International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Design and Verification of Truncation-Error-Tolerant 8 Bit Signed-Multiplier

Shruti Verma[1], Rakesh Jain[2]

[1]Department of Electronics & Communication, Suresh Gyan Vihar University, Jaipur, India
[2]Assistant Professor, Department of Electronics & Communication, Suresh Gyan Vihar University, Jaipur, India

Abstract: ***Multiplier is one of the essential element for microprocessors, digital signal processors etc. In this project, we had proposed architecture for high speed Truncation Multiplier Algorithm. In modern VLSI technology, the occurrence of all kinds of errors has always to be expected. There are some applications which accept small errors such as multimedia processing. Designing accurate circuit for these applications is waste of area/power. The proposed multiplier outperforms and provides significant improvement in power, area, and delay at the cost of little degrade in accuracy. In this paper, we designed and implemented a new high speed signed booth multiplier. We implemented 8 bit multiplier using the Radix -4 Booth Algorithm. The proposed multiplier reduces the partial product array due to which the area is minimized. This reduction in partial product increases the speed of the multiplier. For addition of partial product we use Ripple Binary Adder. The proposed multiplier is designed and implemented using Verilog HDL in XILINX 9.2 version. Experimental results demonstrate that the proposed 8 bit approximate multiplier is 49.74% faster and 77.83% area efficient than conventional 8 bit signed multiplier.***

Keywords: error-tolerant multiplier, Radix-4 booth multiplier, Ripple binary adder, Verilog, FPGA

1. Introduction

In VLSI number of gates per chip area is constantly increasing, while gate switching energy does not decrease at the same rate, so power dissipation raises producing more heat and to remove heat various cooling techniques are used which are expensive and area taking. Power is the most important parameter for battery operated devices like laptops, cell phones etc. Multiplier is the core component in the processor of most of the digital signal processing applications and other arithmetic oriented applications [1-2]. So if approximate design of multiplier is used than there will be a significant improvement in performance of processor. Some an application in which small amount of error is tolerable are called error tolerant applications. These applications are related to human sense such as vision, smell, hear, touch; these do not require exact result. For these applications designing accurate multiplier is wastage of area/power hence this multiplier is design that takes less area, less power consumption and less delay as compared to parallel multiplier designed at the cost of small degradation in accuracy. Presented multipliers are based on truncating the least significant bits. One can get fixed-width multiplier by directly truncating about half the adder cells of parallel multiplier [3].For example [4] gave a simple method by truncating the half least significant partial product terms and to reduce the error probabilistic constant bias is added to the remaining cells. This design takes 50% area as compared to parallel multiplier but has large error because the constant bias is independent to the truncated bits. [5] Presented the fixed width multiplier to reduce the error depending on the partial product. This design gives less error as compared to previous one but error is still large for small numbers. [6] Presented low power shift and add multiplier for high speed multiplication. Some components of conventional serial multiplier which are more responsible for switching activities are replaced. This design provides better result in terms of area and accuracy at the cost of increased delay.

To evaluate the performance of the approximate design following parameters are used [7].

Overall Error, OE: It is defined as | Rc-Ra|, where Rc denotes the accurate result and Ra denotes the result from approximate multiplier.

Accuracy: It is defined as (1- OE/Rc) x 100%, its value ranges from 0% to 100%.
Minimum acceptance accuracy (MAA): It is the threshold value of the results derived from the proposed multiplier. If they are higher than the minimum acceptance accuracy (MAA), they are called accepted results and are often defined by the customers/designers according to the specific applications.
Acceptance probability (AP): It is the probability that the accuracy of a multiplier is higher than minimum acceptance accuracy (MAA). It can be expressed as AP =P(ACC> MAA) and its value ranges from 0 to 1.

This paper is organized as follows. Section II explains the unsigned approximate multiplier, whereas section III contains the proposed multiplier which is approximate signed multiplier. Section IV describes the implementation results and detailed comparison of Conventional signed multiplier and approximate multiplier. This includes the RTL schematic and stimulation result of proposed work. Section V gives Conclusion of proposed work. In section VI, the Future scope of this paper is written.

