# NOVEL SYSTOLIC SCHEMES FOR SERIAL-PARALLEL MULTIPLICATION

*I. Sideris, K. Anagnostopoulos, P. Kalivas, and K. Pekmestzi*

Department of Electrical and Computer Engineering, National Technical University of Athens
Iroon Polytexneiou 9, 15780, Athens, Greece
phone: + 30 2107722500, fax: + 30 2107722428, email: {isidoros, kanagno, paraskevas, pekmes}@microlab.ntua.gr
web: www.microlab.ntua.gr

## ABSTRACT

In this paper two new schemes of systolic multipliers are proposed, one based on Modified Booth encoding and the other is based on the selection of one of the terms 0, X, 2X, 3X where x is the serial input of the multiplier. The proposed multipliers operate with 100% efficiency that is without zero words inserted between successive data. Systolisity and the continuous operation are achieved without an increase in hardware complexity. The proposed schemes are especially suited for long number multiplication.

## 1. INTRODUCTION

Bit-serial signal processing has gained significant impact in recent years. The decreased area of the serial modules leads to low power consumption due to decreased leakage current, that in the sub-micron technology becomes the dominant factor. Additionally in applications like cryptography, where long number computations are required, serial parallel schemes are used to reduce the hardware complexity and the wiring, down to a reasonable level.

The serial/parallel multiplier [1] is the most widely used architecture for implementing multiplication because of its simple structure. The demerit of this scheme lies on the fact that a zero word must be inserted after each data in order to empty the product from the multiplier. Due to the insertion of these zero words, the efficiency of this unit is 50%. Additionally this scheme is not systolic, a severe drawback when multiplying large numbers.

The multiplier proposed in [2] is based on Modified Booth encoding of the parallel factor, is systolic, but its operation is limited to 50%. In [3] the authors propose a systolic serial-parallel multiplier which operates with 100% efficiency without increasing the hardware complexity compared to the previous schemes.

In this paper we apply the technique introduced in [3] to the above Modified Booth serial-parallel multiplier in order to double its operational rate. To avoid the overhead of Modified Booth encoding another scheme is devised where parallel factor is in usual binary form. In this scheme bits are handled in pairs and select the addition of 0, X, 2X or 3X. The term that is not available is 3X and must be computed on the fly. This can be easily implemented due to the serial nature of X.

The rest of the paper is organised as follows. In section 2 the scheme for serial-parallel multiplication based on Modified Booth Algorithm (MBA) with 100% efficiency is presented. In Section 3 the scheme for serial-parallel multiplication based on the generation of 3X is presented and finally in section 4 there are comparisons between the proposed and the existing architectures.

## 2. SYSTOLIC SERIAL-PARALLEL MODIFIED BOOTH MULTIPLIER

The multiplication based on the MBA [4] reduces the number of partial products by half.

The algorithm is based on the following relations:

$$A=\sum_{i=0}^{n-1}\alpha_i 2^i=\sum_{i=0}^{n-1}(\alpha_{i-1}-\alpha_i)2^i=\sum_{i=0}^{n-1}z_i 2^i=$$
$$\sum_{j=0}^{\frac{n}{2}-1}(z_{2j}+2z_{2j+1})2^{2j}=\sum_{j=0}^{\frac{n}{2}-1}w_j 4^j \quad (1)$$

The values of variable w are {-2,-1,0,1,2}. The $MB_j$ cell, shown in Fig. 1, is used to produce the partial product between $w_j$ and X. It is composed of 2:1 MUX which selects X or 2X, an AND gate when $w_j$ is zero and a XOR gate which complements in the case of negative $w_i$

