INTEGRATION the VLSI journal 56 (2017) 115–127

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Fast clock scheduling and an application to clock tree synthesis

Rickard Ewetz*, Cheng-Kok Koh

School of Electrical and Computer Engineering, Purdue University, 1285 Electrical Engineering Building, West Lafayette, IN 47907-2035, United States

ARTICLE INFO

Keywords:
Algorithms
Clock network synthesis
Low-power
Physical design
Skew

ABSTRACT

Clock networks are required to be constructed with adequate safety margins in the skew constraints to operate correctly even under the influence of variations. In this work, a scalable clock scheduler is developed to drive a synthesis framework that constructs useful skew clock trees with large safety margins that are tailored to the tree topology. Sequential elements are clustered early in the topology, if it is impossible provide adequate robustness to variations using only safety margins. Compared to earlier studies, the proposed framework performs the clock scheduling one to two orders of magnitude faster and improves yield and capacitive cost on several synthesized circuits.

1. Introduction

In the synthesis of clock networks for modern VLSI circuits, it is becoming more important to exploit the non-uniformity of the slacks in the timing constraints as process, voltage and temperature variations continue to diminish the available timing margins. The clock network of a VLSI circuit is constructed after an initial placement and delay estimation of the data and control paths. Skew is the difference of the arrival time of the clock signal between a pair of sequential elements. The clock network must meet setup and hold time constraints (i.e., skew constraints) imposed by the data and control paths. Moreover, the skew constraints must be satisfied even when the clock network is under the influence of variations. Such a problem is the emphasis of two clock contests organized by the International Symposium on Physical Design in 2009 [3] and 2010 [4]. However, the two contests considered uniform and regular skew constraints. In reality, the slacks in the skew constraints imposed by the data and control paths are typically non-uniform and it is necessary to use useful skew to obtain high yield.

The Greedy-useful skew tree/deferred-merge embedding (Greedy-UST/DME) algorithm was proposed to construct clock trees meeting arbitrary skew constraints [1]. The algorithm is based on integrating tree construction and dynamic clock scheduling. Clock scheduling is the process of specifying the arrival time of the clock signal to the sequential elements, or the clock sinks. In dynamic clock scheduling, a clock schedule is incrementally specified by committing the skew between pairs of sequential elements. It was shown that if and only if each skew is committed within an feasible skew range (FSR), the skew constraints are satisfied. In the Greedy-UST/DME algorithm, the skew constraints are captured in a skew constraint graph (SCG) and a clock tree is constructed by joining subtrees pair-wise to form larger subtrees. Moreover, subtrees are joined such that the skew of the each subtree-pair is committed within the FSR of the pair, and the FSRs are found by computing shortest paths in the SCG.

To satisfy skew constraints under variations, safety margins can be inserted in the skew constraints. In [5], it was shown that there exists a maximum uniform safety margin M that can be inserted uniformly in the skew constraints. In [1], a user specified uniform safety margin $M_{user} \leq M$ was inserted, which resulted in a smooth trade-off between capacitive cost and robustness to variations. In [2], safety margins of different magnitudes were inserted in different skew constraints during the construction of a clock tree. Smaller safety margins were inserted closer to the bottom of the tree and larger safety margins were attempted to be inserted closer to the top of the tree. The drawbacks of these works are that the dynamic clock scheduling is not very scalable (practically limited to designs with a few thousand sequential elements) and yield loss may be suffered due to inadequate safety margins.

We propose a framework to construct useful skew clock trees with large tailored safety margins (UST-LTSM), driven by a scalable clock scheduler. The main innovations of the framework are: (1) a dynamic clock scheduler that is based on sparse-graph algorithms. (2) Robustness to variations is provided by combining the use of safety margins and placing certain sequential elements close in the tree topology to lower the point of divergence. (3) The inserted safety margins are tailored to the tree topology by reducing safety margins that are larger than required.

Compared with the clock scheduler in [1], our proposed clock

[*] Corresponding author. Present address: 4328 Scorpius St., ECE Department, University of Central Florida, Orlando, FL 32816-2362, United States.
E-mail addresses: rewetz@purdue.edu (R. Ewetz), chengkok@purdue.edu (C.-K. Koh).

http://dx.doi.org/10.1016/j.vlsi.2016.10.012
Received 28 April 2016; Received in revised form 7 September 2016; Accepted 13 October 2016
Available online 19 October 2016
0167-9260/

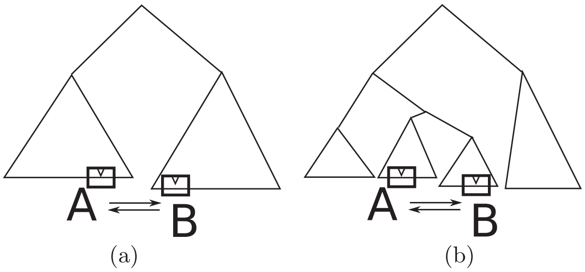

Fig. 1. Sequential elements *A* and *B* have tight skew constraints. If *A* and *B* are distant in the topology, as in (a), the safety margin may be inadequate. The two elements should be placed closer, as in (b), in the topology.

scheduler utilizes that the SCG is sparse, which results better scalability. We overcome the limitation of $M_{user} \le M$ in [1]. Specifically, we ensure that the full safety margin M_{user} is provided in a majority of the skew constraints and in skew constraints where** it is impossible to provide a safety margin M_{user}, a safety margin of at least M is provided and the corresponding sequential elements are clustered early in the tree construction process. Compared with the tailoring of safety margins in [2], our tailoring techniques can never "run-out" of safety margin at the top of the clock tree. The techniques are also scalable as they are based on the developed dynamic clock scheduler. Moreover, we avoid constructing tree topologies where an adequate safety margin can never be provided in certain tight skew constraints, as is illustrated in Fig. 1.

