Journal of Engineering Sciences
Vol 12, Issue 06, JUNE /2021
ISSN NO:0377-9254

DESIGN OF LOW POWER AND HIGH SPEED MULTIPLIER USING SPURIOUS POWER SUPPRESSION TECHNIQUE (SPST)

Mr. Nookala Sairam[1], Gatla Priyanka[2], Bavani Navya sree[3], Bandi Sai Siddhardh[4], T. K. R. Nitin Srivastav[5]

[1]Assistant Professor, Dept. of ECE, Vignana bharathi institute of technology.

[2345]B.Tech Students, Dept. of ECE, Vignana bharathi institute of technology.

ABSTRACT

Spurious-power suppression technique (SPST) which can dramatically reduce the power dissipation of combinational VLSI designs for multimedia/DSP purposes. 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. 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. To save significant power consumption of a VLSI design, it is a good direction to reduce its dynamic power that is the major part of total power dissipation.

This paper presents the design exploration and applications of a spurious-power suppression technique (SPST) which can dramatically reduce the power dissipation of combinational VLSI designs for multimedia/DSP purposes. 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. Furthermore, this paper proposes an original glitch diminishing technique to filter out useless switching power by asserting the data signals after the data transient period.

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. To save significant power consumption of a VLSI design, it is a good direction to reduce its dynamic power that is the major part of total power dissipation. In this paper, we propose a high speed / low-power multiplier 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.

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. 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. 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.

Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement 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, whole spectrums of

multipliers with different area-speed constraints are designed with fully parallel processing. In between are digit serial multipliers where single digits consisting of several bits are operated on. These multipliers have moderate performance in both speed and area. However, existing digit serial multipliers have been plagued by complicated switching systems and/or irregularities in design. Radix 2^n multipliers which operate on digits in a parallel fashion instead of bits bring the pipelining to the digit level and avoid most of the above problems. They were introduced by M. K. Ibrahim in 1993. These structures are iterative and modular. The pipelining done at the digit level brings the benefit of constant operation speed irrespective of the size of' the multiplier. The clock speed is only determined by the digit size which is already fixed before the design is implemented.

LITERATURE SURVEY

The growing market for fast floating-point co-processors, digital signal processing chips, and graphics processors has created a demand for high speed, area-efficient multipliers. Current architectures range from small, low-performance shift and add multipliers, to large, high-performance array and tree multipliers. Conventional linear array multipliers achieve high performance in a regular structure, but require large amounts of silicon. Tree structures achieve even higher performance than linear arrays but the tree interconnection is more complex and less regular, making them even larger than linear arrays. Ideally, one would want the speed benefits of a tree structure, the regularity of an array multiplier, and the small size of a shift and add multiplier. To reduce the size of the multiplier a partial tree is used together with a 4-2 carry-save accumulator placed at its outputs to iteratively accumulate the partial products. This allows a full multiplier to be built in a fraction of the area required by a full array. Higher performance is achieved by increasing the hardware utilization of the partial 4-2 tree through pipelining. To ensure optimal performance of the pipelined 4-2 tree, the clock frequency must be tightly controlled to match the delay of the 4-2 adder pipe stages. The SPST (Spurious Power Suppression Technique) is used for digital signal processing (DSP), Transformations of Digital Image Processing and versatile multimedia functional unit (VMFU) etc. The Booth's radix-4 algorithm, Modified Booth Multiplier, 34-bit CSA are improves speed of Multipliers and SPST adder will reduce the power consumption in addition process.

