

DESIGN TECHNIQUES FOR PULSED STATIC CMOS

Kavitha Seshadri, Adrianne Pontarelli, Gauri Joglekar and Gerald E. Sobelman
Department of Electrical and Computer Engineering,
University of Minnesota, Minneapolis, MN U.S.A.
{kavi3480, sobelman}@ece.umn.edu

ABSTRACT

This paper gives new results in the design of Pulsed Static CMOS circuits. In particular, a new method of circuit duplication has been proposed which is particularly useful for the implementation of arithmetic functions. An array multiplier and a carry-select adder are used as representative design examples. Simulation results confirm that these Pulsed Static CMOS circuits operate correctly and have greater throughput than traditional static designs.

1. INTRODUCTION

Digital CMOS circuits are implemented using either static or dynamic design techniques. In static CMOS, the output is tied to V_{DD} or ground via a low resistance path (except during switching) and this leads to circuits which are very robust with good noise immunity. Dynamic circuits on the other hand are less stable and more susceptible to noise due to the presence of high impedance circuit nodes and charge sharing effects. The main limitation of static circuits is their slower speed as compared to dynamic circuits. The reasons for this include increased gate capacitance (due to the presence of both PMOS and NMOS transistors), output dependence on the previous cycle's inputs (due to charges that may be present at internal nodes) and multiple switching of the output within a cycle (depending on the input switching pattern) [1].

Pulsed Static CMOS (PS-CMOS) circuits combine the advantages of both static and dynamic circuits in being faster than traditional static designs and having better noise immunity than dynamic designs. The patent of Chen and Ditlow [1] gives a description of the PS-CMOS design technique and its advantages. We have extended these concepts and proposed a new method of circuit duplication which is particularly useful when applying the technique to arithmetic functions.

The remainder of this paper is organized as follows: Section 2 gives a brief review of PS-CMOS. The proposed method of duplication is explained in Section 3. In Sections 4 and 5 we present two circuits that have employed this method along with simulation results. Finally, we summarize our conclusions in Section 6.

2. PULSED STATIC CMOS

The idea underlying Pulsed Static CMOS (PS-CMOS) design is to improve the speed of static circuits

through the use of node pre-conditioning. One of the main limitations of static CMOS circuits is the need to charge or discharge the output node through a series chain of transistors. Such devices must be made larger in order to reduce the total on-resistance of the path, but this leads to increased gate capacitances which adds to the overall delay budget. In PS-CMOS, the static circuitry operates in such a way that signal evaluation through a chain of series transistors is minimized. This is achieved by pre-conditioning the static circuits in a manner that resembles pre-charging in dynamic circuits [1]. The pre-conditioning process involves the propagation of two input patterns through the static circuit; one pattern causes the circuit to evaluate and hold its output and the second pattern causes the circuit to be reset. The circuit is reset into a state from which its subsequent evaluation will be fast, i.e. one which does not require charging or discharging through a series chain.

Static logic circuits predominantly consist of a combination of NAND, NOR and NOT gates. Hence, it is crucial to reduce the evaluation time of these gates in order to ensure circuit speedup. On analyzing NAND and NOR gates it is seen that NAND gates have series NMOS transistors (pull-down path is penalized) and NOR gates have series PMOS transistors (pull-up path is penalized). Therefore, during the reset phase, it is favorable to preset the NAND outputs to a reset low level and the NOR outputs to a reset high level, in order to minimize the evaluation time. As a result of this, NAND gates are fed by tri-state inverters which are reset high elements and NOR gates are fed by reset-low elements [1].

Earlier work [1] has shown that a series chain of alternating NAND and NOR static gates would be the fastest and the most optimum topology. This optimal topology is difficult to realize for many applications and so in many cases we settle for less optimal combinations. However, we must still ensure that the inputs to a given static gate are all at the same logic level during the reset phase. In certain logic circuits, some components do not meet this criterion. When this happens, the circuit does not exhibit the required PS-CMOS behavior. In order to overcome this problem, we have proposed a new method of circuit duplication and this is explained in the next section.

3. METHOD OF DUPLICATION

As mentioned in the previous section, it is essential that all of the inputs to any given static logic gate be at the same logic level during the reset phase in order to

satisfy the PS-CMOS criterion. This criterion is not satisfied in static logic circuits that use XOR functions or multiplexers. An XOR implementation using NAND and NOR gates is shown in Fig. 1.

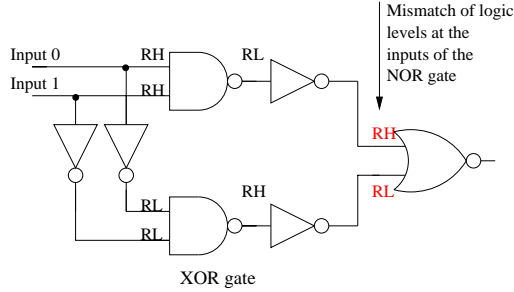


Fig. 1: Reset levels in an XOR implementation.

