International Journal of Latest Trends in Engineering and Technology Vol.(8)Issue(1), pp.222-229 DOI: http://dx.doi.org/10.21172/1.81.030 e-ISSN:2278-621X

# DESIGNING OF MODIFIED BOOTH ENCODER WITH POWER SUPPRESSION TECHNIQUE

Vasudeva G<sup>1</sup> and Venkatesh S N<sup>2</sup>

Abstract— The project titled "Designing of Modified Booth Encoder with power suppression technique" is useful technique in reducing the area, delay and power consumption. Recent studies in designing of DSP systems have revealed loss of high performance due to huge area utilization and delay. In Radix-4 Modified Booth encoder the partial products has been reduced to half of the earlier one. Hence the number of adders are reduced, so the area consumption will be reduced and delay in the output also reduced by a factor. The power Suppression technique used along with the Modified booth Encoder is SPST (Spurious Power Suppression Technique). By using this technique the glitches can be reduced and avoids unwanted operation such as addition of repeated zeros. Therefore the power consumption will be less. Hence this project is useful in the field of signal processing and also very useful in portable systems.

#### I. INTRODUCTION

As integrated circuit technology has improved to allow more and more components on a chip, digital systems have continued to grow in complexity. As digital systems have become more complex, detailed design of the systems at the gate and flip-flop level has become very tedious and time consuming. For this reason, use of hardware description languages in the digital design process continues to grow in importance. A hardware description language allows a digital system to be designed and debugged at a higher level before conversion to the gate and flip-flop level. Digital signal processing is one of the core technologies in rapidly growing application areas such as wireless communications, audio and video processing, and industrial control. Digital signal processing (DSP) applications constitute the critical operations which usually involve many multiplications.

Fast multipliers are essential parts of digital signal processing systems. 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

<sup>1</sup> Dept of ECE Atria Institute of Technology Bangalore-560024 <sup>2</sup> Dept of ECE Brindavan college of Engineering Bangalore almost always replaced by an algorithm that makes use of positional representation. 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. The basic multiplication principle is twofold 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. 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.

#### **II.NORMAL MULTIPLICATION METHOD**

Generally in multiplication there will be an two terms name multiplicand and multiplier. The multiplicand each terms will be multiplied with each terms of multiplier and produces partial product for each term and finally all the partial products are added to get the final result. In this the partial products will the terms of multiplicand values. Example If the multiplier has 8 bits and multiplicand will have 5 bits then there will be 5 partial products and hence the five adders required.



Fig:1 Existing Normal Multiplication method

### **III.MODIFIED BOOTH ALGORITHM**

Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950. In booth multiplication, partial products generation is done based on recoding scheme. e.g., radix 4 encoding. Bits of Multiplicand (Y) are grouped from left to right and corresponding *operation on* Multiplier (X) is done in order to generate the partial product. In radix-4 booth multiplication partial product generation is done on based on encoding which is as given by Table1. Parallel multiplication using Booth's recording algorithm technique based on the fact that partial product can be generated for group of consecutive 0's and 1's which is called Booth's recording. This recording algorithm is used to generate efficient partial product. These partial products always have large number of bits than the input number of bits. This increase in the width of partial products usually depends upon the radix scheme used for recording. So, these scheme uses less partial product

generation which in turn provides low power and are in the Modified Booth encoding multiplier.

Blocks	Recoding	Operation	Partial
C <sub>k</sub>	Digit S <sub>k</sub>	on M	product
			PP <sub>k</sub>
000	0	0*M	S <sub>0</sub> *M
001	+1	+1*M	S <sub>1</sub> *M
010	+1	+1*M	S <sub>2</sub> *M
011	+2	+2*M	S <sub>3</sub> *M
100	-2	-2*M	S <sub>4</sub> *M
101	-1	-1*M	S <sub>5</sub> *M
110	-1	-1*M	S <sub>6</sub> *M
111	0	0*M	S <sub>7</sub> *M

 Table 1: Booth's encoding Table

Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values A and S to a product P, then performing a rightward arithmetic shift on P. Let **m** and **r** be the multiplicand and multiplier, respectively; and let x and y represent the number of bits in **m** and **r**.

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).

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

- ii. S: Fill the most significant bits with the value of  $(-\mathbf{m})$  in two's complement notation. Fill the remaining (y + 1) bits with zeros.
- iii. P: Fill the most significant x bits with zeros. To the right of this, append the value of **r**. Fill the least significant (rightmost) bit with a zero.
  - 2. Determine the two least significant (rightmost) bits of *P*.
- i. If they are 01, find the value of P + A. Ignore any overflow.
- ii. If they are 10, find the value of P + S. Ignore any overflow.
- iii. If they are 00, do nothing. Use *P* directly in the next step.

iv. 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 r.

