Reduction on the usage of intermediate registers for pipelined circuits

Bakhtiar Affendi
Comm. and Integrated Systems Department
Tokyo Institute Of Technology
Ookayama, Meguro-ku, Tokyo Japan
fendi@lab.ss.titech.ac.jp

Atsushi Takahashi
Comm. and Integrated Systems Department
Tokyo Institute Of Technology
Ookayama, Meguro-ku, Tokyo Japan
atsushi@lab.ss.titech.ac.jp

Abstract— Conventional pipelining consumed lots of areas due to intermediate registers need to be inserted between the stages. While wave-pipelining is a method of the circuit design which implements pipelining in logic without the use of intermediate latches or registers. However, to achieve the highest possible wave-pipelining frequency, delay balancing is required that will increase the area of a circuit. In this paper, we propose a new pipeline synthesis method that reduces the usage of intermediate registers by making use of the multi-clock cycle path and the semi-synchronous circuit technique. A multi-clock cycle path is introduced if an intermediate register can be removed without an excessive delay balancing. We will show the constraints that need to be satisfied by a multi-clock cycle path in the given clock period range. Also we will propose an algorithm to reduce the usage of intermediate registers for pipelined circuits.

I. Introduction

The sustained progress of VLSI technology leads to an increasing wire delay, shrinking clock period and growing chip size. In order to shrink clock period, circuit pipelining is one of the techniques that has been used. Pipelining is the method where a circuit is being divided into few stages and the intermediate registers are inserted between the stages to store the intermediate data. However, conventional pipelining methods consumed lots of areas due to intermediate registers need to be inserted between all the stages. It may also increase the latency due to the setup or hold time of the intermediate registers itself.

Recently, to overcome the above stated problem, several researches on the wave-pipelining have been carried out [6]. Wave pipelining is a method of speeding up the circuit without the insertion of intermediate registers. The wave pipeline enjoys several advantages over the conventional pipelining methods. It can achieve high clock period without partitioning the logic and add the intermediate registers thereby reducing quantizations overhead associated with the conventional pipeline as well as the clock buffers and routing which reduce the layout congestion. However, wave pipelining does require tighter timing constraints as there are no intermediate registers to store the intermediate data. In the wave pipelining, there exists a number of 'waves' of the data at a time inside a circuit. Therefore to avoid the data from collision with each other, delay balancing is required that will increase the area of a circuit.

In this paper, we discuss a new pipeline synthesis method which implements the multi-clock cycle path technique together with the semi-synchronous method that reduces the usage of intermediate registers compared to the conventional pipelining methods. Normally, in a synchronous circuit, there are some cases where the multi-clock cycle path technique will be used on certain path to achieve the target clock period without the insertion of the intermediate registers. We discuss the constraints that need to be satisfied in the semi-synchronous circuit method in order to implement the multi-clock cycle path technique.

The algorithm that can reduce the usage of intermediate registers for a 2-stage pipelined circuit while the circuit works correctly in the given clock period range will be proposed. Also, we will discuss the constraints that need to be satisfied in order to implement a 2-clock cycle path technique. Note that in general, a circuit will be designed to achieve the target clock period range. Our algorithm will remove all of the intermediate registers at first and then recover back the intermediate registers to make the circuit works correctly in the given clock period range. In this paper, we show by the experiments that with few intermediate registers a 2-stage pipelined circuit will work correctly in the given clock period range.

II. Preliminaries

We consider a circuit with a single clock consisting of registers and combinatorial circuits between them. The clock timing $s(u)$ of register u is defined as the difference in clock arrival time between u and an arbitrary chosen reference register. The set of clock timings is called a clock schedule.

We assume the framework that a circuit works correctly with clock period T if the following two types of constraints are satisfied for every register pair with signal propagation [2].

