

# High-throughput and Low-power DSP Using Clocked-CMOS Circuitry

Manjit Borah

Robert Michael Owens

Mary Jane Irwin

Department of Computer Science and Engineering  
Pennsylvania State University  
University Park, PA 16802

## Abstract

We argue that the clocked-CMOS ( $C^2$ MOS) circuit family provides a very high throughput and low power alternative to other existing circuit techniques for the fast developing market of portable electronics. By the virtue of self latching gates allowing very fine-grained pipelining, avoidance of precharge and short circuit power consumption, the  $C^2$ MOS circuit offers very good power-delay efficiency. We support our claims through the design of an 8-bit unsigned binary multiplier with pipelining at the gate level which can produce 500 million multiplications per second consuming only 0.8 W power using 1.0 micron technology and 3.3V power supply. By comparison the fastest previously existing pipelined multiplier has a throughput rate of 400 million multiplications per second consuming 0.8 W power at 0.8 micron technology, 5V, using wave-pipelining.

## 1 Introduction

Low power consumption and high throughput are the two important requirements for portable and real time electronic equipment. Pipelining has been used successfully to attain high throughput in digital signal processing (DSP) systems. When the throughput demand is not very high, static CMOS is considered the most power efficient circuit family. However, for very high throughput applications, static CMOS tends to lose its power-delay efficiency. This is due to the fact that static CMOS requires extra pipelined latches which add extra delay, limiting the throughput rate and increasing the power consumption. If we consider the deepest level of pipelining, then each pipeline block consists of

only simple two or three input gates (e.g., AND/NAND, OR/NOR, XOR/XNOR), avoiding long chains of transistors in series.

The clock rate of a pipelined circuit is determined by the slowest pipeline block. Non-clocked logic families, like static CMOS and pass-transistor logic require separate latches adding extra delay to the clock cycle time and increasing the power consumption. Clocked logic families offer better power-delay characteristics than non-clocked logic families for deep pipelined circuits.

In this paper we show that  $C^2$ MOS circuits, due to their low power consumption and ability to apply pipelining at much finer level, can be used to build very high throughput circuits with low power consumption. We present the design of an 8-bit pipelined multiplier for unsigned numbers using  $C^2$ MOS to demonstrate the advantages of  $C^2$ MOS logic for the domain. The multiplier, implemented in 1.0 micron technology, can produce throughput at the rate of 500 million multiplications per second with only 44 nS initial latency consuming a mere 0.8 W power including clock driving circuitry. The fastest existing pipelined multiplier is wave-pipelined using normal process complementary pass transistor logic (NPCPL) and has a throughput rate of 400 million multiplications per second, consuming 0.8 W power with 0.8 micron technology[7]. We compare our design with several other pipelined multiplier designs, including true single-phase logic based, CPL-based, NMOS based and quasi n-p domino logic based designs exploiting pipelining at various granularities.

We briefly describe the  $C^2$ MOS circuit family in section two. Section three details the structure of the multiplier and its basic building blocks. Section four analyzes its speed and power consumption, validating the claims with SPICE simulation results. Conclusions and future research goals are presented in section five.

## 2 The $C^2$ MOS logic family

The  $C^2$ MOS logic family was first proposed in 1973 for calculator circuits as a low-power and smaller area logic alternative[8]. However, at that time dynamic

---

\*this work was partially supported by NSF grant no. CDA-8914587

logic, especially logic requiring complicated clocking was not considered very practical. With the advance in VLSI techniques the generation of high-speed clocks with controlled skew have become possible. Using very fine-grained pipelining (at the NAND/NOR gate level) C<sup>2</sup>MOS circuits can be built to produce very high throughput.