PROPOSED DESIGN

In the majority of digital signal processing (DSP) applications the critical operations usually involve many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput Multiplier-Accumulator (MAC) is always a key to achieve a high performance digital signal processing system and versatile Multimedia functional units. In the last few years, the main consideration of MAC design is to enhance its speed. This is because; speed and throughput rate is always the concern of VMFU. But for the epoch of personal communication, low power design also becomes another main design consideration. This is because; battery energy available for these portable products limits the power consumption of the system. Therefore, the main motivation of this work is to investigate various Pipelined multiplier/accumulator architectures and circuit design techniques which are suitable for implementing high throughput signal processing algorithms and at the same time achieve low power consumption. A conventional VMFU unit consists of (fast multiplier) multiplier and an accumulator that contains the sum of the previous consecutive products. The function of the VMFU unit is given by the following equation:

$$F = \Sigma A i Bi$$

The main goal of a VMFU design is to enhance the speed of the MAC unit, and at the same time limit the power consumption. In a pipelined MAC circuit, the delay of pipeline stage is the delay of a 1-bit full adder. Estimating this delay will assist in identifying the overall delay of the pipelined MAC. In this work, 1-bit full adder is designed. Area, power and delay are calculated for the full adder, based on which the pipelined MAC unit is designed for low power.

PROPOSED SPST

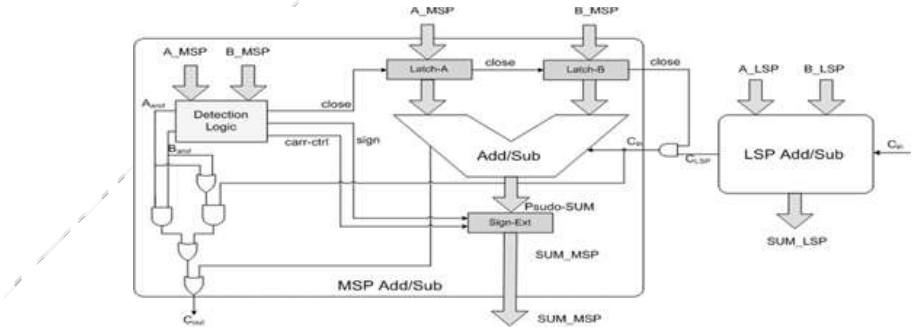
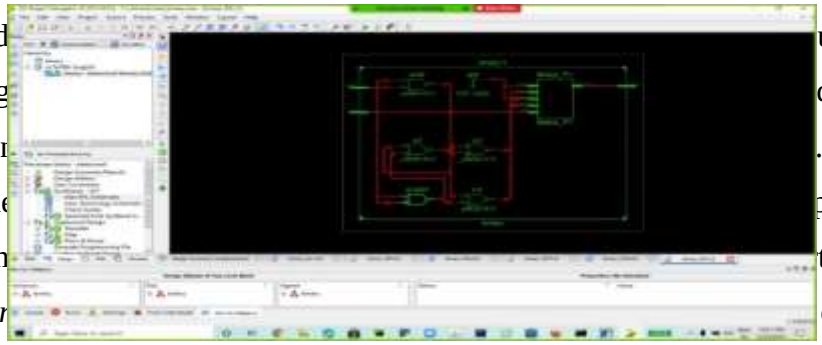


Fig.1: 16-bit adder/subtractor design example adopting the proposed SPST.

In this example, the 16-bit adder/subtractor 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 operand  ubtractor, so that the detection log d on the derived Boolean equatio . 6(a), which can determine whethe ropose the novel glitch-diminishin tion of the *close*, *sign*, and *carr-ctr* cascaded circuits which are usually adopted in VLSI architectures designed for multimedia/DSP applications. The timing diagram is shown in Fig. 6(b). 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.

BINARY MULTIPLER

In the binary number system the digits, called bits, are limited to the set. The result of multiplying any binary number by a single binary bit is either 0, or the original number. This makes forming the intermediate partial-products simple and efficient. Summing these partial-products is the time consuming task for binary multipliers. One logical approach is to form the partial-products one at a time and sum them as they are generated. Often implemented by software on processors that donot have a hardware multiplier, this techniqueworks fine, but is slow because at least one machine cycle is required to sum each additional partial-product. For applications where this approach does not provide enough performance, multipliers can be implemented directly in hardware.

