Concurrent Timing Optimization of Latch-Based Digital Systems [1]

Hong-Yean Hsieh, Wentai Liu, Ralph K. Cavin
Electronics Research Laboratory
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Abstract

Many design techniques have been proposed to optimize the performance of a digital system implemented in a given technology. Each of these techniques can be advantageous in particular applications, and they are often applied individually to enhance performance. Previous results had shown that significant enhancement could be achieved when several optimizations were applied concurrently to systems with edge-triggered flip-flops. However concurrent optimization framework does not exist for systems with transparent latches. It motivates us to formulate the concurrent optimization as a mixed integer linear programming for digital systems with transparent latches. This methodology is applicable to optimize a broad range of digital systems originally designed under either single or multi-phase clocking. In addition, we present a new optimization technique - resynchronization, which allows the insertion of latches in the shortest paths and thus avoids race condition. As a result, our framework is ready to include the resynchronization technique. Our formulation has been applied to several design examples and is able to significantly reduce the clock period.

[1] This research is partially supported by NSF grant MIP-92-12346

1 Introduction

Optimization for clock period is crucial in high performance digital systems. Various techniques for clock period reduction have been proposed, including transistor sizing, logic resynthesis, intentional clock skew [4], cycle stealing with transparent latches [6, 9, 11, 13], retiming [5], and wave pipelining [16, 17, 18, 19, 20]. Each of these techniques can be advantageous in particular applications, and they are often applied individually to enhance performance. The work [2] presented a framework for timing optimization by *simultaneously* applying retiming, intentional clock skew, and wave pipelining for a system with edge-triggered flip-flops (registers). As a result, the concurrent application of optimization techniques achieves significantly reduced clock period. In contrast, the characteristic of a transparent latch is different from that of a register. Mathematical constraints and thus framework of concurrent optimization for latch-based designs do not exist. Accordingly this paper mainly focuses on the concurrent optimization of designs with either single-phase or multi-phase transparent latches, hereafter called latches.

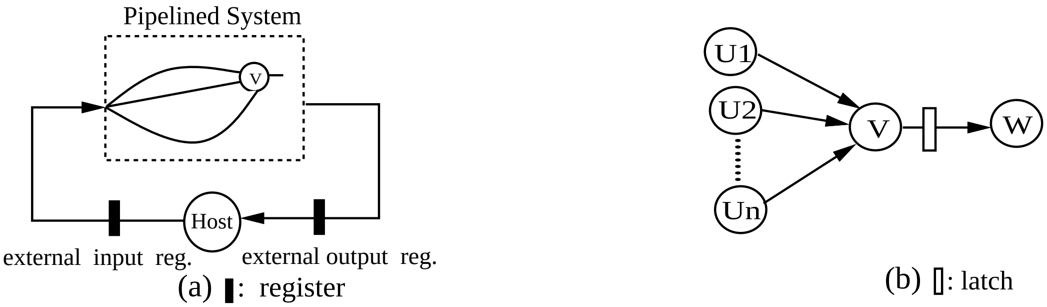
Digital designs usually take advantage of the transparency property of latches to reduce the clock period of a system by the technique of cycle stealing. A combinational block can borrow a portion of cycle time from its neighboring one, thus effectively reducing the clock period. This implies that there might be two "waves" of data presented in a combinational block that, in a sense, is a type of wave pipelining. Also, as proposed by Fishburn [4], the insertion of intentional clock skew introduces another type of wave pipelining in a combinational block. Due to more stringent synchronization requirement for wave pipelining, previous works on wave pipelining are restricted to register based designs with few exceptions [21, 22]. However, works in [21, 22] deal with a circular wave pipelined loop with latches by using a special clocking scheme, called coincident clocking, and a set of predefined clock skews. Our new formulation differs from the previous ones [21, 22] in three aspects: (1) a general system including feedback loops and reconvergent fanout paths is handled; (2) clock skew is not predefined but is optimized so that the clock period is minimal; and (3) the number of data waves in each combination block depends on its capacity and the system in the optimization process, rather than being predefined. In a sense, our formulation is the first one, to our knowledge, for wave pipelining in a general system with latches.

Retiming equalizes propagation delays of different stages of a system by relocating registers or latches [5]. Further reduction of the clock period is possible by combining retiming with other techniques [1, 13, 14, 15]. Insertion of latches in the shortest paths could remove race conditions and thus reduce the clock period. This new technique is called resynchronization. In this paper, we present a mixed integer linear programming formulation in which concurrent optimization by (1) intentional clock skew, (2) wave pipelining, and (3) either retiming or resynchronization is formulated for latch-based designs.

Due to process and environmental variations, it is unavoidable that design parameters, such as the longest and shortest propagation delays of a gate, intentional clock skews, etc., would deviate from the specifications. In order to optimize the yield of a design, tolerance to these variations should be maximized. The tolerance can be quantitatively measured by *safety margin*, δ, by which the clock occurrence is allowed to deviate from the design specification time, t. As pointed out by work [2], safety margin can be maximized by increasing the system clock period, prolonging the short paths, and/or reducing the long paths of a design. However, for a design of which the clock period is valid only for an interval, (*i.e.* it is *bounded above*), this amount can no longer increase when the system clock period is beyond the interval. To avoid this, a system should be designed carefully and verification criteria must be developed. The criteria can be established in terms of the number of latches at each edge.

This paper is organized as follows. Section 2 defines the terminologies used for retiming, transparent latches, and wave pipelining. In Section 3, we discuss in detail the difficulty and complexity incurred by feedback loops and reconvergent fanout paths when optimizing single or multi-phase designs via wave pipelining. Section 4 provides a graph model for a pipelined system with fixed locations of latches. When optimizing wave pipelined systems, LP constraints are formulated to preserve these systems' functionality and temporality. In Section 5, these constraints are modified to accommodate the retiming process in the optimization formulation. In Section 6, the motivation of resynchronization is discussed first and then the corresponding mixed integer formulation is presented. In Section 7 the safety margin of a design is defined and its upperbound is derived. Section 8 establishes the criteria to verify if the feasible clock period is unbounded above and then shows the necessary conditions under which the safety margin will exempt from saturation as the clock period increases. Experimental data for other circuits is provided in Section 9. Finally, directions for future research are given.

For clarity, all the proofs will be put in the appendix.



(a) ▮: register

(b) ▯: latch

Figure 1: (a) A System View (b) A Circuit

2 Background

In this section, we first define the proposed timing model and the variables used here. Then a brief literature review for retiming, transparent latch, and wave pipelining is given.

2.1 Timing Model

The longest propagation delay, $t_{max}(v)$, and the shortest propagation delay, $t_{min}(v)$, are defined over every gate or combinational block, which is referred to as a node v in this paper. These amounts are assumed to be measured under worst case conditions. Although the timing model assumes the constant longest and shortest propagation delays for all input-output pairs of an individual node, it can be easily generalized to the case in which propagation delays for input-output pairs of a node are not equal.

2.2 Definitions of Variables

As shown in Fig. 1(a), a host machine applies data, d_i, to a pipelined system through *external input registers* at time $i \cdot c$, where c is the system cycle time, and extracts data from it through *external output registers*. Unless stated explicitly, these external registers will not be shown in figures. Data d_i arrives at node v through different paths. Calculated from the host, variables $s(v)$ and $b(v)$, the latest and earliest arrival times respectively at the output of node v, are defined as follows:

$$\begin{aligned} s(v) &= \text{the longest path delay to the output of node } v \text{ plus } i \cdot c \\ b(v) &= \text{the shortest path delay to the outputs of node } v \text{ plus } i \cdot c \end{aligned}$$

For a circuit shown in Fig. 1(b), node v is connected to node w through a latch. The time for the latching and enabling edge (defined in Fig. 2(a)) to capture data d_i is denoted by $l(v)$ and $e(v)$

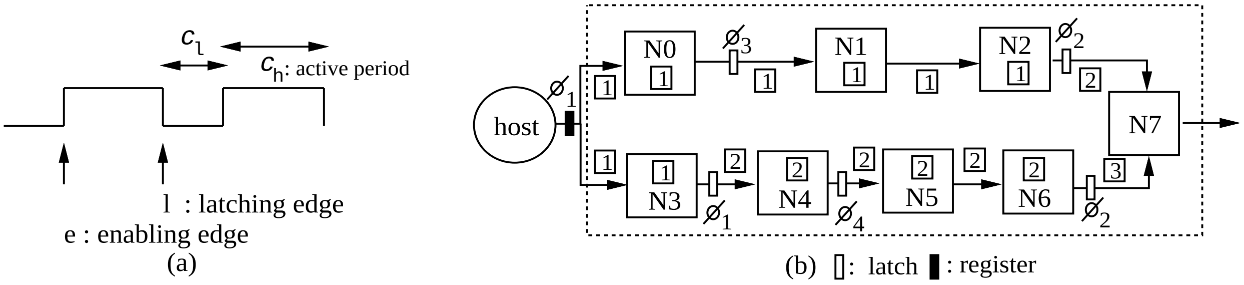


Figure 2: (a) A Typical Clock Phase (b) A Multiple Phase System (external input registers are shown in the figure.)

respectively. The latest departure time, $D(v)$, of this latch is defined as the latest time of data d_i available at node w. Similarly, the earliest departure time, $d(v)$, is defined as the earliest time of data d_i available at node w.