2. Previous Work

The algorithm for approximate multiplier is shown in fig.1.First; input operands are divided into two parts: accurate part and approximate part. Left part containing the most significant bits is the accurate part and the right part containing least significant bits is called the approximate part. Since least significant bits contribute less as compared to most significant bits, for most-

significant bits accurate multiplication is applied. The length of each part need not necessary be equal. The multiplication process starts from the starting point in two opposite directions simultaneously as shown in fig.1 ,the two 8-bit input operands, the multiplicand "11100110" (230) and multiplier "00110010"(56),are divided into two equal-sized parts , each part containing 4 input bits. As for the least significant bits of input operands (approximate part), a special mechanism is applied. The carry generation part which is responsible for more power consumption is removed and partial products are not generated. Every bit position from left to right is checked and if both input bits are 1 or one of two input bit is 1, the corresponding product term is 1 and from this bit onwards all the product bits of right side are 1, if both the input bits are 0,the corresponding product bit is 0 and it has no effect on the next right side bits.fig.1 shows the operation of approximate multiplier.

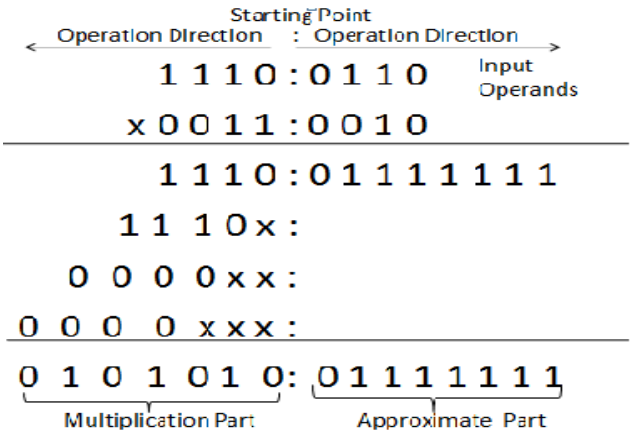

Figure 1: Operation of approximate multiplier

For the most significant bits, input operands fall into accurate part; the operation is conducted as per the normal multiplication operation, from right to left. Its circuit thus designed in the conventional way.

Hardware Implementation

The block diagram of 8-bit approximate multiplier is shown in fig.2. In the proposed design, input A and B are divided into two 4-bit blocks each. The control block contains two, 4-input NOR gates. In the first NOR gate input bits A7-A4 inputs are applied and on the second NOR gate input bits B7-B4 inputs are applied and the outputs of these two NOR gates are applied to the input of NAND gate. The control block is first used for detecting the logic "1" in the MSB position of the inputs, (A7-A4) and (B7-B4). When logic "1" is found, the "ctrl" signal will be activated and the input operands are high enough to operate in: (i) approximate part to give the lower order bits of the final output (P7-P0) and (ii) accurate part to generate the higher order bits of the product (P15-P8). If no logic "1" is detected by the control block, all the most significant bits are 0, the multiplexer selects accurate part to generate the lower order bits of the product (P7-P0).Fig.2 given below shows the hardware for 8-bit approximate multiplier.

