Clock Period Minimization of Non-Zero Clock Skew Circuits

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

ABSTRACT

It is known that the clock skew can be exploited as a manageable resource to improve the circuit performance. However, due to the limitation of race condition, the optimal clock skew scheduling does not achieve the lower bound of the clock period. In this paper, we propose a polynomial time complexity algorithm, which incorporates optimal clock skew scheduling and delay insertion, for the synthesis of non-zero clock skew circuits. The main advantages of our algorithm include two parts. First, it guarantees to achieve the lower bound of the clock period. Secondly, it also tries to minimize the required inserted delays under the lower bound of the clock period. Experimental data shows that, even though we only use the buffers in a standard cell library to implement the delay insertion, our approach still works well.

1. INTRODUCTION

Although the optimal clock skew scheduling [1,2] can enhance the circuit performance, it does not achieve the lower bound of the clock period. In fact, the maximum delay-to-register ratio of the cycles in the circuit gives a lower bound of the clock period. No sequential timing optimization technique can derive a sequential circuit whose clock period works less than this lower bound. Previous works [3,4] have tried to improve the optimal clock skew scheduling by combining with retiming transformation. However, retiming transformation is likely to have the following effects in the design flow: (1) an extra circuitry may be needed for initial states; (2) the forward retiming may increase the number of registers; and (3) verification issues.

This paper investigates the clock period minimization problem from a different standpoint. The main distinction of our approach is that it uses delay insertion instead of retiming transformation. The proposed algorithm guarantees to obtain a non-zero clock skew circuit, which works with the lower bound of the clock period, in polynomial time complexity. Specifically, since our algorithm also tries to minimize the required inserted delays, it makes the permissible range of delay insertion quite large. As a result, even though we only use the buffers in a standard cell library to realize the delay insertion (i.e., the amount of inserted delay is not continuous), a feasible solution is easily found. Thus, the proposed approach can work well in practice.

2. PRELIMINARIES

A sequential circuit can be modeled as a circuit graph G(V,E), where V is the set of vertices and E is the set of directed edges. Each vertex $R_i \in V$ represents a register. A special vertex called the *host* is introduced for the synchronization with primary inputs and primary outputs. A directed edge (R_i,R_j) represents a data path from register R_i to register R_j. Each directed edge (R_i,R_j) is associated with a weight $(T_{PDi,j(min)}, T_{PDi,j(max)})$, where $T_{PDi,j(min)}$ and $T_{PDi,j(max)}$ are the minimum delay and the maximum delay, respectively. Let's use the sequential circuit *ex* given in Figure 1 as an example. This circuit has four registers and eleven logic gates. The corresponding circuit graph G_{ex} is given in Figure 2.

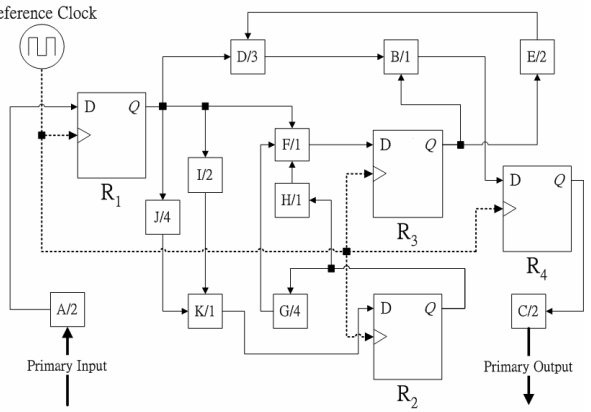

Figure 1: A sequential circuit *ex*.