2.3 Retiming

Retiming is a technique which can improve circuit performance by relocation of latches or registers [5]. In a directed graph, node v represents a combinational block and a directed edge, $e(u, v)$ or $u \xrightarrow{e} v$, represents the connection between u and v. In this case, u is called a *fan-in node* of v, and v is a *fanout node* of u. Edge $e(u, v)$ is a *fan-in edge* of node v and a *fanout edge* of u. The weight, $w(e)$, is the number of latches associated with edge $e(u, v)$. At node v, the retiming process moves an equal number of latches, represented by an integer variable $r(v)$, from every fanout edge to every fan-in edge. After retiming, the number of latches with an edge $u \xrightarrow{e} v$ is $w(e) - r(u) + r(v)$. A valid retiming process must preserve the property of nonnegative edge weight and is represented mathematically by

$$w(e) - r(u) + r(v) \geq 0 \text{ for all edges } u \xrightarrow{e} v$$

2.4 Transparent Latch Models

Transparent latches provide advantages over registers in digital system designs. For example, the area of a latch is generally smaller than that of a register. Also, cycle stealing is only possible with latches. Refer to the example shown in Fig. 1(b) and assume that the propagation delay, setup time and hold time of a latch are all equal to 0. Several models [6, 7, 8, 9, 11], shown in Table 1, have been proposed to verify or optimize latch-based designs. The model shown in the second column is adopted by this paper. Since SMO model results in a non-convex solution set, which in turn leads to the problem of

Unger and Tan's Model	Adopted Model	SMO Model
$D(v) = l(v)$	$D(v) = \max\{e(v), s(v)\}$	$D(v) = \max\{e(v), s(v)\}$
$d(v) = e(v)$	$d(v) = e(v)$	$d(v) = \max\{e(v), b(v)\}$

Table 1: Various Models for Latch-Based Designs

reachability and startability [10, 21], it will not be considered here.

2.5 Wave Pipelining in Latch-Based Designs

Wave pipelining is a timing methodology used in digital systems to achieve maximal rate operation [4, 16, 17, 18, 19, 20]. Using this technique, new data is applied to the inputs of a combinational block before the previous result is clocked out, thus effectively pipelining the logic and maximizing the utilization of the logic without inserting latches or registers.

Applying wave pipelining to latch-based designs imposes new constraints. For explanation, constraints are referred to Fig. 1(b). In order to capture data d_i correctly, the latest arrival time, $s(v)$, at node v should be smaller than or equal to the latching time, $l(v)$, minus the setup time of the latch. This is also known as the zero clocking constraint [4]. In order to prevent the later data from overriding the previous data, the earliest arrival time, $b(v)$, at node v plus clock period, c, should be greater than or equal to the sum of $l(v)$ and the hold time of this latch. This is also known as the double clocking constraint [4].

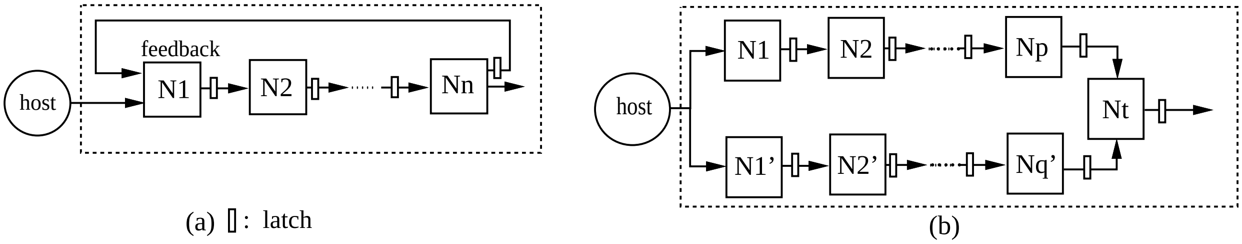
3 Calculation of Shift of Time Origin

In this section, issues related to wave pipelining in a system containing feedback loops and reconvergent fanout paths are discussed. We first consider designs with single phase and then multi-phase clocking. Although these issues are discussed with latch-based designs, they are also true for register based designs.

As shown in Fig. 1(a), for a path p from the host to node v, *temporality*, $\varphi(v, p)$, is defined as follows [12]:

Definition 1 *Temporality, $\varphi(v, p)$, is defined as the number of clock cycles for a data originating from the host to reach node v along path p.*

3.1 Single Phase Designs



(a) ▯ : latch (b)

Figure 3: (a) Feedback Loop (b) Reconvergent Fanout Paths

Original Formulas	Relaxed Formulas
$s(t) = \max\{s(p), s(q')\} + t_{max}(t)$	$s(t) \geq s(p) + t_{max}(t)$ and $s(t) \geq s(q') + t_{max}(t)$
$b(t) = \min\{b(p), b(q')\} + t_{min}(t)$	$b(t) \leq b(p) + t_{min}(t)$ and $b(t) \leq b(q') + t_{min}(t)$

Table 2: Original and Relaxed Formulas

In this subsection, designs with a single phase clocking are investigated. Fig. 3(a) shows an example in which there are n latches along the feedback loop. If no skew is allowed at these latches, data d_{i-n} arrives at node 1 at the i-th cycle when it traverses the feedback loop once. Thus both d_{i-n} and d_i are simultaneously available at node 1. If d_{i-n} interferes with either d_{i+1} or d_{i-1} ($n \neq 1$) after applying wave pipelining, then the system might function incorrectly. In order to preserve the functionality of a wave pipelined system, there should be n waves of data in the system. If the system supports $m \cdot n$ waves of data (m is a positive integer), correct operation can be obtained if the system includes an additional $m \times 1$ multiplexer and $1 \times m$ demultiplexer such that input/output data are carefully scheduled via multiplexer/demultiplexer. In this paper, m is defined as the *multiplexing degree* of a pipelined system. It is clear that with this new construct, d_i should be paired with $d_{i-m\cdot n}$, instead of d_{i-n} (if $m > 1$).

Fig. 3(b) shows the case of a system with two reconvergent fanout paths, which are defined as two simple paths sharing both start and end nodes, and form a semi-loop in graph terminology. Thus these two terms are used interchangeably in this paper. In our example, data arrives at node t by traversing either p or q latches (excluding the external registers). Note that a feedback loop can be regarded as a special case of semi-loop (*i.e.* $p = n$, $q = 0$). In a single phase pipelined system, d_i arrives at node t via the top route after p cycles meanwhile $d_{i-(q-p)}$ arrives at node t via the bottom route. In the process of wave pipelining, this property should be taken into account.

When the value of $q - p$ is 0, variables $s(t)$ and $b(t)$ at node t are derived from its fan-in nodes respectively and shown in the first column of Table 2. The result is derived with an assumption that there are no latches at edges $e(p,t)$ and $e(q',t)$. As shown in the second column of Table 2, it has been proven that operators *max* and *min* can be replaced with operators $\geq$ and $\leq$ respectively such that the solution set for c is neither expanded nor shrunk [8, 18]. If the value of $q-p$ is not 0, a suitable amount should be subtracted from variables s and b at the fan-in nodes of node t. This is discussed in the next paragraph. There are two alternative routes taken by d_i. Constraint formulation based on either of these two routes is plausible. Thus constraint formulation can be done by arbitrarily choosing the alternatives.

If d_i reaches node t by the top route, constraints derived along the bottom one for node t, are applicable if the values of variables $s(q')$ and $b(q')$ are reduced by the amount of $(q-p)c$ first. This effectively preserves the constraint that d_i should be paired with $d_{i-(q-p)}$ at node t when $m=1$. In a pipelined system with a multiplexing degree m, the correct amount of reduction is $m(q-p)c$. The amount of $m(q-p)$ is then referred to as the *shift of time origin* for the bottom route. Similarly, if d_i reaches node t by bottom route, the shift of time origin for the top route should be $m(p-q)$.

For a single phase clocking design as shown in Fig. 3(b), temporality $\varphi(t, top\_path)$ is $p+1$ and temporality $\varphi(t, bottom\_path)$ is $q+1$. The shift of time origin can be defined as follows:

Definition 2 *With reconvergent fanout paths of $u \xrightarrow{p} v$ and $u \xrightarrow{q} v$, the shift of time origin for path p is $m(\varphi(t,p)-\varphi(t,q))$ and for path q is $m(\varphi(t,q)-\varphi(t,p))$, where m is the multiplexing degree.*

In a general pipelined system, a spanning tree is identified first. Then each of the remaining edges, called chord, defines a semi-loop. The shift of time origin associated with each chord is then calculated accordingly.

3.2 Multi-Phase Designs

In a design with n-phase clocking, different phases ϕ_i are ordered in a predefined sequence such that $l_g(1) \leq l_g(2) \leq l_g(3) \cdots \leq l_g(n-1) \leq l_g(n)$, where $l_g(i)$ represents the latching time of phase ϕ_i, $1 \leq i \leq n$ [8, 11]. It is worth pointing out that $l_g(i)$ is defined in a global reference frame such that $0 < l_g(i) \leq c$, and differs from the one defined in Section 2.2. When latch s_x with phase ϕ_i feeds directly to latch s_y with phase ϕ_j, if $l_g(i) < l_g(j)$, data departing from latch s_x should be captured by the immediately following latching edge at latch s_y. Otherwise, it should be captured by the latching edge in the next clock cycle. Here external registers are assumed to latch data at phase ϕ_1.

