International Journal Of Engineering Research & Management Technology ISSN: 2348-4039

IJERMT Email: editor@ijermt.org March 2014 Volume 1, Issue 2 Website: ijermt.org

DESIGN OF BOOTH MULTIPLIER AND ALU

Shalee Teotia, Arun Kumar,

SRM University, NCR Campus

Abstract-

This paper presents the design and implementation of signed-unsigned Modified Booth multiplier. The present MBE multiplier and Baugh-wooley multi- pliers performs multiplication on signed numbers only. Therfore, this paper presents the design and implementation of signed-unsigned Modified Booth multiplier. The MBE circuit generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the SUMBE multiplier is obtained. The CSA (carry save adder) tree and the final CLA (carry look ahead adder) used to speed up the multiplier operation. Since signed and unsigned multiplication operation is performed by the same multiplier unit. So the required hardware and chip area reduces and in turn reduces power dissipation and cost.

Keywords:- Modified Booth multiplier, CSA, CLA, partial products, signed-unsigned.

1. Introduction

In digital computing systems multiplication is an arithmetic operation. The multiplication operation consists of producing partial products and then adding these partial products the final product is obtained. Thus the speed of the multiplier depends on the number of partial products and the speed of the adder. So the multipliers have a significant impact on the performance of the entire system. The high speed Booth multipliers and pipelined Booth multipliers are used for digital signal processing applications such as for multimedia and communication systems. High speed DSP computation applications such as Fast Fourier transform (FFT) require additions and multiplications. The conventional modified Booth encoding (MBE) generates an irregular partial product array because of the extra partial product bit at the least significant bit position of each partial product row. Therefore papers [2,3] presents a simple approach to generate a regular partial product array with fewer partial product rows and negligible overhead, thereby lowering the complexity of partial product reduction and reducing the area, delay, and power of MBE multipliers. But the drawback of this multiplier is that it functions only for signed number operands.

The modified-Booth algorithm is extensively used for high speed multipliers circuits. Once when array multipliers were used, the reduced number of generated partial products significantly improved multiplier performance. The Baugh - Wooley algorithm is a different scheme for signed multiplication, but is not so widely adopted because it may be complicated to deploy on irregular reduction trees. Again the Baugh-Wooley algorithm is for only signed number multiplication. The array multipliers and Braun array multipliers operates only on the unsigned numbers. Thus, the requirement of the modern computer system is a dedicated and very high speed multiplier unit that can perform multiplication operation on both signed and unsigned numbers, and this multiplier is called as SUMBE multiplier.

2. Conventional MBE Multipliers

The new MBE recorder was designed according to the following analysis. Table 1 presents the truth table of the new encoding scheme. The Z signal makes output zero to compensate the in correct X2_b and neg signals. Fig. 1 presents the circuit diagram of the encoder and decoder. The encoder generates X1_b, X2_b, Z signals by encoding the three x-signals. The yLSB signal is the LSB of the y signals and the combination with x-signals to determine the Row_LSB and the Neg_cin signals. Similarly,ymsb is combined with x-signals to determine the sign extension signals. Fig. 2 shows an overview of the partial product array for an 8 * 8 multiplier. The sign extension circuitry developed in [22] and [23]. The conventional MBE partial products array has two drawbacks: 1) an additional partial product term at the (n-2)th bit position; 2) poor performance at the LSB-part. To remedy the two drawbacks, the new equations for the Row_LSB and Neg_cin can be written as (1) and (2) respectively.

TABLE 1: Truth Table of MSB Scheme

b_{i+1} b_i b_{i-1}	Value	X1_a	X2_b	Z	Neg
0 0 0	0	1	0	1	0
0 0 1	1	0	1	1	0
0 1 0	1	0	1	0	0
0 1 1	2	1	0	0	0
1 0 0	-2	1	0	0	1
1 0 1	-1	0	1	0	1
1 1 0	-1	0	1	1	1
1 1 1	0	1	0	1	1

$$Row\_LSB = y_{LSB}(x2i + x2i) \quad (1)$$

$$Neg\_cin_i = x_{2i+1}(\overline{x_{2i-1} + x_{2i-1}})\,(\overline{x_{2i-1} + y_{LSB}})\,(\overline{x_{2i} + y_{LSB}}) \quad (2)$$

