Optimal Clocking of Circular Pipelines*

Karem A. Sakallah Trevor N. Mudge Timothy M. Burks† Edward S. Davidson
Advanced Computer Architecture Lab
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122

Abstract

This paper presents a timing model for circular pipelines and uses it to obtain the minimum cycle time in terms of circuit delays and clock skews. The model accounts for short- and long-path delays, the effects of clock skew, and the use of both latches and flip-flops as synchronizing elements. We describe the formulation and implementation of algorithms to find the minimum cycle time for both single-phase and a restricted class of multi-phase clocks.

1 Introduction

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 non-pipelined case. This maximum is rarely achieved, however, because of dependencies among the operations and overhead due to clocking [1]. Performance can be defined as the sustained number of operations per unit time, and can be expressed as:

$$MOPS = \frac{U(n) \times n}{T_c(n)} \quad (1)$$

where $MOPS$ stands for millions of operations per second, $0 \leq U(n) \leq 1$ is the *utilization* of the pipeline, and $T_c(n)$ is the clock cycle time per 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, in general, but it also 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 operation interdependencies for an appropriate set of benchmark computations. This is a purely "architectural" analysis which disregards all implementation details. 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 which yield the minimum cycle times.

*This work was supported in part by NSF Grant MIP-9014058.
†Mr. Burks is supported by a DoD NDSEG fellowship.

Why Circular Pipelines? 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 and clock skews. This problem has been addressed previously by a number of authors including [1, 2, 3, 4]. This previous work was exclusively concerned, however, with open-ended pipes, and dealt mostly with simple clocking paradigms. In contrast, we consider in this paper circular pipelines, Fig. 1, which more accurately model the flow of computation in real systems by accounting for the timing of the source and sink of the data flowing in the pipeline. For example, one or more stages in such a pipeline can be used to model the "memory" used to supply operands for the computation and to receive results from it.

2 Pipeline Model

Consider the n-stage circular pipeline in Fig. 1. The pipe stages are numbered consecutively from 1 to n, with stage n also referred to as stage 0 to indicate its predecessor relationship to stage 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 synchronizing elements (level-sensitive latches or edge-triggered flip-flops) followed by combinational circuitry. Data flow through the pipeline is regulated by a k-phase clock, where $1 \leq k \leq n$. Stage i is characterized by the parameters defined in Table 1.

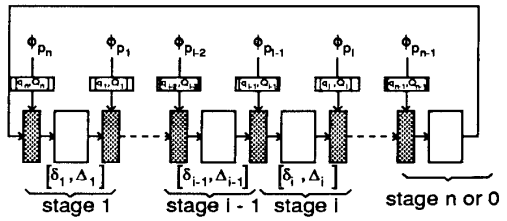

Figure 1: n-stage circular pipeline. Shaded boxes represent the synchronizers

CH3040-3/91/0000/0642$01.00 © 1991 IEEE

p_i	clock phase used to control synchronizer at the output of stage i (synchronizer i).
q_i Q_i	minimum and maximum clock skew to synchronizer i. $(0 \leq q_i \leq Q_i)$
S_i	non-negative setup time of synchronizer i relative to latching edge of phase p_i.
H_i	non-negative hold time of synchronizer i relative to latching edge of phase p_i.
δ_i Δ_i	minimum and maximum propagation delays from the input of synchronizer $i-1$ to the input of synchronizer i. Note that this definition of stage delay lumps together the synchronizer delay of stage $i-1$ and the combinational logic delay of stage i. $(0 \leq \delta_i \leq \Delta_i)$

Table 1: Pipeline Parameters

2.1 Clocking Model

Save for the inclusion of clock skew, the clock model we use here is essentially the same as the general model of clocking introduced in [5]. In particular, we assume a *temporal* rather than a *logical* framework based on the concept of *periodic phases* which define *local time zones* related by *phase shift operators*. A k-phase clock in this model is characterized by $2k$ parameters: $T_1, \cdots, T_k$ denoting the duration of the active interval of each nominal phase, and $e_1, \cdots, e_k$ denoting the time, relative to an arbitrary global time reference, at which each nominal phase *ends* (i.e. when its latching edge occurs). This global reference could be chosen so that $e_k = T_c$, the common cycle time. The phase relations among the k nominal phases are captured by the phase-shift operator E_{pr} defined by:

