

Performance and Reliability Driven Clock Scheduling of Sequential Logic Circuits

Atsushi TAKAHASHI and Yoji KAJITANI

Department of Electrical and Electronic Engineering
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152, Japan
Tel: +81-3-5734-2565 Fax: +81-3-5734-2902
E-mail: {atushi,kajitani}@ss.titech.ac.jp

Abstract— It is known that the clock-period in a sequential circuit can be shorter than the maximum signal delay between registers if the clock arrival time to each register is controlled. We propose an algorithm to find the minimum clock-period of a circuit whose signal propagation delays are given. Experimental results on LGSynth93 benchmarks show that this technique achieves as much as about 16% reduction of clock-period compared with the conventional maximum signal delay based methods. An application of this technique to improve the reliability of circuits is considered.

I. INTRODUCTION

In the design of a sequential circuit with globally clocked registers, there are various techniques to achieve a shorter clock-period [1, 2, 3, 4, 5, 6, 9, 13, 15].

A signal starts to propagate from a register when the register is ticked by a clock-edge. It must arrive at the next register before it is ticked by the next clock-edge. If every register in the circuit is ticked exactly simultaneously, the maximum signal propagation delay along functional elements (wires, transistors, etc.) from a register to another register is a lower bound of the clock-period. The purpose of retiming and performance driven layout is to reduce this delay in the circuit. Retiming investigated in [9, 12] relocates the registers of a given circuit while preserving its functionality. In a unit delay circuit, Papaefthymiou [12] gave an exact characterization of the minimum clock-period that can be achieved by applying a retiming technique in terms of the minimum cycle-mean introduced by Karp [7].

The clock-delay of a register is the time needed for a clock-edge to propagate from the clock source to the register. However, by various reasons caused mainly by the layout, a clock-edge arrives at registers with a not negligible difference of time each other, which brings about the *clock-skew*. Conventional design usually sets the clock-period no smaller than the maximum signal propagation

delay plus the maximum clock-skew to guarantee the correct function. Thus the clock-skew has been considered as a negative effect against speeding up a sequential circuit and efforts have been towards its elimination, i.e. zero clock-skew routing [3, 5, 15].

However there is a different point of view which makes use of the clock-skew to shorter the clock-period. For the circuit whose signal propagation delays are fixed, minimizing the clock-period by controlling arrival times of clock-edges to registers is called here the *clock scheduling problem*. The decision version of the problem to determine if a given clock-period works correctly was formulated as a *feasibility decision problem* of a system of linear inequalities [1, 4]. Similar ideas are found in the multi-phase clock scheduling problem [6, 13, 14]. Since their algorithms only answer the decision problems, they can only find a solution with precision k bits after k trials following the binary search strategy.

Due to routings, process variations, etc., it may be hard to estimate accurately signal propagation delays. It may also be hard to supply clock in designated timing. In such cases, we should take these deviations into consideration. The clock scheduling can be used to improve the reliability of a circuit.

In this paper, we solve the clock scheduling problem: the exact solution is characterized graph theoretically and a polynomial time algorithm to find it is presented. The clock scheduling problem is formulated on a directed weighted graph consisting of two kinds of edges to find a particular cycle. A polynomial time algorithm is provided to find it. Accordingly, the minimum clock-period and the clock-timing of each register are determined. Experimental results on LGSynth93 benchmarks revealed the significance of such considerations on clock scheduling by showing 16% reduction of the clock-period on average. They also show that a circuit reliability is improved by clock scheduling when the clock-period is fixed.

To achieve such a controlled clock scheduling, techniques for clock distribution routing must be developed. For the purpose, it is believed the technology of zero clock-

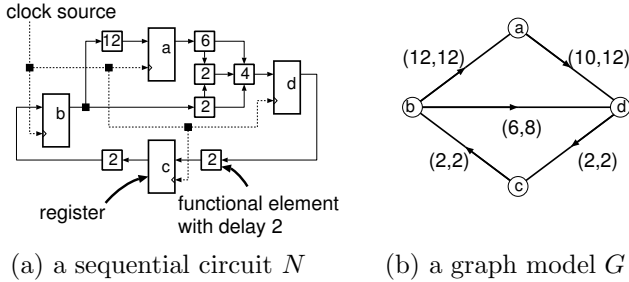


Fig. 1. A graph model of a sequential circuit.

skew routing is available [3, 5, 15].

