Edge-Triggering vs. Two-Phase Level-Clocking

(Extended Abstract)

Marios C. Papaefthymiou
Keith H. Randall
MIT Laboratory for Computer Science
Cambridge, Massachusetts 02139

Abstract

Level-clocked designs that employ a two-phase, nonoverlapping clocking scheme have the theoretical potential to operate up to twice as fast as edge-triggered designs. We have run experiments that demonstrate, however, that edge-triggering is often just as fast as two-phase clocking, and that the speed potential of two-phase clocking is generally not obtained except when the delay between any two consecutive latches is roughly uniform and close to the maximum gate delay. Moreover, our experiments show that asymmetrical clocking of a two-phase circuit often does not provide any speedup over optimal symmetric clocking schemes.

Level-clocking can lead to substantial reductions in the number of storage elements in a circuit, however. Our experiments show that for edge-triggered circuitry that has been retimed to operate with the minimum possible clock period, by replacing each flip-flop by a pair of level-sensitive latches and subsequently retiming the resulting two-phase circuit, the number of storage elements can be reduced by up to 38% without increasing the clock period of the final design or affecting its I/O specification. Reductions of greater than 25% were achieved for more than one third of the circuits we tested.

Our experiments were performed using TIM, a timing optimization package currently under development at MIT. We ran our tests on MCNC benchmark circuits, AT&T communication circuits, and custom circuitry designed for MIT's Alewife machine.

1 Introduction

Level-clocking is becoming an increasingly popular alternative to edge-triggering as a clocking methodology for high-performance designs. In level-clocked circuitry, clocked storage elements are implemented using level-sensitive latches, which become transparent and transmit their inputs unimpeded to their outputs while their clock signals are asserted. Proponents of level-clocking argue that level-clocked circuitry can provide more flexibility in meeting a specific clock period and that it has the theoretical potential to operate faster than the more conventional edge-triggered circuitry. These arguments are based on the fact that in level-clocked circuitry computations are allowed to extend beyond a single clock period during the transparent phase of the level-sensitive latches, in constrast to edge-triggered circuitry, in which computations along every path must complete within a clock period. Advocates of edge-triggering, on the other hand, present simplicity and implementation ease as major advantages of edge-triggering, since edge-triggered flip-flops directly support the abstraction of a storage element that is synchronized by the ticking of a clock.

They also refer to the existence of powerful design tools as another major incentive for designing circuitry with edge-triggered flip-flops.

These arguments in support of edge-triggering and level-clocking are either theoretical or nonquantifiable. We wanted to make an empirical study of the two clocking methodologies. For this purpose, we have run experiments that compare edge-triggered implementations of synchronous circuitry and corresponding level-clocked implementations that employ a two-phase, nonoverlapping clocking scheme. Our empirical comparison focused on two specific quantitative measures: speed and number of storage elements. We ran our tests using TIM, a timing optimization tool currently under development at MIT [17].

Our speedup experiments show that edge-triggered circuitry often operates just as fast as two-phase circuitry, despite the theoretical advantage of two-phase clocking, and that the speed potential of two-phase clocking is generally obtained only when the combinational delay between any two consecutive latches is roughly uniform and close to the maximum gate delay. Our experiments also show that two-phase clocking leads to greater speedups when all gate delays in the circuit are roughly equal. Our experimentation with clock tuning suggests that asymmetric clocking schemes provide little or no speedup over optimal clocking schemes.

With respect to the number of storage elements, however, our experiments demonstrate that two-phase clocking can lead to substantial reductions in aggressive edge-triggered designs that operate at maximum speed. For two of our test circuits, we obtained reductions of 38%; for more than one third of our circuits, we obtained reductions of 25%; and for more than half of our circuits, reductions exceeded 18%. For low-performance designs that operate below their speed potential, however, our experiments show that two-phase clocking does not reduce the number of storage elements.

How can edge-triggering and two-phase clocking be fairly compared on an empirical basis? First, a fair experiment should compare competing circuit implementations that have the same functionality. It would be meaningless to compare two circuits that compute different functions. Second, a fair experiment should compare competing circuit implementations based solely on differences due to their storage elements. It would be unfair to compare two circuits which differ in their combinational elements, for example, because such a comparison would not depend only on the clocking methodology employed.

We did not have the resources to embark on designing pairs of competing circuits for various applications. We settled, therefore, on the strategy of taking edge-triggered circuits and using our timing tool TIM to produce equivalent two-phase circuits. We could have done the reverse, converting two-phase circuits to edge-triggered ones, but since edge-triggered designs are more popular, we were able

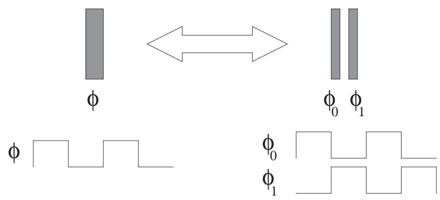


Figure 1: Replacement of an edge-triggered flip-flop by a pair of level-sensitive latches. The flip-flop is clocked by a single clock ϕ, and the two level-sensitive latches are clocked on the phases ϕ_0, ϕ_1 of a two-phase, nonoverlapping clocking scheme.

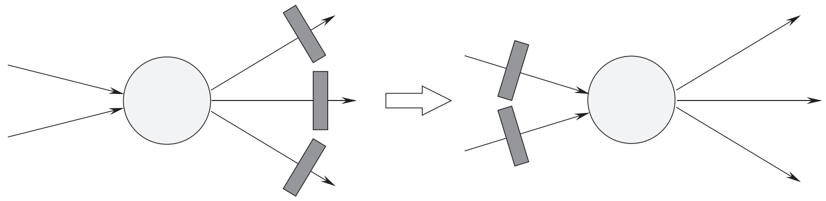


Figure 2: This figure illustrates retiming of a gate with lag 1. In this case, one storage element is removed from each output wire of the gate, and one storage element is inserted at each input wire of the gate. The total number of storage elements is reduced by 1. The critical paths in the circuit may also change as a result of the relocation of the storage elements.

to obtain several interesting edge-triggered circuits. (We hope to also obtain interesting level-clocked designs and remedy this situation before the full paper is published.)

We produced a two-phase circuit from an edge-triggered one by following a two-step procedure. The first step of this procedure was to replace each edge-triggered flip-flop by a pair of back-to-back level-sensitive latches that are clocked by a two-phase, nonoverlapping clocking scheme, as shown in Figure 1. (In fact, it is common in VLSI to implement edge-triggered flip-flops by a pair of back-to-back level-sensitive latches [3, 24].) The two-phase circuit that results after this conversion has the same clock period and the same number of storage elements as the original edge-triggered circuit, assuming that each edge-triggered flip-flop counts as two level-sensitive latches. Moreover, the placement of its latches is dictated by the original edge-triggered design, and the potential of two-phase clocking due to alternate placements of the latches in the circuit is not revealed. Thus, we needed a method to relocate storage elements and explore the space of possible placements in the circuit without changing its functionality or its I/O specification.