$$E_{pr} \equiv \begin{cases} (e_r - e_p)\,, & e_p < e_r \\ T_c\,, & e_p = e_r \\ (T_c + e_r - e_p)\,, & e_p > e_r \end{cases} \qquad (2)$$

and which takes on positive values in the range $(0, T_c]$. This phase-shift operator E_{pr} guarantees that signals which begin propagating on the falling edge of phase p are latched on the next falling edge of phase r. If E_{pr} were allowed to be greater than T_c, a form of *wave pipelining* [6] would arise in which more than one signal could propagate through a combinational block simultaneously. Although other phase shift operators could be chosen, in this paper we limit E_{pr} to avoid wave pipelining effects.

Each nominal phase is distributed to a set of synchronizers. Delay through the clock distribution network causes the clock signal received at synchronizer i to be *skewed* from the corresponding nominal phase ϕ_{p_i}. We account for clock skew with the two delay parameters q_i and Q_i defined earlier. The relationship between a nominal phase and its skewed version is shown in Fig. 2, where the shaded intervals model the uncertainty in the time of occurrence of clock edges.

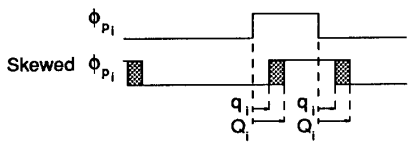

Figure 2: Nominal and skewed clock phase i.

2.2 Timing Constraints

We summarize in Table 2 the constraints and equations which characterize the operation of our pipelines. They are obtained as a special case of the general timing model of synchronous circuits described in [5], slightly modified to account for clock skew effects. In these constraints and equations, w is a specified minimum pulse-width parameter, a_i and A_i refer to the early and late arrival times of a signal at stage i, and d_i and D_i correspond to the corresponding departure times; all times are specified in the local time zone of ϕ_{p_i}, and take values in the interval $(0, T_c]$.

Clock Constraints		
Minimum pulse width:		
T_{p_i}	$\geq$	$w + Q_i - q_i$
$T_c - T_{p_i}$	$\geq$	$w + Q_i - q_i$
Regularity (optional):		
$T_1 = T_2 = \cdots = T_k$		
Latching Constraints		
a_i	$\geq$	$H_i + Q_i$
A_i	$\leq$	$T_c - S_i + q_i$
Synchronization Equations		
Level-sensitive latches:		
d_i	$=$	$\max(a_i, T_c - T_{p_i} + q_i)$
D_i	$=$	$\max(A_i, T_c - T_{p_i} + Q_i)$
Edge-triggered flip-flops:		
d_i	$=$	$T_c + q_i$
D_i	$=$	$T_c + Q_i$
Propagation Equations		
a_i	$=$	$d_{i-1} + \delta_i - E_{p_{i-1}p_i}$
A_i	$=$	$D_{i-1} + \Delta_i - E_{p_{i-1}p_i}$

Table 2: Clocking Constraints

Note that in the propagation equations the phase-shift operators are used to change the frame-of-reference from the local time zone of phase p_{i-1} to that of phase p_i. One way to view these phase shifts is as *negative delays* which effectively reduce the propagation delays between stages.

3 Optimal Cycle Time Calculation

Consider first the case of zero clock skew, i.e. $q_i = Q_i = 0$. If waveform pipelining is disallowed, it is easy to show that the minimum cycle time occurs when the number of clock cycles "covering" the n pipe stages is equal to n. That is, when the phase shift across each pipe stage is exactly equal to T_c. Clocking schemes having this *maximum-phase-shift* property

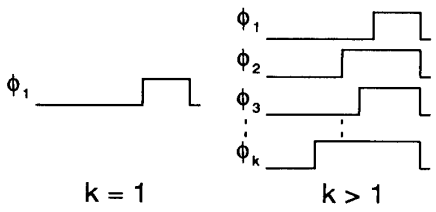

Figure 3: Clocks with maximum possible phase shift between phases

are shown in Fig. 3. They include single-phase clocks and the restricted form of multi-phase clocking shown which will be referred to as *coincident* multi-phase clocking since the latching edges of all k phases *coincide* in time.