No-Zero-Clocking(Setup) Constraint

$$s(u) - s(v) \leq T - d_{\max}(u, v)$$

No-Double-Clocking(Hold) Constraint

$$s(v) - s(u) \leq d_{\min}(u, v)$$

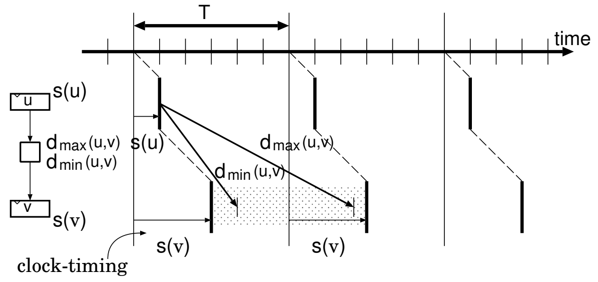

Fig. 1. Timing chart for a single-clock cycle path.

where $d_{\max}(u, v)$ ($d_{\min}(u, v)$) is the maximum (minimum) propagation delay from register u to register v along a combinatorial circuit. See Fig. 1. More precisely, $d_{\max}(u, v)$ is defined as

$$d^{'}_{\max}(u, v) + setup(v) + d_{\max}(v)$$

where $d^{'}_{\max}(u, v)$ is the maximum propagation delay from u to v, $setup(v)$ is the setup-time of register v, and $d_{\max}(v)$ is the maximum delay of register v itself. Similarly, $d_{\min}(u, v)$ is defined as

$$d^{'}_{\min}(u, v) - hold(v) + d_{\min}(v)$$

where $d^{'}_{\min}(u, v)$ is the minimum propagation delay from u to v, $hold(v)$ is the hold-time of register v, and $d_{\min}(v)$ is the minimum delay of register v itself.

In a conventional zero-skew based synchronous circuit, the maximum signal propagation delay between registers gives a lower bound for the clock period. Therefore, the multi-clock cycle path is often used in a circuit in order to avoid this kind of lower bound of the clock period. The constraints for a multi-clock cycle path are defined similarly as a single-clock cycle path.

No-Zero-Clocking(Setup) Constraint

$$s(u) - s(v) \leq aT - d_{\max}(u, v)$$

No-Double-Clocking(Hold) Constraint

$$s(v) - s(u) \leq d_{\min}(u, v) - bT$$

where a and b are given constant ($b < a$). Note that for a single-clock cycle path, a and b are given as 1 and 0, respectively. Note that this formulation enables us to handle the circuit with multi-clocks that have different clock periods.

In this paper, we assume that the clock timing of each register is controlled as we design. If b is 0 for each path, the feasible clock period has no upper bound. That is, if T is feasible then for any $T^{'}$ ($T^{'} \geq T$) is feasible. However, the feasible clock period has an upper bound if b is not 0 for some paths. For the circuit consists of single-clock cycle path only, the delay variation and clock jitter do not effect too much because there are no upper bound of the feasible clock period. We can set the target clock period by considering these effects. We can obtain a circuit tolerable to these effects by setting target clock period small enough. However, if a circuit contains a multi-clock cycle path, we cannot obtain a circuit tolerable to delay variation and clock jitter by setting the target clock period small enough only, because there is an upper bound of the feasible clock period. So, to use the multi-clock cycle path technique, we need to design a circuit that works correctly within the **clock period range**.

These constraints are represented by the constraint graph. The constraint graph $G(V, E)$ of a circuit is defined as follows: a vertex $v \in V$ corresponds to a register; a directed edge $(u, v) \in E$ corresponds to either type of constraints; edge (u, v) corresponding to the setup (hold) constraint is called setup-edge (hold-edge), and the weight $w(u, v)$ of (u, v) is $aT - d_{\max}(u, v)(d_{\min}(u, v) - bT)$. It is known that the constraints can be satisfied if and only if the constraint graph contains no negative weight cycle [4].

Let G be the constraint graph of a circuit and $T_{\min}(G)$ be the minimum feasible clock period of the circuit under the assumption that the clock-timing of each register can be controlled as we design.

If the clock schedule is given, by checking Hold and Setup constraints, the upper and lower bound of feasible clock period of the clock schedule is obtained in $O(m)$ where m is the number of register pairs with signal propagation.

