

# Design and Implementation of 64-bit MAC Unit for DSP Applications using verilog HDL

<sup>1</sup> Shaik. Mahaboob Subhani  
Subhanisk491@gmail.com<sup>1</sup>

<sup>2</sup> L.Srinivas Reddy  
lsr@ngi.ac.in<sup>2</sup>

<sup>1</sup> PG Scholar Dept of ECE Nalanda institute of engineering & technology, kantepudi, sattenapalli, Guntur, AP.

<sup>2</sup> Associate professor Dept of ECE Nalanda institute of engineering & technology, kantepudi, sattenapalli, Guntur, AP.

**Abstract:** In this paper, we proposed a new architecture of multiplier -and- accumulator (MAC) for high-speed arithmetic and low power. With the rapid advances in multimedia and communication system, high capacity signal processing are in demand, so High Speed MAC are essential to improve performance of signal processing System. Multiplication occurs frequently in finite impulse response filters, fast Fourier transforms, discrete cosine transforms, convolution, and other important DSP and multimedia kernels. The objective of a good multiplier and accumulator (MAC) is to provide a physically compact, good speed and low power consuming chip. The proposed SPST separates the target designs into two parts, i.e., the most significant part and least significant part (MSP and LSP), and turns off the MSP when it does not affect the computation results to save power. In this paper, we propose a high speed MAC adopting the new SPST implementing approach. This multiplier and accumulator is designed by equipping the Spurious Power Suppression Technique (SPST) on a modified Booth encoder which is controlled by a detection unit using an AND gate. The modified booth encoder will reduce the number of partial products generated by a factor of 2. The SPST adder will avoid the unwanted addition and thus minimize the switching power dissipation.

**Keywords:** Booth encoder, computer arithmetic, digital signal processing, spurious power suppression technique, low power.

## I. INTRODUCTION

One of the accompanying challenges in designing ICs for portable electrical devices is lowering down the power consumption to

prolong the operating time on the basis of given limited energy supply from batteries. Owing to the vigorous development of the wireless infrastructure and the personal electronic devices like video mobile phones, mobile TV sets, PDAs, etc., multimedia and DSP applications have been adopted in wireless environments. Increasing demands of high speed data signal processing motivated the researchers to seek fastest processors. The multiplier and multiplier-and-accumulator (MAC) [1] are the building blocks of the processor and has a great impact on the speed of the processor. MAC is the necessary element of the digital signal and image/audio processing system such as filtering, convolution and inner products hence high speed is crucial to develop for real processing applications. Many researchers have attempted in designing MAC for high computational performance and low power consumption. High throughput MAC is always a key factor to achieve high performance digital signal processing applications for real time signal processing applications. Since the multiplier requires the longest delay among the basic operation in digital system, the critical path is limited by the multiplier. Multiplier basically consists of three operational steps: Booth Encoder, Partial product reduction network (Wallace Tree) and final adder. For high speed multiplication, Modified Booth Algorithm (MBA) [4] is most commonly used, in which partial product is generated from Multiplicand (X) and Multiplier (Y). Booth multiplication allows for the smaller, faster multiplication circuits through encoding the signed bits to 2's complement which is also the standard technique in chip design and provide substantial improvement by reducing the partial products. Although the partial products are

further reduced by using higher radix (4, 8, 16, 32) Booth Encoder which increases complexity and improves the performance[1].

## II. OVER VIEW OF MAC

In this section, basic MAC operation is introduced. A multiplier can be divided into three operational steps. The first is radix-2 Booth encoding in which a partial product is generated from the multiplicand ( $X$ ) and the multiplier ( $Y$ ). The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry. The last is the final addition in which the final multiplication result is produced by adding the sum and the carry. If the process to accumulate the multiplied results is included, a MAC consists of four steps, as shown in Fig. 1, which shows the operational steps explicitly.