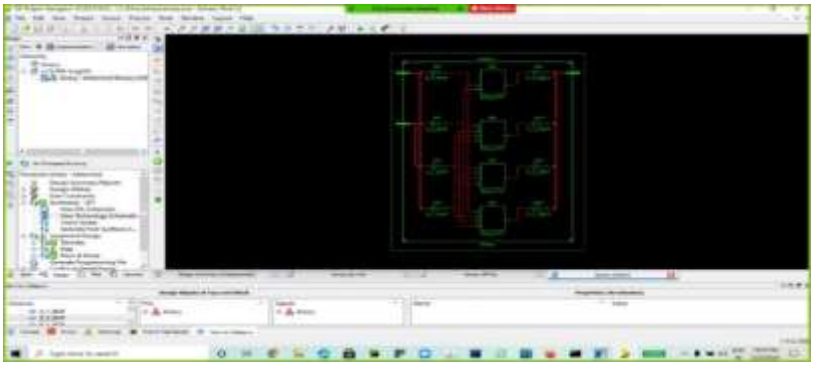


Fig.2: Technical Schematic of Binary multiplier

	Voltage (V)	Current (mA)	Power (mW)
Dynamic		509.45	611.34
Quiescent		102.92	123.51
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		18.00	45.00
Vcco25	2.5		
Dynamic		877.90	2194.75
Quiescent		2.00	5.00
Total Power			2979.60
Startup Current (m		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

HARDWARE MULTIPLIERS

Direct hardware implementations of shift and add multipliers can increase performance over software synthesis, but are still quite slow. The reason is that as each additional partial-product is

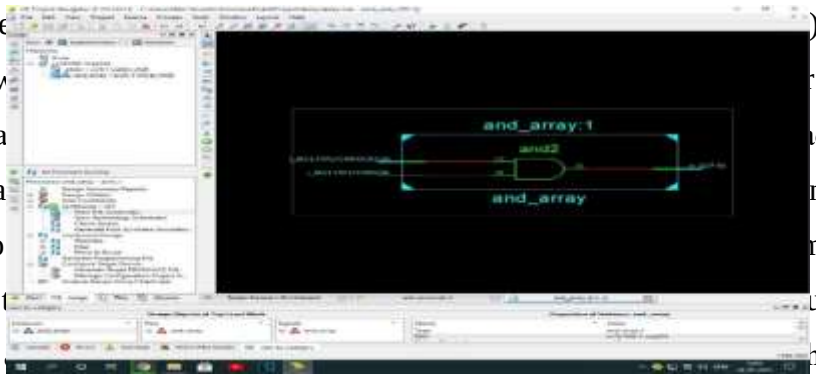
summed a carry must be propagated from the least significant bit **(LSB)** to the most significant bit **(MSB).** This carry propagation is time consuming, and must be repeated for each partial product to be summed.

One method to increase multiplier performance is by using encoding techniques to reduce the number of partial products to be summed. Just such a technique was first proposed by Booth [BOO 511. The original Booth's algorithm ships over contiguous strings of l's by using the property that: 2" + 2(n-1) + 2(n-2) + . . . + 2hm) = 2(n+l) - 2(n-m). Although **Booth's algorithm** produces at most N/2 encoded partial products from an N bit operand, the number of partial products produced varies. This has caused designers to use modified versions of Booth's algorithm for hardware multipliers. Modified **2**-bit Booth encoding halves the number of partial products to be summed.

Since the resulting encoded partial-products can then be summed using any suitable method, modified 2 bit Booth encoding is used on most modern floating-point chips LU 881, MCA 861. A few designers have even turned to modified 3 bit Booth encoding, which reduces the number of partial products to be summed by a factor of three IBEN 891. The problem with 3 bit encoding is that the carry-propagate addition required to form the 3X multiples often overshadows the potential gains of 3 bit Booth encoding.