The second step of the procedure was to use the "retiming" transformation to relocate the storage elements of the two-phase circuit that resulted from the first step. Retiming relocates storage elements in both edge-triggered and level-clocked circuitry without changing its functionality [8, 9, 10]. In addition, retiming is a "universal" transformation for speeding up circuits, in the sense that any other functionality-preserving transformation that did better than retiming would depend on the functionality of the gates in the circuit [9]. Figure 2 illustrates the retiming operation for a gate in a circuit. Observe that retiming can change the clock period as well as the number of storage elements in a circuit.

Our experimental procedure compared an optimal edge-triggered implementation that we obtained from an original edge-triggered circuit with an optimal two-phase implementation of the corresponding two-phase circuit. The use of an optimal edge-triggered implementation as a reference point was essential to ensure that we did not penalize edge-triggering due to suboptimalities in the original edge-triggered circuit that depended on the placement of the storage elements by the circuit designer and were not intrinsic to edge-triggering.

We performed two kinds of experiments. The first kind of experiments compared edge-triggered and two-phase circuits with respect to speed. Our basic experimental approach was the following. First, we retimed a given edge-triggered implementation for maximum speed. Then, by using retiming in conjunction with tuning of the clocking schemes, we obtained the fastest possible implementation of the corresponding two-phase circuit, and we compared the speed of the two optimal implementations. The second kind of experiments compared edge-triggered and two-phase circuits with respect to their number of storage elements, when operating at some specified clock period. We first retimed a given edge-triggered implementation without changing its I/O specification, in order to achieve the specified clock period with the minimum number of flip-flops. We then retimed the corresponding two-phase circuit without changing its I/O specification, in order to achieve the same clock period with the minimum number of latches. We compared the number of storage elements in the two optimal implementations, under the reasonable assumption that each edge-triggered flip-flop is counted as two level-sensitive latches.

Timing verification and optimization of synchronous circuitry has been the subject of extensive study [1, 2, 4, 5, 6, 7, 8, 10, 11, 12, 18, 19, 20, 22, 23]. The concept of replacing each edge-triggered flip-flop by a pair of back-to-back level-sensitive latches, and then using retiming for speed optimization has been mentioned in [1, 6, 11]. The potential of level-clocking for reducing the number of storage elements has been mentioned in [11]. The idea of using latches instead of flip-flops has been also used in [21]. Retiming for speed has been studied in the context of single-phase level-clocked circuits in [19]. Despite the large amount of work in this area, our contribution is (we believe) the first attempt to quantify empirically the performance differences of edge-triggering and two-phase clocking.

The remainder of the paper has three sections. Section 2 describes our experimental methodology and reports our results on the relative speed of the two implementation approaches. In Section 3 we present our experimental results on latch-minimization. Section 4 concludes with directions for further research. Appendix A presents an upper bound on the relative speedup that can be achieved by two-phase clocking over edge-triggering and explains intuitively how two circuit characteristics, the maximum gate delay $d_{\max}$ and the critical ratio R of the circuit, determine the speedups that can be achieved by two-phase clocking. In Appendix B we show that the problem of retiming level-clocked circuitry in order to minimize the number of latches can be solved in polynomial time for symmetric clocking schemes.

2 Speedup experiments

In this section, we present our investigation of edge-triggering and two-phase clocking with respect to speed. First, we briefly refer to our tools and test circuits. We move on to describe and motivate our experimental methodology, and then we discuss our results.

Our experiments were performed using TIM, a timing optimization package currently under development at MIT [17]. TIM performs a variety of functions on two-phase circuitry, such as timing verification, retiming, clock tuning, and sensitivity analysis. It also performs timing verification, retiming, and sensitivity analysis on edge-triggered circuitry. TIM has been implemented in C, and most of its functions have been integrated in the SIS tools from Berkeley. Our test circuits were MCNC benchmark circuits and AT&T communication circuits, all of which were originally designed with edge-triggered flip-flops. The largest among these circuits had 290 gates, and the current version of TIM performed retiming and simultaneous tuning on that circuit in less than 2 minutes on a 16MB SPARCstation.

Experimental methodology

In our speedup experiments we employed the following three optimizations:

OP1 Retiming of edge-triggered circuitry for maximum speed of operation (minimum clock period).

OP2 Retiming of two-phase circuitry for maximum speed of operation with a symmetric clocking scheme.

OP3 Retiming and simultaneous clock tuning of two-phase circuitry for maximum speed of operation.

Using these three optimizations, we initially performed experiments SP1 and SP2.

SP1 We compared the speed of each original edge-triggered circuit that was optimized using OP1 with the speed of the corresponding two-phase circuit that was optimized using OP2.

SP2 We compared the speed of each original edge-triggered circuit that was optimized using OP1 with the speed of the corresponding two-phase circuit that was optimized using OP3.

The goal of our experimentation was not only to investigate whether two-phase clocking could speed-up the particular edge-triggered circuits in our test suite. We also wanted to determine specific design characteristics that may lead to faster two-phase circuits. To that effect, we performed experiments SP3 and SP4 on altered versions of our original test circuits that were obtained by modifying them in two ways.

The first modification changed the number of storage elements in the original circuits by pipelining, a transformation that increases the latency of a computation without decreasing its throughput. A degree-P pipelining of each circuit was obtained by multiplying the number of storage elements on each wire of the circuit by the integer P. The purpose of this transformation was to investigate which of the two implementation approaches is favored more when the number of storage elements is increased.

The second modification changed the gate delays of the original circuits. For each circuit G, we created four additional circuits G_i for $i = 0.8, 0.6, 0.4, 0.2$. Each G_i was topologically identical to G, but its gate delays d_i were modified. For each circuit G_i, each gate delay $d_i(v)$ was set equal to $d^i(v)$, where $d(v)$ was the delay assigned to v using the library corresponding to the circuit G. Thus, for smaller values of the exponent i, the gate delays in the circuits G_i became increasingly uniform. The objective of this modification was to see how uniformity of gate delays affects the speed of the two implementations.

Using the three optimizations on the modified circuits we performed experiments SP3 and SP4.

SP3 On each circuit G_i for $i = 1.0, 0.8, 0.6, 0.4, 0.2$, we applied the following procedure for $P = 1, 2, \ldots, 6$. We optimized the edge-triggered circuit using OP1, and we compared its speed with its corresponding two-phase circuit that was optimized using OP2.

