Design of Low Power and High Speed Configurable Booth Multiplier

Pichingla Kharei[1], Owais Ahmad Shah[2]

[1]M.Tech. ECE, [2]Assistant Professor, Noida International university, Uttar Pradesh, India

Abstract -* Multipliers play a very important role in the design of microprocessor, graphical systems, multimedia systems, DSP system etc .They typically require flexible processing ability, low power consumption, and short design cycle, and have become increasingly popular over the past few years. Many multimedia and DSP applications are highly multiplication intensive so that the performance and power consumption of these systems are dominated by multipliers .Nearly 15 percent of total IC power is consumed by multiplication alone. It is therefore very important to have an efficient design in terms of performance, area, speed of the multiplier, and for the same Booth's multiplication algorithm provides a very fundamental platform for all the new advances made for high end multipliers meant for faster multiplication with higher performance. The computation of the multipliers manipulates two input data to generate many partial products for subsequent addition operations, which in the CMOS circuit design requires many switching activities. Thus, switching activity within the functional unit requires for majority of power consumption and also increases delay. This approach dynamically detects the input range of multipliers and disables the switching operation of non effective ranges .Therefore, minimizing the switching activities can effectively reduce power dissipation and increase the speed of operation without impacting the circuit's operational performance. Here attempt is made to combine configuration, partially guarded computation, and the truncation technique to design a high speed and power efficient configurable BM (CBM). The main concerns are speed, power efficiency and structural flexibility. The proposed multiplier not only perform single 16-b, single 8-b, or twin parallel 8-b multiplication operations but also offer a flexible tradeoff between output accuracy and power consumption to achieve more power savings.

***Keywords:* Booth multiplier (BM), configurable booth multiplier (CBM).**

I. INTRODUCTION

In today's world of ever-increasing computational demands, complex mathematical operation plays a key role in deciding system performance. Multipliers used in DSP and multimedia applications require flexible processing ability, low power consumption and high performance. The demand for high speed processing has been increasing day by day as a result of expanding computer and signal processing applications. Hence modifications are made to their architecture to achieve all these requirements. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low-power VLSI system design. The multiplication operation consists of simply producing partial products and then adding these partial products,the final product is obtained. There are several techniques available [1]–[3] to improve the speed and power efficiency. Guarded evaluation, clock gating, signal gating, truncation etc. reduce the power consumption and increase the speed of multipliers by eliminating spurious computations according to the dynamic range of the input operands. The unnecessary computations in the sign extension part are truncated or removed, hence reducing the power consumption. In [4] the pipeline architecture of high speed modified Booth Multiplier is described in which pipeline technique is used to accelerate the multiplication speed. The speed of multiplier is greatly improved by properly deciding the number of pipeline stages and the positions for the pipeline registers to be inserted. Various other multiplication algorithms such as Booth, modified Booth, Braun and Baugh Wooley have been proposed. The modified booth algorithm reduces the number of partial products to be generated. In [5] techniques that can dynamically adjust two voltage supplies based on the range of the incoming operands and disable ineffective ranges with a zero-detection circuitry were presented to decrease the power consumption of multipliers. The work in [6] separated the arithmetic units into the most and least significant parts and turned off the most significant part when it did not affect the computation results to save power. In [7] a dynamic-range detector to detect the effective range of two operands was developed. The one with the smaller dynamic range is processed to generate booth encoding so that partial products have a greater opportunity to be zero, thereby reducing power consumption maximally. Furthermore, in many multimedia and DSP systems they are frequently truncated due to the fixed register size and bus width inside the hardware. With this characteristic, significant power saving can be achieved by directly omitting the adder cells for computing the least significant bits of the output product, but large truncation errors are introduced. Various error compensation approaches and circuits, which add the estimated compensation carries to the carry inputs of the retained adder cells to reduce the truncation error. In the constant scheme [8], constant error compensation values were pre-computed and added. On the contrary, data-dependent error compensation approaches [9]–[11] were developed to achieve better accuracy than that of the constant scheme. Here, we attempt to combine configuration, partially guarded computation, and the truncation technique to design a power-efficient configurable BM (CBM). Our main concerns are power efficiency and structural flexibility. Most common multimedia and DSP applications are based on 8–16 bit operands, the proposed multiplier is designed to not only perform single 16 bit but also performs single 8 bit, or twin parallel 8 bit multiplication operations. The experimental results demonstrate that the proposed multiplier can provide various configurable characteristics for multimedia and DSP systems and achieve more power savings with slight area overhead.

