LOW-LATENCY AND HIGH-EFFICIENCY BIT SERIAL-SERIAL MULTIPLIERS

Paraskevas Kalivas, Kiamal Pekmestzi, Paul Bougas, Andreas Tsirikos and Kostas Gotsis

School of Electrical and Computer Engineering, National Technical University of Athens,
157 73 Zographou, Athens, Greece.
phone: +30 210 7722500, fax: +30 210 7722428, e-mail: pekmes@microlab.ntua.gr
web: www.microlab.ntua.gr

ABSTRACT

A new bit serial-serial multiplier for unsigned numbers is presented in this paper. The numbers being multiplied enter the circuit simultaneously in LSB first bit-serial form. The multiplier is of immediate response, pipelined at the bit level and has small combinational delay. A variation of this multiplier that operates with 100% efficiency, namely it does not require zero bits to be inserted between successive input data words, produces the product in full-precision is also proposed. The proposed schemes are well suited for DSP applications, compared with other schemes exhibit superior performance in terms of hardware complexity and throughput.

1. INTRODUCTION

Bit-serial arithmetic is often used in order to reduce the wiring and the hardware complexity. Serial-parallel multiplier [1] is the most prevalent multiplication architecture because of its simple structure and low hardware complexity, but it has the disadvantage that it the one factor has to be loaded in parallel. In [2] a modification of the serial/parallel multiplier has been presented so that both numbers being multiplied enter the circuit serially, but it has the drawback that the entrance of the one factor must start a number of clock cycles prior to the entrance of other. In DSP applications where real-time response and low-latency is required this is a serious disadvantage.

The bit serial-serial multipliers have been proposed for applications where both the multiplicand and the multiplier arrive simultaneously. Such a multiplier for two's complement numbers has been presented in [3] but it has the drawback that it truncates the result and introduces latency dependent on the bit-length of the multiplied numbers. Full-precision serial multipliers, in order to keep the round-off error of the intermediate results low, are often needed. Full-precision serial-serial multipliers for unsigned numbers are presented in [4] and [5]. A full-precision scheme for two's complement numbers has been presented in [6]. The major drawbacks of the above designs are that they require double circuit complexity and are not of immediate response, namely at least one clock cycle is required between the first bit of the input and the first bit of the output. A full-precision multiplier of immediate response is presented in [7].

Another desirable feature of serial multipliers is the 100% operational efficiency. Normally, a number of zero bits equal to the coefficient word length are required between successive input data words in order to obtain the full result. Consequently, in case that both multiplied numbers have the same number of bits the multiplier operates with 50% efficiency. The multiplier presented in [3] operates with 100% efficient but at the expense of high latency proportional to the length of the multiplication factors and truncated result. Another technique applied to the serial/parallel has been presented in [2] and [8] and achieves 100% operational efficiency.

In this paper we propose a new bit serial-serial multiplier for unsigned numbers based on a special multiplication algorithm. This algorithm has been presented in [9] and applied to the design of parallel multipliers yielding better results in terms of hardware complexity. The same algorithm is applied in this paper to the design of a bit serial-serial multiplier that is of immediate response and has smaller hardware complexity and combinational delay compared with other bit serial-serial multipliers. A second scheme based on the first is also proposed that operates with 100% efficiency, where the technique presented in [2] is employed. Both schemes are compared with the scheme presented in [7].

The organization of the papers is the following: In Section 1, the multiplication algorithm that the proposed multiplier is based on is presented concisely. The proposed schemes are presented in Section 2. The comparison between those schemes and the scheme in [7] is presented in Section 3.

2. THE MULTIPLICATION ALGORITHM

Let us consider two positive integer numbers X and Y:

$$X = x_{n-1}x_{n-2}\cdots x_0 = \sum_{j=0}^{n-1} x_j 2^j \quad (1)$$

$$Y = y_{n-1}y_{n-2}\cdots y_0 = \sum_{j=0}^{n-1} y_j 2^j \quad (2)$$

The algorithm presented in [9], requires both multiplied numbers X and Y to be of the same length. If one of them is shorter then we assume that zero bits are adding to its most significant bit positions so that both numbers are of the same length.