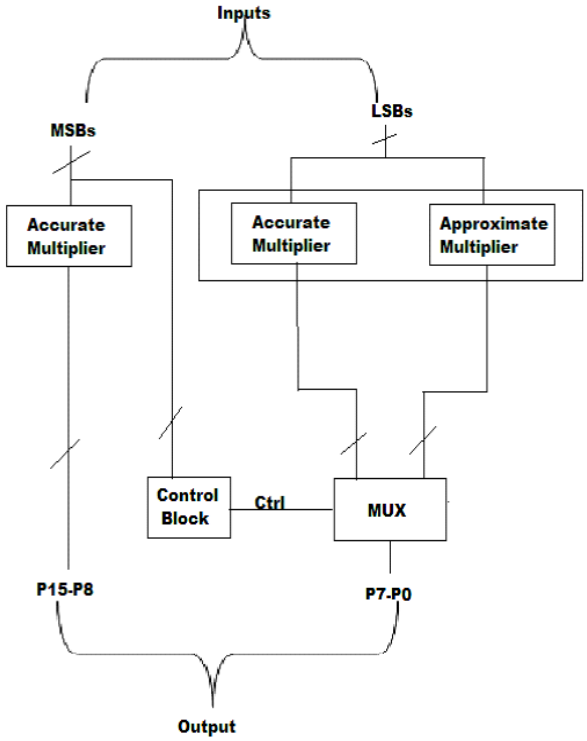

Figure 2: Architecture of Approximate Multiplier

In the accurate part, the standard 4-bit parallel multiplier is used to produce higher order output product terms (P15-P8) as shown in figure 3.

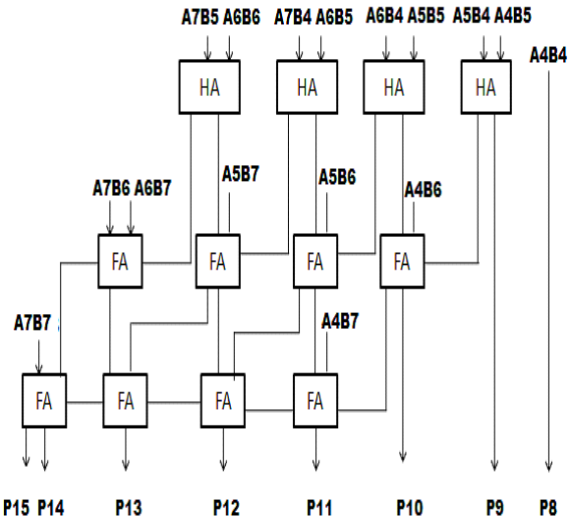

Figure 3: Design of Accurate Part

The approximate part is designed using APGs. For designing 4-bit approximate part 4 APGs are cascaded which generate lower output product (P7-P0).The APG circuit consists of one NOR gate with the inverter at its output.Fig.4 shows the architecture of approximate part using approximate product generators and internal circuit of APG.

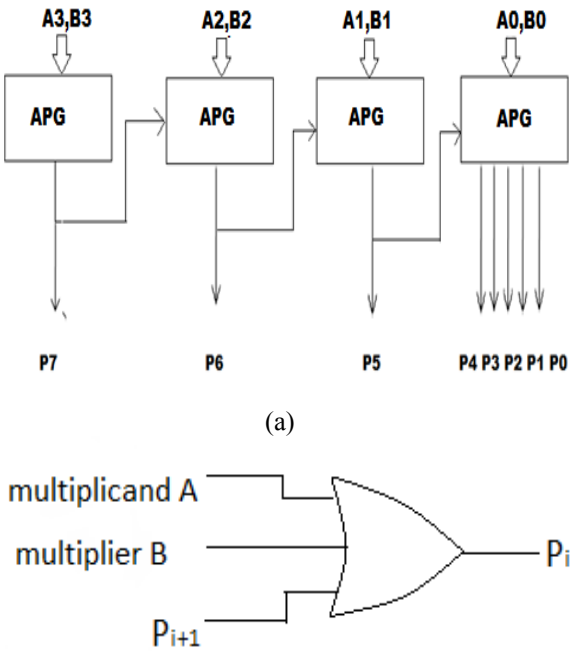

Figure 4: Approximate part (a) Overall architecture (b) schematic diagram of PGC

As shown in fig.4 (a), when inputs A3 and B3 both are 0, the product P7 is 0 and it has no effect on the next product bits but if P7 is 1 then P6 to P0 all the product terms are set to 1 means if Pi+1 is 1, the product term pi will be set to high and in this way, the high signal will be propagated to all the bit positions on its right.

