

HIGH-SPEED ADDER DESIGN USING TIME BORROWING AND EARLY CARRY PROPAGATION

Gunok Jung and Gerald E. Sobelman

Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455, USA
Phone: (612) 625-8041, Fax: (612) 625-4583
e-mail: sobelman@ece.umn.edu

ABSTRACT

This paper presents a novel application of skew-tolerant domino circuit design by combining the effects of time borrowing and early carry propagation in a 64-bit adder. Skew-tolerant circuit design softens the clock edges and allows time borrowing from one stage of logic to the next in a pipeline. Early carry propagation also softens clock edges by allowing useful work to be done during the precharge phase. In this paper, these two effects are used simultaneously, resulting in a significant reduction in latency. Simulation results show that up to a 42% decrease in delay can be achieved over a traditional two-phase clocking design when both techniques are combined in the same circuit.

I. INTRODUCTION

Domino logic has become a popular choice for high speed circuit design. However, since these circuits are typically used with a two-phase clocking scheme, they experience the constraints and overheads from hard clock edges and the need for mid-cycle latches. Skew-tolerant domino CMOS circuit design has been recently introduced [1], [2], [3], [4] to overcome these limitations. In this approach, the computation during one phase can borrow time from a subsequent phase by using overlapping clocks and eliminating intermediate latches.

A different technique for softening clock edges in the context of adder design is known as early carry propagation [5]. In this approach, fighting is intentionally introduced while a circuit is precharging. As a result, intermediate nodes that will be discharged to ground during the evaluate phase are only precharged up to a fraction of the supply voltage. This, in turn, leads to a faster discharge time during the evaluate phase. In some sense, this can be viewed as complementary to time-borrowing, since a logic block, in

effect, makes use of additional time from the *preceding* stage of logic.

In this paper, we propose using a novel combination of these two speed-up techniques. We illustrate the idea for the specific case of a 64-bit EMODL (enhanced multiple output domino logic) carry look-ahead adder [6], [7]. In earlier work, we showed how time borrowing alone can be used to enhance the performance for this type of adder [8]. In this paper, we extend our previous work by demonstrating the effectiveness of using both time borrowing and early carry propagation in the same circuit. The performance is improved because we can make the optimum use of time from both the succeeding *and* the preceding stages of logic. Comparisons between several different design implementations, including a baseline design without time borrowing or early carry propagation, are presented.

II. TIME BORROWING IN SKEW-TOLERANT DOMINO

Traditional domino circuit design uses two phase clocking, with intermediate results stored in latches. As explained in References [1] and [2], this results in a significant clocking overhead due to skew, latch propagation delay and imbalances in the delays in different stages. In essence, the problem can be traced to the existence of hard clock edges that require intermediate computations to be completed well within their allotted phases.

On the other hand, skew-tolerant domino circuit design uses overlapping clock phases, which results in softer clock edges. One of the benefits of this is to enable time borrowing between adjacent phases. If two clock phases are used, then the only way to obtain overlapping phases is to use asymmetric clock waveforms having greater than a 50% duty cycle. However, if more than two phases are used, overlaps can be ob-

tained between adjacent phases with 50% duty cycle clocks. Systems with several clock phases are often criticized for having complex clock distribution problems, but the difficulties can be minimized through proper design. In particular, only a single global clock signal is distributed throughout the chip. The individual phases are derived locally in the vicinity of each module or functional unit. In this way, global clock routing congestion is limited and skews can be better managed.

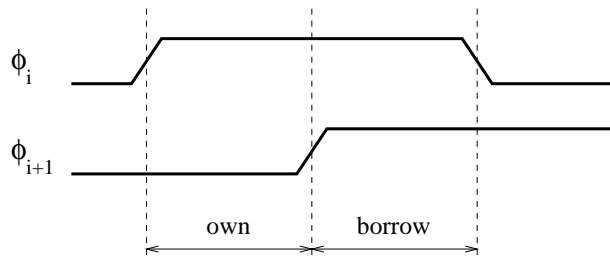


Figure 1: Time periods owned and potentially borrowed by a clock phase.

