

**Analysis and Design of  
Latch-Controlled Synchronous  
Digital Circuits**

By

K.A. Sakallah, T.N. Mudge and  
O.A. Olukotun

CSE-TR-31-89

**THE UNIVERSITY OF MICHIGAN**

**COMPUTER SCIENCE AND ENGINEERING DIVISION**

**DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

Room 3402, EECS Building

Ann Arbor, Michigan 48109-2122

USA

# Analysis and Design of Latch-Controlled Synchronous Digital Circuits<sup>1</sup>

Karem A. Sakallah , Trevor N. Mudge and Oyekunle A. Olukotun

Department of Electrical Engineering and Computer Science  
University of Michigan  
Ann Arbor, MI 48109-2122  
Phone: 313-936-1350  
FAX: 313-763-1503  
E-mail: karem@razi.eecs.umich.edu

October 1989

## Abstract

We present a succinct, yet complete, formulation of the timing constraints for latch-controlled synchronous digital circuits. We show that the constraints are mildly nonlinear, and prove the equivalence of the nonlinear optimal cycle time calculation problem to an associated and simpler linear programming (LP) problem. We present an LP-based algorithm which is guaranteed to obtain the optimal cycle time for arbitrary circuits controlled by a general class of multi-phase overlapped clocks.

---

<sup>1</sup>This work was supported in part by NSF Grant MIP-8802771

## 1 Introduction

The analysis and design of synchronous digital circuits which are controlled by level-sensitive latches is generally acknowledged to be a difficult problem [1,2,3]. The difficulty is due mainly to the coupling between the input and output terminals of a latch while the latch is enabled. This in turn leads to a set of cyclic timing constraints which must be satisfied by a properly-designed circuit. The analysis problem seeks to determine if these constraints are indeed satisfied for a given circuit and a given clocking scheme. The design problem, on the other hand, attempts to find, for a given circuit and clocking scheme, the minimum clock cycle time which would not violate these constraints. In both cases the cyclic nature of the constraints frustrates intuitive solution approaches based on simple graph traversal methods, such as CPM [4], which require the constraint set to be acyclic.

Recently it was suggested [3] that the design problem can be formulated as a linear program (LP). While this statement is true, it is not at all obvious how one would go about it. In fact, in [5] it is correctly pointed out that the timing constraints are *nonlinear*, implying that an LP formulation is at best an approximation.

This paper has two main goals. The first is to present a succinct, yet complete, formulation of the timing constraints for latch-controlled synchronous digital circuits. The constraints in this formulation are easily constructed, almost by inspection, for any circuit topology and clocking scheme, and are clearly seen to be nonlinear, though mildly so. The second goal of the paper is to formally prove the equivalence of the nonlinear design problem (i.e. the minimum clock cycle optimization problem with nonlinear timing constraints) to an associated and simpler LP problem.

Most current methods for the analysis and design of level-sensitive synchronous digital circuits assume edge-triggering to simplify the analysis and then apply some heuristics to approximate the level-sensitive constraints. As a consequence, in the analysis case, they may declare a design to be in violation of timing constraints when in fact it is not, and in the design case, they may not produce the minimum cycle time. Our modeling of the level-sensitive constraints is not an approximation and so avoids both of these problems.

This paper is organized as follows. Section 2 reviews previous work in the area. Section 3 presents a formulation of the timing constraints for level-sensitive synchronous digital circuits. They are seen to be non-linear. Section 4 shows that the solution of the apparently nonlinear optimal cycle time design problem can in fact be found by solving an associated LP problem, and therefore that the entire body of LP theory can be brought to bear on design and analysis problems. Section 5 illustrates the formulation and solution of the design problem for two examples. Finally, Sec. 6 contains some concluding remarks and discussion of future directions.

## 2 Previous Work

The timing analysis of digital logic circuits goes back at least to the work of Kirkpatrick in the 1960's [6]. However, this and much subsequent work was concerned with timing analysis for edge-triggered logic. It has only been during this decade, with MOS VLSI emerging as the leading technology for implementing digital systems, that the timing analysis and design of level-sensitive logic has become important. In this period several authors have addressed the question of level-sensitive latches including Agrawal [7], Jouppi [1], Ousterhout [2], Unger [8], Szymanski [9], and Dagenais [5,3]. One of the earliest, Agrawal, attempts to find the maximum frequency of operation of a logic circuit through a bounded binary search algorithm. Jouppi proposed an iterative scheme based on the concept of "borrowing." In the first iteration, the critical path(s) in the circuit are determined by pretending that the latches are edge-triggered. This approximation is removed in subsequent iterations. In each of these iterations an attempt is made to reduce the clock cycle time to a value determined by the second most critical path. This is accomplished by trading (borrowing) the slack time in the sub-critical path. In practice, only one borrowing iteration is performed to limit the computation cost. The TV program incorporates this borrowing algorithm, and has been effectively applied for the verification of several large commercial chips. Ousterhout developed Crystal, a MOS timing verification program similar in many respects to TV. However, Crystal makes no attempt at dealing with clocking issues and confines its attention

to the proper modeling of signal delay through trees of MOS transistors. In fact, Ousterhout acknowledged the inherent difficulty of dealing with level-sensitive latches. Unger developed a set of timing constraints for a limited form of 2-phase clocking with level-sensitive latches. He considered both the short-path (early arrival) as well as the long-path (late arrival) problems and presented a heuristic procedure for computing the minimum cycle time subject to these constraints. This, to our knowledge, was the first explicit formulation of the timing constraints of latch-controlled circuits as a system of linear inequalities. The LEADOUT program, developed by Szymanski, is an equation-based MOS timing analysis tool which handles multi-phase clocking and level-sensitive latches. The temporal behavior of latches is specified by “max” constraints similar to those encountered in CPM graphs. To eliminate the inevitable cyclic dependencies among these constraints, the circuit is first partitioned into its strongest-connected components, and constraints are generated for each cycle-free path within the components. It is not clear, however, how the cyclic dependencies induced by clock periodicity are removed. A unique feature of LEADOUT is the compilation of the timing constraints into a fast-executing program which allows repeated analysis of a circuit with different clocking or device parameters. The most recent effort at addressing this problem is due to Dagenais. He developed a MOS timing analysis and design tool, TAMIA, which represents the timing behavior of general multi-phase-clocked latch-controlled circuits by a set of nonlinear coupled relations. The design problem, which aims at finding the optimal clock parameters for a given circuit, is then solved approximately by an iterative graph-based algorithm.

