An Efficient Heuristic Algorithm For Fast Clock Mesh Realization

P.Saranya, A.Sridevi

Abstract— The application of multiple clocking domains with dedicated clock buffer will be implemented. In this paper, an algorithm is proposed for determining the minimum number of clock domains to be used for multi domain clock skew scheduling. Non-tree based distributions provides a high tolerance towards process variations. The clock mesh constraints are overcome by two processes. First a simultaneous buffer placement and sizing is done which satisfies the signal slew constraints while minimizing the total buffer size by heuristic algorithm. The second one reduces the mesh by deleting certain edges, thereby trading off skew tolerance for low power dissipation by post processing techniques. Thus comparisons of wire length, power dissipation, nominal skew and variation skews using H-SPICE software for various sized benchmark circuits are performed.

Index Terms—. Clock skew, Clock Distribution Network, Heuristic, Low Power Variation.

I. Introduction

Clock skew refers to the relative difference of the clock latencies of registers. Clock skew is one of the design parameter that is very sensitive to process variations.Clock skew computes a set of individual delays for the clock signals of the registers and latches of synchronous circuits to minimize the clock period. In practice, a clock schedule with a large set of arbitrary delays is becoming unreliable. This is because the implementation of dedicated delays using additional buffers and interconnections is highly susceptible to within-die variations of process parameters.

The function of clock distribution network is to deliver the clock signal from the clock source to the clock sinks.In the formulation of the problem, process variation is not a direct consideration. The fact is,this technique helps to make it practical to inject a limited number of skew values into the circuit. A fast and efficient combinatorial algorithm is presented to design or optimize the distribution network and is also proposed that it can be used at design time to decide the number of distribution network to be used and their associated skew values such that the number of required buffers, as well as the resulting clock period, is minimized.Our proposal includes the following contributions:

P.Saranya, P.G.Scholar
Department of Electronics &Communication Engineering,
SNS College of Technology,Coimbatore, India
Email: digisaran@gmail.com
A.Sridevi
Department of Electronics &Communication Engineering,
Assistant Professor,
SNS College of Technology,Coimbatore, India
Email:srideviarumugam07@gmail.com

i. To find the mesh buffer locations and their sizes using greedy algorithm on a discrete set of libraries of buffer sizes.
ii. The process of removing the noncritical wire segments in a clock mesh to minimize the power dissipated was to be formulated by network survivability theory.
iii. We have to achieve the most desirable skew-power tradeoff, which becomes flexible enough to allow a high range of tradeoff.

Higher skew would bring down the maximum permissible delay. A mesh architecture is suited well for high-performance systems since it mitigates clock skew at the expense of high resource consumption. With power occupying an increasingly important role in chip design, it is necessary to find the most desirable skew-power tradeoff.

The remainder of this paper is organized as follows. Section II contains the list of Literature Survey done. Section III comprises of the methodology used in the work. Section IV gives the results and discussions. Section V gives the work done and conclusion.