3. Proposed Multiplication Algorithm for Approximate Multiplication

In the previous paper, the 8 bit multiplication was done for unsigned bits whereas, the proposed algorithm is for multiplication of signed 8-bit. The multiplication is performed in two parts: the approximate part and the accurate part. The approximate multiplication will be performed on 4 bits of LSB where as the accurate multiplication will be performed on 4 bits of MSB. As small change in LSB bits will create small change in the value of the number whereas, the small change in MSB bits can create greater change in the value of the original number. Hence accurate multiplication method is compulsory for MSB whereas approximate multiplication can be tolerated for LSB in some applications. This is the reason behind dividing the multiplier and multiplicand in two halves and multiplying by two different methods. The architecture for the proposed multiplier is drawn below.

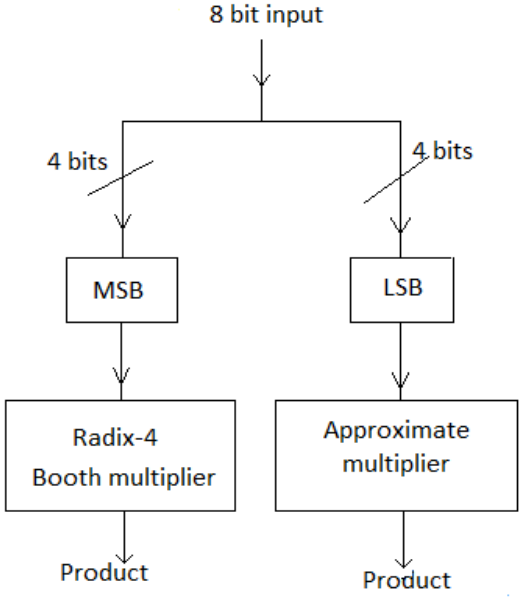

Figure 5: Architecture of Truncation-error-tolerant signed Multiplier i.e. Proposed Multiplier

The approximate part multiplication result is found through performing the OR operation upon multiplier and multiplicand. In the approximate part, multiplication operation is exactly same as explained above in previous paper. The modification has been done on the method of multiplication of 4 bit MSB part. The multiplication method on accurate part used in this paper is based on Booth algorithm. Booth multiplication technique allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is the standard technique used in chip design. In this technique the no. of partial products are lesser and hence provides significant improvements over the "long multiplication" technique.

Hardware Implementation

The hardware implementation for Approximate Multiplier is not explained in proposed work because this part is same as explained in previous work. Accurate multiplication method has been changed from previous work. The hardware for accurate multiplier is drawn below.

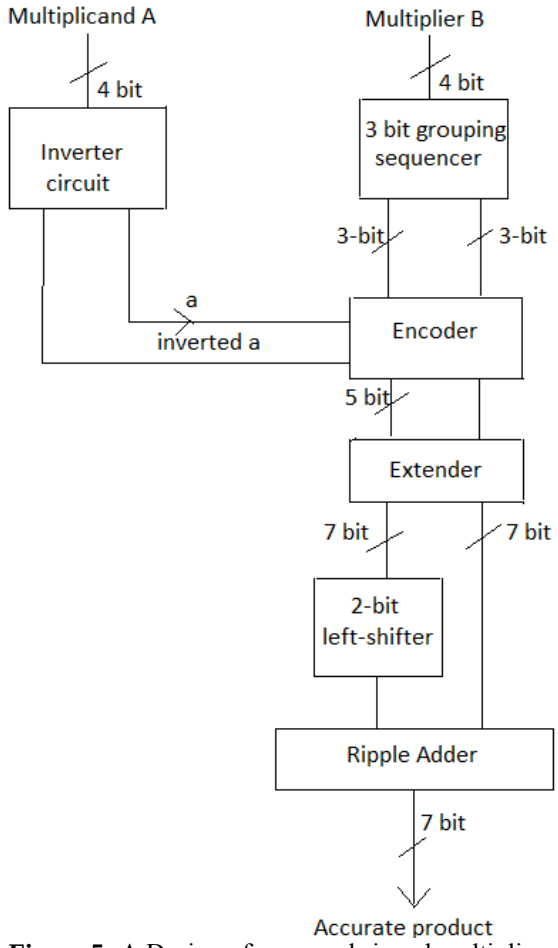

Figure 5: A Design of proposed signed multiplier

Inverter circuit: The purpose of inverter circuit is to convert the sign of a binary number. For implementation of the proposed multiplier, at first the multiplicand A performs sign inversion operation. 2's complement operation is widely used in digital system for bit inversion. 2's compliment method is used for bit inversion in this method first the bit is complimented n then 1 is added to the number.

