Experimental results: To verify the effectiveness of the charge pump boosting circuit, one PLL with a charge pump boosting circuit and another one without a boosting circuit (the same as the conventional PLL) have been implemented in a 0.8μm CMOS technology. Fig. 3 shows the maximum frequency to which both PLLs can lock with a supply voltage change. The proposed PLL can be locked to 672MHz at 3.3V supply, which is ~ 1.9 times the maximum frequency of the conventional PLL. The jitter measurement results are as follows. When the supply noise is not injected, the rms jitter is 8.3ps and the peak-to-peak jitter is 56ps. When 0.6V peak-to-peak supply noise is superposed with V_{dd}, the rms jitter is 67ps and the peak-to-peak jitter is 270ps when the operating frequency is < 592MHz. When the operating frequency is > 592MHz, the source follower enters the linear region. As the PLL does not have enough noise margin at that condition, the jitter becomes larger abruptly, resulting in 161ps (rms) and 512ps (peak-to-peak) jitter.

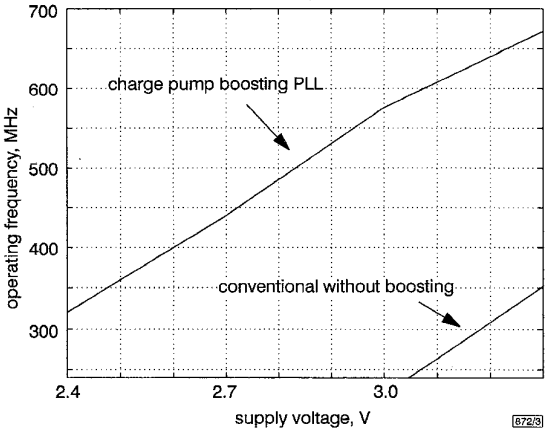

Fig. 3 *Maximum PLL lock frequency against supply voltage*

Conclusion: We have designed and implemented a PLL which can endure a 0.6V peak-to-peak power supply noise. By using the proposed charge pump boosting technique, we could extend the operating frequency range of the PLL with minimum added complexity and power consumption, achieving low jitter.

© IEE 1998
2 June 1998
Electronics Letters Online No: 19981057

Kwangho Yoon and Wonchan Kim (*School of Electrical Engineering, Seoul National University, San 56-1, Shilim-dong, Kwanak-gu, Seoul 151-742, Korea*)

E-mail: kwangho@iclab.snu.ac.kr

References

1 ZHANG, , DU, , and LEE, : 'A 360MHz 3 V CMOS PLL with 1V peak-to-peak power supply noise tolerance'. ISSCC Dig. Technical Papers, February 1996, pp. 134–135

2 VON KAENEL, V., AEBISCHER, D., PIGUET, C., and DIJKSTRA, E.: 'A 320MHz, 1.5mW @1.35V CMOS PLL for microcessor clock generation', *IEEE J. Solid-State Circuits,* 1996, **31**, pp. 1715–1722

3 KONDOH, H., and NOTANI, H.: 'A 1.5V 250MHz to 3.0V 622-MHz operation CMOS phase-locked loop with precharge type phase-detector', *IEICE Trans. Electron.,* 1995, **E78-C**, (4), pp. 381–388

Radix-4 multiplier with regular layout structure

Bongil Park, Myoungcheol Shin, In-Cheol Park and Chong-Min Kyung

A new parallel multiplier with a regular layout structure is described. To achieve a regular structure without sacrificing performance, a new circuit called the weighted carry save adder is proposed, which enables the multiplier not only to have a very regular layout but also to have a smaller operating size at the final adding stage than that of conventional schemes.

