Research Journal of Recent Sciences ISSN 2277-2502
Vol. **3(7),** 39-43, July **(2014)** *Res.J.Recent Sci.*

Clock Mesh Synthesis with Blockage Placement for Stub and Mesh Wire Minimisation

Koundinya C.[1] and J Selva Kumar[2]
[1]VLSI Design, SRM University, Chennai, Tamil Nadu, INDIA
[2]Electronics and Communication Dept., SRM University, Chennai, Tamil Nadu, INDIA

Available online at: www.isca.in, www.isca.me
Received 11th April 2014, revised 26th May 2014, accepted 28th May 2014

Abstract

A method for optimizing clock mesh is proposed which diminishes the power in network significantly while considering timing slack. The proposed paper implements the methodology by introducing placement blockages during the IC design flow. Stub wire minimization is achieved by using register placement. Placement of blockages in order to further reduce the power dissipation along with optimized power density and guaranteed non negative timing slack is the main essence of the paper. The advantages of the proposed method when implemented over ISCAS'89 benchmark circuits are reduced power dissipation—26% on an average when analogized to previous works and significant reduction in wire length by 50% when compared to earlier uniform mesh works. Further, reduction of timing slack (1.2% of the clock period) provides added benefit.

Keywords: Clock mesh, CTS, tree hierarchy, Iterative placement.

Introduction

In real time designs where high performance and low skew variations are required, recurrence in clock structures is preferred. Spines within clock tree or cross links among these clock structures are implemented to minimize clock skew in local areas while methods like clock mesh sizing of clock distribution networks for high performance[1], clock distribution network for microprocessors [2] and the design and analysis of the clock distribution network for a 1.2 GHz Alpha microprocessor[3] assures low skew variation in global level. The shorting of all the sink registers to the horizontal and vertical metal lines induces redundancy which minimizes the skew at the cost of high power consumption compared to other clock structures.

Design automation methods have been evolved in clock mesh synthesis and optimization due to the popularity of clock mesh structures in high performance designs. In method proposed in combinatorial algorithms for fast clock mesh optimization[4], insertion and sizing off buffer drivers as well as reduction of mesh were proposed for power reduction. The method proposed in, meshworks: An efficient framework for planning, synthesis and optimization of clock mesh networks[5] involves reduction of mesh wires under skew constraints. In Timing slack aware incremental register placement with non-uniform grid generation for clock mesh synthesis[6] and in non-uniform clock mesh optimization with linear programming buffer insertion[7] iterative movement of the mesh wires is introduced in order to reduce the stub wires from sink to the mesh. In the paper, an algorithm for routing with capacitance/distance constraints for clock distribution in microprocessors[8], connections similar to steiner tree between register and mesh are developed to reduce the stub wire length.

The novel approach proposed in this paper is to synthesize clock mesh combines traditional physical design flow of placement and clock network synthesis through iterative placement. In this paper, the placement of blockages is done while taking the timing slack into account such that the mesh wire along with the stub wire is reduced significantly thereby improving the power consumption of the overall design.

The rest of the paper is categorized as follows. In section II, the history of clock mesh is discussed. Later in section III, the proposed method is presented. The results are summarized in section IV. The paper is concluded in section V.

Preliminaries: Clock meshes are preferred to minimize the crosstalk in physical IC design. Optimization is done by using the full redundancy of the mesh tracks.

The two major factors when defining a clock mesh are total wire length which in turn has an impact over the total power and the clock skew which is caused due to both internal and environmental variations. In section II-A the wire length of the mesh is discussed and in section II-B skew variations and timing slack are presented.

Wirelength Determination in Clock Mesh: Clock mesh network is illustrated in the figure 1. The mesh wire length is contributed by the stub wire length and grid wire length. These play an important role in the dynamic power dissipation of the design. The total wire length depends directly on the switching capacitance in the clock network. Thus various methods have been developed in order to reduce the power consumption in clock networks. The works done in combinatorial algorithms for fast clock mesh optimization[4], meshworks: An efficient framework for planning, synthesis and optimization of clock mesh networks[5] minimizes mesh wire and the methods implemented in, An algorithm for routing with capacitance/distance constraints for clock distribution in microprocessors[8], Non-uniform clock mesh optimization with linear programming buffer insertion[7] diminish stub wire. The method proposed in this work implements placement blockages along with the method proposed by J.Lu et.al, which reduces both stub wire and mesh wire length.