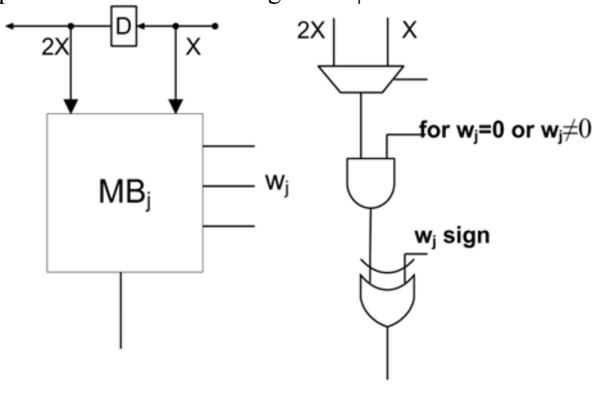

**Fig. 1** Modified Booth partial product generator.

We use the above recoding to design a serial-parallel pipeline multiplier as shown in Fig. 2. Here the number X is in binary-serial form while the parallel factor W is considered to be already transformed in Modified Booth form, so we do not include the necessary hardware for this encoding.

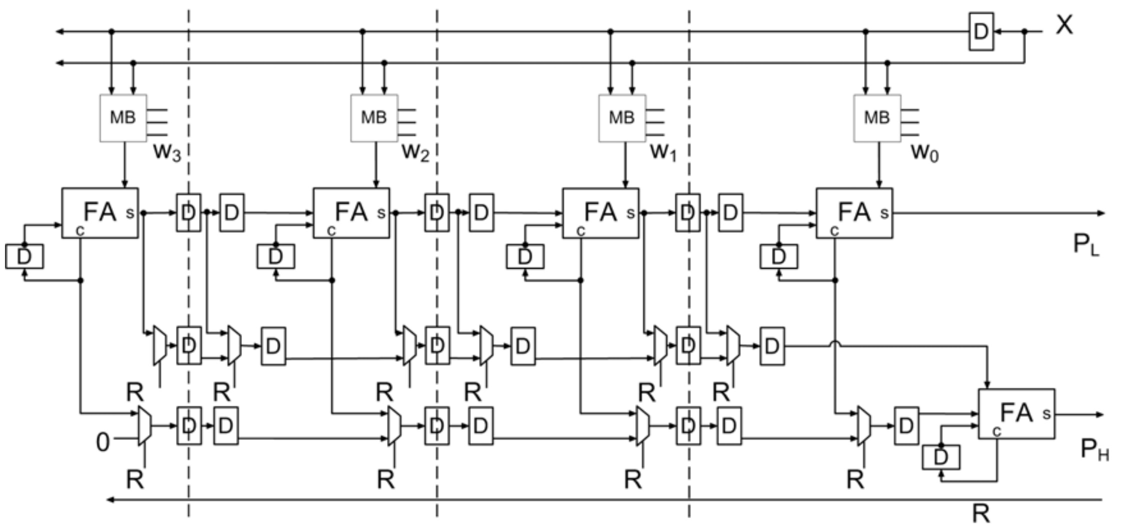

**Fig. 2** Serial-parallel multiplier with one multiplicand in Modified Booth form.

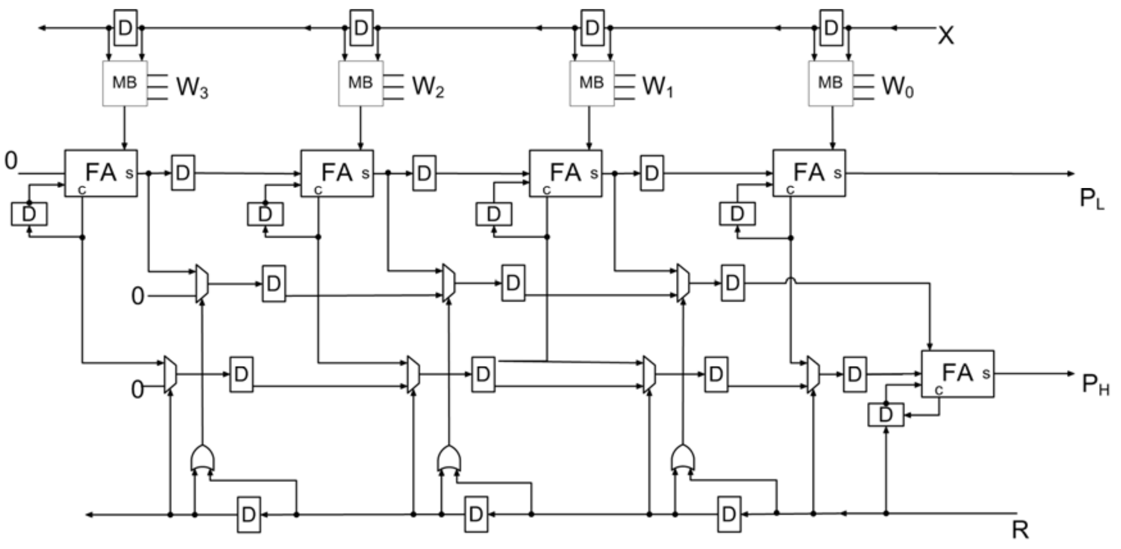

**Fig. 3** Transformed scheme of serial-parallel multiplier of Fig. 2.

Because of the Modified Booth encoding adjacent cells partial products differ by a factor 4. Therefore 2 delay elements must be inserted between the cells as shown in Fig. 2. The least significant part (LSP) of the product is obtained by the output $P_L$ while the most significant part (MSP) by the output $P_H$. When the output of the LSP is completed, the two sum bits and one carry-output of each multiplier cell contain the MSP of the product in carry-save form. Next they are downloaded into a double 2-bit shift register and at the end a serial adder converts them in binary form. R is used as a control signal which initiates the loading to the shift registers making the multiplier available for the next multiplication. We initialize the carry flip-flops with “1” so as to take care the two-complement transformation of A.

A retiming technique [5] can be applied on this circuit in order to convert it into systolic form. According to this technique, if we assume a cut across the graph, we can remove one delay element of all lines of the same direction and insert a delay element into all lines of the opposite way. By applying this transformation, the delay elements located next to a multiplier cell can be removed, and a new delay element at the corresponding position in the data and control signal lines must be inserted. The final circuit is shown in Fig. 3 where we have superseded the two input lines of X with one equivalent. A zero bit must be inserted between successive data words to form and add the term 2X or -2X at each cell. The signal R, which enters synchronously with the above zero bit, is now a travelling one in order to perform the downloading progressively. Also the two switches for the downloading of sum bits shown in Fig.2, are combined to one in Fig.3, triggered in two successive cycles of R using an OR gate.

## 3. SYSTOLIC SERIAL-PARALLEL MULTIPLIER BASED ON 3X PRODUCT

The previous multiplier requires the multiplicand A be in modified Booth form which is assumed to be available. This is true for multiplications with a fixed coefficient, but in the general case is not, thus introducing an overhead for encoding. In order to overcome this difficulty a new multiplier

