Synchronization of Pipelines

Karem A. Sakallah, *Senior Member, IEEE*, Trevor N. Mudge, *Senior Member, IEEE*, Timothy M. Burks, *Student Member, IEEE*, and Edward S. Davidson, *Fellow, IEEE*

Abstract—**In this paper we apply a recently formulated general timing model of synchronous operation to the special case of latch-controlled pipelined circuits. The model accounts for multiphase synchronous clocking, correctly captures the behavior of level-sensitive latches, handles both short- and long-path delays, accommodates wave pipelining, and leads to a comprehensive set of timing constraints. Pipeline circuits are important because of their frequent use in computer systems. We define their concurrency as a function of the clock schedule and degree of wave pipelining. We then identify a special class of clock schedules, *coincident multiphase clocks*, which provide a lower bound on the value of the optimum cycle time. We show that the region of feasible solutions for single-phase clocking can be nonconvex or even disjoint, and derive a closed-form expression for the minimum cycle time of a restricted but practical form of single-phase clocking. We compare these forms of clocking on three pipeline examples and highlight some of the issues in pipeline synchronization.**

List of Symbols

i, j	Indexes used to identify pipeline stages/synchronizers.
p, r	Indexes used to identify clock phases.
p_i	Index of clock phase used to control synchronizer i.
a_i, A_i	Early and late signal arrival times at stage i.
d_i, D_i	Early and late signal departure times from stage i.
δ_i, Δ_i	Minimum and maximum propagation delays from synchronizer $i-1$ to synchronizer i.
C	Concurrency in the pipeline.
e_p	Time, in global frame-of-reference, at which clock phase p ends (i.e., when its latching edge occurs).
E_{pr}	Phase shift from clock phase p to clock phase r.
$\mathcal{E}_{i-1,i}$	Phase shift from stage $i-1$ to stage i.
H_i	Hold time of synchronizer i.
k	Number of clock phases.
m	Width (in bits) of the pipeline datapath.
n	Number of pipeline stages.
ν_i	Degree of wave pipelining in stage i.
R_i^E	Region of feasibility corresponding to the early arrival (hold) constraints of pipe stage i.
R^L	Region of feasibility corresponding to the late arrival (setup) constraints.
R^P	Region of feasibility corresponding to the pulsewidth constraints.
S_i	Setup time of synchronizer i.
T_c	Clock cycle time.
T_p	Width of active interval of phase p.
U	Utilization of the pipeline.
ϕ_p	Name of clock phase whose index is p.
w	Minimum allowable clock pulsewidth.

I. Introduction

IN THIS PAPER we extend the work reported in [1] which applied a recently formulated general timing model of synchronous operation [2] to the special case of pipelined circuits. The model accounts for multiphase synchronous clocking, correctly captures the behavior of level-sensitive latches, handles both short and long paths, and leads to a comprehensive, yet simple, set of timing constraints. It has been successfully applied, for general circuit topologies, to the problems of clock cycle minimization using linear programming methods [3] and timing verification using an iterative relaxation algorithm [4].

By applying this model to a simple circuit structure, this paper helps to clarify various aspects of the clocking of level-sensitive latches as a function of circuit propagation delays.[1] These include the following:

- Defining pipeline concurrency as a function of the clock schedule and the degree of wave pipelining.
- Identifying a special class of clock schedules, *coincident multiphase clocks*, which yield the smallest possible cycle times for a specified degree of wave pipelining.
- Demonstrating that the space of physically realizable single-phase clock schedules derived from this model can be nonconvex or even disjoint, complicating the search for the optimal cycle time.

Manuscript received February 19, 1991; revised November 9, 1992. This work was supported in part by the NSF under Grant MIP-9014058. This paper was recommended by Associate Editor K. Keutzer.

The authors are with the Advanced Computer Architecture Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122.

IEEE Log Number 9207266.

[1]Using edge-triggered flip-flops rather than level-sensitive latches yields a much simpler model that is considerably easier to analyze. Latches, however, are interesting for two reasons: first, they hold the promise of lower cycle times because they allow their data signals to "flow through" unhindered during the active interval of their controlling clock phase; and second, they are typically "cheaper" than flip-flops as measured by gate count or area.

- Providing a closed-form solution for the optimal cycle time of a restricted, but practical, form of single-phase clocking.

In addition to helping understand the above issues, the study of pipelines is further justified by their increasing use, even in the instruction execution units of single-chip microprocessors (commonly referred to as "super-pipelining" in RISC machines) [5]. In particular, the above-mentioned closed-form expression for minimum cycle time may be directly applied in the design and optimization of high performance processors which use single-phase clocking.

The remainder of this paper is organized as follows. The pipeline model is developed in Section II. In Section III we examine the dependence of pipeline concurrency on clocking and wave pipelining, and define coincident multiphase clocks. In Section IV we derive the regions of feasibility for coincident multiphase clocking and for two modes of single-phase clocking. Section V illustrates the application of the model on three example pipelines using an experimental program, *pipe*T_c, which computes the optimal clock schedules for single-phase as well as coincident multiphase clocking. Conclusions and suggestions for future work are summarized in Section VI.

II. Pipeline Model

Pipelining is frequently used to speed up the execution of a sequence of computations by dividing each into n consecutive subcomputations and overlapping their execution. Theoretically, this should yield a factor of n performance improvement over the nonpipelined case. This maximum is rarely achieved, however, because of dependencies among the operations and overhead due to clocking [6]. Performance can be defined as the sustained number of operations per unit time, and can be expressed as:

$$MOPS = \frac{U(n) \times 10^3}{T_c(n)}$$

where *MOPS* stands for millions of operations per second, $0 \leq U(n) \leq 1$ is the *utilization* of the n-stage pipeline, and $T_c(n)$ is the clock cycle time, in nanoseconds, at each pipe stage. Typically, $U(n)$ is a decreasing function of n which is determined empirically through simulations or benchmarking. $T_c(n)$ is also a decreasing function of n that depends on circuit delays and clocking parameters. Optimal pipeline design seeks to find the value of n which maximizes *MOPS*. This is usually done in two steps. 1) Determining $U(n)$ for a suitable range of n by analyzing the dependencies among the operations of an appropriate set of benchmark computations. This is a purely "architectural" analysis which disregards hardware implementation details, but may consider software restructuring to decrease the dependence effects. 2) Determining the minimum $T_c(n)$ for the same range of n. Generally, this is a synthesis problem which involves examining the logic design of various pipelines, and finding those that yield the minimum cycle times.

This paper addresses one aspect of the second step, namely, determining the minimum cycle time, $T_{c,min}$, for an n-stage pipeline in terms of circuit delays. The problem has been addressed previously by a number of authors including [6]–[9]. This previous work dealt mostly with simple clocking paradigms. Furthermore, the analysis was typically based on examination of a single pipe stage. In contrast, in this paper we propose a pipeline timing model that accounts for more complex clocking and for the temporal interactions among the various pipe stages.

Our pipeline model is shown in Fig. 1. The pipe stages are numbered consecutively from 0 to $(n - 1)$. The datapath through the pipeline is assumed to be m bits wide, $m \geq 1$. Each pipe stage consists of a bank of m level-sensitive latches used as synchronizing elements followed by combinational circuitry.[2] Data flow through the pipeline is regulated by a k-phase clock, where $1 \leq k \leq n$. Stage i is characterized by the following parameters:

- p_i: an integer denoting the clock phase used to control the synchronizing element at the output of stage i (henceforth referred to as synchronizer i).
- S_i: nonnegative setup time of synchronizer i relative to latching edge of phase p_i.
- H_i: nonnegative hold time of synchronizer i relative to latching edge of phase p_i.
- δ_i, Δ_i: minimum and maximum propagation delays ($0 \leq \delta_i \leq \Delta_i$) from the input of synchronizer $i - 1$ to the input of synchronizer i. Note that this definition of stage delay lumps together the two components of signal delay, namely the synchronizer delay and the combinational logic delay.

Note that, unlike earlier open-ended pipeline formulations, such as those given in [6]–[9], our model includes a virtual pipe stage, labeled 0, which forms a simple loop with stages 1 through $n - 1$. Stage 0 is used to model the times of data arriving from, and data departing to, the environment surrounding the pipeline. For example, it can be used to model the timing attributes of the memory or register file used to supply operands for the computation, and to receive results from it. The use of such a virtual stage provides a consistent mechanism to account for the boundary conditions of pipeline operation. Furthermore, as shown in Section 2.3, the open-ended pipeline is a special case of the closed pipeline.

We base the steady-state behavior of such pipelines on the general model of synchronous operation introduced in [2]. The salient features of this model, as they relate to the pipeline, are summarized below. In addition, we extend the model to allow for *wave pipelined* [10], [11] op-

[2]The value of m may vary from stage to stage and actually has no effect on the model.

