

# VLSI implementation of modified Booth Algorithm

Rasika Nigam, Jagdish Nagar

**Abstract**— Design a Modified Booth Encoding Radix-4 8-bit Algorithm using 0.5um CMOS technology. Booth Algorithm allows for smaller, faster multiplication circuits through encoding the signed numbers to 2's complement, which is also a standard technique used in chip design, and provides significant improvements by reducing the number of partial product to half over "Complex multiplication" techniques. In this paper, we demonstrate an extendable system technique for 8-bit radix-4 MBE algorithm. Encoder, decoder and Carry Look Ahead Adder (CLA) are design in this paper. The modified Booth Encoder circuit generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the AMBE multiplier is obtained.

**Index Terms**— Adder, Encoder, Decoder, Booth Multipliers, SA, FSM.

## I. INTRODUCTION

Signal processing applications typically exhibit high degree of parallelism and are dominated by a few regular kernels of computation such as multiplication, that are responsible for a large fraction of execution time and energy. In such systems, multiplier is a fundamental arithmetic unit. Custom applications for computation in constrained real time systems require area, delay and power optimal architectures. Mission critical applications such as Autonomous robot navigation, avionics and satellite control systems are severely constrained by considerations of payload and on-board power. A variety of processes in such applications ranging from filters for elimination of sensor noise, multi-sensor data fusion, computation of control commands for actuation mechanisms involve multiplication of operands of different word lengths. 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. There has been extensive work on low-power multipliers at technology, physical, circuit and logic levels. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area- speed constraints has been designed with fully parallel. Often a general purpose or application specific processor based architecture does not provide the desired

performance measures for real time applications. Custom architectures are explored for meeting performance constraints through full custom application specific integrated circuits (ASIC) or programmable ASICs which include Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs). The design of a low power high speed Booth multiplier and its implementation on reconfigurable hardware is being proposed. For arithmetic multiplication, various multiplication architectures like array multiplier, Booth multiplier, Wallace tree multiplier and Booth Wallace multiplier have been analyzed. Then it has been found that Booth Wallace multiplier is most efficient among all, giving optimum delay, power and area for multiplication. Low power modified Booth decoder and pipelining techniques have been used to reduce power and delay.

Booth multiplier reduce the number of iteration step to perform multiplication as compare to conventional steps. Booth Algorithm Scans the multiplier operand and spikes chains of this algorithm can. This algorithm can reduce the number of addition required to produce the result compare to conventional multiplication method[5]. With the help of this algorithm reduce the number of partially product generated in multiplication process by using the modified booth algorithm .

The multiplication operations have the fixed-width property. That is, their input data and output results have the same bit width. For example, the  $(2W-1)$  - bit product obtained from  $W$ -bit multiplicand and  $W$ -bit multiplier is quantized to  $W$ -bits by eliminating the  $(W-1)$  least-significant bits.

## II. SHIFT-ADD ARCHITECTURE

A general architecture for shift-add based multiplier for 8 bit operands and radix-4 booth encoding is shown in figure-1. The multiplicand is in register A and the multiplier in register B. A FSM based sequence controller working with a reference clock sequences the operands on the data path for multiplication. The booth encoder converts a signed binary number into a number system where digits take values in the set  $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ . Let us consider the multiplier  $B = b_{n-1}b_{n-2} \dots b_1b_0$ . This can be represented in the twos complement form as  $B = -b_{n-1}2^{n-1} + \sum b_k 2^k$ . This may be rewritten as  $B = \sum (-4b_{2k-1} + 2b_{2k+1} + b_{k+1} + b_{k+2})$  with  $b_{-1} = 0$ . The multiplier is partitioned into sets of 4 bits with 1 overlapping bit. 2 bits with value zero are added one at the beginning and one at the end as  $b_{-1}$  and  $b_8$  respectively. In this form there will be 3 groups in the number, each taking value in the set  $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ . Initially multiplicand is loaded into A register and the temporary register.

