Available online at: http://www.ijmtst.com/vol5issue08.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 05, Issue No: 08, August 2019

Implementation of 16-Bit Multiplier-Accumulator Using Modified Booth Algorithm

K Yedukondalu[1] | P Aswani Kumar[2]

[1]PG Scholar, Department of Electronics & Communication Engineering, Eluru College of Engineering & Technology, Eluru, AP, India.
[2]Assistant Professor, Department of Electronics & Communication Engineering, Eluru College of Engineering & Technology, Eluru, AP, India.

To Cite this Article
K Yedukondalu and P Aswani Kumar, "Implementation of 16-Bit Multiplier-Accumulator Using Modified Booth Algorithm", *International Journal for Modern Trends in Science and Technology*, Vol. 05, Issue 08, August 2019, pp.: 66-73.

Article Info
Received on 06-July-2019, Revised on 03-August-2019, Accepted on 08-August-2019, Published on 13-August-2019.

ABSTRACT

With the recent rapid advances in multimedia and communication systems, real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing are increasingly being demanded. The multiplier and multiplier-and-accumulator (MAC) are the essential elements of the digital signal processing such as filtering, convolution, transformations and Inner products. There are different entities that one would like to optimize when designing a VLSI circuit. These entities can often not be optimized simultaneously, only improve one entity at the expense of one or more others The design of an efficient integrated circuit in terms of power, area, and speed simultaneously, has become a very challenging problem. Power dissipation is recognized as a critical parameter in modern the objective of a good multiplier is to provide a physically compact, good speed and low power consuming chip.

This project proposes a new architecture of multiplier-and-accumulator (MAC) for high speed and low-power by adopting the new SPST implementing approach. This multiplier 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. By combining multiplication with accumulation and devising a low power equipped carry save adder (CSA), the performance was improved. In this project we used Model sim for logical verification, and further synthesizing it on Xilinx-ISE tool using target technology and performing placing & routing operation for system.

KEYWORDS: *MAC, SPST, CSA*

I. Introduction

Power dissipation is recognized as a critical parameter in modern VLSI design field. To satisfy MOORE'S law and to produce consumer electronics goods with more backup and less weight, low power VLSI design is necessary. Fast multipliers are essential parts of digital signal processing systems[1]. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation[2]. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them[3].

The basic multiplication principle is to fold i.e. evaluation of partial products and accumulation of the shifted partial products. It is performed by the successive additions of the columns of the shifted partial product matrix. The 'multiplier' is successfully shifted and gates the appropriate bit of the 'multiplicand'. The delayed, gated instance of the multiplicand must all be in the same column of the shifted partial product matrix[4]. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation. To extend the multiplication to both signed and unsigned numbers, a convenient number system would be the representation of numbers in two's complement format[5].

The MAC (Multiplier and Accumulator Unit) is used for image processing and digital signal processing (DSP) in a DSP processor[6]. Algorithm of MAC is Booth's radix-2 algorithm, Modified Booth Multiplier; 17-bit SPST adder improves speed and reduces the power.

In this, when performance of circuits is compared, it is always done in terms of circuit speed, size and power. A good estimation of the circuit's size is to count the total number of gates used. The actual chip size of a circuit also depends on how the gates are placed on the chip – the circuit's layout. Since we do not deal with layout in this report, the only thing we can say about this is that regular circuits are usually smaller than non-regular ones (for the same number of gates), because regularity allows more compact layout[8]. The physical delay of circuits originates from the small delays in single gates, and from the wiring between them. The delay of a wire depends on how long it is. Therefore, it is difficult to model the wiring delay; it requires knowledge about the circuit's layout on the chip. The gate delay, however, can easily be modeled by saying that the output is delayed a constant amount of time from the latest input. What we can say about the wiring delay is that larger circuits have longer wires, and hence more wiring delay. It follows that a circuit with a regular layout usually has shorter wires and hence less wiring delay than a non-regular circuit[10].

II. BASICS OF MULTIPLIER

Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer to itself a specified number of times[11]. A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form a result (product The final product is determined by summation of all the partial-products. Although most people think of multiplication only in base 10, this technique applies equally to any base, including binary. Figure 1 shows the data flow for the basic multiplication technique just described. Each black dot represents a single digit.

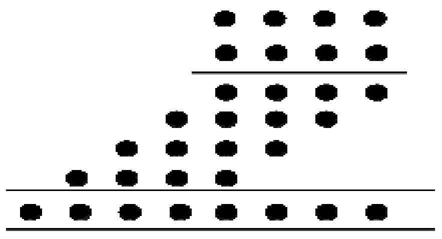


Figure 1.Basic Multiplication

Here, we assume that MSB represent the sign of the digit. The operation of multiplication is rather simple in digital electronics. It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers.