The inputs shown in Fig. 1 are obtained after propagation through static latches and tri-state inverters. The choice of reset high or reset low tri-state inverters depends on the gates to which the inputs are fed. In accordance with this, the inputs are generated as reset high (RH) or reset low (RL) signals and fed to the first level of logic gates. The mismatch in logic levels of the inputs occurs at the last level (the NOR gate), and hence the criterion is violated.

Any other static implementation of the XOR function would also have a mismatch somewhere within the circuit, leading to glitches in the output. Thus, even though the circuit would function correctly, the basic principle of PS-CMOS would not be satisfied and hence it will not be faster than the static design. The same situation occurs in multiplexers as well. As XOR functions and multiplexers are building blocks of many arithmetic functions, it is important to find a way to address this issue. In this regard, we have developed an innovative method of logic duplication where both reset high and reset low elements are used for all the inputs. The modified architecture of the XOR gate with duplication is shown in Fig. 2.

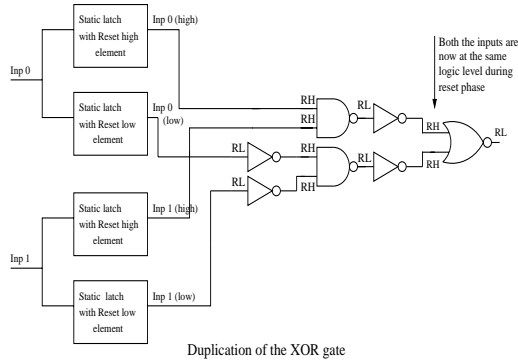


Fig. 2: XOR implementation after duplication.

Through the duplication of the circuitry, each input is generated as a reset high and as a reset low signal by both the tri-state inverters and is given to the gates in the XOR architecture. As a result of this, both the inputs to the NOR gate are observed to be at the same level during the reset phase and hence the criterion for PS-CMOS design is satisfied.

Although, the duplication of circuitry results in more area and a greater number of transistors when compared to a static or dynamic circuit, faster evaluation and an associated improvement in throughput is achieved. This has been demonstrated using two representative circuits, namely a 4-bit combinational multiplier and an 8-bit carry-select adder, which are described in the next two sections.

4. 4-BIT PS-CMOS MULTIPLIER

A 4-bit combinational multiplier uses an array of AND gates to generate the partial products in parallel and a sequence of half adders and full adders to sum them, as shown in Fig. 3 [2].

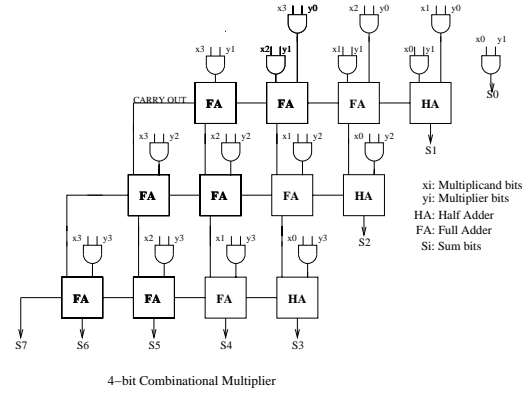


Fig. 3: Architecture of the 4-bit array multiplier.

In static CMOS, the AND and OR gates would be replaced by NAND and NOR gates followed by inverters. In the PS-CMOS design, duplication is incorporated, as both the half and full adders include XOR operations. Hence, all the inputs to the half adders and full adders are generated as both reset high and reset low signals. This requires the duplication of the partial product generation stage as the outputs generated by this stage are the inputs to the various adders. In addition to this we need to duplicate all of the half and full adders as they are connected in tandem. A full adder circuit with the required duplication is shown in Fig. 4.

The resulting space-time diagram for the reset and evaluate wavefronts has been obtained through HSPICE simulations using the 0.18 μ TSMC process models and is shown in Fig. 5.

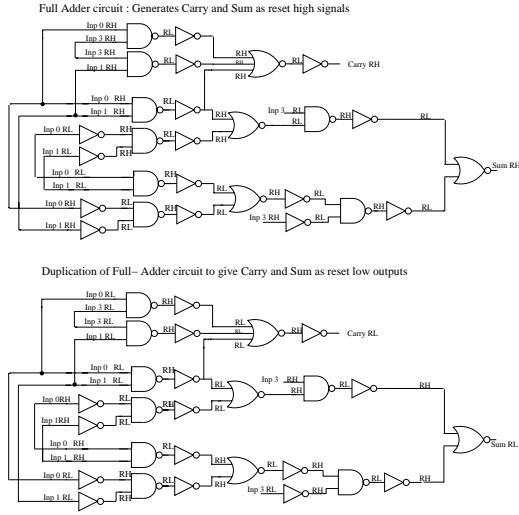


Fig. 4: Full adder with duplication.