SP4 On each circuit G_i for $i = 1.0, 0.8, 0.6, 0.4, 0.2$, we applied the following procedure for $P = 1, 2, \ldots, 6$. We optimized the edge-triggered circuit using OP1, and we compared its speed with its corresponding two-phase circuit that was optimized using OP3.

Note that for $i = 1.0$ and $P = 1$, experiments SP3 and SP4 were identical to experiments SP1 and SP2, respectively.

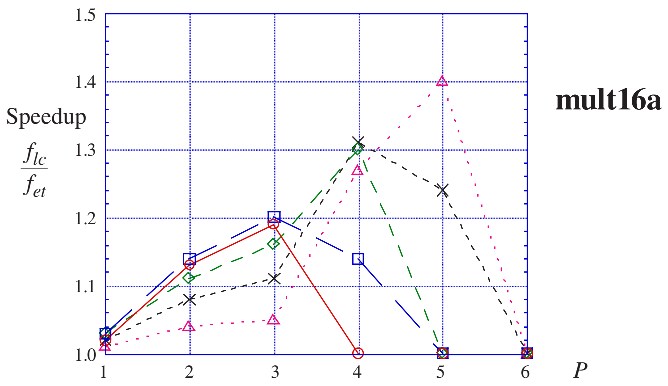
Experimental results

Remarkably, our initial experiments SP1 and SP2 indicated that two-phase clocking was no better than edge-triggering for any of our test circuits. The application of the three optimizations OP1, OP2, and OP3 on the original circuits, with gate delays assigned by their corresponding libraries, showed no speedup by switching to two-phase clocking. Although this result was surprising and unexpected, it could not have been a mere coincidence. Our subsequent empirical investigation with experiments SP3 and SP4 led us to the conclusion that there are two important circuit characteristics that determine the relative speed of the two implementation approaches: the maximum gate delay $d_{\max}$ and the *critical ratio* R, which is defined as the maximum ratio of total delay over total number of storage elements around the cycles in the edge-triggered circuit.

Our experimental results for SP3 are illustrated in the plots of Figure 3. Each plot gives data for an original test circuit $G_{1.0}$ and its four delay-modified versions G_i for $i = 0.8, 0.6, 0.4, 0.2$. For each of the five delay configurations of a test circuit and for degrees of pipelining up to 6, each plot gives two numbers: the speedup f_{lc}/f_{et} achieved by two-phase clocking over edge-triggering, and the ratio $d_{\max}/R$, where $d_{\max}$ is the maximum gate delay and R is the critical ratio of the circuit. For each circuit G_i, the value of the ratio $d_{\max}/R$ closest to 1 is boldfaced.

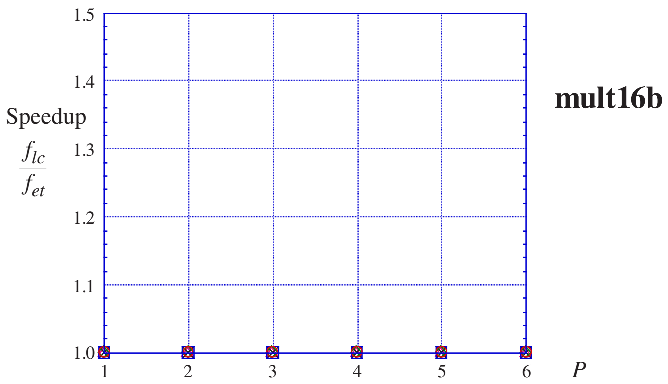
As is apparent from the graphs, for almost every delay configuration, the maximum speedup is achieved when the ratio $d_{\max}/R$ is closest to 1. In the five configurations of `mult16a`, for example, as the ratio $d_{\max}/R$ increases from small values and approaches 1 from below, the speedup constantly increases. When the ratio exceeds 1, the speedup soon drops down to 1. A similar pattern is revealed for almost all of our test circuits. This phenomenon can be justified as follows. The critical ratio R is a lower bound on the clock period of both the edge-triggered and the level-clocked circuit [6, 15]. Consequently, the longest combinational delay in the circuits is at least R under any transformation that does not change the number of storage elements around the cycles in the circuit. Retiming distributes the storage elements, however, so that combinational path delays are roughly equal across the circuit and close to the critical ratio R. When R becomes comparable to the maximum gate delay $d_{\max}$, then the longest combinational delay also tends to approach $d_{\max}$, and then the potential of two-phase clocking becomes apparent. Intuitively, when R approaches $d_{\max}$, level-clocking evens out differences among path delays more effectively than edge-triggering by letting the computations ripple through the transparent latches.

Let us examine more closely some characteristic graphs in Figure 3. Our initially surprising results from experiments SP1 and SP2 can be explained by looking at the ratios $d_{\max}/R$ of the original circuits, which correspond to $P = 1$ and $i = 1.0$. For every such circuit, the ratio $d_{\max}/R$ is smaller than 0.67. The only exceptions are `mult16b`, `ampseq2`, and `ampseq1`. `mult16b` has a ratio greater than 1, and consequently, it is already heavily pipelined. For higher degrees of pipelining, `mult16b` does not become any faster, which leads us to the conclusion that the original design of `mult16b` takes full advantage of any existing speed potential. The situation with `ampseq2` is similar. The original design has already no margin for improvement, and for higher degrees of pipelining there are sufficiently many storage elements for edge-triggering to be as fast as two-phase clocking. The situation with `ampseq1` is



P	1	2	3	4	5	6
i=0.2	0.20	0.39	0.59	0.79	**0.98**	1.17
i=0.4	0.23	0.46	0.68	**0.91**	1.14	1.37
i=0.6	0.26	0.53	0.79	**1.05**	1.31	1.58
i=0.8	0.31	0.60	**0.90**	1.20	1.50	1.80
i=1.0	0.34	0.67	**1.00**	1.34	1.67	2.01

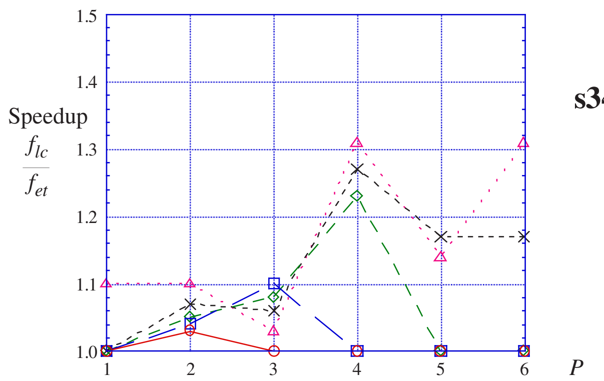
$\frac{d_{max}}{R}$



P	1	2	3	4	5	6
i=0.2	0.59	**1.18**	1.77	2.34	2.95	3.54
i=0.4	**0.68**	1.36	2.04	2.68	3.40	4.02
i=0.6	**0.76**	1.51	2.28	3.05	3.78	4.50
i=0.8	**0.83**	1.65	2.48	3.33	4.16	4.96
i=1.0	**1.02**	2.04	3.06	4.11	5.14	6.13