Depending upon the encoded value, temporary register is shifted and added with the contents of the product register, which is cleared initially, with proper sign. The sequence

**Manuscript received April 08, 2015.**

Rasika Nigam (M. Tech. Scholar), Embedded System and VLSI Design, Acropolis Institute of Technology and Research, Indore (India).

Jagdish Nagar, Asst. Professor, Acropolis Institute of Technology and Research, Indore (India)

controller selects addition or subtraction of the shifted multiplicand through a multiplexer (MUX). In subsequent stages contents of A register are shifted left 3 times and loaded to temporary register which is again shifted according to the code. The sequence of activities are repeated till all the bits in the multiplier have been encoded and the conditional addition of the partial products have been performed with the product register.

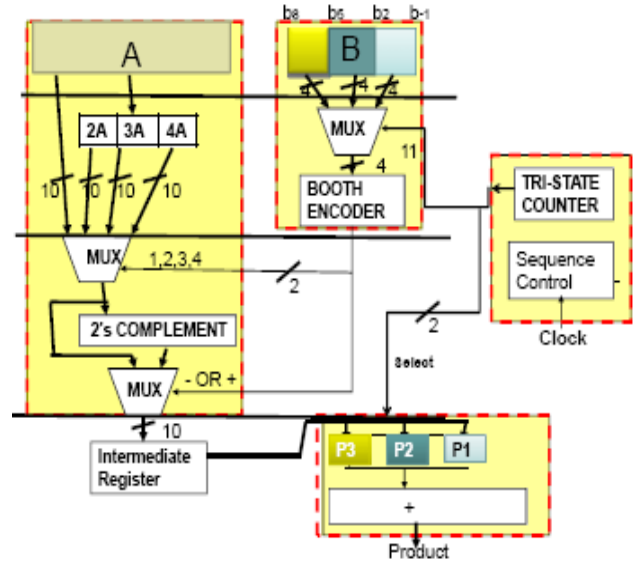
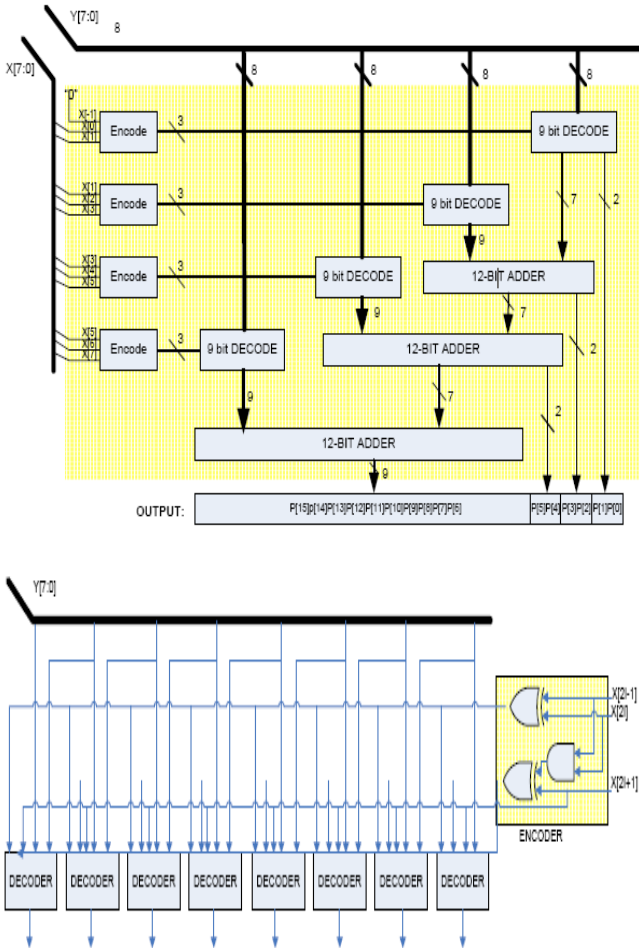


Fig. 2 Architecture for reducing switching activity in the radix-4, 8 bit Multiplier.