Assume that at most only one latch is at each edge. If there is no latch at edge $e(u,v)$, the temporality, $\varphi(e(u,v),p)$, for an edge $e(u,v)$ is defined as the number of clock cycles for a data originating from the host to reach node u along path p. If there exists a latch at edge $e(u,v)$, it is defined as the number of clock cycles for a data along path p to the output of this latch. The temporality of a node can then be defined recursively with the temporality of an edge. For node v with its fan-in edge $e(u,v)$ along path p, $\varphi(v,p)$ is defined as follows:

$$\varphi(v,p) = \begin{cases} 0 & \text{if } v \text{ is a host} \\ \varphi(e(u,v),p) & \text{otherwise} \end{cases}$$

Let $\alpha(e(u,v),p)$ denotes the clock phase of the last latch along path p to node u. If there exists a latch at edge $e(u,v)$, $\beta(e(u,v),p)$ denotes the clock phase of the latch at edge $e(u,v)$. If not, $\beta(e(u,v),p)$ will be set to $n+1$. To meet the boundary condition, $\alpha(e(u,v),p)$ is initially set to $n+1$. The temporality of an edge, $\varphi(e(u,v),p)$, is defined as follows:

$$\varphi(e(u,v),p) = \begin{cases} \varphi(u,p) & \text{if } l_g(\alpha(e(u,v),p)) < l_g(\beta(e(u,v),p)) \\ \varphi(u,p)+1 & \text{otherwise} \end{cases}$$

A four phase design example is shown in Fig. 2(b). Along the top path, the temporality of node N_7 is 2. In contrast, the temporality along the bottom path is 3. The temporality for each node and edge is shown in the figure. It should be noted that this definition assumed that external registers are shown explicitly in the graph.

For a multi-phase design, if the latches that fan in to a node have the same phase, the shift of time origin of reconvergent fanout paths can then be calculated without ambiguity by calculating the temporality of each path.

4 Clock Optimization via Wave Pipelining of Latch-Based Designs

In this section, based on the graph model, we develop constraints required for wave pipelining in latch-based pipelined systems.

4.1 Graph Model for Pipelined Systems

A pipelined system is modeled as a directed graph $G = (V, V_{fo1}, V_{fo2}, V_l, V_{pi}, V_{po}, E, w, t_{max}, t_{min}, os)$, where

V set of functional nodes in the pipelined system

V_{fo1} set of dummy fanout nodes (explained later)

V_{fo2} set of dummy fanout nodes (explained later)

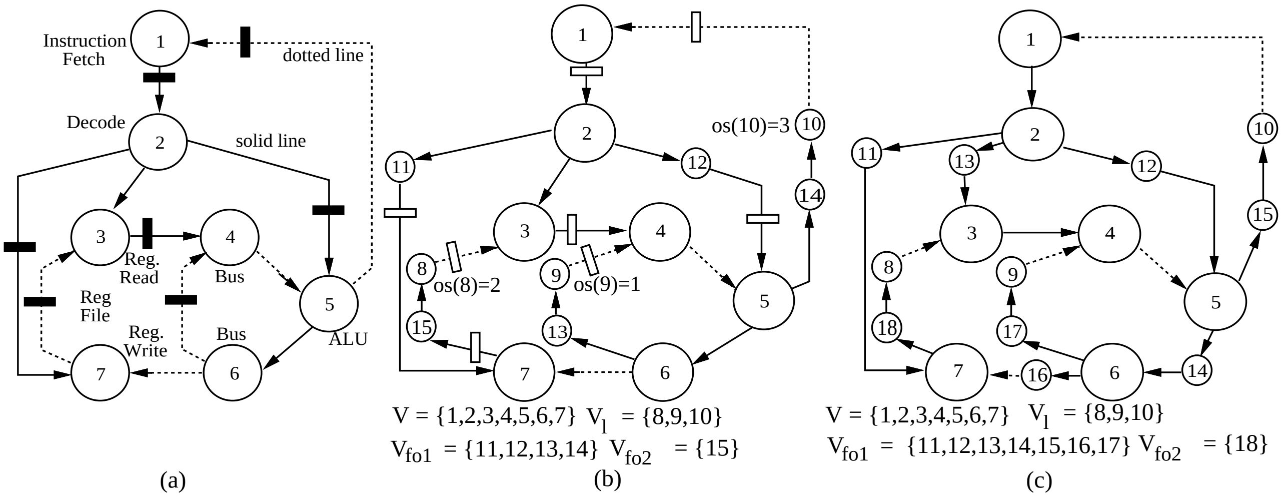


Figure 4: (a) A RISC Design (b) The Graph Model (c) Revised Graph Model

V_l set of dummy loop nodes (explained later)

V_{pi} vertex set as a host which drives primary inputs

V_{po} vertex set as a host which is driven by primary outputs

E set of directed edges

w edge weight (number of latches at each edge)

$t_{max}(v)$ longest propagation delay at each node v of $V \cup V_{fo1} \cup V_{fo2} \cup V_{po}$

$t_{min}(v)$ shortest propagation delay at each node v of $V \cup V_{fo1} \cup V_{fo2} \cup V_{po}$

$os(v)$ shift of time origin at each node v of V_l if its multiplexing degree is 1

A functional node refers to either a gate or a complex module. Vertex sets of V_{pi} and V_{po} depend on the system under study. In a closed system such as Fig. 4(a), they are empty sets. Otherwise, all primary inputs of a system are driven through external input registers by a host node, $v \in V_{pi}$, while all primary outputs are sent through external output registers to a host node, $v \in V_{po}$. All external input registers are assumed to be clocked without skew among them, and so are external output registers. Therefore, these external registers will not be shown in graph model, and for simplicity their propagation delays are assumed to be 0. But the model can easily be extended to the case of different latching times for each external register. The example shown in Fig. 4(a) is a RISC microprocessor design with single phase clocking scheme. Fig. 4(b) illustrates its graph model of the latch-based design. The solid lines and nodes form a spanning tree and the dotted lines are chords. The register at edge $e(7,3)$ shown in Fig. 4(a) is a register file. By combining the formulation described in work [2], it can be optimized as a register. But for simplicity, it is modeled by two latches in this paper. Three kinds of dummy nodes are created in the model. A dummy node $v_{f1} \in V_{fo1}$ is inserted in an edge $e(u,v)$ if there are more than one fanout edge for node u and $w(e) \geq 1$. This removes the restriction of applying the same latching time to the latches at every fanout edge of node u. In this procedure, edge weight $w(e(u, v_{f1}))$ is set to 0 and $w(e(v_{f1}, v))$ is $w(e)$. If more than one latch is required to appear at edge $e(u,v)$ (e.g. edge $e(7,3)$), another dummy node $v_{f2} \in V_{fo2}$ can be inserted to keep at most one latch appearing at each edge. Edge weight $w(e(u, v_{f2}))$ is then set to 1 and $w(e(v_{f2}, v))$ is $w(e) - 1$. For node $v \in V_{fo1} \cup V_{fo2} \cup V_{po}$, $t_{max}(v)$ and $t_{min}(v)$ are set to 0. As explained in the previous section, a chord, $e(u,v)$, defines a loop or semi-loop in a spanning tree. A dummy loop node $v_l \in V_l$ is inserted along the chord if the shift of time origin is not equal to 0. The *shift of time origin* for node v_l with a multiplexing degree of 1 is $os(v_l)$. In this insertion process, edge weight $w(e(u, v_l))$ is set to zero and $w(e(v_l, v))$ is $w(e)$.

4.2 Constraints for Wave Pipelining Latch-Based Designs

Constraints for valid wave pipelining in latch-based designs are presented in this section. These constraints are classified into the categories of pulse width, delay, synchronization, loop/semi-loop, latching, initiation, and temporal equivalence.

Let t_s and t_h denote the setup time and hold time of a latch. And t_{cl}, t_{cs}, and t_{dl} denote the longest propagation delay from clock input, shortest propagation delay from clock input, and longest propagation delay from data input to the output of a latch. The minimum allowable pulse width is denoted by w_l. Variables $s(v)$ and $b(v)$ are the latest and earliest arrival times respectively at the output of node v. For a node u connected to another node through a latch, $e(u)$ represents the time for the enabling edge of this latch. The width of the active period is represented by c_h, and c is the clock period. To simplify the design of a clock generator, all clocks use the same active period, c_h.

pulse width constraints:

$$c_h \geq w_l \tag{1}$$

$$c - c_h \geq w_l \tag{2}$$

delay constraints:

$$s(u) + t_{max}(v) \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \text{ and } w(e) = 0 \tag{3}$$

$$b(v) \leq b(u) + t_{min}(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \text{ and } w(e) = 0 \tag{4}$$

synchronization constraints:

$$s(u) + t_{max}(v) + t_{dl} \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \text{ and } w(e) = 1 \tag{5}$$

$$e(u) + t_{max}(v) + t_{cl} \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \text{ and } w(e) = 1 \tag{6}$$

$$b(v) \leq e(u) + t_{min}(v) + t_{cs} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \text{ and } w(e) = 1 \tag{7}$$

loop/semi-loop constraints:

$$s(v) = s(u) - m \cdot os(v) \cdot c \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V_l \tag{8}$$

$$b(v) = b(u) - m \cdot os(v) \cdot c \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V_l \tag{9}$$

latching constraints:

$$s(u) \leq e(u) + c_h - t_s \qquad \text{for } u \xrightarrow{e} v \text{ and } w(e) = 1 \tag{10}$$

$$e(u) + c_h + t_h \leq b(u) + c \qquad \text{for } u \xrightarrow{e} v \text{ and } w(e) = 1 \tag{11}$$

$$s(v) + t_s + t_h \leq b(v) + c \qquad \text{for } v \in V_{po} \tag{12}$$

initiation constraint:

$$b(v) \leq s(v) \qquad v \in V_{pi} \tag{13}$$

temporal equivalence constraint: optional

$$s(v) + t_s - s(u) = (k-1) \cdot m \cdot c \qquad u \in V_{pi} \text{ and } v \in V_{po} \tag{14}$$

