Race-Condition-Aware Clock Skew Scheduling

Shih-Hsu Huang
Department of Electronic Engineering, Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.
shhuang@cycu.edu.tw

Yow-Tyng Nieh
Department of Electronic Engineering, Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.
ytnieh@vlsi.el.cycu.edu.tw

Feng-Pin Lu
Department of Electronic Engineering, Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.
lfb@vlsi.el.cycu.edu.tw

ABSTRACT

The race conditions often limit the smallest feasible clock period that the optimal clock skew scheduling can achieve. Therefore, the combination of clock skew scheduling and delay insertion (for resolving the race conditions) may lead to further clock period reduction. However, the interactions between clock skew scheduling and delay insertion have not been well studied. In this paper, we provide a fresh viewpoint to look at this problem. A novel approach, called race-condition-aware (RCA) clock skew scheduling, is proposed to determine the clock skew schedule by taking the race conditions into account. Our objective is not only to optimize the clock period, but also to heuristically minimize the required inserted delay. Compared with previous work, our approach has significant improvement in the time complexity.

Categories and Subject Descriptors

B.6.3 [**Logic Design**]: Design Aids – *Optimization*.

General Terms

Algorithms, Performance.

Keywords

High performance, Sequential circuits, Timing optimization.

1. INTRODUCTION

Although the optimal clock skew scheduling [1,2] can enhance the circuit performance, it has not achieved the lower bound of sequential timing optimization [3]. The hold constraints often limit the smallest feasible clock period that the optimal clock skew scheduling can achieve. However, in fact, the race conditions can be resolved by applying the delay insertion, which is referred to as the *padding* method [4]. Therefore, the combination of clock skew scheduling and delay insertion can improve the optimal clock skew scheduling. Huang and Nieh [5] ever proposed an approach to this problem. Their objective is not only to optimize the clock period, but also to heuristically minimize the required inserted delay. However, the time complexity of their algorithm is $O(|E|^2*|T_{OCSS}|)$, where $|E|$ is the number of data paths and $|T_{OCSS}|$ is the time complexity of the used optimal clock skew scheduling algorithm. Clearly, their approach [5] is very time-consuming in the worst case.

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

In this paper, we provide a fresh viewpoint to look at the interactions between clock skew scheduling and delay insertion. An approach, called *race-condition-aware* (RCA) clock skew scheduling, is proposed to determine the clock skew schedule by taking the race conditions into account. Our approach achieves the same results as [5], but the time complexity of our approach is only $\mathbf{O(|E|*|T_{OCSS}|)}$. Compared with [5], our approach has significant improvement in the time complexity.

2. PRELIMINARIES

Let the notation $T_{s,i}$ (the notation $T_{h,i}$) denote the setup time (hold time) of register R_i, and let the notation $T_{PDi,j(max)}$ (the notation $T_{PDi,j(min)}$) denote the maximum delay (the minimum delay) of a data path $R_i \rightarrow R_j$. The constraints of clocking hazards in an edge-triggered circuit can be modeled as a *constraint graph* G(V,E), where each vertex $R_i \in V$ represents a register and each directed edge $e \in E$ represents a constraint. There are two types of directed edges: S-edges and H-edges. The setup constraint of data path $R_i \rightarrow R_j$ is modeled as an S-edge $e_s(R_j,R_i)$, which is from register R_j to register R_i and associated with a weight $P\text{-}T_{PDi,j(max)}\text{-}T_{s,j}$; and the hold constraint of data path $R_i \rightarrow R_j$ is modeled as an H-edge $e_h(R_i,R_j)$, which is from register R_i to register R_j and associated with a weight $T_{PDi,j(min)}\text{-}T_{h,j}$. The constraint graph that models all the constraints of clocking hazards in an edge-triggered circuit is referred to as the *optimal clock skew scheduling* (OCSS) problem, and we let G_{OCSS} denote this constraint graph.