Analysis of Clock Skew in Mesh Networks: Skew of the mesh is estimated as follows[9]:

$$t_{skew} = t_{skew}^{buf} + D_{mesh}(d_{\max}) + D_{stub}(L_{stub}^{\max}) \tag{1}$$

Where t_{skew}^{buf} , $D_{mesh}(d_{\max})$ and $D_{stub}(L_{stub}^{\max})$ are the skew introduced by the buffer drivers of mesh, the maximum delay from a buffer driver on the mesh to the stub wire tapping point, and the utmost delay from a tapping point of a stub wire to the sink register, respectively.

According to (1), the contradicting feature is if low skew is required then a denser mesh is opted which in turn increases the power consumption. Thus a tradeoff is required between power and skew. This can be maintained by using the method proposed by J.Lu, et.al[10].

Timing Constraints: As shown in figure 1 a local data path $R_i \longrightarrow R_f$ consisting of two registers R_i ($R_{initial}$) and R_f (R_{final}) and a combinational block is considered. D_{PMin}^{if} and D_{PMax}^{if} represent the minimum and maximum propagation delays on the combinational block respectively. D_{CQ}^{i} represents initial register (R_i) clock-to-output delay, whereas S_f is setup time of R_f. The clock delays to the registers $\mathrm{R_i}$ and $\mathrm{R_f}$ are represented by parameters t_i and t_f, respectively, and T represents the clock period.

By complying with the setup and hold timing constraints the timing analysis is performed for each data path

Setup: $$t_i + D_{CQ}^{i} + D_{PMax}^{if} \le t_f + T - S_f - L_{if} \tag{2}$$

Hold: $$t_i + D_{CQ}^{i} + D_{PMin}^{if} \ge t_f \tag{3}$$

The placement and routing is done after choosing the time period T, which guarantees the setup and hold time constraints. The slackness in the setup time constraint is considered to be more vulnerable as the hold timing violations can be fixed by inserting delay on a data path. The timing slack L_{if} on each path in timing analysis can be evaluated as follows

$$L_{if} = T - S_f - D_{CQ}^{i} - D_{PMax}^{if} \tag{4}$$

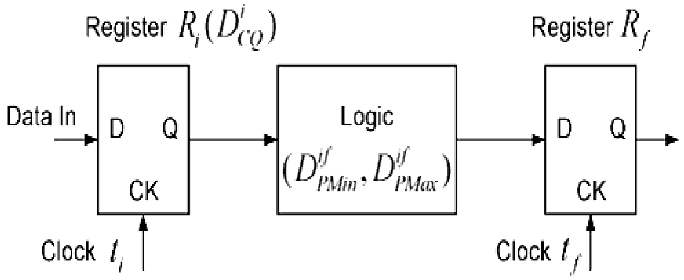

Figure-1
Register to register timing path illustration

Proposed Method

The proposed method modifies the work done in Lu,et.al.,[10]. A timing analysis is performed and placement blockages are generated. This method considers the usage of uniform mesh and thus the wire minimization is done by placing blockages. An algorithm is then formulated depending on the timing constraints to place the blockages. Finally, top level tree is generated by inserting buffer drivers. The stages of the proposed method are discussed through the following sections.

Feasible Moving Regions Generation: This method implements iterative placement of registers towards the clock mesh. The timing slacks are taken into consideration in order to ensure functional correctness of the design. The feasible moving region of each register is based on the timing slack to implement the clock mesh. The feasible moving region of a register R_{i} is generated on the assumption that the other registers are unmoved. The register placement is done later after the placement blockages are generated depending on the feasible moving regions of the registers.

The timing slack of any path is associated with the physical paths between the registers. Thus the movement of the registers affects the timing slack of the entire design i.e. only the initial register's fan-out and the final register's fan-in are affected. Thus the path delay D_{PMax}^{if} is composed of three parts as illustrated in figure 4.

$$D_{PMax}^{if} = D_{fo_k}^{i} + D_{m_k}^{if} + D_{fi}^{f} \tag{5}$$

$D_{fo_k}^{i}$ is the delay of the wire from output of the initial register R_{i} to the input of the k^{th} fan-out of the register R_{i} . The gates and the wire delay between the input of kth fan-out gate of register R_{i} and the fan-in gate input of register R_f is $D_{m_k}^{if}$. The delay between the input of the fan-in gate to the input of register R_f is D_{fi}^{f} .