$\frac{d_{max}}{R}$

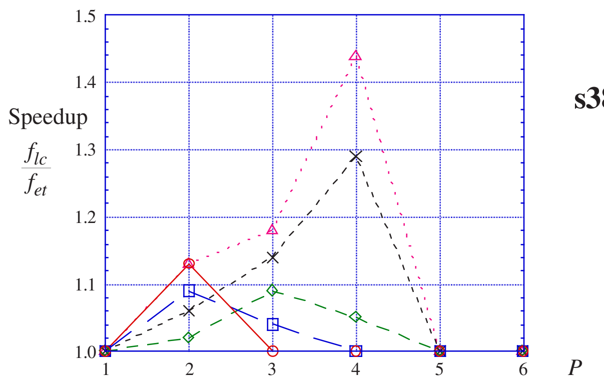
Figure 3: Results of experiment SP3 on the MCNC benchmark and the AT&T circuits. Each plot corresponds to a test circuit. The first row of the horizontal axis gives the pipelining degree P. Each of the next five rows corresponds to a circuit G_i for $i = 0.2, 0.4, 0.6, 0.8, 1.0$, and it gives the ratio $d_{\max}/R$ for each pipelining degree. In each row, the ratio $d_{\max}/R$ closest to 1 is boldfaced. The vertical axis gives the speedup f_{lc}/f_{et} obtained for a specific i and a specific P. The clock frequency f_{et} was obtained by applying OP1. The clock frequency f_{lc} was obtained by applying OP2. For almost every test circuit, maximum speedups are achieved when the ratio $d_{\max}/R$ is closest to 1, or equivalently, when R is closest to $d_{\max}$. Greater peak speedups are achieved as we move from $G_{1.0}$ to $G_{0.2}$, that is, as the gate delays become roughly equal across the entire circuit. The results of experiment SP4 have no significant differences from the results in this figure.



s344

P	1	2	3	4	5	6
i=0.2	0.15	0.29	0.44	0.59	0.73	**0.88**
i=0.4	0.19	0.39	0.58	0.77	**0.96**	1.16
i=0.6	0.25	0.50	0.75	**1.00**	1.25	1.50
i=0.8	0.32	0.65	**0.97**	1.29	1.61	1.93
i=1.0	0.41	**0.83**	1.24	1.65	2.07	2.48

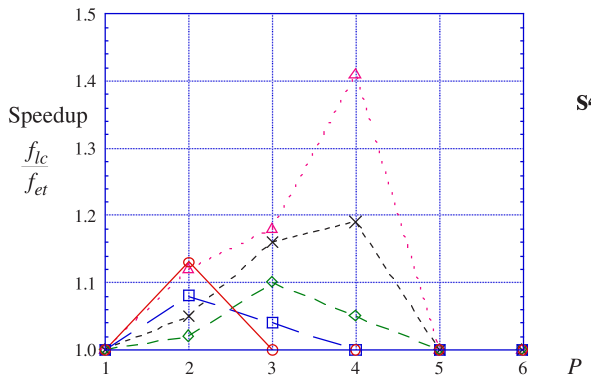
$\frac{d_{max}}{R}$



s382

P	1	2	3	4	5	6
i=0.2	0.23	0.47	0.70	**0.93**	1.16	1.39
i=0.4	0.27	0.54	0.81	**1.08**	1.35	1.62
i=0.6	0.31	0.62	**0.93**	1.24	1.55	1.86
i=0.8	0.35	0.70	**1.05**	1.40	1.75	2.10
i=1.0	0.39	0.78	**1.16**	1.55	1.94	2.33

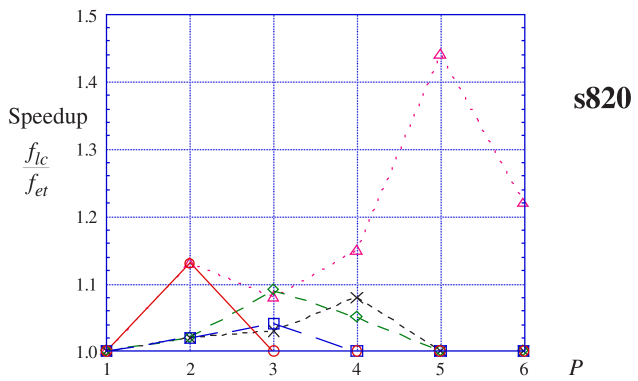
$\frac{d_{max}}{R}$



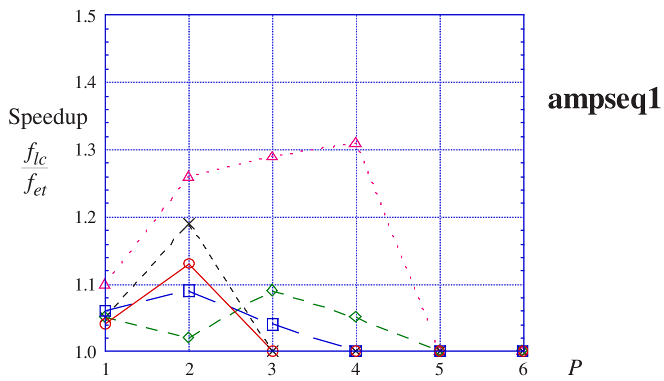
s400

P	1	2	3	4	5	6
i=0.2	0.24	0.47	0.71	**0.95**	1.19	1.43
i=0.4	0.28	0.56	0.83	**1.11**	1.39	1.67
i=0.6	0.32	0.65	**0.97**	1.30	1.62	1.94
i=0.8	0.37	0.74	**1.11**	1.49	1.86	2.22
i=1.0	0.42	**0.84**	1.26	1.68	2.10	2.52