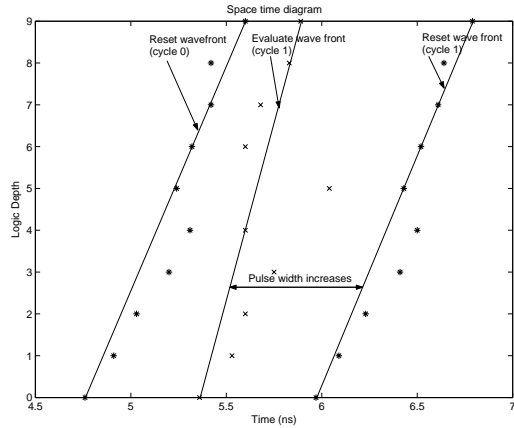


Fig. 5: Space-time diagram for the multiplier.

Ideally, it is desirable to have the reset wavefront of the current cycle slower by a factor of 1.5 than the evaluate wavefront so that the next evaluate wavefront never overlaps with the reset wavefront, assuming that each logic level takes one unit of time to evaluate and 1.5 units of time to reset [1]. In the above space-time diagram, this ratio has been achieved through careful skewing of the transistor widths. The width of the evaluate pulse also grows with increased logic depth, which enables correct latching of the output after the last level of logic. The PS-CMOS design of the 4-bit multiplier has also been found to be 1.4 times faster than a corresponding static implementation. Note that in order to make a fair comparison, the minimum (i.e. unit) width of the transistors for both the PS-CMOS and static designs are kept the same (0.36 μ).

5. 8-BIT PS-CMOS ADDER

A carry look-ahead adder (CLA) block avoids the rippling of carry signals through multiple bit positions, and it is used as a component in many types of adder

architectures. A CLA block makes use of generate and propagate signals as follows:

$$C_{out,k} = G_k + P_k(G_{k-1} + P_{k-1}(\dots + P_1(G_0 + P_0C_{i,0})))$$

where:

k is the number of bits in the computation,
 G_k is the generate signal for the k th bit,
 P_k is the propagate signal for the k th bit,
 $C_{i,0}$ is the carry input and
 $C_{out,k}$ is the carry output for the k th bit.

The sum outputs are obtained by XORing the propagate and carry input signals [3, 4]. The architecture of a 4-bit carry look-ahead block is shown in Fig. 6.

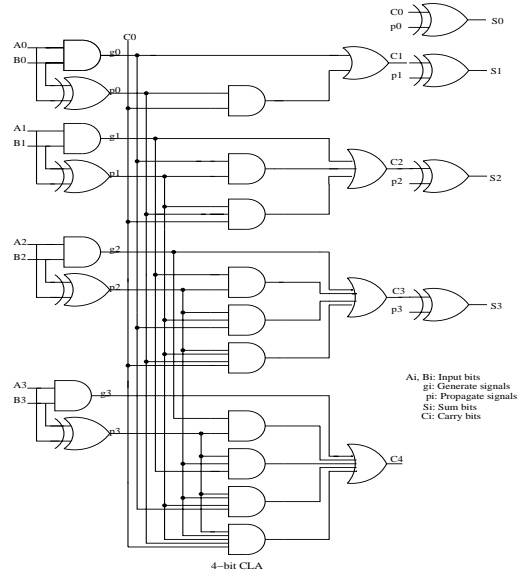


Fig. 6: 4-bit CLA block.

The PS-CMOS design of a 4-bit CLA block is simplified due to the absence of XOR gates in the carry generation path. Thus, with appropriate skewing, the carry signals can be generated with minimal delay. However, the presence of XOR gates in the sum path makes duplication necessary at this stage. The increase in area consumption caused by duplication can be minimized by restricting this process to the inputs of the sum generation stage alone. Thus, the carry signals can be generated in one stage and all the sum signals can be generated simultaneously in the subsequent stage. It can be seen that the delay is primarily due to the carry signals since the sum signals are generated subsequently in parallel.

We have designed and simulated an 8-bit adder which uses three of these 4-bit CLA blocks within an overall carry-select adder architecture. The block diagram of the resulting 8-bit adder is shown in Fig 7.

