

NWR: Net Weighing Based Timing Driven Routing Algorithm

Geetanjali Udgirkar and G. Indumathi

Cambridge Institute of Technology, Krishnarajapuram – 560036, Bangalore, Karnataka, India; geetanjali.udgirkar@gmail.com, indumathi.ece@citech.edu.in

Abstract

Objectives: Timing driven VLSI routing is a challenging problem considering that the number of interconnects in today's technology design nodes is growing rapidly. VLSI or FPGA global routing is difficult problem considering the complexity and large size of present day IC designs. **Methods/Statistical analysis:** Net weighing algorithms for timing driven placement are effective way of optimizing delays during routing of designs. Timing driven routing with net weighing approach has not been explored in the past. **Findings:** We present two novel timing driven routing algorithm which is based on weighting of the critical nets. In the first method slack of the net is considered in evaluating the criticality of the net, whereas, in the second method we consider exponent of the criticality of the pin. These weights are then applied to all the nets while performing timing driven global routing. **Improvements:** The results of our experiments are encouraging, wherein, we obtain an improvement of 17.38% and 22.35% over VPR in the weighing schemes called Method A and Method B respectively.

Keywords: Global Routing, Net Weighting Method, VLSI Routing

1. Introduction

As per the Moore's law gate count is increasing twice every year. With more number of cells added to a design, the complexity of placement and routing of a design also increases. It is the placement and routing of the designs that determine the delays and clock speed after the design is fabricated. To optimize the delay or clock period of the design, methods have been proposed at placement level as well as routing level. In this paper, we investigate the application of net weighing during the routing of FPGA based designs.

Timing optimization algorithms have been traditionally applied to placement problems. There are two types of approaches in the placement and routing methods: 1. Path based, and 2. Net based. In path based approaches, the critical path of the design under consideration are analyzed and optimized. These approaches are complex to analyze and difficult to implement which is primarily due to exponential number of paths in a design. Net based methods are easy to implement compare to path

based approaches and hence are very popular. Weights of nets are computed based on the criticality of the net in the design, and then these weights are applied during the run of the routing tool.

Resources present in FPGA routing architecture use segments of various lengths. Due to this reason, the traditional (as in ASIC) methods of route length estimation do not apply. Timing driven routing becomes even more difficult under such scenario.

In this paper we investigate the net weighing schemes for FPGA based designs. Our approach in this paper is based on application of weights to the critical nets during the global routing flow. We use VPR tool for our experiments with a number of FPGA designs have proposed approximation algorithm for timing aware routing has been proposed in Kai Zhu et.al¹. In their implementation they consider segments of routing resources present in the routing architecture of an FPGA. They explore the complexity of routing tree and present approximation algorithms as the solution. Di Wu et al² have proposed a timing driven routing approach which optimizes cou-

*Author for correspondence

pling capacitance related delays. In the unified graph model, they consider the coupling capacitance as well as the detour routing wire lengths. The problem is modeled as a Sequential Ordering Problem (SOP).

Seokjin Lee et.al.³ use lagrangian relaxation based approach to timing driven routing for FPGA based designs. They use Lagrangian multipliers and sub-gradient projection method to guide the algorithm to form a routing tree. They also handle congestion constraints during global routing and produce results that outperform the state-of-the-art routing tools.

Shih Hsu et.al.⁴ proposes a method that optimizes crosstalk between routes along with the optimization of timing. After the initial timing driven routing run of the router, they form a delay degradation graph for adjacent wires on the die. Then, the delay degradation graph is iteratively improved by doing reassignment of the wires. They meet the goal of crosstalk constraints for all the nets in the design.

Yao-Wen Chang et.al.⁵ show that, in general, the timing driven routing problem is NP-complete. They also give approximation algorithms for this problem with guaranteed performance bound. They have considered a new model for timing driven routing in their implementation flow which is based on finding Minimum Spanning Tree (MST) with bounded delays. Xun Chen et.al.⁶ present two heuristics for the problem of dealing with high fan out nets during timing driven global routing.

Hanyu Liu et.al.⁷ has proposed a novel way of realizing of architecture of a design for FPGA which is time multiplexed. They analyze whether congestion problem can be solved by time-sharing a route, considering various nets of the routing channel in an FPGA. Kai Zhu et al.⁸ present an algorithm for determining nets with non-uniform upper bound. They have shown in their experiments that this algorithm can significantly improve routability and reduce delay bound violations.