The basic concept involved in time borrowing can be seen from Figure 1. In this figure, two adjacent clock phases are shown with an overlap between them. If clock skew is present, then the rising edge of ϕ_{i+1} corresponds to the latest possible time for this edge, while the falling edge of ϕ_i corresponds to the earliest possible time for this edge. The interval when both clocks are high is the interval when time borrowing can occur. If there is no intermediate latch, then the logic that is clocked on ϕ_{i+1} simply waits for the evaluation wavefront to arrive from the previous stage. In this way, the logic that is clocked on ϕ_i effectively borrows time from the subsequent phase.

III. EARLY CARRY PROPAGATION IN SKEW-TOLERANT DOMINO

As shown above, time borrowing occurs in the forward direction, i.e. one stage of logic borrows time from its successor stage. As we will show, early carry propagation can be viewed as performing the complementary function of allowing time to be effectively borrowed from the preceding stage of logic.

Early carry propagation is usually discussed in connection with the Manchester carry chain or the regenerative carry chain [5]. The basic idea is to remove the NMOS evaluate or “foot” transistors from the pull-down networks of these carry chains, so that intentional fighting occurs during the precharge phase. The fighting is beneficial in the sense that it reduces

the precharge voltage on those nodes that will eventually be discharged to ground during the subsequent evaluate phase. Since the effective precharge voltage is reduced, it will take a shorter amount of time to discharge those nodes, which increases the speed of the carry computation. While the technique does dissipate DC power along the paths that are fighting, this is generally considered to be an acceptable trade-off in high-speed designs. Since the fighting occurs at only a small fraction of nodes on a complete chip, the incremental increase in power dissipation from early carry propagation will be relatively minor.

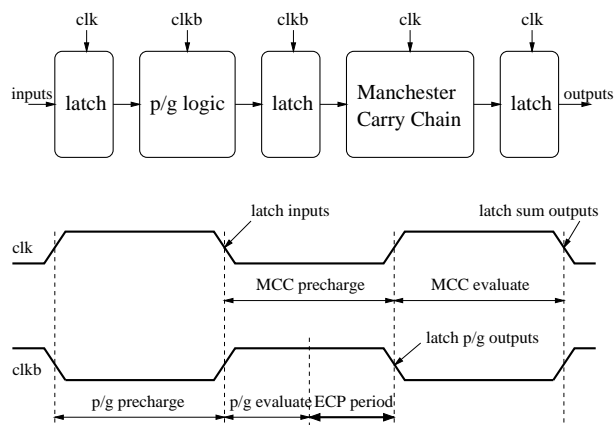


Figure 2: Structure and timing for a simple adder that supports early carry propagation.

Figure 2 shows the basic set-up for an adder design that allows early carry propagation. The propagate and generate (p/g) functions evaluate while the carry chain is precharging. Since these are relatively simple functions, they will complete their evaluations well before the end of clk_b . If any of the g values evaluates to 1, then there will be a fighting situation created at that point in the carry chain. Moreover, if a succession of p values are 1 connecting into that point of the carry chain, then the fighting will be propagated to other nodes in the chain, albeit at a somewhat reduced degree. The point is that all of the nodes that experience fighting are precisely those nodes that will be discharged to ground once the carry chain goes into its evaluation. Thus, they get a “head start” on this discharge by only being precharged up to a fraction of the supply voltage. In other words, the carry chain is effectively borrowing time from the p/g logic stage, which is the preceding stage of logic.

While the early carry propagation concept was illustrated above for the simple case of non-overlapping clocks, the same idea can be applied to the overlapping clocking schemes of interest in this paper. A typ-

ical situation is shown in Figure 3, where the “dead time” that occurs in the absence of early carry propagation can now be utilized. Notice that traditional time borrowing also takes place at another point in the cycle, so that both mechanisms are active in this case.

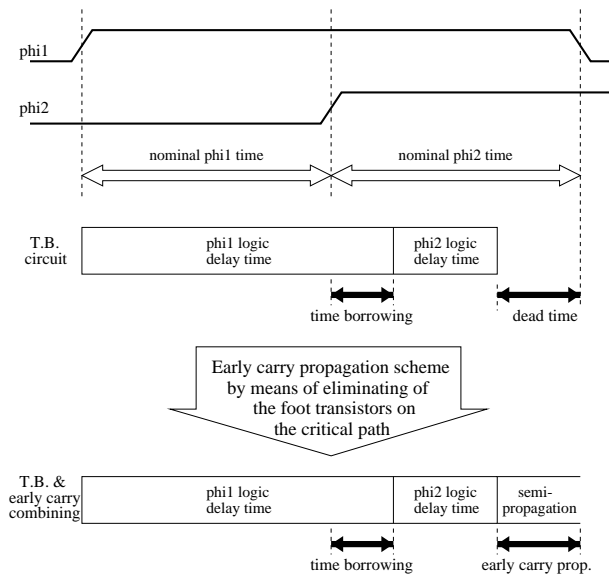


Figure 3: Early carry propagation with overlapping clocks.