A. 12-Bit Adder

We use 3 4-bit CLA units to build up our 12bit adder circuit. The diagram is shown below.

4-bit CLA Adder

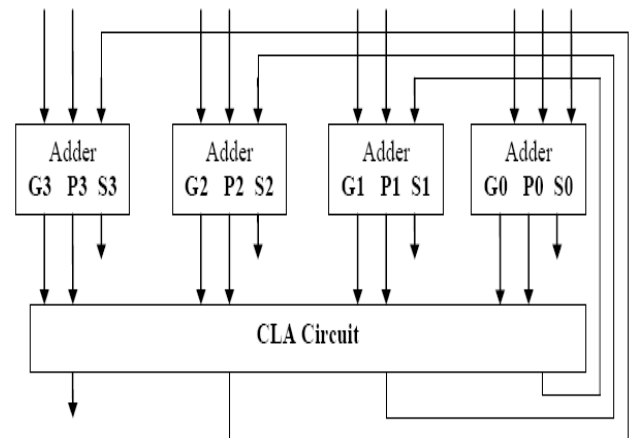


Fig.3 4-Bit CLA Logic equations

III. FSM BASED ARCHITECTURE

The architecture shown in figure-2 uses area trade off for reducing the switching activity in the multiplier. The sequence control block has three states of computation. In state 1, the least significant group of four bits is loaded into the booth encoder to generate the encoded multiplier bits. Depending upon the encoded value, terms from a pre-computed array [A, 2A, 3A, and 4A] are selected through a multiplexer. A second multiplexer is used to select the terms with the proper sign. This value is stored in the partial product register P1 with proper sign extension. In state 2, the next group of four bits of the multiplier is loaded to the encoder circuit. The correct value of the product term is selected through the multiplexer stages as before, and this value is stored in the partial product register P2 with proper zero padding at the right and sign extension on the left. This process is repeated in state 3 with the most significant group of four bits of the multiplier used for encoding and the partial product term being stored in register P3 with proper zero padding and sign extension.

The critical path for system observe that Each adder will give a 10 gate delay time for the 12-bit adder, Including the encoder, decoder and 3 adder, have 38 gate delay

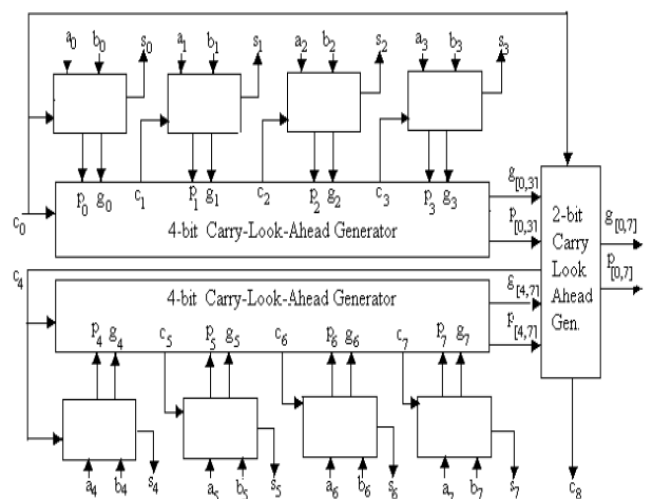
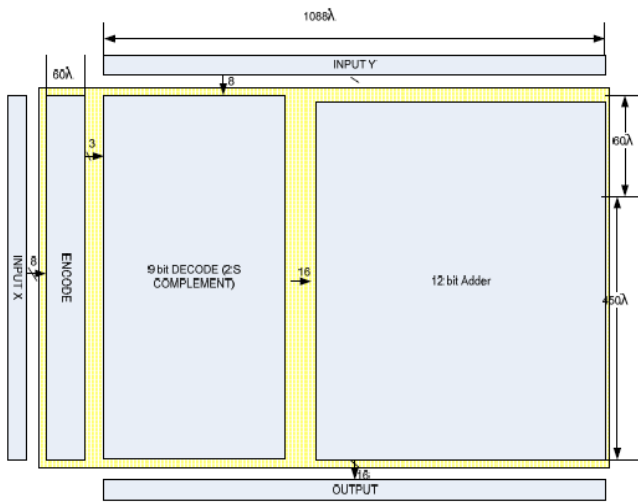


