Research on Net Weighting Schemes in Performance Driven Global Routing

Geetanjali Udgirkar, G. Indumathi

Abstract— In today's VLSI technology nodes, interconnect delay plays an important part in deciding the performance of the chip designs. Various methods are introduced at the level of placement and routing to address this problem. To address this problem at the level of global routing, net weighting methods are being explored in the industry and academia. We investigate four methods for weighting the critical nets during performance driven global routing. This paper presents a comparative study conducted on the four methods for net weighting proposed by us in our previous works.

Keywords: FPGA Routing, Net Weighting Methods.

1. INTRODUCTION

According to Moore's law the gate count increases twice every year. As more and more cells are added to the design, placement and routing complexity of the design increases many folds. This is primarily due to the fact that the interconnect delay becomes significant. The efficient placement and routing of the design decides the performance of the final chip. Many methods have been proposed in the literature at the level of placement of the design and also at the global routing level to address this issue. In this paper, we focus on methods to perform efficient timing driven global routing for 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.

Peishan Tu et al.[1] proposed a novel timing driven routing tree construction method for tradeoff between wirelength and timing. Their approach is based on shortest path trees and minimum spanning trees.

Dongsheng Wang et al.[2], proposed a method in which graph connectivity of graph is obtained at first which is based on observing Elmore delay model. This graph is known as DCG (directed connectivity graph) which describes connectivity of all the edges of the graph. They also propose a C-Tree algorithm which is a timing-driver Steiner tree routing approach.

Dongsheng Wang et al.[3] propose a new algorithm called MCM/IC multilayer routing algorithm, named MLR, which is timing driven and considers the Elmore delay as a performance issue. It also consider number of layers, total wirelength and the vias. Based on layer assignment problem, this algorithm assigns all the nets into the routing layers in a layer-pair by later-pair fashion.

Jin-Tai Yan et al.[4], proposes a simulated-annealing based approach, wherein, the input is an initial routing tree. They consider routing flexibility in a SRT and flexibility of the Steiner-point in one Y-type wire. The flexibility of the points is timing constrained. The simulated-annealing-based approach obtains a better timing-constrained flexibility-driven SRT by performing reassignment of the possible locations of the Steiner points in a SRT.

Tsung-Yi Ho et al.[5], propose a new framework for rapid multilevel routing considering performance optimization and crosstalk. To optimize performance-driven routing, they have proposed a new method of minimum-radius minimum-cost spanning tree algorithm which is applied in global routing. To reduce the crosstalk problem, they have added an intermediate stage of layer/track into their routing framework.

Fang-Jou Liu et al.[6] propose a novel timing model which is based on the layout of the design. This is based on Asymptotic Waveform Evaluation (AWE) which helps in analysis of timing paths during global routing. This computation of moments of the interconnect tree are enabled

Revised Manuscript Received on July 10, 2019.

Geetanjali Udgirkar, CMR Institute of Technology, Bangalore , Karnataka India.**(e**-mail: geetanjali.udgirkar@gmail.com)

G. Indumathi Cambridge Institute of Technology, Bangalore, Karnataka India.**(e**-mail: indumathi.ece@citech.edu.in)

by this model in a bottom-up fashion, and can be integrated global router without much effort. This integration helps in layout optimization in an incremental manner, i.e., routing and timing analysis are performed simultaneously, along with exchange of information between them.

Jiang Hu et al.[7], propose a method for global routing that can optimize routing delay, routing congestion, and number of bends. This approach solves problems which are competing objectives. Based on the timing constraints, routing flexibilities are obtained and used to reduce congestion given the timing constraints.

Kaushik Roy et al. [8], proposes a solution for the problem of place and route in field programmable gate arrays (FPGA's) targeted for low power dissipation constrained by critical path delays. There is a capacitive loading for each net of the large number of unprogrammed anti-fuses present in the routing architecture. Due to signal transitions happening at the output of logic gates, a substantial amount of power gets dissipated in the routing architecture. The signal transitions within internal nodes of the circuit are estimated based on primary input signal distributions.

Jucemar Monteiro et al. [9], proposed an algorithm in which both routing metrics and timing are considered during detailed placement. They also present a detailed analysis of a score for the timing quality and the number of routing overflows. They also provide a trade-off between them and experiment their algorithm on the International Conference on Computer Aided Design (ICCAD) 2015 timing-driven contest benchmarks.