The basic principle of the C<sup>2</sup>MOS circuit is simple: separate the p and the n blocks of the conventional static CMOS gate by two clock transistors T1 and T2 (Figure 1(a)). The p-type block is enabled by T1 while the

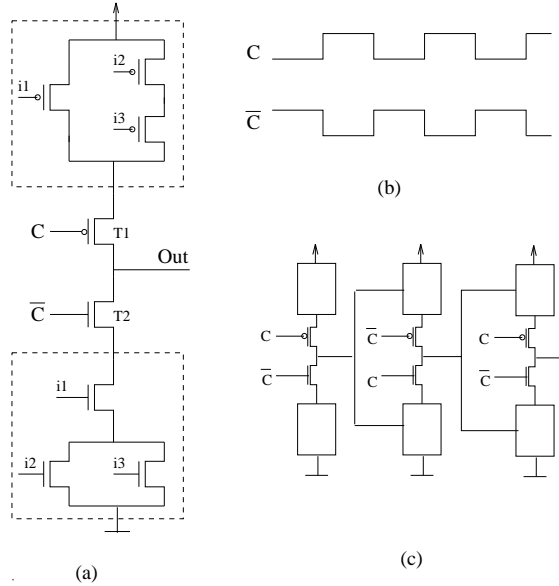


Figure 1: (a)C<sup>2</sup>MOS gate (b)complementary clocks (c)consecutive gates compute in alternate phases

n-type block is enabled by T2. The two clock transistors are driven by two complementary clocks  $C$  and  $\bar{C}$ . The typical waveforms of  $C$  and  $\bar{C}$  are given in Figure 1(b). When  $C$  is low and  $\bar{C}$  is high, the gate evaluates the logic function. When  $C$  is high and  $\bar{C}$  is low, the output of the gate is disconnected from the logic blocks and the gate ‘holds’ the output state.

During the period when the output of a gate is on ‘hold’, the computation of the next gate can take place. To allow this, consecutive gates are driven by complementary clocks (Figure 1(c)). As a result, waves of computation can propagate in a pipelined manner through the gates. Thus, acting as self-latched compute blocks, the C<sup>2</sup>MOS circuit provides gate level pipelining at no extra cost.

The advantages of C<sup>2</sup>MOS logic are many:

- The inputs to a gate are stable (on hold) before the gate starts computing. Therefore, short circuit current is eliminated – only one of the two logic blocks conduct at any given time.
- The self latching property of the C<sup>2</sup>MOS circuit eliminates the need for any extra registers in the

design.

- The delay of a pipeline stage in C<sup>2</sup>MOS circuit is determined by only a single gate, which may involve only three transistors in series, enables a C<sup>2</sup>MOS circuit to run at very high speed.
- Transistor/gate sizing in C<sup>2</sup>MOS circuit is simplified due to the fact that it depends only on the fan-out load of the gate.
- C<sup>2</sup>MOS does not require precharge, and it does not suffer from the charge-sharing problem typical of a precharge circuit.
- C<sup>2</sup>MOS logic does not require complementary inputs (unlike CPL or dual-rail logic) and it can generate output signals with transitions in both directions (in contrast to precharge gates), making it compatible with conventional CMOS circuits.
- The output signals of C<sup>2</sup>MOS has complete swing of the voltage range, providing good noise immunity.

There are, however, a couple of issues that need to be addressed while using C<sup>2</sup>MOS circuitry:

- C<sup>2</sup>MOS uses complementary clocks, requiring two clock signals to be routed to each gate, resulting in an increase in the global routing. Careful clock routing with balanced paths and a fat-tree structure for clock drivers is used in this work to limit the clock skew. SPICE simulations show that the circuit is capable of tolerating a skew of 0.3nS at 500MHz, 3.3V.
- A C<sup>2</sup>MOS circuit may suffer from *capacitive coupling*. This happens when a signal line that crosses over a charged signal changes its voltage causing a small capacitive discharge on the charged signal. The voltage drop is usually very small. Moreover, in a pipelined design the placement of the gates are usually such that gates from the same pipeline stage are placed in the same *slice*. Therefore the lines that cross over between *slices* belong to the same pipeline stage which ‘hold’ and ‘evaluate’ at the same time, hence capacitive coupling is eliminated.
- Like static CMOS, C<sup>2</sup>MOS, gates are inverting (e.g., NAND, NOR, INV); to obtain non-inverted signal after an odd number of pipeline stages (or inverted signal after even stages), static CMOS inverters are used. Proper logic decomposition and mapping can be used to reduce such cases to a large extent.
- Like any other dynamic logic families, C<sup>2</sup>MOS does not allow power-down by disabling the clock.

