Statistical Retiming Based Timing Analysis and Application to Placement

Yu Hu

January 15, 2006

1 Introduction

[1] proposed the conception of retiming based timing analysis (RTA) for sequential circuits performance optimization in deterministic scenario. A sequential arrival time (SAT) and required time (SRT) are found in RTA and the sequential slacks (SRT-SAT) are calculated for each timing edge and used as the weight in the placement. The minimal clock period is found by binary search. If the interconnect delay is considered, the possible clock periods change in a continuous range, which means that the binary search approach can not obtain an accurate solution. To attach this problem, a push down based approach is proposed in [3], which calculates the optimal clock period and retiming without binary search. The basic idea is to move FFs from the downstream to upstream iteratively, which has been proved to has a worse complexity of $O(|V|^2|E| \cdot (N+1))$, where V and E is vertices and edges set in retiming graph and N is the total FF number. However no process variations are considered in [3].

In the statistical scenario, [4] is the first work attack the statistical retiming. Unfortunately, interconnect delay is not considered and the retiming is not leveraged with other design process, e.g. placement and routing. The most recently work [2] follows the RTA model in [1] and performs it with statistical calculation. However, this work suffers from the following weakness. Firstly, it can not be performed in an incremental fashion as binary search is employed to find an optimal clock period. Secondly, the timing weight in the proposed work is based on the sequential slack. However, it is hard to compare the slacks when they become random variables in the statistical scenario. Of course we can compare two slacks by employ a certain function, e.g. $f(a) = a_0 + 3\sigma^2$, but it is a heuristic way. Thirdly, the retiming values can not be obtained as the ceiling and dividing operation in the deterministic scenario are not easy to extend to handle random variables.

In this paper, we study on the statistical simultaneously retiming and placement for performance optimization. The simulated annealing based placer is used. In each temperature, a statistical retiming based timing analysis (will be detailed in the next section) is performed to calculate the statistical criticality

of each timing node[1]. The edge cost will be updated according to the statistical criticality so that the placer will be expected to improve the timing yield iteratively.

2 Statistical Push Down Based Retiming Algorithm

In the retiming graph $G = (V, E)$, each edge $e(u, v)$ is associated with a delay value $d(u, v)$, which is a random variable in statistical scenario, and a weight $w(u, v)$ denoting the number of FFs in this edge. Each node v is associated with a retiming value $r(v)$ and a arrival time $t(v)$. Based on [3], we improve the algorithm to handle statistical delay. The following is the pesudo-code for the proposed algorithm and the underlined lines are statistical operations that we need to pay attention to.

Algorithm 1 INIT(G,r)

1: $w_r(u, v) \leftarrow w(u, v) + r(v) - r(u), \forall (u, v) \in E$;
2: $t(v) \leftarrow 0, \forall v \in V; T \leftarrow 0; Q \leftarrow V$;
3: **while** $Q \neq \emptyset$ **do**
4: $u \leftarrow$ dequeue(Q);
5: **for** each $(u, v) \in E$ **do**
6: **if** $w_r(u, v) = 0$ **then**
7: $\underline{t(v) \leftarrow \max(t(v), t(u) + d(u, v))}$;
8: $\underline{T \leftarrow \max(T, t(u) + d(u, v))}$;

References

[1] J. Cong and S. K. Lim, "Retiming-based timing analysis with an application to mincut-based global placement," *TCAD*, 2004.

[2] M. Ekpanyapong, T. Watewai, and S. K. Lim, "Retiming-based timing analysis with statistical bellman-ford algorithm," in *ASPDAC*, 06.

[3] C. Lin and H. Zhou, "Optimal wire retiming without binary search," in *iccad*, 04.

[4] J. Wang and H. Zhou, "Minimal period retiming under process variations," in *GLSVLSI*, 04.

[1]The statistical criticality means the probability of the edge lying on the critical path

Algorithm 2 SRTA(G) INPUT: A retiming graph $G = (V, E)$, OUTPUT: Retiming values, criticality and statistical minimal clock period

INIT(G,0); $T^{opt} \leftarrow T; r^{opt}(u) \leftarrow 0, \forall u \in V$;
while true **do**
  {Identify critical edges in E}
  $E_c \leftarrow \{(u,v) \in E | t(v) = t(u) + d(u,v) - w_r(u,v)T\}$;
  if E_c contains a cycle **then**
    Report T, r and exit;
  Topological sort $G_c = (V, E_c)$;
  {Compute max FF number on paths from roots in G_c}
  for $v \in V$ in topological sort order of G_c **do**
    if v is a root in G_c **then**
      $\Delta(v) \leftarrow 0$;
    else
      for each $(u,v) \in E_c$ do
      $\Delta(v) \leftarrow \max(\Delta(v), \Delta(v) + w_r(u,v))$
  $\theta \leftarrow \infty$;
  for each $(u,v) \in E$ **do**
    if $(\Delta(u) + w_r(u,v) > \Delta(v))$ **then**
      $\theta \leftarrow \min\{\theta, \frac{t(v)-t(u)-d(u,v)+w_r(u,v)T}{\Delta(u)+w_r(u,v)-\Delta(v)}\}$;
  for each $v \in V$ **do**
    $\theta \leftarrow \min\{\theta, \frac{T-t(v)}{\Delta(v)+1}\}$;
    if $(\theta = 0)$ **then**
      $r(v) \leftarrow r(v) + 1$;
      ADJUST(G, v);
      if $\forall r > 0 \vee \exists r = \sum_{(u,v)\in E} w(u,v)$ **then**
        Report T^{opt}, r^{opt} and exit;
        INIT(G, r);
  {Update t and T};
  if $(\theta > 0)$ **then**
    $T \leftarrow T - \theta$;
    for each $\theta \in V$ **do**
      $t(v) \leftarrow t(v) + \theta \cdot \Delta(v)$;
      if $T < T^{opt}$ **then**
        Update T^{opt} and r^{opt} with T and r;