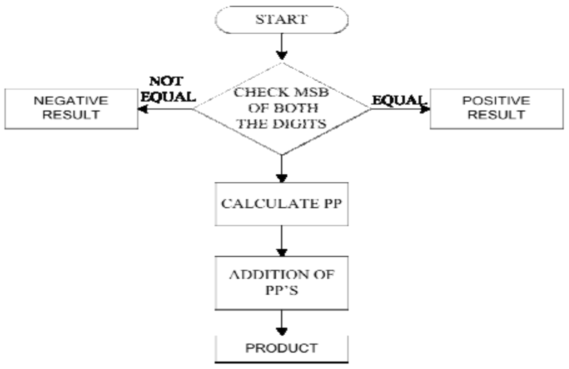


Figure 2: Signed Multiplication Algorithm

III. PROPOSED METHODOLOGY

This paper proposes a new architecture of multiplier-and-accumulator (MAC) for high speed and low-power by adopting the new SPST implementing approach. This multiplier 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. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved.

OBJECTIVES:

- Identify the Architecture From the literature survey
- Model the Architecture into RTL [register transfer level] modeling
- Verify the functionality of the Modeled architecture in MODELSIM®
- Synthesis of the design will be done in Xilinx ISE®
- Generation of Bit map file for Dump into Spartan 3E FPGA
- Program the Bit map file into FPGA.
- Post simulation in ChipScope-pro®.

APPLICATIONS:

- Multimedia and communication systems,
- Real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing.

IV. 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. 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-2 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 ±1, ±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.

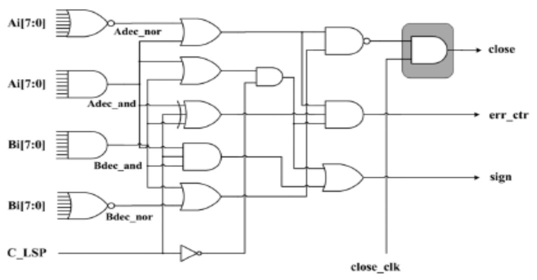
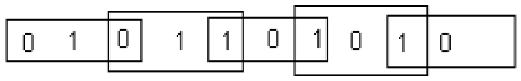


Figure 3:Detection logic circuit using AND gates to assert the control signals

Table 1:Grouping of bits from the multiplier term



0	1	0	1	1	0	1	0	1	0

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.

Table 2:Operations on the multiplicand

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

The PP generator generates five candidates of the partial products, i.e., {-2A,-A, 0, A, 2A}. These are then selected according to the Booth encoding results of the operand B. When the operand besides the Booth encoded one has a small absolute value, there are opportunities to reduce the spurious power dissipated in the compression tree.

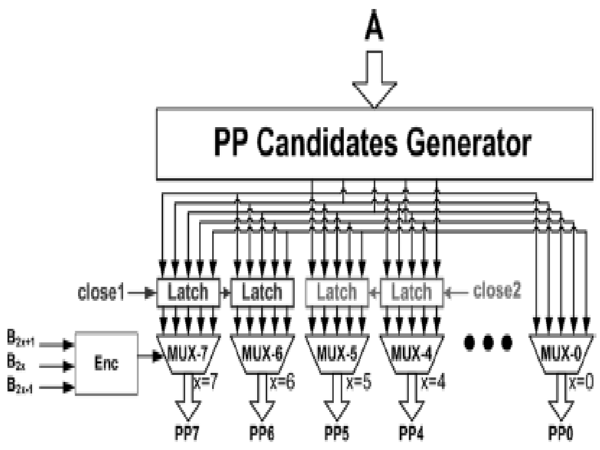


Figure 4:SPST equipped modified Booth encoder

V. MODIFIED BOOTH ENCODER

Multiplication consists of three steps: 1) the first step to generate the partial products; 2) the second step to add the generated partial products until the last two rows are remained; 3) the third step to compute the final multiplication results by adding the last two rows. The modified Booth algorithm reduces the number of partial products by half in the first step. We used the modified Booth encoding (MBE) scheme proposed in. It is known as the most efficient Booth encoding and decoding scheme. To multiply X by Y using the modified Booth algorithm starts from grouping

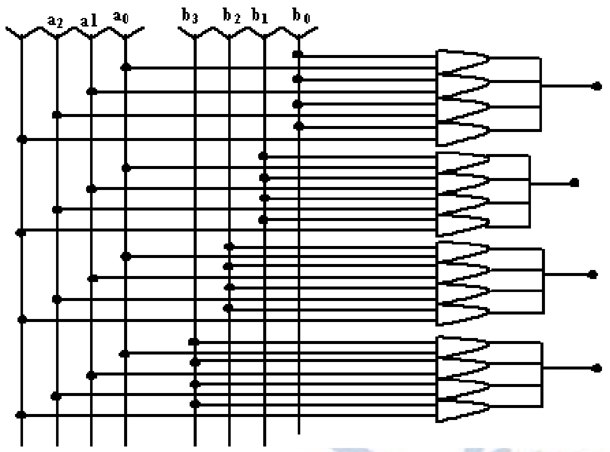


Figure5 :Booth single partial product selector logic

Y by three bits and encoding into one of {-2, -1, 0, 1, 2}. Table I shows the rules to generate the encoded signals by MBE scheme and Fig. 1 (a) shows the corresponding logic diagram. The Booth decoder generates the partial products using the encoded signals as shown in Fig. 1

