown below obtained. **Figure 8b** shows the shortest path between the source n_1 and n_2 . Solid line represents the direct connection and dashed line represents the extended connection:

- C_{ij} is critical connection from the source of net i to one of its sinks j
- I_d is the intrinsic delay of routing node n
- P_c is the present congestion cost of node n

Mobbing/Deferral evasion Algorithm

- 1: $C_{ij} \leftarrow 1$ for each signal net i and each sink j
- 2: while shared routing nodes exist do
- 3: for all nets i do
- 4: rip up routing tree RT_i
- 5: initialize the queue PQ
- 6: for all sinks t_{ij} do
- 7: enqueue each node n in RT_i at costs $C_{ij}I_d$ to PQ
- 8: while t_{ij} is not found do
- 9: dequeue node m with the lowest cost from PQ

```

10: for allfanout node n of m do
11:   if node n is unseen then
12:     mark node n as seen
13:     enqueue n to PQ with the cost of  $C_{ij} I_d + (1-C_{ij}) I_d P_c$ 
14:   end if
15: end for
16: for all node n in the routed path  $t_{ij}$  to  $s_j$  do
17:   update the cost of node n
18:   add n to  $RT_i$ 
19: end for
20: end while
21: end for
22: mark all nodes in PQ as unseen
23: update  $C_{ij}$  for net i
24: end for
25: end while
    
```