Pulse width constraints restrict the minimum allowable pulse width. Delay and synchronization constraints define the timing relationship between node v and its fan-in node u. This depends on if there exists a latch in the edge $e(u, v)$. Constraint (3) specifies that if $w(e) = 0$, the latest arrival time, $s(v)$, must be greater than or equal to the sum of the latest arrival time, $s(u)$, and the longest propagation delay, $t_{max}(v)$, of node v. If $w(e) = 1$, as shown in constraint (5), the latest arrival time, $s(v)$, should be greater than or equal to the sum of $s(u)$, $t_{max}(v)$, and the propagation delay, t_{dl}, of the latch. Due to the synchronization requirement, as shown in constraint (6), $s(v)$, should also be greater than or equal to the sum of the time for the enabling edge, $e(u)$, $t_{max}(v)$, and the propagation delay, t_{cl}, of the latch. Constraint (4) specifies that if $w(e) = 0$, the earliest arrival time, $b(v)$, must be less than or equal to the sum of the earliest arrival time, $b(u)$, and the shortest propagation delay, $t_{min}(v)$, of node v. Otherwise, $b(v)$, must be less than or equal to the sum of $e(u)$, $t_{min}(v)$, and the propagation delay, t_{cs}, of the latch. Loop/semi-loop constraints incorporate the effects of semi-loops and feedback loops. Constraints (10) and (11) prevent the setup time and hold time violation of a latch respectively. Constraint (12) ensures that double clocking doesn't happen in the external output registers whose latching time is $s(v) + t_s$ where $v \in V_{po}$. If external temporal equivalence is preserved, temporal equivalence constraint should be satisfied where k is the temporality along the path in the spanning tree from node $u \in V_{pi}$ to node $v \in V_{po}$ [12].

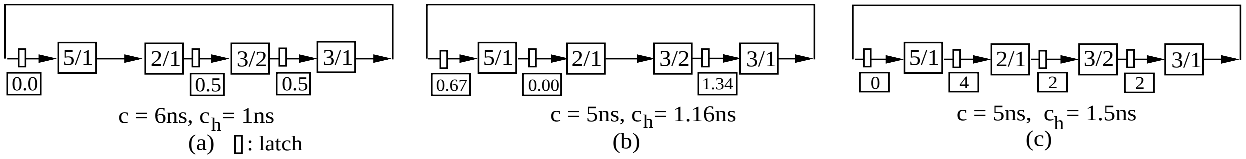


Figure 5: Retiming of a Circular Loop: $t_s = t_h = 0.5ns$ (a) Before Retiming, $w_l = 1ns$ (b) After Retiming, $w_l = 1ns$ (c) Resynchronization, $w_l = 1.5ns$ (number shown in a box near to a latch is the skew for latching edge. Numbers shown inside a node represent longest and shortest propagation delays.)

5 Clock Optimization via Retiming

In the previous section, an LP is formulated for clock period minimization if the locations of the latches are fixed. Retiming reduces the clock period by means of re-positioning latches or registers. Inclusion or modification of constraints is needed in LP formulation to allow retiming and wave pipelining to be applied to latch-based designs simultaneously. The optimization formulation becomes a mixed integer LP problem.

5.1 Retiming Example

For a circular loop as shown in Fig. 5(a), the optimal clock period is $6ns$ if 3 waves of data are propagating around the loop. For simplicity, the propagation delays of the latches are set to 0. By moving one latch from the fanout edge of the second node to its fan-in edge as shown in Fig. 5(b), the optimal clock period is reduced to $5ns$. It can be seen that re-positioning latches can further reduce the clock period.

5.2 Mixed Integer LP Formulation of Retiming

Since no advanced information about latches' positions is available after applying retiming, the graph model must be modified accordingly. For a node with multiple fanout edges, a dummy node is inserted at each fanout edge. For example, Fig. 4(c) illustrates the modified graph model for the system shown in Fig. 4(a). In this example, at most only one latch will appear at each edge.

For an edge $u \xrightarrow{e} v$, retiming sets the number of latches at edge e to $w(e) - r(u) + r(v)$, which must be a nonnegative integer. These retiming constraints are expressed as follows:

$$w(e) = r(u) - r(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V_{fo1} \cup V_l \tag{15}$$

$$w(e) \geq r(u) - r(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \tag{16}$$

$$w(e) \leq r(u) - r(v) + 1 \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \tag{17}$$

As described in Section 4.1, node $v \in V_{fo1}$ is used to remove the restriction of applying the same latching time to the latches on every fanout edge of node u, and node $v \in V_l$ is used to subtract the amount of the shift of time origin with the chord. Therefore, by definition, the number of latches at the fan-in edge of node $v \in V_{fo1} \cup V_l$ is set to 0 as shown in constraint (15). After retiming, constraints (16) and (17) restrict the number of latches at an edge to either zero or one. For some special cases such as $w(e(7,3))$ in Fig. 4(a), the edge weight can be set accordingly.

Some constraints formulated in the proceeding section will be modified by including these new integer variables $r(v)$. Thus, the constraint set becomes as follows:

delay constraints:

$$s(u) + t_{max}(v) \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) \tag{18}$$

$$b(v) \leq b(u) + t_{min}(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) \tag{19}$$

synchronization constraints:

$$s(u) + t_{max}(v) + t_{dl} \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) + 1 \tag{20}$$

$$e(u) + t_{max}(v) + t_{cl} \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) + 1 \tag{21}$$

$$b(v) \leq e(u) + t_{min}(v) + t_{cs} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) + 1 \tag{22}$$

latching constraints:

$$s(u) \leq e(u) + c_h - t_s \qquad \text{for } u \xrightarrow{e} v \text{ and } w(e) = r(u) - r(v) + 1 \tag{23}$$

$$e(u) + c_h + t_h \leq b(u) + c \qquad \text{for } u \xrightarrow{e} v \text{ and } w(e) = r(u) - r(v) + 1 \tag{24}$$

Since the constraint set contains qualifying clauses, we are going to develop a transformation to remove these clauses. In order to make the transformation to be a mixed integer LP, two constraints are added to the delay constraint set, and then the qualifying clauses in constraints (23) and (24) are removed. Lemma 1 shows the conditions that preserve the solution set of the clock period. Note that variable $e(u)$ appears only in the original constraint set when $u \xrightarrow{e} v$ and $w(e) = r(u) - r(v) + 1$. With the additional constraints, $e(u)$ is defined for each node $u \in V \cup V_{fo1} \cup V_{fo2} \cup V_l$.

delay constraints:

$$e(u) + t_{max}(v) \leq s(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) \tag{25}$$

$$b(v) \leq e(u) + t_{min}(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \text{ and } w(e) = r(u) - r(v) \tag{26}$$

latching constraints:

$$s(u) \le e(u) + c_h - t_s \qquad u \in V \cup V_{fo1} \cup V_{fo2} \cup V_l \tag{27}$$

$$e(u) + c_h + t_h \le b(u) + c \qquad u \in V \cup V_{fo1} \cup V_{fo2} \cup V_l \tag{28}$$

Lemma 1 *For the new constraint set as described above, if $c_h \ge t_s$ and $c - c_h \ge t_h$, the solution space of clock period c is identical to the original one.*

The other qualifying clauses can be removed by appropriate transformation described in the following theorem.

