

# EFFICIENT DIGIT-SERIAL FIR FILTERS WITH SKEW-TOLERANT DOMINO

*Sungwook Kim and Gerald E. Sobelman*

Department of Electrical and Computer Engineering  
University of Minnesota  
Minneapolis, MN 55454, USA  
Phone: (612) 625-8041, FAX: (612) 625-4583  
e-mail: sobelman@ece.umn.edu

## ABSTRACT

A novel connection between digit-serial computation and skew-tolerant domino circuit design is exploited to create very efficient implementations of FIR digital filters. In our approach, a digit size of  $N$  bits is mapped onto an  $N$ -phase overlapping clocking scheme in such a way that  $N$  bits are processed in each full clock cycle. In addition, a VHDL-based verification strategy is used to capture the essential time-borrowing behavior of skew-tolerant domino circuits in an accurate and efficient manner. The simulation results show that an 8-tap digit-serial FIR filter constructed with skew-tolerant domino is up to 36% faster than one built using traditional domino circuits.

## 1. INTRODUCTION

Finite impulse response (FIR) filters are used extensively in a wide variety of DSP systems. Bit-serial, bit-parallel, and digit-serial VLSI architectures have all been used to implement these structures. Bit-serial designs are very area-efficient but may be too slow for certain applications. On the other hand, bit-parallel designs may be faster than necessary and occupy a considerable amount of area. Digit-serial systems [1], [2], [3] have been proposed as a flexible, intermediate approach which may avoid these disadvantages in many situations. Several previous designs for digit-serial FIR filters have been presented in the literature [4] [5]. However, most of these designs are based on conventional circuit styles such as static CMOS.

In recent years, domino CMOS circuit design has been used to improve the performance of microprocessors and other high-speed digital systems [6], [7]. However, the traditional domino design methodology [8] based on two-phase clocking suffers from several sources of overhead such as clock skew, intermediate latches and imbalances in the delays of different blocks. Skew-tolerant domino circuits [9] make use of multiple overlapping clock phases in such a way that all of this overhead can be eliminated, and this

leads to a significant performance improvement. In previous work, we have shown that there is a natural mapping between digit-serial multipliers and skew-tolerant domino circuits [10]. This paper extends that work to digit-serial FIR filters composed of skew-tolerant domino multiplier, adder and accumulator blocks. An efficient simulation methodology is used to demonstrate that this design approach can achieve better performance than one based on a traditional domino implementation.

## 2. DIGIT-SERIAL FIR FILTER

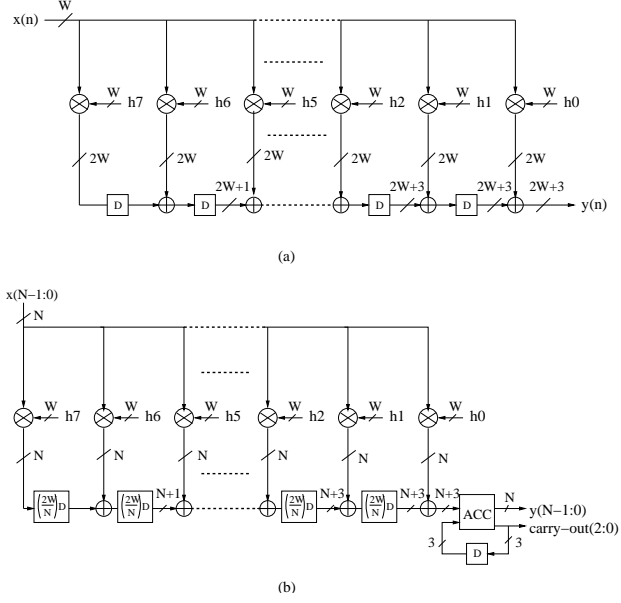
The equation for an  $L$ -tap FIR filter is given by

$$y(n) = \sum_{k=0}^{L-1} h_k \cdot x(n-k), \quad (1)$$

where  $x(n)$  and  $y(n)$  denote the input and output samples, respectively, and where the  $h_k$  are the filter tap coefficients. A typical bit-parallel hardware implementation is given in Fig. 1(a), which shows the inverted-form structure for an 8-tap FIR filter having a word size of  $W$  bits. To maintain full precision along the data path, a word size of  $2W+3$  bits is required at the final output.