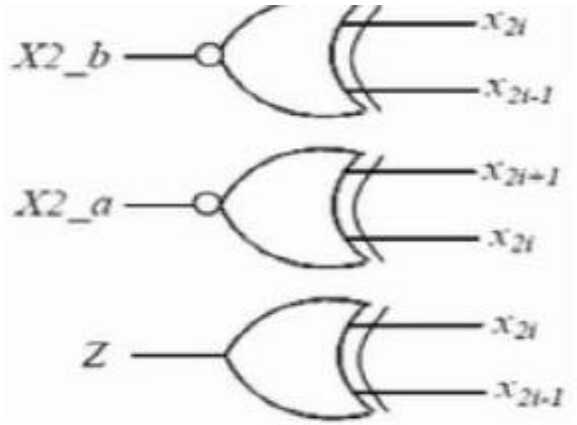


Fig 1(b): Decoder for MBE

b_{i+1} b_i b_{i-1}	Value	X1_a	X2_b	Z	Neg
0 0 0	0	1	0	1	0
0 0 1	1	0	1	1	0
0 1 0	1	0	1	0	0
0 1 1	2	1	0	0	0
1 0 0	-2	1	0	0	1
1 0 1	-1	0	1	0	1
1 1 0	-1	0	1	1	1
1 1 1	0	1	0	1	0

Fig 1(a): Simple encoder

The Fig. 2(a) has widely been adopted in parallel multiplier since it can reduce the number of partial product rows to be added by half, thus reducing the size and enhancing the speed of reduction tree. However, as shown Fig. 1(a), the conventional MBE algorithm generates n/2 + 1 partial product rows rather than n/2 due to the extra partial product bit (neg bit) at the least significant bit position of each partial product row for negative encoding, leading to an irregular partial product array and a complex reduction tree. Therefore, the modified booth multiplier with a regular partial product array [2] produces a very regular partial product array, as shown in Fig. 3. Not only each Neg is shifted to left and replaced by Ci but also the last neg bit is removed this approach reduces the partial product n/2+1 to n/2 by incorporating the last *neg* bit into the sign extension bits of the first partial product row, and almost no overhead is introduced to the partial product array and fewer partial product rows result in a small and fast reduction tree, so that area, delay, and power of MBE multipliers can further be reduced.

3. PROPOSED SUMBE MULTIPLIER

The main goal of this paper is to design and implement 16 x 16 multiplier for signed unsigned numbers using MBE technique. Table2 shows that truth table of MBE scheme. From table2 the

MBE logic and considering other conditions the Boolean expression for one bit partial product generator is given by equation 3.

TABLE2: Truth table of MBE scheme

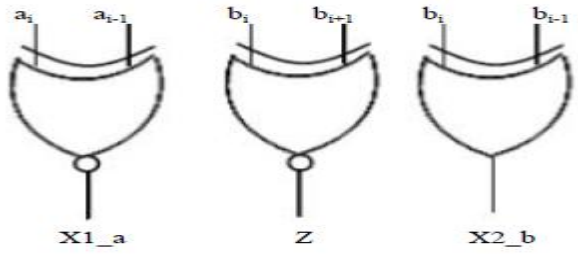


Fig 2: Logic Diagram of MBE

$$P_{ij} = \overline{(\overline{a_i+b_{i+1}}+\overline{b_{i-1}+b_i})\,(\overline{a_{i-1}+b_{i+1}}+\overline{b_i+b_{i+1}}+b_{i-1}+b_i)} \qquad (3)$$

Equation 3 is implemented as shown in Fig. 3. The SUMBE multiplier does not separately consider the encoder and the decoder logic, but instead implemented as a single unit called partial product generator as shown in Fig. 3. The negative partial products are converted into 2's complement by adding a negative (Ni) bit. An expression for negate bit is given by the Boolean equation 4. This equation is implemented as shown in Fig. 4. The required signed extension to convert 2's complement singed multiplier into both signed-unsigned multiplier is given by the equation 3 and 4. For Boolean equations 5 and 6 the corresponding logic diagram is shown in Fig. 5.