### 3 Problem Formulation

We consider synchronous digital circuits controlled by arbitrary  $k$ -phase clocks (to be defined shortly). We assume, see Fig. 1, that the circuits can be decomposed into stages of feedback-free combinational logic blocks with clocked inputs and outputs<sup>2</sup>. The clocked elements at the inputs and outputs of the combinational stages are level-sensitive latches which provide

---

<sup>2</sup>Figure 1 is adapted from Glasser and Dobberpuhl [10] Fig. 6.7 p. 335.

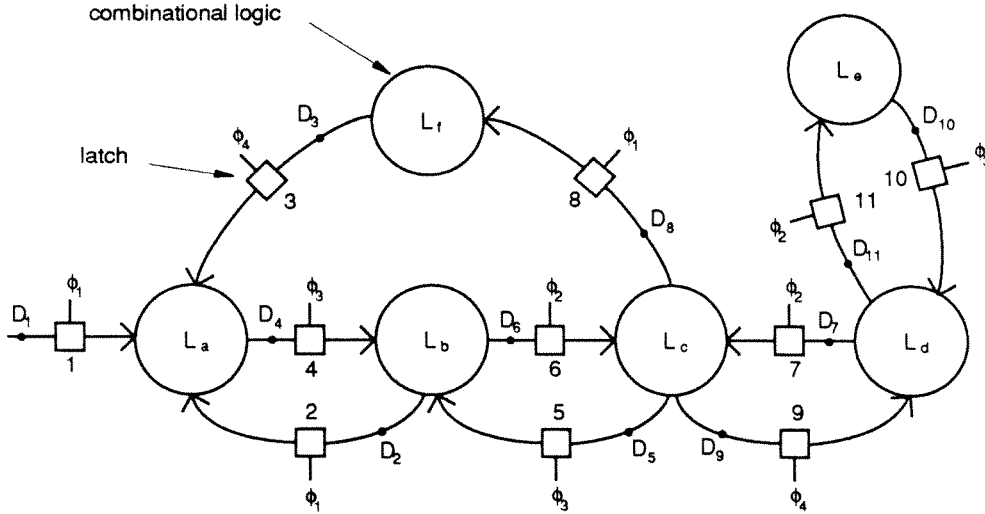


Figure 1: Generalized logic model.

temporary storage of data and act as *synchronizers*. These latches can be either static (for example cross-coupled NAND gates) or dynamic (for example MOS pass transistors). Regardless of the implementation, the functional and timing behavior of these latches is similar, and the formulation presented here applies to both types.

We place no restrictions on the combinations of clock phases used to control the input and output latches of a combinational stage other than a requirement that the set of clock phases controlling each feedback loop in the circuit be non-overlapping; specifically, we require the logical AND of this set of phases to be identically equal to 0 at all times.

### 3.1 Clocking Methodology

An arbitrary  $k$ -phase clock is defined to be a collection of  $k$  periodic signals  $\phi_1, \phi_2, \dots, \phi_k$  — referred to as *phases* — with the same periodicity. Each phase consists of two intervals: an *active* interval during which the latches controlled by the phase are enabled; and a *passive* interval when the latches are disabled. Without loss of generality we assume that all clock phases are active high, i.e., their active intervals occur when the phase signal assumes the

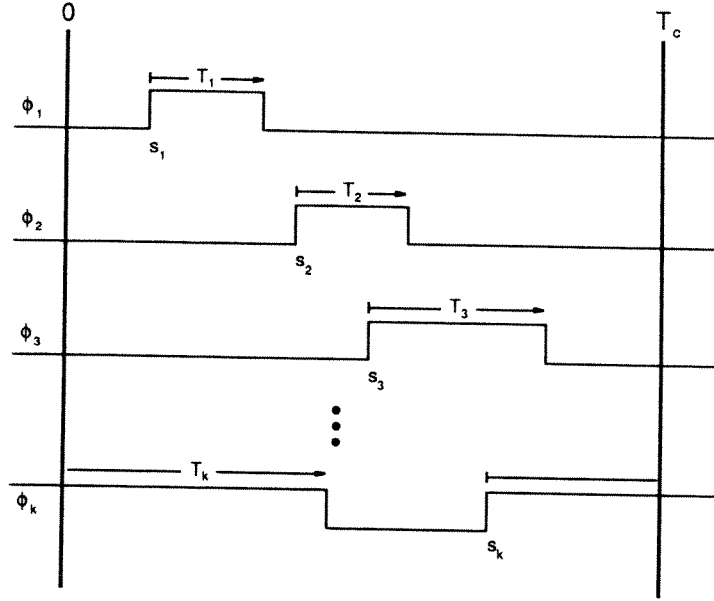


Figure 2: Clock signal variables.

logic 1 state.

We define the following clock variables (see Fig. 2):

- $T_c$ : the clock cycle time, or period.
- $s_i$ : the start time, relative to the beginning of the common clock cycle, of the active interval of  $\phi_i$ .
- $T_i$ : the duration of the active interval of  $\phi_i$ .