III. Proposed Algorithm

We propose an algorithm that reduces the usage of intermediate registers while the circuit works correctly with the given clock period range $T^S_{\min}$ and $T^S_{\max}$.

In this paper, we consider the 2-stage pipelined circuit with I/O pins. The clock timing for I/O pins (vertices *in* and *out* in constraint graph) are fixed to 0. Therefore, the *in* and *out* vertices in the constraint graph can be combined to one vertex. The input of our algorithm is the constraint graph of 2-stage pipelined circuit with intermediate registers that works correctly with the clock period $T^S_{\min}$, and our target is to get a circuit with smaller area that works correctly with the given clock period range [$T^S_{\min}$,$T^S_{\max}$]. At first, our algorithm will remove all the intermediate registers and intermediate registers will be inserted to the circuit to satisfy the constraints.

A. Scheduling

When we remove the intermediate register v from the pipelined circuit as shown in Fig. 2, the circuit contains a 2-clock cycle path as shown in Fig. 3. Then the total propagation delay from register u to w will be changed as follows.

Before removing the intermediate register v

$$\begin{aligned} d^{\max}_b(u, w) &= d^{'}_{\max}(u, v) + setup(v) + d_{\max}(v) \\ &\quad + d^{'}_{\max}(v, w) + setup(w) + d_{\max}(w) \\ d^{\min}_b(u, w) &= d^{'}_{\min}(u, v) - hold(v) + d_{\min}(v) \\ &\quad + d^{'}_{\min}(v, w) - hold(w) + d_{\min}(w) \end{aligned}$$

After removing the intermediate register v

$d_a^{\max}(u,w) = d'_{\max}(u,w) + setup(w) + d_{\max}(w)$
$d_a^{\min}(u,w) = d'_{\min}(u,w) - hold(w) + d_{\min}(w)$

When $d'_{\max}(u,w) = d'_{\max}(u,v) + d'_{\max}(v,w)$ and $d'_{\min}(u,w) = d'_{\min}(u,v) + d'_{\min}(v,w)$, the difference of total propagation delay between before and after removing the intermediate register v is as follows.

$d_b^{\max}(u,w) - d_a^{\max}(u,w) = setup(v) + d_{\max}(v)$
$d_b^{\min}(u,w) - d_a^{\min}(u,w) = d_{\min}(v) - hold(v)$

Since our problem is to make the circuit works correctly with the given clock period range $[T^S_{\min}, T^S_{\max}]$, the constraints for a 2-clock cycle path that is obtained by removing the intermediate register v can be defined as follows.

No-Zero-Clocking (Setup) Constraint

$$s(u) - s(w) \leq 2T^S_{\min} - d_{\max}(u,w)$$

No-Double-Clocking (Hold) Constraint

$$s(w) - s(u) \leq (d_{\min}(u,w) - \delta) - T^S_{\min}$$

where δ is the difference between the given maximum and minimum clock period ($\delta = T^S_{\max} - T^S_{\min}$). See Fig. 4.

The constraints for a single-clock cycle path can be defined as follows.

No-Zero-Clocking (Setup) Constraint

$$s(u) - s(w) \leq T^S_{\min} - d_{\max}(u,w)$$

No-Double-Clocking (Hold) Constraint

$$s(w) - s(u) \leq d_{\min}(u,w)$$

If the above stated constraints are satisfied for a circuit, the circuit will work correctly within the clock period range $[T^S_{\min}, T^S_{\min} + \delta]$

B. Necessary condition for 2-clock cycle path

Theorem 1 *A circuit does not work correctly with the clock period range $[T^S_{\min}, T^S_{\max}]$, unless the following constraints are satisfied.*

$$T^S_{\max} \leq d_{\min}(in,u) + d_{\min}(u,w) + d_{\min}(w,out) \tag{1}$$

$$T^S_{\min} \geq d_{\max}(u,w) - d_{\min}(u,w) + \delta \tag{2}$$

where $(in,u), (u,w), (w,out)$ corresponds to the path from in to register u, the 2-clock cycle path from register u to register w, the path from register w to out, respectively.