In digit-serial computation,  $W$ -bit data words are partitioned into digits of size  $N$  bits. Computational blocks process one digit at a time, starting with the least-significant digit. The digit-serial structure for the same 8-tap FIR filter is illustrated in Fig. 1(b). There are four components: a digit-serial multiplier, a digit-serial adder, a digit-serial accumulator and a delay stage. A total of  $2W/N$  clock cycles are required in order to obtain the full-precision result from each multiplier. Therefore,  $2W/N$  delay stages are used to hold the intermediate results at each tap location. The data path width is increased from  $N$  bits to  $N+3$  bits as the computations proceed through the digit-serial adders so that a full-precision output can be obtained. The digit-serial accumulator produces a digit-size filter output  $y(N-1:0)$  and

the 3-bit quantity carry-out(2:0) on each clock cycle. After  $2W/N$  successive clock cycles, a total of  $(2W/N) \times N = 2W$  bits are produced at the  $y$ -output. During the last clock cycle, the 3 carry-out bits are considered to be the 3 most-significant output bits. Therefore, a  $(2W+3)$ -bit full-precision filter output word is obtained after  $2W/N$  successive clock cycles.



**Fig. 1.** Block diagram for an inverted-form FIR filter. (a) Bit-parallel implementation, with a single delay stage between each adder. (b) Digit-serial implementation having a digit size of  $N$  bits, with  $2W/N$  delay stages between each adder.

### 3. MAPPING ONTO SKEW-TOLERANT DOMINO

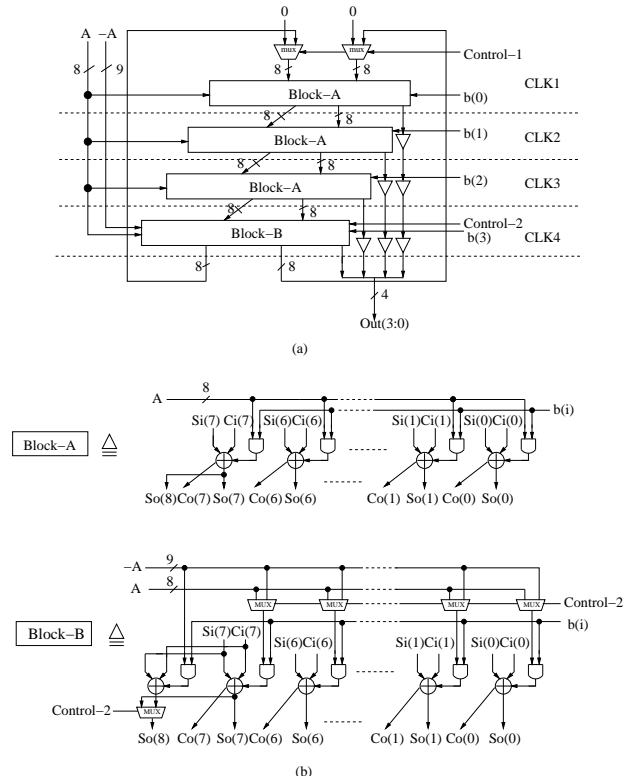
Skew-tolerant domino circuits make use of overlapping clock phases in such a way that all sources of clocking overhead can be eliminated, which results in a significant performance improvement. We have designed skew tolerant domino implementations of the three basic blocks used in the FIR filter. The following sub-sections give examples for a digit-size of  $N = 4$  bits, but other designs having different values of  $N$  may be constructed in an analogous fashion.

#### 3.1. Digit-Serial Multiplier

A signed digit-serial multiplier with a word size of 8 bits and a digit size of 4 bits is implemented using 4 overlapping clock phases, as illustrated in Fig. 2(a). The multiplicand,  $A$ , and its two's complement,  $-A$ , are fed into the structure in bit-parallel form. The multiplier,  $B$ , is partitioned into

4-bit digits  $b(3:0)$  which are applied sequentially, with the least-significant digit first.

