World Applied Sciences Journal 2 (4): 341-347, 2007
ISSN 1818-4952

A New Low Power 32×32-bit Multiplier

[1]Pouya Asadi and [2]Keivan Navi

[1]Islamic Azad University Science and Research Branch, Tehran, Iran
[2]Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran

Abstract: Multipliers are one of the most important building blocks in processors. This paper describes a low-power 32×32-bit parallel multiplier, designed and fabricated using a 0.13 µm double-metal double-poly CMOS process. In order to achieve low-power operation, the multiplier was designed utilizing mainly pass-transistor logic circuits, without significantly compromising the speed performance of the overall circuit implementation. New circuit implementations for the partial-product generator and the partial-product addition circuitry have been proposed, simulated and fabricated. An efficient radix-2 recoding logic generates the partial products. The multiplier supports 32×32-bit integer multiplication of both signed and unsigned operands. The multiplication time is 3.4 ns at a 1.3-V power supply. Our multiplication algorithm showed 7.4 percent speed improvement, 11 percent power savings and 9.5 percent reduction in transistor count when compared to the conventional multiplication algorithms.

Key words: Adder . booth algorithm . CMOS . compressor . pass-transistor logic

INTRODUCTION

Fast integer multipliers are a key topic in the VLSI design of high-speed microprocessors. Multiplication is one of the basic arithmetic operations. In fact 8.72% of all instructions in a typical scientific program are multiplies [1]. In addition, multiplication is a long latency operation. In typical processes, multiplication takes between two and eight cycles [2]. Consequently, having high-speed multipliers is critical for the performance of processors. Processor designers have recognized this and have devoted considerable silicon area for the design of multipliers [3]. Recent advances in integrated circuit fabrication technology have resulted in both smaller feature sizes and increased die areas. Together, these factors have provided the processor designer the ability to fully implement high-speed floating-point multipliers in silicon.

Most advanced digital systems today incorporate a parallel multiplication unit to carry out high-speed mathematical operations. In many situations, the multiplier lies directly in the critical-path, resulting in an extremely high demand on its speed. In the past, considerable efforts were put into designing multipliers with higher speed and throughput, which resulted in fast multipliers which can operate with delay time as low as 4.1 ns [4]. However, with the increasing importance of the power issue due to the portability and reliability concerns of electronic devices [5], recent work has started to look into circuit design techniques that will lower the power dissipation of multipliers [6].

This paper describes the design and fabrication of a 32×32-bit parallel multiplier, based on a 0.13 µm CMOS process, for low-power applications. Pass-transistor (PT) logic is chosen to implement most of the logic functions within our multiplier. Emerging as an attractive replacement for the conventional static CMOS logic, especially in the design of arithmetic macros PT logic requires fewer devices to implement basic logic functions in an arithmetic operation, such as the XOR function. This translates into lower input gate capacitance and power dissipation as compared to conventional static CMOS [7]. In the PT circuit implementations reported so far [8], transmission on-gate (TG) design techniques which provide full voltage swings were widely adopted. In this paper, we present several circuits that fully exploit the inherent nonfull-swing (NFS) nature of PT logic. These circuits were used as basic building blocks within our multiplier to achieve low-power operation.

MATERIALS AND METHODS

To date, the most widely adopted technique for partial-product generation in large multipliers (16-bit and above) is the Modified Booth's Algorithm (MBA). The main attraction of MBA is that instead of generating n partial-products for an n-bit multiplication,

Corresponding Author: Dr. P. Asadi, Islamic Azad University Science and Research branch, Tehran, Iran