3 bit grouping sequencer: In this operation we add 0 at the right LSB of multiplier and then grouping them in a sequence, here its grouping is done on 3 bits and the last bit of these three bit also includes in the next grouping of 3 bit, By which we Find vector of three bits of multiplier ({Bn+1, Bn, Bn-1}.

Encoder: This block encodes the bits. Radix-4 Booth algorithm is used as encoding method. Encoding process is based on the following table 1.

RADIX-4 BOOTH ALGORITHM

Radix-4 Booth algorithm examines strings of three bits according to the following algorithm given below:

- The right side of the LSB of the multiplier adds with 0
- Find vector of three bits of multiplier ({ Bn+1 ,Bn , Bn-1})
- Using this vector and following table I find encoded bits (z)

Table 1: Radix-4 Booth Encoding

Multiplier bit (Bn+1)	Multiplier bit (Bn)	Multiplier bit (Bn-1)	Encoded bits (z)
0	0	0	0*A
0	0	1	1*A
0	1	0	1*A
0	1	1	2*A
1	0	0	-2*A
1	0	1	-1*A
1	1	0	-1*A
1	1	1	0*A

- Every next z will be shifted two bits left.
- It will be partial product (pp).

From an operational point of view, multiply by zero means the multiplicand (A) is multiplied by "0".Multiply by "1" means the product still remains the same as the multiplicand value. Multiply by "-1" means that two's complement of the multiplicand value. Multiply by "2" means just left shift of the multiplicand by one place. Multiply by "-2" is to shift left one bit the two's complement of the multiplicand value.

- All partial products will be added which gives the final product.

Extender: In this circuit the input is of 5 bit and gives output of 7 bits. It repeats the MSB of input twice at MSB position and hence 7 bit output is generated.

2-bit left shifter: After encoding and extending, first partial product got from extender is copied then second partial product is 2 bit left shifted with help of this circuit.

Ripple binary adder: This circuit will be used to add the partial products. The result of ripple binary adder will be the final product.

4. Results and Comparison

In this paper we have calculated the performance of approximate multiplier for unsigned bits and radix-4 booth multiplier for signed bits and implemented them. We also have synthesized these multipliers using Xilinx ISE 9.2 and used Verilog as hardware description language. We have designed 8-bit Truncation-Error-Tolerant multiplier, which analyze that delay of Truncation-Error-Tolerant 8-bit signed multiplier is faster and area efficient as comparison to the delay of conventional 8-bit signed multiplier. As we can see that the no. of partial product in conventional multiplier is more than the no. of partial products in the proposed multiplier which results that speed is increased by 49.74% and area is reduced by 77.83%. With this achievement in size and area, the proposed multiplier is area efficient and faster. The RTL/technology schematic of proposed multiplier is shown in figure 7 and the simulation waveform is shown in figure 8.

Table 2: Comparison of Proposed Signed Multiplier with Conventional Multiplier

Name of Multiplier	Time Delay	No. of Slices used
Accurate 8 Bit signed multiplier	11.458 ns	96
Error-truncated 8-bit signed multiplier	5.759 ns	22