## 2.1 Power-delay comparisons

As mentioned earlier, when the throughput demand for the circuit is very high, then pipelining should be applied at the single gate level. Due to the requirement of extra latches, non-clocked circuit families show significant extra overhead in terms of delay and power consumption. To evaluate the power-delay tradeoff offered by different logic families for very deep pipelined circuits, we consider a pipelined two-input NOR/OR gate implementation using static CMOS, pass-transistor logic and C<sup>2</sup>MOS. We also used two types of latches, the

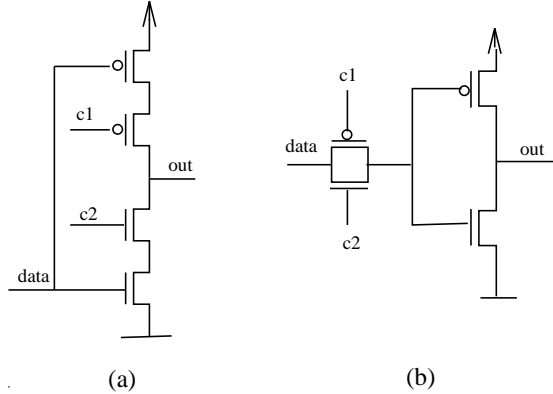


Figure 2: (a) C<sup>2</sup>MOS and (b) transmission-gate inverting latches

C<sup>2</sup>MOS latch and the transmission-gate latch (figure 2).

The circuit consists of two input latches driving the two inputs to the actual logic gate which is also latched at the output (figure 3). The inputs to the input latches

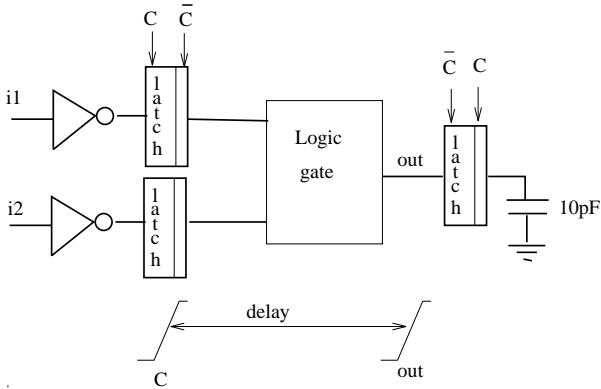


Figure 3: The sub-circuit used for comparison

are derived from static CMOS inverters and the output of the latched gate drives a capacitive load of 10pF. For our discussion we assume that all the transistors are uniformly sized to  $4\lambda$ . We performed SPICE simulations to determine the maximum possible clock speed and the power consumption for each implementation. We measured the delay between the clock edge at the input latches and the signal at the output of the logic block before the output latch. We also measured the average power consumption when the circuit is simulated for

all possible combination of input vectors. Since all the blocks require the same number of clock transistors and the clock routing is also similar, we expect the power-consumption of the clock circuitry to be the same in all the cases.

Table 1: Power-delay characteristics (2-input NOR/OR)

gate type	latch type	delay (nS)	power ( $10^{-5}W$ )	power $\times$ delay
C <sup>2</sup> MOS		1.32	5.209	6.876
CMOS	C <sup>2</sup> MOS	1.66	6.056	10.053
CMOS	trans-gate	1.87	8.332	15.581
pass-gate	C <sup>2</sup> MOS	2.2	7.487	16.471

Table 1 shows delay, power and power-delay product of the basic pipeline block for the circuit implementations. The fully C<sup>2</sup>MOS circuit shows much superior clock-to-output delay compared to the other types of circuits. Moreover, the power consumption of the C<sup>2</sup>MOS circuit is much smaller than the other implementations which results in a significantly smaller power-delay product for C<sup>2</sup>MOS.