To achieve even higher performance advanced hardware multiplier architectures search for faster and more efficient methods for summing the partial-products. Most increase performance by eliminating the time consuming carry propagate additions. To accomplish this, they sum the partial-products in a redundant number representation. The advantage of a redundant representation is that two numbers, or partial-products, can be added together without propagating a carry across the entire width of the number. Many redundant number representations are possible. One commonly used representation is known as carry-save form. In this redundant representation two bits, known as the carry and sum, are used to represent each bit position. When two numbers in carry-save form are added together any carries that result are never propagated more than one bit position. This makes adding two numbers in carry-save form much faster than adding two normal binary numbers where a carry may propagate. One common method that has been developed for summing rows of partial products using a carry-save representation is the array multiplier.

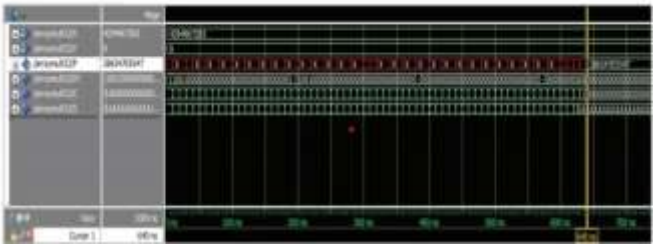
ARRAY MULTIPLIERS

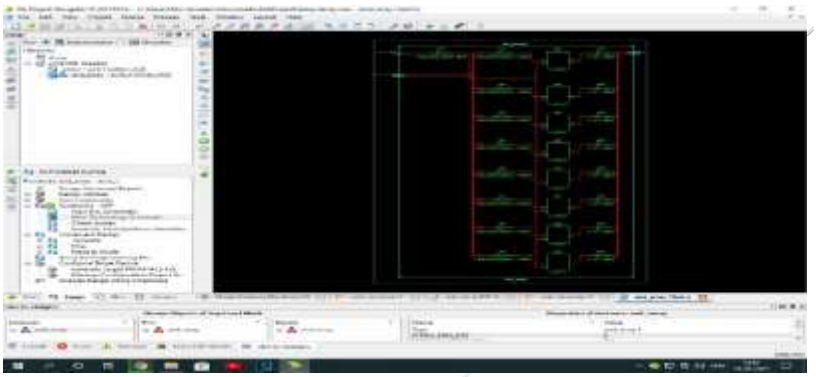
Conventional line 

). A portion of an array multiplier w r array multiplier, as the data propa dditional partial-product to the pa ndant, carry-save form there is no multiplier is only dependent upon t uct width. Linear array multipliers high performance and regular structure have perpetuated the use of array multipliers for VLSI math co-processors and special purpose DSP chips.

The biggest problem with full linear array multipliers is that they are very large. As operand sizes increase, linear arrays grow in size at a rate equal to the square of the operand size. This is because the number of rows in the array is equal to the length of the multiplier, with the width of each row equal to the width of multiplicand. The large size of full arrays typically prohibits their use, except for small operand sizes, or on special purpose math chips where a major portion of the silicon area can be assigned to the multiplier array.

Another problem with array multipliers is that the hardware is underutilized. As the sum is propagated down through the array, each row of CSA's computes a result only once, when the active computation front passes that row. Thus, the hardware is doing useful work only a very small percentage of the time. This low hardware utilization in conventional linear array multipliers makes performance gains possible through increased efficiency. For example, by overlapping calculations pipelining can achieve a large gain in throughput [NOL 861. Figure 1.3 shows a full array pipelined after each row of CSA's. Once the partial sum has passed the first row of CSA's, represented by the shaded row of GSA's in cycle 1, a subsequent multiply can be started on the next cycle. In cycle 2, the first partial sum has passed to the second row of CM's, and the second multiply, represented by the cross hatched row of CSA's, has begun. Although pipelining a full array can greatly increase throughput, both the size and latency are increased due to the additional latches