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 2 shows the grouping of bits from the multiplier term for use in modified booth encoding.



Fig 2: 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.

								0	0	0	0	0	1	0	0 0	8
							X	0	0	0	0	1	0	1	0 0	20
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0	- 0 × Y
	0	0	0	0	0	0	0	0	0	0	1	0	0	0		1×Y
╀	0	0	0	0	0	0	0	0	1	0	0	0				1×Y
	0	0	0	0	0	0	0	0	0	0						$0 \times Y$
	0	0	0	0	0	0	0	0								$0 \times Y$

0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 160

## IV. SPST Adder

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 3 shows the design of low power adder/subtraction with SPST.

The adder /subtraction is divided into two parts, the most significant part (MSP) and the least significant part (LSP). The MSP of the original adder/subtraction 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 3.



Fig 3: Low Power Adder/Subtraction Adopting the SPST

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 3. 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. Spurious transitions (also called glitches) in combinational CMOS logic are a well known source of unnecessary power dissipation. Reducing glitch power is a highly desirable target because in the vast majority of digital CMOS circuits, only one signal transition per clock cycle is functionally meaningful. Unfortunately, glitch power is heavily dependent on the low-level implementation details, namely, gate propagation delays and input transitions misalignments.

The MSP of the original adder is modified to include detection logic circuits, data controlling circuits, sign extension circuits, logic for calculating carry-in and carry-out signals. Simple logic gates are used to implement the latches and the sign extension circuits in order to reduce the additional overhead as for as possible.

Low power adder/subtraction consists of

- $\succ$  latch,
- Detection logic,
- Sign extension logic.

Detection Logic using registers



Fig4 : Detection logic using Registers

### V. MODIFIED BOOTH ENCODER WITH SPST

Figure 5 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 8, 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 6.

Fig 5 Multiplication using Modified Booth Encoding

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.



Fig 6: SPST Equipped Modified Booth Encoder

#### Applying the SPST on the Compression Tree

The proposed SPST -equipped multiplier is illustrated in figure 7. 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.



Fig 7: Proposed low power Equipped Multiplier

### VI. RESULTS AND CONCLUSION

In this project we are examining the performance of the proposed high speed low power multiplier with respect of array multiplier. This multiplier can be implemented using Verilog-HDL coding. In order to get the power report and delay report we are synthesizing this multiplier using Xilinx.

Nour		560.3												
1000 ns		0	2	00 	4	00 		600 		800		1000		
🗉 💦 p3_out(8:0)	9	9°hXXX												
S0	0													
3 <mark>.</mark> S1	0													
61 S2	0	888 <b>8</b> 888												
👌 S3	0													
p0_out[8:0]	9			9'n000			X							
🖬 🛃 p[15:0]	1	(16'hXXXX)	16'h0000X 16'h0000 X											
🖽 🚮 p1_out(8:0)	9	9°5 Y Y Y	9hvvv V. 9h000 V. 9h025											
p1_out1[10:0]	1	( 11) Run	ning sirr	iulation	Simulat	ion time	3203	01900	J ns. 11	'h094				
p2_out1[12:0]	1	13'hXXX0												
🖪 🛃 p2_out[8:0]	9	9°hXXX	9'hXXX X 9'h000											
PERIOD[31:0]	3		32%00000000											
JI DUTY_CYCLE	0.5	0.5												
OFFSET[31:0]	3	32'h0000000												
👌 🛛 clk	1													
🖬 😽 x_i[7:0]	8'h25			X				8'h2	5					
🖬 🚮 y_i[7:0]	8'h05			8'h00		X	8°h05							

Fig8: Waveforms of Booth Multiplier

Now: 1000 ns		40	0	5	00 	61	00	70	00 	80	10	1
c_out	0								ĺ			
🖬 🚮 sum[3:0]	4"hD	4'h1 )	4'h(	) (4.,	4X 4'hC X 4'hD				41hC 41hD			
PERIOD[31:0]	3						h000000C	3				
DUTY_CYCLE	0.5											
OFFSET[31:0]	3						h0000000					
SII cin	0											
🚴 🛛 rst	0											
🖿 😽 a[3:0]	4'h3		4'n0	X				4'h3				
🖬 🛃 b[3:0]	4'hE		4'h0 X 4'hE									