$$N_i = b_{i+1}\,\overline{(b_{i-1}\,b_i)} \qquad (4)$$

$$a_{16} = s\_u.a_{15} \qquad (5)$$

$$b_{16} = s\_u.b_{15} \qquad (6)$$

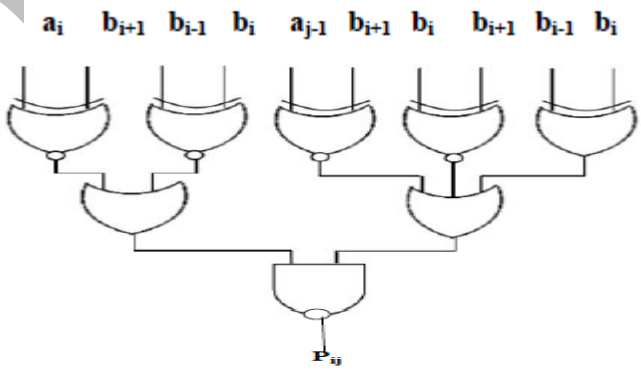


Fig 3: Logic Diagram of i^{th} bit partial product generator

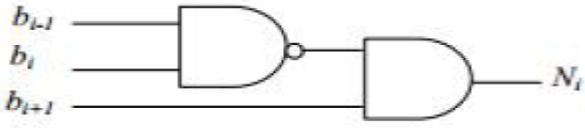


Fig 4: Logic Diagram of negate bit generator

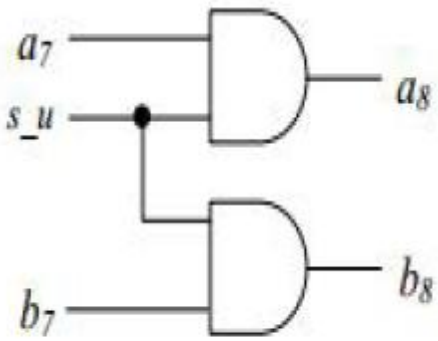


Fig 5: Logic Diagram of sign converter

The working principle of sign extension that converts signed multiplier signed-unsigned multiplier as follows. One bit control signal called signed-unsigned (s_u) bit is used to indicate whether the multiplication operation is signed number or unsigned number. When sign-unsigned (s_u) = 0 it indicates unsigned number multiplication, and when s_u=1, it indicates signed number multiplication. It is required that when the operation is unsigned multiplication the sign the extended bit both multiplication and multiplier should be extended with 0, that is a16 = a17 = b16 = b17 = 0. Is required that when the operation is signed multiplication the sign extended bit depends on whether the multiplication is negative or the multiplier is negative or both the operands are negative. For this when the multiplicand operand is negative and multiplier operand is positive operand is negative and multiplier operand is positive the sign extended bit should be generated are s_u=1, a15 = 1, b15 = 0, a16 = a17 =0, and b16 = b17 =1. Table 3 shows the SUMBE multiplier operation.

Table 3: SUMBE operation

Sign-unsigned	Type of operation
0	Unsigned multiplication
1	Signed multiplication

Fig.6. shows the partial products generated by partial product generator circuit which is shown in Fig. 3. There are 5-partial products which sign extension and negate bit Ni. All the 9-partialproducts are generated in parallel

Fig 6: 16x16 multiplier for signed-unsigned numbers

In Fig. 6 there are 9-partial products namely X1, X2, X3, X4, X5, X6, X7, X8 and X9. These are 9-partial products are added by the carry save adders (CSA) and the final stage is carry look ahead (CLA) adder as shown in Fig.7. Each CSA adder takes three inputs and produce sum and carry in parallel. There are three CSAs, five partial products are added by the CSA tree and finally when there are only two outputs left out then finally CLA adder is used to produce the final result. Assuming each gate delay an unit delay,

including partial product generator circuit delay, then the total through the CSA and CLA is 3+4=7 unit delay. Thus with present very large scale integration (VLSI) the total delay is estimated around 0.7 nano-second and the multiplier operates at giga hertz frequency.