We say that a constraint graph G works with clock period P = *c*, where *c* is a constant, if and only if there is a feasible clock skew schedule that satisfies all the clocking constraints represented by the constraint graph G when clock period P = *c*. From [2], we have the following lemmas.

Lemma 1: A constraint graph G works with clock period *c*, provided that it has no negative cycle when clock period P = *c*.

Lemma 2: There is at least one critical cycle in the constraint graph G when clock period $P = P_{MIN}(G)$, where $P_{MIN}(G)$ denotes the smallest feasible clock period of the constraint graph G.

3. MOTIVATIONAL EXAMPLES

3.1 The Limitation of OCSS

For an edge-triggered circuit ckt, the lower bound of sequential timing optimization [3], denoted as $P_B(ckt)$, is determined by the setup constraints. Due to the limitation of hold constraints, the optimal clock skew scheduling often does not achieve this lower bound. Let's use the edge-triggered circuit ex1 shown in Figure 1 as an example. This circuit has four registers and seven logic gates. Note that the delay of each logic gate is within a range. For example, the delay of logic gate B is within the range from 4 to 5. Assume that the setup time and the hold time of each register are

1. If this circuit is fully synchronous, the smallest feasible clock period is 12 since $T_{PDhost,3(max)}+T_{s,3} = 11+1 = 12$.

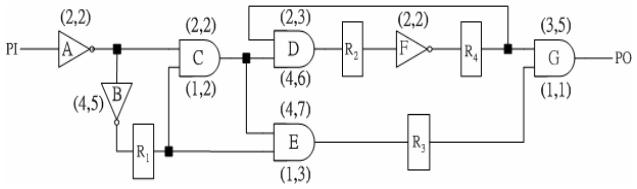

Figure 1. Edge-triggered circuit ex1.

Figure 2 gives the constraint graph $G_{OCSS}(ex1)$. As shown in Figure 3, we have $P_{MIN}(G_{OCSS}(ex1)) = 10$ under the clock skew schedule in which $T_{host} = 0$, $T_{C1} = 5$, $T_{C2} = 5$, $T_{C3} = 5$ and $T_{C4} = 4$. However, with an analysis to Figure 2, we find that the cycle consisting of S-edge $e_s(host,R_3)$ and S-edge $e_s(R_3,host)$ determines the lower bound of sequential timing optimization, and we have $P_B(ex1) = (11+T_{s,3}+1+T_{s,host})/2 = 7$. From Figure 3, we find that the critical cycle consists of S-edge $e_s(R_3, R_1)$ and H-edge $e_h(R_1,R_3)$. Thus, if we can relax the hold constraint in the data path $R_1 \rightarrow R_3$, the clock period can be further reduced.

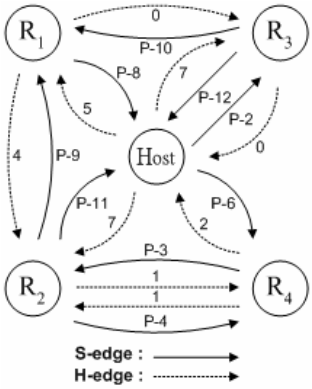

Figure 2. Constraint graph $G_{OCSS}(ex1)$.

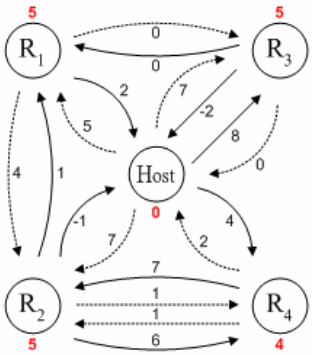

Figure 3. Constraint graph $G_{OCSS}(ex1)$ when P =10.

3.2 The Limitation of Padding

Intuitively, we can use the following two-stage process to achieve the lower bound of sequential timing optimization. In the first stage, we use only the setup constraints to determine the clock skew schedule. Then, in the second stage, all the race conditions (under the determined clock skew schedule) are resolved by applying the padding method [4]. However, since the first stage does not consider the hold constraints, the second stage often suffers from extra delay insertion.

