International Symposium on Circuits and Systems, pp. 117-120, May 2003, Bangkok, Thailand

5 GHZ PIPELINED MULTIPLIER AND MAC IN 0.18μM COMPLEMENTARY STATIC CMOS

Jos B. Sulistyo and Dong Sam Ha

VTVT (Virginia Tech VLSI for Telecommunications) Lab
Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061, USA
Web: www.ee.vt.edu/ha

ABSTRACT

Wave pipelining improves the throughput of a circuit by exploiting the delays of combinational elements, rather than register clocks, for synchronization. Our proposed approach, called HyPipe, combines conventional register-based pipelining with wave pipelining and aims to take advantage of both pipelining methods [5]. In this paper, we applied HyPipe to develop 4-bit signed multipliers and 4-bit MACs targeting for high speed or low-power applications. The circuits were implemented in fully complementary CMOS in TSMC 0.18 μm technology. SPICE simulation results indicate that the two circuits can operate at around 5 GHz under the supply voltage of 1.8 V or at around 1 GHz under 0.8 V with about 25 times less power dissipation.

1. INTRODUCTION

Pipelining is widely employed to circuits to increase throughput. The throughput of a pipelined circuit is determined by the worst delay of a pipeline stage, and register clocks are responsible for synchronization. Another method for improving throughput is *wave pipelining* [1], which exploits the delays of a combinational circuit to process multiple data simultaneously. To distinguish the two pipelining schemes, we call the former conventional method *register pipelining* in this paper.

Wang et al. used register pipelining to design an 8×8 multiplier [6]. Their multiplier reaches the speed of 630 MHz in a 0.6 μm technology through bit-level pipelining, which inserts a register stage after every adder cell. Ghosh and Nandy attained about the same speed using a 0.8 μm technology using wave pipelining [2]. Ghosh and Nandy suggested the use of differential logic based on pass transistors, which leads to reasonably balanced delays for logic 1 and logic 0. In our earlier work, we proposed a new design approach, called HyPipe, which combines the two pipelining methods, register pipelining and wave pipelining [5]. HyPipe aims to take the advantages of both pipelining methods to improve the throughput of a circuit.

In this paper, we investigated high-speed/low-power multipliers and MACs (Multiplier ACcumulators) based on the HyPipe approach. We improved the speed of our earlier 1-bit adders in [5], and then constructed a signed multiplier and a MAC unit using the 1-bit adders. Spice simulation indicates that the resulting pipelined multiplier and the MAC unit reach the speed of 5 GHz in the TSMC 0.18 μm CMOS technology under the supply voltage of 1.8 V.

This paper is organized as follows. Section 2 provides background for wave pipelining and reviews our HyPipe approach briefly. Section 3 discusses the design of proposed multipliers and MAC units. Section 4 contains simulation results and observations, and Section 5 concludes the paper.

2. WAVE PIPELINING AND THE HYPIPE ARCHITECTURE

In this section, we briefly describe terms and conditions for wave pipelined circuits. For details, an excellent tutorial on wave pipelining is available in [1].

The general structure of wave pipelining is shown in Figure 1. It consists of input and output registers and a combinational logic block (CLB). The following terms are defined for wave pipelined circuits.

W_{MIN}, W_{MAX}	minimum and maximum propagation delays from the input of the input register (including the setup time) to the input of the output register
T_{CK}	clock period of a wave pipelined circuit

It is important to note that a minimum (maximum) delay in the above should be obtained considering all possible pairs of input transitions, since a delay often depends on the sequence of input patterns applied.

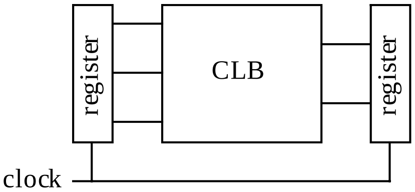


Figure 1. Structure of a Wave-Pipelined Circuit

The minimum allowable clock period for a non-wave-pipelined circuit is simply W_{MAX}. In contrast, the clock period for a wave pipelined circuit is limited by the difference between the slowest and the fastest delay as shown below.

$$T_{CK} \geq W_{MAX} - W_{MIN} \tag{1}$$

The above relationship in (1) determines the maximal operating frequency at which a wave pipelined circuit can operate.