Theorem 1 *A pipelined system is defined by G. If G has a multiplexing degree of m, $c_h \ge t_s$, $c - c_h \ge t_h$, and $c_h + t_h \ge t_{cs}$, then G operates correctly at clock period c if and only if there exists a feasible solution for the following mixed integer LP constraints, where R, T, X, c_h are real variables and r is an integer variable.*

retiming constraints: same as constraints (15), (16), and (17)

pulse width constraints: same as stated in Section 4.2

delay constraints:

$$R(u) - R(v) + r(u) - r(v) \le -\frac{t_{max}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \tag{29}$$

$$X(u) - R(v) + r(u) - r(v) \le -\frac{t_{max}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \tag{30}$$

$$T(v) - T(u) \le w(e) + \frac{t_{min}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \tag{31}$$

$$T(v) - X(u) \le w(e) + \frac{t_{min}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \tag{32}$$

synchronization constraints:

$$R(u) - R(v) \le -\frac{t_{max}(v)}{c} - \frac{t_{dl}}{c} - w(e) + 1 \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \tag{33}$$

$$X(u) - R(v) \le -\frac{t_{max}(v)}{c} - \frac{t_{cl}}{c} - w(e) + 1 \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \tag{34}$$

$$T(v) - X(u) - r(u) + r(v) \le \frac{t_{min}(v)}{c} + \frac{t_{cs}}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \tag{35}$$

loop/semi-loop constraints:

$$R(v) - R(u) = -m \cdot os(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V_l \tag{36}$$

$$T(v) - T(u) = -m \cdot os(v) \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V_l \tag{37}$$

latching constraints:

$$R(u) - X(u) - \frac{c_h}{c} \leq -\frac{t_s}{c} \qquad \textit{for } u \in V \cup V_{fo1} \cup V_{fo2} \cup V_l \tag{38}$$

$$X(u) - T(u) + \frac{c_h}{c} \leq -\frac{t_h}{c} + 1 \qquad \textit{for } u \in V \cup V_{fo1} \cup V_{fo2} \cup V_l \tag{39}$$

$$R(v) - T(v) \leq -\frac{t_s}{c} - \frac{t_h}{c} + 1 \qquad \textit{for } v \in V_{po} \tag{40}$$

initiation constraint:

$$T(v) - R(v) \leq 0 \qquad v \in V_{pi} \tag{41}$$

temporal equivalence constraint: optional

$$R(v) - R(u) = -\frac{t_s}{c} + (k-1) \cdot m \qquad u \in V_{pi} \textit{ and } v \in V_{po} \tag{42}$$

Proof: These mixed integer LP constraints can be obtained by substituting $c(R(v) + r(v))$ for $s(v)$, $c(T(v) + r(v))$ for $b(v)$, and $c(X(v) + r(v))$ for $e(v)$. The details will be given in appendix □

The above theorem is only valid if $c_h \geq t_s$, $c - c_h \geq t_h$, and $c_h + t_h \geq t_{cs}$. Also, these conditions are implied by $w_l \geq t_s$, $w_l \geq t_h$, and $w_l + t_h \geq t_{cs}$, which are true for most practical designs.

6 Clock Optimization via Resynchronization

With the insertion of latches, the clock period can be reduced by (1) introducing pipelining along the longest paths, or (2) providing synchronization on the shortest paths of a design. The first reason is the basis for conventional pipelining in a combinational logic. The latter one is called resynchronization and was not studied vigorously. As shown in this section, the optimal resynchronization is also a mixed integer LP problem. Thus our framework allows concurrent optimizations including resynchronization.

6.1 Resynchronization Example

As shown in Fig. 3(a), there are n latches around the loop l and the shift of time origin around the loop is denoted by $os(l)$. Let $T_{max}(i)$ and $T_{min}(i)$ define the longest and shortest propagation delays of stage i, where $1 \leq i \leq n$. For a circular loop, the following theorem can be established:

Theorem 2 *For a circular loop l, if $(n - m \cdot os(l))c < n(c_h + t_h - t_{cs}) - \sum_{i=1}^{n} T_{min}(i)$, then there exists no feasible solution for the loop.*

From the above theorem, it is clear that when $m = 1$ and $n = os(l)$, the width of the active period , c_h, must be smaller than or equal to $t_{cs} - t_h + \frac{1}{n}\sum_{i=1}^{n} T_{min}(i)$, which is independent of the clock period. So if $w_l > t_{cs} - t_h + \frac{1}{n}\sum_{i=1}^{n} T_{min}(i)$, then there exists no feasible solution. However, such a situation can be remedied by resynchronization. As an example shown in Fig. 5(a), if w_l is increased to $1.5ns$, due to hold time violation, no feasible solution can be obtained even by concurrent application of intentional skew, wave pipelining, and retiming. However, by inserting a latch as shown in Fig. 5(c), the optimal clock period becomes $5ns$.

6.2 Mixed Integer LP Formulation of Resynchronization

For an edge $u \xrightarrow{e} v$, $W(e(u,v))$ denotes whether a latch exists along this edge. To allow more than one latch at an edge, an additional node $v \in V_{fo2}$ can be inserted along the edge. The LP constraints still hold except that these qualifying clauses are now expressed with new variables $W(e(u,v))$. Again, by an appropriate transformation, the optimization process can be formulated as a mixed integer LP problem. In the optimization process, the total number of latches can be minimized.

Theorem 3 *A pipelined system is defined by G. If G has a multiplexing degree of m, $c_h \geq t_s$, $c - c_h \geq t_h$, and $c_h + t_h \geq t_{cs}$, then G operates correctly at clock period c if and only if there exists a feasible solution for the following mixed integer LP constraints, where R, T, X, c_h are real variables and W is a binary integer variable.*

pulse width constraints: same as stated in Section 4.2

delay constraints:

$$R(u) - R(v) \leq -\frac{t_{max}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \quad (43)$$

$$X(u) - R(v) \leq -\frac{t_{max}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \quad (44)$$

$$T(v) - T(u) - W(e(u,v)) \leq \frac{t_{min}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \quad (45)$$

$$T(v) - X(u) - W(e(u,v)) \leq \frac{t_{min}(v)}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo1} \cup V_{fo2} \cup V_{po} \quad (46)$$

synchronization constraints:

$$R(u) - R(v) + W(e(u,v)) \leq -\frac{t_{max}(v)}{c} - \frac{t_{dl}}{c} + 1 \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \ (47)$$

$$X(u) - R(v) + W(e(u,v)) \leq -\frac{t_{max}(v)}{c} - \frac{t_{cl}}{c} + 1 \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \ (48)$$

$$T(v) - X(u) \leq \frac{t_{min}(v)}{c} + \frac{t_{cs}}{c} \qquad \text{for } u \xrightarrow{e} v \text{ and } v \in V \cup V_{fo2} \cup V_{po} \ (49)$$

Constraints of loop/semi-loop, latching, initiation, and temporal equivalence are the same as stated Theorem 1.

Proof: The substitution rules are $cR(v)$ for $s(v)$, $cT(v)$ for $b(v)$, and $cX(v)$ for $e(v)$. $\square$

7 Design Robustness

Due to process and environmental variations, design parameters would deviate from the specifications. Tolerance to these variations can be quantitatively measured by safety margin, δ, by which the clock occurrence is allowed to deviate from the design specification time, t. In other words, the system still operates correctly as long as the clock is delivered in the interval of $[t - \delta,\ t + \delta]$. For a given clock period c, this amount can be maximized by incorporating the variable δ into the latching constraint set as follows:

latching constraints:

$$s(u) + 2\delta \leq e(u) + c_h - t_s \quad \text{for } u \xrightarrow{e} v \text{ and } w(e) = 1 \tag{50}$$

$$e(u) + c_h + t_h + 2\delta \leq b(u) + c \quad \text{for } u \xrightarrow{e} v \text{ and } w(e) = 1 \tag{51}$$

$$s(v) + t_s + t_h + 4\delta \leq b(v) + c \quad \text{for } v \in V_{po} \tag{52}$$

Constraints (50) and (51) are used to provide the safety margin of δ against the setup and hold time violation of a latch. The latching time of the external output register now becomes $s(u) + t_s + 2\delta$. The additional amount of 2δ is needed to prevent zero clocking in the presence of process and environmental variations. And constraint (52) prevents double clocking by the safety margin of δ

Assume that there are n latches around the loop l as shown in Fig. 3(a), and the shift of time origin around the loop is denoted by $os(l)$. An upper bound of the safety margin can then be calculated . Let $T_{max}(i)$ and $T_{min}(i)$ are defined as the longest and shortest path delays of each stage i where $1 \leq i \leq n$.

Theorem 4 *A circular loop l is operated at a clock period c and a multiplexing degree m. The safety margin of δ is limited by* $\frac{1}{2}(\frac{m \cdot os(l) \cdot c}{n} - \frac{\sum_{i=1}^{n} T_{max}(i)}{n} + c_h - t_s - t_{cl})$, $\frac{1}{2}((1 - \frac{m \cdot os(l)}{n})c + \frac{\sum_{i=1}^{n} T_{min}(i)}{n} + t_{cs} - c_h - t_h)$, *and* $\min_i\{\frac{1}{4}(c - T_{max}(i) + T_{min}(i) + t_{cs} - t_s - t_h - t_{cl})\}$.

Theorem 4 tells that the upper bound for the safety margin can be raised by increasing the system clock period, prolonging the short paths, and/or reducing the long paths of a design.

8 Feasible Clock Period

For a given clock period c, the mixed integer LP constraints can be tested if c is a feasible solution. In a single closed loop l whose shift of time origin is $os(l)$, the longest path delay around the loop is $T_{max}(l)$. Then the feasible clock period is bounded below by $\max_{\forall\ l} \frac{T_{max}(l)}{m \cdot os(l)}$ for a multiplexing degree m [19]. This serves as a starting point for searching for an optimal clock period. But what is the upper bound of the clock period?

8.1 Upper Bound of Clock Period

For the semi-loop as shown in Fig. 3(b) where m is the multiplexing degree, edge $e(q', t)$ is selected as a chord, whose associated shift of time origin, denoted by $os(q')$, is assumed to be smaller than or equal to 0. And there are p latches along the top path from host to node t (excluding the external register). The following lemma can be established.

Lemma 2 *If $p < -m \cdot os(q') - 1$, then the feasible clock period c is bounded above. Otherwise, if there exists at least one feasible solution, the feasible clock period is unbounded above (i.e. upper bound is infinity).*

Lemma 3 *For a feedback loop as shown in Fig. 3(a), the shift of time origin of the loop is denoted by $os(l)$ and the number of latches around the loop is n. If $n < m \cdot os(l)$, the feasible clock period is bounded above. On the other hand, if there exists a feasible solution, the feasible clock period is unbounded above.*

Note that if temporal equivalence constraint is required to be satisfied, node $v \in V_{po}$ and node $u \in V_{pi}$ should be seen as a feedback loop.