According to this algorithm the product of X and Y is given by the following equation

$$P = \sum_{j=0}^{n-1} x_j y_j 2^{2j} + \sum_{j=1}^{n-1} Z_j 2^j \quad (3)$$

where

$$Z_j = x_j Y_j + y_j X_j,\ X_j = \sum_{i=0}^{j-1} x_i 2^i \text{ and } Y_j = \sum_{i=0}^{j-1} y_i 2^i \quad (4)$$

The values that the quantity Z_j can take depend on the values of x_j and y_j and are shown in Table I.

Table I: The values of Z_j.

x_j	y_j	Z_j
0	0	0
0	1	X_j
1	0	Y_j
1	1	$X_j + Y_j = S_j$

A graphical representation of the above algorithm is shown in Fig. 1.

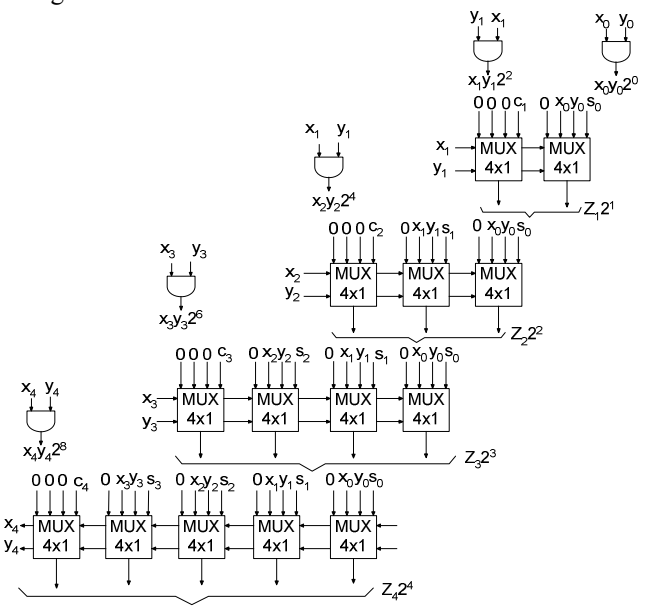

Figure 1: The implementation of the terms $Z_j 2^j$ and $x_i y_j 2^{2j}$ used in the multiplication algorithm.

The chain of 4x1 multiplexers in each row computes the quantity $Z_j 2^j$ according to Table I. The quantity $S_j = c_j s_{j-1} s_{j-2} \ldots s_0$ is not required to be computed in each row separately, since $S_j = S_{j-1} + x_{j-1} 2^{j-1} + y_{j-1} 2^{j-1}$.

3. THE PROPOSED MULTIPLIER

The timing diagram in Fig. 2 clarifies the operation of the propose multiplier. In this diagram with X_j and Y_j are represented the two numbers being multiplied, which have length equal to *w* and *m* bits respectively. The least and most significant part of the corresponding products are represented as $P_{j,LSP}$ and $P_{j,MSP}$. The diagram shows the necessity of padding both multiplied numbers with zeros so that their length to become equal to the length of the product.

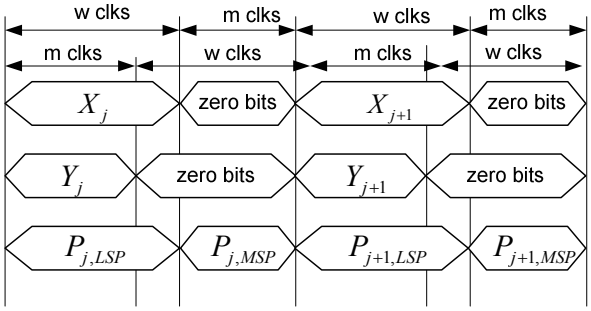

Figure 2: The timing diagram for the proposed multiplier which operates with 50% efficiency.

The circuit implementation of the proposed scheme is shown in Fig. 5. The multiplier implements the previously described algorithm as it is depicted in Fig. 1. During every clock cycle the multiplier adds the quantity that corresponds to a row of the diagram in Fig. 1 and because the length of these rows increases progressively from top to down the multiplier cells are involved progressively into the computation starting with the right most cell. The *j*-th cell of the multiplier starts to participate in the multiplication at the *j*-th clock cycle of the current multiplication.