Thus using (5) the timing slack L_{if} in (4) can be rewritten as follows:

$$L_{if} = T - S_f - D^i_{CQ} - D^i_{fo_k} - D^{if}_{m_k} - D^f_{fi} \qquad (6)$$

In order to ensure functional correctness the timing slack of each register should be non negative.

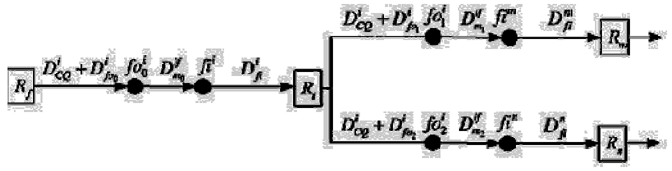

Figure-4
Delay compositions

Delays $D^i_{fo_k} + D^i_{CQ}$ and D^f_{fi} are monotonically increasing functions as they are dependent on the wire lengths $w^i_{fo_k}$ and w^f_{fi} i.e. the fan-out and fan-in wire length, respectively. Considering a non negative slack, the maximum wirelength $W^i_{fo_k}$ for every fan-out gate and maximum wirelength W^i_{fi} for each fan-in gate can be calculated for register R_i. The feasible moving region generation is illustrated in fig. 5. A rectangular region with a radius of W^i_{fo1} is generated such that the manhattan distance of any point on the register is equal to the radius $W^i_{fo_k}$. Thus as long as the register stays within the rectangular boundary the timing slack of the path is satisfied. This process is repeated for all the fan-ins and fan-outs associated with the register $R_{i.}$

After the creation of the feasible moving region, it is not necessary that the moving of the register always guarantees minimum timing slack due to the following reasons: the delay is generally affected by the total wirelength but here only the radius of the rectangular region is considered and when one register moves it changes the feasible regions of the other registers. In case of some registers the timing slack is best when unmoved. The iterative placement blockages presented in following section considers the feasible moving region generation and then generates the placement blockages.

Generation of Placement Blockages: The placement blockages are inserted so as not to allow free movement of registers due to several factors. Some of them include, aggressive movement may introduce overlapping and it may negatively affect the timing slack of the data path. In order to overcome these short comings a flow is proposed for placing blockages: i. Stub wire minimization problem as illustrated in, Integrated clock mesh synthesis with incremental register placement is used to obtain iterative register placement results[10]. ii. If the moving of registers introduces negative effect on the timing slack then a placement blockage is introduced. iii. If the feasible regions of the two regions are overlapping then depending on the overlapping constraints placement blocks are introduced. iv. If given two registers do not have any kind of intersection then the placement blockages are eliminated.

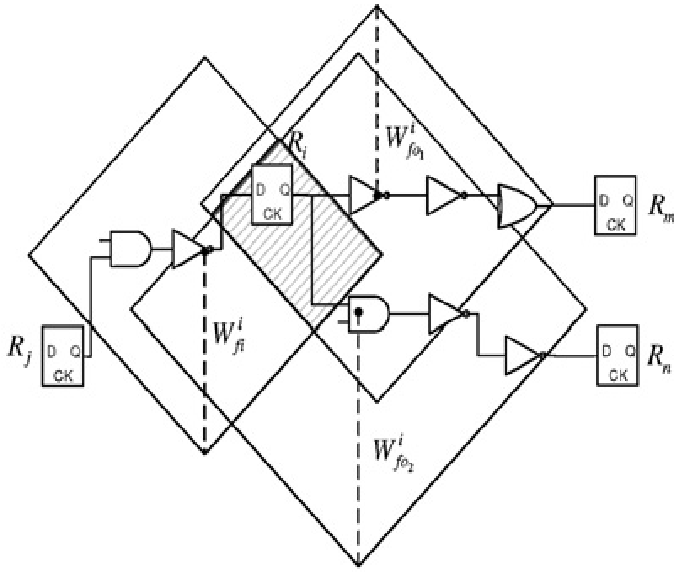

Figure-5
Feasible moving region of a register

Creation of Clock Mesh and placement: The clock mesh comprises of mesh network formed by horizontal and vertical placement of grids along with stub wires for connecting the registers to the grids. In this paper a uniform mesh is generated in order to minimize the complexity due to placement blockages. The main objective of the mesh is to minimize the global skew.