Let's use the constraint graph $G_{OCSS}(ex2)$ shown in Figure 4 as an example. The cycle consisting of S-edge $e_s(host,R_1)$ and S-edge $e_s(R_1,host)$ determines the lower bound of sequential timing optimization, and we have $P_B(ex2) = (9+3)/2 = 6$. As shown in Figure 5, we find that there is no negative cycle in the constraint graph $G_{OCSS}(ex2)$ when clock period P = 6. Obviously, the lower bound of sequential timing optimization is achieved without any delay insertion.

On the other hand, Figure 6 gives a sketch of the intuitive two-stage process. In the first stage, since only setup constraints are taken into account, we may obtain a clock skew schedule in which $T_{host} = 0$, $T_{C1} = -3$, $T_{C2} = 3$ and $T_{C3} = 0$ to work with clock period P = 6. Figure 6 (a) gives the process of the first stage. In the second stage, the delay insertion is applied to resolve the race conditions under the clock skew schedule in which $T_{host} = 0$, $T_{C1} = -3$, $T_{C2} = 3$ and $T_{C3} = 0$. As shown in Figure 6 (b), the required increase of the minimum delay from *host* to R_2 is at least 3.

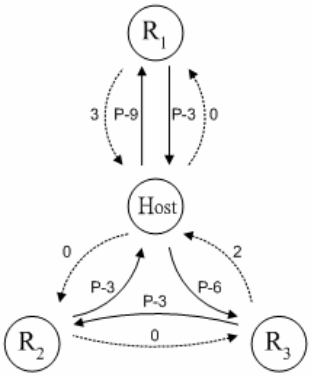

Figure 4. Constraint graph $G_{OCSS}(ex2)$.

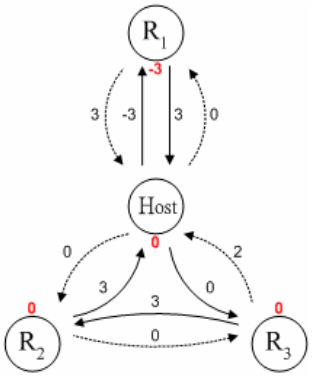

Figure 5. Constraint graph $G_{OCSS}(ex2)$ when P = 6.

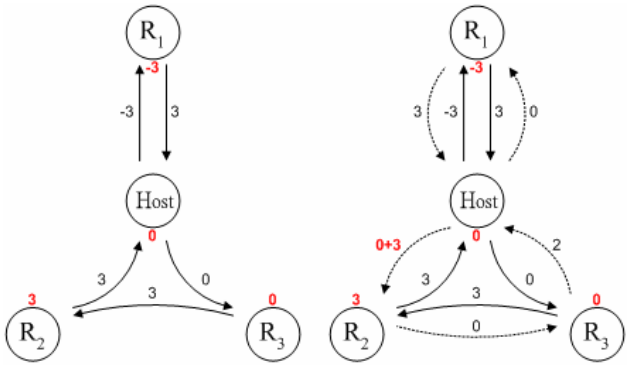

Figure 6. The intuitive two-stage process.

4. PROBLEM FORMULATION

To improve the optimal clock skew scheduling, we should have a mechanism to model the relaxation of hold constraints in a constraint graph. We define a *relaxed constraint graph* as below.