As shown in Fig. 2(b), Block-A and Block-B each contain partial product generation and carry-save adder logic. Note that when Control-2 is 0, Block-B performs the same function as Block-A. On the other hand, when Control-2 is 1, the two's complement of the multiplicand is used. This latter mode is activated when the most-significant digit of the multiplier is applied. The low-order two digits of the product are produced at  $Out(3:0)$  in the first two clock cycles. The high-order two digits are obtained by applying zeros at the  $b(3:0)$  inputs during the second two clock cycles. Thus, a complete 16-bit product is obtained in four full clock cycles.

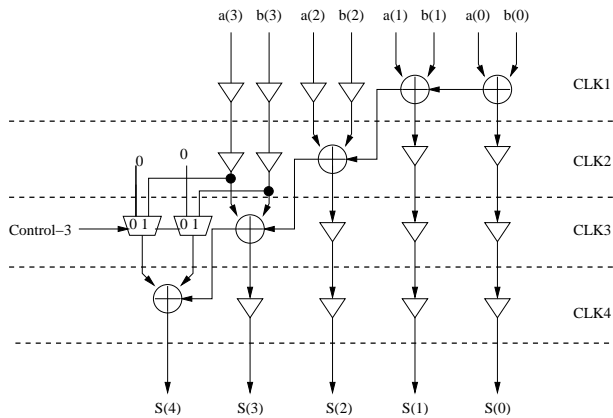


**Fig. 2.** Digit-serial multiplier with a word size of 8 bits and a digit size of 4 bits. (a) Top-level design using 4 overlapping clock phases. (b) Implementations of Block-A and Block-B.

#### 3.2. Digit-Serial Adder

A digit-serial adder with a digit size of 4 bits using 4 overlapping clock phases is shown in Fig. 3. The digit-serial adder is composed of full adders, MUXs and domino buffers. Two digits,  $a(3:0)$  and  $b(3:0)$ , are added to produce the 5-bit output digit  $S(4:0)$  in one full clock cycle. Note that when Control-3 is 0, the digit-serial adder performs an unsigned

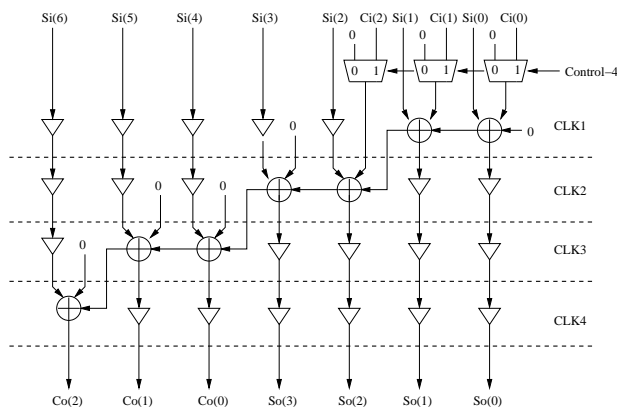
addition. (This is needed because all digits except the most-significant digit are unsigned binary numbers.) On the other hand, when Control-3 is 1, the most significant bit is sign-extended to perform a signed addition.



**Fig. 3.** Digit-serial adder with a digit size of 4 bits using a 4-phase overlapping clocking scheme.

### 3.3. Digit-Serial Accumulator

The digit-serial accumulator of Fig. 4 produces a 4-bit sum-out ( $So(3)-So(0)$ ) and a 3-bit carry-out ( $Co(2)-Co(0)$ ) in 4 overlapping clock phases. The sum-in inputs ( $Si(6)-Si(0)$ ) come from the previous digit-serial adder. The carry-in inputs ( $Ci(2)-Ci(0)$ ) are cleared to zero through MUXs during the first cycle. After that, the carry-outs are fed back to the carry-ins to be added during the next cycle. When the last digit of each input word has been processed, the final 7-bit output is formed as the concatenation of the 3 carry-out bits and the 4 sum-out bits. Thus, after four full clock cycles, a total of  $3 \cdot 4 + 7 = 19$  bits are produced, which comprises the full-precision result.



