Statistical Timing Driven Partitioning for VLSI Circuits

Cristinel Ababei, Kia Bazargan
ECE Department, University of Minnesota, Minneapolis, MN 55455
{ababei, kia}@ece.umn.edu

Abstract – In this poster we present statistical-timing driven partitioning for performance optimization. We show that by using the concept of *node criticality* we can enhance the Fiduccia-Mattheyses (FM) partitioning algorithm to achieve, on average, around 20% improvements in terms of timing, among partitions with the same cut size. By incorporating mechanisms for timing optimization at the partitioning level, we facilitate wire-planning at high levels of the design process.

1. Introduction

Partitioning is an early and very important step during the design process. Until now, all previous timing-driven partitioning approaches used *static* timing analysis. In this poster, we propose to use the *statistical* timing analysis incorporated in two modified versions of the FM algorithm [1]. This methodology incorporates a better gate and wire delay models at the partitioning level and provides better estimates of the timing of the circuit.

2. Statistical Timing Driven Partitioning

In order to model uncertainties in both gate and wire delays (such as fabrication variations, changes in supply voltage and temperature), the statistical timing analysis considers gate and wire delays as stochastic variables with given means and standard deviations. We adopt the statistical timing analysis method presented in [2] due to its introduction of the *criticality* concept that fits well into the partitioning framework. Criticality is meant to represent the timing criticality at each gate, i.e. the contribution to the circuit delay of all paths that pass through that gate. The gate with the largest criticality in a circuit is the most critical in terms of timing since its contribution to the circuit output delays is the most significant among all gates in the circuit.

Our modified FM partitioning algorithms use the statistical timing analysis to compute the criticality of each gate in the circuit. Then the criticality is used in two different ways to guide the FM algorithm such that the circuit delay is minimized.

Assuming that the reader is familiar with the FM algorithm and its data structure, the following two simple strategies describe the two proposed statistical timing driven partitioning algorithms.

- *Strategy I*. During each pass, we record the cut-criticalities of the cells that are moved. When choosing the sequence of moves to accept, we pick the one that offers minimum cut-criticality at no more than a user set percentage decay in cut size, compared to the standard FM algorithm.
- *Strategy II*. We first order all hyperedges in non-increasing magnitude of their weights, i.e. criticalities, and then constrain the standard FM algorithm not to cut any of the critical hyperedges. A hyperedge is *critical* if it is among the first DONOTCUT% (set by the user) of all hyperedges ordered in non-increasing order of their associated criticalities. All vertices connected by critical hyperedges are called *critical* cells.

3. Experimental Results

We observe that both proposed modified FM partitioning algorithms (Str. I and Str. II in Table 1) lead to better timing compared to the standard FM algorithm, at the expense of increase in the cutsize. The run time of the second modified FM algorithm is the same as that of the standard FM algorithm and slightly larger for the first modified FM algorithm.

Table 1:Recursive partitioning, balance ratio=0.48, 10 random runs, minimum timing reported

		FM			Str.I			Str.II		
Circuit	No. cells	Cutsize	Cutcrit	Delay	Cutsize	Cutcrit	Delay	Cutsize	Cutcrit	Delay
rd84	481	269	33.42	11.37	261	30.26	**10.97**	273	29.74	11.26
table3	686	426	185.26	16.57	421	168.7	**15.92**	444	142.7	16.22
cordic	856	383	18.29	41.33	361	15.62	40.53	437	14.59	**38.58**
apex2	2388	979	33.01	133.42	965	30.13	130.66	1444	14.59	**79.82**
des	2557	424	1493.1	193.71	446	1502	193.71	971	703.18	**173.13**
	Av.	**496.2**	**352.6**	**79.2**	**490.8**	**349.3**	**78.3**	**713.8**	**180.9**	**63.8**

References

[1] C.M. Fiduccia, R.M. Mattheyses, 'A Linear-Time Heuristic for Improving Network Partitions', *Proc. ACM/IEEE DAC*, 1982.

[2] M. Hashimoto, H. Onodera, 'A Performance Optimization Method by Gate Resizing Based on Statistical Static Timing Analysis', *IEICE Trans. Fundamentals*, Dec. 2000.

Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition (DATE'02)
1530-1591/02 $17.00 © 2002 **IEEE**