Theorem 5 *If, for every semi-loop of G, $p \geq -m \cdot os(q') - 1$, for every feedback loop of G, $n \geq m \cdot os(l)$, and there exists at least one feasible solution, the feasible clock period is unbounded above.*

8.2 Safety Margin

As stated in Section 7, safety margin will saturate finally if the feasible clock period is bounded above. There are other situations that will lead to a saturated safety margin as described in the following theorem.

Theorem 6 *If, for any semi-loop, $p = -m \cdot os(q') - 1$, or for any feedback loop, $n = m \cdot os(l)$, the safety margin of G is bounded above by a constant value, which is independent of the clock period.*

For a graph G and its spanning tree T, node v is called a *joint node* if node v has more than one fan-in or fanout edge. Along path p_i in G, two joint nodes are *neighboring* if there exists no other joint node between them. Fig. 4(c) shows an example, in which nodes 2, 3, 4, 5, 6, 7 are joint nodes and others are not. Nodes 2 and 3 are neighboring, while nodes 2 and 4 are not. For two joint nodes u and v which are neighboring along a path p_i, variable $\chi(p_i)$, which must be a nonnegative integer, is associated with path p_i. If every node and edge in path p_i is also in path p, then p_i is said to belong to path p, denoted by $p_i \in p$.

Theorem 7 *For a graph G and its spanning tree T, if the safety margin of G exempts from saturation as the clock period is increased, the following three constraints (53), (54), and (55) will be satisfied. For each node $v_l \in V_l$, and the corresponding semi-loop $u \xrightarrow{p} v$ and $u \xrightarrow{q} v$ derived by T, if v_l is in path q, the following constraint is formed.*

$$\sum_{p_i \in p} \chi(p_i) - \sum_{p_i \in q} \chi(p_i) = -m \cdot os(v_l) \tag{53}$$

For each node $v_l \in V_l$, and the corresponding feedback loop l derived by T, the following constraint is formed.

$$\sum_{p_i \in l} \chi(p_i) = m \cdot os(v_l) + 1 \tag{54}$$

For two neighboring joint nodes u and v along path p_i, the following constraint is formed.

$$\sum_{e(u_i,v_i) \in p_i} W(e(u_i, v_i)) \geq \chi(p_i) \tag{55}$$

Proof: Variable $\chi(p_i)$ can be thought as the least number of latches at path p_i to satisfy the requirement on the minimum number of latches at each edge as described in Theorems 5 and 6. □

Further research is needed to prove that the constraints described in Theorem 7 are also sufficient conditions.

9 Design Examples

Our formulation has been applied to several design examples. The first two introduce wave pipelining in latch-based designs. A binary search is employed to find the optimal clock period when retiming or

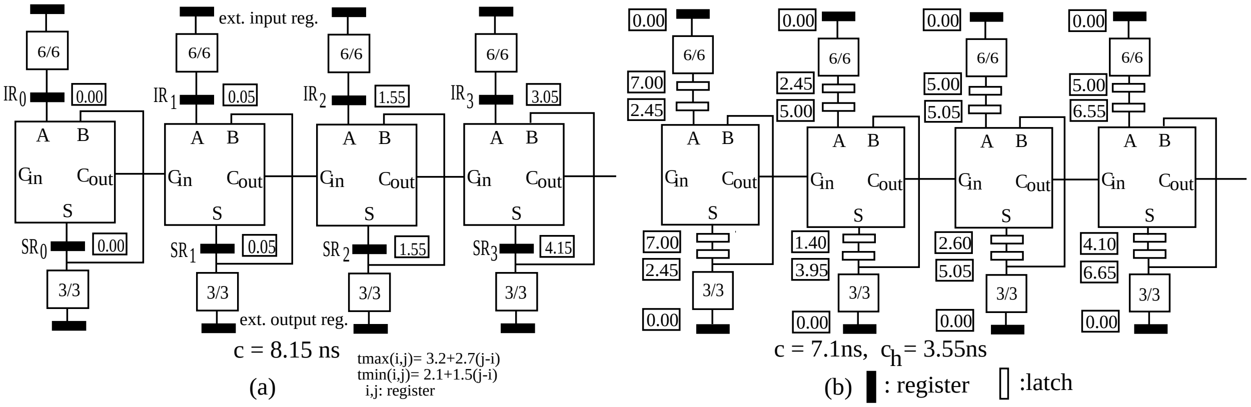


Figure 6: Adder Design: (a) Edge-Triggered Design (b) Latch-Based Design $t_s = t_h = 1ns$,$w_l = 3.5ns$

w_l	0.0	1.0	1.5	2.0	2.5	3.0	3.5
# of latches	7	7	11	11	12	12	12

Table 3: Minimal Number of Latches v.s. Minimal Allowable Pulse Width

resynchronization is applied. Once the constraint set is captured, a mixed integer linear programming solver, LINDO, is used. For simplicity, the propagation delay of a latch or register is set to 0 in these examples.

Example 1: This example is a ripple adder originally studied by Fishburn [4]. As shown in Fig. 6(a), Fishburn reported an optimum clock period at $8.15ns$ by having clock skews in internal registers and restricting zero skew in external registers. If each register is replaced by two latches, the optimum clock period is reduced to $7.1ns$ as shown in Fig. 6(b). During the optimization process, the temporal equivalence constraint is required to be satisfied. And the longest and shortest propagation delays between a pair of registers are respectively estimated as $t_{max}(i,j) = 3.2 + 2.7(j - i)$ and $t_{min}(i,j) = 2.1 + 1.5(j - i)$, where $j \geq i$. At the clock period $7.1ns$, the minimum number of latches (*i.e.* the latches replaced registers IR and SR), versus minimum allowable pulse width w_l is shown in Table 3. It can be seen that as w_l increases, the required minimum number of latches is also increased.

Example 2: This example consists of two circular loops in [22, 23]. The resulting design with intentional skews is shown in Fig. 7 and in the fourth column of Table 4. The first three columns of

	restricted single phase	single phase	coincident multi-phase	non-zero skew
d1	18ns	-	14ns	12ns
d2	-	14ns	-	13ns

Table 4: Optimal Clock Period for Different Clocking Schemes (d1: ICCD design, d2: ICCAD design.)

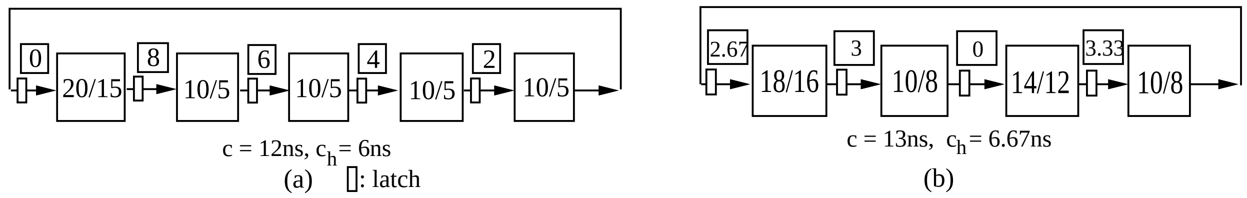


Figure 7: Circular Loop Designs: (a)ICCD design: $t_s = 2ns$, $t_h = 1ns$, $w_l = 5ns$ (b)ICCAD design: $t_s = t_h = 2ns$, $w_l = 5ns$

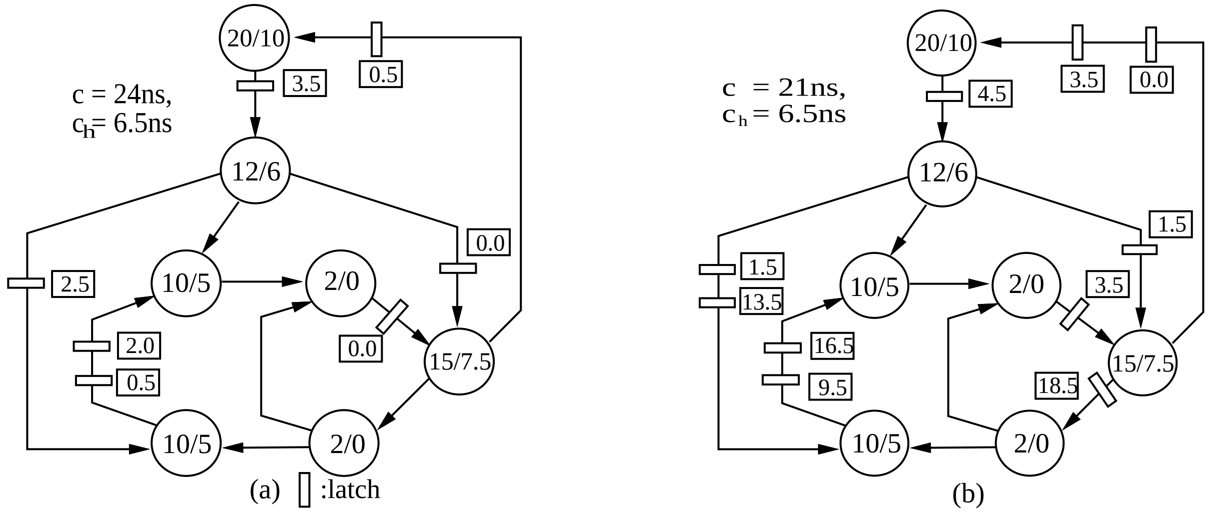


Figure 8: Retiming or Resynchronization of RISC: $w_l = 6.5ns$ (a) $t_s = t_h = 0.5ns$ (b) $t_s = t_h = 2.5ns$

Table 4 are derived by works [22, 23]. Restricted single phase designs, as shown in the first column, use the adopted model, while SMO model is used in single phase designs as shown in the second column. Coincident multi-phase designs set the latching edge for each phase at the same time but different active period width for each phase. There are 5 waves of data around the loop in Fig. 7(a) and 4 waves of data in Fig. 7(b).