Introduction: Fast and area efficient multipliers are indispensable for high-performance computers and data-processing systems. Parallel array multipliers achieve good performance at the expense of a large amount of silicon, while low-performance multipliers based on the add-and-shift algorithm require less hardware [1]. Since the multiplication time is proportional to the number of partial products to be added, one of the most important objectives of multiplier design is to reduce the number of partial products. The most popular method is to use a modified Booth's algorithm which can reduce the number of partial products by one-half [5]. However, parallel multipliers based on the modified Booth's algorithm are not regular enough in terms of layout. Regularity is necessary not only for enhancing design productivity and multiplier bit-width extendibility, but also for improving performance [1].

In this Letter, we propose a new circuit called the weighted carry save adder (WCSA) to achieve a very regular structure and to enhance performance.

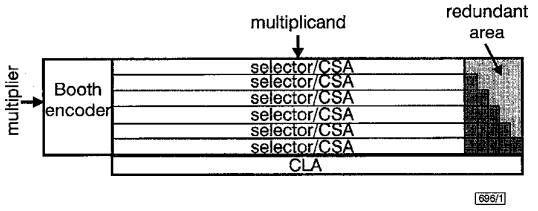

Fig. 1 *Conventional n-bit Booth encoded multiplier*

Multiplier architecture: As shown in Fig. 1, a parallel array multiplier based on a radix-4 modified Booth's algorithm consists of a Booth encoder, rows of carry save adders (CSAs), rows of selectors, and a carry lookahead adder (CLA) [4]. The selector chooses a partial product among five candidates: 0, $+A$, $+2A$, $-A$, and $-2A$.

This multiplier has two main drawbacks which need to be improved. One is that the number of bits to be added is increased by one bit at every successive row of CSAs, and the other is that the number of bits to be added at the final adding stage is greater than n when the multiplier manipulates n bits. These drawbacks result from the fact that each successive partial product should be shifted by two bits. In order not to increase the delay of each row, one more bit is necessary at each successive row. These additional bits are shown in the box on right-hand side of Fig. 1. Because the bits increase at each CSA stage, the final adder should handle $n + (n–2)/2$ bits. The additional $(n–2)/2$ bits cause the resulting layout to be trapezoid, and degrade the multiplier performance because the operation size of the CLA should be lengthened. If we could eliminate these bits, we would reduce silicon area and enhance performance. In order not to increase the number of bits at each CSA state, while keeping the delay, a new multiplier structure is proposed based on the weighted carry save adder (WCSA). As

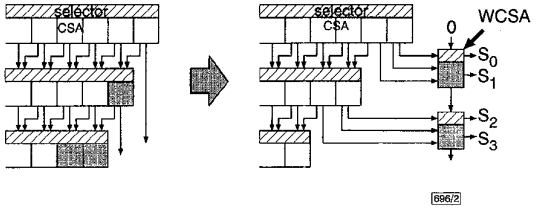

Fig. 2 *CSA rows with WCSA*

shown in Fig. 2, the WCSA accepts four inputs, three from the previous CSA row and one from the previous WCSA, and then generates 2 bit sum outputs and a 1 bit carry output. The WCSAs are connected like a ripple carry adder, meaning that the carry output of the previous WCSA is connected to the carry input of the following WCSA.

Given four inputs $\{x_{i+1}, y_{i+1}, x_i, c_i\}$, the arithmetic operation of the WCSA is described as follows:

$$4 \cdot c_{i+2} + 2 \cdot s_{i+1} + s_i = 2 \cdot x_{i+1} + 2 \cdot y_{i+1} + x_i + c_i \quad (1)$$

$$s_i = x_i \oplus c_i \quad (2)$$

$$s_{i+1} = x_{i+1} \oplus y_{i+1} \oplus x_i c_i \quad (3)$$

$$c_{i+2} = x_{i+1}y_{i+1} + x_i c_i(x_{i+1} + y_{i+1}) \quad (4)$$

where s_i, s_{i+1}, c_{i+2} are the ith sum, $(i+1)$th sum, and $(i+2)$th carry signals, respectively.