Definition 1: A constraint graph G'(V',E') is a *relaxed constraint graph* of constraint graph G(V,E), if and only if G'(V',E') satisfies the following properties:

(1) V' = V;
(2) E' = E;
(3) Suppose that an S-edge $e_s(R_j,R_i)$ in G is associated with a weight $P-T_{PDi,j(max)}-T_{s,j}$, where P is the clock period. Then, the corresponding S-edge $e_s(R_j,R_i)$ in G' is also associated with the weight $P-T_{PDi,j(max)}-T_{s,j}$.
(4) Suppose that an H-edge $e_h(R_i,R_j)$ in G is associated with a weight $T_{PDi,j(min)}-T_{h,j}$. Then, the corresponding H-edge $e_h(R_i,R_j)$ in G' is associated with a weight $T_{PDi,j(min)}-T_{h,j}+p_{i,j}$, where $p_{i,j}$ is called the *relaxed parameter* and $p_{i,j} \geq 0$. An H-edge $e_h(R_i,R_j)$ is a *relaxed H-edge* if and only if $p_{i,j} > 0$.

To minimize the required inserted delay (for resolving the race conditions), we should not over-relax the hold constraints. We define an *irredundant-relaxed constraint graph* as below.

Definition 2: Let G' be a relaxed constraint graph of constraint graph G, and for each H-edge $e_h(R_i,R_j)$, the relaxed parameter is $a_{i,j}$. We say that G' is the *irredundant-relaxed constraint graph* of G for clock period $P_{MIN}(G')$, if and only if there does not exist another relaxed constraint graph G" of G (i.e., G"≠G'), which satisfies the following two properties:

(1) G" also works with clock period $P_{MIN}(G')$.
(2) G" is also a relaxed constraint graph of constraint graph G, and for each H-edge $e_h(R_i,R_j)$, the relaxed parameter is $b_{i,j}$, where $b_{i,j} \le a_{i,j}$.

Using the constraint graph G_{OCSS}(ex2) shown in Figure 4 as an example, both the constraint graph shown in Figure 5 and the constraint graph shown in Figure 6 (b) are the relaxed constraint graph of G_{OCSS}(ex2) to work with clock period P = 6. However, only the constraint graph shown in Figure 5 is the irredundant-relaxed constraint graph of G_{OCSS}(ex2) to work with clock period P = 6.

We formally formulate our problem as below. Given an edge-triggered circuit ckt, our objective is to construct the irredundant-relaxed constraint graph of G_{OCSS}(ckt), denoted as G_{RCA}(ckt), for the lower bound of sequential timing optimization P_B(ckt).

5. THE APPROACH

5.1 The Algorithm

Figure 7 gives the pseudo code of our RCA clock skew scheduling. The input is the constraint graph G_{OCSS}. The outputs include the irredundant-relaxed constraint graph G_{RCA}, the clock period P_{RCA}, and the corresponding schedule S_{RCA}. Note that our algorithm needs *not* to know the lower bound of sequential timing optimization in advance. Our algorithm automatically achieves this lower bound, i.e., $P_{RCA} = P_B$.

The procedure *Relax_Hold* is an iteration process of clock period reduction by deleting the critical hold constraints that actually limit the circuit performance. Figure 8 gives the pseudo code. We let $G_{D(0)} = G_{OCSS}$, $G_{D(k)}$ denote the constraint graph in the *k*th loop iteration, $P_{D(k)}$ denote the clock period $P_{MIN}(G_{D(k)})$, and $S_{D(k)}$ denote a feasible schedule of $G_{D(k)}$ to work with clock period $P_{D(k)}$. A critical H-edge is called an *actual critical H-edge*, if and only if there is a critical cycle containing this critical H-edge. Clearly, actual critical H-edges in $G_{D(k)}$ limit the smallest feasible clock period $P_{D(k)}$. Thus, we construct a new constraint graph $G_{D(k+1)}$ by deleting all the actual critical H-edges in the constraint graph $G_{D(k)}$ with respect to $S_{D(k)}$. The iteration process repeats until the smallest feasible clock period cannot be further reduced. It is easy to prove that P_{RCA} always achieves the lower bound of sequential timing optimization P_B.