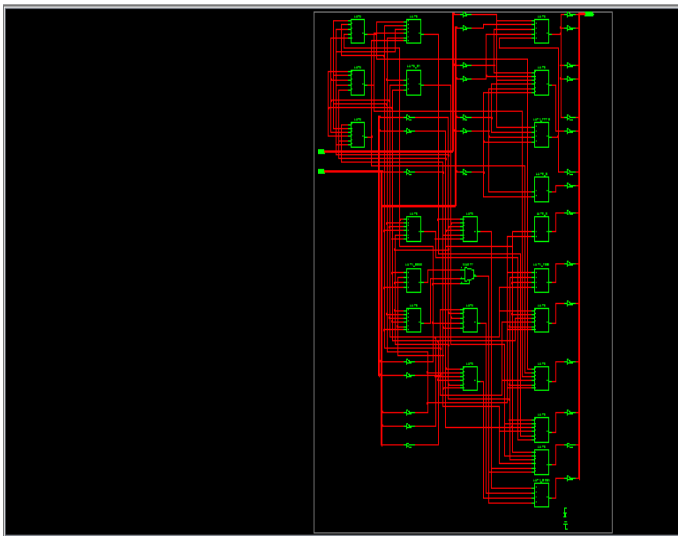

Figure 7: RTL schematic of proposed multiplier

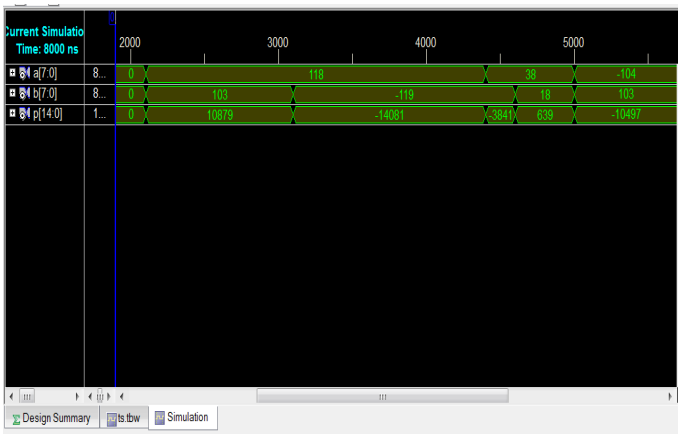

Figure 8: Simulation waveform of proposed multiplier

5. Conclusion

In the proposed design strategy, we have achieved 8 bit truncation-error-tolerant 8-bit signed multiplier with lower area and a shorter critical path than traditional multipliers for Error-Tolerant Applications & high-performance DSP applications. Multiplier circuit is designed into two different parts –Accurate part, which is implemented using Radix 4 booth multiplier and approximate part, which generates small amount of error but this error is acceptable to circuit designer/customer because with small sacrifice of accuracy. The proposed design provides noticeable saving in power and high speed operation. In this paper we have achieved 49.74% faster speed and 77.83% reduced size than conventional signed multiplier.

6. Future Scope

This truncation-error-tolerant multiplication is designed for signed 8 bit multiplication by radix-4 booth algorithm, but it can be implemented in future for higher no. of bits using radix-8 or radix 16. With the use of these algorithms, the area and time delay can be minimized in greater extent. Pipelining technique can be used for further improvement in speed and clock-gating technique can be used for power reduction.

References

[1] S.R Kuang , J.M Jou and Y.L Chen ,"The design of an adaptive on-line binary arithmetic coding chip", IEEE Trans. Circuits syst. 1.vol.45,pp, 693-706, July 1998.
[2] T.J Terrell, Introduction to digital filters. New York: Wiley, 1998.
[3] M. J. Schutle and E. E. Swartzlander Jr, "Truncated multiplication with correction constant," VLSI Signal Processing, vol. 6, pp. 388-396, 1993
[4] S. S. Kidambi, F. EI-Guibaly, Senior Member and A. Antoniou, Fellow IEEE, "Area-Efficient Multipliers for Digital Signal Processing Applications," IEEE Trans. On Circuits and Systems-IL vol. 43, no. 2, Feb 1996.
[5] J.M. Jou, S.R. Kuang, and R.D. Chen, "Design of low-error-fixed-width multipliers for DSP applications, " IEEE Trans, Circuits System If, Analog and Digital Signal Processing, vol. 46, no. 6, pp. 836-842, lun 1999.
[6] K.N Vijeykumar, V.Sumathy, "Design of low-power high-speed error tolerant shift and add multiplier," journal of computer science 7(12) pp. 1839-1845, 2011.
[7] Khaing Yin Kyaw, Wang Ling Goh, and KiatSeng Yeo, "Low-power high-speed multiplier for error-tolerant application", IEEE Trans. Circuits syst. pp, 978-981, 2010