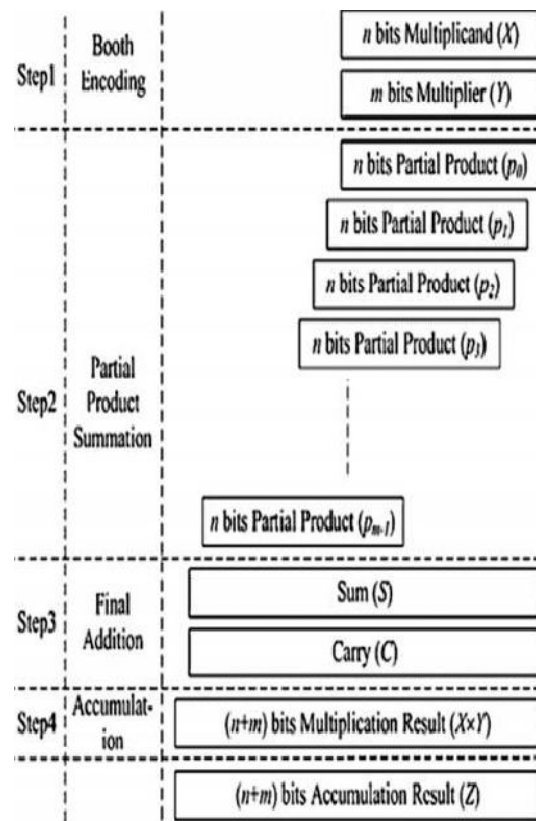


Figure 1. Basic Arithmetic Steps of Multiplication and Accumulation

In this paper, a new architecture for a high-speed MAC is proposed. In this MAC, the computations of multiplication and accumulation are combined and a hybrid-type CSA structure is proposed to reduce the critical path and improve the output rate. It uses MBA algorithm based on 1's complement number system. A modified array structure for the sign bits is used to increase the density of the operands. A carry look-ahead adder (CLA) is inserted in the CSA tree to reduce the number of bits in the final adder. In addition, in order to increase the output rate by optimizing the pipeline efficiency, intermediate calculation results are accumulated in the form of sum and carry instead of the final adder outputs.

A general hardware architecture of this MAC is shown in Fig. 2. It executes the multiplication operation by multiplying the input multiplier and the multiplicand. This is added to the previous multiplication result as the accumulation step.

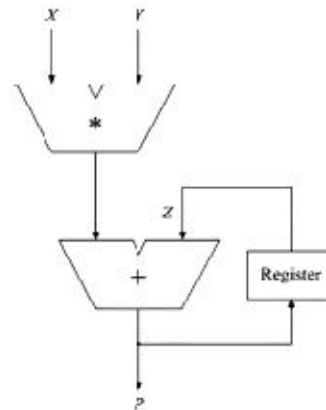
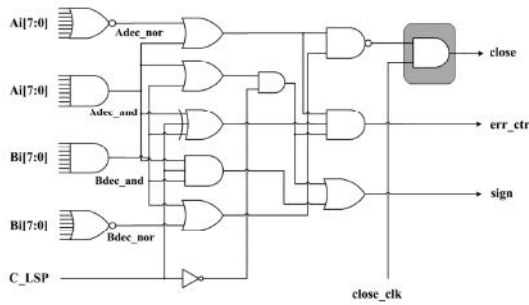


Fig. 2. Hardware architecture of general MAC.

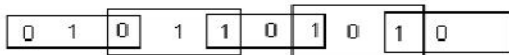
### Modified Booth Encoder

In order to achieve high-speed multiplication, multiplication algorithms using parallel counters, such as the modified Booth algorithm has been proposed, and some multipliers based on the algorithms have been implemented for practical use. This type of multiplier operates much faster than an array multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands.