The procedure *Parameter_Assign* is to construct the relaxed constraint graph G_{INS} of G_{OCSS} with respect to clock period P_{RCA}. Let the notation E_{DEL} denote the set of deleted H-edges during the procedure *Relax_Hold*. In the constraint graph G_{INS}, we let the relaxed parameter $p_{i,j}$ of each H-edge $e_h(R_i,R_j) \in E_{DEL}$ be $\max(T_{Cj}-T_{Ci}-T_{PDi,j(min)}+T_{h,j},0)$ with respect to the clock skew schedule S_{DEL}. Clearly, the relaxed constraint graph G_{INS} works with clock period P_{RCA} under the schedule S_{DEL}. Thus, $P_{MIN}(G_{INS}) = P_{RCA}$.

The procedure *Parameter_Minimization* is to construct the irredundant constraint graph G_{RCA} of G_{OCSS} with respect to clock period P_{RCA}. Note that, for the same H-edge $e_h(R_i,R_j)$, the value of relaxed parameter $p_{i,j} = s_{i,j}$ in G_{INS} gives an upper bound of relaxed parameter in G_{RCA}. We examine all the relaxed H-edges in arbitrary sequence. When a relaxed H-edge $e_h(R_i,R_j)$ is examined, we let the relaxed parameter $p_{i,j}$ be a variable. We apply the binary search strategy[1] between the range $[0,s_{i,j}]$ to find the minimum value $p_{i,j} = a_{i,j}$ to work with clock period P_{RCA}. Then, we let the relaxed parameter $p_{i,j} = a_{i,j}$ in the constraint graph G_{RCA}.

```
Procedure RCA_Clock_Skew_Scheduling(G_OCSS)
 begin
 (G_DEL,P_RCA,S_DEL) = Relax_Hold(G_OCSS);
 (G_INS) = Parameter_Assign(G_DEL,S_DEL);
 (G_RCA,S_RCA) = Parameter_Minimization(G_INS,P_RCA);
 return(G_RCA,P_RCA,S_RCA);
 end.
```

Figure 7. Our approach.

```
Procedure Relax_Hold(G_OCSS)
begin
k = 0; G_D(k) = G_OCSS;
derive S_D(k) and P_D(k) with respect to G_D(k);
repeat
 obtain G_D(k+1) by deleting all the actual critical H-edges in G_D(k)
with respect to S_D(k);
 derive S_D(k+1) and P_D(k+1) with respect to G_D(k+1); k = k+1;
until (P_D(k) == P_D(k-1));
G_DEL = G_D(k-1); P_RCA = P_D(k-1); S_DEL = S_D(k-1);
return(G_DEL,P_RCA,S_DEL);
end.
```

Figure 8. Procedure *Relax_Hold*.

The time complexity of procedure *Relax_Hold* is $O(|E|*|T_{OCSS}|)$, the time complexity of procedure *Parameter_Assign* is $O(|E|)$, and the time complexity of procedure *Parameter_Minimization* is $O(|E|*|T_{OCSS}|)$.

5.2 An Example

We use the circuit ex1 shown in Figure 1 to demonstrate our approach. Our input is G_{OCSS}(ex1) as shown in Figure 2.

First, we start with the procedure *Relax_Hold*. Initially, $G_{D(0)} = G_{OCSS}$. We have $P_{D(0)} = 10$. Figure 3 gives the constraint graph $G_{D(0)}$ when clock period P = 10. With an analysis to Figure 3, we find that there is a critical cycle consisting of S-edge $e_s(R_3,R_1)$ and H-edge $e_h(R_1,R_3)$. By deleting the actual critical H-edge $e_h(R_1,R_3)$, we have the constraint graph $G_{D(1)}$. We find that $P_{D(1)} = 8$ under the clock skew schedule $S_{D(1)}$ in which $T_{host} = 0$, $T_{C1} = 2$, $T_{C2} = 3$, $T_{C3} = 6$ and $T_{C4} = 2$. We find that there is a critical cycle consisting of S-edge $e_s(R_2,R_4)$ and H-edge $e_h(R_4,R_2)$. By deleting the actual critical H-edge $e_h(R_4,R_2)$, we have the constraint graph $G_{D(2)}$. We find that $P_{D(2)} = 7$ under the clock skew schedule $S_{D(2)}$ in which $T_{host} = 0$, $T_{C1} = 2$, $T_{C2} = 5$, $T_{C3} = 5$ and $T_{C4} = 1$. Figure 9 gives the constraint graph $G_{D(2)}$ when clock period P = 7. We find that there is no actual critical H-edge in Figure 9. Thus, the smallest feasible clock period cannot be further reduced. We have $G_{DEL} = G_{D(2)}$, $S_{DEL} = S_{D(2)}$ and $P_{RCA} = 7$.