Ramin Hojati et.al.⁹ present two methods to reduce resistivity minimization which are Topological Resistivity Minimization (TRM) and Constrained Resistivity Minimization (CRM). The idea behind their approach is to optimize the performance of the design by restricting the use of higher resistance layers. Some increase in the routing area is observed by them in their implementation.

Yan JT¹⁰ presents a timing driven routing algorithm which tries to reduce the congestion in the routing resources. They exploit the flexibility improvement of the edges within the routing tree, to reduce routing and

improve routability. They consider dynamic routing tree re-construction in their algorithm.

Zhaoyun Xing et.al.¹¹ device a parallel algorithm for timing drives routing. Their algorithm minimizes the circuit area as well as the timing delay of the critical path.

In Section 2, we discuss the implementation aspects of the proposed routing tool. Experiments are presented in Section 3. Finally, we present the results of our approach in Section 3 which is followed by conclusion in Section 4.

2. Implementation of the Router

We implement timing driven routing for FPGA routing architecture. A typical FPGA routing architecture is show in Figure 1.

Our work is based on the framework of VPR. We implement net weighing algorithms within the framework of VPR. Source code of VPR is modified to implement net weighing schemes A and B. In method A, as shown in Figure 2, we traverse through all the nets of the design, which is followed by traversal for all the pins for each net. In line 1 we compute the Pin_{crit} for each pin which is maximum of the ration of net slack divided by time period of the critical path of the design. Next, we assign Pin_{crit} to be equal to 10 raised to the power of Pin_{crit} . After this we choose the maximum of $Max_{criticality}$ and Pin_{crit} .

In method B, as shown in Figure 3, in line 1 we assign Pin_{crit} to be equal to Pin_{crit} raised to the power of 10. After this we choose the maximum of $Max_{criticality}$ and Pin_{crit} . Then we assign $pin_criticality$ to variable Pin_{crit} .

3. Experiments and Results

We carry out our experiments on twenty FPGA designs. The experiments are run on a 64-bit MAC running on UNIX operating system. We used VPR version 2.4 from internet.

The results obtained by using method A and method are shown in Table 1 and 2 respectively. As shown in Table 1, the third column represents the time period for each design, and column 4 and column 5 represent the time period obtained in timing analysis after running the router VPR and NWR respectively.

The comparison of time periods obtained for VPR and NWR (Method A and B) are shown in Figure 4 and 5 respectively. Note that while extracting the results for VPR we have used criticality exponent as 0.01 for VPR.

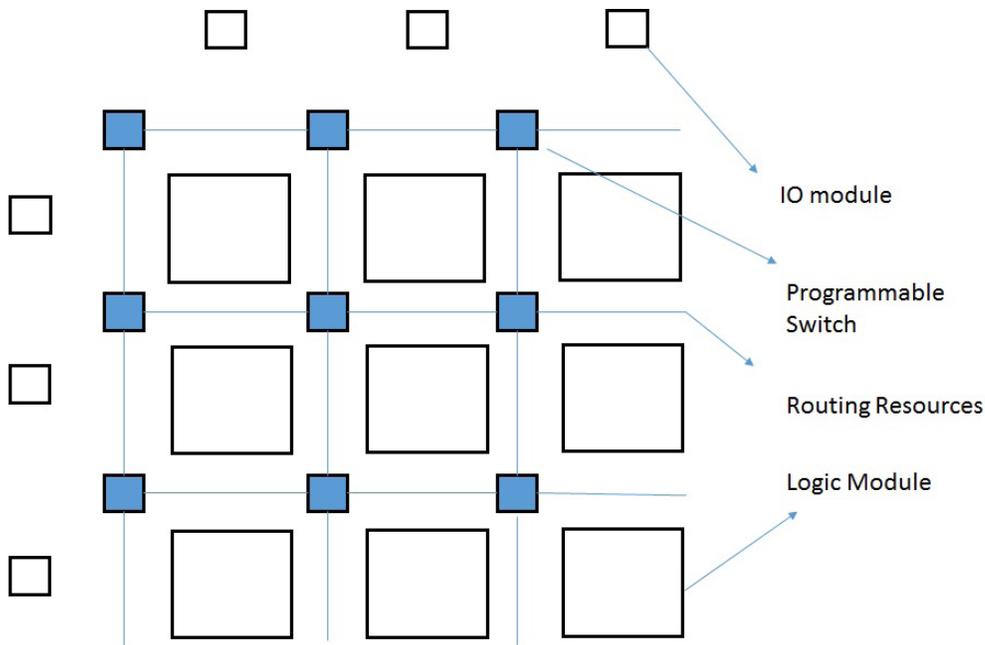


Figure 1: Typical FPGA resources.