## 2.2 Comparisons with other techniques

True single phase logic proposed in [3] has a simpler clocking structure and the advantage of a single phase clock. Even though the global clock routing in true single phase logic is simpler than C<sup>2</sup>MOS, the number of clock transistors increase compared to C<sup>2</sup>MOS (figure 4). The actual logic also becomes more complicated resulting in a slower gate. SPICE simulations show that the clock-to-output delay for a true-single-phase 2-input NOR gate is 1.7 nS compared to 1.32 nS for C<sup>2</sup>MOS while the power consumption without including the clock-driving circuitry is  $8.185 \times 10^{-5}W$  as compared to  $5.209 \times 10^{-5}W$  for C<sup>2</sup>MOS when simulated for a circuit similar to figure 3. Moreover, the

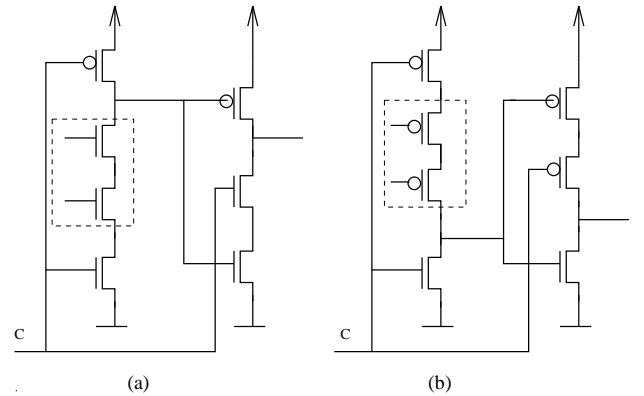


Figure 4: True-single-phase gates: (a) n-block (b) p-block

clock drivers in a true single phase logic are expected to

provide much sharper rise and fall times which in turn makes the clock generation more difficult[5].

Wave pipelining[9] has been used with complementary pass-transistor logic (CPL) [10] to attain high throughput with low power consumptions[1]. But as the authors in [1] point out, the design of a CPL based wave-pipelined circuit is mostly dependent on balancing the delay along different paths and setting the transistor width ratios to achieve proper logic threshold. All these properties vary with process, temperature and other external factors. Therefore the performance of such a design may degrade significantly in a real environment. Since the proper functioning of the circuit without data overrun is dependent on the cumulative path delays in the whole design, the design of large circuits using wave-pipelining is very difficult. While implementing a large system the wave speeds of different modules need to be matched, which is another challenging requirement.

### 3 The 8-bit multiplier design

Multipliers are essential parts of a digital signal processing circuit and it is the most time critical component. Therefore pipelined multipliers are highly desirable which is evident from the abundant literature on highly pipelined multiplier designs [4, 7, 1, 6, 2]. Earlier pipelined multipliers[2, 6, 4] were based on using one full-adder stage as one pipelined unit. More recent pipelined multipliers, exploit pipelining at the half-adder or XOR gate level [7, 1].

The obvious speed advantage of finer level of pipelining is, however, accompanied by the increased number of pipelined registers and expansive clock circuitry which result in more power consumption. But the power-delay product comparison has always been favorable to the finer pipelined designs. Here we have exploited the advantage of C<sup>2</sup>MOS logic gates to design circuit with pipelining at the single NAND/NOR gate level, resulting in clock speed much faster than the existing designs, while the lower power-delay characteristic of C<sup>2</sup>MOS has made it possible to maintain fairly low power consumption compared to all existing designs.

#### 3.1 The structure of the multiplier

Array architectures are commonly used for pipelined multiplier design due to their regular structure and easy interconnects. We designed a pipelined array multiplier using carry-save adder arrays, as discussed in [2]. An array of full-adder cells are used to accumulate and propagate the partial sum and carry. Each row of full-adders also adds a new row of partial products to the partial sum and carry. Therefore skewing registers are used to delay the inputs for the partial product generation logic such that they are presented to the proper row at the proper time. For the final adder stage, which accumulates the partial sum and carry values into the

final product, we use a triangular vector merge array of half-adder as described in [2] for the same reasons, i.e., reduction of pipeline latency, reduction in the extra de-skewing registers and regular structure.