The *wave number N* of a wave-pipelined circuit is defined as the number of clock cycles needed for a signal to propagate through the combinational logic block before latched by the output register. A wave number N represents the degree of wave pipelining, and it bounds the clock period T_{CK} as shown in (2) [1],[5].

$$W_{MAX}/N \leq T_{CK} \leq W_{MIN}/(N\text{-}1) \qquad (2)$$

In addition, the clock period is also bounded from below by rise/fall times of gate input/output signals, gate delays, and the delays of the input and output registers [5]. Those additional constraints result in the operable frequency of a wave pipelined circuit being much lower than the theoretical maximum frequency given in (1).

Wave pipelining offers potential for high speed. However, this high speed is attainable only if a good delay balance is attained. Further, if the logic depth is large, delay imbalances may accumulate over signal paths of a large combinational logic block and reduce the actual operable frequency significantly.

HyPipe partitions a combinational logic block into smaller blocks and to apply wave pipelining to each smaller block, while the entire combinational block operates in a register pipelined manner as shown in Figure 2 [5].

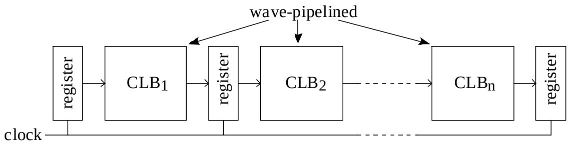


Figure 2. Block Diagram of Our HyPipe Approach

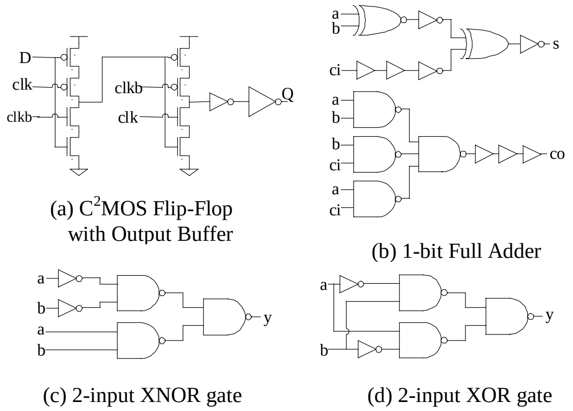
The finer granularity for HyPipe decreases the minimum and maximum delays of a subblock, W_{MIN} and W_{MAX} of each pipeline stage, resulting in a significant increase of the maximum operable clock speed. HyPipe is particularly attractive for a combinational logic block that is composed of a few identical building blocks such as ripple-carry adders and array multipliers, since it requires well crafted design of a few building blocks. In the next section, we present design of pipelined signed multipliers and MACs based on the HyPipe approach.

3. PROPOSED MULTIPLIER AND MAC ARCHITECTURE

We briefly describe full adders and flip-flops first, which are based on the HyPipe architecture. We constructed a 4-bit signed multiplier and a 4-bit signed MAC unit using the full adders and the flip-flops.

3.1 Flip-flops, Full Adders, and Deskewing Registers

A flip-flop and a full adder circuits used for constructing multipliers and MAC units are shown in Figure 3. The design goals of the circuits are good delay balance and short rise/fall times, both of which are necessary to attain high operating speed for wave pipelining.



(a) C^{2}MOS Flip-Flop with Output Buffer

(b) 1-bit Full Adder

(c) 2-input XNOR gate

(d) 2-input XOR gate

Figure 3. Circuit diagram of the building blocks

C^{2}MOS flip-flops are used due to its better immunity to the clock overlap than simple dynamic flip-flops such as the one used in our previous design in [5]. A tapered buffer, i.e., a cascade of two inverters, is added at the output of the flip-flop, as it should drive a large number of gate loads of the following adder, four gate loads for inputs a and b and three gate loads for carry-in ci. The fanout of all other gates except the large inverter at a flip-flop output is limited to one to reduce rise/fall times. Those modifications increase the speed of the full adder (including flip-flops at the input and output of the adder) from the wave number N=2 in our previous design to N=3.