We summarize below the optimal solutions obtained for three pipeline synchronization schemes:

1. Negative edge-triggered flip-flops.
2. Level-sensitive latches and a single-phase clock.
3. Level-sensitive latches and a coincident n-phase clock.

Detailed derivations can be found in [7].

3.1 Flip-Flops

A solution exists if $\delta_i \geq H_i$ for all stages; otherwise hold requirements will be violated at one or more stages and the problem would be infeasible. The minimum cycle time is:

$$T_{c,\min} = \max_i(\Delta_i + S_i) \tag{3}$$

and the phase widths must satisfy

$$w \leq T_{p_i} \leq T_{c,\min} - w \tag{4}$$

Note that $T_{c,\min}$ in (3) is independent of the phase widths because of edge-triggering. Hence this solution is equally applicable to single- as well as to coincident multi-phase clocking. Since multi-phase clocking offers no advantage in this case, a single-phase clock that satisfies (4) can be used.

3.2 Latches—Single-Phase Clock

Denoting the single phase ϕ_1, in [7] we show that the setup constraints require the cycle time T_c and phase width T_1 to satisfy the following set of constraints at each latch i:

$$(1+l)T_c + T_1 \geq \sum_{j=i-l}^{i} \Delta_j + S_i, \quad l = 0, 1, \ldots, n-1$$

Ensuring that the hold times are satisfied requires that *at least one* of the following constraints be satisfied for each latch i:

$$lT_c + T_1 \leq \sum_{j=i-l}^{i} \delta_j - H_i, \quad l = 0, 1, \ldots, n-1$$

In [7], we show that the solution space defined by these inequalities is non-convex and present a graphical method for finding the optimal clock schedule.

3.3 Latches—Coincident Multi-phase Clocks

For this case, the individual control we have over the clock phase widths allows us to write a set of linear constraints which define a convex region. The optimal clock schedule can then be determined by solving a linear program. The problem size is linear in the number of stages; in fact the number of inequalities needed to model the n-stage pipeline is equal to $5n$.

We have developed a computer program ($pipeT_c$) which automatically finds solutions under both the single and coincident multi-phase clocking methodologies. Using this program, we have studied a number of different circuits to evaluate the potential cycle-time reduction which can be obtained using coincident multi-phase clocking. Since we can easily incorporate clock skew in our models, we also were able to observe some effects of clock skew on circuit timing designs.

3.4 Taking Clock Skew into Account

The above results remain valid in the presence of non-zero clock skew if the algebraic transformations in Table 3 are made. Substituting these definitions in the pipeline timing solutions and restricting the clocks to be single- or coincident multi-phase (i.e. $E_{pr} = T_c$), we obtain a timing model in the transformed variables and parameters which has the same form as the timing model in the original variables and parameters (Table 2) with the skew parameters set to 0. Thus, the 0-skew results can be used if the above transformations are made first. Note that these transformations effectively move the skews from the clock lines to the data lines, and permit the analysis of the pipeline as though it had no clock skew.[1]

a_i'	$\equiv$	$a_i - q_i$	A_i'	$\equiv$	$A_i - Q_i$
d_i'	$\equiv$	$d_i - q_i$	D_i'	$\equiv$	$D_i - Q_i$
H_i'	$\equiv$	$H_i + Q_i - q_i$	S_i'	$\equiv$	$S_i + Q_i - q_i$
δ_i'	$\equiv$	$\delta_i + q_{i-1} - q_i$	Δ_i'	$\equiv$	$\Delta_i + Q_{i-1} - Q_i$

Table 3: Clock skew transformation equations

4 Examples and Results

In this section we present an example circuit and use the techniques of sections 3.2 and 3.3 to find the optimal clocking scheme under each methodology. We also consider the effects of adding clock skew to our circuit, giving us both a more realistic and less restrictive method for clocking circuits. The circuit we will study is a five-stage pipeline with the parameters specified in Table 4. The average maximum delay in each stage is $\overline{\Delta} = 12$, giving us an absolute lower limit on $T_{c,\min}$.

[1] A transformation similar to that in Table 3 is described in [8, p. 345].