The rest of the paper is organized as follows. Section II analyzes the clock scheduling problem and formulates it as a graph problem. Section III introduces some terms which are defined on the graph and discusses their implications. Section IV proves the main theorem that claims how the minimum clock-period is determined, and realizes the claim as an algorithm. Experimental results are presented in Section V. Section VI is the conclusion.

II. PRELIMINARIES

Let G be a directed edge-weighted graph. $V(G)$ and $E(G)$ denote the vertex set and edge set of G , respectively. An edge (u, v) with weight $w(u, v)$ has direction from u to v as u being the tail and v head. A (directed) *walk* in G is a sequence $(v_0, e_1, v_1, \dots, e_k, v_k)$, whose terms are alternately vertices and edges of G such that vertices v_i and v_{i-1} are the head and tail of edge e_i , respectively. A walk P in G is called a (directed) *path* if the both edges and vertices of P are distinct. A walk $(v_0, e_1, \dots, e_k, v_k)$ is *closed* if $v_0 = v_k$. A closed walk P is called a (directed) *cycle* if both the edges of P and the vertices v_0, v_1, \dots, v_{k-1} are distinct. For any walk $P = (v_0, e_1, v_1, \dots, e_k, v_k)$, the weight of P , denoted as $w(P)$, is the sum of the edge weights of P , that is, $w(P) = \sum_{1 \leq i \leq k} w(e_i)$.

The sequential circuit N under consideration consists of registers and gates, and wires connecting them. Every register is clocked equi-period but not necessarily simultaneously. In clocking design, only the signal propagation delay between registers is concerned, which is not unique because of signal propagations on various paths, different rise and fall gate delays, etc. In our discussion, only the maximum and minimum propagation delays are significant which are assumed to be estimatable.

We model N by a directed graph G as follows: a vertex $v \in V(G)$ represents a register and an edge $(u, v) \in E(G)$ does the signal transmission from register u to register v along functional elements of the circuit. The weight of edge (u, v) is a pair $(w_{min}(u, v), w_{max}(u, v))$ where $w_{min}(u, v)$ and $w_{max}(u, v)$ are the minimum and maximum signal propagation delays, respectively. See an example in Fig. 1.

A clock from the clock source arrives at each regis-

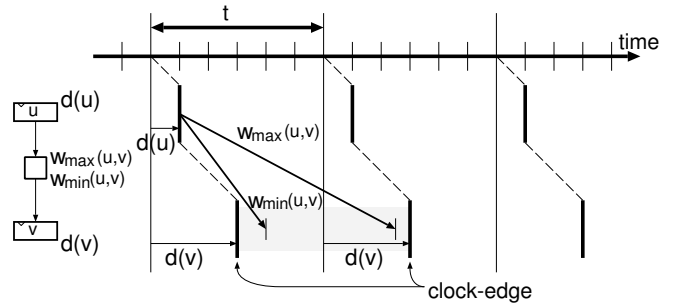


Fig. 2. Legal clocking: A signal started from register u at time $d(u)$ must arrive at register v between $d(v)$ and $t + d(v)$ for correct function.

ter with some delay which is called the *clock-delay*. But our concern is not with the clock-delay itself but relative difference. For convenience to describe it, we take an arbitrary register s as the standard register such that it is ticked by a clock-edge with *standard clock-delay*. Then the register v is ticked $d(v)$ time after the standard register is ticked. If the standard register s is ticked on time $(\dots, -t, 0, t, 2t, \dots)$ where t is the clock-period of the circuit concerned, then v is ticked on time $(\dots, -t + d(v), d(v), t + d(v), 2t + d(v), \dots)$. Our problem is to find the minimum clock-period by clock scheduling, that is, designing clock-timing $d(v)$ of every register.