Given a circuit graph G and clock period P, we can model the constraints of clocking hazards by a constraint graph $G_{cg}(G,P)$. The constraint graph $G_{cg}(G,P)$ is a directed graph, where a vertex corresponds to a register and a directed edge corresponds to a constraint. For each directed edge (R_i,R_j) in the circuit graph G, it is replaced by a D-edge $e_d(R_i,R_j)$ and a Z-edge $e_z(R_j,R_i)$ in the corresponding constraint graph $G_{cg}(G,P)$. The D-edge $e_d(R_i,R_j)$, which is associated with a weight $-T_{PDi,j(min)}$, corresponds to the double clocking constraint $T_{Ci}-T_{Cj} \geq -T_{PDi,j(min)}$. The Z-edge $e_z(R_j,R_i)$, which is associated with a weight $T_{PDi,j(max)}-P$, corresponds to the zero clocking constraint $T_{Cj}-T_{Ci} \geq T_{PDi,j(max)}-P$.

The clock period P is feasible, if and only if the corresponding constraint graph contains no positive cycle [2]. Based on this property, [2] proposed a binary search strategy to solve the optimal clock skew scheduling problem in polynomial time complexity. The lower bound and upper bound of the clock period can be specified as 0 and longest path delay, respectively. The binary search procedure continues until the smallest feasible clock period is found. If the clock period P is feasible, then a set of clock arrival times can be found by solving the longest path problem on the corresponding constraint graph. In the following, we use the notation T_{Ci} to denote the clock arrival time to register R_i. Note that T_{Ci} may be a negative value (relative to a global time reference).

Let's use the sequential circuit *ex* for illustration. The longest path delay is 6 tu (time units). The constraint graph $G_{cg}(G_{ex},P)$ is shown in Figure 3. For the convenience of presentation, we label T_{Ci} for each vertex R_i in the constraint graph and an optimal clock skew schedule is denoted as $(T_{host},T_{C1},T_{C2},T_{C3},T_{C4})$. After optimal clock skew scheduling is applied, we have (0,0,0,0,1). Compared with fully synchronous system, the minimum clock period is reduced from 6 tu to 5 tu.

ICCAD'03, November 11-13, 2003, San Jose, California, USA.

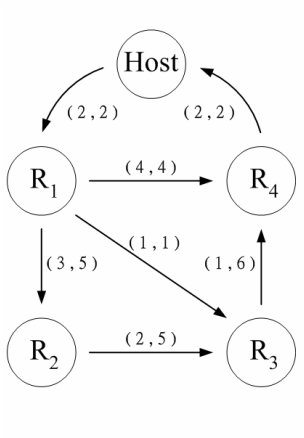

Figure 2: The circuit graph G_{ex}.

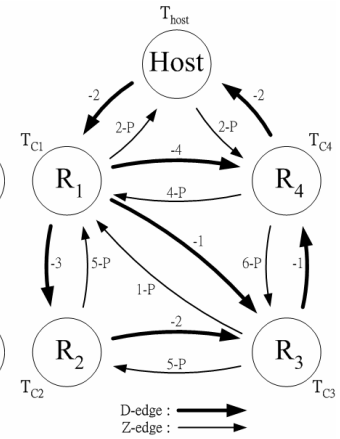

Figure 3: The constraint graph $G_{cg}(G_{ex},P)$.

A cycle is defined as a critical cycle if and only if the summation of weights is zero. Given a circuit graph G, there exists at least one critical cycle in the constraint graph $G_{cg}(G,P_{OCSS})$, where P_{OCSS} is the minimum clock period obtained by optimal clock skew scheduling (OCSS). For example, in Figure 3, $R_3 \rightarrow R_4 \rightarrow R_3$ is a critical cycle when P = 5. In a constraint graph, an edge from register R_i to register R_j is defined as a critical edge if and only if T_{Ci}-T_{Cj} is equal to the weight of this edge. Note that all the directed edges in a critical cycle are critical edges.