Figure 4 - 8-bit carry look ahead generator (using 2-bit CLA)

IV. DESIGN STRUCTURE OF ALGORITHM

A. Algorithm floor plan



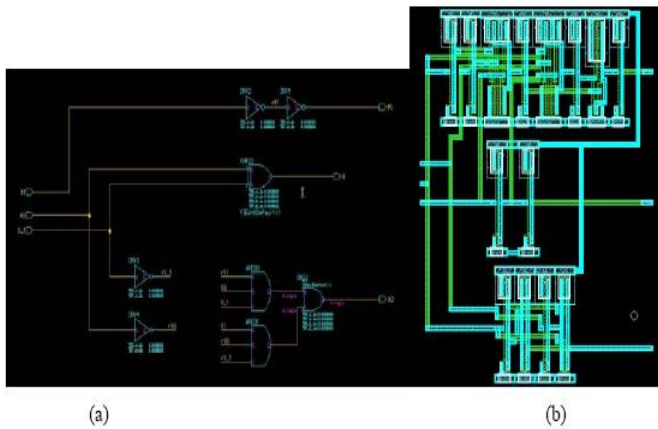
B. Sub Circuit Design: Encoder Block

The encoder block generates the selector signals for each 3 bits of multiplicand. This is the logic for the encoder block:

$$M_i = X_i;$$

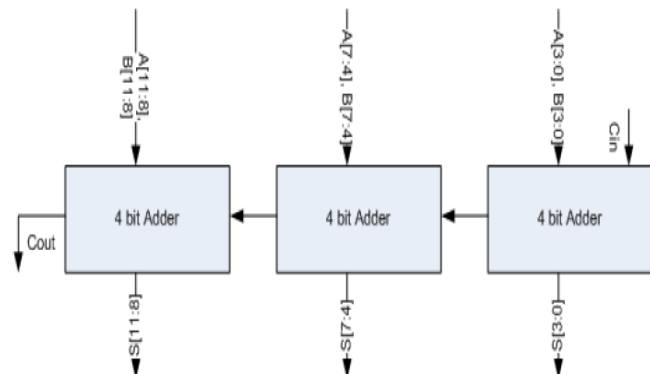
$$X = X_0 \oplus X_{-1};$$

$$X_2 = \bar{X}_1 X_0 X_{-1} + X_1 \bar{X}_0 \bar{X}_{-1}$$

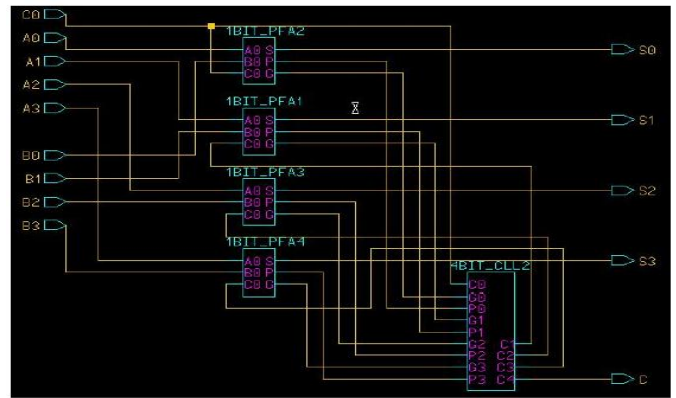


C. 12 Bit Adder circuit

We use three 4bit Carry Look Ahead (CLA) Adder to make up the 12bit adder. The 4bit CLA contains 4 PFA units and one CLL unit, which will increase the computation time.