II. SYSTEM MODEL

Booth multiplier (BM):

Andrew Donald Booth in 1951, devised a multiplication algorithm which was named after his name as Booth‘s Algorithm. Signed multiplication is a vigilant process. Through unsigned multiplication there is no need to take the sign of the number into consideration. Same procedure cannot be applied for signed multiplication due to the reason that the signed numbers are in a 2’s compliment form which would give us inaccurate result if multiplied in an analogous manner to unsigned multiplication [12].Unsigned multipliers cannot be applied to most of the multimedia and DSP applications due to their signed multiplication operation [13]. Thus here Booth’s algorithm comes in rescue. The Booth recoding multiplier scans the three bits at a time to reduce the number of partial products generated [14].Booth’s algorithm conserves the sign of the end result, thus showing better performance in terms of operating speed ,time delay, power dissipation and area. From the basics of Booth Multiplication it can be proved that the addition/subtraction operation can be skipped if the successive bits in the multiplicand are same, thus reducing the delay to a greater extent.

Booth’s Multiplier Algorithm:

Following steps are used for implementing the booth algorithm:-

It makes repeated addition of one of two predetermined values A and S to a product P after which it performs a rightward arithmetic shift on P. Let M and Q be the multiplicand and multiplier respectively, and let x and y represent the number of bits in M and Q. It involves various steps as follows:

(1) Determine the values of A and S, and the initial value of P. All of these numbers should have a length equal to (x + y + 1).

A: Fill the most significant (leftmost) bits with the value of M. Fill the remaining (y + 1) bits with zeros.

S: Fill the most significant bits with the value of (–M) in two’s complement notation. Fill the

remaining (y + 1) bits with zeros.

P: Fill the most significant x bits with zeros. To the right of this, append the value of Q. Fill the least significant bit with a zero.

(2) Determine the two least significant (rightmost) bits of P.

If they are 01, find the value of P + A. Ignore any overflow.

If they are 10, find the value of P + S. Ignore any overflow.

If they are 00, do nothing. Use P directly in the next step.

If they are 11, do nothing. Use P directly in the next step.

3) Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.

(4) Repeat steps 2 and 3 until they have been done y times.

(5) Drop the least significant (rightmost) bit from P. This is the product of M and Q.

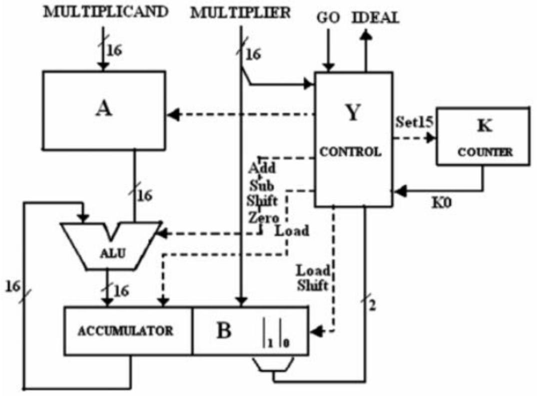
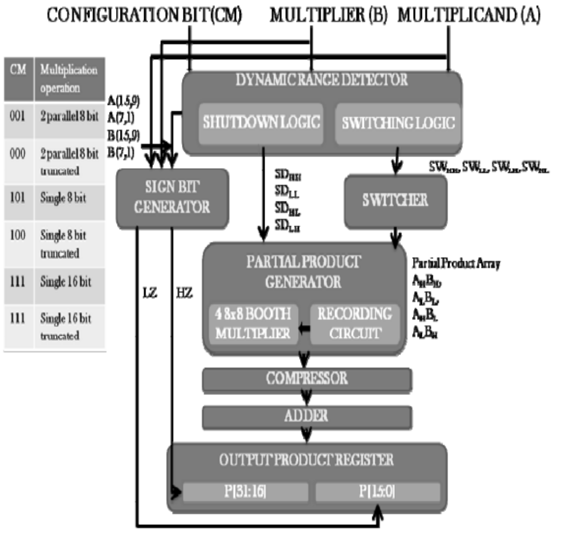


Figure 2.1 Booth Multiplier Architecture

Configurable Booth Multiplier (CBM) design:

Figure 2.2 shows the block diagram of the proposed 16-b CBM. In this section, partially guarded computation and the truncation technique are integrated into the configurable multiplication to construct a 16-b low-power

CBM. The configuration signals are utilized to configure the operation of the proposed multiplier into six modes as shown. When CM [2:1]=11 or 10, the single 16-b or single 8-b multiplication operation is performed. On the other hand, two parallel 8-b multiplication operations that satisfy the high- throughput requirement are carried out if CM [2:1] = 00. The Bit CM [0] decides whether truncation has to be done or not, if it is 0 then truncation will be done through which more power saving and speed is obtained else the output product will not be truncated. Whenever truncation is done error compensation values will be added to maintain output precision.