For brevity, we will identify  $\phi_i$  with its active interval, and simply refer to  $s_i$  as the start of  $\phi_i$  and to  $T_i$  as the duration or width of  $\phi_i$ . In addition, if  $\phi_i$  and  $\phi_j$  control an input latch and an output latch, respectively, of a combinational logic block  $L$ , we will simply refer to  $\phi_i$  as an input phase,  $\phi_j$  as an output phase, and  $\phi_i/\phi_j$  as an input/output-phase pair of  $L$ . We also introduce two  $k \times k$  matrices  $C$  and  $K$  with elements  $C_{ij}$  and  $K_{ij}$  defined as follows:

$$C_{ij} \equiv \begin{cases} 0 & i < j \\ 1 & i \geq j \end{cases} \quad (1)$$

$$K_{ij} \equiv \begin{cases} 1 & \text{if } \phi_i/\phi_j \text{ is an I/O-phase pair of any logic block} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The  $K$  matrix identifies all I/O-phase pairs for a particular circuit. The  $C$  matrix is used to determine if a clock cycle boundary must be crossed when going between an I/O-phase pair  $\phi_i/\phi_j$ .

We can now state the relations among the various clock variables as a set of inequalities which are collectively referred to as the *clock constraints*:

**C1. Periodicity Constraints:**

$$T_i \leq T_c \quad i = 1, \dots, k \quad (3)$$

$$s_i \leq T_c \quad i = 1, \dots, k \quad (4)$$

**C2. Phase Ordering Constraints:**

$$s_i \leq s_{i+1} \quad i = 1, \dots, k-1 \quad (5)$$

**C3. Phase Non-overlap Constraints:**

$$s_i \geq s_j + T_j - C_{ji}T_c \quad \forall (i, j) \ni K_{ij} = 1 \quad (6)$$

**C4. Clock Non-negativity Constraints:**

$$T_c \geq 0 \quad (7)$$

$$T_i \geq 0 \quad i = 1, \dots, k \quad (8)$$

$$s_i \geq 0 \quad i = 1, \dots, k \quad (9)$$

These inequalities, except for the non-overlap constraints C3, are intuitively obvious. Constraints C3 insure that the output phase  $\phi_j$  of every I/O-phase pair must end before the input phase  $\phi_i$  starts. This in turn guarantees that a set of clock phases controlling a feedback loop in the circuit are never simultaneously overlapping.

Constraints C1-C4 should be viewed as the minimum set of requirements that must be satisfied by a  $k$ -phase clock. Further requirements, such as minimum phase width and



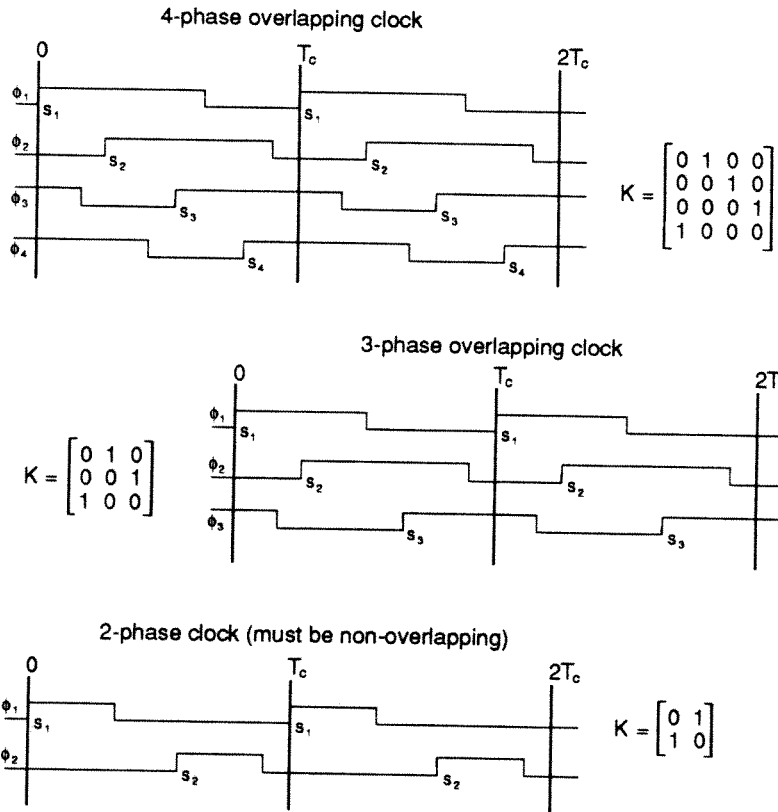


Figure 3: Clocks with 2-, 3-, and 4-phases.

minimum phase separation, can be easily added to this minimum set, but will not be treated here.

Figure 3 shows how these inequalities might be satisfied for commonly used 2-, 3-, and 4-phase clocking schemes. Note in particular that for  $k = 2$ , the inequalities insure that the two phases are non-overlapping as they should be.

### 3.2 Latch Constraints

As will become evident, the simplicity of the formulation we present stems from a careful choice of time variables, and naturally leads to a solution by linear programming.

We describe here the timing constraints necessary for the correct operation of D-type

latches. Such latches have three terminals representing data input, data output, and control (*clock*) input (see Fig. 1). The circuit is assumed to contain  $l$  latches numbered from 1 to  $l$ . For each of these latches we define the following variables and parameters:

- $p_i$ : denotes the clock phase used to control latch  $i$  (e.g., latch 3 in Fig. 1 has  $p_3 = 4$ )
- $A_i$ : denotes the arrival time, relative to the beginning of phase  $p_i$ , of a valid data signal at the input to latch  $i$ .
- $D_i$ : denotes the departure time, that is the earliest time, relative to the beginning of phase  $p_i$ , when the signal available at the data input of latch  $i$  starts to propagate through the latch.
- $Q_i$ : denotes the earliest time, relative to the beginning of phase  $p_i$ , when the signal at the data output of latch  $i$  starts to propagate through the succeeding stage of combinational logic.
- $\Delta_{DCi}$ : denotes the setup time for latch  $i$  required between the data input and the trailing edge of the control input.
- $\Delta_{DQi}$ : denotes the propagation delay of latch  $i$  from the data input to the data output of the latch while the control input is high. It is assumed that  $\Delta_{DQi} \geq \Delta_{DCi}$ .
- $\Delta_{ij}$ : denotes the propagation delay from an input latch  $i$  through a combinational logic block to an output latch  $j$ . If latches  $i$  and  $j$  are not directly connected by a combinational block, then  $\Delta_{ij} \equiv -\infty$ .