A signal started from u triggered by a clock-edge at time $d(u)$ arrives at v on time $d(u) + w$ if the signal propagation delay from u to v is w . Since v is clocked as above, the arrival time $d(u) + w$ must be between $d(v)$ and $t + d(v)$ for correct function. See Fig. 2. If it is earlier than $d(v)$, the signal is doubly clocked by a same clock-edge. If it is later than $t + d(v)$, the signal is not clocked by the next due clock-edge. Since a signal propagation delay is between $w_{min}(u, v)$ and $w_{max}(u, v)$, these requirements for all $(u, v) \in E(G)$ are stated as two constraints[4]:

No-Double-Clocking Constraint:

$$d(v) \leq d(u) + w_{min}(u, v)$$

No-Zero-Clocking Constraint:

$$d(u) + w_{max}(u, v) \leq t + d(v).$$

Other technology dependent constraints related to the setup and hold time of the registers and the deviation of due propagation delays, etc.[14], are assumed to be contained in the signal propagation delays. If there are registers constrained to be triggered simultaneously, such as these concerned with primary inputs and/or outputs, then we consider these registers as one register. Thus our framework is defined only by the no-double-clocking and no-zero-clocking constraints. The period t is called *feasible* if there exist $d(v)$'s that satisfy both of them. Note that if t is the minimum clock-period, any $t' \geq t$ is also a feasible clock-period.

These constraints are formulized in linear inequalities in general as $d(v) - d(u) \leq f(u, v)$ where $f(u, v) =$

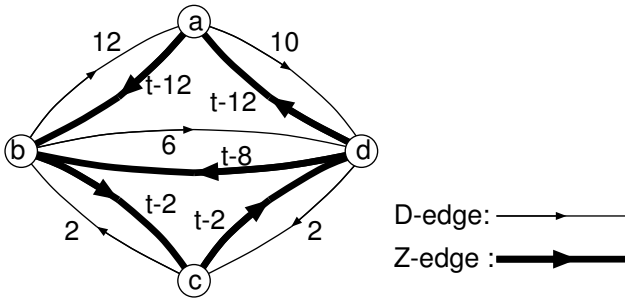


Fig. 3. Constraint graph G_t for the circuit N shown in Fig. 1.

$w_{min}(u, v)$ or $= t - w_{max}(u, v)$. While we regard $d(v)$'s being unknown variables and $f(u, v)$'s given constants, the system is called the *difference constraint*. The following well-known fact enables us to solve the decision version of the problem [9, 8, 11, 12].

Lemma 1 *Let L be a difference constraint in the form of $x(v) - x(u) \leq g_{u,v}$. Let \tilde{G} be a directed graph obtained from L as follows: $V(\tilde{G}) = \{u \mid L \text{ contains } x(u)\}$, and $E(\tilde{G}) = \{(u, v) \mid L \text{ contains } x(v) - x(u) \leq g_{u,v} \text{ where weight } w(u, v) = g_{u,v}\}$. Then the difference constraint is feasible if and only if any directed cycle C in \tilde{G} satisfies $w(C) \geq 0$.*

Our problem is to find the minimum clock-period efficiently. We define the *constraint graph* G_t , where t is a variable standing for the clock-period, is obtained from the graph model G of the circuit by replacing each edge $(u, v) \in E(G)$ with two edges (u, v) and (v, u) with weight $w_{min}(u, v)$ and $t - w_{max}(u, v)$, respectively. The former edge is called the *D-edge* and the latter the *Z-edge*, respectively. D-edges (Z-edges) correspond to the No-Double (No-Zero) Clocking Constraints. For example, the constraint graph G_t for the circuit shown in Fig. 1 is shown in Fig. 3. We claim that the minimum t and associated clock scheduling are determined from this graph.

Once a feasible t is given, we can determine clock-timing $d(v)$ of every register by solving shortest path problems on G_t . The clock-timing of $d(v)$'s that makes t feasible will be in some range. We can use this margin to improve the reliability of the circuit by choosing the clock-timing appropriately. A related discussion with experiments will be given.

III. FORMULATION OF THE MINIMUM CLOCK-PERIOD

Let G_t be the constraint graph. For a subgraph or an edge set S of G_t , $E_Z(S)$ denotes the set of Z-edges contained in S . For an edge set S which contains at least one Z-edge, $w(S)/|E_Z(S)|$ is called the *Z-mean* of S , and denoted as $\overline{w_Z}(S)$. Particularly, if S is a cycle, it is called a *Z-cycle* and $\overline{w_Z}(S)$ is called the *cycle Z-mean*. Its minimum over all Z-cycles in G_t , called *minimum cycle Z-mean* of G_t and denoted as $Z(G_t)$, is a key index in the following discussion.