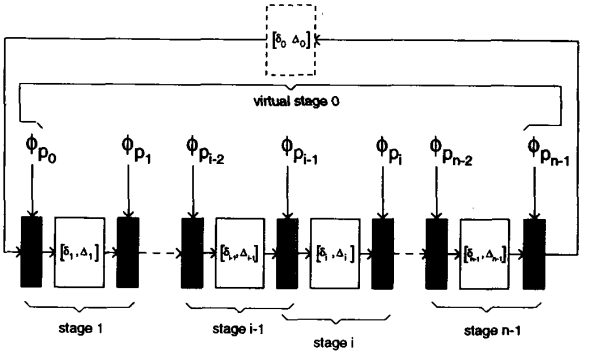


Fig. 1. n-stage pipeline. Shaded boxes represent the synchronizers.

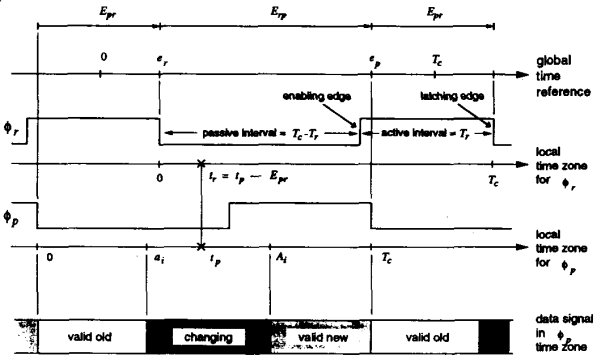


Fig. 2. Key model parameters.

eration. Fig. 2 depicts the relationships among the key parameters used in the model.

2.1. Clocking Model

The clocking model is described in terms of a *temporal* rather than a *logical* framework based on the concept of *periodic phases* which define *local time zones* related by *phase shift operators*. In this model, a k-phase clock is considered to be a collection of k periodic signals $\phi_1, \phi_2, \cdots, \phi_k$—referred to as the *phases*—with a common cycle time T_c. Each phase ϕ_p divides the clock cycle into two intervals: an *active* interval of duration T_p, and a *passive* interval of duration $(T_c - T_p)$. During the active interval of a given phase, the synchronizers it controls are *enabled*; during its passive interval, they are *disabled*. The transitions into and out of the active interval are called, respectively, the *enabling* and *latching* edges of the phase. We assume, without loss of generality, that all phases are active high; thus, the enabling and latching edges correspond to the rising and falling transitions of the phase signal. Associated with the phase is a *local time zone* such that the passive interval of the phase starts at $t = 0$, its enabling edge occurs at $t = T_c - T_p$, and its latching edge occurs at $t = T_c$. The temporal relationships among the k phases (i.e., among the different time zones) are established by an arbitrary choice of a *global time reference*. We introduce e_p to denote the time, relative to this global time reference, at which phase ϕ_p *ends* (i.e., when its latching edge occurs). Finally, we define a *phase shift* operator:

$$E_{pr} \equiv \begin{cases} (e_r - e_p), & e_r > e_p \\ (T_c + e_r - e_p), & e_r \le e_p \end{cases} \tag{1}$$

which takes on positive values in the range $(0, T_c]$. When subtracted from a time variable, t_p, in the *current* local time zone of ϕ_p, E_{pr} changes the frame of reference to the *next* local time zone of ϕ_r, taking into account a possible cycle boundary crossing.

2.2. Timing Constraints

For timing purposes, it is sufficient to characterize a data signal with respect to one clock cycle by two, possibly simultaneous, events which demark the interval during which the signal is switching between its old and new values. For the signal *arriving* at the input of synchronizer i these two events are defined to occur at $t = a_i$ and $t = A_i$ in the local time zone of phase p_i. The corresponding events of the data signal *departing* from the input of synchronizer i are defined to occur at $t = d_i$ and $t = D_i$. It will be convenient to refer to a_i and A_i as the *early* and *late* arrival times, and to d_i and D_i as the *early* and *late* departure times. The timing model of the pipeline can now be expressed by the following constraints and equations [2] for $i = 0, \cdots, n - 1$.

Clock Constraints express limitations on clock generation and distribution. This set should at least include the following minimum pulsewidth constraints:

$$T_{p_i} \ge w_{p_i} \tag{2}$$

$$T_c - T_{p_i} \ge w_{p_i} \tag{3}$$

where w_{p_i} are specified pulse width parameters. In addition, to simplify the design of the clock generator we may include "regularity" constraints such as

$$T_1 = T_2 = \cdots = T_k. \tag{4}$$

It is important to point out that the phase signals are not required to be nonoverlapping.

Latching Constraints express the conditions necessary for capturing valid data values at each of the synchronizers. They consist of two sets of requirements which, together, insure that the data signal at the input of a synchronizer is stable for a sufficient period of time before and after the occurrence of the latching edge of the corresponding clock. Mathematically,

$$a_i \ge H_i \tag{5}$$

$$A_i \le T_c - S_i. \tag{6}$$

Synchronization Equations *macromodel* the temporal behavior of different types of synchronizing elements. Specifically, for D-type level-sensitive latches, they express the departure times of each output data signal as the later of the arrival time of the corresponding input data

signal and the enabling clock edge:

$$d_i = \max (a_i, T_c - T_{p_i}) \quad (7)$$

$$D_i = \max (A_i, T_c - T_{p_i}). \quad (8)$$

Propagation Equations model the delay of the combinational stages in the pipeline, including the propagation through the input synchronizer. They express the arrival times of data at the input of synchronizer i in terms of the corresponding departure times from the input of synchronizer $(i - 1) \bmod n$, taking into account the change in the frame-of-reference from phase p_{i-1} to phase p_i:[3]

$$a_i = d_{i-1} + \delta_i - \mathcal{E}_{i-1,i} \quad (9)$$

$$A_i = D_{i-1} + \Delta_i - \mathcal{E}_{i-1,i} \quad (10)$$

where $\mathcal{E}_{i-1,i}$ is the amount of phase shift *from stage* $i - 1$ *to stage* i. In [2], this was defined to be equal to the phase shift *from clock phase* p_{i-1} to *clock phase* p_i, i.e., $\mathcal{E}_{i-1,i} \equiv E_{p_{i-1}p_i}$. This definition limited signal propagation to consecutive cycles of phases p_{i-1} and p_i, i.e., signals launched from stage $i - 1$ in any given cycle of phase p_{i-1} had to arrive and be correctly latched at stage i by the immediately following cycle of phase p_i. We extend this definition here to allow for signal propagation over multiple clock cycles by introducing the nonnegative integer parameter ν_i to indicate the number of *additional* clock cycles available for signals to propagate from stage $i - 1$ to stage i. Thus,

$$\mathcal{E}_{i-1,i} \equiv E_{p_{i-1}p_i} + \nu_i T_c. \quad (11)$$

Note that the addition of an integer number of clock cycles to the clock phase shift has the effect of changing the frame-of-reference from the current local time zone of phase p_{i-1} to the local time zone of phase p_i that begins ν_i cycles *after* its next local time zone. In particular, for $\nu_i = 0$ the phase shift reverts to its earlier definition.

2.3. *Open-Ended Pipelines*

Characterizing open-ended pipelines using the above model is a simple matter of replacing the departure time equations for virtual stage 0 with *specified* values that represent the pipeline boundary conditions. Specifically, the following equations for signal departure times from stage 0

$$d_0 = \max (a_0, T_c - T_{p_0}) \quad (12)$$

$$D_0 = \max (A_0, T_c - T_{p_0}) \quad (13)$$

are simply replaced by

$$d_0 = \hat{d}_0 \quad (14)$$

$$D_0 = \hat{D}_0 \quad (15)$$

[3]Index arithmetic in what follows will always be modulo n. To keep the equations from becoming too cluttered, the mod operator will be dropped and assumed to be implied.

where $\hat{d}_0$, and $\hat{D}_0$ denote the specified signal "departure" times from the pipeline source to its first stage. This immediately leads to the following arrival time equations at the first pipe stage:

$$a_1 = \hat{d}_0 + \delta_1 - \mathcal{E}_{0,1} \quad (16)$$

$$A_1 = \hat{D}_0 + \Delta_1 - \mathcal{E}_{0,1}. \quad (17)$$

The equations for signal arrival times at virtual stage 0

$$a_0 = d_{n-1} + \delta_0 - \mathcal{E}_{n-1,0} \quad (18)$$

$$A_0 = D_{n-1} + \Delta_0 - \mathcal{E}_{n-1,0} \quad (19)$$

which capture the signal propagation delays through the pipeline "environment" are dropped altogether. Instead, the corresponding *actual* signal departure times computed from the model equations:

$$d_{n-1} = \max (a_{n-1}, T_c - T_{p_{n-1}}) \quad (20)$$

$$D_{n-1} = \max (A_{n-1}, T_c - T_{p_{n-1}}) \quad (21)$$

are checked against the *required* signal departure times, $\hat{d}_{n-1}$ and $\hat{D}_{n-1}$ from the last pipe stage (stage $n - 1$) to the pipeline environment.