II. Related Work

In combinatorial optimization, the most important challenges are presented by problems belonging to the class NP-hard. The solution returned will be within a relative distance from the optimum. The design of approximation algorithm relies in a number of techniques, which usually involve clever analysis of the relaxation of integer problems to linear or semidefinite formulations. One of the motivations for developing approximation algorithms is to find good solutions for problems that are provably difficult. Another motivation is to better understand the intrinsic difficulty of the problems.[11] Link based non-tree clock distribution is a cost-effective technique for reducing clock skew variations. It is limited to unbuffered clock networks and neglected spatial correlations in the experimental validation. We overcome the shortcomings and make the link based non-tree approach feasible for realistic designs. The short circuit risk and multi-driver delay issues in buffered non-tree clock networks are examined. Skew tuning is used to synthesize a clock tree with low nominal skew under a higher order delay model. The effect of link insertion depends on a well-designed buffered clock tree which enhances the effectiveness of link insertion. Link insertion in a buffered network may result in multiple drivers for a subnet.[7]. Process variation, RLC delay matching and scalability with die size and process generation are the classic challenges in high speed clock distribution design. An architecture for achieving sub-10ps global clock uncertainty that addresses each one of these issues without additional clock jitter or layout area is presented. Included is a method to limit global clock skew to a single inverter stage delay independent of die size and management of practical constraints due to schedule, changing floor plan, die size, clock loading and process independence. Wirelength of clock routing trees should be minimized in order to reduce system power requirements and deformation of the clock pulse at the synchronizing elements of the system. The deferred-merge embedding (DME) algorithm, embeds any given connection topology to create a clock tree with zero skew while minimizing total wirelength. The algorithm always yields exact zero skew trees with respect to the appropriate delay model. The circuit speed is increasingly limited by two factors: i) delay on the longest path through combinational logic, and ii) clock skew, which **is** the maximum difference in arrival times of the clocking signal at the synchronizing elements of the design by the inequality governing the clock period of **a** clock signal.[3] A mesh construction procedure which guarantees Zero skew under the Elmore delay model, using a simple and efficient linear programming formulation. Buffers are inserted to reduce the transition time (or rise time). As a post-processing step, wire width optimization under an accurate higher-order delay metric is performed to further minimize the transition time and propagation delay skew[12]. Thus the hybrid mesh tree construction scheme can provide smaller propagation delay and transition time than a comparable clock tree.

III. Preliminaries

Certain conventions and notations that are to be used in this paper are introduced below as

i. Clock mesh has set of nodes with dimension A_n x B_n. Buffer sizes ranges from 1 to K in a non-decreasing order. The ith buffer can drive a load capacitance of atmost C_i.

ii. The buffer mapping functions maps each node location i and j to set of nodes and buffer sizes respectively. The problem implies that the ith value of buffer mapping function becomes null, thus it has no buffer in the ith location of the clock mesh.

iii. The set of clock sinks ranges from S_1 to S_c without any loss of generality, thereby every buffer is connected to the set of nodes in the clock mesh. But these clock sinks are connected to the closest point in the mesh which are not available in the covering regions.

iv. The minimum distance between nodes i and j in the mesh is given by A_{ij}.The maximum permissible delay, P^{ij}_{delay} between two registers i & j is given as,$P_{delay}=\min_{(i,j)}P^{ij}_{delay}$.

IV HEURISTIC ALGORITHMS

A. Algorithm Description

A greedy algorithm is any algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding the global optimum. For example, applying the greedy strategy to the traveling salesman problem yields the following algorithm: "At each stage visit the unvisited city nearest to the current city". In general, greedy algorithms have five pillars:

1. A candidate set, from which a solution is created.
2. A selection function, which chooses the best candidate to be added to the solution.
3. A feasibility function, that is used to determine if a candidate can be used to contribute to a solution.
4. An objective function, which assigns a value to a solution, or a partial solution.

A solution function, which will indicate when we have discovered a complete Greedy algorithms appear in network routing. Using greedy routing, a message is forwarded to the neighbouring node which is "closest" to the destination. The notion of a node's location (and hence "closeness") may be determined by its physical location, as in geographic routing used by ad-hoc networks. Location may also be an entirely artificial construct as in small world routing and table. A greedy algorithm can be thought of as a backtracking algorithm where at each decision point "the best" option is already known and thus can be picked without having to recurse over any of the alternative options. Greedy algorithms tend to be very efficient and can be implemented in a relatively straightforward fashion. Many a times in O(n) complexity as there would be a single choice at every point solution.

Clock mesh implementation requires an array of mesh drivers, shown in fifth level of second block in fig1 is to drive the massive RC network of the clock mesh. The benefit of the mesh net is that it smoothes out the arrival time differences from the multiple mesh drivers that drive it. The top trace is the ideal clock, the top pair of traces shows the skew just before the mesh, and the bottom pair of traces shows the skew just after the mesh.