Hsiao-Ping Tseng et al.[10], propose a crosstalk and timing driven router for the task of VLSI design which is used for detailed and global routing. This method targets to process the timing constraints and crosstalk by tuning wire spacing and by performing ordering of nets and by quantitative analysis. This novel heuristic fits for detailed routing and global routing along with the flow of layout design.

Takahiro DEGUCHI et al.[11], present a method for performance driven routing. This method is based on a multi-layer routing model. The use linear programming in this method by considering buffer insertion and wire sizing under timing constraints. The routes for each net are determined in a hierarchical level.

Sung-Woo Hur et al.[12], this paper performs the study of the performance driven objective of maze routing. They adopt a graph model appropriate for applications to global and detailed routing. This model captures blockages naturally, limits the routing layer assignment and wire-sizing resources. The perform annotation of each edge in the graph with capacitance and resistance values pertaining to the particular wire segment. The objective is to find low resistance-capacitance delay paths or obtains a tradeoff between total capacitance and resistance-capacitance delay.

Jin-Tai Yan et al.[13] present a method for performance driven routing which is based upon the concept of assignment of hidden Steiner points and insertion of sharing buffer. This is achievable for input graph where a set of connecting nodes of a net are given. They construct a rectilinear Steiner tree which is performance driven by inserting Steiner points and sharing buffers into all possible positions that given a better performance.

Christophe Alexandre et al.[14], propose a routing tool called TSUNAMI, which is more of a platform based on an C++ database. All the tools interact with this database consistently and collaborate on the problems at hand. A timing driven global routing and a timing driven placement tool has been developed on this platform.

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.

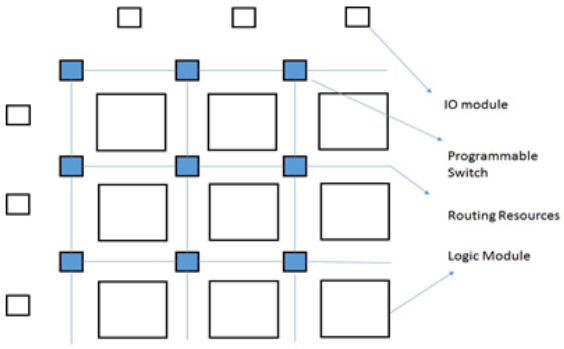

Figure 1: Typical FPGA Resources

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}.

Algorithm 1 Method A for weighting nets

```
1: for All nets n do
2:     for All pins p of net n do
3:         pin_crit = max((max_criticality - net_slack[p]) / T_crit, 0)
4:         pin_crit = pow(e, pin_crit)
5:         pin_criticality[p] = pin_crit
6:     end for
7: end for
```

Figure 2: Method A

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}.

Algorithm 2 Method B for weighting nets

```
for All nets n do
    for All pins p of net n do
        pin_crit = max((max_criticality - net_slack[p]) / T_crit, 0)
        pin_crit = pow(pin_crit, e)
        pin_criticality[p] = pin_crit
    end for
end for
```

Figure 3: Method B

In method C, as shown in Figure 3, we traverse through all the nets of the design, which is followed by traversal for all the pins for each net. Initially, 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 e raised to the power of Pin_{crit}. After this we choose the maximum of $Max_{criticality}$ and Pin_{crit}.

Algorithm 3 Method C for weighting nets

```
for All Nets n do
    for All Pins p of net n do
        pin_crit = max(-net_slack[p]/T_crit, 0)
        pin_crit = pow(e, -net_slack[p]/T_crit)
        pin_crit = min(pin_criticality[p], max_criticality)
        pin_criticality[p]=pin_crit;
    end for
end for
```

Figure 4: Method C

In method D, as shown in Figure 4, we traverse through all the nets of the design, which is followed by traversal for all the pins for each net. Initially, 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 Pin_{crit} raised to the power of e. After this we choose the maximum of $Max_{criticality}$ and Pin_{crit}.

