TIME BORROWING IN HIGH-SPEED FUNCTIONAL UNITS USING SKEW-TOLERANT DOMINO CIRCUITS

Gunok Jung, Victoria Perepelitsa and Gerald E. Sobelman

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

ABSTRACT

We present results on time borrowing in skew-tolerant domino logic circuits for a 32-bit adder, a 64-bit adder and a 32-bit pipelined multiplier. The adders are built using enhanced multiple output domino logic and the multiplier uses modified Booth encoding and a Wallace tree. We illustrate how the resulting soft clock edges allow advantageous time borrowing to occur in these functional units. In this way, limitations due to delay imbalances between stages are removed, allowing the circuits to operate at a higher speed.

1. INTRODUCTION

Skew-tolerant domino CMOS circuit design has been recently introduced as a means for overcoming the limitations of imperfect clock distribution in high-speed digital chips [1], [2]. The basic idea is to make use of overlapping clock phases in such a way that the circuit becomes insensitive to skew.

On the other hand, in systems where skew is carefully controlled through precise engineering of the clock distribution network, the same mechanism can be applied to enhance performance through the use of time borrowing between adjacent clock phases. Time borrowing has been traditionally used in latched-based designs using static CMOS combinational logic. Discussion of time borrowing in conjunction with domino CMOS has not appeared in the open literature until recently [1], [2], [3], [4].

In this paper, we present design examples of this technique for three important functional units, namely a 32-bit adder, a 64-bit adder and a 32-bit multiplier. Our clocking scheme uses four overlapping phases, where each clock phase has a 50% duty cycle. In the case of the adder circuits, we make use of the recently introduced EMODL (enhanced multiple output domino logic) circuit technique with a carry look-ahead architecture to achieve a high-speed single-cycle design [5], [6]. For the multiplier, we consider a highly pipelined design using modified Booth encoding and a Wallace tree [7]. This multiplier could be used in feed-forward data paths requiring high-throughput, such as in digital signal processing applications involving FIR filters or orthogonal transforms.

In these three design examples, we show how the logic can be partitioned into blocks that are nominally associated with each of the four clock phases. Moreover, we point out specific instances where time borrowing occurs, resulting in an increase in operational speed. The overlapping phases give rise to softer clock edges that allow blocks to continue evaluating beyond their nominal phase time. In this way, we avoid wasteful delays associated with the inevitable imbalances in the propagation delay associated with each stage of a pipeline.

This paper is organized as follows: In Section 2, the key ideas of skew-tolerant domino CMOS circuits and time borrowing are briefly reviewed. Then, in Section 3, we present our design and simulation results for time borrowing in the EMODL adders. Section 4 contains our design and simulation results for the 32-bit multiplier. Finally, in Section 5, we present our conclusions.

2. TIME BORROWING IN SKEW-TOLERANT DOMINO

Conventional or “textbook” domino circuit design uses two phase clocking in which the precharge and evaluation of adjacent blocks of logic occurs on alternate phases and where the intermediate results are stored in transparent 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 blocks. 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 resulting 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 obtained 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. All of the designs in this paper utilize a clocking scheme having four overlapping clock phases, each of which has a 50% duty cycle.

The basic concept involved in time borrowing can be seen from Figure 1. In the 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 when time borrowing can occur.

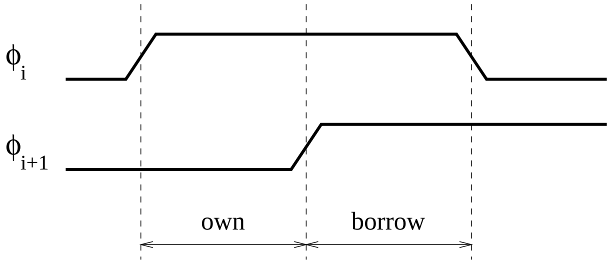

Figure 1: Time periods owned and potentially borrowed by a clock phase.