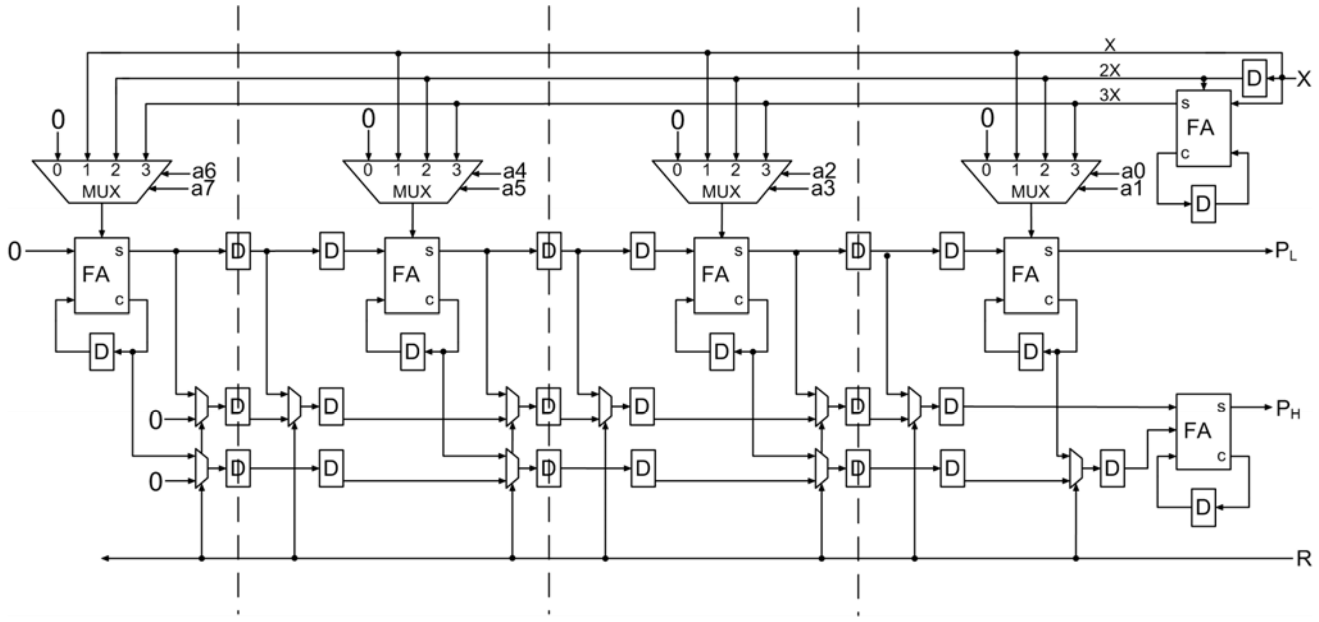

**Fig. 4** Serial-parallel multiplier based on product 3X.

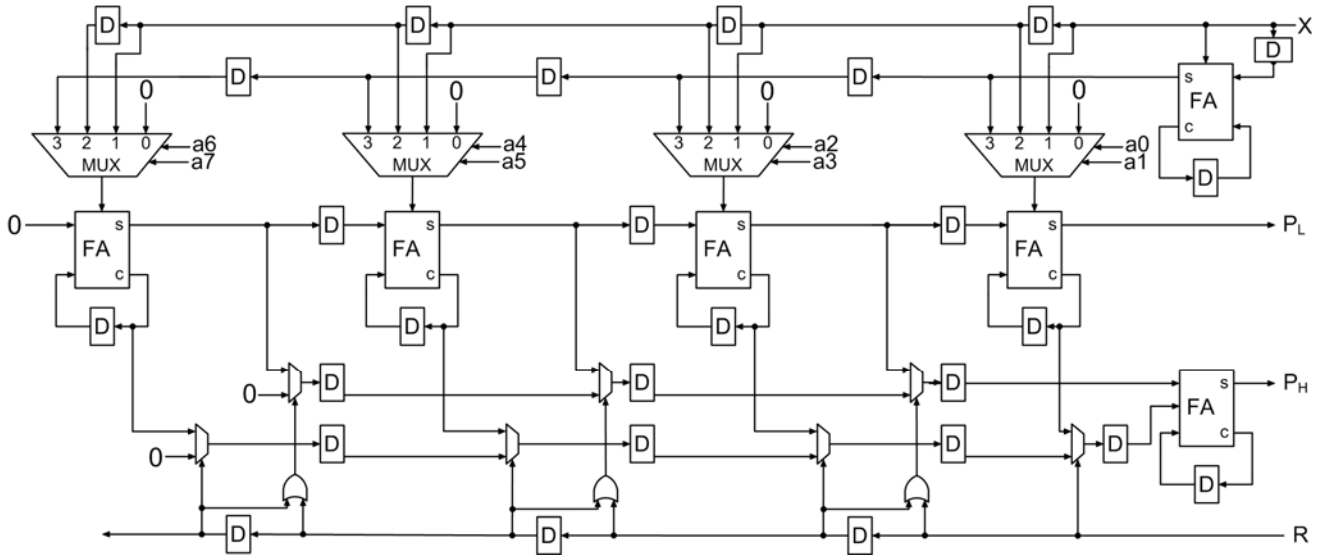

**Fig. 5** Systolic form of the multiplier of Fig. 4.

scheme is proposed. Consider the multiplicand $A$ in normal binary form. The product $A \cdot X$ can be expressed as

$$A \cdot X = \left( \sum_{i=0}^{n-1} a_i 2^i \right) X = \sum_{j=0}^{n/2-1} \left( a_{2j} + 2a_{2j+1} \right) 2^{2j} X$$

$$= \sum_{j=0}^{n/2-1} \left[ \left( a_{2j} + 2a_{2j+1} \right) X \right] 2^{2j} \quad (2)$$