From eqns. 2 – 4, the sum and carry equations of the WCSA are slightly different from those of the CSA [5]. This implies that the circuit delay of the WCSA is similar to that of the CSA. As WCSAs are connected like a ripple adder, the critical output of the WCSA is c_{i+2} which has less delay than the sum output of the CSA. The overall structure of our multiplier is shown in Fig. 3, where the additional bits inevitable in the conventional CSA array are eliminated using the WCSA. Furthermore, we improve overall performance because the operating size of the CLA is reduced by eliminating the additional bits which appear in conventional implementations. To handle two's complement numbers, an $(n+3)$-bit CSA is necessary for each row, for the following reasons.

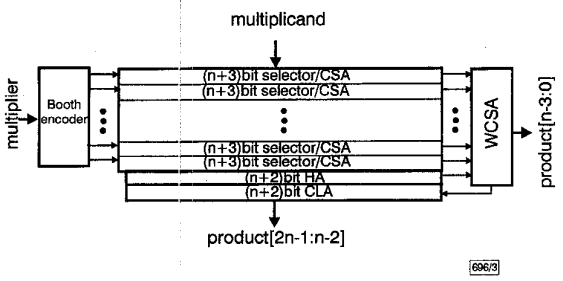

Fig. 3 *Block diagram of proposed multiplier*

In the radix-4 modified Booth's algorithm, five different operations can be encountered at each row: 0, $+A$, $+2A$, $-A$, and $-2A$. If two input operands are represented by n bit two's complement numbers, the negative operand, $-A$ or $-2A$, is calculated by shifting and inverting the original operand ($\bar{A}$, $\overline{2A}$) and then adding 1 at the least significant bit of the next CSA row.

For negative numbers represented in the two's complement method, we have to make sure that the sign bit is properly extended before the addition of the partial product takes place. We complement the original sign bit s to obtain $\bar{s}$ and add 1 [5].

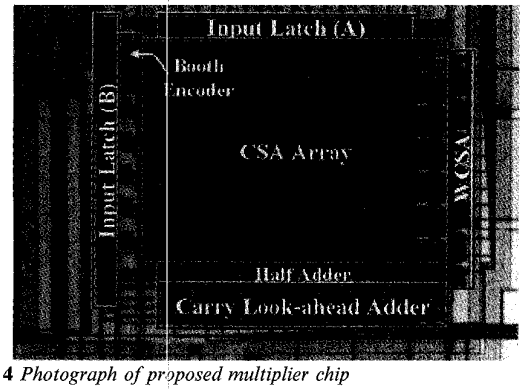

Fig. 4 *Photograph of proposed multiplier chip*

Simulation and implementation: In a 0.8μm CMOS DLM technology, HSPICE simulation shows that the carry delays of the CSA and WCSA are 1.2 and 1.3ns, respectively, while the sum delay of the CSA is 1.5ns. The delay is measured for the worst case input pattern transition. As expected, the carry evaluation delay of the WCSA is placed between the carry delay and sum delay of the CSA. Since the critical path of the WCSA chain is not in sum generation but in carry generation, there is no performance penalty in using the WCSA because the carry delay of the WCSA is less than the sum delay of the CSA.

The 16 × 16 multiplier based on the WCSA was implemented in 0.8μm CMOS DLM technology, which contains 7000 transistors with a core size of 1.04 × 1.15mm². Fig. 4 shows a photograph of the chip which consists of the modified Booth encoder, six stages of 19 bit CSAs, seven WCSAs, an 18 bit half adder (HA), and an 18 bit CLA. The upper part, $s_{31:14}$, is calculated at the CLA whose operation size is 18 bit. If we used the conventional implementation, the CLA should handle 24 bit. All possible input patterns were tested for this chip and the multiplication time measured with the worst case pattern was 16.5ns (60MHz).