The XOR and XNOR gates of full adders are built using NAND gates and inverters instead of complex gates as shown in Figure 3 (c) and (d), since complex gates tend to have long output rise/fall times.

For the purpose of minimizing delay variations along different paths, we use the same full adder in Figure 3 (b) to implement minterm generators (which performs an AND operation necessary for multipliers), as well as inverting and noninverting combinational delay elements. A delay element used in our design, called Δ delay element, is shown in Figure 4. A delay element delays the signal by N clock cycles, where N is the wave number. It is possible to use a shift register instead of the delay element for a fixed N; however, it causes inflexible operating speed. The cost of the use of full adders as minterm generators and delay elements is increased circuit complexity.

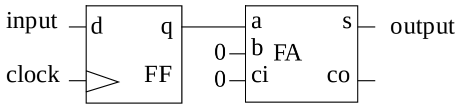


Figure 4. Δ Delay element

3.2 Multiplier

We present design of a 4×4 signed multiplier, which is bit-level pipelined based on the HyPipe architecture. The logic function evaluated by the proposed architecture is described as follows.

Consider two n-bit 2's complement numbers, A and B:

$A = -2^{n-1}a_{n-1} + 2_{n-2}a_{n-2} + \ldots + 2^{1}a_{1} + 2^{0}a_{0} = -S_{A} + L_{A}$

$B = -2^{n-1}b_{n-1} + 2_{n-2}b_{n-2} + \ldots + 2^1b_1 + 2^0b_0 = -S_B + L_B$

where S is the signed part of a number, e.g., $S_A = 2^{n-1}a_{n-1}$, while L is the unsigned part of the number. Then,

$$A\times B = (S_A\times S_B) + (L_A\times L_B) - (S_A\times L_B) - (S_B\times L_A) \qquad (3)$$

Note that $(S_A \times S_B) = 2^{2n-2}a_{n-1}b_{n-1}$ and $-X = /X + 1$ for a 2's-complement number X. Hence, with $X = (S_A\times L_B) + (S_B\times L_A)$, (3) becomes:

$$A\times B = [a_{n-1}b_{n-1} << (2n-2)] + (L_A\times L_B) + (/\,[(S_A\times L_B) + (S_B\times L_A)] + 1) \qquad (4)$$

where << is the left shift operator and / is the bitwise complement operator. As $/[X + Y] = /X + /Y +1$ for 2's complement numbers X and Y, (4) can be expressed as:

$$A\times B = [a_{n-1}b_{n-1} << (2n-2)] + (L_A\times L_B) + [/(S_A\times L_B) + /(S_B\times L_A) + 1] + 1 \qquad (5)$$

The two terms, $S_A\times L_B$ and $S_B\times L_A$, have (n-2) significant bits and two leading 0s each. So it is important to take complement of the two leading 0s as well as the significant bits. It should be noted that Baugh-Wooley's 2's complement multiplication algorithm essentially relies on the above expression [4].

The architecture given in Figure 5 evaluates (5). It composed of two ripple-carry adders, one array multiplier, one preskewing register, and three delay elements. The 3×3 multiplier computes $L_A\times L_B$, and the 3-bit adder computes the third term in (5). The first and the last terms are taken care of by the 5-bit adder. One MSB input of the 5-bit adder is set 1, as the resultant MSB of the third term in (5) is 1.

Calculation of $/(S_A\times L_B)$ does not require a multiplier: since S_A has only one possibly nonzero bit, NAND gates with shifted bit positions would suffice. To balance path delays, a NAND gate is constructed in two stages, an AND stage and an inverting stage, constructed from the same full adders discussed earlier.