Each multiplier cell includes a Full-Adder, a 4:1 multiplexer and three registers for storing the bits x_j, y_j and s_j. The terms $x_j y_j$, s_j and c_j are computes by the AND gate and the serial adder at the right side of the circuit. The control signal R is a travelling one that enters the circuit a clock cycle before the entrance of the least significant bits of X_j and Y_j. Actually, this signal determines the cell that will start to participate in the multiplication. Because the output of the 4:1 multiplexer must be zero during the j-th clock cycle, where the quantities $x_j y_j$ and c_j are added through the 4:1 multiplexer, the signal R resets the three registers that hold the bits x_j, y_j and s_j of the previous multiplication. The clock input of these register is driven by the control signal R. Hence, from the *j*+1-th clock cycle and for the rest of the multiplications these registers store the bits x_j, y_j and s_j.

According to Fig. 1 when the last bit of X_j renters the multiplier, the output of least significant part of the product $P_{j,LSP}$ is being completed while the most significant part is stored in carry-save form into the delay elements at the sum and carry outputs of each Full-Adder. The circuit operation must be continued for additional *m* clock cycles with the data input kept to zero in order the most significant part $P_{j,MSP}$ to be obtained.

The proposed scheme that operates with 100% efficiency is shown in Fig. 5. The timing diagram in Fig. 4 clarifies its operation. According to this figure the least and most significant parts of the product are obtained by different outputs noted with P_L and P_H in Fig. 5. The computation of the

most significant part $P_{j,LSP}$ is overlapped with that of the least significant part of the next product $P_{j+1,MSP}$. Thus, no zero bits have to be inserted between successive X data words. Only Y data words must be extended by $w-m$ zero bits, when Y is shorter than X.

The operation of the circuit in Fig. 6 has as following: During the first w clock cycles of the multiplication the least significant part of the result $P_{j,LSP}$ is produced in binary form while the most significant part $P_{j,MSP}$ in carry-save form and is stored in the delay elements at the sum carry outputs of each Full-Adder. Afterwards $P_{j,MSP}$ is downloaded into a double shift register at the bottom of the circuit, while the multiplier starts the computation of the least significant part of the next product $P_{j+1,LSP}$. At the same time the content of the double register starts shifting through the bit-serial adder at the right of the multiplier end and the most significant part $P_{i,MSP}$ is getting out in binary form.

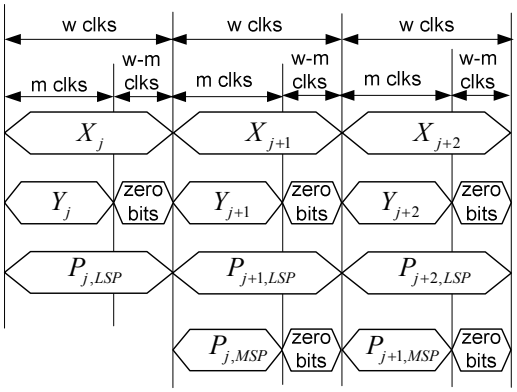

Figure 4: The timing diagram for the proposed multiplier that operates with 100% efficiency.