Stage	Δ_i	δ_i	S_i	H_i
1	20	15	2	1
2	10	5	2	1
3	10	5	2	1
4	10	5	2	1
5	10	5	2	1

Table 4: Example circuit parameters

Single-Phase Clocking

For flip-flops, the minimum cycle time is obtained directly from the setup time and the maximum delay between pipeline stages; for this example, $T_{c,\min} = 22$. The single-phase latch solution is found using the constraints described in section 3.2. We begin by finding the range of values of (T_c, T_1) that yield the minimum cycle time while still satisfying the setup constraints. We then search this range until we find the minimum T_c which satisfies the hold time requirements. Using this method, we find the minimum cycle time to be $T_{c,\min} = 18$ with $T_1 = 4$. A plot of this solution is shown in Fig. 4. The bars beneath the clock waveform show the status of signals at the inputs of each synchronizer; signals are stable in the interval $[0, a_i)$, possibly changing in the interval $[a_i, A_i]$, and stable again in $(A_i, T_c]$.

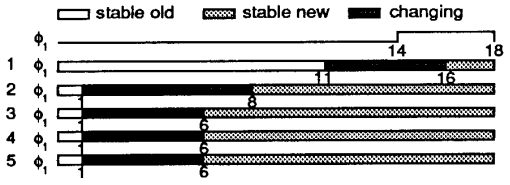

Figure 4: Single-Phase Clocking Results

Coincident Multi-Phase Clocking

Using the *pipe*T_c program, we can find the optimal coincident multi-phase (with latches) solution for our example circuit. Although each of the five synchronizers theoretically receives a separate phase width, in the LP solution only three distinct phases were required. $T_{c,\min}$ is now 14. A plot of the coincident multi-phase timing is shown in Fig. 5.

Single-Phase Clocking with Skew

If clock skews are treated as designable parameters, we may obtain lower cycle times by adjusting them along with the clock parameters [9]. To observe this effect, we will apply a clock skew to synchronizer 1 of $Q_1 = q_1 = 5$. Using the skew transformation equations in Table 3, we replace each variable in our model with its primed equivalent, and observe that the only values that change are Δ_1 and δ_1, which are reduced by 5, and Δ_2 and δ_2, which are increased by 5. Thus our circuit description is the same as before but with $\Delta_1 = \Delta_2 = 15$ and $\delta_1 = \delta_2 = 10$. Applying

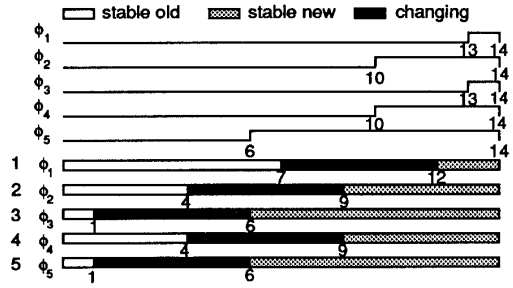

Figure 5: Multi-Phase Clocking Results

the single-phase latch solution procedure gives $T_{c,\min} = 14$, which (coincidentally) gives the same performance as the coincident multi-phase approach. Fig. 6 is a plot of the circuit timing for the skewed single-phase approach. Note Φ_1 is the original clock and Φ_2 the skewed clock; the arrival times for synchronizer 1 are now referred to the skewed clock signal Φ_2.

Adding clock skew has allowed us to achieve a lower cycle time than the single phase design; however, we have also introduced an implicit form of waveform pipelining in our circuit's operation. By adding skew, we have changed the effective phase shifts E_{pr} to $E'_{pr} = E_{pr} + Q_r - Q_p$. For cases when $Q_r > Q_p$ (inevitable for circular pipes unless all skews are equal), $E'_{pr} > T_c$. Since new waves of data are initiated every cycle (T_c), we can have more than one wavefront propagating simultaneously along each path where $E'_{pr} > T_c$. Therefore, by adding clock skew, we can reduce cycle time, but we must also recognize that waveform pipelining is introduced.