Proof : Let consider the constraint graph shown in Fig. 3. Note that vertices *in* and *out* can be combined to one vertex because the clock timing for input and output pins are fixed to 0. Therefore path (in,u,w,out) will be a cycle. So the total weight of the cycle (in,u,w,out) must be larger than or equal to 0 unless the circuit will not work correctly with clock period $T^S_{\max}$. See Fig. 3.

$0 \leq d_{\min}(in,u) + d_{\min}(u,w) - T^S_{\max} + d_{\min}(w,out)$.

Let consider the constraint graph shown in Fig. 5. The total weight of cycle (u,w,u) must be larger than or equal to 0 unless the circuit will not work correctly with clock period $T^S_{\min}$. See Fig. 5.

$0 \leq 2T^S_{\min} - d_{\max}(u,w) + d_{\min}(u,w) - \delta - T^S_{\min}$. ∎

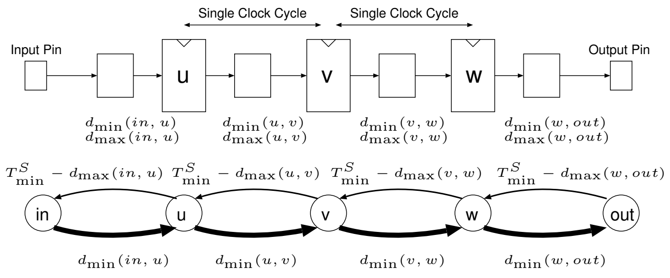
Single Clock Cycle
Single Clock Cycle
Input Pin
Output Pin
u
v
w
$d_{\min}(in,u)$ $d_{\max}(in,u)$
$d_{\min}(u,v)$ $d_{\max}(u,v)$
$d_{\min}(v,w)$ $d_{\max}(v,w)$
$d_{\min}(w,out)$ $d_{\max}(w,out)$
$T^S_{\min} - d_{\max}(in,u)$
$T^S_{\min} - d_{\max}(u,v)$
$T^S_{\min} - d_{\max}(v,w)$
$T^S_{\min} - d_{\max}(w,out)$
in
out
$d_{\min}(in,u)$
$d_{\min}(u,v)$
$d_{\min}(v,w)$
$d_{\min}(w,out)$

Fig. 2. The circuit before removing the intermediate register v : single-clock cycle-path

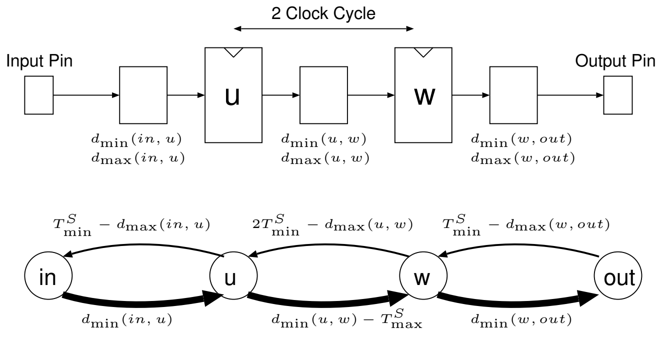
2 Clock Cycle
Input Pin
Output Pin
u
w
$d_{\min}(in,u)$ $d_{\max}(in,u)$
$d_{\min}(u,w)$ $d_{\max}(u,w)$
$d_{\min}(w,out)$ $d_{\max}(w,out)$
$T^S_{\min} - d_{\max}(in,u)$
$2T^S_{\min} - d_{\max}(u,w)$
$T^S_{\min} - d_{\max}(w,out)$
in
out
$d_{\min}(in,u)$
$d_{\min}(u,w) - T^S_{\max}$
$d_{\min}(w,out)$

Fig. 3. The circuit after removing the intermediate register v : 2-clock cycle-path

C. Algorithm

Inputs : Constraint graph G^{in} of a 2-stage pipelined circuit with intermediate registers, the target minimum (maximum) Clock Period $T^S_{\min}$ ($T^S_{\max}$)

Outputs : Constraint graph G^{out} of the circuit after removing the intermediate registers

Step 0 : Remove all of the intermediate registers. Let G^0 be the constraint graph of the obtained circuit.