IV. TIME BORROWING IN A 64-BIT EMODL CARRY LOOK-AHEAD ADDER

In this section, we show the speed advantage that can be obtained using time borrowing (but not early carry propagation) for the case of a 64-bit EMODL carry look-ahead adder [6], [7]. EMODL is a modification to the original concept of multiple output domino logic [9] that allows shared devices to be used to implement common sub-expressions that are not necessarily function outputs.

The partitioning of the 64-bit EMODL adder into clock phases is shown in the block diagram of Figure 4. Note that the figure shows the phase assignments for either a two-phase or a four-phase clocking scheme.

Our circuit simulations were performed using a 0.35 micron TSMC process that is available from MO-SIS. All of the simulations were performed under nominal process, temperature and voltage conditions. We used skewed static gates at the output of domino stages in order to improve the evaluation times. We also included estimated wire capacitances on the circuit nodes.

Figure 5 shows the simulation results for the case of traditional two-phase clocking. We have overhead

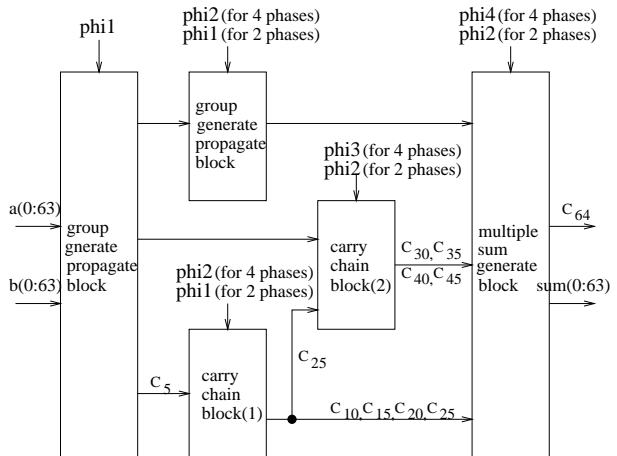


Figure 4: Partitioning the 64-bit EMODL adder.

from two latch delays and one phase imbalance (the “dead time”), which results in the relatively long cycle time of 2.04 ns.

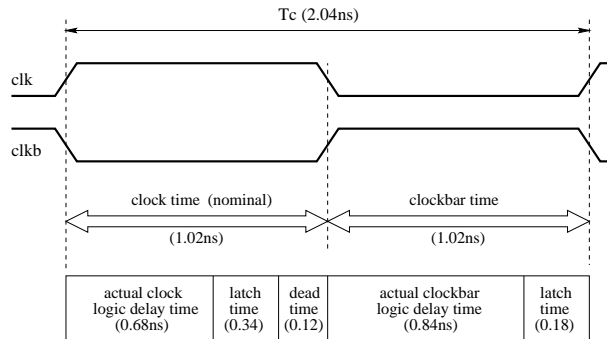


Figure 5: Simulation results for the 64-bit EMODL adder using traditional two-phase clocks.

The results for overlapping two-phase clocks are shown in in Figure 6. It can be seen that there is a certain amount of the overlap time can be used for time borrowing. However, in this case, the actual phase 1 delay time is less than the nominal phase 1 time, so there is no need to borrow any time from phase 2. Still, we obtain a shorter cycle time of 1.74 ns because of the elimination of the latch delays.

The use of overlapping four-phase clocks is illustrated in Figure 7. This configuration produced an even faster cycle time of 1.44 ns because of the time borrowing action. As shown in the figure, 0.10ns and 0.17ns time intervals were used to effectively balance the delays between the clock phases.