3. THE MOTIVATION

The delay-to-register ratio of a directed cycle C in a circuit graph G is defined as (the maximum delay of C) / (number of registers in C). Clearly, the maximum delay-to-register ratio of directed cycles in the circuit graph gives a lower bound of the clock period. Let $P_B(G)$ denotes this lower bound. No sequential circuit can work with clock period less than $P_B(G)$, even if retiming techniques are applied, unless delays of elements are decreased or the number of registers in a cycle is increased.

Let's consider the circuit graph G_{ex}. The lower bound $P_B(G_{ex})$ is 4 tu. However, the P_{OCSS} is 5 tu. We find that the optimal clock skew scheduling is limited by the critical cycle $R_3 \rightarrow R_4 \rightarrow R_3$. If we increase $T_{PD3,4(min)}$, the cycle $R_3 \rightarrow R_4 \rightarrow R_3$ becomes non-critical.

Definition 1: A circuit graph G'(V',E') is a delay-inserted circuit graph of a circuit graph G(V,E), if and only if G'(V',E') satisfies the following three properties: (1) V'=V. (2) E'=E. (3) Suppose that $T_{PDi,j(max)} = D_{i,j}$ and $T_{PDi,j(min)} = d_{i,j}$ for each edge $(R_i,R_j) \in E$. Then, for the corresponding edge $(R_i,R_j) \in E'$, we have that $T_{PDi,j(max)} = \max(D_{i,j}, d_{i,j}+p_{i,j})$ and $T_{PDi,j(min)} = d_{i,j}+p_{i,j}$, where $p_{i,j} \geq 0$. We say that $p_{i,j}$ is the increase of the minimum delay from register R_i to register R_j.

Definition 2: A directed cycle in the constraint graph is a Z-cycle, if and only if all the directed edges in this cycle are Z-edges.

Theorem 1: Suppose that G is a circuit graph, whose minimum clock period is P. If no critical cycle in the constraint graph $G_{cg}(G,P)$ is Z-cycle, there exists a delay-inserted circuit graph G', whose minimum clock period P'<P.

Proof: We prove the theorem by providing a method to construct the delay-inserted circuit graph G'. Suppose that $T_{PDi,j(max)} = D_{i,j}$ and $T_{PDi,j(min)} = d_{i,j}$ in the circuit graph G. In the delay-inserted graph G', the increase of the minimum delay from register R_i to register R_j is denoted as $p_{i,j}$. We let $p_{i,j} = 0$ except for the following considerations.

It is clear that $G_{cg}(G,P)$ contains no positive cycle. Each critical cycle C in the constraint graph $G_{cg}(G,P)$ has at least one critical D-edge $e_d(R_i,R_j)$. An increase of the minimum delay $T_{PDi,j(min)}$ can make the cycle C become non-critical. To increase $T_{PDi,j(min)}$, we consider the following two conditions.

(1) If $d_{i,j}<D_{i,j}$, we let $0<p_{i,j}\leq D_{i,j}-d_{i,j}$. The weight of D-edge $e_d(R_i,R_j)$ in $G_{cg}(G',P)$ is $-(d_{i,j}+p_{i,j})$, which is less than that in $G_{cg}(G,P)$ by $p_{i,j}$. As a result, the cycle C in the constraint graph $G_{cg}(G',P)$ becomes non-critical.

(2) If $d_{i,j}=D_{i,j}$, we let $0<p_{i,j}<P$. The weight of D-edge $e_d(R_i,R_j)$ in $G_{cg}(G',P)$ is $-(d_{i,j}+p_{i,j})$, which is less than that in $G_{cg}(G,P)$ by $p_{i,j}$. As a result, the cycle C in the constraint graph $G_{cg}(G',P)$ becomes non-critical. On the other hand, the weight of Z-edge $e_z(R_j,R_i)$ in $G_{cg}(G',P)$ is $d_{i,j}+p_{i,j}-P$, which is larger than that in $G_{cg}(G,P)$ by $p_{i,j}$. Since $e_d(R_i,R_j)$ is a critical D-edge in $G_{cg}(G,P)$, we have that $T_{Ci}-T_{Cj}=-D_{i,j}=-d_{i,j}$. The Z-edge $e_z(R_j,R_i)$ in $G_{cg}(G',P)$ corresponds to $T_{Cj}-T_{Ci} \geq d_{i,j}+p_{i,j}-P$. Since $0<p_{i,j}<P$, in $G_{cg}(G',P)$, the cycles containing the Z-edge $e_z(R_j,R_i)$ are non-critical .**Q.E.D.**