Notice that both  $D_i$  and  $Q_i$  will always be non-negative quantities, whereas  $A_i$  is unrestricted in sign.

The constraints governing latch operation fall into two categories: *setup* constraints and *propagation* constraints. The setup constraints guarantee that a latch has sufficient time to lock (store) the signal at the data input before that signal is allowed to change again. Thus,

$$A_i + \Delta_{DCi} \leq T_{p_i}, \quad i = 1, \dots, l \quad (10)$$

Since  $A_i$  can be negative, signifying that valid data has arrived *before* the onset of phase  $p_i$ , (10) may sometimes be satisfiable by a clock phase whose width  $T_{p_i}$  is 0! A more realistic setup constraint is obtained if  $A_i$  is replaced by  $D_i$  yielding:

$$D_i + \Delta_{DCi} \leq T_{p_i} \quad i = 1, \dots, l \quad (11)$$

In this case, since  $D_i$  is always non-negative, the constraint places a lower bound on the width of phase  $p_i$  equal to the required setup time. We will adopt this more realistic constraint in our analysis.

Unlike the setup constraints which are local, the propagation constraints are global. They relate the departure times of signals at different latches in the circuit using the combinational propagation delay parameters. Since latch variables are referenced to the beginning of their corresponding clock phase, it is convenient to define the following phase-shift operator:

$$S_{ij} \equiv s_i - (s_j + C_{ij}T_c) \quad (12)$$

Adding  $S_{ij}$  to a time variable moves its referenced point (origin) ahead from  $s_i$  to  $s_j$ .

The propagation of signals from the inputs to the outputs of latches is simply described by:

$$Q_i = D_i + \Delta_{DQ_i} \quad i = 1, \dots, l \quad (13)$$

Now consider a combinational path which starts at latch  $j$  and ends at latch  $i$ . The data signal at latch  $j$  starts to propagate at time  $Q_j$ , and thus reaches latch  $i$  at  $Q_j + \Delta_{ji}$ , all times being referenced to the beginning of phase  $p_j$ . Therefore, relative to the beginning of phase  $p_i$ , the signal *arrives* at latch  $i$  at time  $Q_j + \Delta_{ji} + S_{p_j p_i}$ . The data signal at latch  $i$  becomes valid when *all* relevant input signals had had sufficient time to propagate through the combinational circuitry leading to latch  $i$ . Thus, the *arrival time* of a valid signal at

latch  $i$  becomes:

$$A_i = \max_j(Q_j + \Delta_{ji} + S_{p_j p_i}) \quad j = 1, \dots, l \quad (14)$$

The propagation constraints through the combinational logic can now be expressed as follows:

$$D_i = \max(0, A_i) \quad i = 1, \dots, l \quad (15)$$

which express the fact that if a valid signal arrives at latch  $i$  before the start of phase  $p_i$ , then the departure time of that signal must be delayed to the beginning of phase  $p_i$ .

By eliminating the  $Q$  and  $A$  variables using (13) and (14), the latch constraints can be written exclusively in terms of signal departure times  $D_i$  along with the various clock variables, as follows:

**L1. Setup Constraints:**

$$D_i + \Delta_{DCi} \leq T_{p_i} \quad i = 1, \dots, l \quad (16)$$

**L2. Propagation Constraints:**

$$D_i = \max(0, \max_j(D_j + \Delta_{DQj} + \Delta_{ji} + S_{p_j p_i})) \quad i, j = 1, \dots, l \quad (17)$$

**L3. Latch Non-negativity Constraints:**

$$D_i \geq 0 \quad i = 1, \dots, l \quad (18)$$

Using the notation scheme defined in this section, it is now possible to write down the set of timing constraints for arbitrary circuits by inspection. It is assumed that the circuit has been decomposed into clocked combinational stages, and that the various delay parameters have been calculated. We illustrate this process in the Appendix for the circuit shown in Fig. 1.

## 4 Optimal Clock Cycle Calculation

The minimum clock cycle time can be calculated by solving the following optimization problem, denoted by P1:

### P1: Optimal Cycle Time

$$\begin{aligned} &\text{Minimize} && T_c \\ &\text{Subject to} && \text{Clock Constraints C1, C2, C3, and C4} \\ &&& \text{Latch Constraints L1, L2, and L3} \end{aligned}$$

P1 is a nonlinear optimization problem. The nonlinearity is due to the max functions in the latch propagation constraints L2. A linear optimization problem is obtained if these propagation constraints are *relaxed* as follows:

### L2R. Relaxed Propagation Constraints:

$$D_i \geq D_j + \Delta_{DQ_j} + \Delta_{j_i} + S_{p_j p_i} \quad i, j = 1, \dots, l \quad (19)$$

Thus, we define the following linear program:

### P2: Modified Optimal Cycle Time

$$\begin{aligned} &\text{Minimize} && T_c \\ &\text{Subject to} && \text{Clock Constraints C1, C2, C3, and C4} \\ &&& \text{Latch Constraints L1, L2R, and L3} \end{aligned}$$

If we denote the optimal value for P1 and P2 by  $T_{c,min}^{(P1)}$  and  $T_{c,min}^{(P2)}$  respectively, we can state the following theorem:

### Theorem:

$$T_{c,min}^{(P1)} = T_{c,min}^{(P2)} \quad (20)$$

*Proof:* The critical element of this proof is the observation that the optimal value of a linear program cannot be improved with the addition of extra constraints [4]. It follows that, since P2 is a relaxed version of P1, that  $T_{c,min}^{(P1)} \geq T_{c,min}^{(P2)}$ . Therefore the theorem is proved if we can show that P2 can be augmented with constraints such that the following two stipulations are true:

1. The optimal value of the augmented problem is not *greater* (worse) than  $T_{c,min}^{(P2)}$ .
2. The constraints of the augmented problem are equivalent to those of P1.