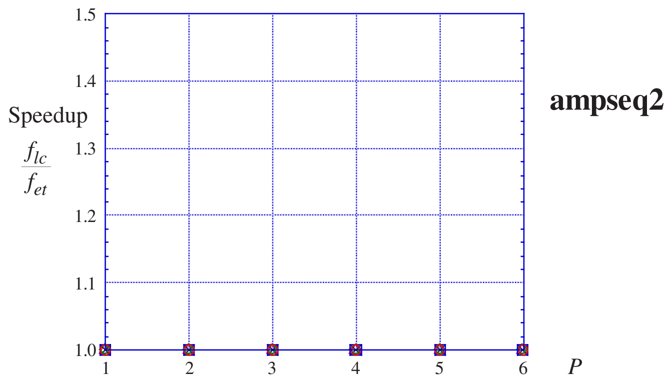
$\frac{d_{max}}{R}$



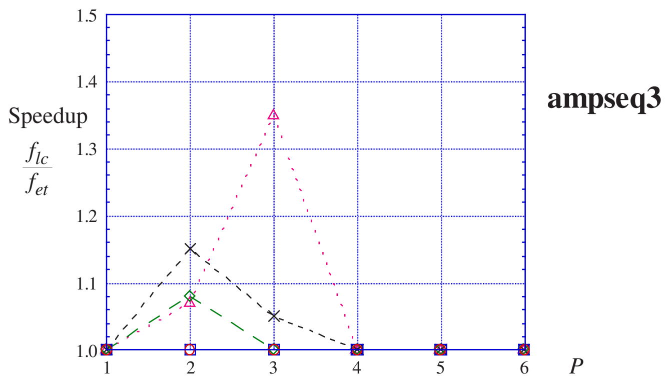
							$\frac{d_{max}}{R}$
i=0.2	0.17	0.33	0.50	0.67	0.84	**1.01**	
i=0.4	0.24	0.48	0.72	**0.97**	1.20	1.44	
i=0.6	0.33	0.66	**1.00**	1.33	1.66	2.00	
i=0.8	0.43	**0.87**	1.30	1.73	2.16	2.59	
i=1.0	0.53	**1.07**	1.61	2.14	2.68	3.21	



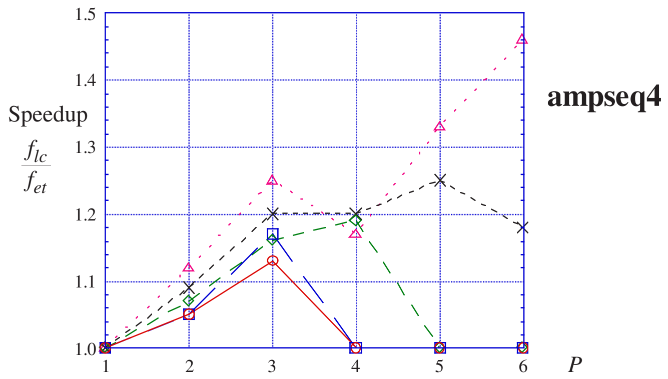
							$\frac{d_{max}}{R}$
i=0.2	0.32	0.65	**0.97**	1.29	1.61	1.93	
i=0.4	0.45	**0.90**	1.35	1.80	2.20	2.65	
i=0.6	0.61	**1.21**	1.82	2.43	3.03	3.64	
i=0.8	**0.80**	1.60	2.40	3.19	4.00	4.80	
i=1.0	**0.97**	1.93	2.90	3.87	4.83	5.80	



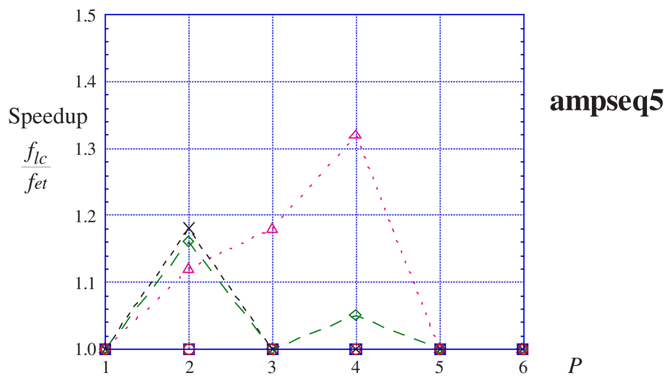
							$\frac{d_{max}}{R}$
i=0.2	0.59	**1.17**	1.76	2.32	2.93	3.51	
i=0.4	0.66	**1.34**	1.99	2.65	3.36	3.97	
i=0.6	**0.74**	1.47	2.20	2.94	3.70	4.40	
i=0.8	**0.80**	1.59	2.39	3.20	3.98	4.73	
i=1.0	**0.85**	1.69	2.54	3.40	4.23	5.07	



	1	2	3	4	5	6	
i=0.2	0.33	0.65	**0.97**	1.30	1.62	1.96	$\frac{d_{max}}{R}$
i=0.4	0.41	**0.82**	1.23	1.64	2.05	2.46	
i=0.6	0.51	**1.02**	1.54	2.04	2.56	3.07	
i=0.8	0.59	**1.19**	1.78	2.38	2.97	3.55	
i=1.0	**0.67**	1.35	2.03	2.69	3.38	4.04	



	1	2	3	4	5	6	
i=0.2	0.17	0.35	0.52	0.70	0.86	**1.05**	$\frac{d_{max}}{R}$
i=0.4	0.21	0.42	0.63	0.84	**1.05**	1.26	
i=0.6	0.26	0.51	0.77	**1.02**	1.28	1.54	
i=0.8	0.31	0.61	**0.92**	1.23	1.53	1.84	
i=1.0	0.37	0.73	**1.10**	1.46	1.83	2.19	



	1	2	3	4	5	6	
i=0.2	0.29	0.59	**0.88**	1.17	1.46	1.76	$\frac{d_{max}}{R}$
i=0.4	0.41	**0.82**	1.23	1.64	2.05	2.46	
i=0.6	0.50	**1.00**	1.50	2.00	2.50	3.00	
i=0.8	0.59	**1.17**	1.76	2.35	2.94	3.52	
i=1.0	**0.67**	1.34	2.02	2.68	3.36	4.03	

somewhat different. The ratio $d_{\max}/R$ is close to 1, but there is still room for improvement, since without any pipelining, that is, for $P = 1$, all versions of `ampseq1` become faster by level-clocking.

Another conclusion that we can draw from the plots in Figure 3 is that two-phase clocking leads to greater speedups when the gate delays are more uniform. For every test circuit, peak speedups increase as the exponent i decreases, that is, as the gate delays become more uniform. This observation suggests that standard-cell designs in which gate delays are roughly equal are likely to benefit from two-phase clocking.

The data shown in Figure 3 are the results of experiment SP3, in which the two-phase circuits were clocked by symmetric clocking schemes. We also performed experiment SP4 that combines retiming with tuning of the clocking schemes. In all cases, however, OP3 did not provide any speedup greater than 2% over OP2. Thus, our experiments suggest that clocking with asymmetrical schemes often does not provide any speed advantage over symmetric schemes.

3 Latch-minimization experiments

In this section we present our experimental comparison of edge-triggering and two-phase clocking in terms of the number of storage elements required by each implementation approach. We first describe our methodology, and then we present and discuss our experimental results.