Lemma 1: Let G be the circuit graph of a sequential circuit *ckt*. Let G' be a delay-inserted circuit graph of G. Assume that G' works with clock period P'. We can implement a sequential circuit works with clock period P' by applying the padding method [5] to the sequential circuit *ckt*. Note that a feasible padding can be found in polynomial time complexity.

4. THE APPROACH

Given a circuit graph G_{in}, our optimization goal is to derive a delay-inserted graph G_{opt}, whose minimum clock period P_{opt} can achieve $P_B(G_{in})$. In addition to optimize the clock period, the proposed algorithm also tries to minimize the required inserted delays under the clock period $P_B(G_{in})$.

Figure 4 gives the pseudo code of the proposed algorithm, where G_{in} is the initial circuit graph. The subroutine *OCSS* denotes the optimal clock skew scheduling. In the *k*th iteration, the notation $G_{INS(k)}$ denotes the circuit graph obtained by delay insertion, the notations $S_{INS(k)}$ and $P_{INS(k)}$ denote the optimal clock skew schedule and minimum clock period with respect to the $G_{INS(k)}$, and $G_{MIN(k+1)}$ and $S_{MIN(k+1)}$ denote the circuit graph and optimal clock skew schedule obtained by delay minimization. Suppose that $T_{PDi,j(max)} = D_{i,j}$ and $T_{PDi,j(min)} = d_{i,j}$ in the circuit graph G_{in}. Since the circuit graphs $G_{INS(k)}$ and $G_{MIN(k+1)}$ are delay-inserted graphs of G_{in}, in the following, they are represented in the form as $T_{PDi,j(max)} = \max(D_{i,j}, d_{i,j}+p_{i,j})$ and $T_{PDi,j(min)} = d_{i,j}+p_{i,j}$, where $p_{i,j}$ is the increase of the minimum delay and $p_{i,j} \geq 0$.

```
Procedure Our_Approach(G_in)
{ k = 1; G_MIN(k) = G_in;
(S_MIN(k),P_MIN(k)) = OCSS(G_MIN(k));
repeat
G_INS(k)= Delay_Insertion(G_MIN(k),S_MIN(k),P_MIN(k));
(S_INS(k),P_INS(k)) = OCSS(G_INS(k));
(G_MIN(k+1),S_MIN(k+1),P_MIN(k+1))= Delay_Minimization(G_INS(k),P_INS(k));
k = k+1;
until (P_MIN(k-1)==P_INS(k-1));
G_opt = G_MIN(k-1); S_opt = S_MIN(k-1); P_opt = P_MIN(k-1);
return(G_opt,S_opt,P_opt); }
```

Figure 4: The proposed algorithm.

The subroutine *Delay_Insertion* is to obtain a circuit graph $G_{INS(k)}$ by inserting delays to the data paths, whose minimum delays are critical with respect to $S_{MIN(k)}$, in $G_{MIN(k)}$. Figure 5 gives the pseudo code. We examine every data path in arbitrary sequence. For every critical minimum delay $T_{PDi,j(min)}$, we consider the following two conditions. If $T_{PDi,j(min)}<T_{PDi,j(max)}$, then we let the increased delay $p_{i,j}$ be $D_{i,j}-d_{i,j}$; if $T_{PDi,j(min)}=T_{PDi,j(max)}$, then we let the increased delay $p_{i,j}$ be $p_{i,j}+(P_{MIN(k)}/2)$. After every data path is examined, we obtain a circuit graph $G_{INS(k)}$, which is a delay-inserted graph of $G_{MIN(k)}$. All the D-edges in the constraint graph $G_{cg}(G_{INS(k)},P_{MIN(k)})$ are non-critical. Thus, applying OCSS to $G_{INS(k)}$ may lead to further clock period minimization.