Thus the augmented problem, which we will call P3, has the same optimal solution (i.e., the same values for all variables) as the original problem P1. Since the only difference between P1 and P2 are the latch propagation constraints, we need to examine the following two cases:

1. All  $D$  variables in the optimal solution to P2 are at their minimum values. Thus, the optimal solution to P2 satisfies all the constraints of P1, including the latch propagation constraints L2. In this case, P2 is equivalent to P1, and their optimal values are the same.
2. One or more of the  $D$  variables is not at its minimum value. Thus the optimal solution to P2 violates some of the latch propagation constraints L2. To force the solution to satisfy the L2 constraints we augment the constraints of P2 with equality constraints as follows:

- (a) If  $A_i \leq 0$  and  $D_i > 0$  for some latch  $i$ , add the following equality constraint:

$$D_i = 0 \tag{21}$$

- (b) If  $A_i > 0$  and  $D_i > A_i$  for some latch  $i$ , add the following equality constraint:

$$D_i = A_i \tag{22}$$

The addition of these equality constraints may in some cases cause the departure time at some other latch  $j$  which previously satisfied constraints L2 to now violate them. In such cases we add further equality constraints (either (a) or (b), as appropriate) for all such affected latches, and repeat the procedure as often as necessary, until the constraints of P3 become equivalent to those of P1. It should be obvious that the addition of such equality constraints does not increase the cycle time. Thus the two stipulations stated above are true, and the theorem is proved. □

This theorem forms the basis for the following algorithm to find the optimal solution of P1 by Linear Programming:

**Algorithm MLP:** Optimal Cycle Time Calculation by Modified LP

1. Solve P2.
2. Construct P3 by augmenting P2 with the necessary equality constraints, as described in the above theorem.
3. Solve P3.

While this algorithm involves the solution of two linear programs, P2 and P3, the construction as well as the solution of P3 are trivially obtained by “sliding” each of the  $D_i$  variables found in the solution to P2 towards the time origin until  $D_i = \max(0, A_i)$ . Thus this algorithm obtains the solution to the original *nonlinear* optimization problem, P1, by solving a *linear* optimization problem, P2, followed by a simple correction.

## 5 Examples

In this section we illustrate our proposed formulation with two examples. The first, adapted from [3], is shown in Fig. 4. It is a simple two-stage system connected in a loop and controlled by a two-phase clock. To facilitate comparison with [3] we assume that all latches have equal setup and propagation delays of 10 nS. We also assume the same values for the combinational logic delays, except for block  $L_4$  whose delay  $\Delta_{41}$  will be varied to study its effect on the optimal cycle time. The resulting set of timing constraints are:

- Periodicity Constraints:  $T_i, s_i \leq T_c \quad i = 1, 2$
- Phase Ordering Constraints:  $s_1 \leq s_2$
- Phase Non-overlap Constraints:  $s_1 \geq s_2 + T_2 - T_c$  and  $s_2 \geq s_1 + T_1$
- Latch Setup Constraints:  $D_1 + 10 \leq T_1 \quad D_2 + 10 \leq T_2$   
 $D_3 + 10 \leq T_1 \quad D_4 + 10 \leq T_2$

- Latch Propagation  $D_1 = \max(0, D_4 + 10 + \Delta_{41} + s_2 - s_1 - T_c)$   
 Constraints:  $D_2 = \max(0, D_1 + 10 + 20 + s_1 - s_2)$   
 $D_3 = \max(0, D_2 + 10 + 20 + s_2 - s_1 - T_c)$   
 $D_4 = \max(0, D_3 + 10 + 60 + s_1 - s_2)$
- Non-negativity Constraints:  $T_c \geq 0$ ;  $T_{p_i}, s_{p_i} \geq 0$   $p_i = 1, 2$ ; and  $D_i \geq 0$   $i = 1, \dots, 4$

Figures 5 and 6 compare the results obtained by the MLP algorithm with the *null retardation in the initial phase* (NRIP) algorithm described in [3]. We make the following observations on these results.

- Unless additional constraints are placed on the minimum widths and separations of clock phases, the optimal solution will not be unique. This is illustrated with two such solutions for the  $\Delta_{41} = 80$  nS case, see the top of Fig. 5. The apparent uniqueness of the solution found by the NRIP algorithm is due to its implicit *minimum* constraints on phase widths and separations.
- The NRIP algorithm produces an optimal solution for  $\Delta_{41} = 60$ nS. For all other values of  $\Delta_{41}$ , the cycle time found by NRIP is suboptimal, see Fig. 6.
- The piece-wise linear dependence of  $T_c$  on  $\Delta_{41}$  in the optimal solution has three distinct segments. For  $0 \leq \Delta_{41} \leq 20$ nS,  $T_c$  is independent of  $\Delta_{41}$ , and is set by some other delay in the circuit. When  $\Delta_{41} \geq 20$ nS, block  $L_d$  becomes critical, and any increase in  $\Delta_{41}$  causes  $T_c$  to also increase. For  $20 \leq \Delta_{41} \leq 100$ nS,  $T_c$  increases by one nanosecond for every two-nanosecond increase in  $\Delta_{41}$  because the added delay is shared between the two clock cycles (“borrowed” from  $\phi_1$ ). For  $\Delta_{41} \geq 100$ ,  $T_c$  increases in direct proportion to  $\Delta_{41}$ , since the additional delay can no longer be shared between the two clock cycles, and slack is inevitably introduced in the cycle with shorter delay. The rather simple dependence of the optimal cycle time on  $\Delta_{41}$  in this case is due to the simplicity of the topology of this particular circuit. In fact, one can show that since the feedback loop consists of two complete clock cycles, the optimal cycle time is the maximum of the average delay around the loop and the difference between the



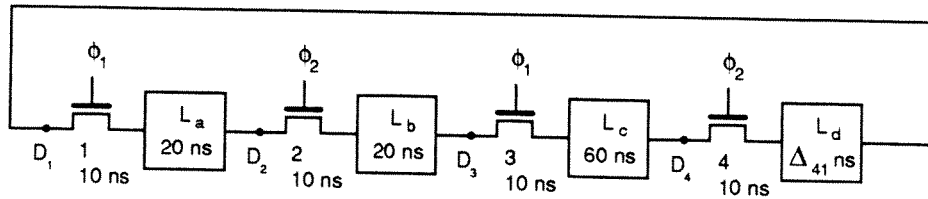


Figure 4: Block diagram for example 1.