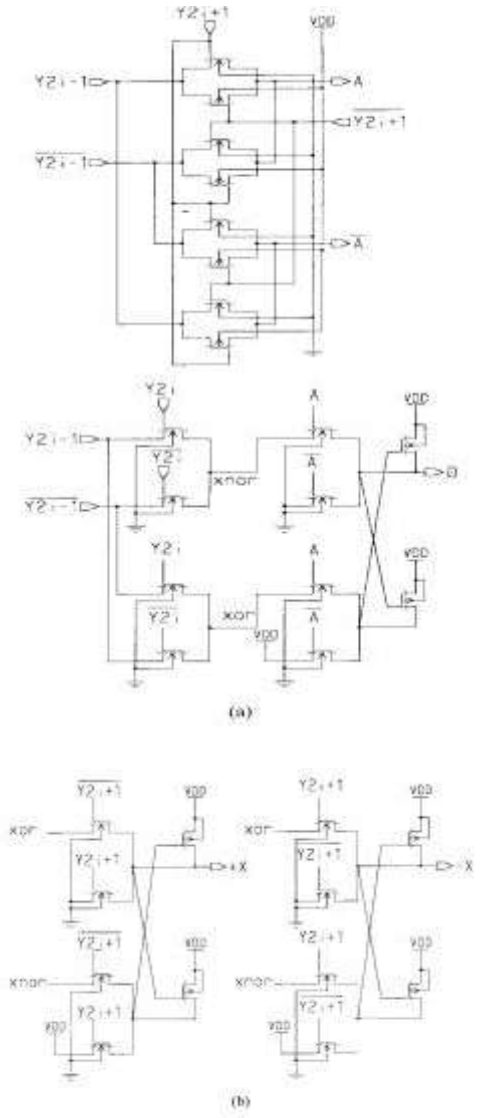

Fig. 1: Proposed circuit implementations for control-bits (a) "0" and (b) "+X" and "-X"

it only generates half of that. According to MBA, a signed binary number in its two-complement form can be partitioned into overlapping groups of three bits. By coding each of these groups, an n-bit signed binary number can be represented as a sum of n/2 signed digits. As each signed digit takes the possible values of zero, ±1 and ±2, the required partial-products are all power-of-two multiples of the multiplicand (X), which are readily available.

The standard PPG circuit implementation requires five control bits, each representing a "0", "+X", "-X", "+2X" or "-2X" operation. The truth table for the control bits are shown in Table 1. When implemented in full CMOS, the encoders only exhibit moderate performance [9].

Table 1: Coding of Y in a standard PPG implementaion

3-b group of Y			Control bits				
Y_{2i+1}	Y_{2i}	Y_{2i-1}	0	+X	-X	+2X	-2X
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	0	1	0	0	0
0	1	1	0	0	1	0	0
1	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0
1	1	0	0	0	0	1	0
1	1	1	1	0	0	0	0

Table 2: Improvement of proposed encoder over CMOS and CPL encoders

				Improvement over	
	CMOS	CPL	Proposed	CMOS (%)	CPL (%)
Delay (ps)	63.00	59.00	50.00	21	15
Power (mW)	0.05	0.90	0.05	53	49
Power×delay (pJ)	0.47	0.41	0.27	43	34
Transistor count	48.00	122.00	50.00	-	59

To improve its performance, complementary PT logic (CPL) family cells have been used in [9]. Although significant improvement in power dissipation (30%) has been reported, the CPL encoder requires 122 transistors to implement, a 150% increase compared to the CMOS encoder (48 transistors) and provides only 6% improvement in speed. We present a PT Booth's encoder (Fig. 1) which offers better performance over both the CMOS and CPL implementations in terms of power, speed and transistor count. From Table 1, it is obvious that the control bit for "0" is high when Y_{2i-1}, Y_{2i} and Y_{2i+1} are the same. Therefore, it can be expressed as:

$$"0" = \overline{(Y_{2i-1} \oplus Y_{2i})}.\overline{(Y_{2i-1} \oplus Y_{2i+1})} \quad (1)$$

The control bit for "+X" is high when Y_{2i-1} and Y_{2i} are different, provided Y_{2i+1} is low. The same is true for "-X", except that Y_{2i+1} must now be high. The expressions for these control bits are:

$$"+X" = (Y_{2i-1} \oplus Y_{2i}).\overline{Y_{2i+1}} \quad (2)$$

$$"-X" = (Y_{2i-1} \oplus Y_{2i}).Y_{2i+1} \quad (3)$$

It is clear from equations 1 to 3 that an XOR operation between Y_{2i-1} and Y_{2i} should be performed to generate all three control bits.