**Fig 3. Modified Booth Encoder**

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is possible to reduce the number of partial products by half, by using the technique of radix-4 Booth recoding. The basic idea is that, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, we only take every second column, and multiply by  $\pm 1$ ,  $\pm 2$ , or 0, to obtain the same results. The advantage of this method is the halving of the number of partial products. To Booth recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier. Figure 3 shows the grouping of bits from the multiplier term for use in modified booth encoding.



**Fig.3.1 Grouping of bits from the multiplier term**

Each block is decoded to generate the correct partial product. The encoding of the multiplier Y, using the modified booth

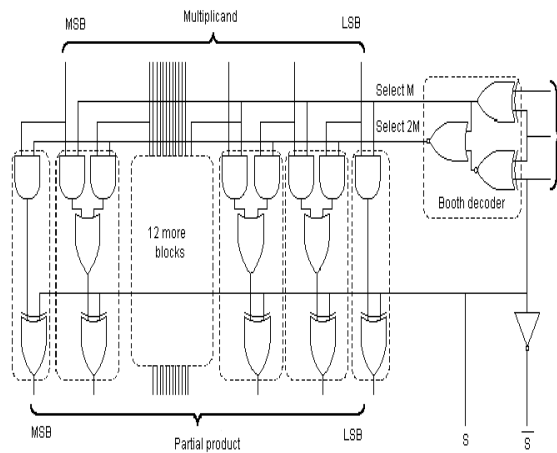
algorithm, generates the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand, X, as illustrated in Table 1

Block	Re - coded digit	Operation on X
000	0	0 X
001	+1	+1 X
010	+1	+1 X
011	+2	+2 X
100	-2	-2 X
101	-1	-1 X
110	-1	-1 X
111	0	0 X

For the partial product generation, we adopt Radix-4 Modified Booth algorithm to reduce the number of partial products for roughly one half. For multiplication of  $2^n$ 's complement numbers, the two-bit encoding using this algorithm scans a triplet of bits. When the multiplier B is divided into groups of two bits, the algorithm is applied to this group of divided bits. Figure 4, shows a computing example of Booth multiplying two numbers "2AC9" and "006A". The shadow denotes that the numbers in this part of Booth multiplication are all zero so that this part of the computations can be neglected. Saving those computations can significantly reduce the power consumption caused by the transient signals. According to the analysis of the multiplication shown in figure 4, we propose the SPST-equipped modified-Booth encoder, which is controlled by a detection unit. The detection unit has one of the two operands as its input to decide whether the Booth encoder

calculates redundant computations. As shown in figure 9. The latches can, respectively, freeze the inputs of MUX-4 to MUX-7 or only those of MUX-6 to MUX-7 when the PP4 to PP7 or the PP6 to PP7 are zero; to reduce the transition power dissipation. Figure 10, shows the booth partial product generation circuit. It includes AND/OR/EX-OR logic.

### III. Partial product generator:



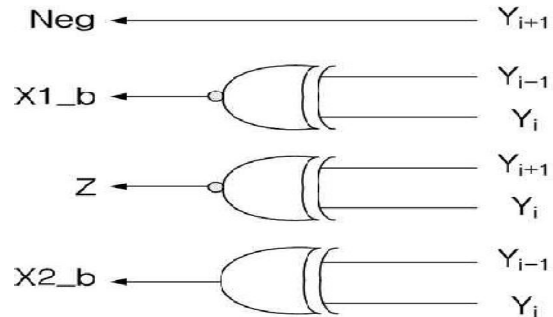
**Fig 4. Booth partial product selector logic**

The multiplication first step generates from A and X a set of bits whose weights sum is the product P. For unsigned multiplication, P most significant bit weight is positive, while in 2's complement it is negative.