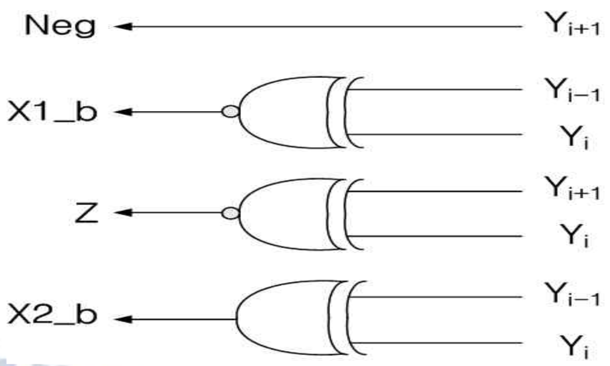


Figure 6:Booth Encoder

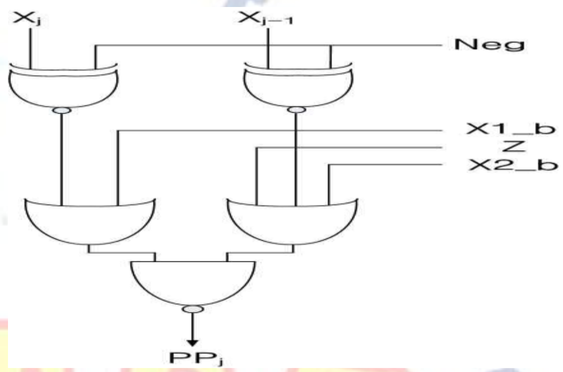


Figure 7 :Booth Decoder

Fig. shows the generated partial products and sign extension scheme of the 8-bit modified Booth multiplier. The partial products generated by the modified Booth algorithm are added in parallel using the Wallace tree until the last two rows are remained. The final multiplication results are generated by adding the last two rows. The carry propagation adder is usually used in this step.

Y_{i+1}	Y_i	Y_{i-1}	Value	X1_b	X2_b	Neg	Z
0	0	0	0	1	0	0	1
0	0	1	1	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	2	1	0	0	0
1	0	0	-2	1	0	1	0
1	0	1	-1	0	1	1	0
1	1	0	-1	0	1	1	1
1	1	1	0	1	0	1	1

Table 3: Truth table for MBE Scheme

A. SPURIOUS POWER SUPPRESSION TECHNIQUE

The former SPST has been discussed in (7) and (8).figure 2 shows the five cases of a 16-bit addition in which the spurious switching activities occur. The 1st case illustrates a transient state in which the spurious transitions of carry signals occur in the MSP though the final result of the MSP are unchanged. The 2nd and the 3rd cases describe the situations of one negative operand adding another positive operand without and with carry from LSP, respectively. Moreover, the 4th and the 5th cases respectively demonstrate the addition of two negative operands without and with carry-in from LSP. In those cases, the results of the MSP are predictable Therefore the computations in the MSP are useless and can be neglected. The data are separated into the Most Significant Part (MSP) and the Least Significant Part (LSP).To know whether the MSP affects the computation results or not. We need a detection logic unit to detect the effective ranges of the inputs. The Boolean logical equations shown below express the behavioral principles of the detection logic unit in the MSP circuits of the SPST-based adder.

where A[m] and B[n] respectively denote the mth bit of the operands A and the nth bit of the operand B, and AMSP and BMSP respectively denote the MSP parts, i.e. the 9th bit to the 16th bit, of the operands A and B. When the bits in AMSP and/or those in BMSP are all ones, the value of A_{and} and/or that of Band respectively become one, while the bits in AMSP and/or those in BMSP are all zeros, the value of

A_{nor}, and/or that of B_{nor} respectively turn into one. Being one of the three outputs of the detection logic unit, close denotes whether the MSP circuits can be neglected or not. When the two input operand can be classified into one of the five classes as shown in figure 8,