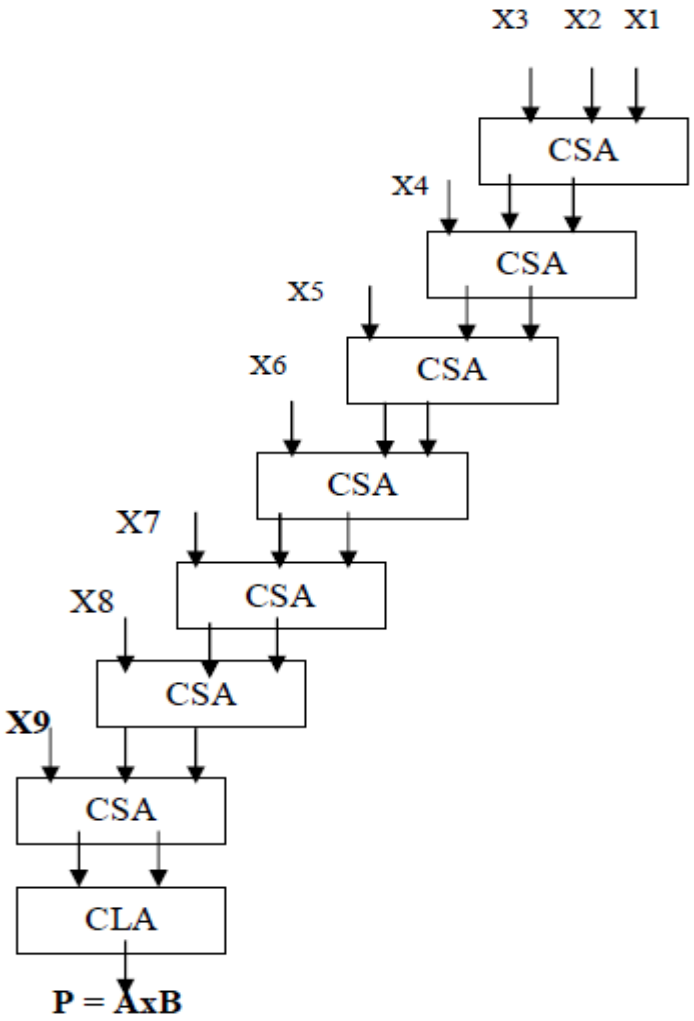


Fig 7: Partial product adder logic

4. SIMULATION RESULT

Verilog code is written to generate the required hardware and to produce the partial product, for CSA adder, and CLA. After the successful compilation the RTL view generated I shown in Fig.8.

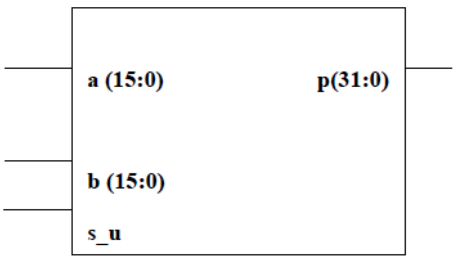


Fig 8: RTL view of 16x16 signed-unsigned multiplier

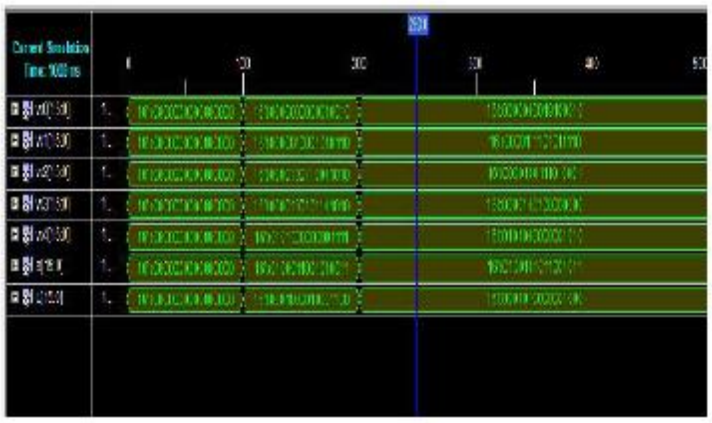
Fig.8 shows the simulation result of signed-unsigned numbers. Fig. 9(a) shows the simulation result of signed – unsigned number in binary. Here when the control signals s_u = 0, the 8-bit operands are considered as unsigned and the product of 0001000000000001×0000000000000011=00000000000000

And when the control signal s_u = 1, the 8-bit operands are considered as signed and the product of 1111111111111111×1111111111111111=00000000000000 01.Fig.9 (b) shows the simulation result of unsigned operands in decimal that is when the control signal s_u = 0, the 8-bit operands are considered as unsigned and the product of 00000000 11111111(255)×0000000011111111(255)=1111111000000