Algorithm 4 Method D for weighting nets

```
for All Nets n do
    for All Pins p of net n do
        pin_crit = max(-net_slack[p]/T_crit, 0)
        pin_crit = pow(-net_slack[p]/T_crit, e)
        pin_crit = min(pin_criticality[p], max_criticality)
        pin_criticality[p]=pin_crit;
    end for
end for
```

Figure 5: Method D

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

Table 1 Comparison of time period between VPR and proposed for Method A

Index	Designs	Clock Period in nanosecond	Period VPR in nanosecond	Period of MethodA in nanosecond	% Improvement in Slack
1	alu4	82.00	146.491	110.43	43.97683
2	apex2	130.00	149.168	126.73	17.26
3	apex4	150.00	144.419	121.98	14.95933
4	bigkey	100.00	144.95	141.22	3.73
5	clma	300.00	292.266	278.407	4.619667
6	des	120.00	151.944	118.95	27.495
7	diffeq	100.00	96.4889	97.154	-0.6651
8	dsip	65.00	98.5863	83.45	23.28662
9	elliptic	150.00	215.927	250.144	-22.8113
10	ex1010	189.00	324.849	336.741	-6.29206
11	ex5p	150.00	136.444	98.34	25.40267
12	frisc	150.00	234.588	196.601	25.32467
13	misex3	120.00	131.223	133.561	-1.94833
14	pdc	380.00	315.524	247.065	18.01553
15	s298	250.00	237.959	255.431	-6.9888
16	s38417	125.00	260.179	244.969	12.168
17	s38584.1	91.00	150.017	101.1	53.75495
18	seq	140.00	153.597	149.459	2.955714
19	spla	120.00	212.304	135.31	64.16167
20	tseng	80.00	84.2235	88.5287	-5.3815
Average					14.65118

Table 2 Comparison of time period between VPR for Method B

Index	Designs	Clock Period in nanosecond	Period VPR in nanosecond	Period of MethodB in nanosecond	% Improvement in Slack
1	alu4	82.00	146.491	110.43	43.97683
2	apex2	130.00	149.168	126.73	17.26
3	apex4	150.00	144.419	121.98	14.95933
4	bigkey	100.00	144.95	141.22	3.73
5	clma	300.00	292.266	253.611	12.885
6	des	120.00	151.944	118.95	27.495
7	diffeq	100.00	96.4889	88.2141	8.2748
8	dsip	65.00	98.5863	83.45	23.28662
9	elliptic	150.00	215.927	182.579	22.232
10	ex1010	189.00	324.849	247.535	40.90688
11	ex5p	150.00	136.444	98.34	25.40267
12	frisc	150.00	234.588	168.839	43.83267
13	misex3	120.00	131.223	106.322	20.75083
14	pdc	380.00	315.524	238.912	20.16105
15	s298	250.00	237.959	198.24	15.8876
16	s38417	125.00	260.179	177.33	66.2792
17	s38584.1	91.00	150.017	101.1	53.75495
18	seq	140.00	153.597	126.938	19.04214
19	spla	120.00	212.304	135.31	64.16167
20	tseng	80.00	84.2235	90.0419	-7.273
Average					26.85031

Table 3 Comparison of time period between VPR for Method C

Index	Designs	Clock Period in nanosecond	Period VPR in nanosecond	Period NWR in nanosecond	% 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

Table 4 Comparison of time period between VPR for Method D

Index	Designs	Clock Period in nanosecond	Period VPR in nanosecond	Period NWR in nanosecond	% 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	7.099235081
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

4. RESULTS & DISCUSSIONS

As shown in Table 1, Table 2, Table 3 and Table 4, the routing Method C works the best on Design Apex4. We list down the pin criticality for design Apex4 considering for Methods A, B, C and D in Table 5. As show in Figure 6, the value of weights for Method C are the highest, which is the reason for best improvement in slack for the design Apex4 using method C.