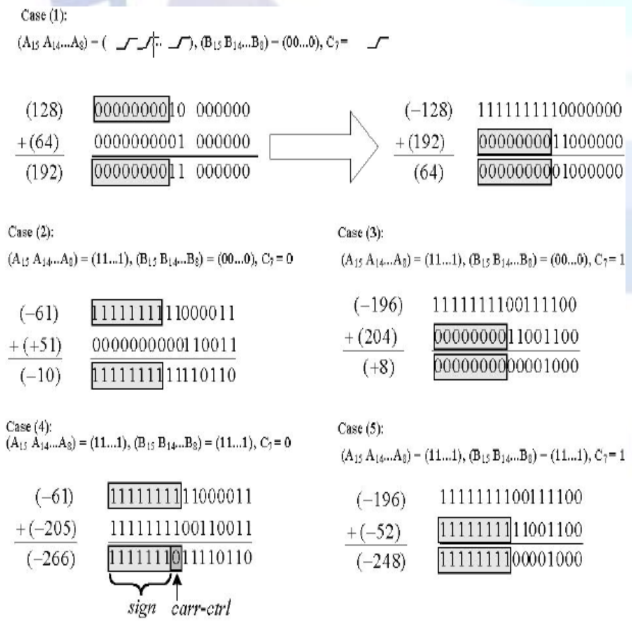


Figure 8 :Spurious transition cases in multimedia/ DSP processing

AMSP = A[15:8]; BMSP = B[15:8] ; Aand = A[15] A[14] A[8]; Band = B[15] B[14] B[8];]

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

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

$$\text{Close} = \overline{(Aand + Anor) \cdot (Band + Bnor)};$$

the value of close becomes zero which indicates that the MSP circuits can be closed. Figure8. also shows that it is necessary to compensate the sign bit of computing results Accordingly, we derive the Karnaugh maps which lead to the Boolean equations (7) and (8) for the Carr_ctrl and the sign signals, respectively. In equation (7) and (8), CLSP denotes the carry propagated from the LSP circuits.

$$\begin{aligned}\text{Carr\_ctrl} = {} & \overline{CLSP} \cdot A_{and} \cdot A_{nor} \cdot B_{and} \cdot \overline{Bnor} + \overline{CLSP} \cdot \\ & A_{and} \cdot \overline{Anor} \cdot \overline{Band} \cdot B_{nor} + C_{LSP} \cdot \\ & \overline{Aand} \cdot A_{nor} \cdot \overline{Band} \cdot B_{nor} + C_{LSP} \cdot A_{and} \cdot \\ & \overline{Anor} \cdot B_{and} \cdot \overline{Bnor};\end{aligned}$$

$$\begin{aligned}= {} & C_{LSP} (\overline{Aand} \cdot B_{and} + A_{and} \cdot \overline{Band}) \cdot (A_{and} \cdot B_{and} \\ & + A_{and} \cdot B_{nor} + A_{nor} \cdot B_{and} + A_{nor} \cdot B_{nor}) + \\ & C_{LSP} \cdot (A_{and} \cdot B_{and} + A_{and} \cdot B_{and} + A_{and} \cdot B_{nor} + \\ & A_{nor} \cdot B_{and} + A_{nor})\end{aligned}$$

$$\begin{aligned}= {} & (C_{LSP} \oplus A_{and} \oplus B_{and}) \cdot (A_{and} + A_{nor}) \cdot \\ & (B_{and} + B_{nor});\end{aligned}$$

$$\text{sign} = \overline{CLSP} \cdot (A_{and} + B_{and}) + C_{LSP} \cdot A_{and} \cdot B_{and};$$

Figure shows a 16-bit adder design example based on the proposed SPST. In this example, the 16-bit adder is divided into MSP and LSP at the place between the 8th bit and the 9th bit. Latches implemented by simple AND gates are used to control the input data of the MSP. When the MSP is necessary, the input data of MSP remain the same as usual, while the MSP is negligible, the input data of the MSP become zeros to avoid switching power consumption. From the derived Boolean equations (1) to (8), the detection logic unit of the SPST is designed as shown in figure 8.

Which can determine whether the input data of MSP should be latched or not. Moreover, we add three 1-bit to control the assertion of the close, sign, and Carr-ctrl signals in order to further decrease the glitch signals occurred in the cascaded circuits which are usually adopted in VLSI architectures designed for video coding.