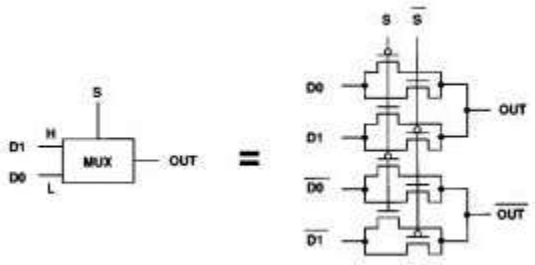

Fig. 2: Pass transistor multiplexer circuit

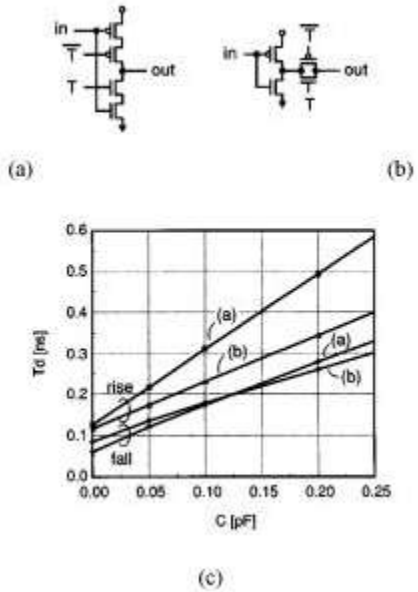

Fig. 3: Comparison of CMOS and pass-transistor circuits. (a) CMOS tristate inverter. (b) Pass-transistor tristate inverter. (c) Comparison of delay time

A PT XOR-XNOR pair carries out this operation and the results (XOR and XNOR) are fed simultaneously in three PT AND-NAND pairs to generate the respective control bits. The control bits for "+2X" and "-2X" are generated using conventional CPL logic style and therefore will not be discussed. The proposed circuit was compared with the CMOS and CPL circuits and the results are shown in Table 2. The proposed encoder outperforms the CMOS implementation by 21% in speed and over 50% in power dissipation, with approximately the same transistor count. When compared to the CPL encoder, our circuit is faster by 15% and achieves about 50% improvement in power and transistor count.

Pass transistor multiplexer: The pass-transistor multiplexer used in the 4-2 compressor is shown in Fig. 2. When the control signal S is low, data D_0 is selected and when the control signal S is high, data D_1

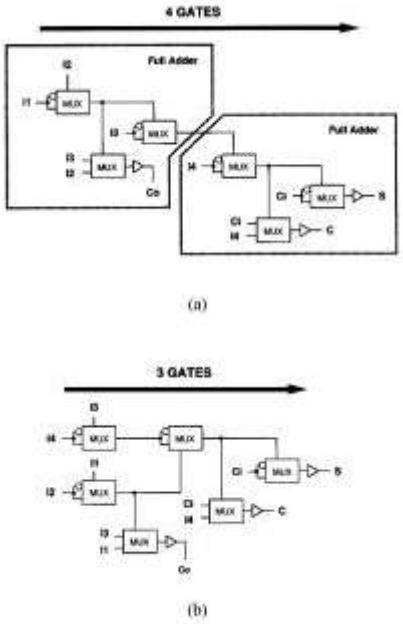

Fig. 4: 4-2 compressor circuits using pass-transistor multiplexers. (a) Full-adder-based construction. (b) Proposed construction

is selected. The output is used as the control signal input for the next-stage multiplexer. Thus, the multiplexer has both positive and negative output. It can reduce the propagation delay by eliminating an inverter.

Several pass-transistor logic circuits have been proposed to improve the performance of CMOS circuits. The NMOS pass-transistor logic circuits [10] is one example. It has been shown to result in high speed due to its low input capacitance and high logic functionality. However, particularly in reduced supply voltage designs, it is important to take into account the problems of noise margins and speed degradation. These are caused by mismatches between the input signal levels and the logic threshold voltage of the CMOS gates, which fluctuates with process variations. To avoid these problems, the multiplexer in this design consists of both NMOS and PMOS pass transistors.