The mesh tracks are created on the chip area where mesh track represent the placement of the mesh wire on the chip. The figure 6 illustrates the placement of the mesh tracks as well as feasible moving regions. M_{hi} and M_{vi} represent the horizontal and vertical locations where mesh can be placed, respectively. The clock mesh implemented on s13207 benchmark circuit is illustrated in figure 7. The clock mesh is implemented over 6*6 mesh track.

Experimental Results

To drive the mesh grid a top level clock tree is generated using H-tree algorithm. The proposed method flow is implemented using encounter 11.13 of *cadence* in order to perform initial placement, routing, timing analysis and power analysis. The benchmark circuits over which the method in this paper is implemented are the five largest circuits from ISCAS'89 benchmark.

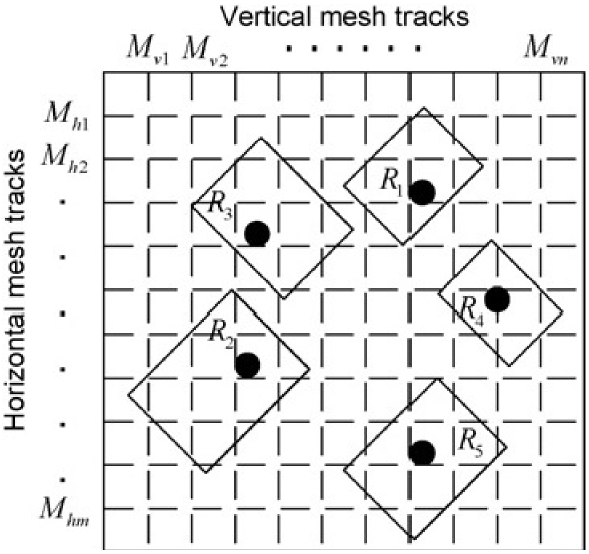

Figure-6
Mesh tracks and feasible moving regions

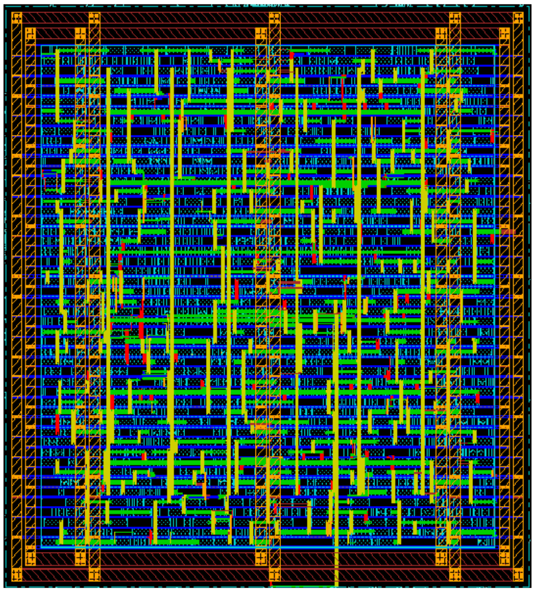

Figure-7
Clock mesh placement of s13207 benchmark circuit with 6*6 mesh

Table-1
Wire length comparison between proposed and optimized method in non -uniform clock mesh optimization with linear programming buffer insertion[7]

Circuit	Proposed method in clock mesh optimization with linear programming[7]			Proposed method			Improvement		
	Grid	*Stub wire (µm)*	*Mesh (µm)*	*Grid*	*Stub wire (µm)*	*Mesh (µm)*	*Stub*	*Mesh*	*Total*
s13207	8*8	3281	4848	8*8	489	3538	85.3%	27.06%	50.4%
S15850	8*8	2226	4062	8*8	288	2285	87.06%	43.7%	59.08%
S35932	12*12	10112	10871	12*12	1085	8157	89.2%	24.1%	55.9%
S38417	12*12	8839	10794	12*12	1392	8332	84.2%	22.8%	50.4%
S38584	11*11	8533	12668	11*11	774	10794	90.9%	14.7%	45.4%