Consider two adjacent logic modules, A and B, in which A evaluates when ϕ_i is high, B evaluates when ϕ_{i+1} is high and where the inputs to B come from the outputs of A. The nominal evaluation time for A is the period labeled as “own” in the figure. However, if necessary, the evaluation of A can extend into the region labeled as “borrow” in the figure. If the evaluation of A has not completed, its output(s) will be low since A is a domino circuit. This, in turn, will temporarily suspend the evaluation of B since all of the logical inputs to B are low. As soon as the result from A becomes available, then B will immediately start its own evaluation. Thus, the boundary between when A finishes and B begins is fuzzy, i.e. the clock edges have been effectively softened. This effect enables the circuit to automatically adjust to the imbalances that inevitably exist between the propagation delays of adjacent stages. Hence, the clock period does not have to be degraded to accommodate the worst-case stage delay, which improves the overall speed of the circuit.

3. TIME BORROWING IN EMODL CARRY LOOK-AHEAD ADDERS

In this section, we present our design and simulation results for 32-bit and 64-bit single-cycle carry look-ahead adders that are based on EMODL functional blocks [5], [6]. EMODL is a modification to the original concept of multiple output domino logic [8] that allows shared devices to be used to implement common sub-expressions that are not necessarily function outputs.

The partitioning of the 32-bit EMODL adder is shown in the block diagram of Figure 2. In this design, only the first three phases are nominally allocated to the addition operation itself, with the fourth phase just latching the result. Specifically, the generate, propagate, group generate and group propagate functions are formed by circuits clocked on ϕ_1. These values are then fed into the carry chain, which is clocked on ϕ_2. Finally, the sum values are created in circuits that are clocked on ϕ_3.

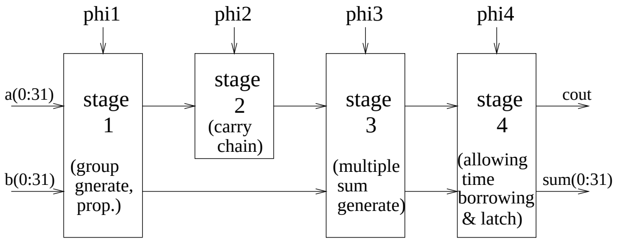

Figure 2: Partitioning the 32-bit EMODL adder into 4 phases.

However, as shown in the simulation results of Figure 3, the actual operation is enhanced through the use time borrowing. Due to delay imbalances within the circuit, the nominal ϕ_1 computations borrow 0.039 ns from ϕ_2, the nominal ϕ_2 computations borrow 0.187 ns from ϕ_3 and the nominal ϕ_3 computations borrow 0.11 ns from ϕ_4. However, stage 4 still completes by the end of ϕ_4.

The 64-bit EMODL adder has a more complex logical structure, and the partitioning has been done differently, as shown in Figure 4. Here, all four phases are required to compute the results, since two phases are allocated to the operation of the carry chain. Figure 5 shows the time borrowing that occurs in this design.

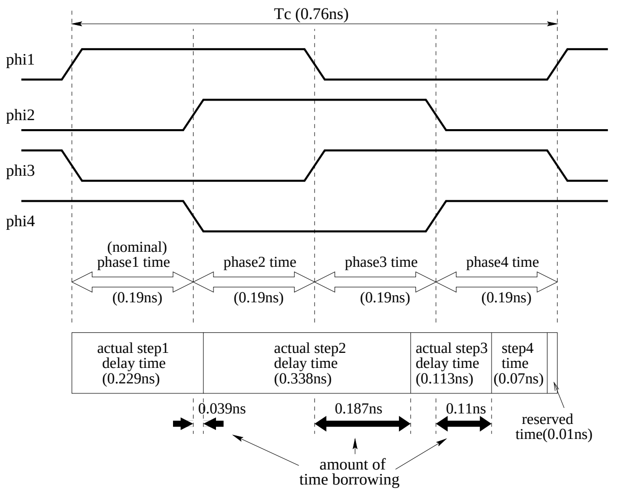

Figure 3: Time borrowing in the 32-bit EMODL adder.