Step 1 : Insert the intermediate registers to the 2-clock cycle paths that violate the constraints shown in inequalities (1) and (2) and update the constraint graph. If there is no more 2-clock cycle path, output the constraint graph and terminate. Otherwise, let G^1 be the constraint graph of the obtained circuit and go to the next step.

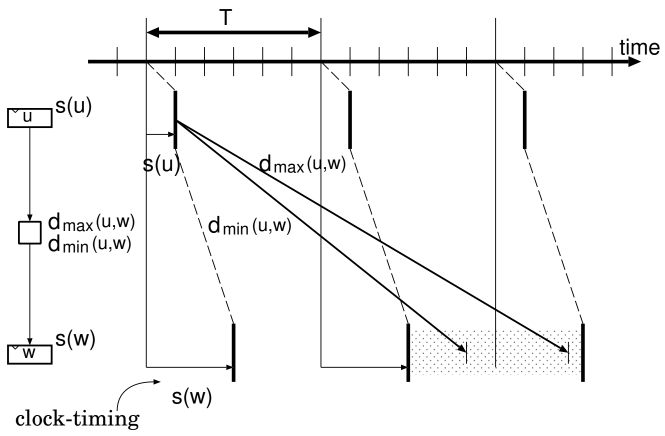

Fig. 4. Timing chart for a 2-clock cycle path (Pipelined without intermediate register)

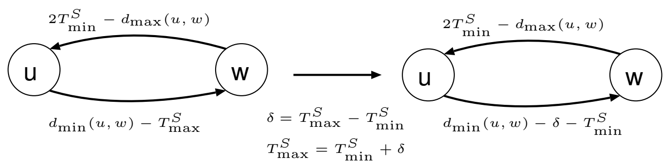
$2T^S_{\min} - d_{\max}(u,w)$
u
w
$d_{\min}(u,w) - T^S_{\max}$
$\delta = T^S_{\max} - T^S_{\min}$
$T^S_{\max} = T^S_{\min} + \delta$
$2T^S_{\min} - d_{\max}(u,w)$
$d_{\min}(u,w) - \delta - T^S_{\min}$

Fig. 5. The cycle contains a 2 clock cycle path

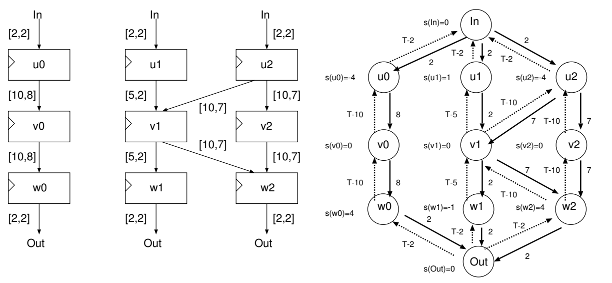

Fig. 6. Pipelined circuit with intermediate registers and the corresponding constraint graph G^{in} , $(T_{\min}(G^{in}) = 6)$

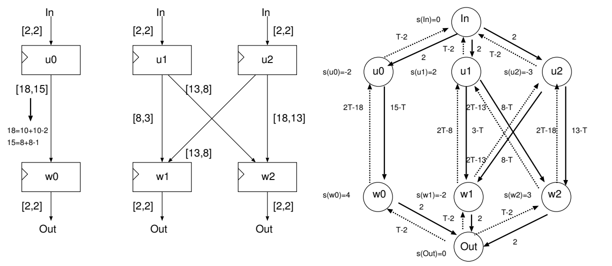

Fig. 7. Pipelined circuit after removing all intermediate registers and the corresponding constraint graph G^0, $(T_{\min}(G^0) = 6)$