Table-2
Capacitance and clock skew

Circuit	Proposed method in clock mesh optimization with linear programming[7]		Proposed method	
	Skew (ps)	*Cap (fF)*	*Skew (ps)*	*Cap (fF)*
S13207	0.7	5656	0.4	3127
S15850	0.8	4935	0.7	1837
S35932	0.7	13169	1.7	7112
S38417	1.4	12791	1.0	7280
S38584	1.1	13531	0.9	7428

The proposed placement of blockages flow is implemented using C. The LP formulation as proposed in, Integrated clock mesh synthesis with incremental register placement was implemented using *GLPK* in Linux machine[10]. The results obtained are compared with the results obtained in proposed method in clock mesh optimization with linear programming after performing the iterative register placement for uniform meshes[7]. Since the proposed flow doesn't consider the non uniform mesh placement only the uniform mesh analysis of the proposed method in, Non-uniform clock mesh optimization with linear programming buffer insertion is considered[9]. In order to demonstrate the reduction in the power consumption and the wire length these comparisons are performed.

The results demonstrated in table I illustrate the comparison of the bench mark circuits with uniform clock mesh. The stub wire reduction is by 89% and the mesh wirelength by 25%. The overall reduction on an average is by 53.2%.

The power and slack comparisons are shown in table II. The reduction in power is synchronous to the length of the wire reduced. The switching capacitance as well implies the power consumption in the circuit. Compared to the method used in, Non-uniform clock mesh optimization with linear programming buffer insertion [7] the total capacitance is reduced by 43.5% . The overall skew on an average is reduced by 0.3ps using the proposed method.

Conclusion

A clock mesh synthesis along with placement flow is proposed in the paper. This method considers power density as well as timing slack and creates placement blockages for iterative register placement for a uniform clock mesh. This method is able to reduce the power by reducing the switching capacitance by wirelength reduction. The proposed method discusses methodology flow on uniform meshes, further it can be extended to non uniform clock mesh and yet powerful algorithm for register placement can be formulated which allows simultaneous movement of registers. The results propose effective flow and can be integrated into industrial designs.

References

1. Desai M., Cvijetic R. and Jensen J., Sizing of clock distribution networks for high performance cpu chips, in *Proc. ACM/IEEE DAC*, 389–394 **(1996)**
2. Restle P.J., Mcnamara T.G., Webber D.A., Camporese P.J., Eng K.F., Jenkins K.A., Member S., Allen D.H., Rohn M.J., Quaranta M.P., Boerstler D.W., Alpert C.J., Carter C.A., Bailey R.N., Petrovick J.G., Krauter B.L. and Mccredie B.D., A clock distribution network for microprocessors, *IEEE J. Solid-State Circuits*, **36(5)**, 792–799 **(2001)**
3. Xanthopoulos J.T., Bailey D., Gangwar A., Gowan M., Jain A. and Prewitt B., The design and analysis of the clock distribution network for a 1.2 GHz Alpha microprocessor, in *Proc. IEEE ISSCC***,** 402–403 (**2001**)
4. Venkataraman G., Feng Z., Hu J. and Li P., Combinatorial algorithms for fast clock mesh optimization, *IEEE Trans. Very Large Scale Integr. Syst.*, **18(1)**, 131–141 **(2010)**
5. Rajaram A. and Pan D., MeshWorks: An efficient framework for planning, synthesis and optimization of clock mesh networks, in *Proc. ASPDAC*, 250–257 (**2008**)
6. Lu J., Mao X. and Taskin B., Timing slack aware incremental register placement with non-uniform grid generation for clock mesh synthesis, in *Proc. ISPD*, 131–138 (**2011**)
7. Guthaus M.R., Wilke G. and Reis R., Non-uniform clock mesh optimization with linear programming buffer insertion, in *Proc. ACM/IEEE DAC*, 74–79 **(2010**)
8. Shelar R.S., An algorithm for routing with capacitance/distance constraints for clock distribution in microprocessors, in *Proc. ISPD*, 141–148 (**2009)**
9. Lu J., Aksehir Y. and Taskin B., Register on MEsh (ROME): A novel approach for clock mesh network synthesis, in *Proc. IEEE ISCAS*, 1219–1222 (**2011**)
10. Lu J., Mao X. and Taskin B., Integrated clock mesh synthesis with incremental register placement, in *Proc. IEEE*, 217–227 **(2012)**
11. Sumitha J., Routing Algorithms in Networks, *Res. J. Recent Sci.,* **3(ISC-2013),** 1-3 **(2014)**