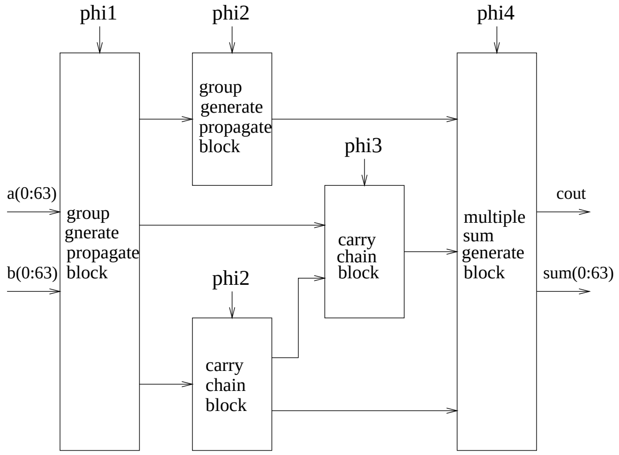

Figure 4: Partitioning the 64-bit EMODL adder into 4 phases.

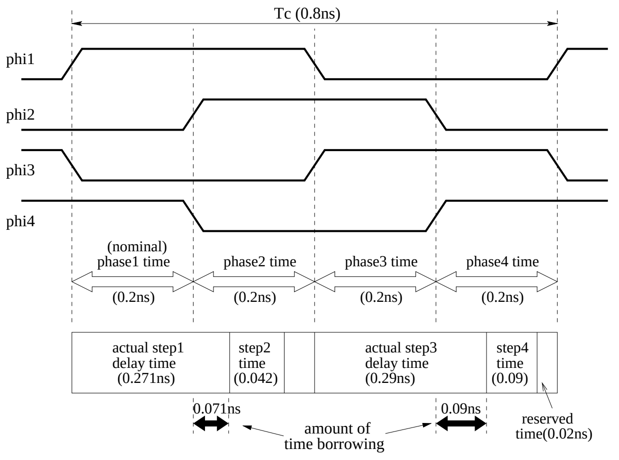

Figure 5: Time borrowing in the 64-bit EMODL adder.

4. TIME BORROWING IN A PIPELINED MULTIPLIER

As a final example, we consider the design of a deeply pipelined 32-bit by 32-bit parallel multiplier. We use a conventional architecture based on modified Booth encoding to generate a reduced number of partial products and a Wallace tree to sum the partial products [7]. Each modified Booth encoder cell examines three contiguous bits of the multipier operand, X, to determine whether to add 0, +1, -1, +2 or -2 times the multiplicand, Y. This is accomplished in several steps: During ϕ_2 of the first cycle, a set of three control signals are generated: *n* to invert the multiplicand, *m1* to multiply the multiplicand by 1 and *m2* to multiply the multiplicand by 2. The equations for the above three functions are:

$$\begin{aligned} m_1 &= x_{2i} \oplus x_{2i-1} && (1)\\ m_2 &= x_{2i+1}x'_{2i}x'_{2i-1} + x'_{2i+1}x_{2i}x_{2i-1} && (2)\\ n &= x_{2i+1}(x_{2i}x_{2i-1})' && (3) \end{aligned}$$

These three signals are used to control the operation of select cells that send the correct partial product bit into the Wallace tree. The governing equation is:

$$pp_i = (m_1 y_i + m_2 y_{i-1}) \oplus n \qquad (4)$$

The selection process is mapped into the last two clock phases of the first cycle. During ϕ_3, the intermediate quantities $p_i = m_1 y_i + m_2 y_{i-1}$ are computed. Then, during ϕ_4, the partial product bits $pp_i = p_i \oplus n$ are created.

The Wallace tree implements a series of 3-to-2 compressions on the set of partial products. This occurs over 6 phases (i.e., 1.5 cycles), and spans the entire second cycle plus the first half of the third cycle.