CM	Multiplication operation
001	2 parallel 8 bit
000	2 parallel 8 bit truncated
101	Single 8 bit
100	Single 8 bit truncated
111	Single 16 bit
111	Single 16 bit truncated

Figure 2.2 Block diagram of the Configurable Booth Multiplier

Dynamic Range Detector (DRD):

The proposed dynamic-range detector (*DRD)* in Figure 2.2 generates switching signals SWLH, SWHH, SWHL and SWLL for each 8-b Booth multiplication to pick the operand that leads more partial products to zero for Booth encoding. In addition to switching signals, *DRD* produces several extra shutdown signals including SDLH, SDHH, SDHL, and SDLL to dynamically disable the redundant computation of the multiplier by forcing unnecessary partial-product bits and carry propagations to zero based on the multiplication mode and the effective range of the input operands.

Switching logic:

If the output of a comparator is 1, it indicates that the input 3-bit group is successive zeros or ones so that it's Booth encoded product will be zero. Finally, each operand is compared to generate the switching signal that is used to determine which operand is a multiplier. In our design, the input operands will be exchanged if the switching signal is one. Aside from increasing the probability of Booth encoded products becoming zero, the switching logic can aid in detecting the length of the sign-extension bits of the input operands and shut down unnecessary computation.

Shutdown logic:

Given the multiplication mode and the effective range of the input operands, the shut down logic shown in Figure 2.3 produces shutdown signals SDLH, SDHH, SDHL and SDLL, to individually shut down AHBH, AHBL, ALBH and ALBL multiplications by setting the signals to be zero to dynamically disable the redundant computation of the multiplier by forcing unnecessary partial-product bits and carry propagations to zero, based on the multiplication mode and the effective range of the input operands.

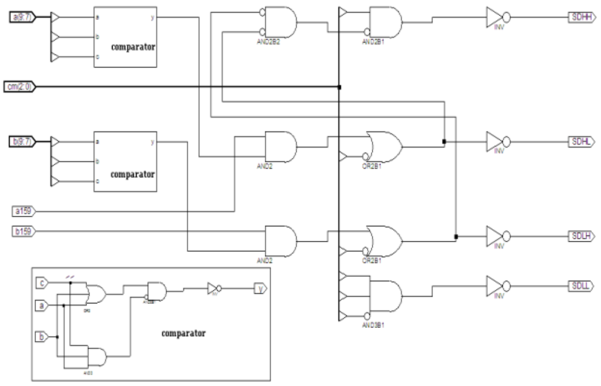


Figure 2.3 Shut down logic of the dynamic-range detector.

Sign Bit Generator:

If one of the input operands is zero, the entire operation of the configurable multiplier can be shut down to obtain more power savings by preventing input registers from loading new data and directly resetting the output registers to zero thereby increasing the speed of operation. Therefore, we develop an SBG as shown in Figure 2.4 to generate an SB, LZ and HZ and shut down the entire multiplier when one of the input operands is zero.

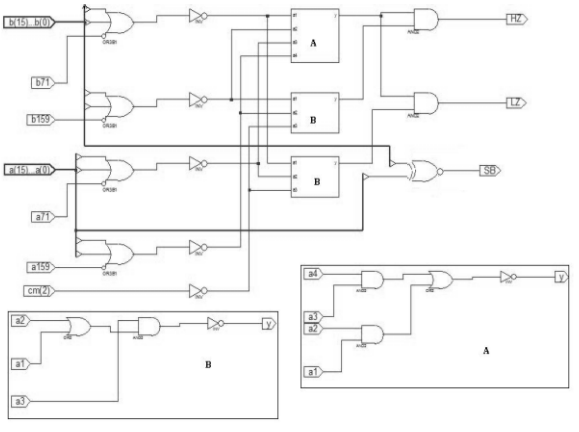


Figure 2.4 Sign Bit Generator

Radix-4 booth encoding:

Radix 2 booth algorithm does not work well when the multiplier has isolated ones. In such case the recorded multiplier has more number of one's when compared to the actual multiplier. So we group 3 bits for finding the recorded multiplier which will help to overcome the above said disadvantage. To multiply A by X, the Radix 4 Booth algorithm starts from grouping X by three bits and encoding into one of {-2, -1, 0, 1, 2}.