Procedure Delay_Insertion($G_{MIN(k)}$,$S_{MIN(k)}$,$P_{MIN(k)}$)
{$G = G_{MIN(k)}$;
for each data path (R_i,R_j) in $G_{MIN(k)}$ **do**
if $T_{PDi,j(min)}$ is critical to the schedule $S_{MIN(k)}$ **then**
if ($T_{PDi,j(min)}<T_{PDi,j(max)}$) **then**
let the increased delay $p_{i,j} = (D_{i,j}-d_{i,j})$ in G;
else let the increased delay $p_{i,j} = (p_{i,j}+(P_{MIN(k)}/2))$ in G;
$G_{INS(k)} = G$;
return($G_{INS(k)}$);**}**

Figure 5: The procedure *Delay_Insertion*.

Lemma 2: If there is no critical Z-cycle in the constraint graph $G_{cg}(G_{MIN(k)},P_{MIN(k)})$, then $P_{INS(k)}<P_{MIN(k)}$.

The subroutine *Delay_Minimization* is to obtain a circuit graph $G_{MIN(k+1)}$ from the circuit graph $G_{INS(k)}$ by minimizing the increased delay $p_{i,j}$ of the minimum delay from register R_i to register R_j with respect to the clock period $P_{INS(k)}$. Figure 6 gives the pseudo code. We examine every delay-inserted data path in arbitrary sequence. When a delay-inserted data path is examined, we let $p_{i,j}$ become a variable. Suppose that $p_{i,j} = a_{i,j}$ in the circuit graph $G_{INS(k)}$. Then, the variable $p_{i,j}$ has the following property: if $p_{i,j}$ =a works with clock period $P_{INS(k)}$, $p_{i,j}$ = a' also works with clock period $P_{INS(k)}$, where $0 \le a \le a' \le a_{i,j}$; if $p_{i,j}$ =a does not work with clock period $P_{INS(k)}$, $p_{i,j}$ = a' also does not work with clock period $P_{INS(k)}$, where $0 \le a' \le a \le a_{i,j}$. Thus, we can use the binary search strategy to find the minimum value of variable $p_{i,j}$, where the interval of binary search is $[0,a_{i,j}]$. After every delay-inserted data path is examined, we obtain the circuit graph $G_{MIN(k+1)}$. The $S_{MIN(k+1)}$ can be found by solving the longest path problem on the constraint graph $G_{cg}(G_{MIN(k+1)},P_{INS(k)})$. Note that $G_{INS(k)}$ also works with clock period $P_{INS(k)}$ under $S_{MIN(k+1)}$.

Procedure Delay_Minimization($G_{INS(k)}$,$P_{INS(k)}$);
{ $G=G_{INS(k)}$;
for each delay-inserted data path in $G_{INS(k)}$ **do**
{ let $p_{i,j}$ be a variable in the constraint graph $G_{cg}(G,P_{INS(k)})$ and use the binary search strategy to find the minimum value of $p_{i,j}$; let $p_{i,j}$ in G be the minimum value obtained by the binary search; **}**
$G_{MIN(k+1)} = G$;
find $S_{MIN(k+1)}$ for the constraint graph $G_{cg}(G_{MIN(k+1)},P_{INS(k)})$;
$P_{MIN(k+1)} = P_{INS(k)}$;
return($G_{MIN(k+1)}$,$S_{MIN(k+1)}$, $P_{MIN(k+1)}$); **}**

Figure 6: The Procedure *Delay_Minimization*.