The specification of signal times entering stage 1 and leaving stage $n - 1$ represents a decoupling of the signal propagation equations around the closed pipeline and leads to an easier cycle time optimization problem. However, by explicitly including virtual stage 0 in the pipeline model, we have the added flexibility of optimizing the operation of the pipeline within its environment. Either way, it should be clear that the closed pipeline model above encompasses open-ended pipelines as a degenerate special case. The remainder of the paper focuses on studying closed pipelines.

III. Pipeline Operation Modes

Allowing multiple clock cycles for signals to propagate through a single stage has the potential of reducing the cycle time below what is possible with single-cycle propagation. However, for such operation to be feasible the minimum combinational delay of the stage must be sufficiently large to maintain adequate temporal separation between consecutive *waves* of signals (see Fig. 3). Reliance on logic delay, rather than on synchronizing elements alone, to prevent interference between consecutive data waves has been dubbed *wave pipelining* [10]. This phenomenon will occur in any pipe stage for which $\nu_i > 0$. We thus refer to ν_i as the *degree of wave pipelining* in stage i.

The number of operations concurrently in process in an n-stage pipeline need not be equal to n. Depending on the nature of the clocking scheme, the differences between the minimum and maximum delays in each stage, and the distribution of the maximum delays over all stages, it may be possible to operate the pipeline so that the number of signal waves simultaneously traveling around the closed pipeline is less than or greater than n. We capture this

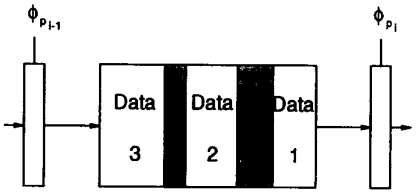


Fig. 3. Wave pipelining.

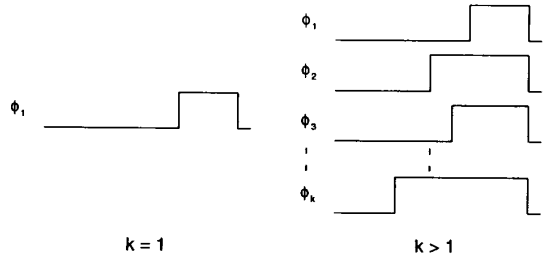


Fig. 4. Clocks with maximum possible phase shift between phases.

notion by introducing C, the *concurrency* in the pipeline, which can easily be related to the clock phase shifts and the degrees of wave pipelining by

$$C = \frac{1}{T_c} \sum_{i=0}^{n-1} E_{p_{i-1}p_i} + \sum_{i=0}^{n-1} \nu_i. \tag{22}$$

C can be thought of as the number of *virtual* pipeline stages. Note that in a closed pipeline C must be an integer; hence $\Sigma E_{p_{i-1}p_i}$ must be an integer multiple of T_c. A particular level of concurrency may be achieved by a variety of combinations of clocking schemes and wave pipelining. For example, a concurrency of 4 in a 4-stage pipeline may be obtained by a 4-phase clock where each pipe stage is allocated a fraction of the clock cycle such that $\Sigma E_{p_{i-1}p_i} = T_c$, and $\Sigma \nu_i = 3$. Alternatively, $\Sigma E_{p_{i-1}p_i} = 2T_c$, and $\Sigma \nu_i = 2$.

We limit our attention in this paper to those clocking schemes which maximize C for a given level of wave pipelining, namely those for which the sum of the clock phase shifts around the pipeline stages is equal to nT_c. Recalling that each of the individual phase shifts is at most one clock cycle, this restriction implies that $E_{p_{i-1}p_i} = T_c$ for each of the n stages. Clocking schemes for which this restriction applies include single-phase clocks and the restricted form of multiphase clocking shown in Fig. 4, which will be referred to as *coincident* multiphase clocking since the latching edges of all k phases *coincide* in time.[4] For simplicity in the equations and analysis that follows we let $\nu_i = \nu$ for all stages. The methods used, however, do handle the general case where ν_i differs from stage to stage.

With these restrictions, the concurrency C becomes

$$C = (1 + \nu)n. \tag{23}$$

[4]For general multiphase clocks, the existence of fractional phase shifts (i.e., phase shifts smaller than a full cycle) limits $\Sigma E_{p_{i-1}p_i}$ to $\leq (n - 1)$ clock cycles.

IV. Optimal Cycle Time Calculation

Subject to the simplifying assumptions made above, namely $E_{p_{i-1}p_i} = T_c$ and $\nu_i = \nu$, the phase shift from stage $i - 1$ to stage i in (11) can be expressed simply as

$$\mathcal{E}_{i-1,i} \equiv (1 + \nu)T_c. \tag{24}$$

The timing model of the pipeline can now be conveniently viewed as consisting of *three* distinct sets of constraints:

- **Pulsewidth Constraints** expressed by (2) and (3).
- **Long-Path** (*Late-Signal*) **Constraints** involving the late arrival and departure times and expressed by the setup inequalities (6), the propagation equations (10), and the synchronization equations (8).
- **Short-Path** (*Early-Signal*) **Constraints** involving the early arrival and departure times and expressed by the hold inequalities (5), the propagation equations (9), and the synchronization equations (7).

Subject to the above constraints, we outline in this section procedures for obtaining the minimum cycle time for latch-controlled pipelines for the following three clocking schemes:

1) a coincident n-phase clock,
2) a general form of single-phase clocking,
3) a restricted form of single-phase clocking.

In all three cases, the calculation of the optimal cycle time starts by finding expressions for the early and late arrival times at stage i in terms of the clock variables and circuit delays. These expressions are then combined with the hold and setup requirements to obtain the short- and long-path constraints. In one case, restricted single-phase clocking, these constraints can be solved to yield a closed-form expression for the minimum cycle time. Numerical solutions are necessary in the other two cases.

It should be noted that single-phase clocks are a special case of the more general coincident n-phase clocks. As such, the minimum cycle time possible with a coincident n-phase clock will always be less than or equal to that obtainable with a single-phase clock. Less obvious, though, is the fact that the solution space for the case of coincident n-phase clocks is convex whereas that for single-phase clocks may in fact be nonconvex or even disconnected. While we do not envision that coincident n-phase clocks are likely to be used in practice, their study is theoretically important because they provide a lower bound on the minimum cycle times possible with single-phase clocks.

4.1. Coincident n-phase Clocks

A coincident n-phase clock is obtained by setting $p_i = i$ and is characterized by $n + 1$ variables: the cycle time T_c, and the n independent phase widths $T_0, \cdots, T_{n-1}$. It is important to note that the freedom to choose a different phase width for each pipe stage is the key to the relatively simple solution procedure of the coincident n-phase case. In particular, it is always possible to choose

the phase widths so that the synchronization equations (7) and (8) are simplified to:

$$d_i = D_i = T_c - T_i \tag{25}$$

This simplification can be justified as follows:

> Suppose that $D_i > T_c - T_i$ for some stage i at the optimal solution. Then $D_i = A_i > T_c - T_i$. Since changing T_i can only directly affect the departure times from stage i, it should be obvious that T_i can be decreased until $D_i = A_i = T_c - T_i$ without affecting the optimal cycle time. Note also that decreasing T_i can only increase the margin by which the hold requirement is satisfied at stage $i + 1$.

The above simplification is significant because it removes the coupling, inherent in the latch synchronization model, between the departure and arrival times. As will become apparent later, this coupling is the primary source of complexity and nonconvexity in the general single-phase case. Specifically, the optimality of the coincident n-phase solution is unchanged if we replace the synchronization equations (7) and (8) and their troublesome max function with the simple equalities (25). This in turn makes it possible to express the feasible region as a set of linear inequalities that define a convex space.