As can be seen, the term in the brackets of eq. 2 can take the values 0, X, 2X, 3X. The term 2X is a one bit shifted form of X, while 3X is generated by the addition of X and 2X in a serial adder. These terms are generated bit after bit, as X enters the circuit. The serial-parallel multiplier based on the generation of the above terms is illustrated in Fig. 4. At each cell, a MUX 4:1 is used to select the proper term, depending on the corresponding pair of A's bits. The circuit in Fig. 4 incorporates the mechanism for downloading the higher order partial results, so as to make the multiplier ready for the next multiplication. The same mechanism was used in the aforementioned multiplier.

Applying the same retiming technique, as in the multiplier of Fig. 3, we obtain the systolic form shown in Fig. 5.

Two zero bits are needed to be interposed between successive input sequences, due to the additional bits needed for the representation of 3X. The signal R, that enables the downloading, is activated concurrently with the second zero bit inserted in X input.

**Table 1** Comparison of serial-parallel multipliers.

| Type of multiplier | Hardware Complexity per bit | Transistors | Operation | E(throughput/area) *100 |
|---|---|---|---|---|
| Proposed in [2] | FA/2+3D+MB/2 | 67 | Systolic 50% | 1.49 |
| Proposed in [3] | FA+AND+11/2D+MUX | 112 | Systolic 100% | 1.79 |
| Proposed multiplier (Fig.3) | FA/2+9/2D+NOR/2+MUX+MB/2 | 99 | Systolic 100% | 2.02 |
| Proposed multiplier (Fig.5) | FA/2+9/2D+5/2MUX+NOR/2 | 100 | Systolic 100% | 2.00 |

FA: Full Adder(22 tr), D: Delay Element(16 tr), MUX: Multiplexer2:1(6 tr), MB: Modified Booth Cell(16 tr) [6]

## 4. COMPARISON

The multiplier proposed in section 2 has the same hardware with that presented in [2], except for the addition of the downloading subcircuit. This addition makes the multiplier available 100% of the time, thus increasing its throughput 100%. Considering the quantity $E = \text{throughput}/\text{area}$ as a degree for circuit performance, and measuring it for the two multipliers, we obtain an increase of 35%, as is shown in table 1.
The second proposed scheme alleviates us from the overhead of encoding, while having the same performance and hardware complexity with that proposed in section 2.
Comparing our multipliers with that proposed in [3], we can see a decrease of 11% in the hardware complexity, while the throughput is the same.
Table 1 summarizes a comparison of the four multipliers.

## 5. CONCLUSION

In this paper we proposed two serial-parallel multipliers with increased throughput and no significant rise in the hardware complexity. The Modified Booth multiplier proposed has reduced hardware and high throughput, but it requires encoding of the parallel factor, as all Modified Booth multipliers do. The second multiplier proposed, goes one step further, having the same performance and complexity, without the need for any encoding.

## REFERENCES

[1] L. Dadda, "On serial-input multipliers for two's complement numbers," IEEE Trans. Comput., vol. 38, pp.1341-1345, Sep. 1989.
[2] C. Wu, "A fast 1-D serial-parallel systolic multiplier," IEEE Trans. Comput., vol. C-36, pp. 1243-1247, Oct. 1987.
[3] K. Pekmestzi, P. Kalivas and N. Mosxopoulos, "Long unsigned systolic serial multipliers and squarers," IEEE Trans. On Circuits Syst. II, vol. 48, no.3, pp. 316-321, Mar. 2001.
[4] O. MacSorley, "High speed arithmetic in binary computers," IRE Proc., vol. 49, pp. 67-91,1961.
[5] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, Oxford University Press, 2000.
[6] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design: A Systems Perspective. Addison-Wesley Publishing Company, 1994.