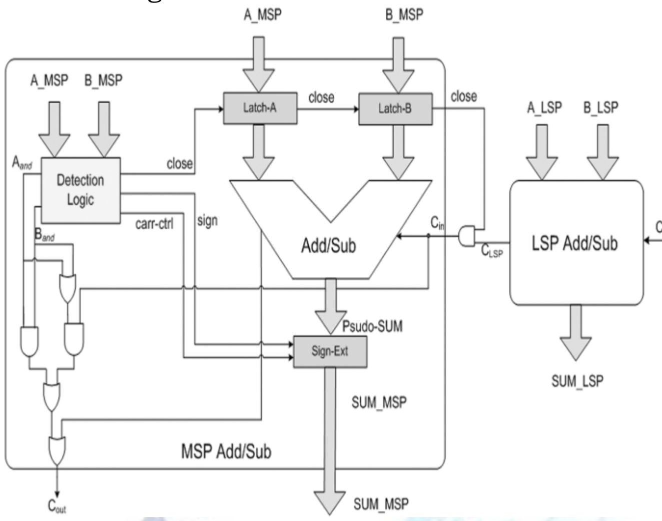


Figure 9:Low-power adder design using SPST

Fig. shows a 16-bit adder design example adopting the proposed SPST. In this example, the 16-bit adder is divided into MSP and LSP between the eighth and the ninth bits. Latches implemented by simple AND gates are used to control the input data of the MSP. When the MSP is necessary, the input data of MSP remain unchanged. However, when the MSP is negligible, the input data of the MSP become zeros to avoid glitching power consumption. The two operands of the MSP enter the detection-logic unit, except the adder, so that the detection-logic unit can decide whether to turn off the MSP or not. Based on the derived Boolean equations (1) to (8), the detection-logic unit of SPST is shown in Fig.8, which can determine whether the input data of MSP should be latched or not. Moreover, we propose the novel glitch-diminishing technique by adding three 1-bit registers to control the assertion of the *close*, *sign*, and *carr-ctrl* signals to further decrease the transient signals occurred in the cascaded circuits which are usually adopted in VLSI architectures designed for multimedia/DSP applications. The timing diagram is shown in Fig. 8 A certain amount of delay is used to assert the *close*, *sign*, and *carr-ctrl* signals after the period of data transition which is achieved by controlling the three 1-bit registers at the outputs of the detection-logic unit.

Hence, the transients of the detection-logic unit can be filtered out; thus, the data latches shown in Fig can prevent the glitch signals from flowing into the MSP with tiny cost. The data transient time and the earliest required time of all the inputs are also illustrated. The delay should be set in the range of, which is shown as the shadow area in Fig, to filter out the glitch signals as well as to keep the computation results correct. Based on Figs. 5 and 6, the timing issue of the SPST is analyzed as follows.

VI. RESULTS AND DISCUSSION

6.1 SYNTHESIS RESULTS

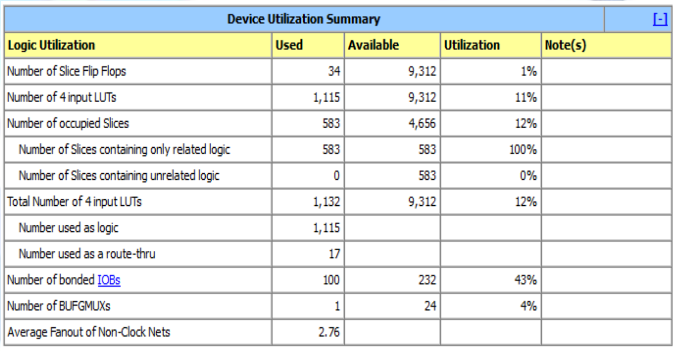
The developed MAC design is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level netlist mapped to a specific technology library. This MAC design can be synthesized on the family of Spartan 3E.Here in this Spartan 3E family, many different devices were available in the Xilinx ISE tool. In order to synthesis this design the device named as "XC3S500E" has been chosen and the package as "FG320" with the device speed such as "-5". The design of MAC is synthesized and its results were analyzed as follows.

6.2 DEVICE UTILIZATION SUMMARY

This device utilization includes the following.

- Logic Utilization
- Logic Distribution
- Total Gate count for the Design

Table 4:Device Utilization Summary



Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	34	9,312	1%	
Number of 4 input LUTs	1,115	9,312	11%	
Number of occupied Slices	583	4,656	12%	
Number of Slices containing only related logic	583	583	100%	
Number of Slices containing unrelated logic	0	583	0%	
Total Number of 4 input LUTs	1,132	9,312	12%	
Number used as logic	1,115			
Number used as a route-thru	17			
Number of bonded IOBs	100	232	43%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	2.76			

Table 5:Synthesis report for array MAC and SPST MAC

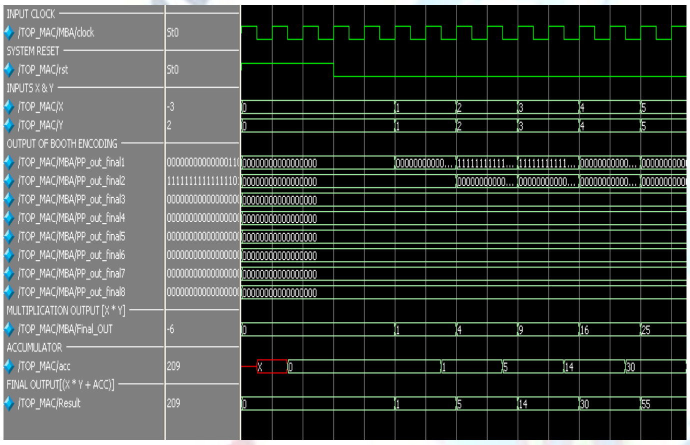
Device parameters	Array MAC	SPST MAC
Number of Slice Flip Flops	34 out of 9312	34 out of 9312
Number of 4 input LUTs	636 out of 9312	1108 out of 9312
Number of IOs	100	100
Number of bonded IOBs	100 out of 232	100 out of 232
Number of GCLKs	1 out of 24	1 out of 24

Table 6:Comparison of SPST adder MAC and array MAC

Multiplier Type	Array Multiplier and Accumulator	MBA with SPST adder MAC
Vendor	Xilinx	Xilinx
Device and family	xc3s500e-5fg320	xc3s500e-5fg320
Estimated delay(ns)	217.8	36.349
Power consumption(mw)	154	144

The device utilization summery is shown above in which its gives the details of number of devices used from the available devices and also represented in %. Hence as the result of the synthesis process, the device utilization in the used device and package is shown above.