Table I Truth Table of Booth Encoding Scheme (Radix 4)

X(i+1)	X(i)	X(i-1)	VALUE
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	2
1	0	0	-2
1	0	1	-1
1	1	0	-1
1	1	1	0

Table I shows the rules to generate the encoded signals by Radix 4 BE scheme. Then with these new multipliers multiplication is done by means of shifting and adding the multiplicand. For negative values 2's compliment is obtained.

Truncation and Error Compensation Circuit:

For fixed-width multiplication operation the least significant bits of the n-bit output product can be disabled to further reduce power consumption and reducing number of adders there by increasing the speed of operation. To incorporate into the proposed multiplier, the partial products of each 8-b Booth multiplication are divided into Higher part (HP), Middle part (MP), and Lower Part (LP). When truncation is performed, the partial products in LP are forced to zero. The partial products in MP are used as inputs to generate approximate carries which are added along with the carry inputs of the adder cells in HP to reduce the truncation error.

16-Bit Multiplication Matrix:

The total Multiplication expression is divided into four sub-expressions AHBH, ALBL, AHBL and ALBH as shown in Figure 2.5. Where AH means A [15:8] and AL means A [7:0] and similarly for operand B. Four independent partial-product arrays are produced by using Radix-4 Booth Encoding approach. The partial products generated for all the individual blocks are grouped as shown in the Figure 2.5 to obtain the final product using adders and compressors.

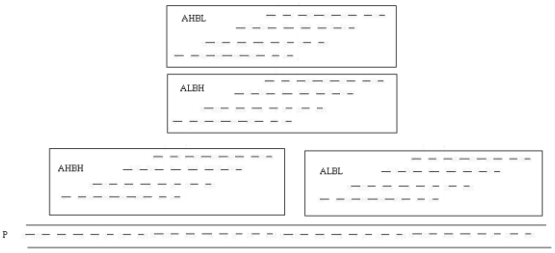


Figure 2.5 Multiplication matrixes for 16-b multiplication

Compressor and Adder:

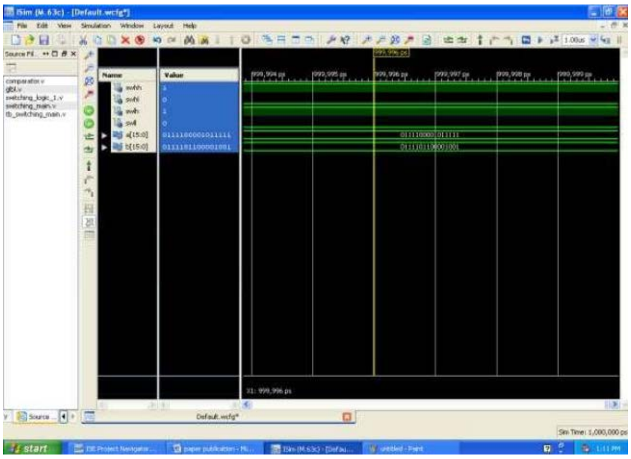
These partial products can be effectively reduced using Dadda tree compression techniques. In the compression algorithm each and every partial product is combined in groups of three and compressed in groups of 2 using full adder which is the 3:2 compressor. This process will be continued until all the partial products along with their carries are compressed. Thus the number of stages and the delay in those stages are reduced effectively using Dadda tree compression technique.

III. EXPERIMENTAL/SIMULATION RESULTS

Radix4 booth encoding for n=8 and n=16 and the proposed CBM for n=16 are designed in Verilog HDL and their simulation are tabulated below and their simulation results were verified. These multipliers were synthesized by using Xilinx ISE 9.2i (and also Synopsys) design compiler with TSMC 90nm CMOS standard cell technology library.