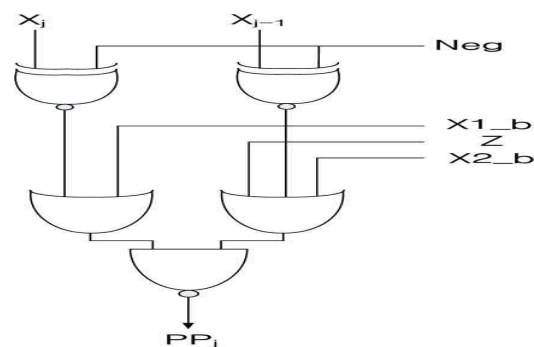
The partial product is generated by doing AND between 'a' and 'b' which are a 4 bit vectors as shown in fig. If we take, four bit multiplier and 4-bit multiplicand we get sixteen partial products in which the first partial product is stored in 'q'. Similarly, the second, third and fourth partial products are stored in 4-bit vector n, x, y.

The multiplication second step reduces the partial products from the preceding step into two numbers while preserving the weighted sum. The sum after product P is the sum of those two numbers. The two numbers will be added during the third step. The "Wallace trees" synthesis follows the Dadda's algorithm, which assures of the minimum counter number. If on

top of that we impose to reduce as late as (or as soon as) possible then the solution is unique. The two binary number to be added during the third step may also be seen a one number in CSA notation (2 bits per digit).



**Fig 5. Booth Encoder**



**Fig 6. Booth Decoder**

### IV. PROPOSED SPST

Besides the explanations presented in our former studies, this paper provides further illustrations of the proposed SPST as described in the following sections.

The SPST uses a detection logic circuit to detect the effective data range of arithmetic units, e.g., adders or multipliers. When a portion of data does not affect the final computing results, the data controlling circuits of the SPST latch this portion to avoid useless data transitions occurring inside the arithmetic units. Besides, there is a data asserting control realized by using registers to further filter out the useless spurious signals of arithmetic unit every time when the latched portion is being turned on. This asserting control brings evident power

reduction. Figure 5 shows the design of low power adder/subtractor with SPST.

Case (1):  
 $(A_{15} A_{14} \dots A_8) = (\_ \_ \dots \_)$ ,  $(B_{15} B_{14} \dots B_8) = (00 \dots 0)$ ,  $C_7 = \_$

$$\begin{array}{r} 128 \quad 0000000010000000 \\ + 64 \quad 0000000001000000 \\ \hline 192 \quad 0000000011000000 \end{array} \rightarrow \begin{array}{r} (-128) \quad 1111111110000000 \\ + 192 \quad 0000000011000000 \\ \hline 64 \quad 0000000001000000 \end{array}$$

Case (2):  
 $(A_{15} A_{14} \dots A_8) = (11 \dots 1)$ ,  
 $(B_{15} B_{14} \dots B_8) = (00 \dots 0)$ ,  $C_7 = 0$

$$\begin{array}{r} (-61) \quad 1111111110000111 \\ + 51 \quad 0000000001100111 \\ \hline (-10) \quad 1111111111101110 \end{array}$$

Case (3):  
 $(A_{15} A_{14} \dots A_8) = (11 \dots 1)$ ,  
 $(B_{15} B_{14} \dots B_8) = (00 \dots 0)$ ,  $C_7 = 1$

$$\begin{array}{r} (-196) \quad 1111111100111100 \\ + 204 \quad 0000000011001100 \\ \hline 8 \quad 0000000000001000 \end{array}$$

Case (4):  
 $(A_{15} A_{14} \dots A_8) = (11 \dots 1)$ ,  
 $(B_{15} B_{14} \dots B_8) = (11 \dots 1)$ ,  $C_7 = 0$

$$\begin{array}{r} (-61) \quad 1111111110000111 \\ + (-205) \quad 1111111100110011 \\ \hline (-266) \quad 1111111011110110 \end{array}$$

Case (5):  
 $(A_{15} A_{14} \dots A_8) = (11 \dots 1)$ ,  
 $(B_{15} B_{14} \dots B_8) = (11 \dots 1)$ ,  $C_7 = 1$

$$\begin{array}{r} (-196) \quad 1111111100111100 \\ + (-52) \quad 1111111111001100 \\ \hline (-248) \quad 1111111100001000 \end{array}$$