In the following, we assume that the weight of each D- and Z-edge is finite since the signal propagation delay is naturally so assumed. Due to the hold time, etc., the weights of some D-edges may be negative. However, if there is a negative weight cycle which consists only of D-edges, there is no feasible clock-period by Lemma 1. Therefore our concern is the circuits in which the weight of any cycle which consists only of D-edges is zero or positive. However if there is a zero weight cycle the following discussion becomes unnecessarily lengthy although the similar discussion is possible. Therefore, for compactness of presentation here, we assume that the weight of any cycle which consists only of D-edges is positive. (Note that a negative or zero cycle is easily checked.)

Our problem is to determine the minimum clock-period τ_N of the circuit N by clock scheduling. The following theorem is claimed by Lawler [8, 14] but we state it differently for our purpose. A proof is given for completeness.

Theorem 1 *For any sequential circuit N and the constraint graph G_t for N with $t = 0$, $\tau_N = -Z(G_0)$.*

Proof: Let C be a Z-cycle of G_0 and C' be the Z-cycle of G_{τ_N} corresponding to C in a natural way. Then $w(C') = w(C) + \tau_N |E_Z(C)|$. Since $w(C') \geq 0$ by Lemma 1 and $|E_Z(C)| \geq 1$, we have $\tau_N \geq -\overline{w_Z}(C)$. Since it holds for any Z-cycle, $\tau_N \geq -Z(G_0)$.

Next we show that $\tau_N \leq -Z(G_0)$. The weight of each Z-cycle of C' of G_{τ_N} is non-negative by Lemma 1. There exists a Z-cycle C' of G_{τ_N} such that $w(C') = 0$, otherwise we can reduce the clock-period further without violating the feasibility (Lemma 1), which contradicts τ_N being minimum. Let C be a Z-cycle of G_0 corresponding to C' . Then $w(C') = w(C) + \tau_N |E_Z(C)| = 0$ and we have $\tau_N = -w(C)/|E_Z(C)| = -\overline{w_Z}(C) \leq -Z(G_0)$. \square

IV. COMPUTATION OF THE MINIMUM CYCLE Z-MEAN

A. Main Algorithm

Now our problem is to provide a polynomial time algorithm to compute the minimum cycle Z-mean $Z(G_0)$.

Graph G_0 is a very special graph such that every edge has a parallel edge of opposite direction and different type. The weights of these two edges correspond to the minimum delay and the maximum delay with minus sign, respectively. An example shown in Fig. 4 is the one derived from a circuit in Fig. 1.

We assume that G_0 is strongly connected since otherwise we can determine the minimum cycle Z-mean as the minimum of cycle Z-means of all strongly connected components of G_0 .

Let $n = |V(G_0)|$. Let s be an arbitrarily chosen vertex of G_0 . For every vertex $v \in V(G_0)$ and every non-negative integers i and j ($0 \leq i \leq j \leq n$), let $\mathcal{P}(v, i/j)$ be the set of

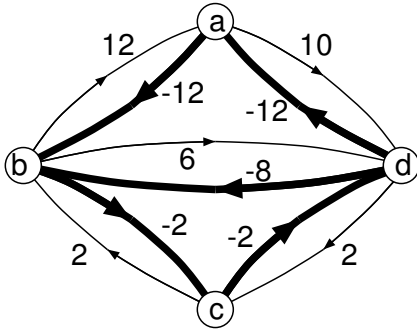


Fig. 4. Constraint graph G_t with $t = 0$ for N shown in Fig. 1.

- Step 1:** Choose a vertex as s , and compute $F(v, i/j)$ for every triple (v, i, j) .

Step 2: Compute $X(v, i)$ for each $v \in V(G_0)$ and i .

Step 3: Compute $Q(v, i)$ for each $v \in V(G_0)$ and i .

Step 4: Take the minimum of all $Q(v, i)$ over all pairs (v, i) . Then, output its minus as τ_N .

Step 5: Find a shortest path in G_{τ_N} from s to each vertex v , and determine $d(v)$ as the length of the shortest path from s to v .