The experimental results show that the proposed clock scheduler is one to two orders of magnitude faster compared with schedulers in previous studies. On a benchmark circuit with 13,216 sequential elements, the framework provides a yield of 97%.

The remainder of the paper is organized as follows: the problem formulation is introduced in Section 2 and previous studies are reviewed in Section 3. An overview of the proposed dynamic clock scheduling and UST-LTSM framework is presented in Section 4. The details of dynamic clock scheduling and the UST-LTSM framework are given in Sections 5 and 6, respectively. Finally, the experimental results are explained in Section 7 and the paper is concluded in Section 8.

2. Problem formulation

This paper considers the problem of constructing a variations aware useful skew clock tree (VA-UST). For a clock tree to be able to satisfy timing constraints while under the influence of variations, safety margins are required to be provided in the skew constraints. Therefore, we present the setup and hold time constraints and the insertion of safety margins before introducing the VA-UST synthesis problem. Next, we present the dynamic clock scheduling problem because it is a key component of constructing useful skew clock trees.

2.1. Skew constraints and safety margins

Setup and hold time constraints are imposed between flip flops (FFs) that are only separated by combinational logic. A launching and a capturing flip flop (denoted as FF_i and FF_j, respectively) are illustrated in Fig. 2(a) and the setup and hold time constraints of the flip flop pair are formulated as follows:

$$t_i + t_i^{CQ} + t_{ij}^{max} + t_j^S \le t_j + T, \tag{1}$$

$$t_i + t_i^{CQ} + t_{ij}^{min} \ge t_j + t_j^H. \tag{2}$$

Here, t_i and t_j are respectively the arrival times of the clock signal to FF_i and FF_j; t_i^{CQ} is the clock to output time of FF_i; t_{ij}^{max} and t_{ij}^{min} are the maximum and the minimum propagation delay, respectively; t_j^S and t_j^H are the setup and the hold time of FF_j, respectively. The constraints in Eqs. (1) and (2) are reformulated with respect to the skew $skew_{ij} = t_i - t_j$ using $l_{ij} = t_j^H - t_i^{CQ} - t_{ij}^{min}$ and $u_{ij} = T - t_i^{CQ} - t_{ij}^{max} - t_i^S$ to obtain

$$l_{ij} \le skew_{ij} \le u_{ij}. \tag{3}$$

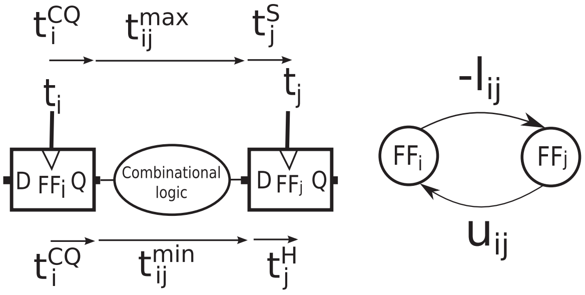

Fig. 2. (a) Two flip flop circuit. (b) SCG of the two flip flop circuit in (a).

The slacks in the lower and upper bound of the skew constraint are formulated as $skew_{ij} - l_{ij}$ and $u_{ij} - skew_{ij}$, respectively. The skew constraints in Eq. (3) can be captured in a skew constraint graph (SCG). In an SCG, $G = (V, E)$, the vertices V represent the flip flops and the edges E represent the skew constraints. A directed edge e_{ij} with a weight $w_{ij} = -l_{ij}$ is added from vertex i to vertex j and a directed edge e_{ji} with a weight $w_{ji} = u_{ij}$ is added from vertex j to i. In Fig. 2(b), the SCG of the two flip flop circuit in Fig. 2(a) is shown. A feasible clock schedule exists if the SCG does not contain any negative cycles. Moreover, one such feasible clock schedule can be determined using the Bellman–Ford algorithm [6].

To provide a (non-negative) safety margin to variations, the skew constraints in Eq. (3) can be tightened with non-uniform safety margins m_{ji} and m_{ij}, respectively, to obtain:

$$l_{ij} + m_{ji} \le skew_{ij} \le u_{ij} - m_{ij} \tag{4}$$

Note that inserting an safety margin in a skew constraint corresponds to reducing the weight of an edge in the SCG, which may create a negative cycle.

2.2. Variations-aware useful skew clock tree (VA-UST) problem

The VA-UST problem is based on an adaptation of the problem formulation in the International Symposium on Physical Design 2010 contest [4]. The main difference is that we consider arbitrary skew constraints.

The problem consists of constructing a clock tree that delivers a synchronizing clock signal from a clock source to a set of clock sinks. The source-to-sink connections are realized using wires and buffers from a wire library and a buffer library, respectively. The clock tree is to be constructed such that the clock signal is delivered meeting the skew constraints specified in Eq. (3), while under the influence of variations. The key idea is to construct a clock tree with adequate safety margins provided in the skew constraints, as detailed in Eq. (4). In particular, the safety margins m_{ij} and m_{ji} in the skew constraints between FF_i and FF_j must be larger than the maximum skew that can be caused by variations between the two flip flops. (The details of the Monte Carlo framework that is used to evaluate the timing performance under variations are given in Section 7.1.)

Moreover, to ensure sharp rise and fall times, the clock network must also satisfy a transition time constraint. Transition time is the 10–90% rise or fall time of the clock signal and the transition time at every point in the clock network must be less than a constraint parameter S_{tran}.

2.3. Dynamic clock scheduling problem

A related problem to the VA-UST problem is the dynamic clock