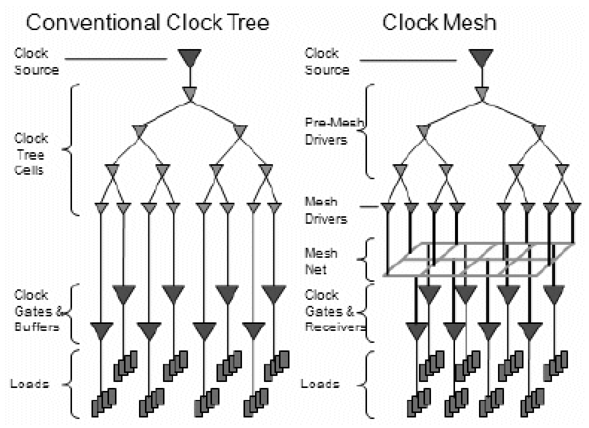

Fig1. Clock Structures - Conventional clock tree and clock mesh

B. Node Co-ordinate Determination

```
Input   :N,Number of buffers to be tested.

Output:S,Set of Integrated nodes.

    1. Initialize S = null
    2. Determine the mesh dimension   $A_n$ X $B_n$
    3. Define the buffer sizes,K.
    4. Calculate Effective capacitance $C_E$= K/|(CR - S)|
    5. Choose least $C_E \rightarrow LC_E$

    6. $S \leftarrow S$ U $LC_E$
```

Fig2. Node co-ordinate integration for mesh by greedy

For the buffer mapping function to be estimated : 1) the requirement of buffer is noted. 2) the size of buffer needed with the each node in the mesh allocated to at least one buffer and each buffer driving less than the maximum load it can drive.The Covering Region(CR) of the node for a particular buffer is defined as the set of nodes around the node in the 2-D mesh such that the total capacitance of the nodes included in the covering region is less than the maximum capacitance that the buffer can drive. If any more edges in the mesh are added, then the capacitance of the region will be greater than the maximum load that the buffer can drive. The greedy algorithm pick the set that covers the most nodes and then throw away the nodes that are covered. The process is repeated until all nodes are covered.

The algorithm may return two buffers for the same location, which is not a feasible solution. Thus, such a situation can be easily avoided by using the observation.

Observation 1: A bigger buffer size can drive a bigger load.For any node i with solution Φ, if there are more than one buffer driving a node,one can pick the biggest buffer without losing feasibility.

C. Steiner Reduction

After the integration of the position of the mesh buffers and their sizes ,the next task is to reduce the size of the mesh. This is done by removing edges such that a certain level of redundancy is still maintained. Remove edges from the mesh such that 1)each sink S_c has at least k node locations such that for each such node locations j, $A_{ij} \leq L_{max}$, BM(j) ≠ Φ and there exists at least l edge disjoint paths between j and i and 2) the number of edges removed is maximized.

```
Input   : The buffer cost and node to be tested.

Output : $B_R$ , Reduced buffer cost.

    1. Define the cost of node, e.
    2. Initialize $B_R \leftarrow \Phi$
    3. Find K nearest nodes for the tested nodes.
    4. Determine the minimum cost , $l_{min}$
    5. For each e □ $l_{min}$
    6. $B_R \leftarrow B_R$ U e
```

Fig3. Steiners Reduction

The user defined parameters control impact the solution as k and l high would mean more redundancy and hence more tolerance to variations but less number of edges removed or more power dissipation. Thus after mesh reduction the buffers could be driving a load that is significantly lesser than that of a complete mesh. Hence, as a postprocessing step we compute the new load and downsize the buffers accordingly.

V EXPERIMENTAL RESULTS

In order to mention the effectiveness of the aforementioned techniques, we conducted the following algorithms 1) Compare our node co-ordinate integration greedy algorithm for various sizings. We do the comparisons for the minimum, medium and maximum sizing of buffers, 2) Comparison of our Steiner reduction algorithm with that of the complete meshes, 3) Finally the run time results are also achieved and compared against various sizing of buffers in a complete mesh.

The greedy and Steiner reduction algorithms were implemented in C++ and the simulations were generated on a Linux work station .The test cases of the benchmark circuits were simulated in HSPICE using 180nm process model from Berkeley Predictive Technology Model and the results are compared for various sizing of circuits.