Experimental methodology

In our experiments, we employed retiming in order to minimize the number of storage elements in the circuits. We retimed both the original edge-triggered circuits and their corresponding two-phase circuits in order to achieve a given clock period with the minimum number of storage elements. In both cases, the retiming transformation was applied without relocating the I/O storage elements of the circuits, and thus the I/O specification remained unchanged.

We compared the two implementations of each circuit by performing experiments LM1 and LM2.

LM1 We retimed the original edge-triggered circuit, in order to achieve the minimum period possible with the minimum number of flip-flops. Then, we retimed the corresponding two-phase circuit, in order to achieve the same period with the minimum number of latches, and we compared the number of storage elements in the two circuits.

LM2 We retimed the original edge-triggered circuit, in order to achieve its original clock period specification with the minimum number of flip-flops. Then, we retimed the corresponding two-phase circuit, in order to achieve the same period with the minimum number of latches, and we compared the number of storage elements in the two circuits.

The motivation behind these two experiments was to investigate the impact of two-phase clocking on the number of storage elements under different conditions of operation. Experiment LM1 was aimed at the typical situation, where speed is the primary concern, and edge-triggered circuits are configured to operate at the maximum of their potential. It is often the case, however, that the clock period is dictated by external system considerations and cannot be changed easily. To

Circuit	clock period	e-t latch count	l-c latch count	reduction
`mult16a`	15.465	62	51	18%
`s344`	20.946	46	34	26%
`s349`	20.945	46	34	26%
`s382`	11.567	68	42	38%
`s386`	22.999	12	12	0%
`s400`	11.759	64	42	34%
`s510`	18.740	16	14	12%
`s641`	95.464	38	38	0%
`s820`	31.993	10	10	0%
`ampseq1`	11.773	86	74	14%
`ampseq3`	9.700	174	152	13%
`ampseq4`	15.004	98	76	22%
`DRAM-ctl`	8.692	52	32	38%

Figure 4: Results of experiment LM1. In this experiment, each edge-triggered circuit and its corresponding two-phase circuit operate at the minimum clock period that can be achieved by retiming the original edge-triggered circuit. For each circuit, the table gives its operating period, the minimum number of latches in the edge-triggered implementation after retiming, the minimum number of latches in the two-phase implementation after retiming, and the reduction in the number of storage elements with respect to edge-triggering. The number of latches in each edge-triggered circuit is twice its number of flip-flops.

that effect, we also performed experiment LM2, which compares the number of storage elements in the two implementations when the clock period equals that of the original edge-triggered circuit.

We performed our experiments using TIM. The latch-minimization algorithms in TIM run in polynomial time and take into account maximal sharing of storage elements. We ran our tests on MCNC benchmark circuits, AT&T communication circuits and custom circuitry designed for MIT's Alewife machine. The largest among these circuits had 340 gates, and the current version of TIM retimed that circuit to minimize its number of latches in less than 20 minutes on a 16MB SPARCstation.

Experimental results

Our experimental results for the two sets of experiments are shown in Figures 4 and 5. There is a striking difference betwen these two sets of results. When the operating period is the minimum clock period that can be achieved by retiming the original edge-triggered circuit, then two-phase clocking leads to substantial reductions in the number of storage elements. When the operating period is that specified for the original circuit, however, there are almost no gains in the number of storage elements when we switch from edge-triggering to two-phase clocking.

In the experimental results of Figure 4 the greatest reductions were achieved for two controller circuits. The number of latches in both `s382` and the DRAM controller `DRAM-ctl` of the Alewife machine was reduced by 38%. Substantial reductions were also achieved for the multiplier circuits `mult16a`, `s344`, `s349`, for the controller circuit `s400`, as well as for the communication circuits `ampseq1`, `ampseq3`, and `ampseq4`. The two circuits `s641` and `s820` for which the number of storage elements did not decrease by two-phase clocking were PLD's.

Circuit	clock period	e-t latch count	l-c latch count
`mult16a`	66.987	32	32
`s344`	28.579	30	30
`s349`	28.579	30	30
`s382`	28.962	32	32
`s386`	22.999	12	12
`s400`	19.272	32	32
`s510`	20.369	12	12
`s641`	95.464	38	38
`s820`	31.993	10	10
`ampseq1`	17.041	70	70
`ampseq3`	12.041	144	141
`ampseq4`	23.087	64	64
`DRAM-ctl`	8.794	32	32

Figure 5: Results of experiment LM2. In this experiment, each edge-triggered circuit and its corresponding two-phase circuit are clocked at the original clock period specification of the edge-triggered circuit. For each circuit, the table gives its operating period, the minimum number of latches in the edge-triggered implementation after retiming, and the minimum number of latches in the two-phase implementation after retiming. The number of latches in each edge-triggered circuit is twice its number of flip-flops. Note that the level-clocked latch count decreases only for `ampseq3`.

Figure 5 shows that for all circuits except `ampseq3`, there was no reduction in the number of storage elements when the circuits were operating at the clock period specification of the original circuit. This seemingly negative result can be explained by comparing the clock periods of the original and the optimally retimed designs. In most cases, the original circuits operate substantially slower than the optimally retimed circuits. Most notably, the original `mult16a` is almost four times slower than its minimum-period retimed version. When the original clock period specification is so far from the minimum achievable, the placement of the storage elements in the edge-triggered circuit is as flexible as in the two-phase implementation, and thus no additional reductions are achieved by two-phase clocking. In the optimally retimed edge-triggered circuits, however, the minimum number of storage elements increases substantially, as it can be verified by comparing the columns that give the edge-triggered latch counts in Figures 4 and 5. Two-phase clocking can decrease this number without degrading circuit performance. In fact, as it is evident from the columns that give the level-clocked latch counts in Figures 4 and 5, the number of latches in more than half of the aggressive level-clocked implementations is not more than 15% higher than the number of latches in the low-performance implementations.

4 Conclusion – Work in progress

We compared edge-triggering and two-phase clocking in terms of speed and number of storage elements. Our methodology was independent of the functionality of the circuit and compared the two design approaches based on the effects of the storage elements in each one of them. In our speedup experiments, edge-triggering was often as fast as two-phase clocking, except when the average delay between any two consecutive latches was roughly uniform over the entire circuit and equal to the maximum gate delay, in which case the potential of two-phase clocking was generally obtained. Our experimental results suggest that circuits designed with standard cells of uniform delay benefit more from two-phase clocking. Moreover, symmetric clocking schemes seem to perform as well as tuned clocking schemes.

In terms of number of storage elements, two-phase clocking led to substantial reductions when the target clock period was set aggressively to the minimum that could be achieved by retiming the original edge-triggered circuit.

We are currently in the process of collecting experimental data from more circuits. We plan to perform experiments with circuits that were originally designed as two-phase circuits. We also plan to perform experiments with circuits in which the nonuniformity in gate delays is amplified, in order to check if, in this case, tuning of the clocking schemes provides any speedups. Another interesting question that remains open, and we plan to address, is whether asymmetric clocking schemes can decrease the number of storage elements further than symmetric ones.