Next, we move to the procedure *Parameter_Assign*. The set E_{DEL} has two H-edges: $e_h(R_1,R_3)$ and $e_h(R_4,R_2)$. We use the clock skew schedule S_{DEL} to derive the relaxed parameters of the two H-edges

[1] Following [5], we use the binary search strategy to minimize the relaxed parameter $p_{i,j}$. In fact, the problem formulation is the same as the optimal clock skew scheduling, except that this problem is to minimize the relaxed parameter $p_{i,j}$, whereas the optimal clock skew scheduling is to minimize the clock period.

in G_{INS}. For the H-edge $e_h(R_1,R_3)$, we let the relaxed parameter $p_{1,3}$ be $\max(T_{C3}-T_{C1}-T_{PD1,3(min)}+T_{h,3},0) = \max(5-2-1+1,0) = 3$. For the H-edge $e_h(R_4,R_2)$, we let the relaxed parameter $p_{4,2}$ be $\max(T_{C2}-T_{C4}-T_{PD4,2(min)}+T_{h,2},0) = \max(5-1-2+1,0) = 3$.

Finally, we move to the procedure *Parameter_Minimization*. We let $G_{RCA} = G_{INS}$. Then, we let the relaxed parameter $p_{1,3}$ in G_{RCA} be a variable. By applying binary search between the range [0,3], we find that the minimum value of $p_{1,3}$ is 3. Thus, we let $p_{1,3} = 3$ in G_{RCA}. Next, we let the relaxed parameter $p_{4,2}$ in G_{RCA} be a variable. By applying binary search between the range [0,3], we find that the minimum value of $p_{4,2}$ is 2. Thus, we let $p_{4,2} = 2$. Consequently, we have the constraint graph G_{RCA} as shown in Figure 10. Based on G_{RCA}, we can derive the clock skew schedule S_{RCA} in which $T_{host} = 0$, $T_{C1} = 2$, $T_{C2} = 4$, $T_{C3} = 5$ and $T_{C4} = 1$.

We can apply the padding method [4] to resolve all the race conditions under the clock skew schedule S_{RCA}. Figure 11 gives the padded circuit ex1'. Delay element H and delay element I are inserted to resolve the race conditions. The padded circuit ex1' works with clock period $P_{RCA} = P_B = 7$ under the clock skew schedule S_{RCA}.

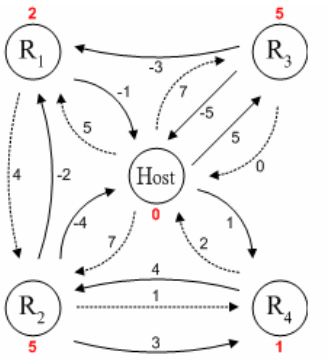

Figure 9. Constraint graph G_{DEL}(ex1) when P = 7.

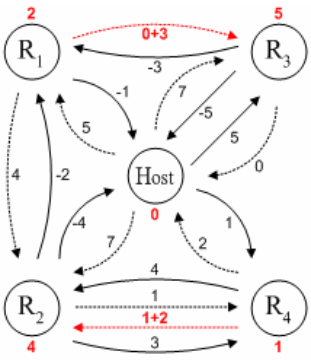

Figure 10. Constraint graph G_{RCA}(ex1) when P = 7.