The delay time of the pass-transistor multiplexer is shorter than that of a CMOS gate, because of the pass-transistor based design where both the NMOS and PMOS are turned on. The CMOS tristate inverter and pass-transistor tristate inverter are shown in Fig. 3 (a) and (b), which are used in CMOS and pass-transistor multiplexers, respectively. The number of transistors in both circuits is the same and both have equal input capacitance. A simulated comparison is shown in Fig. 3(c), showing the dependence of the delay time from "in" to "out" on the output load capacitance. The low driving source impedance attained by using the

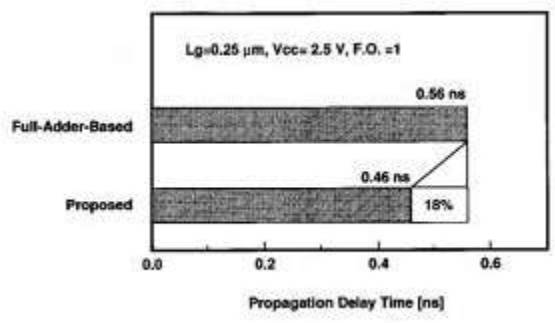

Fig. 5: Simulated comparison of 4-2 compressor circuits

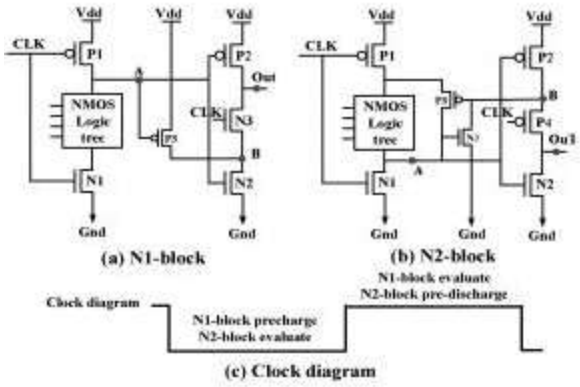

Fig. 6: ANL logic

pass transistor makes the delay time of the pass-transistor shorter than that of a CMOS gate.

4-2 compressor circuit: The 4-2 compressor circuits using pass-transistor multiplexers are shown in Fig. 4. The signal lines in the figure represent positive and negative signals. The inputs of the multiplexers are either two different signal, or one signal and logical invert. All outputs (S, C and C_0) have buffers to enhance driving ability. The 42 compressor circuits add four partial products (I1-I4) and generate a sum signal (S) and two carry signals (C and C_0).

Since the pass-transistor multiplexer circuit shown in Fig. 2 has high logic functionality, a full-adder circuit is constructed from three pass-transistor multiplexers. The 42 compressor is constructed from two full adders, such that there are four critical path gate stages, as shown in Fig. 4(a). This circuit is faster than conventional CMOS circuits due to the use of pass-transistor multiplexers.

For further speed improvement, we developed a new 4-2 compressor. Though the number of multiplexers is the same, the number of critical-path gate stages in this circuit is reduced to three by exploiting parallelism, as shown in Fig. 4(b). In this new configuration, the carry-out (C_0) does not depend on the carry-in (C_i), so, the advantage of the 4-2 compressor is maintained with this new configuration. The simulated delay comparison for these 4-2 compressor circuits is shown in Fig. 5. The propose circuit reduces the propagation delay time by 18% from that of a full-adder-based circuit.

A 32-bit carry lookahead adder: In pipelined systems low-speed pMOS logic blocks are used [10]. For speed improvement, all-N logic (ANL) was introduced to use only high-speed nMOS logic in all stages. All-N transistor (ANT) was developed by using a feedback transistor pair to improve the performance of ANL. For further speed improvement with reduces power consumption, we propose dual-path all-N logic (DPANL). NORA uses two-phase clock signals instead of four-phase clock signals and avoids the race problem caused by clock skews with constrained logic composition. True Single Phase Clock (TSPC) uses only a single-phase clock without inversion. It does not suffer from the clock skew problems and thus can operate at high clock frequency.