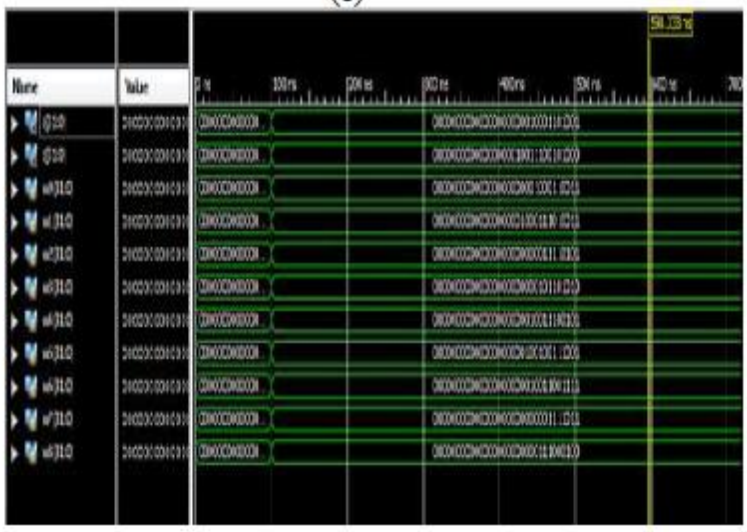
001 (65025), and when the control signals s_u = 1, the 8-bit operands are considered as signed and the product of 11111111 (-1) × 11111111 (-1) = 0000000000000001 (+1). Fig. 9(c) and Fig. 9(d) shows the simulation result of signed-unsigned number in binary and decimal and respectively. When s_u = 0, the 8-bit operands are unsigned and the product of 01111111 (127) × 01111111 (127) = 0111111100000001 (16129). And when the control signal s_u = 1, the 8-bit operands are unsigned and the product of 01111111 (-1) × 00000001 (+1) = 1111111111111111 (- 1).

(a)

(b)



(c)



(d)

Fig 9: Simulation results

V.CONCLUSION

In all multiplication operation product is obtained by adding the partial products. Thus the final speed of the adder circuit depends on the speed of the adder circuit and the number of partial products generated. if radix 8 booth encoding technique is used there are only 3 partial products and for that only one CSA and a CLA is required to produce the final product.

REFERENCES

1. IEEE Recommended Practice for Instrumentation: Specifications for Magnetic Flux Density and Electric Field Strength Meters - 10 Hz to 3 kHz, IEEE Std 1308-1994, 1995. Performance."(n.d.).Federal Highway Administration. Retrieved October 8, 2006, from
2. W. J. Dally, "Interconnect-limited VLSI architecture," in Proceeding of IEEE International Conference on Interconnect Technology, pp.15–17, 1999.
3. J. D. Meindl, ''Beyond Moore's law: The interconnect era,'' Computer Science and Engineering. pp. 20-24, 2003.

4. Ahmed Shebaita and Yehea Ismail, "Multiple threshold voltage design scheme for CMOS Tapered Buffers" IEEE Transactions on circuits and Systems-II, Vol. 55, Page(s): 21 - 25 , January 2008.
5. Ahmed Shebaita and Yehea Ismail "Lower power, lower delay Design scheme for CMOS Tapered Buffers" , Design & Test Workshop (IDT), Page(s): 1 - 5 , 2009
6. N. C. Li, G. L. Haviland and A. A. Tuszynski, "CMOS tapered buffer," IEEE I.S.SC., vol. 25, no. 4, pp. 1005-1008, 1990.
7. B. S. Cherkauer and E. G. Friedman, "A unified design methodology for CMOS tapered buffers," IEEE Transactions on VLSI Syst., vol. 3, no. 1, 1995.
8. K. Roy, S. Mukhopadhyay, and H. Mahmoodi " Leakage current mechanism and leakage reduction techniques in DSM CMOS circuits", IEEE Proceeding., vol 91 No. 2, Feb. 2003
9. HeungJunJeon, Yong-Bin Kim, "Standby Leakage Power Reduction Technique for Nanoscale CMOS VLSI System" IEEE transactions on instrumentation and measurement, VOL. 59, NO. 5, Page(s): 1127 – 1133, MAY 2010
10. Bisdounis, L. "Short-circuit energy dissipation model for sub-100nm CMOS buffers"17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Page(s): 615 - 618, 2010.
11. Kyung Ki Kim; Yong-Bin Kim; Minsu Choi; Park, N. "Leakage minimization technique for nano scale CMOS VLSI" Design &Test of Computers, IEEE Volume: 24 , Issue: 4 , Page(s): 322 – 330, 2007.
12. H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," IEEE J. Solid-State Circuits, vol. SC-19, no. 4, pp.468-473, Aug. 1984.