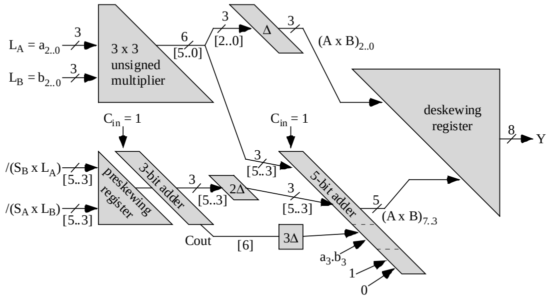


Figure 5. Block diagram of a 4-bit signed multiplier

The 3×3 unsigned multiplier is implemented as an array multiplier based on the HyPipe architecture. Every full adder of the multiplier has an input and an output register to perform register pipelining, and the full adder itself operates under wave pipelining with wave number *N*=3. The two adders are implemented as ripple carry adders, and each full adder operates under wave pipelining with wave number *N*=3.

Now, let's discuss about the timing of signals. The left side of a block in Figure 5 depicts the arrival time of input signals and the right side the available time of the output signals. A delay element Δ delays 3 clock cycles as *N*=3 in our design. A preskewing register, which is composed of delay elements, is added to skew arrival time of input signals to the adder. So that a bit x_i arrives at the adder Δ time (i.e., 3 clock cycles) later than the bit x_{i-1}. The deskew register is added at the output to make all the outputs available at the same time.

The proposed multiplier has 10 pipeline stages, consisting of 7 stages for the unsigned 3×3 multipler (5 stages for the array itself, and 2 for the minterm generators) and 3 stages for the 5-bit adder. Therefore, the latency of the multiplier is 31 clock cycles (=10×*N*+1) under the wave number *N*=3.

3.3 MAC

A MAC unit considered in this paper is shown in Figure 6. It implements a function Y = AB + CD, with 4-bit inputs and an 8-bit untruncated output. It consists of two signed multipliers, an adder and a deskew register. One notable aspect in the design is that deskew registers of the two multipliers are detached from the multipliers and placed at the output to save the hardware. The MAC unit has 11 pipeline stages with the latency of 34 clock cycles.

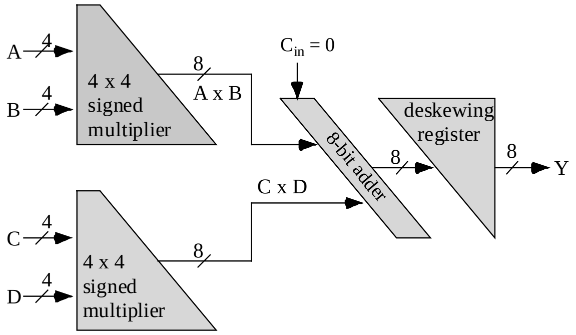


Figure 6. Block Diagram of the MAC

4. SIMULATION RESULTS

We generated netlists of the full adder, the signed multiplier and the MAC unit manually and performed SPICE simulation of the circuits for a wave number *N*=3. Hence, routing delays are not considered in our simulations.

Prior to simulating the multiplier and MACs, a pair of the flip-flops and the full adder were simulated first to find the minimum and maximum propagation delays. The same type of the flip-flop is connected to the full adder output as a load. A full adder has three inputs and hence 8 possible input combinations. Since an input combination can be followed by possibly 7 different input combinations, a total of 56 different input transitions were simulated. Table I shows the simulation results for four different supply voltages, 1.8, 1.5, 1.2 and 0.8 V, as well as permitted T_{CK} and operable frequency ranges predicted using (2).

From Table I, it is apparent that the register and the full adder pair could operate in a reasonably broad range of clock frequency for each supply voltage.

Table I. Performance of a Full Adder and a Flip-Flop Pair

	1.8 V	1.5 V	1.2 V	0.8 V
$W_{MIN/MAX}$	438-533	560-679	808-975	2176-2717
T_{CK}	177-219	226-280	325-404	906-1088
Freq. Range (GHz)	4.6-5.7	3.6-4.4	2.5-3.1	0.9-1.1

Next, we selected one middle operating frequency of the full adder for each supply voltage from Table 1 and simulated the multiplier and the MAC unit for those frequencies. The goal of our simulation is to verify correct operation for each voltage and to estimate its performance. A load of four inverters is placed at each output pin. The simulation results are shown in Table II. In the table, EDP denotes energy-delay product, which is the product of energy consumed during one clock period and the clock period. The latencies were computed as the product of the latency in clock cycles of a circuit and the clock period.