**Fig. 4.** Digit-serial accumulator with a digit size of 4 bits using 4 overlapping clock phases.

## 4. PERFORMANCE EVALUATION

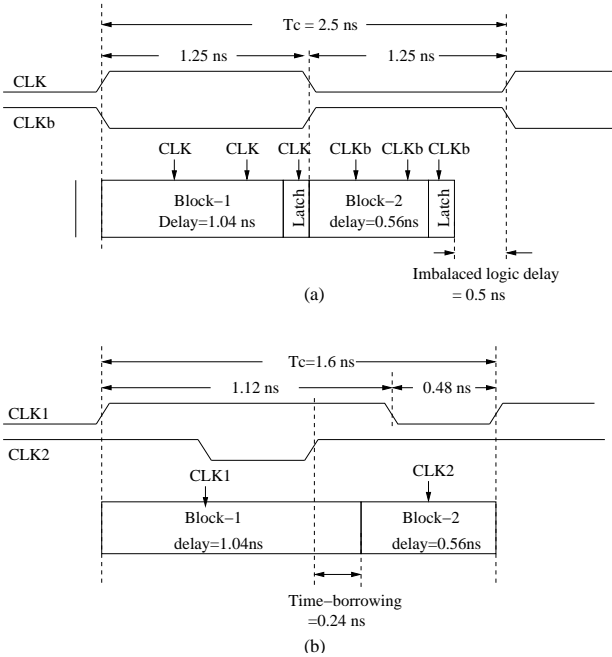
Since the complete 8-tap FIR filter contains a large number of devices, a flat transistor-level circuit simulation would require a very long execution time. Therefore, we applied a much more efficient approach [11]. Behavioral VHDL models were created for each basic logic gate. The delay values in these models were obtained from HSPICE simulations of the underlying domino circuit configurations, including the estimated loading conditions. We used device models for the  $0.25 \mu m$  TSMC CMOS technology that are available from MOSIS, with a supply voltage of 2.5 volts. Once these behavioral models have been created, the full FIR filter can be readily simulated using a VHDL engine. This verification methodology provides a way to capture the essential time borrowing behavior inherent in skew-tolerant domino circuits at a much higher level of abstraction so that large systems can be efficiently and accurately simulated.

For comparison purposes, we created traditional and skew-tolerant domino implementations with a word size of 8 bits and digit sizes of  $N = 2$  and 4 bits. Fig. 5 gives the simulation results for a digit size of 2 bits. The longest logic delay path is in the digit-serial accumulator, which has been partitioned into Block-1 and Block-2. As shown in Fig. 5(a), the traditional two-phase domino design suffers from imbalanced logic and latch delays. On the other hand, Fig. 5(b) shows that time borrowing is taking place in the skew-tolerant design since the computation extends beyond the phase boundary, as indicated by the dashed vertical line. Thus, the clock cycle time is equal to the propagation delay for logic evaluation alone, which results in a speed-up of 36%.

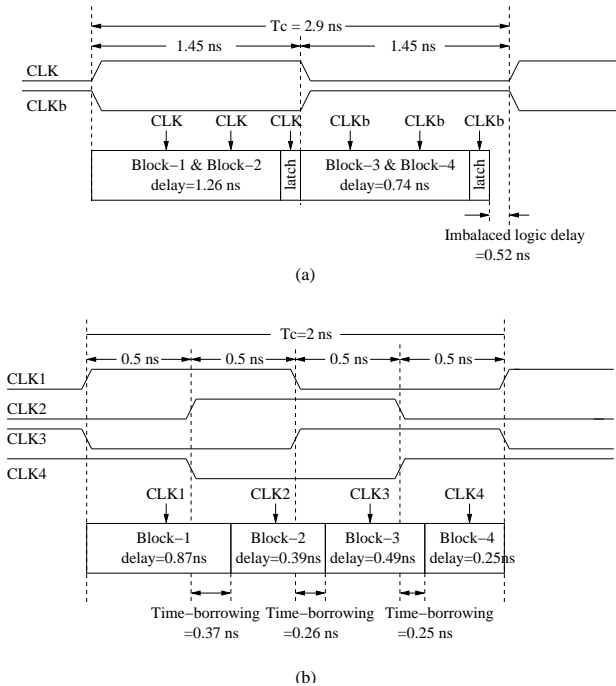
Similarly, Fig. 6 shows the comparative simulation results using a digit size of 4 bits. In this case, the digit-serial accumulator has been partitioned into 4 sections called Block-1 through Block-4. In the skew-tolerant domino design, each of these blocks is allocated to a specific clock phase, as in Fig. 4. For the traditional two-phase domino design, these blocks have been grouped into pairs. In this case, the simulation results show that the skew-tolerant domino circuit is 31% faster than the traditional domino implementation.