Theorem 2: After the procedure Delay_Minimization is completed, in the constraint graph $G_{cg}(G_{MIN(k+1)},P_{INS(k)})$, $e_d(R_i,R_j)$ is a critical D-edge if $p_{i,j} > 0$.

Proof: If D-edge $e_d(R_i,R_j)$ is not a critical D-edge, a smaller value of $p_{i,j}$ can be found by the binary search strategy. **Q.E.D.**

Theorem 3: After the procedure Delay_Minimization is completed, in the constraint graph $G_{cg}(G_{MIN(k+1)},P_{INS(k)})$, $e_z(R_j,R_i)$ is not a critical Z-edge if $p_{i,j} > D_{i,j}-d_{i,j}$.

Proof: From Theorem 2, we know that the D-edge $e_d(R_i,R_j)$ is a critical D-edge if $p_{i,j} > 0$. Therefore, we have $T_{Ci}-T_{Cj} = -T_{PDi,j(min)}$. If $e_z(R_j,R_i)$ is a critical Z-edge, then we have $T_{Cj}-T_{Ci} = T_{PDi,j(max)}-P$. However, if $p_{i,j} > D_{i,j}-d_{i,j}$, we should have $T_{PDi,j(max)}=T_{PDi,j(min)}$. Thus, $e_z(R_j,R_i)$ is not a critical Z-edge. **Q.E.D.**

Theorem 4: The clock period P_{opt} obtained by our approach achieves $P_B(G_{in})$.

Proof: The iteration in our approach terminates only when a critical Z-cycle is formed. Thus, $P_{opt} = P_B(G_{opt})$. Due to delay insertion, we have $P_B(G_{opt}) \ge P_B(G_{in})$. From Theorem 3, we know that $e_z(R_j,R_i)$ is not a critical Z-edge if $p_{i,j} > D_{i,j}-d_{i,j}$. Therefore, our approach still terminates with the Z-cycle that determines $P_B(G_{in})$. Consequently, we have $P_B(G_{opt}) = P_B(G_{in})$. **Q.E.D.**

Theorem 5: Our approach is in polynomial time complexity.

Proof: The $P_{INS(k)}$ can be smaller than $P_{MIN(k)}$, only when in $G_{INS(k)}$ there is at least one data path (R_i,R_j) whose $p_{i,j}$ value is larger than that in $G_{MIN(k)}$. The number of data paths is polynomial. The maximum number of iterations to increase the minimum delay of a data path is also polynomial. **Q.E.D.**

In the following, we use the sequential circuit *ex* to illustrate the proposed algorithm. Initially, we have $G_{MIN(1)}=G_{ex}$. By applying OCSS, we have $S_{MIN(1)}$=(0,0,0,0,1) and $P_{MIN(1)}$=5 tu. Then, our algorithm moves to the procedure *Delay_Insertion*. We find that $T_{PD3,4(min)}$ is critical in $G_{cg}(G_{MIN(1)},P_{MIN(1)})$. Since $T_{PD3,4(max)}$=6 tu and $T_{PD3,4(min)}$=1 tu, we have $p_{3,4}$=5 tu in $G_{INS(1)}$. By applying OCSS to $G_{INS(1)}$, we have $S_{INS(1)}$=(0,0,0.5,1,2.5) and $P_{INS(1)}$=4.5 tu. The constraint graph $G_{cg}(G_{INS(1)},P_{INS(1)})$ is given in Figure 7.

Next, our algorithm moves to the procedure *Delay_Minimization*. We have $p_{3,4}$=0.5 tu in $G_{MIN(2)}$ and $S_{MIN(2)}$=(0,0,0.5,1,2.5). Note that $P_{MIN(2)}=P_{INS(1)}$=4.5 tu. Then, we move to the procedure *Delay_Insertion*. We have $p_{3,4}$=5 tu and $p_{1,3}$= 0+($P_{MIN(2)}$/2)= 2.25 tu in $G_{INS(2)}$. Thus, $S_{INS(2)}$=(0,-2,-1,0,2) and $P_{INS(2)}$=4 tu.