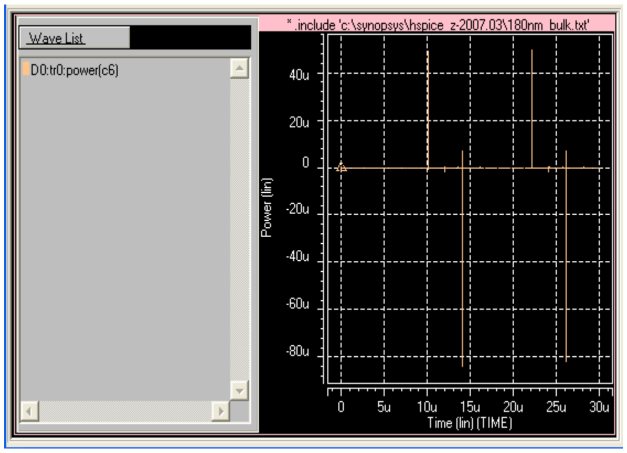

Fig 4. Output waveform with minimum size buffer

The run time of our algorithm is within a few seconds as verified from the Table1.It becomes one of the fine tuning technique and not an optimizing procedure. Such a reduction in area though small could be significant in high performance designs. Hence the minimum buffer sizing achieves the low power dissipation of about 40u approximately while the medium and maximum sizing achieves with a larger power as obtained in waveform simulation.

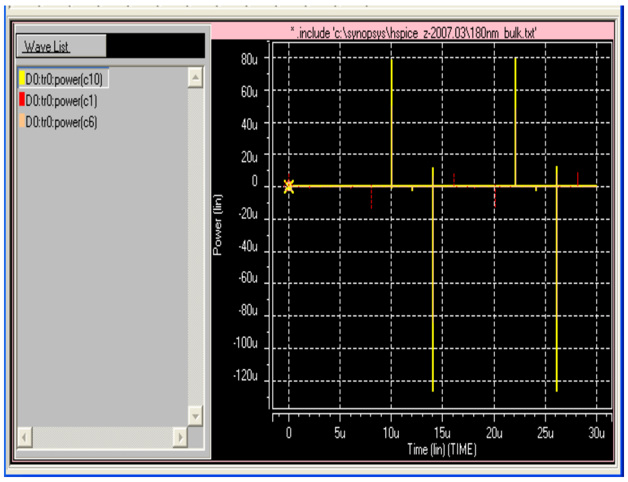

Fig 5. Output waveform with medium size buffer

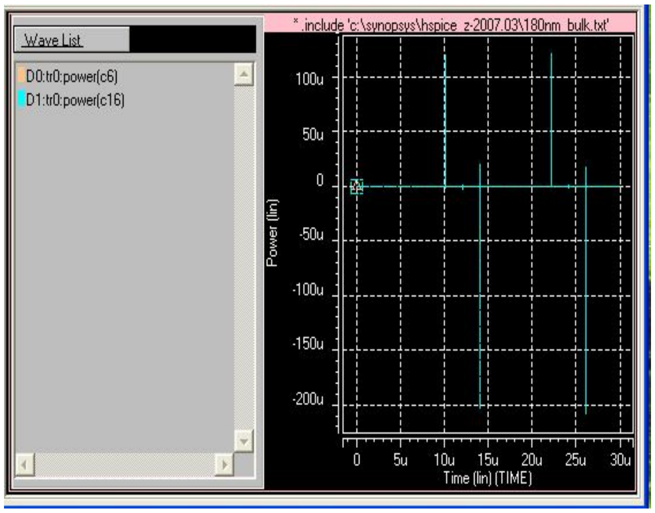

Fig 6. Output waveform with maximum size buffer

Thus the Fig4 ,Fig5 and Fig 6 gives the power versus clock time trade-off curve for test case S9234 .The Y-axis measures the power dissipation and X–axis measures the time value for the various sizing of the circuits done using heuristic algorithms by a HSPICE model. The Table1 provides the comparison of parameters (memory,cputime,etc.,) for various sizings. Thus the following inferences can be made as Using the minimum buffer size results in an area reduction of about 10% comparitively to other sizings. Medium buffer sizes satisfy slew constraints in all but except of one case whereas large buffer size satisfies the slew constraints in all the cases. But it can be achieved only with an area penalty of 18%(35%) for medium (maximum)buffer sizes.