Table II. Performance of the Multiplier and the MAC

V_{DD}	1.8 V	1.5 V	1.2 V	0.8 V
Clock, GHz	5.00	4.00	2.86	1.05
Multiplier				
Power, mW	74.48	39.76	17.83	2.82
EDP, $\times 10^{-21}$ J.s	2.98	2.48	2.18	2.55
Latency, ns	6.20	7.75	10.85	29.45
MAC				
Power, mW	132.02	70.54	31.67	5.01
EDP, $\times 10^{-21}$ J.s	5.28	4.41	3.88	4.52
Latency, ns	6.80	8.50	11.90	32.30

Most of all, our simulation indicated the multiplier and the MAC unit performed correctly for the chosen clock frequencies under the wave number N=3. So the multiplier and the MAC unit indeed achieve the high operating clock frequency of 5 GHz under 1.8 V. When the supply voltage is reduced from 1.8 V to 0.8 V, the power dissipation for the two circuits reduces by about 25 times. So our circuits may be suitable for low-power applications as well as high speed applications.

It is difficult to compare the performance of our circuits with others directly due to the use of different technologies and different performance goals such as high throughput versus short latency. Horowitz et al. suggested the use of FO4 delay, which is the delay of inverter with four identical loads as a performance metric [3]. The FO4 delay of the processing technology used for our circuits is estimated as 82 ps under the supply voltage of 1.8 V, and hence T_{CK}=2.44×FO4 delays for our multiplier and the MAC unit running at 5 GHz. We believe that FO4 delay performance for our multiplier and the MAC unit is quite remarkable.

5. CONCLUSION

We investigated design of multipliers and MAC circuits based on our earlier pipelining method called HyPipe. Due to a stringent requirement on delay balancing for the high speed operation, we employed only one kind of combinational building block, full adder, in our design. Our SPICE simulation results show that our multipliers and MAC circuits can operate at 5 GHz in the TSMC 0.18 μm technology under the supply voltage of 1.8 V. The speed is about three times higher than that would be attained using conventional pipelining only.

If the circuit is laid out and fabricated, the attainable operating frequency would be lower due to both wire loads and other parasitics. However, we believe that the circuits can still operate with the wave number 3, since the wire delays are relatively small due to the regular structure of ripple carry adders and array multiplies. It is open to future research.

6. REFERENCES

[1] W. Burleson, M. Ciesielski, W. Liu, and F. Klass, "Wave Pipelining: A Tutorial and Research Survey". *IEEE Transactions on VLSI Systems*, vol. 6, pp. 464-474, September 1998.

[2] D. Ghosh and S. K. Nandy, "Design and Realization of High-Performance Wave-Pipelined 8b×8b Multiplier in CMOS Technology". *IEEE Transactions on VLSI Systems*, vol. 3, pp. 36-48, March 1995.

[3] M. Horowitz, R. Ho, and K. Mai, "Wires: A User's Guide", http://mos.stanford.edu/papers/rh_srcmarco_99.pdf, Stanford University, 1999.

[4] B. Parhami, *Computer Arithmetic – Algorithms and Hardware Designs*, 1st ed. New York, NY: Oxford University Press, 2000, sec. 11.3.

[5] J. B. Sulistyo and D. S. Ha, "HyPipe: A New Approach for High Speed Design". *Proceedings of the 2002 IEEE ASIC/SOC Conference*, pp. 203-207, September 2002.

[6] J.–S. Wang, P.–H. Yang, and D. Sheng, "Design of a 3-V 300-MHz Low-Power 8-b × 8-b Pipelined Multiplier Using Pulse-Triggered TSPC Flip-Flops". *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 583-592, April 2000.