#### 3.2 The structure of the basic modules

Our goal is to design the multiplier for maximum throughput. The basic pipeline block of our design is limited to 2-input NAND/NOR gates and inverters. Thus, the stage delay of our design can be as small as the delay incurred in a 2-input NOR gate. The circuit diagram of the half-adder and the full-adder are shown in Figure 5. Notice that the number of stages in a full adder is four, requiring two clock cycles for it to compute. The isolated small circles are static CMOS in-

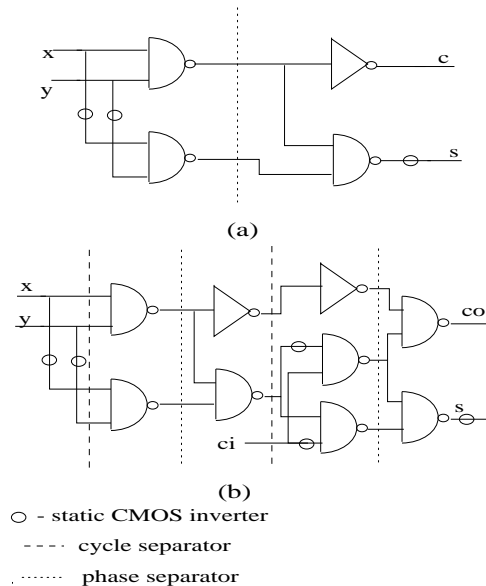


Figure 5: Circuit diagram of (a) half-adder and (b) full-adder

verters used to produce inverted signals without adding extra pipeline stages.

The basic building block of the multiplier consists of a full-adder and an AND gate to compute the partial product to be added to the partial sum and carry signals passed down from the previous row. The partial product computation is overlapped with the first cycle of the full-adder computation and the new partial product is used as the ‘ci’ input in the second cycle of the full-adder. A register is also included to pass on the multiplier bits to the next row for partial product computation.

#### 3.3 Floor-plan and layout

The floor-plan of the 8-bit×8-bit pipelined multiplier is shown in Figure 6. The data flow is strictly vertical. The multiplicand bits percolate down the carry-save adder rows and participate in the partial product generation with the corresponding multiplier bits in the given row. The row of AND gates at the top computes the first two rows of the partial product array using

the least significant two bits of the multiplier ( $y_0, y_1$ ) with the corresponding multiplicand bits. These partial products are used in the next row (half adders) to compute the first stage of carry-save addition. The most significant partial product for each row is computed by the ‘and’ block in the left most column. The output of

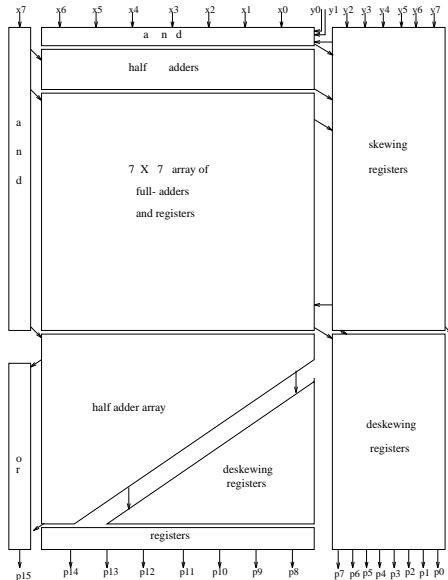


Figure 6: The floor-plan of the 8-bit  $\times$  8-bit multiplier. The last row of full-adders are input to the triangular half-adder array for computing the vector merge addition producing the final most significant product bits. The total latency of the 8-bit multiplier pipeline is 22 cycles.

## 4 Results and analysis

The layout of the CMOS  $8 \times 8$  multiplier is shown in Figure 7. The multiplier contains 6156 transistors and has a silicon area of  $0.90 \text{ mm} \times 0.89 \text{ mm}$  using 1.0 micron technology. This initial layout was not optimized for area and the area can be reduced significantly with better layout techniques.

The clock driver and distribution circuitry accounts for about 70% of the total power consumption of the multiplier. Therefore it is important to design the clock drivers properly to minimize the power consumption. We designed a tree based CMOS clock driver circuit to distribute the load of the clock tree into three driver stages. The third stage of the driver tree consists of sixteen inverting buffers which are driven by four inverting buffers in the second stage and they are in turn driven by a single inverting buffer. We believe that more reduction in the power consumption can be obtained by further optimization in the clock drivers.