Figure 6 shows a circuit diagram of the CMOS dynamic circuit ANL. It removes the drawback of TSPC logic by using an nMOS logic tree in N2-block. To overcome the voltage drop problem in the nMOS logic tree, a positive feedback pMOS P3 in N2-block is used to pull up the evaluation node. pMOS P3 in N1-block and nMOS N3 in N2-block are used to solve the charge sharing problem between the point OUT and the point B. When the clock slew rate is high enough, pMOS P3 in N1-block and nMOS N3 in N2-block can be omitted [11]. A schematic diagram of ANT logic is shown in Fig. 7. It improves the performance using the feedback transistor pair, pMOS P3 and nMOS N3. In evaluation phase, if the nMOS logic tree is evaluated, after the voltage of the evaluation node A drops to below (V_{dd}-V_{th}), pMOS P3 turns on. Then it pulls up point B and turns on nMOS N3. nMOS N3 in turn pulls down evaluation node A and accelerates the evaluation. However, the speedup using the feedback transistor pair is not significant when the number of serial nMOS transistors in the logic tree is small.

Circuit diagram and operating principle: The performance of N1-block is affected by the rise time of the output point since two processes are involved. First, the evaluation node A is pulled down through the current path in the nMOS logic tree. Then pMOS P2 turns on and the output point gets pulled up. The capacitance at the evaluation node A significantly affects the performance. For example, in Fig. 7(a), the

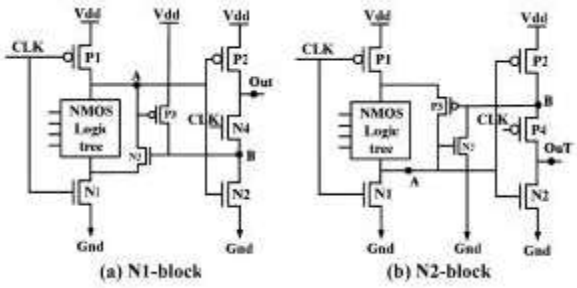

Fig. 7: ANT logic

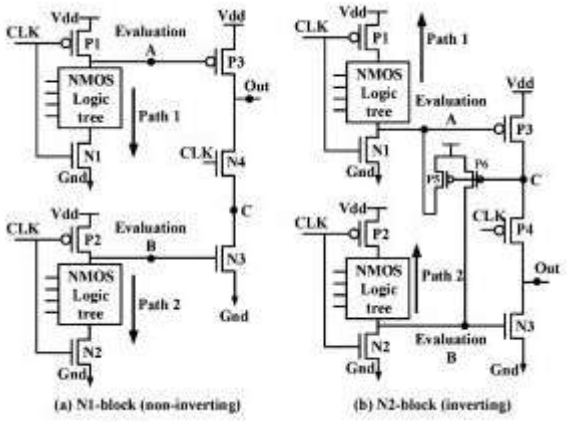

Fig. 8: Circuit diagram of DPANL

gate capacitances of pMOS P2, P3, nMOS N2 and the drain capacitances of nMOS N3, pMOS p1 and the nMOS transistors at the top of the nMOS logic tree are connected to the evaluation node A. To further enhance the performance, we need to reduce the capacitance at the evaluation node.

We have developed DPANL to achieve this goal [9]. N1-block of DPANL is shown in Fig. 8(a). The nMOS logic trees in Path 1 and Path 2 are identical except that Path 1 is made faster than Path 2, since Path 1 influences the rise time of the output. The sizes of the transistors in Path 1 and Path 2 should guarantee that the short circuit current through pMOS P3, nMOS N4 and N3 does not affect the performance. The capacitance at the evaluation node A consists of the gate capacitance of pMOS P3, the drain capacitance of pMOS p1 and the nMOS transistors at the top of the nMOS logic tree. It is much smaller than the corresponding capacitance in ANT and thus helps achieve higher performance.

Power consumption is also less in DPANL. The total width of the two nMOS logic trees in DPANL can be made the same as or even less than that of ANT. In ANT, in order to charge and discharge the large capacitance at the evaluation node A, the sizes of pMOS p1 and nMOS N1 must be large. Also, in order to discharge the capacitance introduced by the feedback transistor pair at point B, nMOS N4 and N2 need to be large. In DPANL, since evaluation nodes A and B have small capacitances, pMOS P1 and P2, nMOS N1 and N2 can be small; nMOS N4 and N3 can also be small since there is no feedback transistor pair attached to point C. So the total channel width of transistors in DPANL can be smaller than that in the ANT. The same principle applies to N2-block. The circuit diagram of N2-block is shown in Fig. 8(b).