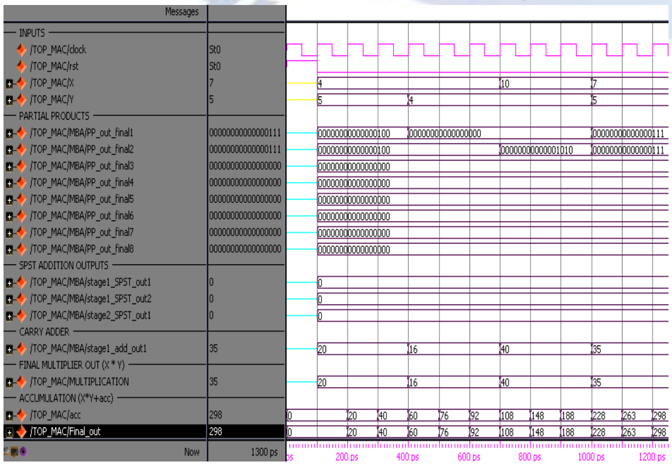
6.3 SIMULATION RESULTS OF MAC



```
INPUT CLOCK
/TOP_MAC/MBA/clock              St0
SYSTEM RESET
/TOP_MAC/rst                    St0
INPUTS X & Y
/TOP_MAC/X                      -3     0  1  2  3  4  5
/TOP_MAC/Y                      2      0  1  2  3  4  5
OUTPUT OF BOOTH ENCODING
/TOP_MAC/MBA/PP_out_final1      000000000000000110  00000000000000000  00000000000...  11111111111...  11111111111...  00000000000...  0000000000
/TOP_MAC/MBA/PP_out_final2      11111111111111110   00000000000000000  00000000000...  00000000000...  00000000000...  0000000000
/TOP_MAC/MBA/PP_out_final3      00000000000000000   00000000000000000
/TOP_MAC/MBA/PP_out_final4      00000000000000000   00000000000000000
/TOP_MAC/MBA/PP_out_final5      00000000000000000   00000000000000000
/TOP_MAC/MBA/PP_out_final6      00000000000000000   00000000000000000
/TOP_MAC/MBA/PP_out_final7      00000000000000000   00000000000000000
/TOP_MAC/MBA/PP_out_final8      00000000000000000   00000000000000000
MULTIPLICATION OUTPUT [X * Y]
/TOP_MAC/MBA/Final_OUT          -6     0  1  4  9  16  25
ACCUMULATOR
/TOP_MAC/acc                    209    X  0  1  5  14  30
FINAL OUTPUT[(X * Y + ACC)]
/TOP_MAC/Result                 209    0  1  5  14  30  55
```

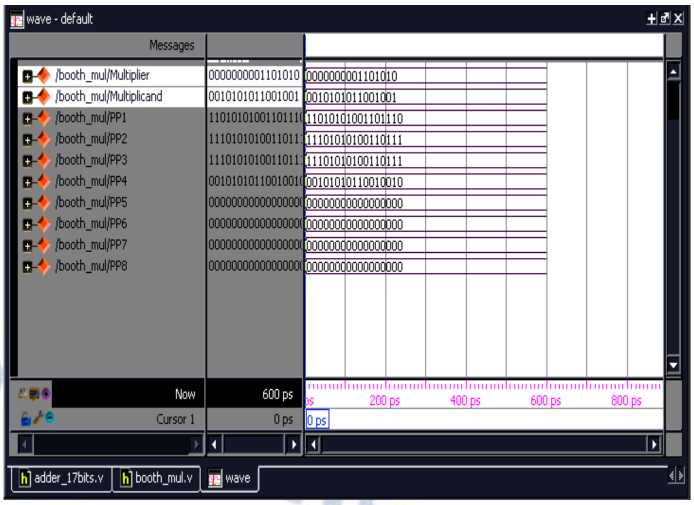
Figure 10 : Verification of Top MAC with inputs

The figures show the verification of SPST MAC with carry save adder in addition stage which is consuming less power and less combinational path delay. The design is verified on Modelsim with given inputs. Figure 26 gives the power analysis of design which conforms that the quiescent power is 144 mw which is 6% lower compared to array MAC.