Table 5 Analysis of Critical paths for design Apex4

Critical Paths in decreasing order of Criticality	Pin crititicality	Method A	Method B	Method C	Method D
1	0.997	2.7	0.992	9.93	0.97
2	0.995	2.7	0.986	9.88	0.95
3	0.994	2.7	0.984	9.86	0.94
4	0.987	2.6	0.965	9.70	0.87
5	0.974	2.6	0.931	9.41	0.76
6	0.956	2.6	0.885	9.03	0.63
7	0.949	2.5	0.867	8.89	0.59
8	0.946	2.5	0.86	8.83	0.57
9	0.94	2.5	0.845	8.71	0.53
10	0.927	2.5	0.814	8.45	0.46
11	0.877	2.4	0.7	7.53	0.26
12	0.864	2.3	0.672	7.31	0.23
13	0.852	2.3	0.647	7.11	0.20
14	0.85	2.3	0.643	7.07	0.19
15	0.844	2.3	0.631	6.98	0.18
16	0.841	2.3	0.625	6.93	0.17
17	0.836	2.3	0.615	6.85	0.16
18	0.833	2.3	0.609	6.80	0.16
19	0.815	2.2	0.573	6.53	0.12
20	0.814	2.2	0.572	6.51	0.12
21	0.812	2.2	0.568	6.48	0.12
22	0.808	2.2	0.56	6.42	0.11
23	0.794	2.2	0.534	6.22	0.1
24	0.791	2.2	0.529	6.18	0.09
25	0.776	2.1	0.502	5.97	0.07
26	0.775	2.1	0.5	5.95	0.07
27	0.759	2.1	0.473	5.74	0.06
28	0.739	2.0	0.439	5.48	0.04
29	0.737	2.0	0.436	5.45	0.04
30	0.735	2.0	0.433	5.43	0.04
31	0.731	2.0	0.427	5.38	0.04
32	0.729	2.0	0.424	5.35	0.04
33	0.715	2.0	0.402	5.18	0.03
34	0.698	2.0	0.376	4.98	0.02
35	0.678	1.9	0.348	4.76	0.02
36	0.654	1.9	0.315	4.50	0.01
37	0.631	1.8	0.286	4.27	0.01
38	0.614	1.8	0.266	4.11	0.00
39	0.61	1.8	0.261	4.07	0.00
40	0.61	1.8	0.261	4.07	0.00
41	0.59	1.8	0.238	3.89	0.00
42	0.586	1.7	0.234	3.85	0.00
43	0.586	1.7	0.234	3.85	0.00
44	0.575	1.7	0.222	3.75	0.00
45	0.568	1.7	0.215	3.69	0.00
46	0.561	1.7	0.208	3.63	0.00
47	0.54	1.7	0.187	3.46	0.00
48	0.538	1.7	0.185	3.45	0.00

49	0.536	1.70	0.184	3.43	0.00
50	0.534	1.7	0.182	3.42	0.00
51	0.523	1.6	0.172	3.33	0.00
52	0.514	1.6	0.164	3.26	0.00
53	0.502	1.6	0.154	3.17	0.00
54	0.502	1.6	0.154	3.17	0.00
55	0.486	1.6	0.141	3.06	0.00
56	0.472	1.6	0.13	2.96	0.00
57	0.469	1.5	0.128	2.94	0.00
58	0.468	1.5	0.127	2.93	0.00
59	0.462	1.5	0.123	2.89	0.0
60	0.457	1.5	0.119	2.86	0.0
61	0.414	1.5	0.091	2.59	0.0
62	0.386	1.4	0.075	2.43	0.0
63	0.368	1.4	0.066	2.33	0.0
64	0.345	1.4	0.055	2.21	0.0
65	0.342	1.4	0.054	2.19	0.0
66	0.334	1.3	0.051	2.15	0.0
67	0.303	1.3	0.039	2.00	0.0
68	0.296	1.3	0.037	1.97	0.0
69	0.262	1.3	0.026	1.82	0.0
70	0.255	1.2	0.024	1.79	0.0
71	0.244	1.2	0.022	1.75	0.0
72	0.235	1.2	0.02	1.71	0.0
73	0.223	1.2	0.017	1.67	0.0
74	0.21	1.2	0.014	1.62	0.0
75	0.196	1.2	0.012	1.57	0.0
76	0.183	1.2	0.01	1.52	0.0
77	0.182	1.2	0.01	1.52	0.0
78	0.169	1.1	0.008	1.47	0.0
79	0.169	1.1	0.008	1.47	0.0
80	0.152	1.1	0.006	1.41	0.0
81	0.133	1.1	0.004	1.35	0.0
82	0.132	1.1	0.004	1.35	0.0
83	0.126	1.1	0.004	1.33	0.0
84	0.12	1.1	0.003	1.31	0.0
85	0.119	1.1	0.003	1.31	0.0
86	0.109	1.1	0.002	1.28	0.0
87	0.105	1.1	0.002	1.27	0.0
88	0.096	1.1	0.002	1.24	0.0
89	0.093	1.0	0.002	1.23	0.0
90	0.054	1.0	0.0	1.13	0.0
91	0.054	1.0	0.0	1.13	0.0
92	0.044	1.0	0.0	1.10	0.0
93	0.039	1.0	0.0	1.09	0.0
94	0.037	1.0	0.0	1.08	0.0
95	0.035	1.0	0.0	1.08	0.0
96	0.027	1.0	0.0	1.06	0.0
97	0.026	1.0	0.0	1.06	0.0
98	0.01	1.0	0.0	1.02	0.0
99	0.997	2.7	0.992	9.93	0.97
100	0.995	2.7	0.986	9.88	0.95