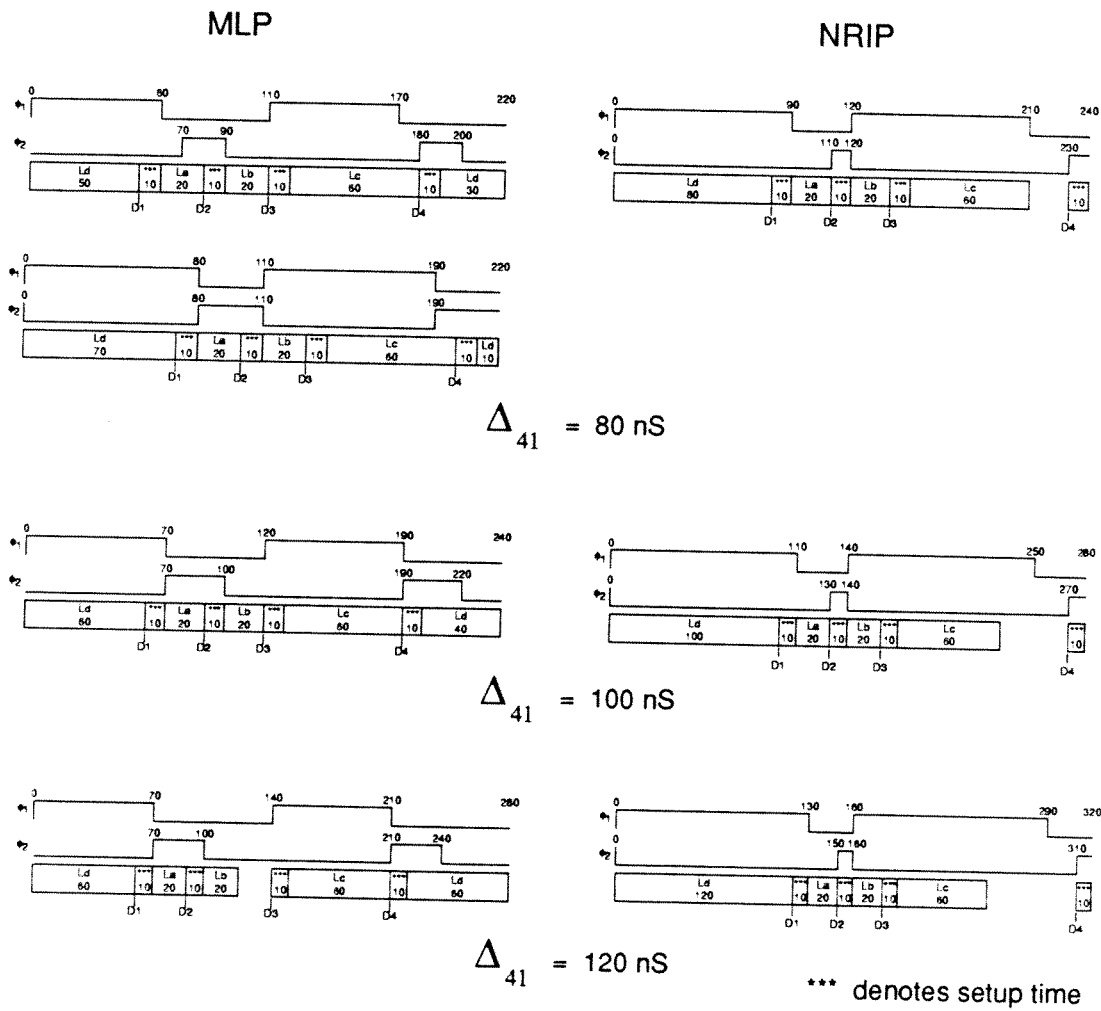


Figure 5: Timing diagrams for example 1.