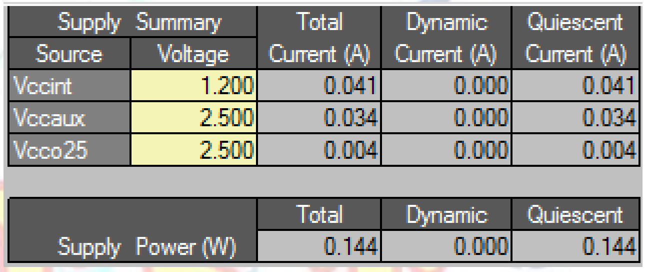
```
Messages
INPUTS
/TOP_MAC/clock                       St0
/TOP_MAC/rst                         St0
/TOP_MAC/X                           7                   4  10  7
/TOP_MAC/Y                           5                   5  4  5
PARTIAL PRODUCTS
/TOP_MAC/MBA/PP_out_final1           0000000000000111    0000000000000100  00000000000000000  0000000000000111
/TOP_MAC/MBA/PP_out_final2           0000000000000111    0000000000000100  0000000000001010  0000000000000111
/TOP_MAC/MBA/PP_out_final3           0000000000000000    00000000000000000
/TOP_MAC/MBA/PP_out_final4           0000000000000000    00000000000000000
/TOP_MAC/MBA/PP_out_final5           0000000000000000    00000000000000000
/TOP_MAC/MBA/PP_out_final6           0000000000000000    00000000000000000
/TOP_MAC/MBA/PP_out_final7           0000000000000000    00000000000000000
/TOP_MAC/MBA/PP_out_final8           0000000000000000    00000000000000000
SPST ADDITION OUTPUTS
/TOP_MAC/MBA/stage1_SPST_out1        0                   0
/TOP_MAC/MBA/stage1_SPST_out2        0                   0
/TOP_MAC/MBA/stage2_SPST_out1        0                   0
CARRY ADDER
/TOP_MAC/MBA/stage1_add_out1         35                  20  16  40  35
FINAL MULTIPLIER OUT (X * Y)
/TOP_MAC/MULTIPLICATION              35                  20  16  40  35
ACCUMULATION (X*Y+acc)
/TOP_MAC/acc                         298                 0  20  40  60  76  92  108  148  188  228  263  298
/TOP_MAC/Final_out                   298                 0  20  40  60  76  92  108  148  188  228  263  298
Now                                  1300 ps             ps  200 ps  400 ps  600 ps  800 ps  1000 ps  1200 ps
```

Figure 11: Simulation results for a 16-bit multiplier and accumulator using SPST



```
wave - default
Messages
/booth_mul/Multiplier     0000000001101010    0000000001101010
/booth_mul/Multiplicand   0010101011001001    0010101011001001
/booth_mul/PP1            11010101001101110   11010101001101110
/booth_mul/PP2            11101010100110111   11101010100110111
/booth_mul/PP3            11101010100110111   11101010100110111
/booth_mul/PP4            00101010110010010   00101010110010010
/booth_mul/PP5            00000000000000000   00000000000000000
/booth_mul/PP6            00000000000000000   00000000000000000
/booth_mul/PP7            00000000000000000   00000000000000000
/booth_mul/PP8            00000000000000000   00000000000000000
Now        600 ps     ps  200 ps  400 ps  600 ps  800 ps
Cursor 1   0 ps       0 ps
adder_17bits.v   booth_mul.v   wave
```

Figure 12:Generation of Partial products using MBA Technique



Supply Source	Summary Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.200	0.041	0.000	0.041
Vccaux	2.500	0.034	0.000	0.034
Vcco25	2.500	0.004	0.000	0.004

	Total	Dynamic	Quiescent
Supply Power (W)	0.144	0.000	0.144

Figure 13: Power analysis of design

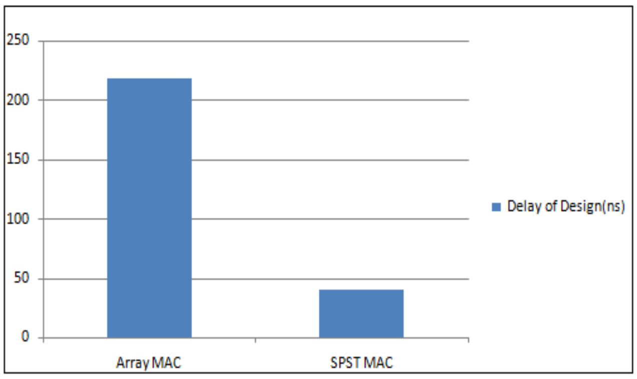


Figure 14: Improvements of delay over conventional design by using the SPST

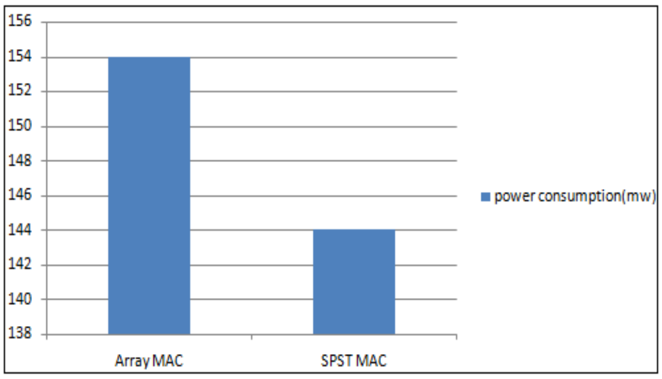


Figure15: Improvements of power over conventional design by using the SPST

The figure 14 shows the improvement of delay over conventional design using the SPST. The SPST with CSA gives a factor of 5 less delay. This improvement in delay is achieved because of using of 16-Bit Carry Save Adder in our design. Carry save addition carry will be saved instead of adding to next stage which will avoid waiting time of next adder. 16-Bit CSA is implemented using ripple carry adder in this design.