Algorithm 1 Method A for weighing nets

```

for All Nets  $n$  do
  for All Pins  $p$  of net  $n$  do
     $pin\_crit = \max(\frac{-net\_slack[p]}{T\_crit}, 0)$ 
     $pin\_crit = \text{pow}(10, \frac{-net\_slack[p]}{T\_crit})$ 
     $pin\_crit = \min(pin\_criticality[p], max\_criticality)$ 
     $pin\_criticality[p] = pin\_crit;$ 
  end for
end for

```

Figure 2: Method A.

Algorithm 2 Method B for weighing nets

```

for All Nets  $n$  do
  for All Pins  $p$  of net  $n$  do
     $pin\_crit = \text{pow}(pin\_crit, 10);$ 
     $pin\_crit = \min(pin\_criticality[p], max\_criticality)$ 
     $pin\_criticality[p] = pin\_crit;$ 
  end for
end for

```

Figure 3: Method B.

Table 1: Comparison of time period between VPR and NWR for method A

Index	Designs	Clock Period in nano-second	Period VPR in nano-second	Period NWR in nano-second	% Improvement in Slack
1	alu4	82.00	72.93	72.93	0
2	apex2	130.00	116.60	101.36	113.7174416
3	apex4	150.00	120.13	104.27	53.11370028
4	bigkey	100.00	83.41	69.17	85.8162355
5	clma	300.00	262.30	245.24	45.25173749
6	des	120.00	94.35	97.52	-12.34861069
7	diffeq	100.00	81.79	68.03	75.57464478
8	dsip	65.00	62.07	61.96	3.773069036
9	elliptic	150.00	143.50	140.46	46.75384615
10	ex1010	189.00	188.27	188.27	0
11	ex5p	150.00	91.33	99.42	-13.79425535
12	frisc	150.00	136.83	139.97	-23.8575983
13	misex3	120.00	99.49	91.40	39.43625575
14	pdc	380.00	193.69	266.74	-39.20735111
15	s298	250.00	152.94	183.20	-31.17581602
16	s38417	125.00	107.10	112.20	-28.47004917
17	s38584.1	91.00	90.59	90.59	0
18	seq	140.00	97.48	83.07	33.90715922
19	spla	120.00	181.94	186.85	7.932381813
20	tseng	80.00	55.94	59.00	-12.75052047
Average					17.18