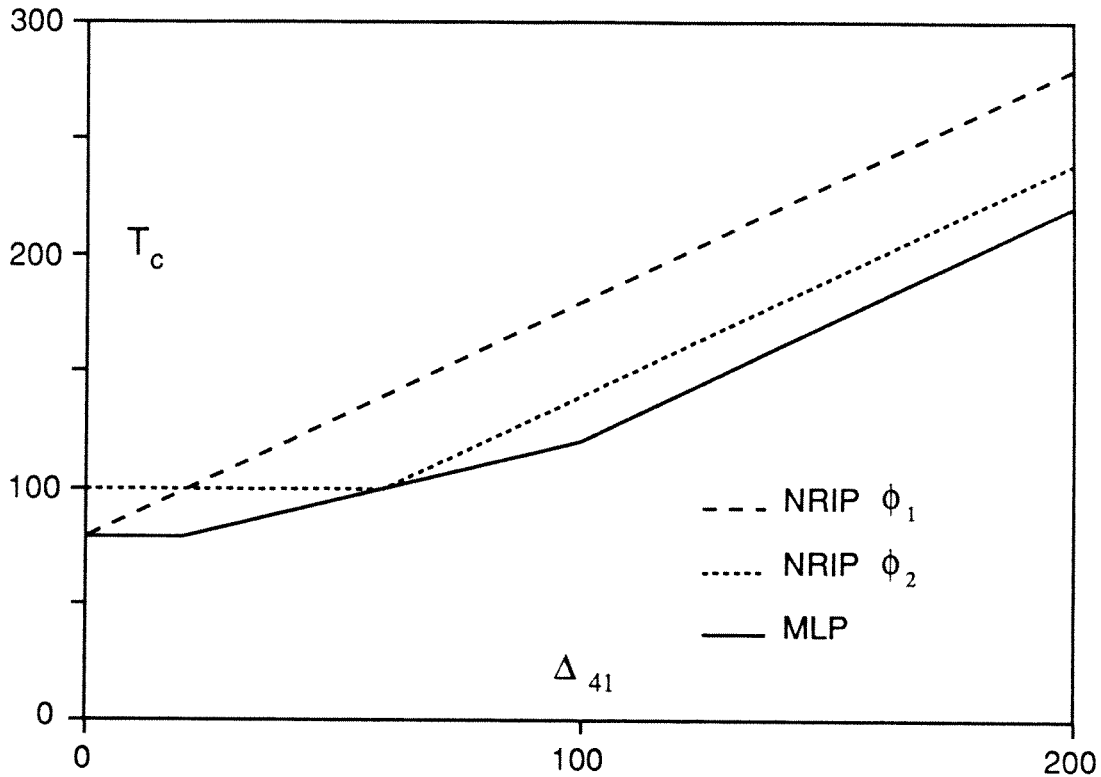


Figure 6:  $T_c$  vs.  $\Delta_{41}$  for example 1.

delays for each of the cycles making up the loop.

The cycle time calculations using the MLP and NRIP algorithms, for a more complicated example are shown in Figs. 7 and 8.

The following additional observations can be made:

- Unlike the previous example, the cycle time found by the NRIP algorithm is significantly higher (35%) than the optimal cycle time. While this result cannot be generalized for other circuits, it does point out that the approximate solution found by NRIP may deviate appreciably from the exact solution, and additional iterations

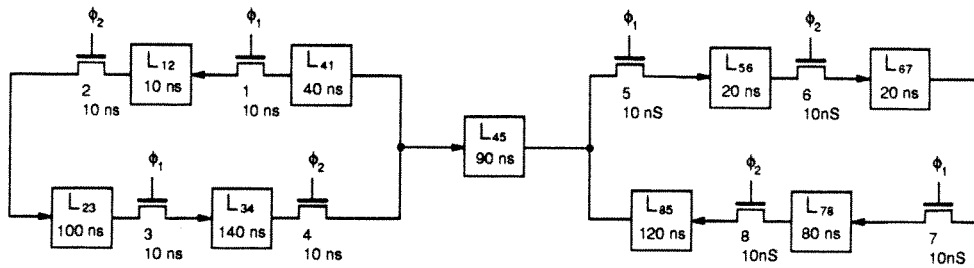


Figure 7: Block diagram for example 2.

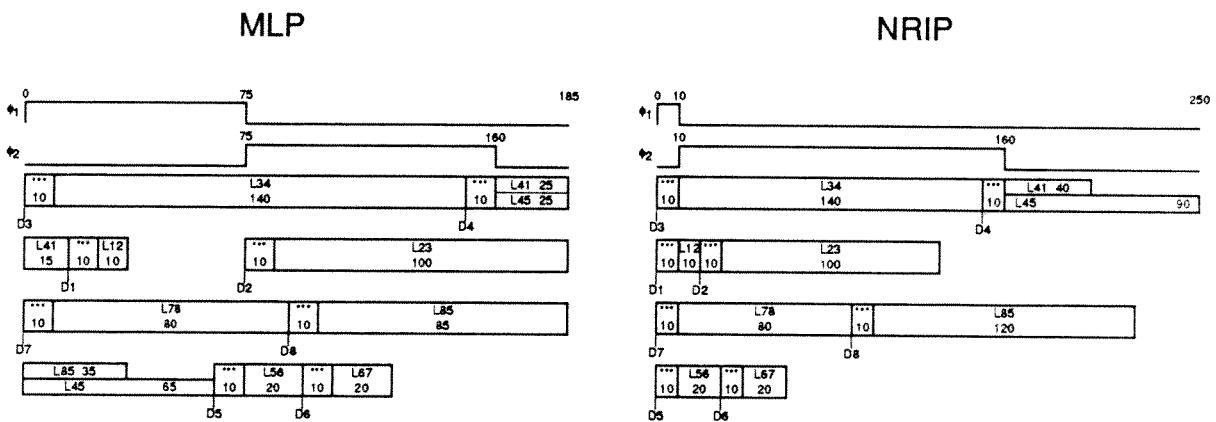


Figure 8: Timing diagrams for example 2.

might be necessary.

- Because of the coupling of the timing constraints through the feedback loops in the circuit as well as through the periodic clock signals, the notion of a *critical path* is clearly inadequate as a basis for discussing the optimality of the solution, and the dependence of that solution on the circuit's delays. Instead of a single critical path the circuit has several critical combinational delay *segments* which may be disjoint. The criticality of these segments, and the subcriticality of others, are directly related to associated slack variables in the inequality constraints. The techniques of parametric analysis in linear programming can be usefully applied here to study the effects of

varying the circuit delays on the optimal cycle time.

## 6 Conclusions

We have shown in this paper that the optimal cycle time for circuits controlled by level-sensitive latches can be determined by solving a linear program. The LP formulation provides a convenient theoretical foundation for analyzing such circuits and for developing algorithms which are potentially more efficient than the simplex algorithm. We are currently investigating just such algorithms, noting the fact that the entries of the constraint matrix for this problem are exclusively topological (i.e.  $0, \pm 1$ ). We also intend to interface the TV program [1] with an LP solver to study several VLSI chips currently being designed at the University of Michigan.

### Acknowledgements

We would like to thank Norm Jouppi for providing the MTV program and for explaining its “borrowing” algorithm.