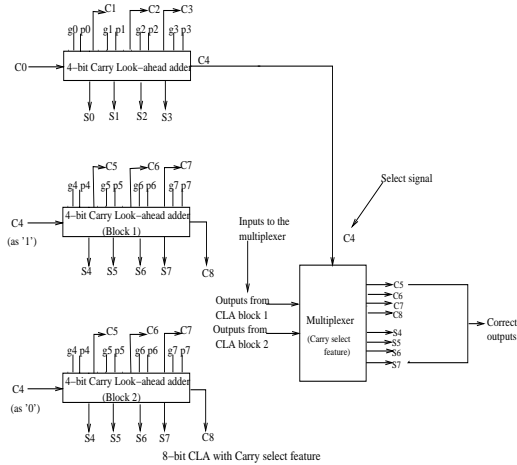


Fig. 7: Architecture of the 8-bit carry-select adder.

The architecture consists of three 4-bit CLA blocks that operate in parallel. For the lower four bits, a single 4-bit CLA block is used to generate the sum and carry signals based on the carry input (C_0). However, for the higher-order bits two 4-bit CLA blocks are used to compute the sum and carry outputs. One of these has a carry input of 0 and the other has a carry input of 1. The carry output (C_4) serves as the select signal for the multiplexer to choose the correct set of outputs from the two CLA blocks (Block 1 and 2). Since all the CLA generators operate in parallel, a rippling delay is avoided. However, in order to provide a matched input signal type to the multiplexer, duplication of the CLAs is necessary.

With careful skewing of the gates, a slope ratio of 1.5 has been achieved between the evaluate and reset wavefronts, as shown in the space-time diagram of Fig. 8. It is also observed that as the logic depth increases, the width of the evaluate wavefront increases, thus enabling correct latching of the output.

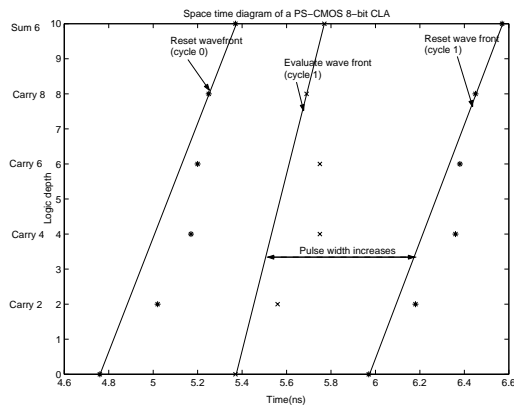


Fig. 8: Space-time diagram for the 8-bit adder.

The speed of this design is observed to be *twice* that of a corresponding static design. The observed delay

between the input and output in the static design is 0.81 ns whereas that between the evaluate edge of the clock and the output in the PS-CMOS design is only 0.40 ns. Note also that some idle time occurs after the completion of the evaluation wavefront. This is unavoidable as the desired ratio of 1.5 would not be achieved if the clock period were reduced any further.

A possible alternative to reduce the extent of duplication would be to implement this design in two stages. The first stage would consist of the three 4-bit CLA blocks and a multiplexer to produce the carry signals. The second level would contain latches and tri-state buffers which serve to duplicate the carry and propagate signals in order to generate the sum outputs. By postponing the sum generation, we would need to duplicate only the select signal C_4 and the inputs to the second level. Thus, the area consumption would be significantly reduced as none of the 4-bit CLAs would have to be duplicated. However, this modification would result in the sum signals being available only after two clock cycles.

6. CONCLUSIONS

This paper introduces a novel method of duplication in PS-CMOS circuits that is extremely useful for arithmetic functions and which leads to circuits having a significant speed improvement compared to static CMOS. We have illustrated the design technique in two representative modules, an array multiplier and a carry-select adder. Simulation results indicate that the circuits operate properly and are significantly faster than corresponding static CMOS designs. It should also be noted that the circuits implemented using this method are most feasible when the number of bits is relatively small. This is because an increase in the number of bits would require careful skewing of a large number of transistor widths in order to achieve the desired speed up. Moreover, the amount of duplication also grows and the design becomes quite large. Although the method has these limitations, the goal of a faster static circuit design can be achieved using duplication in PS-CMOS.

ACKNOWLEDGEMENT

This work was supported in part by an equipment grant from Intel Corporation.

REFERENCES

- [1] Chih-Liang Chen and Gary S. Ditlow, "Pulsed Static CMOS Circuit," U.S. Patent No. 5,495,188, February 27, 1996.
- [2] Jan M. Rabaey, *Digital Integrated Circuits – A Design Perspective*, Prentice Hall, 1996.
- [3] B. H. Parhami, *Computer Arithmetic: Algorithms and Hardware Design*, New York: Oxford University Press, 2000.
- [4] N. H. E. Weste and K. Eshragian, *Principles of CMOS VLSI Design*, 2nd Ed., Addison-Wesley, 1993.