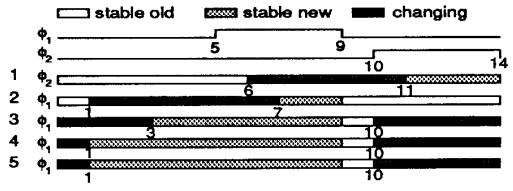

Figure 6: Skewed Single-Phase Clocking Results

Multi-Phase Clocking with Skew

To complete our comparisons, we used the *pipe*T_c program to find $T_{c,\min}$ for our circuit using coincident multi-phase clocking in the presence of skew. For this case, we have complete control over both the active interval and phase shift of each synchronizer's individual clock signal. To observe the effects of clock skew, we used the same skew as in the previous section, $Q_1 = q_1 = 5$, with all other skews set to zero. As might be expected, we found a cycle time lower than that of all of the other approaches, with a value of $T_{c,\min} = 12.\overline{6}$. The signal waveforms for this approach are plotted in Fig. 7.

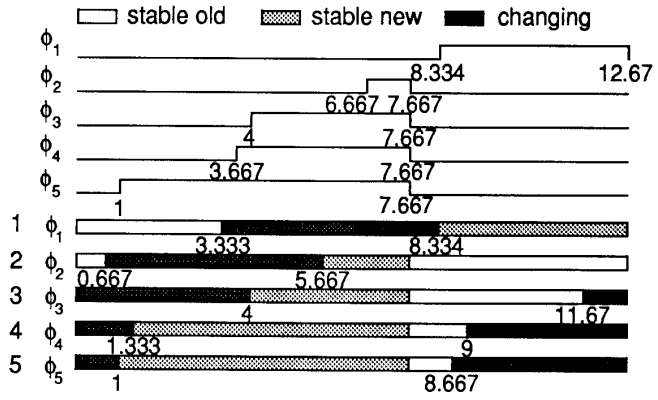

Figure 7: Skewed Multi-Phase Clocking Results

Comments

These examples have demonstrated the effects of varying two important clocking parameters: phase width and skew. We have seen that each level of control allows us to further reduce the cycle time towards the ultimate limit of $T_{c,\min} = \overline{\Delta}$. We have obtained optimal solutions for each clocking approach for a fixed clock skew. As an extension to these methods, we can easily modify the coincident multi-phase linear programs to also find the optimal clock skew to apply to a circuit. This is similar to Fishburn's clock skew optimization methods for edge-triggered flip-flops [9]. However, as we have seen, these approaches implicitly introduce wave pipelining effects.

5 Conclusions

In general, minimizing the clock cycle time for a pipeline requires the consideration of clock generation and distribution, as well as logic and circuit design of the pipe stages. In this paper, we have assumed that the design of the pipe stage circuitry is fixed (i.e. all propagation delays are specified). If the minimum cycle time corresponding to this "fixed" design is unacceptably high, a redesign (resynthesis) would be necessary to reduce some or all of the delays. The results presented in this paper can be used to guide this resynthesis step by identifying the most critical delays in the pipeline. We are currently investigating the integration of these results with a logic synthesis system.

References

[1] S. R. Kunkel and J. E. Smith, "Optimal Pipelining in Supercomputers," in *Proc. of the 13th Annual Symposium on Computer Architecture*, pp. 404–411, 1986.

[2] L. W. Cotten, "Circuit Implementation of High-Speed Pipeline Systems," in *AFIPS Fall Joint Computer Conference*, pp. 489–504, 1965.

[3] T. G. Hallin and M. J. Flynn, "Pipelining of Arithmetic Functions," *IEEE Trans. Computers*, vol. C-21, no. 8, pp. 880–886, August 1972.

[4] B. K. Fawcett, *Maximal Clocking Rates for Pipelined Digital Systems*, Master's thesis, University of Illinois, 1975.

[5] 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 *ICCAD-90 Digest of Technical Papers*, Santa Clara, California, November 1990, IEEE.

[6] D. Wong, G. D. Micheli, and M. Flynn, "Inserting Active Delay Elements to Achieve Wave Pipelining," in *ICCAD-89 Digest of Technical Papers*, pp. 270–273, 1989.

[7] K. A. Sakallah, T. N. Mudge, T. M. Burks, and E. S. Davidson, "Optimal Clocking of Circular Pipelines," Technical Report CSE-TR-97-91, University of Michigan, Dept of EECS, Ann Arbor, MI 48109-2122, June 1991.

[8] L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Addison Wesley, 1985.

[9] J. P. Fishburn, "Clock Skew Optimization," *IEEE Trans. Computers*, vol. 39, no. 7, pp. 945–951, July 1990.