Example3: The final example is shown in Fig. 4(b). By retiming, the optimal clock period is $24ns$ as shown in Fig. 8(a). While increasing both of the setup and hold times to $2.5ns$, no feasible solution can be obtained by using retiming. If at most two latches are allowed at each edge, resynchronization can reduce the clock period to $21ns$ as shown in Fig. 8(b). Constraints (53), (54), and (55) are incorporated into the constraint set when resynchronization.

10 Conclusion

In this paper, we present a LP formulation for the latch-based designs such that timing optimization is achieved by concurrently applying both intentional clock skew and wave pipelining. The formulation becomes a mixed integer LP if retiming or resynchronization is included. By checking the number of latches at each edge, the solution set of the clock period can be tested if it is unbounded above or not. Constraints are then added to provide the minimum number of latches at each edge in order to meet the criteria described in Theorem 7. Further research is needed to investigate whether the safety margin will exempt from saturation as the clock period is increased if the constraints described in Theorem 7 are satisfied, and to unveil a more efficient algorithm for solving this special type of mixed integer LP problem.

Appendix

Proof of Lemma 1: Because the new constraints are more stringent than the original ones, the clock period found as a solution of the new ones must be a feasible clock period of the original ones. The remaining question is that: if a feasible solution exists for the original formulation, can these undecided variables $e(u)$ be assigned such that these new constraints are satisfied? It is worth noting that if the values of all the undecided variables $e(u)$ are assigned such that $b(u) \leq e(u) \leq s(u)$, constraints (25) and (26) can be satisfied. For the following situation: $w(e_1) = r(u) - r(v)$ where $u \overset{e_1}{\rightarrow} v$, and $w(e_2) = r(v) - r(w) + 1$ where $v \overset{e_2}{\rightarrow} w$, consider three different cases: $e(v) > s(v)$, $e(v) < s(v) - s(u) + b(u)$, and $s(v) - s(u) + b(u) \leq e(v) \leq s(v)$. If $e(v) > s(v)$, the value of variable $e(u)$ is set to the value of $s(u)$ and constraints (27) and (28) can be satisfied when $c_h \geq t_s$. If $e(v) < s(v) - s(u) + b(u)$, the value of variable $e(u)$ is set to the value of $b(u)$ and constraints (27) and (28) can be satisfied when $c - c_h \geq t_h$. For the case $s(v) - s(u) + b(u) \leq e(v) \leq s(v)$, the value of variable $e(u)$ is set to the value of $s(u) - s(v) + e(v)$, and constraints (27) and (28) can be satisfied. This illustration does not lose its generality. In this way, the value of all the undecided variables $e(u)$ can be assigned such that the new constraints are satisfied. □

Proof of Theorem 1: Due to similarities between all the constraints, we only prove constraints (29) and (31). If constraints for the qualified LP formulation can be satisfied by the values of the variables s, b, e, r, and c_h, then the corresponding constraints for the mixed integer linear formulation can be satisfied by the values of the variables R, T, X, r, and c_h obtained by the given transformation rules.

Constraint (18) ⇒ Constraint (29):

$$\begin{aligned} & R(u) - R(v) + r(u) - r(v) + \tfrac{t_{max}(v)}{c} \\ = & \tfrac{s(u)}{c} - r(u) - \tfrac{s(v)}{c} + r(v) + r(u) - r(v) + \tfrac{t_{max}(v)}{c} \\ = & \tfrac{1}{c}(s(u) - s(v) + t_{max}(v)) \leq 0 \end{aligned}$$

The final inequality is valid and independent of $w(e) = 0$ or $w(e) = 1$.

Constraint (18) ⇐ Constraint (29):

$$\begin{aligned} & s(u) - s(v) + t_{max}(v) \\ = & c(R(u) + r(u)) - c(R(v) + r(v)) + t_{max}(v) \\ = & c(R(u) - R(v) + r(u) - r(v) + \tfrac{t_{max}(v)}{c}) \leq 0 \end{aligned}$$

Constraint (19) $\Rightarrow$ Constraint (31):

$$\begin{aligned}
& T(v) - T(u) - \tfrac{t_{min}(v)}{c} - w(e) \\
= \quad & \tfrac{b(v)}{c} - r(v) - \tfrac{b(u)}{c} + r(u) - \tfrac{t_{min}(v)}{c} - w(e) \\
= \quad & \tfrac{1}{c}(b(v) - b(u) - t_{min}(v)) - (w(e) - r(u) + r(v))
\end{aligned}$$

When $w(e) = r(u) - r(v)$, (*i.e.* there is no latch at edge e), expression $b(v) - b(u) - t_{min}(v)$ is smaller than or equal to 0. The above expression must therefore be smaller than or equal to 0. If $w(e) = r(u) - r(v) + 1$, (*i.e.* there exists one latch at edge e), expression $\frac{1}{c}(b(v) - b(u) - t_{min}(v))$ is smaller than or equal to 1 when $c_h + t_h \geq t_{cs}$. This can be shown as follows:

$$\begin{aligned}
& b(v) - b(u) - t_{min}(v) \\
\leq \quad & e(u) + t_{cs} + t_{min}(v) - b(u) - t_{min}(v) \\
\leq \quad & e(u) - b(u) + t_{cs} \\
\leq \quad & b(u) + c - c_h - t_h - b(u) + t_{cs} \\
\leq \quad & c - c_h - t_h + t_{cs}
\end{aligned}$$

The second inequality is derived from constraint (22) and the fourth one is from constraint (28). Since expression $b(v) - b(u) - t_{min}(v)$ is smaller than or equal to c when $c_h + t_h \geq t_{cs}$, expression $\frac{1}{c}(b(v) - b(u) - t_{min}(v))$ is smaller than or equal to 1. And $-(w(e) - r(u) + r(v))$ is equal to -1. Expression $T(v) - T(u) - \frac{t_{min}(v)}{c} - w(e)$ is consequently smaller than or equal to 0 and independent of $w(e) = 0$ or 1.

Constraint (19) $\Leftarrow$ Constraint (31):

$$\begin{aligned}
& b(v) - b(u) - t_{min}(v) \\
= \quad & c(T(v) + r(v)) - c(T(u) + r(u)) - t_{min}(v) \\
= \quad & c(T(v) - T(u) - \tfrac{t_{min}(v)}{c} - r(u) + r(v))
\end{aligned}$$

When $w(e) = r(u) - r(v)$, the final expression becomes $c(T(v) - T(u) - \frac{t_{min}(v)}{c} - w(e))$, which must be smaller or equal to 0. □

Proof of Theorem 2: The edge pointed from stage n to stage 1 is selected as a chord. Constraints (7) and (11) for such a spanning tree can be formulated as follows:

$$\begin{aligned}
b(i+1) \leq e(i) + T_{min}(i+1) + t_{cs} \qquad & 1 \leq i \leq n-1 \\
e(i+1) + c_h + t_h \leq b(i+1) + c \qquad & 0 \leq i \leq n-1
\end{aligned}$$

Loop/semi-loop constraint and constraint (7) for the chord can be written together as follows.

$$b(1) \leq e(n) + T_{min}(1) + t_{cs} - m \cdot os(l) \cdot c$$

Summation of the above constraints will lead to the necessary condition for a feasible solution: $(n - m \cdot os(l))c \geq n(c_h + t_h - t_{cs}) - \sum_{i=1}^{n} T_{min}(i)$. □

Proof of Theorem 3: It can be proven similarly as Theorem 1. □

Proof of Theorem 4: The edge pointed from stage n to stage 1 is selected as a chord. Constraints (6) and (50) for such a spanning tree can be formulated as follows:

$$\begin{aligned} e(i) + T_{max}(i+1) + t_{cl} \leq s(i+1) &\qquad 1 \leq i \leq n-1 \\ s(i) + 2\delta \leq e(i) + c_h - t_s &\qquad 1 \leq i \leq n \end{aligned}$$

Loop/semi-loop constraints and constraint (6) for the chord can be written together as follows:

$$e(n) - m \cdot os(l) \cdot c + T_{max}(1) + t_{cl} \leq s(1)$$

Summing all these inequalities leads to the first bound value.

Constraints (7) and (51) for the spanning tree can be written as follows:

$$\begin{aligned} b(i+1) \leq e(i) + T_{min}(i+1) + t_{cs} &\qquad 1 \leq i \leq n-1 \\ e(i) + c_h + t_h + 2\delta \leq b(i) + c &\qquad 1 \leq i \leq n \end{aligned}$$

Loop/semi-loop constraints and constraint (7) for the chord can be formed together as follows.

$$b(1) \quad \leq \quad e(n) - m \cdot os(l) \cdot c + T_{min}(1) + t_{cs}$$

Again summation of the left and right sides of these inequalities gives the second bound value.