V. COMBINED TIME BORROWING AND EARLY CARRY PROPAGATION FOR THE 64-BIT EMODL ADDER

In this section, we will illustrate the further speed advantage that can be obtained by utilizing both time borrowing and early carry propagation. Specifically, we will consider the case of overlapping four-phase clocks, since this showed the best results for time borrowing alone. Early carry propagation is enabled by eliminating the foot device in various sub-circuits of the carry logic. One typical example of this is the module shown in in Figure 8.

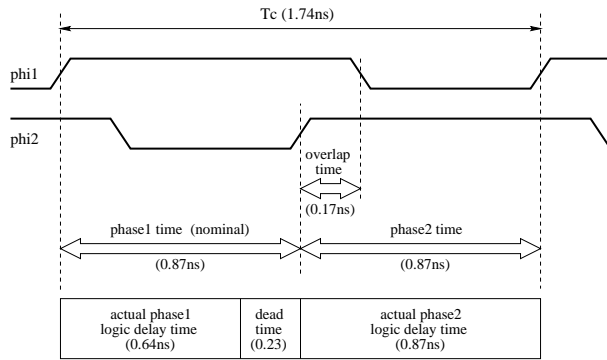


Figure 6: Time borrowing in the 64-bit EMODL adder using two-phase overlapping clocks.

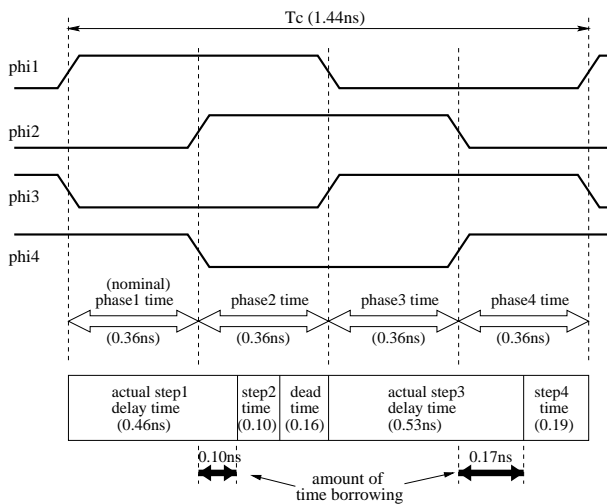


Figure 7: Time borrowing in the 64-bit EMODL adder using four-phase overlapping clocks.

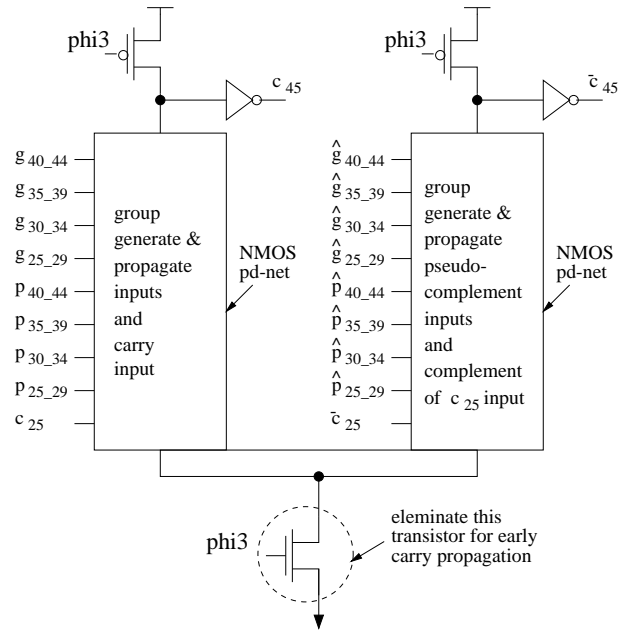


Figure 8: Circuit diagram for the carry chain(2) block as used with four-phase overlapping clocks.

The simulation results for this design are shown in Figure 9. It is observed that this configuration has the fastest cycle time of 1.32 ns. As can be seen, useful work is being performed during the entire period, and the “dead time” has been eliminated.