Arrival Times: Substituting (24) and (25) in (9) and (10), we can express the arrival times at stage i as:

$$\begin{aligned} a_i &= T_c - T_{i-1} + \delta_i - (1 + \nu)T_c \\ &= \delta_i - T_{i-1} - \nu T_c \end{aligned} \tag{26}$$

and

$$\begin{aligned} A_i &= T_c - T_{i-1} + \Delta_i - (1 + \nu)T_c \\ &= \Delta_i - T_{i-1} - \nu T_c. \end{aligned} \tag{27}$$

In addition, from (8), signals must arrive at the latest by the rising edge of the corresponding clock to satisfy (25):

$$A_i \leq T_c - T_i. \tag{28}$$

Long-Path Constraints: Combining (27) with the setup requirement (6), yields

$$(1 + \nu)T_c + T_{i-1} \geq \Delta_i + S_i. \tag{29}$$

Combining (27) with (28) leads to another constraint:

$$(1 + \nu)T_c + T_{i-1} - T_i \geq \Delta_i. \tag{30}$$

Short-Path Constraints: Substituting (26) into the hold requirement (5) yields

$$\nu T_c + T_{i-1} \leq \delta_i - H_i. \tag{31}$$

Solution Procedure: The feasible region for coincident n-phase clocking is defined in the $(n + 1)$-dimensional space of clock variables by $5n$ linear inequalities:

- $2n$ long-path inequalities (29) and (30),
- n short-path inequalities (31),
- $2n$ minimum pulse-width inequalities (2) and (3).

The minimum cycle time can be now be found by solving a linear program. Note that if $T_i \geq S_i$, as might be required for certain types of latches, (29) is subsumed by (30) and the total number of constraints in the linear program can be reduced to $4n$.

4.2. *General Single-Phase Clock*

When the clock phase widths at all pipe stages are forced to be equal, it may no longer be possible to satisfy the simplified latch synchronization equation (25); instead, the general model equations (7) and (8) must be invoked. It is possible under these conditions for some early signals to simply flow through the latches without having to wait for the enabling clock edge (i.e., $a_i > T_c - T_{pi}$), effectively rendering the latches redundant. Such an operation mode has been termed "aggressive" [3] since it allows the latches to be transparent not only for the slow signals but also for the fast signals. In this case, the space of feasible solutions may become nonconvex. If we denote the feasible regions corresponding to pulse-width constraints by R^P, long-path (late-signal) constraints by R^L, and short-path (early-signal) constraints by R^E, then the overall region of feasibility is simply $R^P \cap R^L \cap R^E$. These regions are shown in Fig. 5 and are derived next, except for R^P which follows trivially from (2) and (3).

Arrival Times: The solution in the case of a single-phase clock is considerably more complicated because of the coupling between signal arrival and departure times through the latch synchronization equations (7) and (8). Unlike the coincident n-phase case, obtaining an expression for the arrival time at stage i requires the substitution of the synchronization and propagation equations of *all* pipe stages. Thus, the early arrival time at stage i is calculated by repeated application of (9) and (7) and algebraic simplification. Setting $p_i = 1$ to represent a single-phase clock, we obtain the following expression for the early arrival time at the input to synchronizer i:

$$\begin{aligned} a_i &= d_{i-1} + \delta_i - (1 + \nu)T_c \\ &= \max(a_{i-1}, T_c - T_1) + \delta_i - (1 + \nu)T_c \\ &= \max(a_{i-1} + \delta_i - (1 + \nu)T_c, \delta_i - T_1 - \nu T_c) \\ &= \max(d_{i-2} + \delta_{i-1} + \delta_i - (2 + 2\nu)T_c, \\ &\qquad \delta_i - T_1 - \nu T_c) \\ &= \max(\max(a_{i-2}, T_c - T_1) + \delta_{i-1} + \delta_i \\ &\quad - (2 + 2\nu)T_c, \delta_i - T_1 - \nu T_c) \\ &= \max(a_{i-2} + \delta_{i-1} + \delta_i - (2 + 2\nu)T_c, \delta_{i-1} \\ &\quad + \delta_i - T_1 - (1 + 2\nu)T_c, \delta_i - T_1 - \nu T_c) \\ &= \cdots \\ &= \max(a_i + \delta_{i-n+1} + \cdots + \delta_i - (n + n\nu)T_c, \\ &\qquad \delta_{i-n+1} + \cdots + \delta_i - T_1 \\ &\quad - (n - 1 + n\nu)T_c, \\ &\qquad \cdots, \delta_{i-1} + \delta_i - T_1 - (1 + 2\nu)T_c, \\ &\qquad \delta_i - T_1 - \nu T_c) \end{aligned}$$

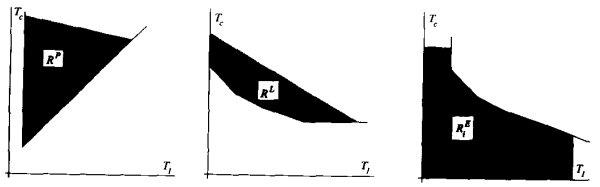


Fig. 5. Single-phase feasible regions for latches (illustrated for $\nu = 0$). (a) Pulsewidth constraints. (b) Long-path constraints. (c) Short-path constraints for stage i.

which can be expressed more conveniently as:

$$a_i = \max\left\{a_i + \left(\sum_{j=i-n+1}^{i} \delta_j\right) - (n + n\nu)T_c, \cdot \max_{0 \le l \le (n-1)}\left[\left(\sum_{j=i-l}^{i} \delta_j\right) - T_1 - (l + \nu + l\nu)T_c\right]\right\} \tag{32}$$

Similarly, the late arrival time at stage i, calculated from (10) and (8), is:

$$A_i = \max\left\{A_i + \left(\sum_{j=i-n+1}^{i} \Delta_j\right) - (n + n\nu)T_c, \cdot \max_{0 \le l \le (n-1)}\left[\left(\sum_{j=i-l}^{i} \Delta_j\right) - T_1 - (l + \nu + l\nu)T_c\right]\right\} \tag{33}$$

Note that the max functions in these expressions involve $n + 1$ arguments in which, except for the first argument, the only variables are the two clock variables T_c and T_1.

Long-Path Constraints: Expression (33) implies the following $n + 1$ inequalities:

$$A_i \ge A_i + \left(\sum_{j=i-n+1}^{i} \Delta_j\right) - (n + n\nu)T_c \tag{34}$$

$$A_i \ge \left[\left(\sum_{j=i-l}^{i} \Delta_j\right) - T_1 - (l + \nu + l\nu)T_c\right], \quad l = 0, \cdots, n-1. \tag{35}$$

Eliminating A_i from the first inequality, we immediately obtain the following lower bound on T_c:

$$T_c \ge \frac{1}{n(1+\nu)} \sum_{j=i-n+1}^{i} \Delta_j = \frac{1}{n(1+\nu)} \sum_{j=0}^{n-1} \Delta_j \equiv \frac{\overline{\Delta}}{1+\nu} \tag{36}$$

which confirms the intuition that the cycle time cannot be less than the average pipeline stage delay, $\overline{\Delta}$, when $\nu = 0$. In general, the C clock cycles during which one signal wave completes its tour of the pipeline must not comprise less total time than the sum of the maximum propagation delays around the pipeline.

Combining each of the remaining n inequalities with the setup constraint (6) we eliminate A_i to obtain:

$$(1 + l)(1 + \nu)T_c + T_1 \ge \left(\sum_{j=i-l}^{i} \Delta_j\right) + S_i, \quad l = 0, \cdots, n-1. \tag{37}$$

While the physical interpretation of each of these inequalities is not as obvious as that of (36), it is still rather simple: the time available for a signal to propagate down the $(l + 1)$ pipe stages ending at stage i, and to be correctly setup for latching at stage i, is $(1 + l)(1 + \nu)$ clock cycles plus the phase width T_1 which represents the "extra" time due to the use of level-sensitive latches. Since each of these inequalities must be true for all n pipe stages, we obtain:

$$(1 + l)(1 + \nu)T_c + T_1 \ge \max_{0 \le i \le (n-1)}\left[\left(\sum_{j=i-l}^{i} \Delta_j\right) + S_i\right], \quad l = 0, \cdots, n-1. \tag{38}$$

Thus the long-path constraints have been reduced to the $n + 1$ inequalities in (36) and (38) which together define a convex set in the T_c/T_1 solution space as shown in Fig. 5(b).