TABLE 1

RESULTS FOR BUFFER SIZING FOR TEST CASE S9234

Test case – S9234	Sizing		
	Minimum	Medium	Maximum
Memory used	1097 kbytes	1865 kbytes	3547 kbytes
CPU Time	3.8 sec	6.48 sec	14.01 sec
Transient time	2.65	5.67	12.96
Mosfets defined	150	290	560
Total Iteration	9900	10894	12130
Output analysis	0.39	0.66	0.86
Operating point	0.05	0.07	0.11
Nodes	141	273	531
Elements	308	592	1138

VI CONCLUSION

The research work has been focused on sizing the interconnect elements within the clock mesh. We believe that sizing the interconnect wire segments of the clock mesh and the buffers driving the mesh simultaneously would yield more improvement in the power of a clock mesh area satisfying the constraints. The work presented here can be easily extended for sizing buffers and mesh elements simultaneously. Since, our design techniques are faster, it offers the flexibility to optimize clock mesh with different design objectives. The slew constraints are also satisfied while minimizing the total buffer size in a simultaneous buffer placement and sizing module. The number of mesh in the distribution network is also reduced by deleting certain edges, thereby trading off skew tolerance for low power dissipation. The various benchmark circuits are considered and their performance measures are determined by above heuristic algorithms.

REFERENCES

[1] J. Burkis, "Clock tree synthesis for high performance ASIC's," in *Proc. IEEE Int. Conf ASIC,* pp. 9.8.1-9.8.4, 1991.
[2] P. K. Chan and K. Karplus, "Computing signal delay in general RC networks by tree/link partitioning," *IEEE Trans. Computer- Aided Design,* pp. 898-902, Aug. 1990.
[3] S. Dhar, M. A. Franklin, and D. F. Warm, "Reduction of clock delays in VLSI structures," in *Proc. IEEE Int. Conf Computer Design,* pp. 778-783, 1984.
[4] A. Rajaram, D. Pan, and J. Hu. Improved algorithms for link based non-tree clock network for skew variability reduction. In Proceedings of the ACM International Symposium on Physical Design, pages 55{62, 2005}.
[5] M. Edahiro. A clustering-based optimization algorithm in zero-skew routings. In Proceedings of the ACM/IEEE Design Automation Conference, pages 612{616, 1993}.
[6] S. Pullela, N. Menezes, J. Omar, and L. T. Pillage. Skew and delay optimization for reliable buffered clock trees. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pages 556{562, 1993}.
[7] J. G. Xi and W. W.-M. Dai. Buffer insertion and sizing under process variations for low power clock distribution. In Proceedings of the ACM/IEEE Design Automation Conference, pages 491{496, 1995}.
[8] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555. Prelim. version in FOCS'92.
[9] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, And M. Protasi, *Complexity and approximation*, Springer-Verlag, Berlin, 1999. Combinatorial optimization problems and their approximability properties, With 1 CD-ROM (Windows and UNIX).
[10] U. FEIGE, *A threshold of* ln n *for approximating set cover (preliminary version)*, in Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996), New York, 1996, ACM, pp. 314–318.
[11] *Approximation thresholds for combinatorial optimization problems*, in Proceedings of the International Congress of Mathematicians, Vol. III (Beijing, 2002), Beijing, 2002, Higher Ed. Press, pp. 649–658.
[12]S. Pullela, *et al.,* "Skew and Delay Optimization for Reliable Buffered Clock Trees," *Proceedings of the /€€HACM Intemational Conference on Computer-Aided Design,* pp. 556-562, 1993.
[13] J.P.Fishbum, "Clock skew optimization," *IEEE Transactiom on Computers,* vol. 39, pp. 945-951,J ul. 1990.