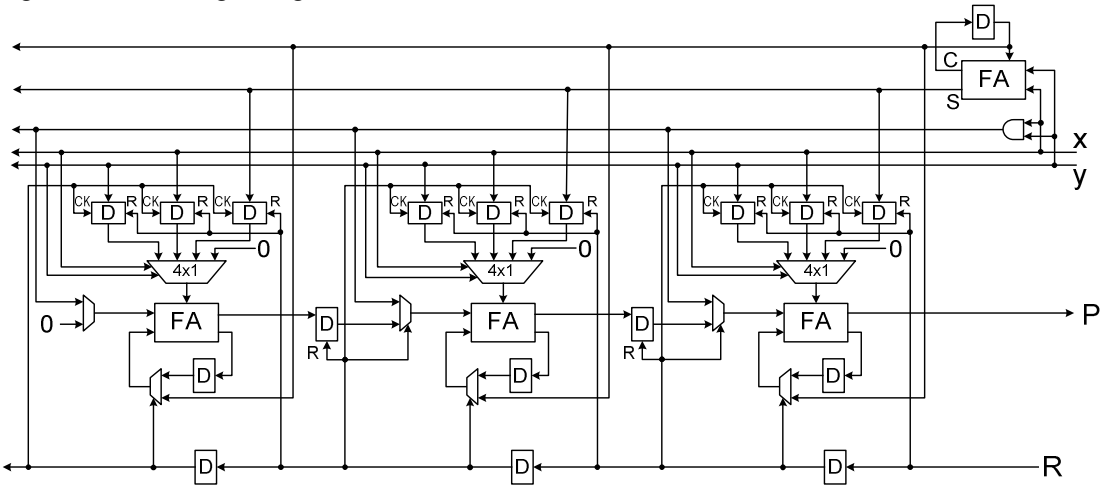

Figure 5: The 50% efficient proposed multiplier for the multiplication of 3-bit long numbers.

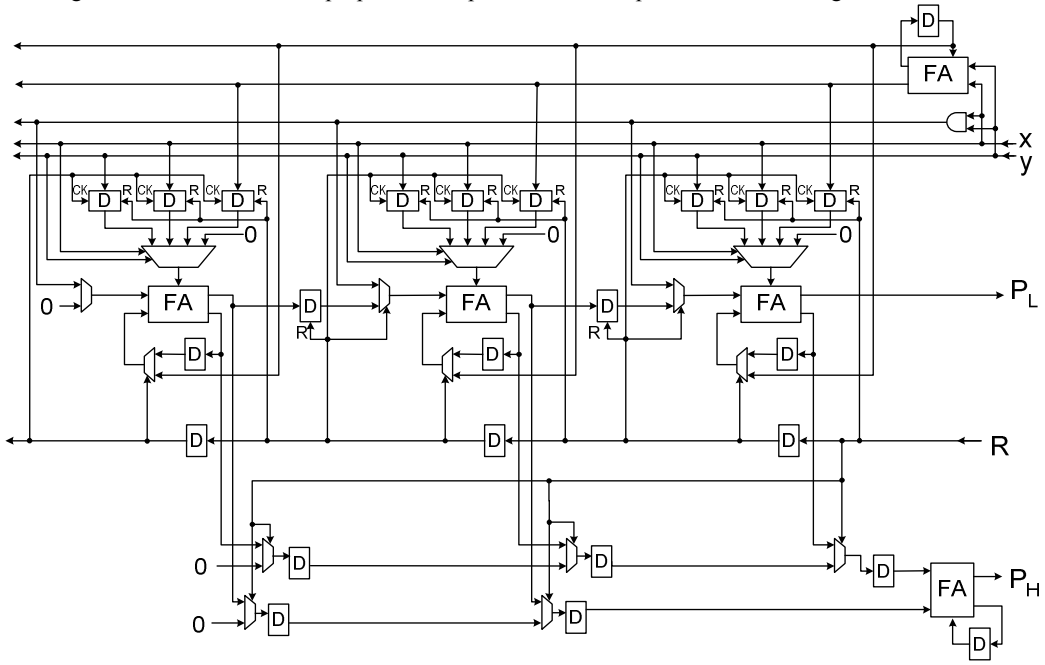

Figure 6: The 100% efficient proposed multiplier for the multiplication of 3-bit long numbers.

4. COMPARISON OF THE PROPOSED MULTIPLIERS WITH OTHER SCHEMES

We compare the proposed multiplier with the multiplier presented in [7], because this multiplier has smaller hardware complexity than schemes and is of immediate response. The hardware complexities of the proposed schemes and the scheme in [7] are shown in Table II. The 50% efficient proposed multiplier requires 20% less hardware than the multiplier presented in [7]. The 100% efficient proposed multiplier has the same hardware complexity as the scheme in [7]. Both proposed schemes have significantly smaller combinational delay than the scheme in [7].