The figure 15 shows the improvement in power by using SPST. 7% less power consumption is achieved as compared to array MAC which is possible because of using SPST in this design SPST will avoid unwanted additions which major area of power consumption in CMOS devices.

VII. CONCLUSION & SCOPE FOR FUTURE WORK

A 16x16 multiplier-accumulator (MAC) is presented in this work. A Radix-2 Modified Booth multiplier circuit is used for MAC architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. The basic building blocks for the MAC unit are identified and each of the blocks is analyzed for its performance. Power and delay is calculated for the blocks. 1-bit MAC unit is designed with enable to reduce the total power consumption based on block enable technique. Using this block, the N-bit MAC unit is constructed and the total power consumption is calculated for the MAC unit. The power reduction techniques adopted in this work. The MAC unit designed in this work can be used in filter realizations for High speed DSP applications. Table 6 summarizes the results obtained. With the above technique a factor of 6 less delay and 7% less power consumption is achieved as compared to array MAC. In this design 16-Bit carry save adder is used in addition stage. The basic building block for 16-Bit carry save adder is 4-Bit ripple carry adder(RCA) if we can replace RCA with carry look ahead adder we can get better results.

REFERENCES

[1] Young-Ho Seo and Dong-Wook Kim,"A new VLSI architecture of parallel multiplier-accumulator based on radix-2 modified Booth algorithm", in IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 2, pp.201-208, February 2010.

[2] C.N. Marimuthu and P. Thangaraj, "Low power high performance multiplier", Proc. ICGST-PDCS, vol. 8, pp.31–38, December 2008.

[3] K.H.Chen, Y.M.Chen, and Y.S.Chu, "A versatile multimedia functional unit design using the spurious power suppression technique", in Proc. IEEE Asian Solid-State Circuits Conf., 2006,pp.111–114.

[4] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[5] G. Lakshmi Narayanan and B. Venkataramani, "Optimization techniques for FPGA -based wave pipelined DSP blocks", IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol.13, no. 7, pp.783-792, July 2005.

[6] K.H.Chen, K.C.Chao, J.I.Guo, J.S.Wang and Y.S. Chu, "An efficient spurious power suppression technique (SPST) and its applications on MPEG-4 AVC/H.264 transform coding design", Proc. IEEE Int. Symps. Low Power Electron. Des., pp.155–160, 2005.

[7] P.Zicari, S.Perri, P.Corsonello and G.Cocorullo, "An optimized adder accumulator for high speed MACs", Proc. ASICON 2005, vol.2, pp.757–760, 2005.

[8] F. Elguibaly, "A fast parallel multiplier–accumulator using the modified Booth algorithm," *IEEE Trans. Circuits Syst.*, vol. 27, no. 9, pp. 902–908, Sep. 2000.

[9] Z. Huang and M. D.Ercegovac, "High-performance low-power left-to right array multiplier design", IEEE Trans.Comput.,vol.54, no.3, pp.272–283, March 2005.

[10] H. Lee, "A power-aware scalable pipelined Booth multiplier", Proc. IEEE Int. SOC Conf., , pp.123–126, 2004.

[11] A. Fayed and M. Bayoumi, "A merged multiplier-accumulator for high speed signal processing applications", Proc. ICASSP, vol. 3, pp.3212–3215, 2002.

[12] J.Choi, J. Jeon and K.Choi,"Power minimization of functional units by partially guarded computation", Proc. IEEE Int. Symp. Low Power Electron. Des., pp. 131–136, 2000.

[13] L.Benini, G.D.Micheli, A.Macii, E.Macii, M.Poncino, and R.Scarsi, "Glitching power minimization by selective gate freezing", IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol.8, no.3, pp.287–297, June2000.

[14] F. Elguibaly, "A fast parallel multiplier–accumulator using the modified Booth algorithm", IEEE Trans. Circuits Syst., vol. 27, no. 9, pp.902–908, September 2000.

[15] J. Fadavi-Ardekani, "M*N Booth encoded multiplier generator using optimized Wallace trees", IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 1, no. 2, pp. 120–125, June1993.

[16] A. R. Cooper, "Parallel architecture modified Booth multiplier",Proc.Inst.Electr.Eng.G, vol.135,pp.125–128, 1988.