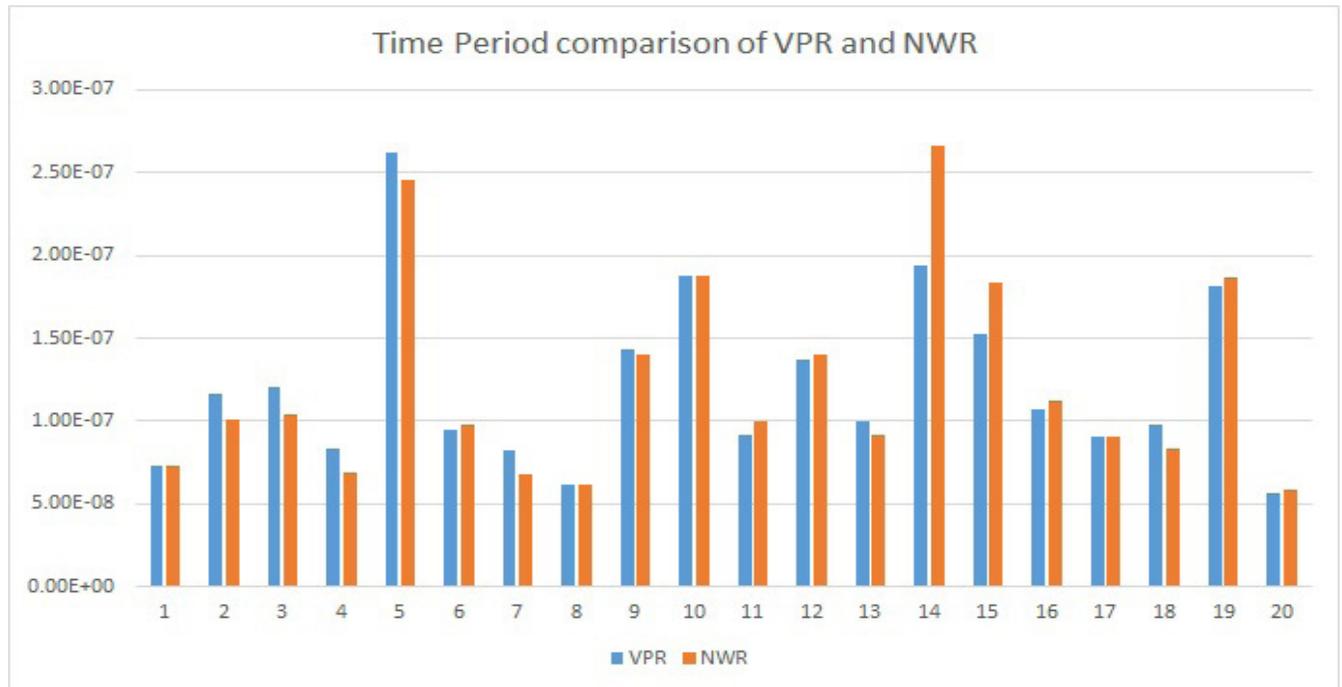
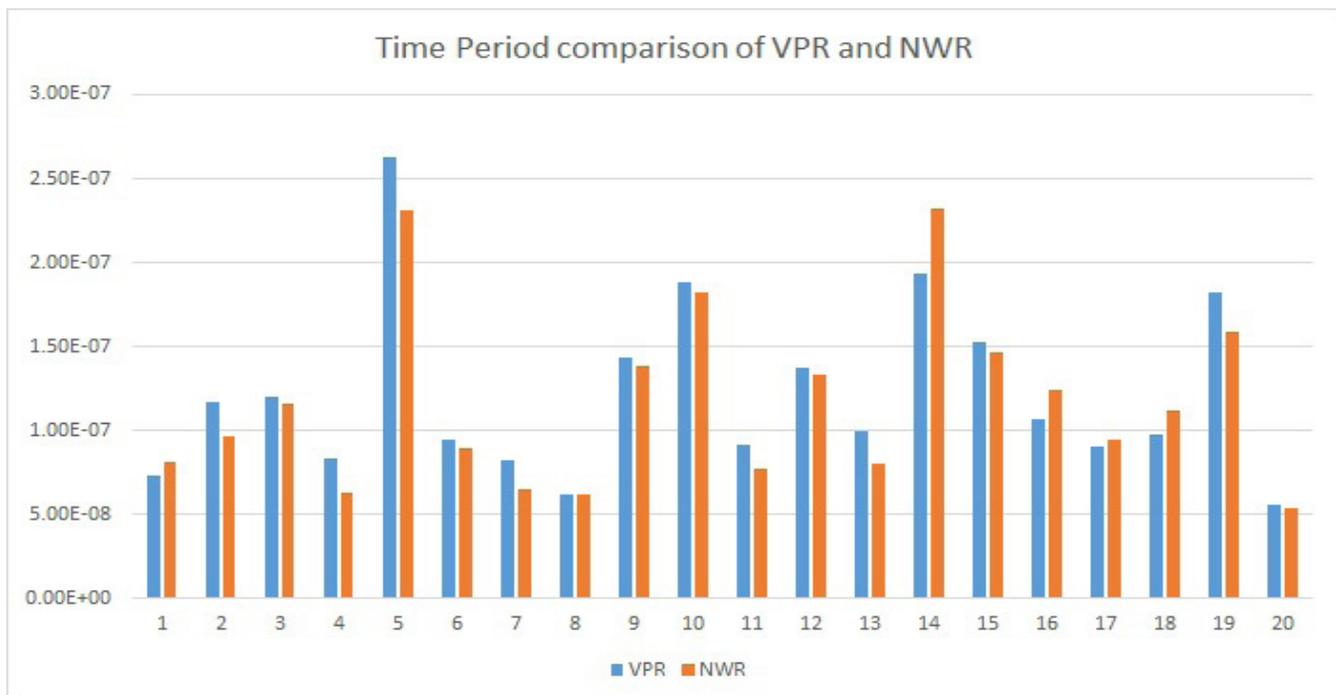