Fig. 5. Algorithm computing the minimum clock-period and clock scheduling.

walks P from s to v such that $|E_Z(P)| = i$ and $|E(P)| = j$. (Necessarily, $j - i$ edges are D-edges.) We define $F(v, i/j)$ as the minimum of weights of all walks from s to v that use exactly i Z-edges and j edges ($0 \leq i \leq j \leq n$), i.e.,

$$F(v, i/j) = \min\{w(P) \mid P \in \mathcal{P}(v, i/j)\} \cup \{\infty\}.$$

In terms of $F(v, i/j)$, two more indexes $X(v, i)$ ($0 \leq i \leq n$) and $Q(v, i)$ ($1 \leq i \leq n$) are defined.

$$X(v, i) = \min_{i \leq j \leq n} F(v, i/j).$$

That is, $X(v, i)$ denotes the minimum of weights of all walks from s to v that use exactly i Z-edges and at most n edges. While $Q(v, i)$ is the upper bound of the Z-mean of a cycle contained in a walk whose weight is minimum in $\mathcal{P}(v, i/n)$. Formally,

$$Q(v, i) = \max_{0 \leq i' \leq i-1} \frac{F(v, i/n) - X(v, i')}{i - i'}$$

if $F(v, i/n) = X(v, i) \neq \infty$, $Q(v, i) = \infty$ otherwise.

Our proposing algorithm is shown in Fig. 5. In the following, we show the correctness of the algorithm.

B. Theorem, lemmas, proofs, and complexity analysis

The correctness of Step 5 in Fig. 5 comes from Lemma 1. Then the main theorem claims that the algorithm in Fig. 5 outputs the minimum clock-period.

Theorem 2 For any sequential circuit N ,

$$\tau_N = - \min_{v \in V(G_0)} \min_{1 \leq i \leq n} Q(v, i),$$

where G_0 is the constraint graph for N with $t = 0$.

We prove Theorem 2 by a series of lemmas.

Lemma 2 For any vertex $v \in V(G_0)$ and any integer i ($1 \leq i \leq n$), $Z(G_0) \leq Q(v, i)$.

Proof: If $F(v, i/n) = \infty$ or $F(v, i/n) > X(v, i)$, then $Q(v, i) = \infty$. Since $Z(G_0)$ is finite, $Z(G_0) \leq Q(v, i)$, so the proof completes. Thus we assume that $F(v, i/n)$ is finite and $F(v, i/n) = X(v, i)$.

Let P be a walk whose weight is minimum in $\mathcal{P}(v, i/n)$, i.e., $w(P) = F(v, i/n)$. Since the number of edges of P is n , P contains cycles. Let arbitrary one of them be C . For simplicity of terminology, let $i^* = |E_Z(C)|$, $j^* = |E(C)|$, $w^* = |w(C)|$. Notice that $0 \leq i - i^* \leq n - j^* < n$. Let P' be the walk obtained from P by deleting the edges of C . (Note that a walk minus a cycle is a walk.) Then, the number of Z-edges, the number of edges, and the weight of the walk P' are $i - i^*$, $n - j^*$, and $F(v, i/n) - w^*$, respectively. Since P' is a member of $\mathcal{P}(v, i - i^*/n - j^*)$ and since $F(v, i - i^*/n - j^*)$ is the minimum weight of such walks, $F(v, i/n) - w^* \geq F(v, i - i^*/n - j^*)$.

If $i^* = 0$, then $w^* > 0$ by the assumption that the weight of a cycle consisting only of D-edges is positive. We have $F(v, i/n) > F(v, i/n - j^*) \geq X(v, i)$. This contradicts the assumption that $F(v, i/n) = X(v, i)$. Thus we have $i^* \geq 1$. Then the cycle Z-mean of C , which is $\overline{w}_Z(C)$, is w^*/i^* . Since $Z(G_0)$ is the minimum of such averages, we have

$$\begin{aligned} Z(G_0) &\leq \frac{w^*}{i^*} \leq \frac{F(v, i/n) - F(v, i - i^*/n - j^*)}{i^*} \\ &\leq \frac{F(v, i/n) - X(v, i - i^*)}{i - (i - i^*)} \\ &\leq \max_{0 \leq i' \leq i-1} \frac{F(v, i/n) - X(v, i')}{i - i'} = Q(v, i). \end{aligned}$$

Thus $Z(G_0) \leq Q(v, i)$ for any v and i . \square

Lemma 3 If $Z(G_0) = 0$, there exist a vertex $v \in V(G_0)$ and an integer i ($1 \leq i \leq n$) such that $Q(v, i) \leq 0$.