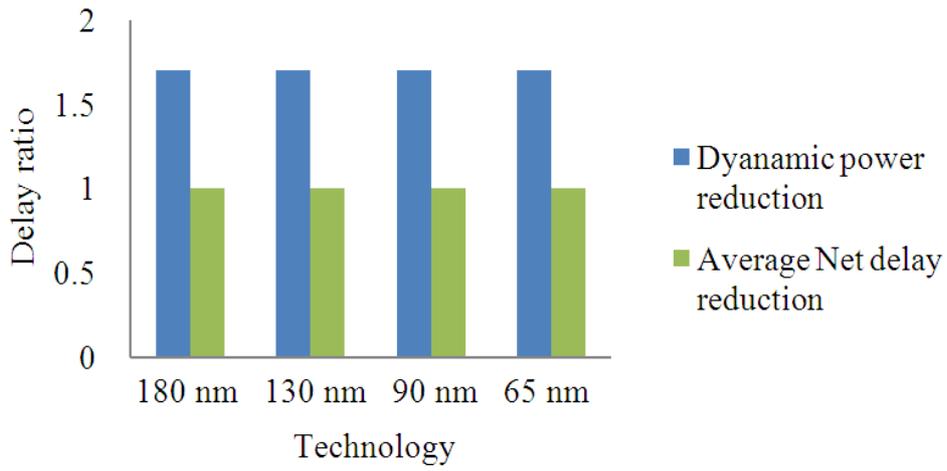


Fig. 9. Technology generation versus delay ratio

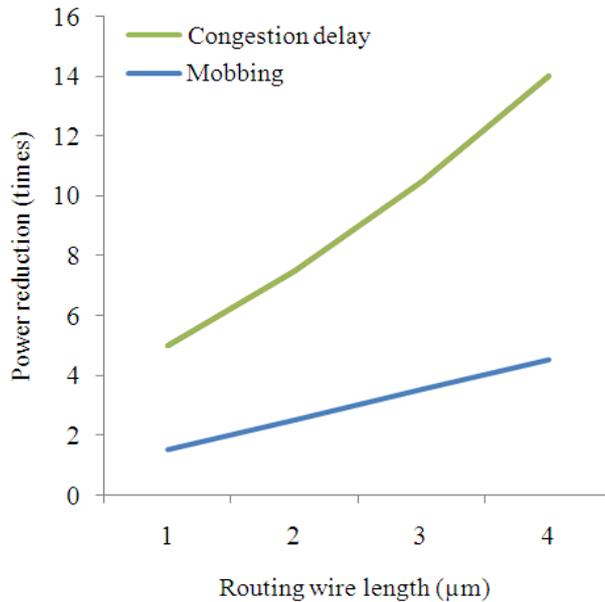


Fig. 10. Power reduction

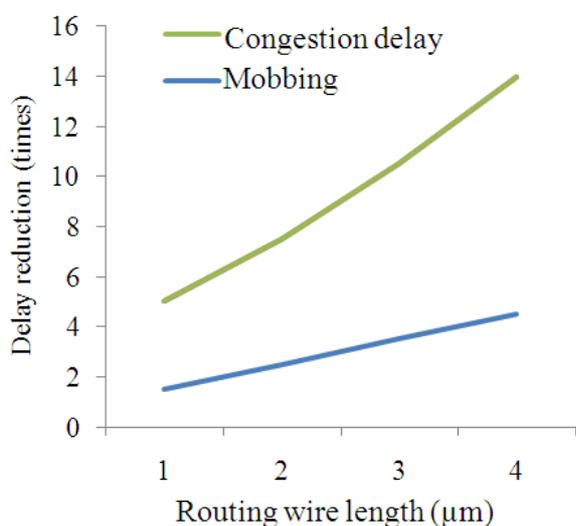


Fig. 11. Delay reduction

4.3. Comparison

Figure 9 show that the power consumption based on four technology nodes and that the delay ratio increases with increase in technology nodes. This is because of increase in the parasitic wires.

Figure 10 and 11 shows the comparison of existing and proposed algorithm which results in the positive way.

5. CONCLUSION

The power inefficiency of FPGA is a major problem. By reducing routed net length and programming overhead the power consumption reduced. Routed net length reduced by inter connect the segments by shortest routing channels. By decreasing the switch box and connection box flexibilities programming overhead reduced. We developed a new routing fabrics and algorithm FPGA can do 1.80 times reduction in the overall dynamic power consumption and 1.50 time reduction in average net delays.

6. REFERENCES

Betz, V. and J. Rose, 1997. VPR: A new packing, placement and routing tool for FPGA research. Proceedings of the 7th International Workshop Field-Programmable Logic Application, Sept. 1-3, Springer Berlin Heidelberg, London, pp: 213-222. DOI: 10.1007/3-540-63465-7_226

Lamoureux, J., G.G. Lemieux and S.J.E. Wilton, 2007. GlitchLess: An active glitch minimization technique for FPGAs. Proceedings of the ACM/SIGDA 15th International Symposium on Field Programmable Gate Arrays, Feb. 18-20, Monterey, ACM Press, New York, USA., pp: 156-165. DOI: 10.1145/1216919.1216946

Li, F., D. Chen, L. He and J. Cong, 2003. Architecture evaluation for power-efficient FPGAs. Proceedings of the ACM/SIGDA 11th International Symposium on Field Programmable Gate Arrays, Feb. 23-25, ACM Press, New York, USA., pp: 175-184. DOI: 10.1145/611817.611844

Li, F., Y. Lin, L. He and J. Cong, 2004. Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics. Proceedings of the ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays, Feb. 22-24, ACM Press, New York, USA., pp: 42-50. DOI: 10.1145/968280.968288

Lin, M., A. El Gamal, Y.C. Lu and S. Wong, 2006. Performance benefits of monolithically stacked 3D-FPGA. Proceedings of the ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays, Feb. 22-24, ACM Press, New York, USA., pp: 113-122. DOI: 10.1145/1117201.1117219

Lin, M. and A. El Gamal, 2007. A routing fabric for monolithically stacked 3D-FPGA. Proceedings of the ACM/SIGDA 15th International Symposium on Field Programmable Gate Arrays, Feb. 18-20, ACM Press, New York, USA., pp: 3-12. DOI: 10.1145/1216919.1216921

Lin, M. and A. El Gamal, 2008. TORCH: A design tool for routing channel segmentation in FPGAs. Proceedings of the 16th International ACM/SIGDA Symposium on Field Programmable Gate Arrays, Feb. 24-26, ACM Press, New York, USA., pp: 131-138. DOI: 10.1145/1344671.1344693

Lin, M. and A. El Gamal, 2009. A low-power field-programmable gate array routing fabric. IEEE Trans. Very Large Scale Integrat. Syst., 17: 1481-1494. DOI: 10.1109/TVLSI.2008.2005098

Tuan, T., S. Kao, A. Rahman, S. Das and S. Trimberger, 2006. A 90 nm low-power FPGA for battery-powered applications. Proceedings of the ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays, Feb. 22-24, ACM Press, New York, USA., pp: 3-11. DOI: 10.1145/1117201.1117203

Wang, L., M. French, A. Davoodi and D. Agarwal, 2006. FPGA dynamic power minimization through placement and routing constraints. EURASIP J. Embedded Syst., 2006: 7-7. DOI: 10.1155/ES/2006/31605