Acknowledgements

We would like to thank Charles Leiserson of MIT for his ideas and suggestions throughout the course of this work. Thanks to Alex Ishii of NEC and Srinivas Devadas of MIT for helpful discussions and for providing us with test circuits. Thanks to John Kubiatowicz of MIT for his suggestions from the designer's perspective and for providing us with circuitry from the Alewife machine. Thanks also to Tom Szymanski of AT&T Bell Labs for enlightening discussions. Marios Papaefthymiou was supported in part by the Defense Advanced Research Projects Agency under Grant N00014-91-J-1698. Keith Randall was supported in part through the Undergraduate Research Opportunities Program at MIT.

A Upper bound on relative speedup

In this section we prove an upper bound on the relative speedup that can be achieved by two-phase level-clocking over edge-triggering. This bound is expressed in terms of the maximum gate delay $d_{\max}$ and the maximum ratio R of total delay over total number of storage elements around the cycles of the edge-triggered circuit.

The following two theorems give bounds on the clock period that can be achieved by retiming and tuning. The first theorem provides a lower bound and an upper bound for retiming edge-triggered circuitry.

Theorem 1 ([16]) *Let $G_{et} = \langle V, E, d, w \rangle$ be an edge-triggered circuit, with delay $d(v)$ for each gate $v \in V$ and $w(e)$ latches on each wire $e \in E$. Let*

$$R = \max_{C \in G_{et}} \frac{\sum_{v \in C} d(v)}{\sum_{e \in C} w(e)}$$

be the maximum ratio of delay over number of flip-flops around the cycles in G_{et}. Moreover, let d_{max} denote the maximum gate delay in G_{et}, and let $\Phi(G_{et})$ denote the minimum clock period that we can obtain by retiming G_{et}. Then

$$\max\{d_{max}, R\} \leq \Phi(G_{et}) \leq R + d_{max} \ . \qquad \square$$

The second theorem provides a lower bound for retiming and tuning two-phase circuitry.

Theorem 2 ([6]) *Let $G_{lc} = \langle V, E, d, w\rangle$ be a two-phase circuit, with delay $d(v)$ for each gate $v \in V$ and $w(e)$ latches on each wire $e \in E$. Let*

$$S = 2 \cdot \max_{C \in G_{lc}} \frac{\sum_{v \in C} d(v)}{\sum_{e \in C} w(e)}$$

be the maximum ratio of delay over number of latches around the cycles in G_{lc}. Then the minimum clock period $\Phi(G_{lc})$ we can obtain by retiming and tuning G_{lc} satisfies

$$\max\{d_{max}, S\} \leq \Phi(G_{lc}) \ . \qquad \square$$

Note that for a given edge-triggered circuit and its corresponding two-phase, level-clocked implementation that is obtained by replacing each edge-triggered flip-flop with a pair of level-sensitive latches, the lower bounds R and S are equal.

We use the bounds in Theorems 1 and 2 to prove the following upper bound on the speedup that can be achieved by switching from edge-triggering to level-clocking.

Lemma 1 *Let G_{et} be an edge-triggered circuit, and let G_{lc} be a two-phase circuit that is obtained by replacing each edge-triggered flip-flop in G_{et} by a pair of level-sensitive latches. Let $\Phi(G_{et})$ be the minimum clock period that we can achieve by retiming G_{et}, and let $\Phi(G_{lc})$ be the minimum clock period that we can achieve by retiming G_{lc} and simultaneously tuning its clocking scheme. Then*

$$\frac{\Phi(G_{et})}{\Phi(G_{lc})} \leq \frac{R + d_{\max}}{\max\{R, d_{\max}\}} \ . \qquad \square$$

The ratio $(R + d_{\max}) / \max\{R, d_{\max}\}$ can be used as a predictor of the speedup that may be achieved by two-phase clocking. The simple intuition behind this heuristic is that for greater values of the upper bound we have more space for improvement. There is no guarantee, however, on the actual improvement, since we have no lower bounds. The upper bound in maximized for $R = d_{\max}$. As we see in the experimental results of Section 2, it is when R approaches $d_{\max}$ that two-phase clocking becomes faster than edge-triggering.

B Retiming with minimum number of latches

In this section we consider the following problem: Given a level-clocked circuit G and a symmetric, k-phase clocking scheme π, compute a retimed circuit G_r that is properly timed by π and uses the minimum number of latches. We show that this problem can be solved in $O(V^3 \lg V)$ time by reducing it to the dual of an uncapacitated minimum-cost flow problem.

For the remainder of the section, we assume that along any path in a k-phase circuit, the phases that clock the latches appear in order modulo k. We also assume that in a symmetric clocking scheme with k phases, the duty cycle of every phase is ϕ, the rise times of any two consecutive phases are π/k units of time apart, and at least one of the k phases is low at any point in time.

The following lemma gives necessary and sufficient conditions for a k-phase circuit to be properly timed by a symmetric clocking scheme. Its proof is based on the notion of "rise-to-fall" times that has been introduced in [6].

Lemma 2 *Let $G = \langle V, E, d, w \rangle$ be a k-phase circuit, let π be a symmetric clocking scheme with k phases, and let ϕ be the duty cycle of each phase. Then the circuit G is properly timed by π if and only if for every path $u \overset{p}{\rightsquigarrow} v$ in G, we have*

$$d(p) \leq \frac{\pi}{k}\left(w(p)+1\right)+\phi \ . \tag{1}$$

$\square$

A circuit $G_r = \langle V, E, d, w_r \rangle$ is a retiming of another circuit $G = \langle V, E, d, w \rangle$ if, for some retiming function $r : V \rightarrow \mathbf{Z}$, we have $w_r(e) = w(e) + r(v) - r(u)$ and

$$w(e) + r(v) - r(u) \geq 0 \tag{2}$$

for every edge $u \xrightarrow{e} v \in E$. The variable $r(v)$ counts the *lag* of vertex v, and indicates the number of latches that the retiming process moves from every output of v to every input of v. The following lemma must hold for a retiming G_r of G to be properly timed by a symmetric clocking scheme π.