In order to have a total estimation of the Hardware Efficiency we introduce the factor $E = \frac{1}{HT_M}$, where H and T_M are the hardware complexity and the critical delay of the multiplier respectively. The value of this factor for each multiplier has also been included in Table II and shows that the proposed schemes are three and five times superior to the previous one.

Another advantage of the proposed schemes is that they require only one control signal, the signal *R,* which is activated at the beginning of every multiplication. The multiplier in [7] requires two control signals one activated at beginning of the multiplication and the other when the last bits of the multiplication factors enter the circuit.

Table II. Comparison of the proposed multipliers and the multiplier presented in [7].

Multiplier Type	Hardware Complexity per multiplication cell	Hardware Complexity (transistors)	Combinational Delay	Operational Efficiency	Hardware Efficiency Factor $E=1/H\cdot T_M$
The 50% efficient proposed multiplier.	$2D+4DR+2SW+MUX+FA$	116	$T_{MUX} + T_{FA}$	50%	$E_{p1} = 1/116\cdot T_{FA}\cdot 2n \approx 3\cdot E$
The 100% efficient proposed multiplier.	$4D+4DR+4SW+MUX+FA$	144	$T_{MUX} + T_{FA}$	100%	$E_{p2} = 1/144\cdot T_{FA}\cdot n \approx 5\cdot E$
The multiplier presented in [7]	$6DR+SW+2.5FA+2AND$	146	$2.5T_{FA}$	50%	$E = 1/365\cdot T_{FA}\cdot 2n$

D:1-bit register without reset(8 transistors), DR:1-bit register with synchronous reset (12 transistors), SW:2-input switch (6 transistors), MUX: 4:1 multiplexor (16 transistors), FA:Full-Adder (24 transistors), AND:2-input AND gate (4 transistors)[10]

4. CONCLUSIONS

A new bit serial-serial multiplier for unsigned numbers is presented in this paper. The multiplier is of immediate response, pipelined at the bit level and has smaller combinational delay compared with previously presented schemes. A variation of this multiplier that operates with 100% efficiency is presented in this paper. The proposed scheme yields significant performance of three and five times superior compared to the previous implementations. It can be easily extended to perform operations with both numbers in two's complement form and is well suited for DSP applications.

REFERENCES

[1] L. Dadda, "On serial-input multipliers for two's complement numbers," *IEEE Trans. On Comput*., vol. 38, no. 9, pp.1341-1345, Sept. 1989.

[2] K.Z. Pekmestzi, P. Kalivas, N. Moshopoulos, "Long unsigned number systolic serial multipliers and squarers", *IEEE Circuits and Systems II: Transaction on Analog and Digital Signal Processing*, vol. 48, no 3, pp. 316 -321, Mar. 2001.

[3] R.F. Lyon, "Two's complement pipeline multipliers", *IEEE Trans. Commiun.*, vol. COM-24, no. 4, pp. 418-425, Apr. 1976.

[4] H. J. Sips, "Comments on 'An O(n) parallel multiplier with bit-sequential input and output", *IEEE Trans. Comput*., vol. C-31, no. 4, pp. 325-327, Apr. 1982.

[5] N. R. Strader and V. T. Rhyne, "A canonical bit-sequential multiplier", *IEEE Trans. Comput.*, vol. C-31, no. 8 , pp. 791-795, Aug. 1983.

[6] R. Gnanasekaran, "On a bit serial-input output multiplier", *IEEE Trans. Comput.*, vol.C-32, no. 9, pp. 878-880, Sept. 1983.

[7] P. Ienne, M. A. Viredaz, "Bit-Serial Multipliers and Squarers," *IEEE Trans. On Comput*., vol. 43, no. 12, pp. 1445-1450, 1994.

[8] G. Even, "Two's complement pipeline multipliers," *Integration, the VLSI journal*, no 22, pp. 23-38, 1997.

[9] K.Z. Pekmestzi, "Multiplexer-Based Array Multipliers", *IEEE Trans. On Comput.*, vol. 48, no. 1, pp. 15-23, Jan. 1999.

[10] N.H.E. Weste, K. Eshraghian: *Principles of CMOS VLSI Design*, Addison-Wesley, 1993.