$\underbrace{\hspace{10em}}_{\text{sign}} \quad \underbrace{\hspace{10em}}_{\text{carr-ctrl}}$

**Fig 7. Spurious transition cases in multimedia/ DSP processing**

$$A_{MSP} = A[15:8]; B_{MSP} = B[15:8];$$

$$A_{and} = A[15] A[14] A[8];$$

$$B_{and} = B[15] B[14] B[8];$$

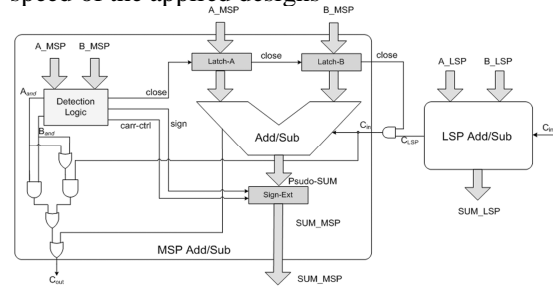
$$A_{nor} = \overline{A[15] + A[14] + A[13] + \dots + A[8]};$$

$$B_{nor} = \overline{B[15] + B[14] + B[13] + \dots + B[8]};$$

$$\text{Close} = \overline{(A_{and} + A_{nor}) \cdot (B_{and} + B_{nor})};$$

The adder /subtractor is divided into two parts, the most significant part (MSP) and the least significant part (LSP). The MSP of the original adder/subtractor is modified to include detection logic circuits, data controlling circuits, sign extension circuits, logics for calculating carry in and carry out signals. The most important part of this study is the design of the control signal asserting circuits, denoted as asserting circuits in Figure 5. Although this asserting circuit brings evident power reduction, it may induce additional delay. There are two implementing approaches for the control signal assertion circuits. The first implementing approach of control signal assertion circuit is using registers. This is illustrated in Figure 6. The three output signals of the detection logic are close, Carr\_ctrl, sign. The three output signals the detection logic unit are given a certain amount of delay before they assert. The delay  $\square$ , used to assert the three output signals,

must be set in a range of  $\square < \square < \square$ , where  $\square$  denotes the data transient period and  $\square$  denotes the earliest required time of all the inputs. This will filter out the glitch signals as well as to keep the computation results correct. The restriction that  $\square < \square$  must be greater than  $\square$  to guarantee the registers from latching the wrong values of control usually decreases the overall speed of the applied designs



**Fig 8. Low-power adder/subtractor design example adopting the proposed SPST.**

**When the detection-logic unit turns off the MSP:**

At this moment, the outputs of the MSP are directly compensated by the SE unit; therefore, the time saved from skipping the computations in the MSP circuits shall cancel out the delay caused by the detection-logic unit.

**When the detection-logic unit turns on the MSP:**

The MSP circuits must wait for the notification of the detection-logic unit to turn on the data latches to let the data in. Hence, the delay caused by the detection-logic unit will contribute to the delay of the whole combinational circuitry, i.e., the 16-bit adder/subtractor in this design example.

**When the detection-logic unit remains its decision:**

No matter whether the last decision is turning on or turning off the MSP, the delay of the detection logic is negligible because the path of the combinational circuitry (i.e., the 16-bit adder/subtractor in this design example) remains the same. From the analysis earlier, we can know that the total delay is affected only when the detection-logic unit turns on the MSP. However, the detection-logic unit should be a speed-oriented design. When the SPST is applied on combinational circuitries, we should

first determine the longest transitions of the interested cross sections of each combinational circuitry, which is a timing characteristic and is also related to the adopted technology. The longest transitions can be obtained from analyzing the timing differences between the earliest arrival and the latest arrival signals of the cross sections of a combinational circuitry. Then, a delay generator similar to the delay line used in the DLL