Short-Path Constraints: Proceeding as we did for the late arrival time at stage i, we obtain the following inequalities that must be satisfied by the early arrival time:

$$a_i \ge a_i + \left(\sum_{j=i-n+1}^{i} \delta_j\right) - (n + n\nu)T_c \tag{39}$$

$$a_i \ge \left[\left(\sum_{j=i-l}^{i} \delta_j\right) - T_1 - (l + \nu + l\nu)T_c\right], \quad l = 0, \cdots, n-1. \tag{40}$$

The first of these is redundant since it is subsumed by the corresponding max-delay inequality (34). The remaining n inequalities in (40) may now be combined with the hold requirement (5) to eliminate a_i and yield the set of short-path constraints. A convenient way to obtain this set is to derive its complement, namely the set of constraints under which the hold requirements are *violated*, and then to invoke De Morgan's Law. Specifically, the hold violation region for stage i is defined by the set of n inequalities:

$$H_i > a_i \ge \left[\left(\sum_{j=i-l}^{i} \delta_j\right) - T_1 - (l + \nu + l\nu)T_c\right], \quad l = 0, \cdots, n-1 \tag{41}$$

which, upon elimination of a_i, leads to the following n hold violation conditions:

$$(l + \nu + l\nu)T_c + T_1 > \left(\sum_{j=i-l}^{i} \delta_j\right) - H_i, \quad l = 0, \cdots, n-1. \tag{42}$$

Introducing $\overline{R}_i^E$ to represent the hold violation region for stage i, where E stands for *early signal* (short path), (42) can equivalently be expressed as

$$\overline{R}_i^E = \overline{R}_{i,0}^E \cap \overline{R}_{i,1}^E \cap \cdots \cap \overline{R}_{i,l}^E \cap \cdots \cap \overline{R}_{i,n-1}^E \tag{43}$$

where $\overline{R}_{i,l}^E$ denotes the region of feasibility for the lth inequality in (42). By applying DeMorgan's Law to the set intersection equation (43), we obtain

$$R_i^E = R_{i,0}^E \cup R_{i,1}^E \cup \cdots \cup R_{i,l}^E \cup \cdots \cup R_{i,n-1}^E \tag{44}$$

Thus, the desired set of short-path constraints is

$$(l + \nu + l\nu)T_c + T_1 \leq \left(\sum_{j=i-l}^{i} \delta_j\right) - H_i$$

$$\text{for } \textit{at least} \text{ one } l \in \{0, \cdots, n-1\} \tag{45}$$

Note that, unlike the corresponding long-path inequalities (38) which must all be satisfied, the above set of n short-path inequalities is satisfied if *at least one* of them is satisfied. In other words, the feasible region defined by the set of n inequalities in (38) is the *intersection* of n separate (linear bounded) convex regions, whereas that defined by the inequalities in (45) is the *union* of n separate (linear bounded) convex regions. This in turn implies that while the region defined by (38) is guaranteed to be convex, the region defined by (45), for each i, is guaranteed to be nonconvex, as shown in Fig. 5(c).

Solution Procedure: Denoting the overall region of feasibility by R, it can be conveniently expressed as:

$$R = R^P \cap R^L \cap R_0^E \cap \cdots \cap R_{n-1}^E. \tag{46}$$

Due to the nonconvexity of R_i^E, R may be nonconvex or even disconnected. Examples of these cases are illustrated in Section V. In any case, assuming that $R \neq \phi$, at the optimal solution one or more of the long-path constraints (36) and (38) must be active (satisfied as an equation). This observation forms the basis for a directed-search algorithm to find the minimum cycle time. Basically, the search begins by finding the smallest possible cycle time that satisfies the minimum pulsewidth and long-path constraints ($R^P \cap R^L$). Except for the degenerate case where the vertex of R^P lies in R^L, this point corresponds to the intersection of $T_c - T_1 = w_1$ and one of the $n + 1$ long-path constraints. This solution is now examined to see if it satisfies all of the short-path constraints. If it does, then it is optimal, otherwise we "climb" up the lower periphery of R^L until either we satisfy all the short-path constraints, or we reach the other end of the minimum pulse width region ($T_1 = w_1$) without satisfying all the short-path constraints. If the latter obtains, the problem is infeasible. A detailed description of the geometric solution approach outlined here can be found in [12].

4.3. Restricted Single-Phase Clock

A conservative application of single-phase clocking is to require that no hold times be violated even if the early signal departure from each latch occurred at the earliest possible time. This is equivalent to conservatively assigning d_i to its worst case value by using $d_i = T_c - T_1$ in place of the general early signal synchronization equation (7) even when $a_i > T_c - T_1$. This restriction restores convexity to the region of feasible solutions and leads to a closed-form expression for minimum cycle time.

Specifically, since the $d_i = T_c - T_1$ part of (25) is satisfied, the short-path constraints can be obtained from (31) by first setting $T_{i-1} = T_1$, leading to

$$\nu T_c + T_1 \leq \delta_i - H_i \tag{47}$$

which must be satisfied for all i, resulting in

$$\nu T_c + T_1 \leq \min_{0 \leq i \leq (n-1)} (\delta_i - H_i) \tag{48}$$

which corresponds to a convex region. Notice that this simplified short-path constraint can also be obtained from (45) by requiring it to be satisfied for $l = 0$, thereby shrinking R_i^E by extending the leftmost boundary edge down to the T_1 axis and removing the other edges.

When (48) is combined with the long-path constraints (36) and (38), and the pulse-width constraints (2) and (3), we obtain the following expression for the minimum cycle time:

$$T_{c,\min} = \max\left\{\frac{\overline{\Delta}}{1+\nu}, \max_l \frac{\max_i \left[\left(\sum_{j=i-l}^{i} \Delta_j\right) + S_i\right] - \min_i (\delta_i - H_i)}{1 + l + l\nu}, \max_l \frac{\max_i \left[\left(\sum_{j=i-l}^{i} \Delta_j\right) + S_i\right] + w_1}{2 + l + \nu + l\nu}, 2w_1\right\}. \tag{49}$$

The feasibility of this minimum cycle time must be checked by substituting it, along with the corresponding phase width T_1, in (48). If (48) is violated, then the restricted single-phase constraints have no feasible solution. This check is necessary only if the minimum cycle time obtained in (49) is set by the third or fourth arguments of the max function; if it is determined by the first or second argument, (48) is automatically satisfied.

4.4. Observations

The solution space becomes nonconvex when the early arriving signals are allowed to flow through the latches unimpeded by the clock, i.e. when $d_i > T_c - T_i$ for one or more stages. If necessary or desired, this can be prevented by using $d_i = T_c - T_i$ instead of the actual synchronization equation $d_i = \max(a_i, T_c - T_i)$ and can be accomplished in two ways:

1) By using a restricted single-phase clock which *assumes* that the early signals always start flowing through latches on the enabling clock edge even though they may not actually start their propagation until after the clock edge. This leads to safe though not generally minimum cycle times.
2) By using a coincident multiphase clock which permits the individual phase widths to be adjusted so that $d_i = D_i = T_c - T_i$ actually occurs at every latch. This choice involves more costly clock generation and distribution, but achieves the minimum possible cycle time of any coincident clocking scheme.

V. Examples and Results

We developed a computer program *pipe*T_c which determines the optimal cycle time for n stage pipes. *pipe*T_c reads in the pipeline parameters (number of stages, stage delays, setup and hold times, and wave pipelining parameters) and produces the optimal clock schedules and signal waveforms for general single-phase, restricted single-phase, and coincident n-phase clocking using latches.

In this section we illustrate the use of *pipe*T_c on three pipeline examples to highlight some of the issues in pipeline synchronization. The results are shown in Figs. 6–12. In each figure we show:

- The pipeline parameters (minimum and maximum delays, hold and setup times, wave pipelining parameter, and minimum pulse width)
- The region of feasible solutions, in the T_c/T_1 space, for general single-phase clocking (part (a) in each figure).
- The optimal clock waveform(s), and corresponding signal waveforms at all synchronizer inputs, for:
  general single-phase clocking (part (b)),
  restricted single-phase clocking (part (c)),
  single-phase clocking with negative edge-triggered flip-flops (part (d)),
  coincident n-phase clocking (part (e)).

The clock and signal waveforms in these figures are depicted using the notation introduced in Fig. 2.

The flip-flop solutions are obtained by substituting the synchronization equations $d_i = D_i = T_c$ in the signal propagation equations and combining the results with the hold and setup constraints. This procedure is analogous to those used in Section IV for latch synchronization; however, it is much less involved due to the simplicity of the flip-flop synchronization equations, and leads to the following simple expression for minimum cycle time

$$T_{c,\min} = \max\left\{\max_i\left(\frac{\Delta_i + S_i}{1+\nu}\right),\ 2w_1\right\} \tag{50}$$

subject to the following feasibility conditions:

$$\nu = 0\text{: } \delta_i \geq H_i, \qquad \text{for } i = 0, \cdots, n-1$$

$$\nu \geq 1\text{: } \min_i\left(\frac{\delta_i - H_i}{\nu}\right) \geq \max_i\left(\frac{\Delta_i + S_i}{1+\nu}\right).$$

The phase widths T_{p_i} can be chosen arbitrarily as long as they satisfy the minimum pulse width constraints. By comparing (49) with (50) we see that operation at the ideal latch cycle time of

$$\frac{\overline{\Delta}}{1+\nu} \tag{51}$$

is never possible with flip-flops.