We simulated the multiplier core together with the clock driver tree using SPICE with 1.0 micron technology parameters at 3.3V supply voltage at room temperature. The SPICE waveforms for the output of the

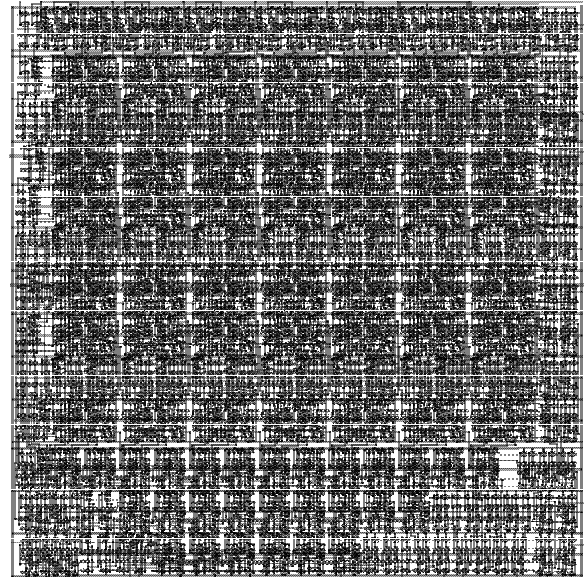


Figure 7: Layout of the 8bit  $\times$  8bit multiplier

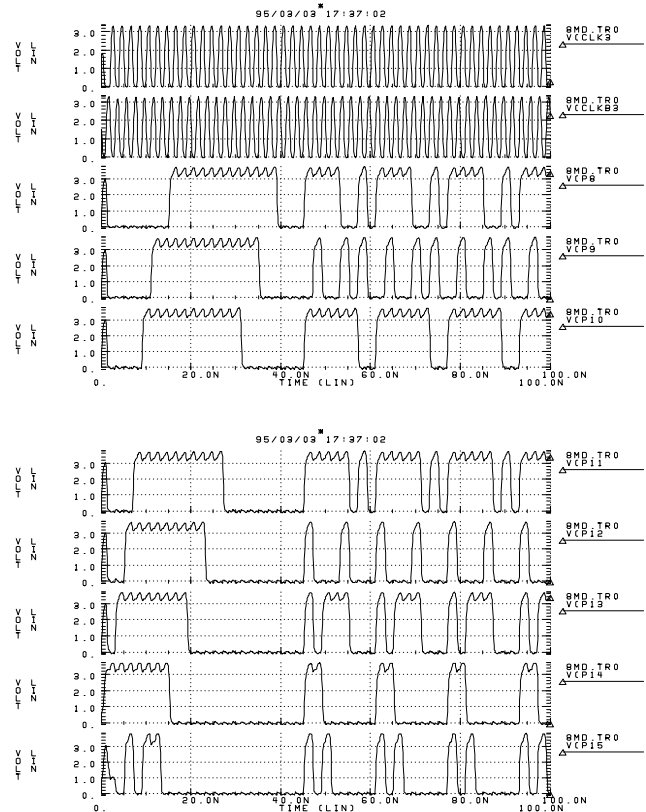


Figure 8: SPICE output of the 8bit  $\times$  8bit multiplier with clock drivers at 500MHz, 3.3V

clock drivers and the eight most significant product bits for the 8-bit×8-bit multiplier running at 500 MHz are shown in figure 8. The average power consumption of the 8-bit×8-bit multiplier including the clock drivers is found to be 0.8 W at 500 MHz.