The final two vectors produced by the Wallace tree are summed in a fast 64-bit adder that uses a carry-select architecture [9]. The adder is divided into 16-bit carry lookahead (CLA) sections. A single 16-bit CLA section is used to sum the least significant 16 bits. However, each of the other three 16-bit sections use two CLAs, one that assumes a carry-in of 0 and another that assumes a carry-in of 1. In each case, the outputs from the appropriate CLA are selected once the carry-in values become known. This final set of selections, together with latching of the 64 product bits, is accomplished in the final MUX/latch block at the bottom of the block diagram. The CLAs and MUX/latch blocks operate during the final 2.5 clock cycles.

Some of the time borrowing effects are illustrated by the simulation waveforms for the first clock cycle, as shown in Figure 7. During this period of time, the input operands are latched, the modified Booth encoder outputs are generated and the partial products are formed. Note that the full clock

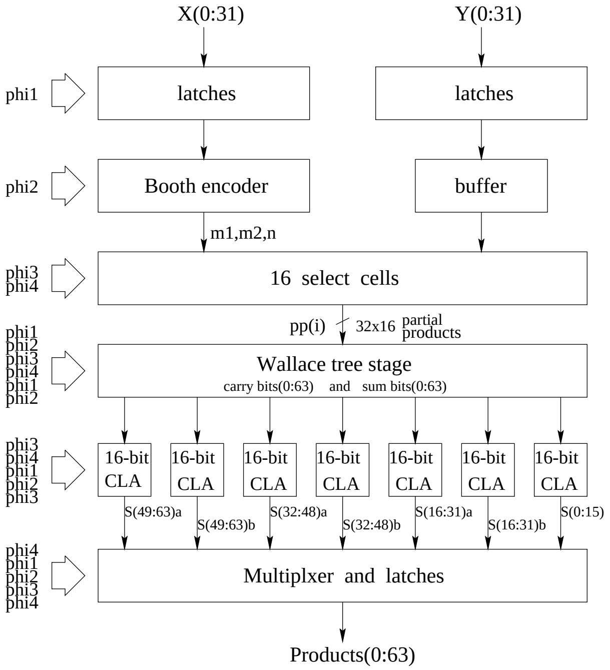

Figure 6: Partitioning the 32-bit multiplier into 5 cycles.

cycle time, Tc, is 0.68 ns, so that each nominal phase time is 0.17 ns. We can see that the modified Booth encoding, which is allocated to ϕ_2, borrows 0.135 ns from the nominal ϕ_3 time. There are also two other minor instances of time borrowing during the cycle, as shown in the figure. (Step 5, which occurs during the next ϕ_1, is the initial stage of the Wallace tree computation.)

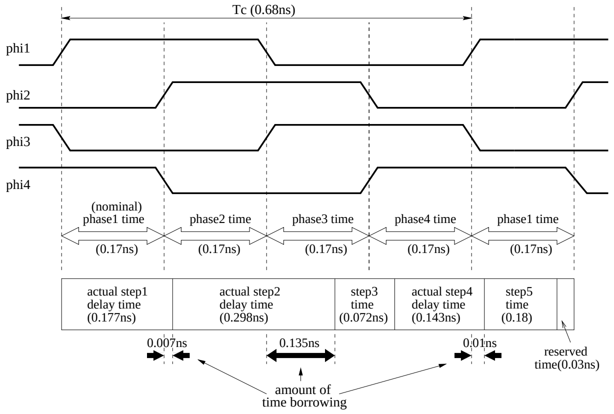

Figure 7: Time borrowing in the first cycle for the 32-bit multiplier.

5. CONCLUSIONS

We have demonstrated the issues and benefits arising from time borrowing in skew-tolerant domino circuits for three important functional units. The 32-bit and 64-bit EMODL adders were mapped into single-cycle designs using four 50% duty cycle clock phases. We also considered a pipelined multiplier which used a total of five cycles of the same overlapping clocking scheme. For these three designs, we showed how and where time borrowing occurs. The time borrowing phenomenon effectively creates softer clock edges which relaxes the need to carefully balance the delays in a pipeline, thereby leading to faster circuits.

6. 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.

[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] 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.

[6] Z. 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.

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

[8] 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.

[9] O. J. Bedrij, "Carry-Select Adder," *IRE Trans. on Electronic Computers*, Vol. EC-11, pp. 340-346, 1962.