Step 2 : Let $T_{\text{check}}(G^1) = \max(\max_{(u,w)\in E^1(G^1)} d_{\max}(u,w), \max_{(u,w)\in E^2(G^1)} \frac{d_{\max}(u,w)}{2})$, where $d_{\max}(u,w)$ is the maximum delay of the register pair with signal propagation, $E^1(G^1)$ is the set of single-clock cycle path and $E^2(G^1)$ is the set of 2-clock cycle path. Let $W_{\min}(G^1)$ be the total minimum delay from *in* to *out* that contains the 2-clock cycle path. If the circuit does not work correctly with the clock period $T_{\text{check}}(G^1)$, insert the intermediate register to the 2-clock cycle path which corresponds to an edge contained in the path that determines $W_{\min}(G^1)$ and update the constraint graph. Repeat until the obtained circuit works correctly with the clock period $T_{\text{check}}(G^1)$. If there is no more 2-clock cycle path, output the constraint graph and terminate. Otherwise, let G^2 be the constraint graph of the obtained circuit and go to the next step.

Step 3 : Compute the minimum clock period $T_{\min}(G^2)$. If $T_{\min}(G^2) > T^S_{\min}$, insert the intermediate register to the 2-clock cycle path which corresponds to a hold-edge contained in the critical cycle of the constraint graph G^2_t, where $t = T_{\min}(G^2)$ and update the constraint graph. Repeat until $T_{\min}(G^2) \leq T^S_{\min}$. If there is no more 2-clock cycle path which corresponds to a hold-edge contained in the critical cycle of the constraint graph G^2_t, output the constraint graph after inserted all of the intermediate registers and terminate.

D. Example

To explain the behavior of the algorithm, we apply it to the pipelined circuit shown in Fig. 6. We fix the clock timings of all input and output pins (*in* and *out*) to 0, assume Setup and Hold Time for registers are 0 and the maximum and minimum delay of an intermediate register are 2 and 1, respectively. Then $T_{\min}(G^{in}) = 6$. Let $T^S_{\min}$ and $T^S_{\max}$ be 7 and 9, respectively. The circuit after removing the intermediate registers v_0, v_1, v_2 is shown in Fig. 7. Since $d_{\min}(in, u_1) + d_{\min}(u_1, w_1) + d_{\min}(w_1, out) = 2 + 3 + 2 = 7 < T^S_{\max} = 9$, the intermediate register v_1 is inserted. The circuit after inserting the intermediate register v_1 is shown in Fig. 8. When the algorithm pro-

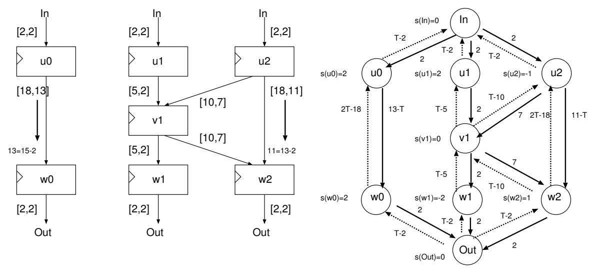

Fig. 8. Pipelined circuit after inserting the intermediate register v_1, changed the 2-clock cycle path minimum delay $d_{\min}$ to $d_{\min} - \delta$ and the corresponding constraint graph G^2, $(T_{\min}(G^2) = 9)$

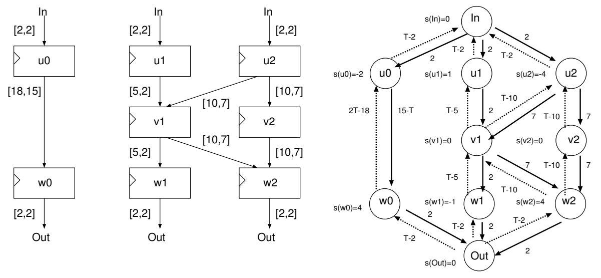

Fig. 9. Pipelined circuit after inserting the intermediate register v_2 and the corresponding constraint graph G^{out}, ($T_{\min}(G^{out}) = 6$)