## 5. CONCLUSIONS

We have proposed a novel design for a high-performance FIR filter using a digit-serial architecture and skew-tolerant domino circuit techniques. A digit-serial implementation with a digit size of  $N$  bits is mapped in a natural fashion onto a skew-tolerant domino clocking scheme having  $N$  overlapping clock phases. In this way, one operand bit is utilized in each of  $N$  phases, so that an  $N$ -bit digit is processed in each full clock cycle. This leads to simple scheduling



**Fig. 5.** Simulation results for the digit-serial FIR filter using a digit size of 2 bits. (a) Traditional two-phase domino. (b) Skew-tolerant domino with 2 overlapping clock phases.



**Fig. 6.** Simulation result for the digit-serial FIR filter using a digit size of 4 bits. (a) Traditional two-phase domino. (b) Skew-tolerant domino with 4 overlapping clock phases.

and straightforward implementations of digit-serial FIR filters. The simulations have been done in a manner that efficiently captures the essential time-borrowing behavior in large skew-tolerant domino circuits. The results confirm that the skew-tolerant domino circuits provide a significant improvement in throughput of up to 36% over a traditional two-phase domino implementation.

## 6. REFERENCES

- [1] Y. -N. Chang, J. H Satyanarayana, and K. K. Parhi, "Design and Implementation of Low-power Digit-serial Multipliers," *IEEE International Conference on Computer Design*, pp.186 -195, 1997.
- [2] H. H Lee and G. E. Sobelman, "Digit-Serial Reconfigurable FPGA Logic Block Architecture," *IEEE Workshop on Signal Processing Systems*, pp.469-478, 1998.
- [3] A. S Ashur, M. K. Ibrahim and A. Aggoun, "Systolic Digit-Serial Multiplier," *IEE Proc.: Circuits, Devices and Systems*, Vol. 143, pp.14-20, Feb. 1996.
- [4] H. H Lee and G. E. Sobelman, "FPGA-Based FIR Filters Using Digit-Serial Arithmetic," in *Proc. of IEEE international ASIC Conference*, pp. 225-228, Sept. 1997.
- [5] J. Valls, M. M. Peiro, T. Sansaloni, E. Boemo, "Design and FPGA implementation of Digit-Serial FIR filters," *IEEE International Conference on Electronics, Circuits and Systems*, Vol. 2, pp. 191 -194, 1998.
- [6] D. H. Allen et al, "Custom Circuit Design as a Driver of Microprocessor Performance," *IBM J. Research and Development*, Vol. 44, No. 6, pp. 799-822, November, 2000.
- [7] C. Lemonds, "A 500 MHz, One Volt 16 by 16 Bit Multiplier for DSP Cores," *VLSI Signal Processing, IX*, pp. 481-484, 1996.
- [8] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison Wesley, pp.344-354, 1993.
- [9] D. Harris and M. A. Horowitz, "Skew-Tolerant Domino Circuits," *IEEE Journal of Solid-State Circuits*, Vol. 32, pp. 1702-1711, Nov. 1997.
- [10] S. Kim and G. E. Sobelman, "Digit-Serial Multiplier Design Using Skew-Tolerant Domino Circuits," *IEEE International ASIC/SOC Conference*, 2001.
- [11] S. Kim and G. E. Sobelman, "Digit-Serial Modular Multiplication Using Skew-Tolerant Domino CMOS," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 1173-1176, 2001.