Operating principles of the DPANL: When the clock is low, N1-block of DPANL begins its precharge phase. The clocked pMOS P1 and P2 are turned on and the evaluation node A and B are precharged to high. The clocked foot transistors nMOS N1 and N2 are turned off, allowing no current through Path 1 and Path 2. Since the evaluation node A is precharged to high, pMOS P3 is turned off. nMOS N4 is turned off by the clock. So the output point keeps its previous state in the capacitance at that point.

When the clock is high, N1-block begins its evaluation phase. If the nMOS logic tree is not evaluated, the evaluation nodes A and B stay high. pMOS P3 is off, nMOS N4 and N3 are on, the output is pulled down. If the nMOS logic tree is evaluated, the evaluation nodes A and B are pulled down through Path 1 and Path 2, respectively. nMOS N3 is turned off. pMOS P3 is turned on and the output is pulled up.

The operating principles of N2-block are similar to those of N1-block. One thing to note is that when the nMOS logic tree is evaluated, the evaluation nodes can not reach full V_{dd} because of the threshold voltage drop in nMOS transistors. The presence of the feedback transistors pMOS P5 and P6 is to pull up the evaluation nodes to full V_{dd}.

Minimal race problem in DPANL: Race exists in TSPC, ANL, ANT and DPANL, namely, output glitches caused by a race between the discharge of the evaluation node in the logic block and the discharge of the output node by the latch block. Let us take the ANT circuit in Fig. 7(a) as an example. Assume the output was high during precharge phase. If the nMOS logic tree is evaluated in the evaluation phase, the output will still be high. But at the beginning of the evaluation phase, node A and CLK are both high, the output will be discharged through nMOS N4 and N2. After the evaluation node A is discharged, nMOS N2 is turned off and pMOS p2 is turned on, the output is pulled up again, thus forming the large glitch. The large output glitch consumes additional dynamic power.

In order to minimize the race problem, we need to speedup the discharge to the evaluation node A and slow down the discharge of the output. As we have discussed before, the capacitance at the evaluation node of DPANL is much smaller than that of ANT, so discharge of the evaluation node of DPANL is much faster. To slow down the discharge of the output, we can do transistor sizing for the latch block so that on the basis of equal rise time and fall time of output, the discharge path nMOS N4 and N2 are chosen as weak as possible.

The Kogge-Stone graph is generally used for tree structure carry lookahead adders. It has a regular structure and the maximum fan out at each cell for each pipeline stage is 2, which leads to high performance. However, it requires many long interconnects, causing much area and thus much power consumption. S. Knowles introduced a new family of adder structures that offer some tradeoff between performance and power. One of the structures is very suitable for pipeline systems because the maximum fan out at each cell for each pipeline stage is three. It requires less wiring than the Kogg-Stone graph. Thus, we adopted this adder structure for low-power adder design.

Simulation: The multiplier was analyzed using HSPICE on extracted layout under conditions of 1.3-volt supply voltage and 100 degree temperature.

To measure its worst case multiplication time, input test patterns are applied to trigger its critical-path, which includes a Booth's encoder, a control-line buffer, a partial-product selector, four 4-2 compressors and a half adder. The worst case (rise) delay is measured to be 3.4 ns. The average power dissipation of the test chip, inclusive of the multiplier core, input/output pads, output multiplexers and testing circuitry, with no probes at the outputs, is 0.8 mW. The multiplication time and power dissipation of the fabricated device are measured for the supply range of 0.8-V to 1.4-V and the results are compared with some of the reported multipliers of the same width. At 1.3-V, our work achieved, better multiplication time compared to the multipliers reported in [11, 12], which used CMOS technology. At 100 MHZ, power consumption is less than half that of [10] and even less when compared to [12]. Similar observation is made at 1.3-V when our multiplier is compared to the one reported in [7] which is based on a 0.18 µm CMOS process, where over 50% reduction in power is obtained.

CONCLUSIONS

We have presented several low-power PT circuit techniques for parallel multiplication. Taking full advantage of the low-transistor-count and NFS nature of PT logic, we have successfully implemented low-power circuit blocks which serve as basic building units within a 32×32-bit multiplier, including a new Booth's encoder and a modified 4-2 compressor. Experimental measurements on the fabricated multiplier and comparisons with other reported multipliers have verified its low-power characteristics.