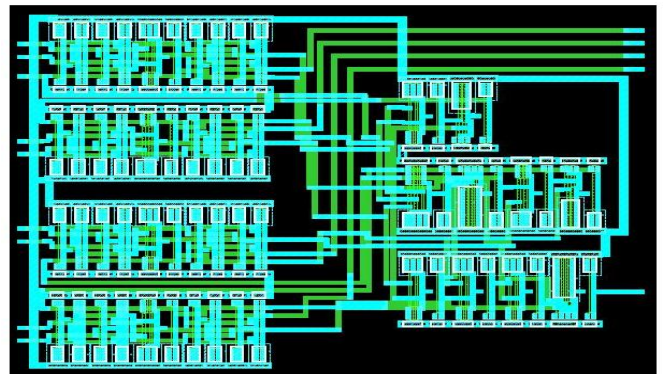


Schematics



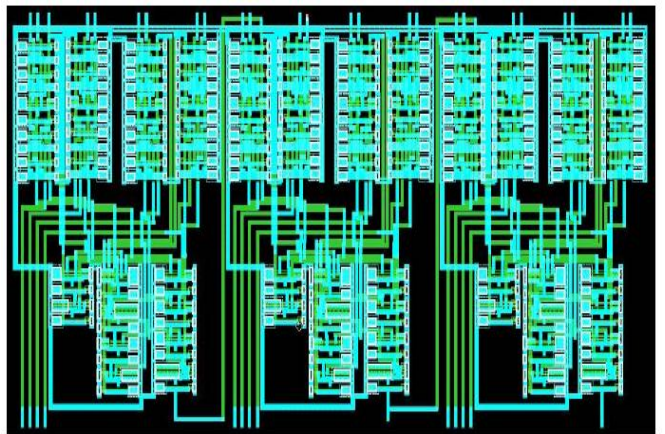
4bit Look Ahead ADDER

Layout 1) for 4 bit adder

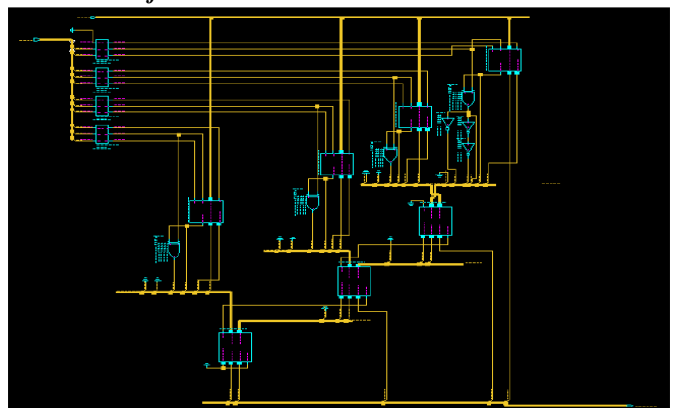


for 4 bit Adder

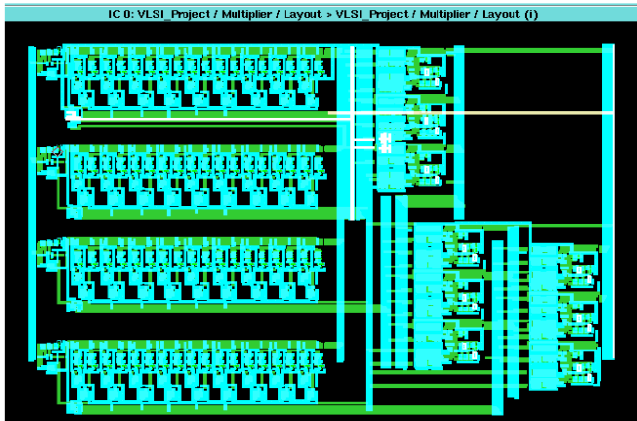
2) For 12 Bit Adder



Schematics of encoder

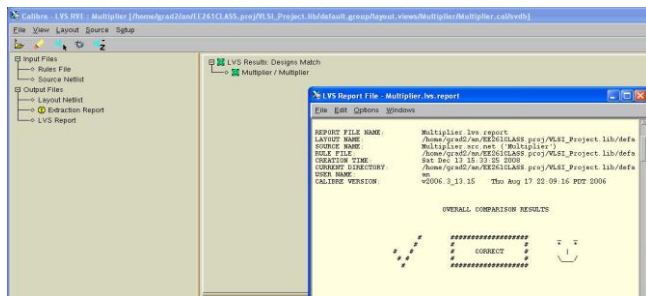
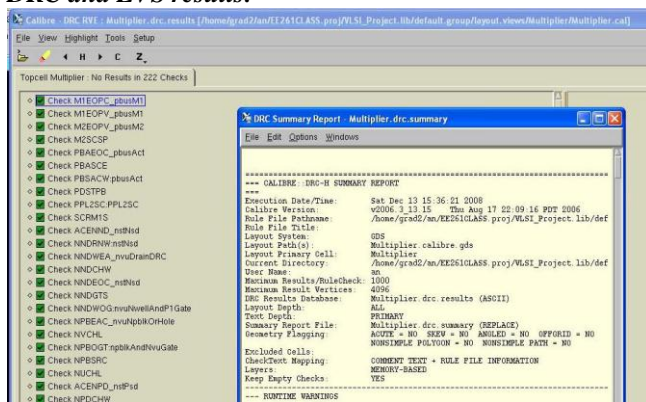


Multiplier Layout



V. SIMULATION AND RESULT

DRC and LVS results:



Digital Simulation Result

The multiplication procedure we wrote the Verilog code for all the blocks and check our multiplier with digital simulation. The table shows our simulation result:

A[7:0]	B[7:0]	Output
00000001	00000001	0000000000000001
00001111	00000010	0000000000011110
00001111	00000100	0000000000111100
00001111	00001111	0000000000110001
10000001	00000100	1111111000000100
10110111	01011010	1110011001010110

VI. CONCLUSION

The implementation result of the multiplication techniques reviewed in this paper. The complete characterization of booth multipliers for 8, 16 and 32 bit operand lengths with 3, 4 and 5 bit encoding. The performance comparison of a conventional shift-add multiplier with a FSM based multiplier and parallel architecture based multiplier has been provided for the above operand lengths and radices of encoding. The results demonstrate the FSM based multiplier architecture is area and power optimal for the operand lengths and radix of encoding