Table 2: Description of the existing multipliers

name	precision	Comments
Noll	8 × 8	1.0μm nMOS, 3V
Hatamian	8 × 8	2.5μm CMOS, 5V
Lu	12 × 12	1.0μm CMOS,5V (quasi-domino) mult-accum.
Somasekhar	8 × 8	1.6μm CMOS,5V(true 1-φ)
Ghosh	8 × 8	0.8 μm CMOS,5V(NPCPL)
This work	8 × 8	1.0 μm CMOS,3.3V(C <sup>2</sup> MOS)

Table 2 lists some of the existing pipelined multiplier designs with respect to their precision and process technology. A comparison of the performance and power consumption of the existing multipliers with the current work is presented in Table 3. The last column in Table 3 shows the power-delay product for the multiplier designs. Our C<sup>2</sup>MOS multiplier, simulated at 1.0 micron CMOS, runs faster than any of the existing designs with a low power consumption. Observe that the C<sup>2</sup>MOS circuit has a much superior power-delay product, which is highly desirable for portable DSP applications.

Table 3: Comparison of different pipelined multipliers

name	clk-rate (MHz)	power (Watts)	latency (nSec)	pow×delay mW/MHz
Noll	330	1.5	54.5	4.54
Hatamian	70	0.25	228.57	3.57
Lu	200	1.3	65.0	6.5
Somasekhar	230	0.54	52.17	2.35
Ghosh	400	0.8	37.5	2.0
This work	500	0.8	44.0	1.6

## 5 Conclusion

Through the example of an 8-bit pipelined multiplier for unsigned numbers, we have shown that C<sup>2</sup>MOS is an energy efficient logic family for very high-throughput applications. The multiplier presented in this paper is the fastest existing pipelined multiplier with a throughput of 500 million multiplications per second and power consumption of 0.8 Watt with 1.0 micron HP technology and 3.3V supply voltage. The multiplier circuit is being fabricated as a tiny-chip using 2.0 micron process for initial testing. The authors are currently investigating the system level design of low-power and high perfor-

mance DSP systems for portable electronic equipment using C<sup>2</sup>MOS.

## References

- [1] D. Ghosh and S. K. Nandy. A 400MHz Wave-Pipelined 8 × 8-bit Multiplier in CMOS Technology. In *Proceedings of ICCD*, pages 198–201, 1993.
- [2] Mehdi Hatamian and Glenn L. Cash. A 70-MHz 8-bit×8-bit Parallel Pipelined Multiplier in 2.5μm CMOS. *IEEE Journal of Solid-State Circuits*, sc-21(4):505–513, August 1986.
- [3] Y. Ji-Ren, I. Karlsson, and C. Svensson. A True Single-Phase-Clock Dynamic CMOS Circuit Technique. *IEEE Journal of Solid-State Circuits*, sc-22(5):899–901, October 1987.
- [4] Fang Lu and Henry Samueli. A 200-MHz CMOS Pipelined Multiplier-Accumulator Using a Quasi-Domino Dynamic Full-Adder Cell Design. *IEEE Journal of Solid-State Circuits*, 28(2):123–132, February 1993.
- [5] E. D. Man and M. Schobinger. Power Dissipation in the Clock System of Highly Pipelined ULSI CMOS Circuits. In *Proc. of International Workshop on Low Power Design*, pages 133–138, 1994.
- [6] T. G. Noll, D. Schmit-Landsiedel, H. Klar, and G. Enders. A Pipelined 330-MHz Multiplier. *IEEE Journal of Solid-State Circuits*, sc-21(3):411–416, June 1986.
- [7] D. Somasekhar and V. Visvanathan. A 230-MHz Half-Bit Level Pipelined Multiplier Using True Single-Phase Clocking. *IEEE Transactions on VLSI*, 1(4):415–422, December 1993.
- [8] Y. Suzuki, K. Odagawa, and T. Abe. Clocked CMOS Calculator Circuitry. *IEEE Journal of Solid-State Circuits*, sc-8(6):462–469, December 1973.
- [9] D. C. Wong, G. De Micheli, and M. Flynn. Designing High-Performance Digital Circuits Using Wave-Pipelining. *IEEE Transactions on Computer-Aided Design*, 12(1):25–46, Jan 1993.
- [10] K. Yano, T. Yamanaka, T. Nashida, M. Saito, K. Shimohigashi, and Akihiro Shimizu. A 3.8-ns CMOS 16 × 16-b Multiplier Using Complementary Pass-Transistor Logic. *IEEE Journal of Solid-State Circuits*, 25(2):388–395, April 1990.