For each stage i, summation of constraints (6), (7), (50), and (51) can give the final bound value. □.

Proof of Lemma 2: Node 0 denotes the host. Constraints (4) and (11) for node 1 can be expressed as follows:

$$b(1) \leq b(0) + t_{min}(1)$$

$$e(1) + c_h + t_h \leq b(1) + c$$

Constraints (7) and (11) for the other nodes along the top path can be formulated as follows:

$$b(i) \leq e(i-1) + t_{min}(i) + t_{cs} \qquad 2 \leq i \leq p$$

$$e(i) + c_h + t_h \leq b(i) + c \qquad 2 \leq i \leq p$$

For node t, constraints formed along edge $e(p,t)$ can be written similarly.

$$b(t) \leq e(p) + t_{min}(t) + t_{cs}$$

$$e(t) + c_h + t_h \leq b(t) + c$$

Constraints (3) and (10) for node $1'$ can be expressed as follows:

$$s(1') \geq s(0) + t_{max}(1')$$

$$e(1') + c_h - t_s \geq s(1')$$

Constraints (6) and (10) for the other nodes along the bottom path can be formulated similarly.

$$s(i') \geq e((i-1)') + t_{max}(i') + t_{cl} \qquad 2 \leq i \leq q$$

$$e(i') + c_h - t_s \geq s(i') \qquad 2 \leq i \leq q$$

Constraint (6), constraint (10), and loop/semi-loop constraints for node t along edge $e(q',t)$ can be written together as follows:

$$s(t) \geq e(q') + t_{max}(t) + t_{cl} - m \cdot os(q')$$

$$e(t) + c_h - t_s \geq s(t)$$

By multiplying the constraints formed along the bottom path with -1 and summing the constraints formed along the top path and initial constraint, the following constraint can be obtained. Let T_{max} denotes the sum of $\sum_{i=1}^{q} t_{max}(i')$ and $t_{max}(t)$. And T_{min} denotes the sum of $\sum_{i=1}^{p} t_{min}(i)$ and $t_{min}(t)$.

$$(-m \cdot os(q') - (p+1))c \quad \leq \quad T_{min} - T_{max} + p \cdot t_{cs} - (p+1)(c_h + t_h) - q \cdot t_{cl} + (q+1)(c_h - t_s)$$

Note that the value of the right side of this inequality is constant. As c approaches infinity, this inequality will not hold if $p < -m \cdot os(q') - 1$, which makes the feasible clock period bounded above.

Assume that a feasible solution exists for $c = c_f$. If $p \geq -m \cdot os(q') - 1$, a feasible solution is constructed below for the clock period of $2 \cdot c_f$. Similarly, the clock period of $4 \cdot c_f$, $8 \cdot c_f$, etc., can be proven feasible. So it is unbounded above. Consider first the case $p \geq -m \cdot os(q')$: Choose any $-m \cdot os(q')$ latches from the latches along the top path, and mark them from 1 to $-m \cdot os(q')$ sequentially. For the i-th marked latch and the latches between the i-th and $(i+1)$-th marked latches, the time for the enabling edge is increased by the amount of $i \cdot c_f$. For the nodes between the i-th and $(i+1)$-th marked latches, the values of their variables s and b are also increased by the amount of $i \cdot c_f$. For the nodes after the $(-m \cdot os(q'))$-th marked latch, the values of variables s, b, and e are all increased by the amount of $-m \cdot os(q') \cdot c_f$. The values of the other nodes' variables are not changed. The solution obtained will be a feasible one. For the case that $p = -m \cdot os(q') - 1$, the latch which is at the fanout edge of node t is selected as the $(-m \cdot os(q'))$-th marked latch. Except that the value of variable $s(t)$ should be increased by the amount of $-m \cdot os(q') \cdot c_f$, the construction method described above can lead to a feasible solution for $c = 2c_f$. □

Proof of Lemma 3: Due to similarity with the proof of Lemma 2, it is omitted here. □

Proof of Theorem 5: For a given spanning tree, the number of marked latches are selected accordingly. The construction method as described in the proof of Lemma 2 can be used to obtain a feasible solution. □

Proof of Theorem 6: As shown in the Theorem 4, the second bound value will be a constant if $n = -m \cdot os(l)$. So the safety margin will saturate finally. Similarly, if $p = -m \cdot os(q') - 1$ for any semi-loop, it can be shown that there also exists a constant bound value, which is independent of clock period. □

References

[1] S. Malik, E. M. Sentovich, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Retiming and Resynthesis: Optimizing Sequential Networks with Combinational Techniques," *IEEE Trans. on Computer-Aided Design,* vol. 10, pp. 74-84, Jan. 1991.

[2] H. Hsieh, W. Liu, and R. Cavin, "Integrated Parametric Timing Optimization of Digital Systems," submitted to *IEEE Trans. on Computer-Aided Design,* Sept. 1994.

[3] H. Hsieh, W. Liu, and R. Cavin, "Concurrent Timing Optimization of Latch-Based Digital Systems," *Technical Report*, Department of Electrical and Computer Engineering, NCSU, Oct. 1994.

[4] J. P. Fishburn, "Clock Skew Optimization," *IEEE Trans. on Computers,* vol. 39, pp. 945-951, July, 1990.

[5] C. E. Leiserson and F. M. Rose, "Optimization Synchronous Circuitry by Retiming," *Proceedings of the Third Caltech Conference on VLSI,* pp. 87-116, 1983.

[6] S. H. Unger and C. Tan, "Clocking Schemes for High-Speed Digital Systems," *IEEE Trans. on Computers,* vol. C-35, pp. 880-895, Oct. 1986.

[7] M. R. Dagenais and N. C. Rumin, "On the Calculation of Optimal Clocking Parameters in Synchronous Circuits with Level-Sensitive Latches," *IEEE Trans. on Computer-Aided Design,* vol. 8, pp. 268-278, Mar. 1989.

[8] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun, "Analysis and Design of Latch-Controlled Synchronous Digital Circuits," *IEEE Trans. on Computer-Aided Design,* vol. 11, pp. 322-333, Mar. 1992.

[9] T. G. Szymanski, "Computing Optimal Clock Schedules," *Proceedings of the IEEE/ACM Design Automation Conference,* pp. 399-404, 1992.

[10] T. G. Szymanski and N. Shenoy, "Verifying Clock Schedules," *Proceedings of the International Conference on Computer-Aided Design*, pp. 124-131, 1992.

[11] N. Shenoy and R. K. Brayton, "Graph Algorithms for Clock Schedule Optimization," *Proceedings of the International Conference on Computer-Aided Design*, pp. 132-136, 1992.

[12] N. Shenoy, K. Singh, R. K. Brayton, and A. Sangiovanni-Vincentelli, "On the Temporal Equivalence of Sequential Circuits", *Proceedings of the IEEE/ACM Design Automation Conference,* pp. 405-409, 1992.

[13] A. T. Ishii and C. E. Leiserson, "Optimizing Two-Phase Level-Clocked Circuitry," *Advanced Research in VLSI and Parallel Systems: Proceedings of the 1992 Brown/MIT Conference,* pp. 239-264, 1992

[14] B. Lockyear and C. Ebeling, "The Practical Application of Retiming to the Design of High-Performance Systems," *Proceedings of the ICCAD*, pp. 288-295, 1993

[15] N. Shenoy and R. K. Brayton, "Retiming of Circuits with Single Phase Transparent Latches," *International Conference on Computer Design,* pp. 86-89, 1991.

[16] D. Wong, G. De Micheli, and M. J. Flynn, "Design High-Performance Digital Circuits Using Wave Pipelining: Algorithms and Practical Experiences," *IEEE Trans. on Computer-Aided Design,* vol. 12, pp. 25-46, Jan. 1993.

[17] D. A. Joy and M. J. Ciesielski, "Clock Period Minimization with Wave Pipelining," *IEEE Trans. on Computer-Aided Design,* vol. 12, pp. 461-472, Apr. 1993.

[18] B. C. Ekroot, "Optimization of Pipelined Processors by Insertion of Combinational Logic Delay," Ph.D. dissertation, Stanford University, Sept. 1987.

[19] C. T. Gray, W. Liu, and R. Cavin III, Wave Pipelining: Theory and CMOS Implementations, Kluwer Academic Publisher, Oct. 1993.

[20] W. K. C. Lam, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Valid Clocking in Wave Pipelined Circuits," *Proceedings of the ICCAD*, Nov. 1992.

[21] K. A. Sakallah, T. N. Mudge, T. M. Burks, and E. S. Davidson, "Synchronization of Pipelines," *IEEE Trans. on Computer-Aided Design,* vol. 12, pp. 1132-1146, Aug. 1993.

[22] K. A. Sakallah, T. N. Mudge, T. M. Burks, and E. S. Davidson, "Optimal Clocking of Circular Pipelines," *Proceedings of the ICCD*, 1992.

[23] C. Chang, E. S. Davidson, and K. A. Sakallah, "Using Constraint Geometry to determine Maximum Rate Pipeline Clocking," *Proceedings of the International Conference on Computer-Aided Design*, pp. 142-148 Nov. 1992.