## A Appendix

To illustrate the notation developed in Sec. 3, we present in this appendix the complete set of timing constraints for the circuit shown in Fig. 1. The circuit has 11 latches and is controlled by a 4-phase clock with the following  $K$  matrix:

$$K = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Thus there are 9 I/O-phase pairs; the corresponding phase-shift operators (used in the latch propagation constraints) are:

$$S_{13} = s_1 - s_3$$

$$S_{14} = s_1 - s_4$$

$$S_{21} = s_2 - s_1 - T_c$$

$$S_{23} = s_2 - s_3$$

$$S_{24} = s_2 - s_4$$

$$S_{31} = s_3 - s_1 - T_c$$

$$S_{32} = s_3 - s_2 - T_c$$

$$S_{42} = s_4 - s_2 - T_c$$

$$S_{43} = s_4 - s_3 - T_c$$

The timing constraints can now be stated as follows:

- Periodicity Constraints:

$$T_i \leq T_c, s_i \leq T_c \quad i = 1, 2, 3, 4$$

- Phase Ordering Constraints:

$$s_1 \leq s_2 \leq s_3 \leq s_4$$

- Phase Non-overlap Constraints:

$$\begin{aligned} s_1 &\geq s_3 + T_3 - T_c & s_2 &\geq s_1 + T_1 & s_3 &\geq s_1 + T_1 & s_4 &\geq s_2 + T_2 \\ s_1 &\geq s_4 + T_4 - T_c & s_2 &\geq s_3 + T_3 - T_c & s_3 &\geq s_2 + T_2 & s_4 &\geq s_3 + T_3 \\ s_2 &\geq s_4 + T_4 - T_c \end{aligned}$$

- Latch Setup Constraints:

$$D_i + \Delta_{DCi} \leq T_1 \quad i = 1, 2, 8$$

$$D_i + \Delta_{DCi} \leq T_2 \quad i = 6, 7, 11$$

$$D_i + \Delta_{DCi} \leq T_3 \quad i = 4, 5, 10$$

$$D_i + \Delta_{DCi} \leq T_4 \quad i = 3, 9$$

- Latch Propagation Constraints:

$$D_2 = \max(0, D_4 + \Delta_{DQ4} + \Delta_{42} + S_{31}, D_5 + \Delta_{DQ5} + \Delta_{52} + S_{31})$$

$$D_3 = \max(0, D_8 + \Delta_{DQ8} + \Delta_{83} + S_{14})$$

$$D_4 = \max(0, D_1 + \Delta_{DQ1} + \Delta_{14} + S_{13}, D_2 + \Delta_{DQ2} + \Delta_{24} + S_{13}, \\ D_3 + \Delta_{DQ3} + \Delta_{34} + S_{43})$$

$$D_5 = \max(0, D_6 + \Delta_{DQ6} + \Delta_{65} + S_{23}, D_7 + \Delta_{DQ7} + \Delta_{75} + S_{23})$$

$$D_6 = \max(0, D_4 + \Delta_{DQ4} + \Delta_{46} + S_{32}, D_5 + \Delta_{DQ5} + \Delta_{56} + S_{32})$$

$$D_7 = \max(0, D_9 + \Delta_{DQ9} + \Delta_{97} + S_{42}, D_{10} + \Delta_{DQ10} + \Delta_{10,7} + S_{32})$$

$$D_8 = \max(0, D_6 + \Delta_{DQ6} + \Delta_{68} + S_{21}, D_7 + \Delta_{DQ7} + \Delta_{78} + S_{21})$$

$$D_9 = \max(0, D_6 + \Delta_{DQ6} + \Delta_{69} + S_{24}, D_7 + \Delta_{DQ7} + \Delta_{79} + S_{24})$$

$$D_{10} = \max(0, D_{11} + \Delta_{DQ11} + \Delta_{11,10} + S_{23})$$

$$D_{11} = \max(0, D_9 + \Delta_{DQ9} + \Delta_{9,11} + S_{42}, D_{10} + \Delta_{DQ10} + \Delta_{10,11} + S_{32})$$

- Non-negativity Constraints:

$$T_c \geq 0$$

$$T_{p_i} \geq 0, s_{p_i} \geq 0 \quad p_i = 1, 2, 3, 4$$

$$D_i \geq 0 \quad i = 1, \dots, 11$$

## References

- [1] Norman P. Jouppi. Timing verification and performance improvement of MOS vlsi designs. Technical Report 84-266, Stanford University, 1984.
- [2] John K. Ousterhout. A switch level timing verifier for digital MOS VLSI. *IEEE Trans. Computer-Aided Design*, CAD-4(3):336-349, July 1985.
- [3] Michel R. Dagenais and Nicholas C. Rumin. On the Calculation of Optimal Clocking Parameters in Synchronous Circuits with Level-Sensitive Latches. *IEEE Trans. Computer-Aided Design*, 8(3):268-278, March 1989.
- [4] Don T. Phillips, A. Ravindran, and James J. Solberg. *Operations Research: Principles and Practice*. John Wiley & Sons, Inc., 1976.
- [5] Michel R. Dagenais. *Timing Analysis for MOSFETs: An Integrated Approach*. PhD thesis, Dept. of Electrical Engineering, McGill University, 1987.

- [6] T. I. Kirkpatrick and N. R. Clark. PERT as an aid to logic design. *IBM Journal of Research and Development*, 10(2):135–141, March 1966.
- [7] Vishwani D. Agrawal. Synchronous path analysis in MOS circuit simulator. In *Proc. ACM/IEEE Design Automation Conf.*, pages 629–635, 1982.
- [8] Stephen H. Unger and Chung-Jen Tan. Clocking schemes for high-speed digital systems. *IEEE Trans. Computers*, C-35(10):880–895, October 1986.
- [9] Thomas G. Szymanski. LEADOUT: A static timing analyzer for mos circuits. In *Proc. IEEE Conf. Computer-Aided Design*, pages 130–133, 1986.
- [10] Lance A. Glasser and Daniel W. Dobberpuhl. *The Design and Analysis of VLSI Circuits*. Addison Wesley, 1985.