Proof: Let C be a Z-cycle of weight 0 in G_0 . Let v' be a vertex in $V(C)$ and P' be a minimum weight walk from s to v' . Since the weight of C is 0, a walk, P' followed by any number of repetitions of C is also a minimum weight walk. Any initial part from s to a vertex on the way of a minimum weight walk is also a minimum weight walk from s to the vertex. So there exists a vertex $v^* \in V(C)$ such that a walk that comprises exactly n edges is a minimum weight walk from s to v^* . Let P be such a minimum weight walk.

Let $i^* = |E_Z(P)|$. Notice that $1 \leq i^* \leq n$. Since $F(v^*, i^*/n) = X(v^*, i^*)$, $Q(v^*, i^*)$ is finite defined as $\max_{0 \leq i' \leq i^* - 1} \frac{F(v^*, i^*/n) - X(v^*, i')}{i^* - i'}$. Furthermore, since $F(v^*, i^*/n) \leq X(v^*, i')$ for any i' ($0 \leq i' \leq i^* - 1$), we have $Q(v^*, i^*) = \max_{0 \leq i' \leq i^* - 1} \frac{F(v^*, i^*/n) - X(v^*, i')}{i^* - i'} \leq 0$. \square

Lemma 4 *There exist a vertex $v \in V(G_0)$ and an integer i ($1 \leq i \leq n$) such that $Q(v, i) \leq Z(G_0)$.*

Proof: Letting $t = Z(G_0)$, G' is a graph obtained from G_0 by reducing each Z-edge weight by t . Then an important fact is that $Z(G') = 0$.

Let $F'(v, i/j)$ be the minimum weight of a walk from s to v in G' using i Z-edges and j edges (i.e. elements of $\mathcal{P}(v, i/j)$ defined in G'). $X'(v, i)$ and $Q'(v, i)$ for G' are similarly defined.

Another important fact is that a minimum weight walk from s to v in G' using i Z-edges and j edges is also such a minimum weight walk in G . Hence $F(v, i/j) = F'(v, i/j) + i \cdot t$ if $F'(v, i/j) \neq \infty$, and $X(v, i) = X'(v, i) + i \cdot t$ if $X'(v, i) \neq \infty$. By Lemma 3, there exist a vertex $v^* \in V(G')$ and an integer i^* ($1 \leq i^* \leq n$) such that $Q'(v^*, i^*) \leq 0$. Since $F'(v^*, i^*/n) = X'(v^*, i^*)$, we have $F(v^*, i^*/n) = X(v^*, i^*)$. Thus

$$\begin{aligned} Q(v^*, i^*) &= \max_{0 \leq i' \leq i^* - 1} \frac{F(v^*, i^*/n) - X(v^*, i')}{i^* - i'} \\ &= \max_{0 \leq i' \leq i^* - 1} \frac{F'(v^*, i^*/n) - X'(v^*, i')}{i^* - i'} + t \\ &= Q'(v^*, i^*) + t \leq t. \end{aligned}$$

Thus $Q(v^*, i^*) \leq Z(G_0)$. \square

Since $\tau_N = -Z(G_0)$, Theorem 2 comes from Lemmas 2 and 4. Notice that Theorem 2 is naturally described independently from the choice of s , the standard register used for clock-timing reference. Notice also that the minimum cycle Z-mean can be computed for any graph G consisting two types of edges, D-edges and Z-edges, if the weight of any cycle consisting only D-edges is positive.

Finally, we show that τ_N and the corresponding clock-timing $d(v)$ for all register v in N can be computed in $\mathcal{O}(n^2e)$ time where $n = |V(G_0)|$ and $e = |E(G_0)|$. Step 1 can be executed in $\mathcal{O}(n^2e)$ time. The following steps totally take further $\mathcal{O}(n^3)$ times. Since N is strongly connected, $n \leq e$, so, the minimum cycle Z-mean can be computed in $\mathcal{O}(n^2e)$ time.