Figure 4: Comparison of time period between VPR and NWR for method A

Table 2.: Comparison of time period between VPR and NWR for method B

Index	Designs	Clock Period in nano-second	Period VPR in nano-second	Period NWR in nano-second	% Improvement in Slack
1	alu4	82.00	72.93	81.17	-90.87190005
2	apex2	130.00	116.60	96.96	146.5810881
3	apex4	150.00	120.13	115.96	13.96477836
4	bigkey	100.00	83.41	62.56	125.6624153
5	clma	300.00	262.30	231.10	82.77892726
6	des	120.00	94.35	89.68	18.21740792
7	diffeq	100.00	81.79	64.47	95.15952743
8	dsip	65.00	62.07	62.06	0.580997949
9	elliptic	150.00	143.50	138.25	80.72307692
10	ex1010	189.00	188.27	181.87	880.0824176
11	ex5p	150.00	91.33	77.43	23.68710342
12	frisc	150.00	136.83	133.22	27.38727797
13	misex3	120.00	99.49	80.33	93.42980989
14	pdc	380.00	193.69	232.23	-20.68680493
15	s298	250.00	152.94	146.11	7.039235081
16	s38417	125.00	107.10	123.94	-94.06012517
17	s38584.1	91.00	90.59	94.18	-879.1187271
18	seq	140.00	97.48	112.07	-34.31383156
19	spla	120.00	181.94	158.94	-37.13349264
20	tseng	80.00	55.94	54.05	7.832153884
Average					22.35

**Figure 5:** Comparison of time period between VPR and NWR for method B.

4. Conclusion

Timing driven routing is a well studied problem and important problem considering the impact of routing to performance of the design after fabrication. In this paper we present a timing driven routing algorithm. We present two methods for weighing the nets of critical path of a design. Our experiments show that an improvement of 17.38 % using Method A and 22.35% using Method B can be obtained. Our approach show promising results over well known FPGA routing tool VPR.

5. Acknowledgements

We would like to thank Department of Electronics and Communication Engineering, Cambridge Institute of Technology, Bangalore for providing the resources to conduct the experiments.

6. References

1. Kai Zhu, Yao-Wen Chang, Wong DF. Timing-driven routing for symmetrical-array-based FPGAS. In: Computer Design: VLSI in Computers and Processors, 1998. ICCD '98. Proceedings. International Conference; Oct 1998. p. 628–33.
2. Di Wu, Jiang Hu, Min Zhao, Rabi Mahapatra. Timing driven track routing considering coupling capacitance. In: Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference; Jan 2005. p. 1156–59. Crossref.
3. Seokjin Lee, Wong MDF. Timing-driven routing for FPGAS based on lagrangian relaxation, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Apr 2003; 22(4):506–10. Crossref.
4. Shih Hsu Huang, Shih Hsu. A timing driven approach for crosstalk minimization in gridded channel routing. In: Circuits and Systems, 2002. APCCAS '02. 2002 Asia-Pacific Conference; 2002. p. 263–66.
5. Yao-Wen Chang, Wong DF, Zhua K, Wong CK. On a new timing-driven routing tree problem [FPGA]. In: Circuits and Systems, 1996. ISCAS '96, Connecting the World, 1996. IEEE International Symposium, May 1996; 4:420–23.
6. Xun Chen, Zhu J, Zhang M. Timing-driven routing of high fanout nets. In: 21st International Conference on Field Programmable Logic and Applications; Sept 2011. p. 423–28. Crossref.
7. Hanyu Liu, Chen X, Ha Y. An area-efficient timing-driven routing algorithm for scalable fpgas with time-multiplexed interconnects. In: Field-Programmable Custom Computing Machines, 2008. FCCM '08. 16th International Symposium; April 2008. p. 275–76. Crossref.
8. Kai Zhu, Wong DF. Switch bound allocation for maximizing routability in timing-driven routing of FPGAS, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Apr 1998; 17(4):316–23. Crossref.
9. Ramin Hojati. Timing driven routing and resistivity minimization. In: Custom Integrated Circuits Conference, 1991, Proceedings of the IEEE, May 1991, p. 28. Crossref.
10. Yan JT. Dynamic tree reconstruction with application to timing constrained congestion-driven global routing, IEE Proceedings – Computers and Digital Techniques. March 2006; 153(2):117–29. Crossref.
11. Zhaoyun Xing, Banerjee P. A parallel algorithm for timing-driven global routing for standard cells. In: Parallel Processing, 1998. International Conference; Aug 1998. p. 54–61.