TABLE I

The changes of the clock period range

Number Of Intermediate Registers	Clock Period Range $[T_{\min}, T_{\max}]$
$3(v0, v1, v2)$	[6, Infinite]
0	[6, 7]
$1(v1)$	[9, 9]
$2(v1, v2)$	[6, 9]

ceeds to step 3, we get $T_{\min}(G^2) = 9$ from the critical cycle (w_2, v_1, u_2, w_2) which is larger than $T^S_{\min} = 7$. There is 2-clock cycle path (path (u_2, w_2)) which corresponds to a hold-edge contained in the critical cycle (w_2, v_1, u_2, w_2) of the constraint graph G^2_9. Therefore, the intermediate register v_2 is inserted. The circuit after inserting the intermediate register v_2 is shown in Fig. 9. Since $T_{\min}(G^2) = 6$ (critical cycle: $(out, w_2, v_2, u_2, in)) < T^S_{\min} = 7$, so no more intermediate register need to be inserted.

The changes of the clock period range and the number of the intermediate registers for above example is shown in Table I.

If we implement the complete-synchronous circuit method (The clock timing for all registers are fixed to 0) together with the 2-clock cycle path technique to the above example, when we remove the intermediate register v_0 (path $(u0, w0)$ will become a 2-clock cycle path) the clock period range will be $[10, 15]$. So we cannot get the circuit that works correctly within our target clock period range which is $[7, 9]$. However, by implementing the semi-synchronous circuit method together with the 2-clock cycle path technique we will get the smaller circuit (circuit without the intermediate register $v0$) that works correctly within our target clock period range.

IV. Experiments

The proposed algorithm was applied to 4-bit and 8-bit 2-stage pipelined adder circuits. The delay statistics of the circuits are shown in Table II. Total number of the registers for 4-bit and 8-bit circuit are 21 and 41, respectively. We used ROHM 0.35 process library for these experiments.

TABLE II

Delay of pipelined adder circuit

Bit	1st Stage[ns]		2nd Stage[ns]	
	min	max	min	max
4	0.575	1.259	0.392	1.965
8	0.584	1.283	0.614	2.890

Results are presented in tables III, IV and V. In tables III, IV and V, "$[T^S_{\min}, T^S_{\max}]$" and " $[T_{\min}(G^{out}), T_{\max}(G^{out})]$ " are the target and output clock period range, respectively. "#FF" is the number of registers. "#Int. FF" is the number of intermediate registers. "Area" is the sum of the gates and registers area.

The first and second experiments on 4-bit and 8-bit adders are to see the effect of the minimum clock period to the number of the intermediate registers. The target minimum clock period $T^S_{\min}$ was set to $110\%, 120\%, 130\%, 140\%, 150\%, 160\%, 170\%, 180\%$ of the minimum clock period $T_{\min}(G^{in})$ of the original circuit. And the clock period range is 10% of the minimum clock period of the original circuit. From the results, it is shown that in most of the cases, by increasing the target minimum clock period we can reduce more on the usage of the intermediate registers.

The third experiment on 8-bit adder is to see the effect of the clock period range to the number of the intermediate registers. The target minimum clock period $T^S_{\min}$ was set to 150% of the minimum clock period $T_{\min}(G^{in})$ of the original circuit. And the clock period range are $0.00ns, 0.050ns, 0.100ns, 0.150ns, 0.200ns, 0.250ns, 0.300ns, 0.350ns$. From the results, it is shown that in most of the cases, the increasing of the clock period range is proportional to the number of the intermediate registers.

V. Conclusions and Future Works

We show that the usage of intermediate registers on pipelined circuit can be reduced by implementing the 2-clock cycle path technique together with the semi-synchronous circuit method.

Our proposed algorithm only inserts the intermediate register without considering the delay padding in order to make the circuit works correctly within the target clock period range. We believe that by considering the delay padding together with the intermediate registers insertion, the circuit will be more smaller.

As future works, the effect on the circuit area by the combination of the delay padding together with the intermediate registers insertion should be investigated.

References

[1] R.B. Deokar and S.S. Sapatnekar. A graph-theoretic approach to clock skew optimization. In *Proc. ISCAS'94*, pages 407–410, 1994.