VI. PERFORMANCE COMPARISONS

The clock cycle times for the four different design/clocking implementations of the 64-bit EMODL carry look-ahead adder are summarized in Table 1. Treating the traditional two-phase clocking design as the baseline, the table also indicates the percentage improvement that is provided by each of the other three designs. The four-phase scheme with time borrowing and early carry propagation exhibits an im-

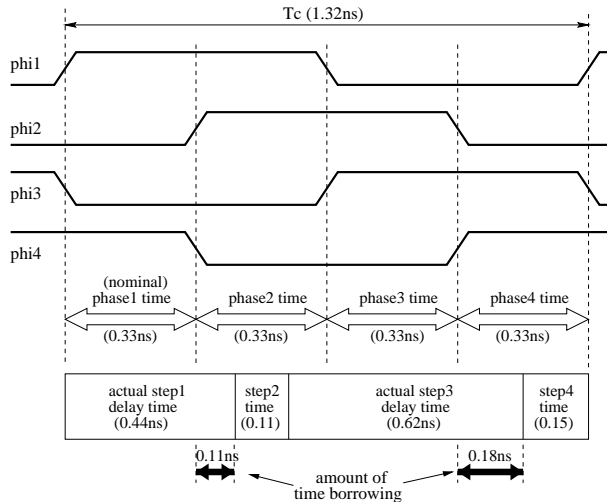


Figure 9: Combined time borrowing and early carry propagation in the 64-bit EMODL adder using four-phase overlapping clocks.

Table 1: Clock cycle time comparisons without clock skew.

Clock Systems	clock cycle time without clock skew	improved amount
2 phase traditional clock scheme	2.04 ns	0 %
2 phase overlapping clock scheme	1.74 ns	15 %
4 phase overlapping clock scheme	1.44 ns	29 %
4 phase overlapping & ECP combining	1.32 ns	35 %

pressive 35% speed-up over the baseline design.

We also considered the effect of non-zero clock skew on these circuits. Specifically, we simulated each of the four circuits assuming a relative skew of 0.17 ns, which corresponds to 1 FO4 (fanout-of-4) delay in our technology. As expected, the traditional two-phase clock cycle time is degraded in this situation, while the cycle times for the three skew-tolerant design are not affected. Thus, we realize an even larger percentage improvement of 42% for the four-phase skew-tolerant, early carry propagation design compared to the baseline design in the presence of this skew.

VII. REFERENCES

[1] D. Harris and M. A. Horowitz, “Skew-Tolerant Domino Circuits,” IEEE Journal of Solid-State Circuits, Vol. 32, No. 11, pp. 1702-1711, November, 1997.

Table 2: Clock cycle time comparisons with clock skew.

Clock Systems	clock cycle time with 0.17ns skew	improved amount
2 phase traditional clock scheme	2.26 ns	0 %
2 phase overlapping clock scheme	1.74 ns	23 %
4 phase overlapping clock scheme	1.44 ns	36 %
4 phase overlapping & ECP combining	1.32 ns	42 %

[2] D. Harris, “Skew-Tolerant Circuit Design,” Ph.D. Thesis, Stanford University, November, 1998.

[3] D. Harris et al, “Opportunistic Time-Borrowing Domino Logic,” U.S. Patent No. 5,517,136, May 14, 1996.

[4] Kerry Bernstein et al, “High Speed CMOS Design Styles,” Chapter 8, Kluwer Academic Publishers, 1998.

[5] R. F. Hobson, *Refinements to the Regenerative Carry Scheme*, IEEE Journal of Solid-State Circuits, Vol. 24, No. 6, pp. 1759-1762, 1989.

[6] J. Wang et al, *Area-Time Analysis of Carry Lookahead Adders Using Enhanced Multiple Output Domino Logic*, Proceedings, IEEE International Symposium on Circuits and Systems, Vol. 4, pp. 59-62, 1994.

[7] J. Wang et al, *Fast Adders using Enhanced Multiple-Output Domino Logic*, IEEE Journal of Solid-State Circuits, Vol. 32, No. 2, pp. 206-214, February, 1997.

[8] Gunok Jung, Victoria Perpelitsa and Gerald E. Sobelman, *Time Borrowing in High-Speed Functional Units Using Skew-Tolerant Domino Circuits*, to appear in Proceedings, IEEE Symposium on Circuits and Systems, May, 2000.

[9] S. Waser and M. J. Flynn, *Introduction to Arithmetic for Digital System Design*, Chapter 3, CBS Publishing, 1982.

[10] I. S. Hwang and A. L. Fisher, “Ultra Fast Compact 32-bit CMOS Adder in Multiple-Output Domino Logic,” IEEE Journal of Solid-State Circuits, Vol. 24, pp. 358-369, 1989.