Example 1: The first example is a 4-stage pipeline with an uneven distribution of stage delays. Optimal clock schedules were computed for two cases: (a) $H_1 = 2.0$, and (b) $H_1 = 2.5$. In both cases, $\nu = 0$ and $w = 1$. The results are shown in Figs. 6 and 7 and are summarized in Table I. Examination of these results leads to the following observations:

1) The general single-phase feasible region in Fig. 6 is nonconvex. It consists of the shaded area in the T_c/T_1 plane as well as the line segment AB.
2) The optimal general single-phase cycle time is the same as the optimal coincident 4-phase cycle time reaching the ideal value ($\overline{\Delta}/(1+\nu) = 10$), and is substantially lower than the restricted single-phase optimum (16.0) and the flip-flop optimum (18.0).
3) Although there are always exactly four signal waves in the pipeline in the general single-phase and coincident 4-phase solutions, during each clock cycle there are two waves traveling in stage 0 (from $t = 2.0$ to $t = 8.0$) and in stage 2 (from $t = 2.0$ to $t = 4.0$). This limited form of wave pipelining in particular stages occurs even though $\nu = 0$ since the delays of both stages 0 and 2 are greater than the cycle time. Examination of the signal waveforms suggests another way to determine if a given stage is wave pipelining: *stage i will "contain" 2 or more waves of data in every clock cycle from $t = D_{i-1}$ to $t = A_i$ if $D_{i-1} < A_i$; otherwise at most one wave can be traveling in stage i.*
4) When H_1 is increased from 2.0 to 2.5, the general single-phase feasible region shrinks and becomes convex (Fig. 7). Now, the general single-phase and the restricted single-phase solutions are identical ($T_{c,\min} = 16.5$), and both are larger than the coincident 4-phase solution ($T_{c,\min} = 10.125$). Note that due to the nonconvexity of the general single-phase feasible region, a 0.5 ns change in the hold time of stage 1 causes a 6.5 ns change in the optimal cycle time. In contrast, the restricted single-phase and coincident 4-phase optima changed by 0.5 ns and 0.125 ns, respectively, in response to the same 0.5 ns change in H_1. The flip-flop solution did not change.

Example 2: The second example is a modification of the first in which the delay of stage 1 has been increased from 4.0 to 8.0. We study the effect of changing the hold time of stage 2 from 6.0 to 7.5 in 0.5 ns increments. The results are shown in Figs. 8, 9, 10, and 11, and are sum-

i	δ_i	Δ_i	H_i	S_i
0	16.0	16.0	2.0	2.0
1	4.0	4.0	2.0	2.0
2	12.0	12.0	2.0	2.0
3	8.0	8.0	2.0	2.0

$\nu = 0$, $w = 1$

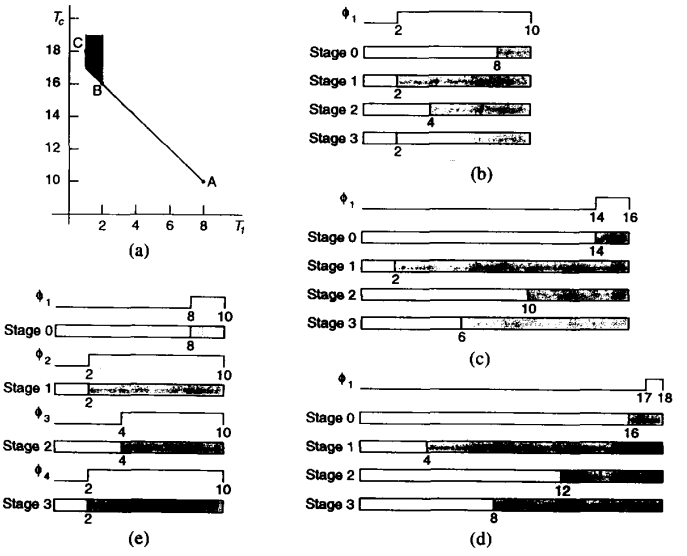


Fig. 6. Example 1, Case (a)—$H_1 = 2.0$ (a) General single-phase feasible region (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 4-phase solution.

i	δ_i	Δ_i	H_i	S_i
0	16.0	16.0	2.0	2.0
1	4.0	4.0	2.5	2.0
2	12.0	12.0	2.0	2.0
3	8.0	8.0	2.0	2.0

$\nu = 0$, $w = 1$

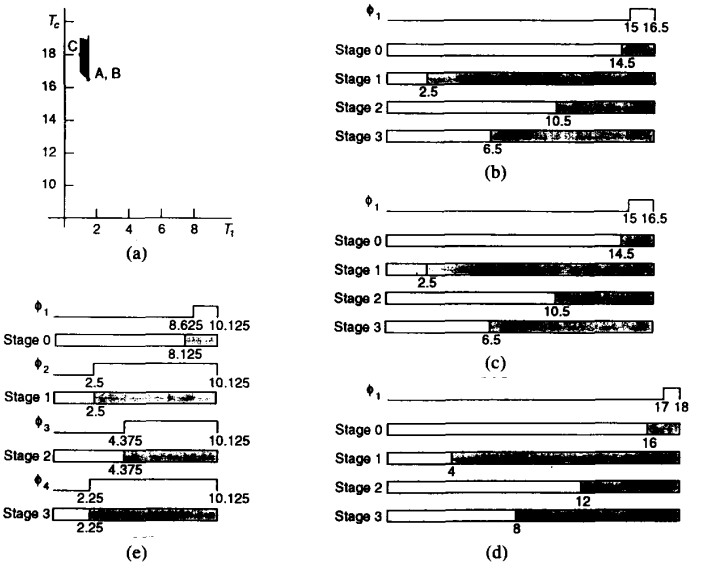


Fig. 7. Example 1, Case (b)—$H_1 = 2.5$. (a) General single-phase feasible region (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 4-phase solution.

TABLE I
SUMMARY OF RESULTS FOR EXAMPLE 1 (ALL TIMES IN UNITS OF NANOSECONDS)

		G 1-Ph[1]		R 1-Ph[2]		C 4-Ph[3]	FF[4]
Case	H_1	T_c	T_1	T_c	T_1	T_c	T_c
(a)	2.0	10.000	8.000	16.000	2.000	10.000	18.000
(b)	2.5	16.500	1.500	16.500	1.500	10.125	18.000

[1]Optimal general single-phase solution
[2]Optimal restricted single-phase solution
[3]Optimal coincident 4-phase solution
[4]Optimal flip-flop solution

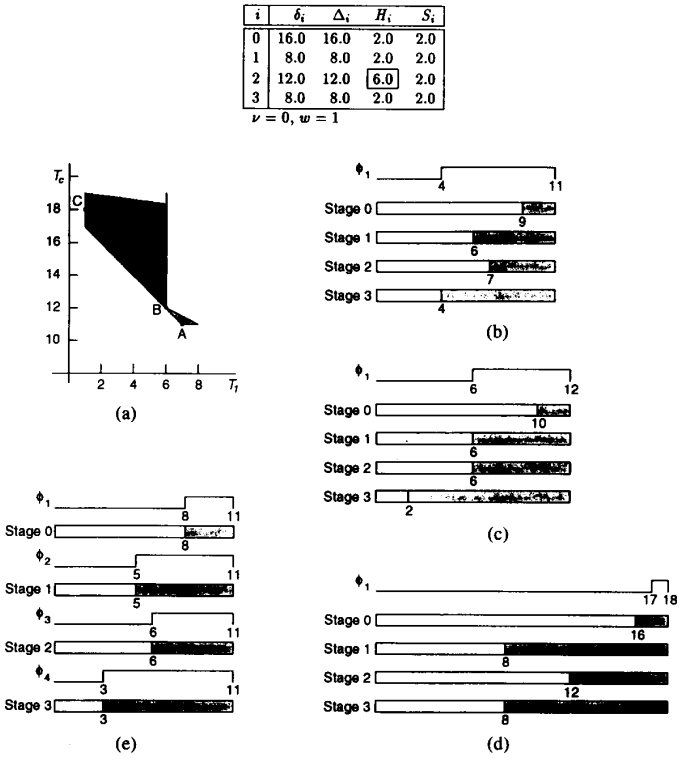


Fig. 8. Example 2, Case (a)—$H_2 = 6.0$. (a) General single-phase feasible region (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 4-phase solution.

marized in Table II. The following additional observations can be made:

1) The general single-phase feasible region is nonconvex (Fig. 8), and becomes disconnected when H_2 is increased to 6.5 and 7.0 (Fig. 9 and Fig. 10). In particular, one of the disconnected subregions shrinks to a point. When H_2 is increased further to 7.5, the feasible region becomes convex (Fig. 11). Further increases in H_2 reduce the size of the feasible region, until it vanishes completely and the problem becomes infeasible.
2) The general single-phase solution is not unique. In fact, for $H_1 = 6.0$ and $H_1 = 6.5$, the same optimal cycle time (11.0) can be achieved with a range of values for T_1. This situation will arise whenever the lower bound constraint on T_c given by (36) is active (this lower bound corresponds to the horizontal line segment).
3) The restricted single-phase solution is now exhibiting wave pipelining in some stages. In fact, only the flip-flip solution is free from wave pipelining (recall that $\nu = 0$ in these experiments).

i	δ_i	Δ_i	H_i	S_i
0	16.0	16.0	2.0	2.0
1	8.0	8.0	2.0	2.0
2	12.0	12.0	6.5	2.0
3	8.0	8.0	2.0	2.0

$\nu = 0, w = 1$

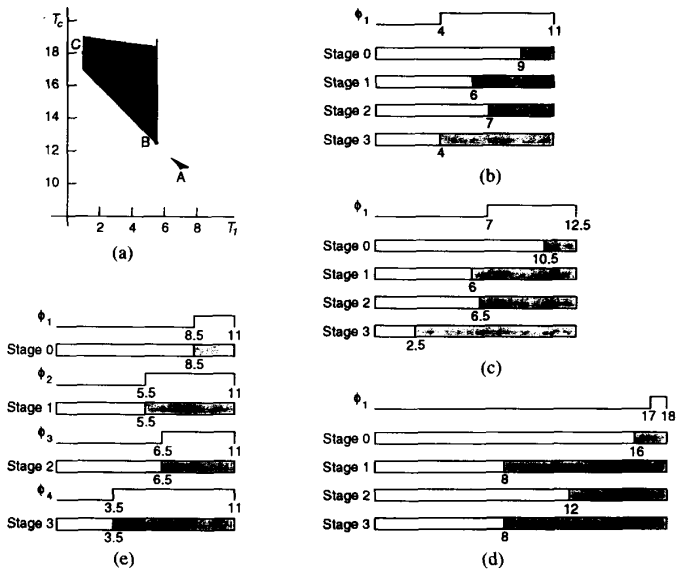


Fig. 9. Example 2, Case (b)—$H_2 = 6.5$. (a) General single-phase feasible region (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 4-phase solution.

i	δ_i	Δ_i	H_i	S_i
0	16.0	16.0	2.0	2.0
1	8.0	8.0	2.0	2.0
2	12.0	12.0	7.0	2.0
3	8.0	8.0	2.0	2.0

$\nu = 0, w = 1$

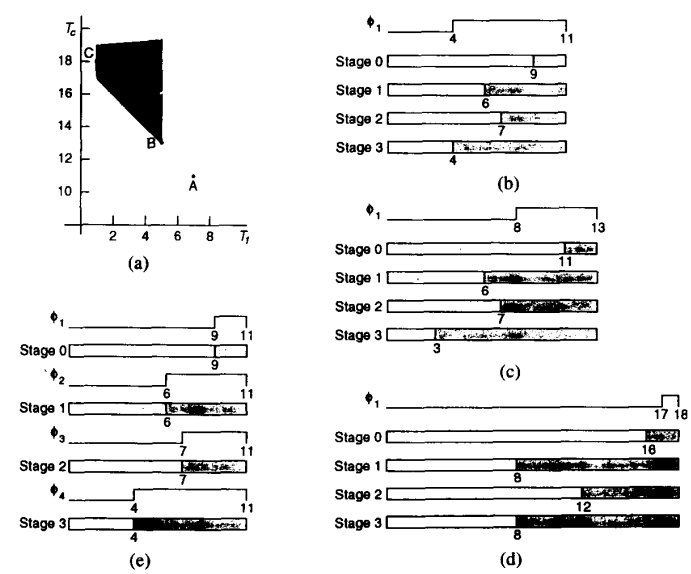


Fig. 10. Example 2, Case (c)—$H_2 = 7.0$. (a) General single-phase feasible (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 4-phase solution.

i	δ_i	Δ_i	H_i	S_i
0	16.0	16.0	2.0	2.0
1	8.0	8.0	2.0	2.0
2	12.0	12.0	7.5	2.0
3	8.0	8.0	2.0	2.0

$\nu = 0, w = 1$

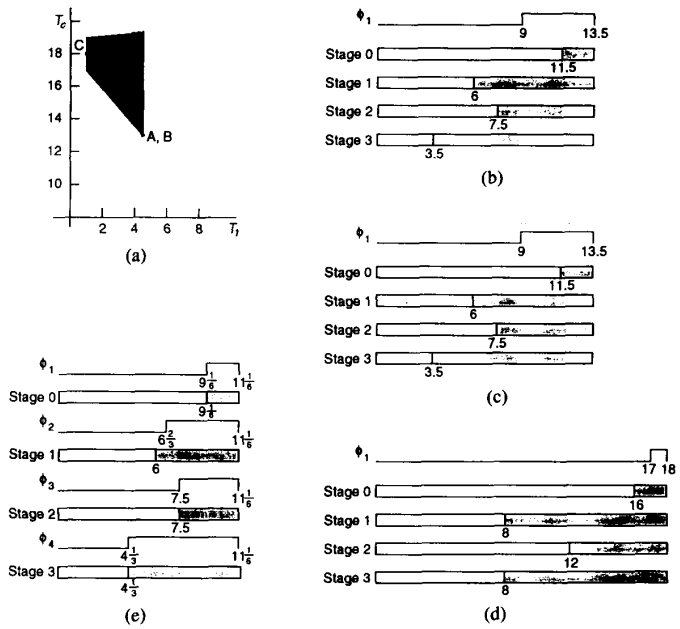


Fig. 11. Example 2, Case (d)—$H_2 = 7.5$. (a) General single-phase feasible region (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 4-phase solution.

TABLE II
SUMMARY OF RESULTS FOR EXAMPLE 2 (ALL TIMES IN UNITS OF NANOSECONDS)

		G 1-Ph[1]		R 1-Ph[2]		C 4-Ph[3]	FF[4]
Case	H_2	T_c	T_1	T_c	T_1	T_c	T_c
(a)	6.0	11.000	[7.0, 8.0]	12.000	6.000	11.000	18.000
(b)	6.5	11.000	[7.0, 7.5]	12.500	5.500	11.000	18.000
(c)	7.0	11.000	7.000	13.000	5.000	11.000	18.000
(d)	7.5	13.500	4.500	13.500	4.500	11.167	18.000

[1]Optimal general single-phase solution
[2]Optimal restricted single-phase solution
[3]Optimal coincident 4-phase solution
[4]Optimal flip-flop solution

4) The 0.5 ns increase in H_2 from 7.0 to 7.5 (cases c and d) causes a 2.5 ns increase in the general single-phase optimum, a 0.5 ns increase in the restricted single-phase optimum, and a *one-sixth* ns increase in the multiphase optimum; the flip-flop optimum does not change. This is consistent with the earlier observation in example 1. The curious one-sixth ns increase in the multiphase case is readily explained in terms of the dual solution of the linear program [13] used to find the optimal cycle time.

5) Note that, for this as well as for the previous example, the optimal cycle time for general single-phase clocking is equal to either the optimal coincident multiphase cycle time or the optimal restricted single-phase cycle time. This is not true in general, as the last example demonstrates.

Example 3: The third example is for a 3-stage pipeline in which, unlike the first two examples, the minimum delays for some stages are strictly less than the corresponding maximum delays. The results are shown in Fig. 12 and suggest the following additional comments:

1) As was just noted, the optimal general single-phase cycle time (13 ns) is strictly less than the restricted single-phase optimum (14 ns) and strictly greater than the coincident 3-phase optimum (10 ns).
2) The optimal coincident 3-phase cycle time (10 ns) is greater than the average maximum stage delay ($\overline{\Delta} = 9.33$ ns) due to the discrepancy between the minimum and maximum stage delays. Specifically, the signal arriving at stage 0 must wait 2 ns before beginning to propagate to stage 1. Averaged over the three pipe stages, this wait time accounts for the observed difference of 0.67 ns (10 − 9.33).

i	δ_i	Δ_i	H_i	S_i
0	4.0	7.0	2.0	2.0
1	5.0	7.0	2.0	2.0
2	14.0	14.0	8.0	2.0

$\nu = 0, w = 2$

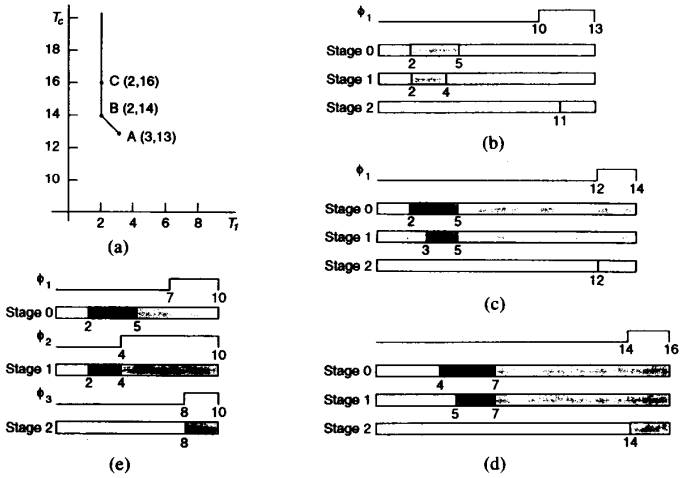


Fig. 12. Example 3. (a) General single-phase feasible region (latches). (b) Optimal general single-phase solution (point A). (c) Optimal restricted single-phase solution (point B). (d) Optimal flip-flop solution (point C). (e) Optimal 3-phase solution.