TABLE III
The effect of the minimum clock period to the number of intermediate registers (4-bit adder).

$T^S_{\min}$ (%)	$[T^S_{\min}, T^S_{\max}]$ (ns)	$[T_{\min}(G^{out}), T_{\max}(G^{out})]$ (ns)	#FF	#Int. FF	Area (%)
100	[1.40, Inf.]	[1.40, Inf.]	21	8	100
110	[1.54, 1.68]	[1.54, 1.68]	13	0	75
120	[1.68, 1.82]	[1.62, 1.82]	13	0	75
130	[1.82, 1.96]	[1.82, 1.96]	13	0	75
140	[1.96, 2.10]	[1.91, 2.10]	13	0	75
150	[2.10, 2.24]	[1.98, 2.24]	13	0	75
160	[2.24, 2.38]	[2.05, 2.38]	13	0	75
170	[2.38, 2.52]	[2.38, 2.52]	14	1	78
180	[2.52, 2.66]	[2.52, 2.66]	14	1	78

TABLE IV
The effect of the minimum clock period to the number of intermediate registers (8-bit adder)

$T^S_{\min}$ (%)	$[T^S_{\min}, T^S_{\max}]$ (ns)	$[T_{\min}(G^{out}), T_{\max}(G^{out})]$ (ns)	#FF	#Int. FF	Area (%)
100	[1.71, Inf.]	[1.71, Inf.]	41	16	100
110	[1.88, 2.05]	[1.71, Inf.]	41	16	100
120	[2.05, 2.22]	[1.71, Inf]	41	16	100
130	[2.22, 2.39]	[1.71, Inf]	41	16	100
140	[2.39, 2.56]	[2.39, 2.56]	38	13	96
150	[2.56, 2.73]	[2.56, 2.73]	37	12	94
160	[2.73, 2.90]	[2.73, 2.90]	36	11	93
170	[2.90, 3.07]	[2.90, 3.07]	36	11	93
180	[3.07, 3.24]	[3.07, 3.24]	38	13	96

TABLE V
The effect of the clock period range to the number of intermediate registers (8-bit adder)

Range (ns)	$[T^S_{\min}, T^S_{\max}]$ (ns)	$[T_{\min}(G^{out}), T_{\max}(G^{out})]$ (ns)	#FF	#Int. FF	Area (%)
	[1.71, Inf.]	[1.71, Inf.]	41	16	100
0	[2.56, 2.56]	[2.56, 2.56]	26	1	78
0.05	[2.56, 2.61]	[2.56, 2.61]	28	3	81
0.10	[2.56, 2.66]	[2.56, 2.66]	28	3	81
0.15	[2.56, 2.71]	[2.56, 2.71]	37	12	94
0.20	[2.56, 2.76]	[2.56, 2.76]	38	13	96
0.25	[2.56, 2.81]	[2.56, 2.81]	38	13	96
0.30	[2.56, 2.86]	[2.56, 2.86]	40	15	99
0.35	[2.56, 2.91]	[2.56, 2.91]	38	13	96

[2] J.P. Fishburn. Clock skew optimization. *IEEE Trans. on Computers*, 39:945–951, 1990.

[3] A. Takahashi, K. Inoue, and Y. Kajitani. Clock-tree routing realizing a clock-schedule for semi-synchronous circuits. In *Proc. 1997 ICCAD*, pages 260–265, 1997.

[4] A. Takahashi and Y. Kajitani. Performance and reliability driven clock scheduling of sequential logic circuits. In *Proc. ASP-DAC '97*, pages 37–42, 1997.

[5] A. Takahashi, W. Takahashi, and Y. Kajitani. Clock-routing driven layout methodology for semi-synchronous circuit design. In *Proc. TAU '97*, pages 63–66, 1997.

[6] Woo Jin Kim and Yong-Bin Kim Clocking for correct functionality on wave pipelined circuits In *SOC Conference, 2003. Proceedings. IEEE International* , pages 161 - 164, Sept. 2003 .