Next, our algorithm moves to the procedure *Delay_Minimization*. We have $p_{3,4}$=1 tu and $p_{1,3}$=1 tu in $G_{MIN(3)}$, $S_{MIN(3)}$=(0,-2,-1,0,2), and $P_{MIN(3)}$=4 tu. The constraint graph $G_{cg}(G_{MIN(3)},P_{MIN(3)})$ is given in Figure 8. Then, we move to the procedure *Delay_Insertion*. We have $p_{3,4}$=5 tu and $p_{1,3} = 1+(P_{MIN(3)}/2) = 3$ tu in $G_{INS(3)}$. However, we cannot further reduce the clock period. Finally, we move to the procedure *Delay_Minimization*. We have $p_{3,4}$=1 tu and $p_{1,3}$=1 tu. Thus, $G_{opt} = G_{MIN(4)} = G_{MIN(3)}$, $S_{opt} = S_{MIN(4)} = S_{MIN(3)}$, and $P_{opt} = P_{MIN(4)} = P_{MIN(3)}$.

We apply the padding method to the sequential circuit *ex*. The padded sequential circuit *ex'* is given in Figure 9. Two delay elements M and N are added. The minimum delays $T_{PD3,4(min)}$ and $T_{PD1,3(min)}$ are increased from 1 tu to 2 tu. The maximum delay $T_{PD1,3(max)}$ is also increased from 1 tu to 2 tu. The sequential circuit *ex*' works with clock period P_{opt}=4 tu under S_{opt}=(0,-2,-1,0,2). Note that, due to delay minimization, the $p_{3,4}$ and $p_{1,3}$ are minimized. However, in fact, the clock skew schedule (0,-2,-1,0,2) works with clock period P_{opt} = 4 tu if $p_{3,4}$ is between 1 tu and 5 tu and $p_{1,3}$ is between 1 tu and 3 tu. Therefore, the permissible range of delay insertion is quite large. Due to this property, it is easy for us to find a feasible solution by using a cell library.

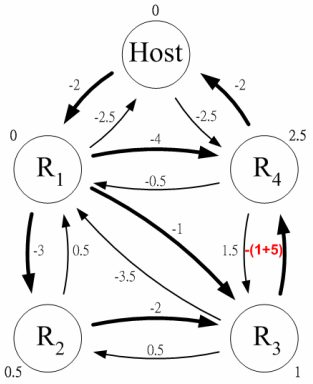

Figure 7: The constraint graph $G_{cg}(G_{INS(1)},4.5)$.

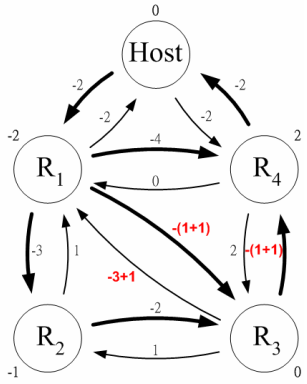

Figure 8: The constraint graph $G_{cg}(G_{MIN(3)},4)$.

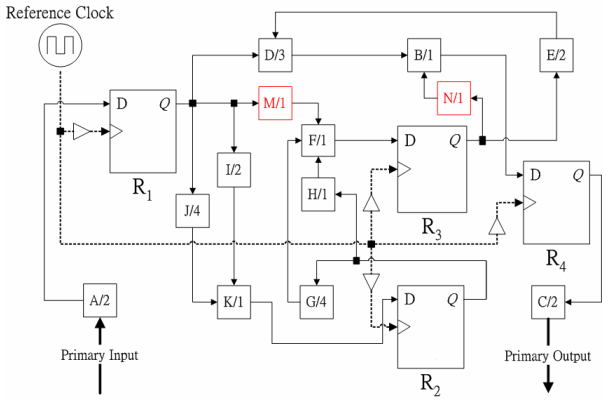

Figure 9: The padded sequential circuit *ex'*.