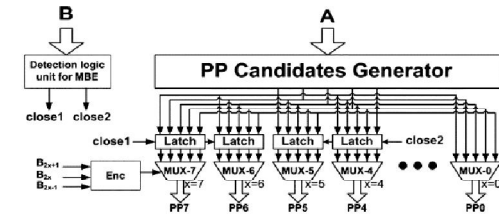


Fig 9.SPST Modified Booth encoder

### Simulation Results of MAC:

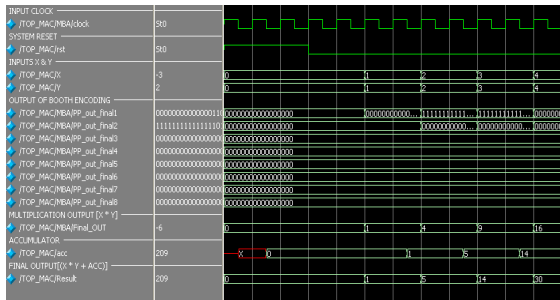


Fig 10. Simulation Waveform of MAC

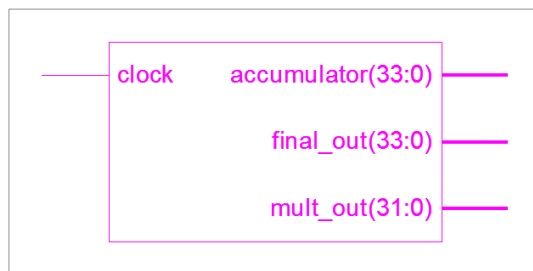


Fig 11.Schematic with Basic Inputs and Output

### CONCLUSIONS

In this project, we propose a high speed low-power multiplier and accumulator (MAC) adopting the new SPST implementing approach. This MAC is designed by equipping the Spurious Power Suppression Technique (SPST)

on a modified Booth encoder which is controlled by a detection unit using an AND gate. The modified booth encoder will reduce the number of partial products generated by a factor of 2. The SPST adder will avoid the unwanted addition and thus minimize the switching power dissipation. The SPST MAC implementation with AND gates have an extremely high flexibility on adjusting the data asserting time. This facilitates the robustness of SPST can attain 30% speed improvement and 22% power reduction in the modified booth encoder. This design can be verified using Modelsim and Xilinx using verilog.

### REFERENCES

- [1] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no. 7, pp. 657–673, Jul. 2003.
- [2] R. Schafer, T. Wiegand, and H. Schwarz, "The emerging H.264/AVC standard," *EBU Technique Review* Jan. 2003 [Online]. Available: [http://www.ebu.ch/trev\\_293-schaefer.pdf](http://www.ebu.ch/trev_293-schaefer.pdf)
- [3] A. Bellaouar and M. I. Elmasry, *Low-Power Digital VLSI Design "Circuits and Systems*. Norwell, MA: Kluwer, 1995.
- [4] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, no. 4, pp. 498–523, Apr. 1995.
- [5] K. K. Parhi, "Approaches to low-power implementations of DSP systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no.10, pp. 1214–1224, Oct. 2001.
- [6] K. Choi, R. Soma, and M. Pedram, "Dynamic voltage and frequency scaling based on workload decomposition," in *Proc. IEEE Int. Symp. Low Power Electron. Des.*, 2004, pp. 174–179.
- [7] J. Choi, J. Jeon, and K. Choi, "Power minimization of functional units by partially guarded computation," in *Proc. IEEE Int. Symp. Low Power Electron. Des.*, 2000, pp. 131–136.

- [8] O. Chen, R. Sheen, and S. Wang, "A low-power adder operating on effective dynamic data ranges," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 4, pp. 435–453, Aug. 2002.
- [9] O. Chen, S. Wang, and Y. W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 3, pp. 418–433, Jun. 2003.
- [10] L. Benini, G. D. Micheli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Glitch power minimization by selective gate freezing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 3, pp. 287–298, Jun. 2000.