Fig9 : waveform of SPST ADDER

CONSTRAINTS	BOOTH MULTIPLIER	BOOTH MULTIPLIER WITH SPST ADDER				
DELAY	23.79ns	10.391ns				
POWER	0.305mW	0.211mW				

## VII. APPLICATIONS

The SPST can drastically reduce the power dissipation of combinational VLSI designs for multimedia/DSP applications. There are several DSP applications, out of which one is an efficient multi Transform Design (ETD) for MPEG-4 and the second one is a Versatile Multimedia Functional Unit (VMFU) design. The ETD can compute forward, inverse, hadamard transforms. On the other hand, the VMFU can compute six commonly used arithmetic operations in multimedia/DSP processing like addition, subtraction, multiplication,

MAC (Multiplier and accumulator), interpolation, Sum-of-Absolute Difference (SAD) Encapsulating the VMFUs, designers can increase the flexibility and scalability of multimedia/DSP processors. When applying the SPST to these two designs, the realization issues in every design highly differ from each other due to the large hardware-configuration differences.

## VIII. CONCLUSION

This work presents the designing of a 16x16 multiplier with high speed low-power technique called SPST (Spurious Power Suppression Technique). A Modified Radix-4 Booth Encoder circuit is used for this multiplier architecture. Compared to array multiplier, the booth multiplier has the highest operational speed and less hardware count.

The presented low-power technique called SPST and the theoretical analysis of the SPST are fully discussed. The proposed SPST can obviously decrease the switching (or dynamic) power dissipation, which decreases a significant portion of the whole power dissipation in integrated circuits. Simulation results and Power analysis reports of SPST-equipped multiplier and array multiplier are shown.

## REFERENCES

- A. D. Booth, "A signed binary multiplication technique," Quarterly Journal of Mechanics and Applied Mathematics, vol. 2, pp. 236-240, 1951.
- [2] O. L. MacSorley, "High-speed arithmetic on binary computers," IRE Transaction on Electronic Computers, vol. 49, pp. 67-91, 1961.
- [3] A. R. Cooper, "Parallel Architecture Modified Booth Multiplier", Proceedings of the Institution of Electrical Engineers, 1989.
- [4] J. Fadavi-Ardekani, "M × N booth encoded mult iplier generator using optimized Wallace trees," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 1, no. 2, pp. 120–125, Jun, 199 3.
- [5] W. -C. Yeh and C. -W. Jen, "High-speed booth encoded parallel multiplier design," IEEE Trans. Comput. , vol. 49, no. 7, pp. 692–701, Jul, 2000.
- [6] B Ayman A. Fayed, Magdy A. Bayoumi, "A Novel Architecture for Low-Power Design of Parallel Multipliers," vlsi,pp. 0149, IEEE Computer Society Workshop on VLSI 2001.
- [7] RazaidiHussin, Ali Yeon Md. Shakaff, Norina Idris1, Zaliman Sauli1, RizalafandeChe Ismail1 and Afzan Kama An Efficient Modified Booth Multiplier Architecture, International Conference on Electronic Design, 2008.
- [8] Shiann-RongKuang and Jiun-Ping Wang "Modified Booth Multipliers with a Regular Partial Product Array" IEEE Transactions on Circuits and Systems—II Volume. 56, No. 5, 2009.
- [9] Soojin Kim and Kyeongsoon Cho, "Design of High-speed Modified Booth Multipliers Operating at GHz Ranges," World Academy of Science, Engineering and Technology, Vol. 7, No. 2, 2010.
- [10] S. K. Sahoo and C. Shekhar, "A fast final adder for a 54-bit parallel multiplier for DSP application", international journal of electronics, vol. 98, no. 12, pp. 1625-1638, 2011.
- [11] S. K. Sahoo and C. Shekhar,"Delay Optimized Array Multiplier for Signal and Image Processing", 2011 International Conference on Image Information Processing, 2011.
- [12] C. Senthilpari "A low power and High performance Radix-4 Multiplier design using a Modified Pass Transistor Logic Technique" IETE Journel of Research Vol. 57, Issue 2, 2012.
- [13] Neil. H. Weste and Kamran Eshraghian, "Principles of CMOS VLSI Design," Addison-Wesley, ISBN 0201733897,pp 361-380, 1993.
- [14] Kiat-Seng Yeo and Kaushik Roy, "Low-Voltage, Low-Power VLSI Subsystems" Tata McGraw-Hill, ISBN-13-978-0-07-067750-0, pp 119- 146, 2009.
- [15] Pratap Kumar Dakua, AnamikaSinha, Shivdhari&Gourab, "Hardware Implementation of MAC unit," International Journal of Electronics Communication and Computer Engineering, vol. 3, 2012.