5. EXPERIMENTAL RESULTS

The ISCAS'89 benchmarks are targeted to a 0.35 μm cell library. Table 1 tabulates the characteristics of test circuits. The columns *#Cells*, *#Registers*, *#DataPaths*, *Longest*, and P_{OCSS} denote the number of cells, the number of registers, the number of data paths, the longest path delay, and the minimum clock period obatined by the OCSS, respectively.

The proposed algorithm has been implemented by using a C programming language on a Sun Ultra-5 workstation. Table 2 tabulates the experimental results of our algorithm, including the number of delay-inserted data paths and minimum clock period (i.e., P_{opt}). Compared with P_{ocss}, the result obtained by our algorithm (i.e., P_{opt}) has significant improvement. Note that, in every test circuit, our optimized result achieves the lower bound of the clock period and can be obtained within only few seconds.

We also realize the delay insertion by using the buffers in the cell library, including buffers *buffa*, *buff7*, *buff3*, and *buff1*. Table 3 tabulates the implementation results. The column *Delay Insertion* gives the numbers of four type buffers for delay insertion. The column P_{imp} is the minimum clock period obtained by the implementation. Since buffer insertion may affect other cells, e.g., wire capacitances and transition times of adjacent cells, P_{imp} may be slightly larger than P_{opt}.

6. CONCLUSIONS

This paper incorporates optimal clock skew scheduling and delay insertion for the synthesis of non-zero clock skew circuits. The proposed algorithm not only optimizes the clock period, but also tries to minimize the required inserted delays. Benchmark data consistently shows that our approach works well in practice.

7. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Council of R.O.C. under contract number NSC 91-2215-E-033-005.

Table 1: The characteristics of ISCAS'89 benchmarks.

Circuit	#Cells	#Registers	#Datapaths	Longest	P_{ocss}
S298	133	14	55	4.13	2.66
S344	175	15	106	6.61	5.37
S349	176	15	106	6.61	5.37
S444	202	21	160	5.02	2.88
S526	214	21	146	4.14	2.67
S526n	215	21	146	4.22	2.73
S1423	735	74	2155	19.93	19.45

Table 2: The results obtained by our algorithm.

Circuit	Number of delay-inserted data paths		**Minimum Clock Period (P_{opt})**
	$T_{PDi,j(min)}<T_{PDi,j(max)}$	$T_{PDi,j(min)}=T_{PDi,j(max)}$	
S298	19	1	**2.25**
S344	7	2	**4.68**
S349	8	2	**4.68**
S444	12	3	**2.55**
S526	18	2	**2.41**
S526n	11	2	**2.47**
S1423	9	1	**18.94**

Table 3: The implementation results by using cell library.

Circuit	Delay Insertion				**Minimum Clock Period (P_{imp})**
	#buffa	#buff7	#buff3	#buff1	
S298	13	0	3	9	**2.35**
S344	1	0	2	3	**4.68**
S349	1	0	2	3	**4.68**
S444	19	1	1	2	**2.56**
S526	3	0	1	9	**2.42**
S526n	5	0	0	8	**2.49**
S1423	1	2	0	1	**18.94**

8. 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", in the Proc. of IEEE International Symposium on Circuits and Systems, Vol. 1, pp.407—410, 1994.

[3] T. Soyata, E.G. Friedman and J.H. Mulligan, "Incorporating Interconnect, Register, and Clock Distribution Delays into the Retiming Process", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vo. 16, No. 1, pp. 105—120, 1997.

[4] X. Liu, M. C. Papaefthymiou and E. G. Friedman, "Retiming and Clock Scheduling for Digital Circuit Optimization", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, No. 2, pp. 184—203, 2002.

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