considered. The parallel architecture has a better power delay product performance. The implementations of the multipliers in two different FPGA fabrics show that parallel multiplier, architecture performs better in a multiplexer based fabric. The performance of the multipliers under different operating voltages and frequencies has been characterized. The scope for future work includes evaluation of performance of the multipliers using full custom design methodology.

REFERENCES

- [1] "The Modified fermat number transform and its application" in Proc. IEEE Int. Symposium on Circuit and System, Bethlehem, 1990, vol.3, pp. 2365-2368.
- [2] Jagadguru Swami Sri Bharath, Krishna Tirathji, "Vedic Mathematics or Sixteen Simple Sutras from The Vedas", Motilal Banarsidas, Varanasi (India), 1992.
- [3] Jeganathan Srisankarajah, "Secrets of Ancient Maths: Vedic Mathematics", Journal of Indic Studies Foundation, California, 2003 pp. 15-16.
- [4] Parth Mehta, Dhanashri Gawali, "Conventional versus Vedic mathematical method for Hardware implementation of a multiplier"; in Proc. IEEE Int. Conf. Advances in Computing, Control, and Telecommunication Technologies, India, Dec. 2009, pp. 640-642.
- [5] "Low Power Shift and Add Multiplier Design" in Proc. IJCSIT, vol.2, number 3 pp 12-22. 'Improved Carry Select Adder with Reduced Area and Low Power Consumption', International Journal of Computer Applications, Vol.3, No.4, pp. 14-18.