C. Example

The constraint graph G_t with $t = 0$ is shown in Fig. 4 for a sequential circuit N in Fig. 1. The minimum cycle Z-mean of G_0 is -9 which cycle (a, b) , (b, d) , (d, a) attains. Then, the clock-timing $d(v)$ of each v is obtained as follows: taking register a as the standard,

$$d(a) = 0, \quad d(b) = -3, \quad d(c) = 4, \quad d(d) = 3.$$

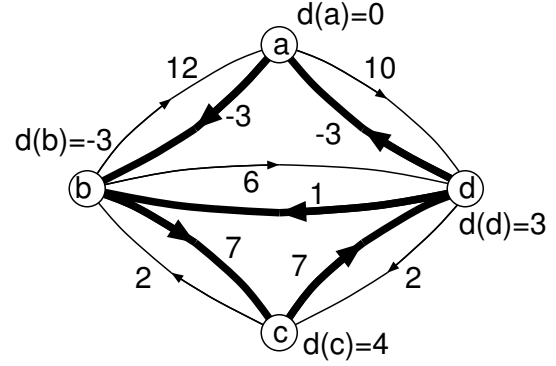


Fig. 6. Constraint graph G_t with optimum clock-period $t = 9$.

See a clock realization in Fig. 6. It is easily verified that the clock assignment as above works correctly.

V. EXPERIMENTS

The clock scheduling technique was applied to LGSynth93 benchmarks using JCC JS5/70 (equivalent to SUN SPARC 5). Results are presented in Tables I and II.

The first experiment is to see the effect of the clock scheduling. In Table I, “Regs” include registers themselves and IO terminals. “ w_{max} ” is the maximum delay between registers estimated by the sum of the maximum gate delays along a path where the maximum gate delay is defined as the larger of raise and fall delays. Similarly, minimum delay is estimated by the sum of the minimum gate delays along a path. While “ τ_N ” is our result. The reduction of clock-period to the maximum signal propagation delay is 16.1% on average.

The second experiment is to see the allowance in clock schedule design. In Table II, the minimum clock-periods are shown in cases that a certain deviation from the estimated propagation delay is inevitable. They are within $\pm 2.5\%$, $\pm 5\%$, or $\pm 10\%$ of each signal propagation delay. The last column shows the allowance of the deviation of propagation delay in percent precision in the case that the clock-period is set equal to the maximum signal propagation delay in the circuit. This shows that the clock-period which equals to the maximum signal propagation delay can be achieved if the deviation of each signal propagation delay is at most 13.7% (average) in the case of LGSynth93 benchmarks. We can also show that we can achieve the clock-period which equals to the maximum signal propagation delay if the deviation of due clock-timing is at most 3.5% (average) of the maximum signal propagation delay (Experimental results are omitted here).

VI. CONCLUDING REMARKS

Considerations of this paper are fully standing on the expectation that some way is available which can control

TABLE I
RESULTS OF CLOCK SCHEDULING. (U: UNIT DELAY)

name	# Regs (# IO)	w_{max} [u]	τ_N [u]	red.[%]	CPU [s]
s27	8 (5)	3.01	2.39	20.6	0.00
s208	21 (13)	7.11	5.01	29.5	0.01
s298	23 (9)	4.94	3.89	21.3	0.02
s349	35 (20)	9.24	6.53	29.3	0.10
s382	30 (9)	5.78	4.07	29.6	0.08
s386	20 (14)	6.81	6.68	1.9	0.02
s400	30 (9)	5.94	4.22	29.0	0.07
s444	30 (9)	6.43	4.72	26.6	0.09
s510	32 (26)	6.02	5.74	4.7	0.06
s526	30 (9)	4.94	3.89	21.3	0.07
s641	78 (59)	30.92	27.49	11.1	1.72
s713	77 (58)	32.12	28.99	9.7	1.66
s820	42 (37)	6.65	6.29	5.4	0.20
s838	69 (37)	25.34	23.51	7.2	0.90
s953	68 (39)	7.59	6.19	18.4	0.88
s1196	46 (28)	12.93	10.96	15.2	0.36
s1238	46 (28)	13.26	11.58	12.7	0.36
s1423	96 (22)	35.72	32.45	9.2	9.61
s1488	33 (27)	10.34	9.45	8.6	0.12
s1494	33 (27)	10.75	9.60	10.7	0.12
avg.	42 (24)	12.29	10.68	16.1	0.82

the clock distribution with arbitrary clock-timing to every register. Though such a technology falls beyond the scope of this paper, it seems probable that zero clock-skew routing is available for the purpose. In fact, some of zero clock-skew routing algorithms, for example [3], are considered. When a large amount of clock-skew is requested, combination of the multi-clock source and clock routing techniques may be valid for our purpose. The idea in [10] may be helpful in which the non-zero clock-skew routing problem is discussed.