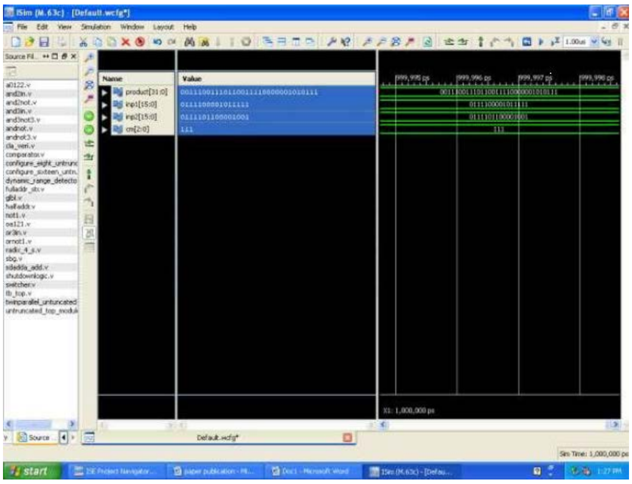
Table 2 Simulation result

Data	Inputs		Product	Sw hh	sw hl	sw lh	sw ll	Sd hh	Sd hl	Sd lh	Sd ll
Data1	01111000 01011111 (30815)	01111011 00001001 (31497)	001110011011001111 0000001010111 (970580055)	1	0	1	0	0	0	0	0
Data2	00000000 01111111 (127)	00000000 00000111 (7)	000000000000000000 000001101111001 (889)	0	1	0	0	1	1	1	0



swhh 1
swhl 0
swlh 1
swll 0
a[15:0] 0111100001011111
b[15:0] 0111101100001001

Figure 3.1



Name	Value
product[31:0]	00111001110110011110000001010111
inp1[15:0]	0111100001011111
inp2[15:0]	0111101100001001
cm[2:0]	111

Figure 3.2

IV. CONCLUSION

A Configurable Booth Multiplier has been designed which provides a flexible arithmetic capacity and a trade off between output precision and power consumption. Moreover, the ineffective circuitry can be efficiently deactivated, thereby reducing power consumption and increasing speed of operation.

V. FUTURE WORK

Further work can be carried out on this project in the power estimation section. Power can be estimated at the gate-level by generating gate-level netlist and also the post layout analysis can be done for this design. Another possible direction can be pursued for higher radix encoding. Another attainable direction will be pursued for higher base encryption.

REFERENCES

[1] J. Choi, J. Jeon, and K. Choi, “Power minimization of function units by partially guarded computation,” in Proc. Int. Symp. Low Power Electron. Des., Jul. 2000, pp. 131–136.

[2] Fayed A and M. A. Bayoumi, “A novel architecture for low-power design of parallel multipliers,” in Proc. IEEE Comput. Soc. Annu Workshop VLSI, Apr. 2001, pp. 149–154.

[3] N.Honarmand and A.Kusha, “Low power minimizetion combinational multipliers using data-driven signal gating,” in Proc. IEEE Int. Conf. Asia-Pacific Circuits Syst., Dec. 2006, pp. 1430– 1433.

[4] Ojin Kim and kyeongsoon Cho “Design of High-speed Modified Booth Multipliers Operating at GHz Ranges” World Academy of Science, Engineering and Technology, 2010.

[5] T.Yamanaka and V.G.Moshnyaga, “Reducing energy of digital multiplier by adjusting voltage supply to multiplicand variation ,” in Proc. 46th IEEE Midwest Symp. Circuits Syst., Dec. 2003, pp. 1423–1426.

[6] K.-H. Chen and Y.-S. Chu, “A spurious-power suppression technique for multimedia/DSP applications,” IEEE Trans. Circuits Sy st. I, Reg. Papers, vol. 56, no. 1, pp. 132–143, Jan. 2009.

[7] N.-Y.Shen and O.T.-C.Chen, “Low-power multiplier by minimizing switching activities of partial products,” in Proc. IEEE Int. Symp. Circuits Syst., May 2002, vol. 4, pp. 93–96.

[8] M.J.Schulte and E. E. Swartzlander Jr., “Truncated multiplication with correction constant,” in Proc. Workshop VLSI Signal Process., Oct.1993, pp. 388–396.

[9] S. J. Jou, M. H. Tsai, and Y. L. Tsao, “Low-error reduced-width Booth multipliers for DSP applications,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 11, pp. 1470–1474, Nov.2003.

[10] K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, “Design of lowerror fixed -width modified booth multiplier,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531, May2004.

[11] T.-B. Juang and S.-F. Hsiao, “Low-power carry-free fixed-width multipliers with low-cost compensation circuit,” IEEE Trans.Circuits Syst.II, Analog Digit. Signal Process. vol. 52, no. 6, pp. 299 –303, Jun. 2005.

[12] Laxman S, Darshan Prabhu R, Mahesh S Shetty ,Mrs. Manjula BM, Dr.Chirag Sharma,FPGA Implementation of Different Multiplier Architectures, International Journal of Emerging.

[13] Shiann-Rong Kuang and Jiun-Ping Wang “Design of Power efficient Configurable Booth Multiplier” Vol.57,No3,March1010

[14] Tam Anh Chu, "Booth Multiplier with Low Power High Performance Input Circuitary", US Patent, 6.393.454 B1,May 21, 2002.