Lemma 3 *Let $G = \langle V, E, d, w \rangle$ be a k-phase circuit, and let π be a symmetric clocking scheme with k phases. Then there exists a retiming $r : V \rightarrow \mathbf{Z}$ of G such that G_r is properly timed by π if and only if there exists an assignment of an integer number $r(v)$ to each vertex $v \in V$ such that*

$$r(u) - r(v) \leq w(e) \tag{3}$$

for every edge $u \xrightarrow{e} v \in E$, and

$$r(u) - r(v) \leq \left\lfloor w(p) - \frac{k}{\pi}\left(d(p) - \phi\right) + 1 \right\rfloor \tag{4}$$

for every path $u \overset{p}{\rightsquigarrow} v$ in G. $\square$

The number of latches in the retimed circuit G_r is given by

$$\begin{aligned} \sum_{e \in E} w_r(e) &= \sum_{e \in E} w(e) + \sum_{e \in E} (r(v) - r(u)) \\ &= \sum_{e \in E} w(e) + \sum_{v \in V} \left(\text{indegree}(v) - \text{outdegree}(v)\right) r(v) \ . \end{aligned}$$

Therefore, the problem of retiming a k-phase circuit that is clocked with a symmetric scheme, in order to minimize the number of latches in it, is expressed as follows.

Definition 1 (Latch-minimization problem) *Let $G = \langle V, E, d, w \rangle$ be a level-clocked circuit, and let π be a symmetric clocking scheme with k phases. The latch-minimization problem asks for a retiming $r : V \rightarrow \mathbf{Z}$ of G such that*

$$\sum_{v \in V} \left(\mathit{indegree}(v) - \mathit{outdegree}(v)\right) r(v)$$

is minimized, subject to Inequality (3)

$$r(u) - r(v) \leq w(e)$$

for every edge $u \xrightarrow{e} v \in E$, and Inequality (4)

$$r(u) - r(v) \leq \left\lfloor w(p) - \frac{k}{\pi}\left(d(p) - \phi\right) + 1 \right\rfloor$$

for every path $u \overset{p}{\rightsquigarrow} v$ in G. $\square$

The following theorem shows that the latch-minimization problem is solvable in polynomial time.

Theorem 3 *The latch-minimization problem can be solved in $O(V^3 \lg V)$ time by reduction to the dual of an uncapacitated minimum-cost flow problem on the graph defined by Inequalities (3) and (4). The cost of each edge equals the right-hand side of the corresponding inequality. The demand/supply of each vertex v equals the difference between the number of edges coming into v and the number of edges coming out of v.*

If every fanout wire of each gate in the circuit has the same output, then we can show that the latch-minimization problem with maximal sharing of latches can be solved even more efficiently in $O(V^3)$ time. If the objective is to achieve the minimum clock period with the minimum number of latches, we can find a solution in $O(V^2E + V^3 \lg^2 V)$ time. First, we compute a set of $O(V^3)$ possible clock periods using the $O(V^2E)$ algorithm from [6], and then we binary search this set for the minimum feasible clock period.

References

[1] T. Burks, K. Sakallah, and T. Mudge. Multiphase retiming using $minT_c$. *τ92 ACM Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, March 1992.

[2] M. R. Dagenais and N. C. Rumin. Automatic determination of optimal clocking parameters in synchronous MOS VLSI circuits. In *Advanced Research in VLSI: Proc. of the 5th MIT Conference*, pages 19–33, 1988.

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

[4] R. B. Hitchcock. Timing verification and the timing analysis program. In *Proc. 19th Design Automation Conference*, pages 594–604, June 1982.

[5] A. T. Ishii and C. E. Leiserson. A timing analysis of level-clocked circuitry. In *Advanced Research in VLSI: Proc. of the Sixth MIT Conference*, pages 113–130. MIT Press, April 1990.

[6] A. T. Ishii, C. E. Leiserson, and M. C. Papaefthymiou. Optimizing two-phase, level-clocked circuitry. In *Advanced Research in VLSI and Parallel Systems: Proc. of the 1992 Brown/MIT Conference.* MIT Press, March 1992.

[7] N. P. Jouppi. Timing analysis for NMOS VLSI. In *Proc. 20th Design Automation Conference*, pages 411–418, June 1983.

[8] C. E. Leiserson, F. M. Rose, and J. B. Saxe. Optimizing synchronous circuitry by retiming. *3rd Caltech Conference on VLSI*, 1983. R. Bryant, ed., pp. 87-116.

[9] C. E. Leiserson and J. B. Saxe. Optimizing synchronous systems. *Journal of VLSI and Computer Systems*, 1(1):41–67, 1983.

[10] C. E. Leiserson and J. B. Saxe. Retiming synchronous circuitry. *Algorithmica*, 6(1), 1991. Also available as MIT/LCS/TM-372.

[11] B. Lockyear and C. Ebeling. Optimal retiming of multi-phase, level-clocked circuits. In *Advanced Research in VLSI and Parallel Systems: Proc. of the 1992 Brown/MIT Conference*. MIT Press, March 1992.

[12] T. M. McWilliams. Verification of timing constraints on large digital systems. In *Proc. 17th Design Automation Conference*, pages 139–147, June 1980.

[13] J. B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Proc. of the 20th Annual ACM Symposium on Theory of Computing*, pages 377–387, May 1988.

[14] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Prentice-Hall, Inc., 1982.

[15] M. C. Papaefthymiou. Understanding retiming through maximum average-weight cycles. *3rd ACM Symposium on Parallel Algorithms and Architectures*, July 1991.

[16] M. C. Papaefthymiou. Understanding retiming through maximum average-delay cycles. *Mathematical Systems Theory*, 1992. To appear. A preliminary version of the paper appeared in the proceedings of the 3rd ACM Symposium on Parallel Algorithms and Architectures, July 1991.

[17] M. C. Papaefthymiou and K. H. Randall. TIM: a timing package for two-phase, level-clocked circuitry. Unpublished manuscript, October 1992.

[18] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun. $checkT_c$ and $minT_c$: Timing verification and optimal clocking of synchronous digital circuits. In *Digest of Technical Papers of the 1990 IEEE International Conference on CAD*, pages 552–555, November 1990.

[19] N. Shenoy, R. K. Brayton, and A. Sangiovanni-Vincentelli. Retiming of circuits with single phase level-sensitive latches. In *International Conference on Computer Design*, October 1991.

[20] T. G. Szymanski. LEADOUT: A static timing analyzer for MOS circuits. In *Digest of Technical Papers of the 1986 IEEE International Conference on CAD*, pages 130–133, November 1986.

[21] T. G. Szymanski. Computing optimal clock schedules. In *Proc. 29th ACM/IEEE Design Automation Conference*, pages 399–404, June 1992.

[22] T. G. Szymanski and N. Shenoy. Verifying clock schedules. In *Digest of Technical Papers of the 1992 IEEE/ACM International Conference on CAD*, November 1992.

[23] S. H. Unger and C. J. Tan. Clocking schemes for high speed digital systems. *IEEE Transactions on Computers*, C-35(10):880–895, October 1986.

[24] S. A. Ward and R. H. Halstead, Jr. *Computation Structures*. McGraw-Hill, MIT Press, 1990.