Table 3: Comparison between 32×32 bit multipliers

	This work	[11]	[12]
Technology (µm)	0.13	0.13	0.13
Transistor counts	21579.00	25258.00	32369.00
Multiplication time (ns)	3.30	3.50	4.10
Chip area (mm^2)	0.69	1.10	1.27
Power diss (mW/MHz)	0.80	1.37	2.23
PDP (pJ@100MHz)	649.00	1300.00	1450.00

A high performance, low hardware cost and low power asynchronous iterative multiplier has been developed in this work. The multiplier totally consists of 21579 CMOS elements and completes a 32×32-bit multiplication in 3.3 ns under the typical-case conditions. It suits for both signed and unsigned operands. The design uses the modified Booth algorithm. An early termination scheme is employed which efficiently speeds up the operation.

Table 3 shows the comparison between 32×32-bit multipliers. Compared to other multipliers, this multiplier is smaller.

Power×Latency is often used as a metric for the power consumption of a CMOS system. Given two designs A and B, if the Power×Latency of A is smaller than that of B, then A consumes less power than B when they operate the same number instructions. The average Power×Latency of this work is only 1/2 of that of [11] multiplier. So this work is much more power efficient than other designs. The reason is eliminating the propagation of the glitches through the whole data path. The unnecessary switches waste quite a lot power especially when they are propagated through the whole data path. The immediately closing of the latches prevents glitches from propagating to the next stages, thus saves power. From this example, we can see that the hybrid handshaking protocol is a good choice for low power circuits. Another reason is that during early termination period, only shift registers row consumes power, while compressors rows are "free". However, for synchronous system, it is very difficult to stop some parts of the data path. Our multiplication algorithm showed 7.4 percent speed improvement, 11 percent power savings and 9.5 percent reduction in transistor count when compared to the conventional multiplication algorithms.

REFERENCES

1. Kyoung, H.L., 2003. Design of an 8-bit multiplier using dynamic pass transistor logic. IEEE J. solid-state Circuits, 40: 279-285.
2. Law, C.F., 1999. A Low Power 16×16-b parallel multiplier utilizing pass-transistor logic. IEEE J. Solid State Circuits, 34: 1395-1399.
3. Yang, G., S.O. Jung, K.H. Baek, S.H. Kim, S. Kim and S.M. Kang, 2005. A 32-bit carry lookahead adder using dual-path All-N logic. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 13: 992-996.
4. Satoro, M.R. and M.A. Horowitz, 2004. A pipelined 64×64-bit multiplier. IEEE J. Solid-State Circuits, 24: 642-651.
5. Zhuo, L. and V.K. Prasanna, 2007. Scalable and Modular Algorithms for Floating-Point Matrix Multiplication on Reconfigurable Computing Systems. IEEE Transactions on Parallel and Distributed Systems, 18: 433-448.
6. Ohkubo, N., M. Suzuki, T. Shinbo and T. Yamanaka, 1995. A 4.4 ns CMOS 54×54-b multiplier using pass-transistor multiplexer. IEEE J. Solid State Circuits, 30: 251-257.
7. Twagiry, H.A. and M.J. Flynn, 2004. Technology scaling effects on multipliers. IEEE Transactions on Computers, 47: 1201-1215.
8. Cheng, F. and M. Theobald, 2000. Self timed carry-lookahead adders. IEEE Transaction on Computers, 49: 659-672.
9. Liu, Y., 2003. An asynchronous pipelined 32×32-bit iterative multiplier using hybrid handshaking protocol. IEEE J. Solid State circuits, 29: 256-262.
10. Wang, Z. and W.C. Miller, 2005. A new design technique for column compression multipliers. IEEE Transactions on Computers, 44: 962-970.
11. Kang, J.Y. and J.L. Gaudiot, 2006. A simple high-speed multiplier design. IEEE Transaction on Computers, 55: 1253-1258.
12. Itoh, N., Y. Naumura, H. Makino, Y. Nakase, T. Yoshihara and Y. Horiba, 2006. A 54×54-bit multiplier with rectangular-styled Wallace tree. IEEE J. Solid-State Circuits, 36: 249-257.