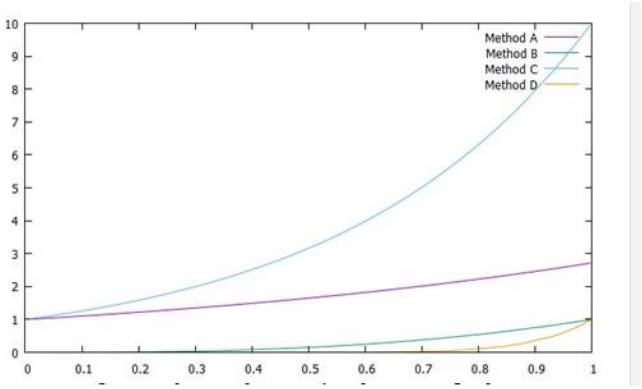

Figure 6: Analysis of Critical paths for design Apex4

5. 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 comparative analysis of timing driven routing algorithms. We present four methods for weighing the nets of critical path of a design and provide a comparative analysis of these methods. The methods A, B, C and D proposed in this work, show an improvement of 14.65 %, 26.85 %, 17.18 % and 22.35 % improvement in slack when compared with a well-known routing algorithm VPR.

6. 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.

7. REFERENCES

1. P. Tu, W. Chow and E. F. Y. Young, "Timing driven routing tree construction," 2017 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP), Austin, TX, 2017, pp. 1-8.
2. Dongsheng Wang and E. S. Kuh, "A new general connectivity model and its applications to timing-driven Steiner tree routing," 1998 IEEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology (Cat. No.98EX196), Lisboa, Portugal, 1998, pp. 71-74 vol.2.
3. Dongsheng Wang and E. S. Kuh, "A new timing-driven multilayer MCM/IC routing algorithm," Proceedings 1997 IEEE Multi-Chip Module Conference, Santa Cruz, CA, USA, 1997, pp. 89-94.
4. Jin-Tai Yan, Chia-Fang Lee and Yen-Hsiang Chen, "A simulated-annealing-based approach for timing-constrained flexibility-driven routing tree construction," The 2004 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS '04., Hiroshima, Japan, 2004, pp. I-461.
5. Tsung-Yi Ho, Yao-Wen Chang, Sao-Jie Chen and Der-Tsai Lee, "Crosstalk- and performance-driven multilevel full-chip routing," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and

Systems, vol. 24, no. 6, pp. 869-878, June 2005.
6. Fang-Jou Liu, J. Lillis and Chung-Kuan Cheng, "Design and implementation of a global router based on a new layout-driven timing model with three poles," Proceedings of 1997 IEEE International Symposium on Circuits and Systems. Circuits and Systems in the Information Age ISCAS '97,
7. Jiang Hu and S. S. Sapatnekar, "Performance driven global routing through gradual refinement," Proceedings 2001 IEEE International Conference on Computer Design: VLSI in Computers and Processors. ICCD 2001, Austin, TX, USA, 2001, pp. 481-483.
8. K. Roy, "Power-dissipation driven FPGA place and route under timing constraints," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 46, no. 5, pp. 634-637, May 1999.
9. J. Monteiro et al., "Routing-Aware Incremental Timing-Driven Placement," 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, 2016, pp. 290-295.
10. Hsiao-Ping Tseng, L. Scheffer and C. Sechen, "Timing- and crosstalk-driven area routing," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, no. 4, pp. 528-544, April 2001.
11. T. Deguchi, T. Koide and S. Wakabayashi, "Timing-driven hierarchical global routing with wire-sizing and buffer-insertion for VLSI with multi-routing-layer," Proceedings 2000. Design Automation Conference. (IEEE Cat. No.00CH37106), Yokohama, 2000, pp. 99-104.
12. Sung-Woo Hur, A. Jagannathan and J. Lillis, "Timing-driven maze routing," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no. 2, pp. 234-241, Feb. 2000.
13. Jin-Tai Yan, Shi-Qin Huang and Zhi-Wei Chen, "Top-down-based timing-driven steiner tree construction with wire sizing and buffer insertion," TENCON 2007 - 2007 IEEE Region 10 Conference, Taipei, 2007, pp. 1-4.
14. C. Alexandre, H. Clement, J. -. Chaput, M. Sroka, C. Masson and R. Escassut, "TSUNAMI: an integrated timing-driven place and route research platform," Design, Automation and Test in Europe, Munich, Germany, 2005, pp. 920-921 Vol. 2.