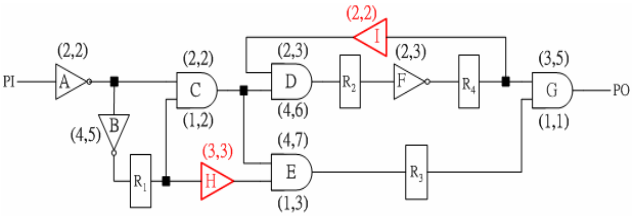

Figure 11. Padded circuit ex1'.

6. EXPERIMENTAL RESULTS

For the purpose of comparisons, we use C++ programming language to implement both our RCA clock skew scheduling and the approach proposed by [5] on a Sun Ultra-Sparc5 workstation. The ISCAS'89 benchmark circuits are targeted to a 0.18 μm cell library for experiments.

Table 1 tabulates the characteristics of benchmark circuits. The column *Regs* denotes the total number of registers. The column *Paths* denotes the total number of data paths. The column *Longest* denotes the longest path delay. The column *OCSS* denotes the smallest feasible clock period obtained by the optimal clock skew scheduling. The column P_B denotes the lower bound of sequential timing optimization. Our RCA clock skew scheduling guarantees achieving this lower bound.

Table 1. Characteristics of benchmark circuits.

Circuit	Regs	Paths	Longest	OCSS	P_B
s3271	116	1688	3.38	2.02	1.57
s3384	183	1937	6.53	5.18	3.27
s4863	104	1144	11.46	10.30	6.10
s6669	239	2481	17.39	14.79	4.99
s9234	228	3256	7.38	4.88	4.59
s13207	669	4660	8.80	6.59	5.37
s15850	597	16863	11.02	8.43	5.93
s35932	1728	5113	3.99	3.82	3.36

Note that, in each benchmark circuit, both RCA clock skew scheduling and [5] achieve the same results. Therefore, Table 2 only compares their time complexities. The column *# of Binary Searches* gives the comparisons on the number of binary searches. The column *CPU Time* gives the comparisons on the CPU time.

Table 2. Comparison of our approach with [5].

Circuit	# of Binary Searches		CPU Time (seconds)	
	[5]	**RCA**	[5]	**RCA**
s3271	96	15	94	21
s3384	3419	210	2773	202
s4863	4035	259	3914	242
s6669	6608	195	6417	285
s9234	60	12	101	21
s13207	207	11	864	70
s15850	403	17	1269	87
s35932	125	9	486	61

7. CONCLUSIONS

This paper proposes an irredundant-relaxed constraint graph to model the interaction between clock skew scheduling and delay insertion. Compared with previous work, our RCA clock skew scheduling has significant improvement in the time complexity. Experimental data consistently show that our RCA clock skew scheduling can solve large circuits within only few minutes.

8. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Council of R.O.C. under grant number NSC 93-2220-E-033-001.

9. REFERENCES

[1] J.P. Fishburn, "Clock Skew Optimization", IEEE Trans. on Computers, Vol. 39, No. 7, pp. 945—951, 1990.

[2] R.B. Deokar and S.S. Sapatnekar, "A Graph-Theoretic Approach to Clock Skew Optimization", Proc. of IEEE International Symposium on Circuits and Systems, Vol. 1, pp.407—410, 1994.

[3] M.C. Papaefthymiou, "Understanding Retiming through Maximum Average-Delay Cycles", Mathematical Systems Theory, Vol. 27, pp. 65—84, 1994.

[4] N.V. Shenoy, R.K. Brayton and A.L. Sangiovanni-Vincentelli, "Minimum Padding to Satisfy Short Path Constraints", Proc. of IEEE/ACM International Conference on Computer Aided Design, pp. 156—161, 1993.

[5] S.H. Huang and Y.T. Nieh, "Clock Period Minimization of Non-Zero Clock Skew Circuits", Proc. of IEEE/ACM International Conference on Computer Aided Design, pp. 809—812, 2003.