The larger the differences of the maximum and minimum propagation delays, the severer the constraints are, so the reduction of the clock-period tends to be smaller. Thus our results have created a new problem to reduce the difference of the maximum and minimum propagation delays from registers to registers.

For reliability concerned, it is essential to take an enough margin for clock-period to ensure a correct performance of a sequential circuit. The idea in this paper will be useful to improve circuit reliability in this way.

The urgent future work in this direction should be to develop a layout optimization techniques based on the clock scheduling technique.

REFERENCES

- [1] R. B. Deokar and S. S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," in *Proc. ISCAS '94*, pp. 1:407–410, 1994.
- [2] R. B. Deokar and S. S. Sapatnekar, "A fresh look at retiming via clock skew optimization," in *Proc. 32nd DAC*, pp. 310–315, 1995.

TABLE II
ROBUSTNESS OF CLOCK SCHEDULING.

name	w_{max} [u]	τ_N [u]			Allow. [%]
		$\pm 2.5\%$	$\pm 5.0\%$	$\pm 10\%$	
s27	3.01	2.45	2.51	2.63	26
s208	7.11	5.24	5.47	5.93	23
s298	4.94	4.04	4.19	4.49	17
s349	9.24	6.83	7.13	7.72	22
s382	5.78	4.26	4.45	4.82	23
s386	6.81	6.85	7.02	7.35	1
s400	5.94	4.42	4.61	4.99	22
s444	6.43	4.93	5.13	5.54	21
s510	6.02	5.88	6.02	6.31	4
s526	4.94	4.04	4.19	4.49	17
s641	30.92	28.35	29.20	30.92	10
s713	32.12	29.87	30.75	32.51	9
s820	6.65	6.44	6.60	6.92	5
s838	25.34	24.19	24.87	26.22	7
s953	7.59	6.35	6.55	7.00	16
s1196	12.93	11.32	11.68	12.40	13
s1238	13.26	11.95	12.33	13.07	10
s1423	35.72	33.43	34.40	36.35	8
s1488	10.34	9.69	9.93	10.40	8
s1494	10.75	9.84	10.08	10.56	12
avg.	12.29	11.02	11.36	12.03	13.7

- [3] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *Proc. 30th DAC*, pp. 612–616, 1993.
- [4] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. on Computers*, vol. 39, no. 7, pp. 945–951, 1990.
- [5] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock routing for high-performance ICs," in *Proc. 27th DAC*, pp. 573–579, 1990.
- [6] D. A. Joy and M. J. Ciesielski, "Placement for clock period minimization with multiple wave propagation," in *Proc. 28th DAC*, pp. 640–643, 1991.
- [7] R. M. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete Mathematics*, vol. 23, pp. 309–311, 1978.
- [8] E. L. Lawler, *Combinatorial Optimization, Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.
- [9] C. E. Leiserson and J. B. Saxe, "Retiming synchronous circuitry," *Algorithmica*, vol. 6, no. 1, pp. 5–35, 1991.
- [10] J. L. Neves and E. G. Friedman, "Circuit synthesis of clock distribution networks based on non-zero clock skew," in *Proc. ISCAS '94*, pp. 4:175–178, 1994.
- [11] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization, Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [12] M. C. Papaefthymiou, "Understanding retiming through maximum average-delay cycles," *Math. System Theory*, vol. 27, pp. 65–84, 1994.
- [13] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun, "Analysis and design of latch-controlled synchronous digital circuits," in *Proc. 27th DAC*, pp. 111–117, 1990.
- [14] T. G. Szymanski, "Computing optimal clock schedules," in *Proc. 29th DAC*, pp. 399–404, 1992.
- [15] R. S. Tsay, "Exact zero skew," in *Proc. 1991 ICCAD*, pp. 336–339, 1991.