VI. Conclusions and Future Work

We have studied the problem of minimizing the cycle time for an n-stage pipeline under a variety of clocking conditions. This study clarified the relationships among concurrency, wave pipelining, and clocking. It has also led to a closed-form expression for minimum cycle time under a restricted form of single-phase clocking.

One of the important results of this study was discovering that even for single-phase clocks and simple circuit structures, the region of feasible solutions may be nonconvex or even disjoint. One must conclude, therefore, that such phenomena can also occur in the case of multiphase clocking of more complex circuits. Nonconvexity implies that slight variations in the circuit delays can cause large variations in the cycle time, leading to possible malfunction. Even more serious, a disjoint solution space poses problems of reachability (i.e., how do you start, stop and single-step the clock). In either case, trying to capitalize on the existence of such effects may lead to unreliable circuit operation and "weird" circuit behavior. Both problems can be traced to exploiting the transparency of latches for *fast* as well as slow signals to achieve lower cycle times.

With the above in mind, practical solutions should include only those clocking schemes whose feasible regions are convex, such as coincident multiphase and restricted single-phase clocks. Additional considerations, such as ease of clock generation and distribution, may further limit the range of options to just a few phases. The closed-form minimum cycle time expression for restricted single-phase clocking thus assumes greater significance as a bound on what is practically achievable.

We have incorporated wave pipelining in the model but have not addressed issues such as startability and stoppability (single-stepping). These issues remain as important open problems that must be solved before wave pipelining becomes viable. Finally, the above results do not take clock skew into account. We conjecture that clock skew, clock phase shifts, and wave pipelining can be integrated into a unified model for clock design.

References

[1] K. A. Sakallah, T. N. Mudge, T. M. Burks, and E. S. Davidson, "Synchronization of pipelines," Tech. Rep. CSE-TR-97-91, Univ. Michigan, Ann Arbor, Feb. 1991.

[2] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun, "*check*T_c and *min*T_c: Timing verification and optimal clocking of synchronous digital circuits," in *Dig. Tech. Pap. IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pp. 552-555, Nov. 1990.

[3] T. G. Szymanski, "Computing optimal clock schedules," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, pp. 399-404, 1992.

[4] T. G. Szymanski, "Verifying clock schedules," in *Dig. Tech. Pap. IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 1992, pp. 124-131.

[5] N. P. Jouppi and D. W. Wall, "Available instruction-level parallelism for superscalar and superpipelined machines," in *Proc. Conf. on Architectural Support for Programming Languages and Operating Systems*, pp. 272-282, Boston, MA, Apr. 1989.

[6] S. R. Kunkel and J. E. Smith, "Optimal pipelining in supercomputers," in *Proc. 13th Ann. Symp. on Computer Architecture*, pp. 404-411, 1986.

[7] L. W. Cotten, "Circuit implementation of high-speed pipeline systems," in *AFIPS Fall Joint Computer Conf.*, pp. 489-504, 1965.

[8] T. G. Hallin and M. J. Flynn, "Pipelining of arithmetic functions," *IEEE Trans. Computers*, vol. C-21, pp. 880-886, Aug. 1972.

[9] B. K. Fawcett, *Maximal Clocking Rates for Pipelined Digital Systems*, Master thesis, Univ. Illinois, 1975.

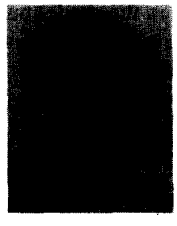
[10] D. Wong, G. D. Micheli, and M. Flynn, "Inserting active delay elements to achieve wave pipelining," in *ICCAD-89 Dig. Technical Pap.* pp. 270-273, 1989.

[11] D. A. Joy and M. J. Ciesielski, "Placement for clock period minimization with multiple wave propagation," in *Proc. of the 28th Design Automation Conf.*, pp. 640-643, 1991.
[12] C.-H. Chang, E. S. Davidson, and K. A. Sakallah, "Using constraint geometry to determine maximum rate pipeline clocking," in *Dig. of Tech. Pap. IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, Nov. 1992, pp. 142-148.
[13] D. T. Phillips, A. Ravindran, and J. J. Solberg, *Operations Research: Principles and Practice*. New York: Wiley, 1976.

Karem A. Sakallah (S'76-M'81-SM'92) received the B.E. degree (with distinction) in electrical engineering from the American University of Beirut, Beirut, Lebanon, in 1975, and the M.S.E.E. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1977 and 1981, respectively.

In 1981 he joined the Department of Electrical Engineering at CMU as a Visiting Assistant Professor. From 1982 to 1988 he was with the Semiconductor Engineering Computer-Aided Design Group at Digital Equipment Corporation in Hudson, Massachusetts, where he headed the Analysis and Simulation Advanced Development team. Since September 1988 he has been at the University of Michigan, Ann Arbor, MI, as Associate Professor of Electrical Engineering and Computer Science. His research interests are primarily in the area of computer-aided design of integrated circuits and systems, with particular emphasis on numerical analysis, multilevel simulation, timing verification and optimal clocking, modeling, knowledge abstraction, and design environments.

Dr. Sakallah is a member of the ACM.



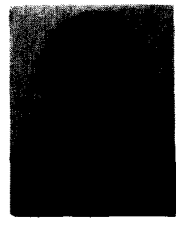
Trevor Mudge (S'74-M'77-SM'84) received the B.S. degree in cybernetics from the University of Reading, England, in 1969, and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana, in 1973 and 1977, respectively.

While at the University of Illinois he participated in the design of several high-performance computers, and did research in computer architecture. Since 1977, he has been on the faculty of the University of Michigan, Ann Arbor, where he has taught classes on logic design, CAD, computer architecture, and programming languages. He is presently a Professor of Electrical Engineering and Computer Science and the Director of the Advanced Computer Architecture Laboratory. He is author of more than 100 papers on computer architecture, programming languages, VLSI design, and computer vision, and he holds a patent in computer-aided design of VLSI circuits. His major research, at present, is the design and construction of high-performance GaAs "microsupercomputer." In addition to his position as a faculty member, he is a consultant for several computer companies in the areas of architecture and CAD.

Dr. Mudge is a member of the ACM and the British Computer Society.

Timothy M. Burks (S'93) received the B.S. degree in computer engineering from the Virginia Polytechnic Institute and State University, Blacksburg, VA, in 1989, and the M.S.E. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1991, where he is currently working toward the Ph.D. degree.

His research interests include timing issues in the design of high-performance digital systems, digital and analog integrated circuits, and applications of object-oriented techniques to the design of VLSI CAD software.



Edward S. Davidson (S'67-M'68-SM'78-F'84) received the B.A. degree in mathematics from Harvard University in 1961, the M.S. degree in communication science from The University of Michigan in 1962, and the Ph.D. degree in electrical engineering from the University of Illinois in 1968.

He worked at Honeywell 1962-1965, Stanford University 1968-1973, and the University of Illinois 1973-1987, managing the hardware design of the Cedar parallel supercomputer at the Center for Supercomputing Research and Development (1984-1987). He then joined The University of Michigan as Professor of EECS and served as its Chair through 1990. His research interests include computer architecture, supercomputing, parallel and pipeline processing, performance modeling, VLSI systems, and computer-aided design. With his graduate students, he developed the reservation table approach to optimum design and cyclic scheduling of pipelines, designed and implemented a shared memory multimicroprocessor system in 1976, and developed a variety of systematic methods for computer performance evaluation. His current research focuses on extending these techniques to evaluate and improve the performance of alternative parallel, vector, and workstation architectures and their compilers on specific application codes. Dr. Davidson was elected Chair of ACM-SIGARCH (1979-1983).