Conclusion: A new circuit called the WCSA is described to overcome the irregular structure caused by the 2 bit shifting of modified Booth encoded multiplication. The WCSA yields a more efficient and regular layout structure than the previous implementations. Furthermore, the WCSA makes it possible to improve overall performance by reducing the bit length to be added at the final adding stage. We are considering a high performance floating-point multipler which is based on both Wallace tree connection [2, 3] and the WCSA.

© IEE 1998
28 May 1998
Electronics Letters Online No: 19981068

Bongil Park, Myoungcheol Shin, In-Cheol Park and Chong-Min Kyung (*Dept. of EE, KAIST, 373-1, Kusong-dong, Yusong-gu, Taejon 305-701, Korea*)

E-mail: {bipark, mcshin}@duo.kaist.ac.kr
{icpark, kyung}@eekaist.kaist.ac.kr

References

1 SANTORO, M.R., and HOROWITZ, M.A.: 'SPIM:A pipelined 64 × 64-bit iterative multiplier', *IEEE J. Solid-State Circuits,* 1989, **24**, (2), pp. 487–493
2 GAMAL, A.E., GLUSS, D., ANG, P.-H., GREENE, J., and REYNERI, J.: 'A CMOS32b Wallace tree multiplier-accumulator'. ISSCC Dig. Tech. Papers, 1986, pp. 194–196
3 KAJI, Y., and SUGIYAMA, N.: 'A 45ns 16 × 16 CMOS multiplier'. ISSCC Dig. Tech. Papers, 1983, pp. 32–33
4 ANDERSON, J.M., TROUTMAN, B.L., and ALLEN, R.A.: 'A CMOS LSI 16 × 16 multiplier/multiplier-accumulator'. ISSCC Dig. Tech. Papers, 1982, pp. 124–125
5 KOREN, I.: 'Computer arithmetic algorithms' (Prentice Hall, 1993)

Reversed ROBDD circuits

A. Bystrov and A.E.A. Almaini

A new class of logic circuits is proposed. Being derived from reduced ordered binary decision diagrams, these circuits inherit compactness and the ablility to represent very large switching functions. The reversed signal propagation results in low dynamic power consumption, complete single stuck-at fault testability and fault security. Simulation results are presented.

Introduction: Reduced ordered binary decision diagrams (ROBDDs) represent switching functions in the form of canonical rooted directed graphs. This representation is based on the Shannon decomposition $f = \overline{x_j} \cdot f_0 + x_j \cdot f_1$ presenting a Boolean function $f(X)$ as a composition of two co-factors: $f_0 = f(X|x_j = 0)$ and $f_1 = f(X|x_j = 1)$. The recursive decomposition results in a binary tree. Non-terminal nodes of such a tree represent the Shannon formula, having two inputs corresponding to the co-factors and one primary input x_j. Terminal nodes (leaves) are 0 and 1 constants. The binary tree is reduced by replacing the equivalent subgraphs by a single instance of the subgraph and a fanout, as described in [1]. The ROBDD size depends on the variable order used in the recursive decomposition. There are many efficient methods for the variable order optimisation [2, 3]. Our experiments with LGSynth-95, ISCAS-85 and ISCAS-89 benchmark sets have shown that ~65% of circuits can be represented by ROBDDs where the number of nodes do not exceed the number of logic gates in the original circuits. This makes ROBDDs extremely attractive for logic synthesis tasks. Unfortunately, the replacement of ROBDD nodes by multiplexers (MX) implementing the Shannon formula results in redundant circuits [4], which are not completely testable in the stuck-at fault model (SAFM). Redundancy elimination changes the structure of some nodes and destroys the regularity of the circuit.

In this Letter, a new method of ROBDD implementation is proposed. It uses a reversed signal propagation, from the root to the terminal nodes, and leads to better testability, low dynamic power consumption and fault secure properties.

Reversed ROBDDs: A single output ROBDD is a directed graph connecting one of the terminal nodes and the root by a path, as