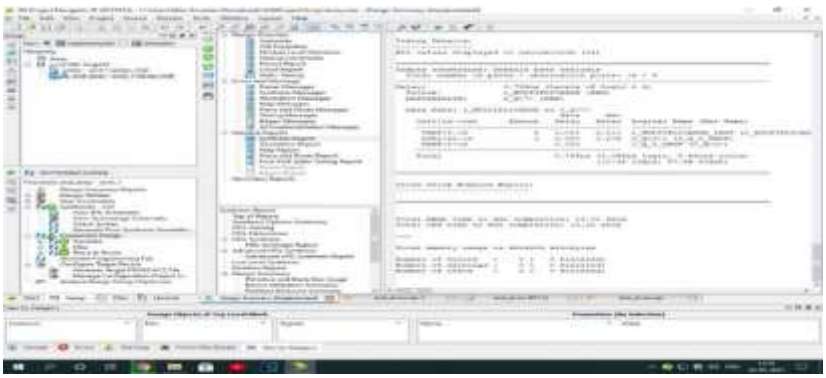


Fig.9: Output of array multiplier

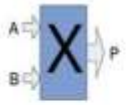
BOOTH MULTIPLIER

The Booth multiplier architecture can be divided into two stages. In the first stage the Partial Products are formed by the Booth encoder and Partial Product Generator (PPG). In the second stage the partial products obtained in the above are merged to form the results. Instead of adders here the 5:2,4:2,3:2 compressors can also be used to reduce the carry propagation delay to a great extent. When the adders alone seen we can have the adder circuits such as Carry Propagation adder,carry save adders. Prior to Multiplication, we require the two operands, a Multiplier and a Multiplicand which are to be stored in the buffer. In normal Binary Multipliers the Partial Products are generated by performing AND operation(multiplying) the bits of Multiplier with the Multiplicand bits.Thus the array of AND gates are used in normal binary multipliers for partial products generation. Here when the multiplier bit is zero then a row of zeros are summed to previous partial product. when the multiplier bit is one then the multiplicand is added once to the previous partial products with a position shift towards left.

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.

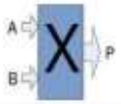
COMPARISON



Comparison of Multipliers

	Array Multiplier	Modified Booth Multiplier	Wallace-Tree Multiplier	Modified Booth-Wallace Tree Multiplier	Twin Pipe Serial-Parallel Multiplier	Behavioral Multiplier

Table.1: Comparison of Multipliers

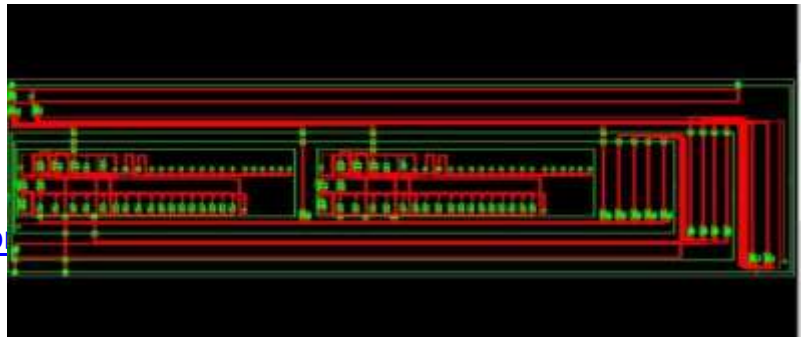


Comparison of Multipliers

	Array Multiplier	Modified Booth Multiplier	Wallace-Tree Multiplier	Modified Booth-Wallace Tree Multiplier	Twin Pipe Serial-Parallel Multiplier	Behavioral Multiplier
Area – Total CLB's (#)	3280.50	2800.00	3321.50	2845.50	487.00	3003.00
Maximum Delay D(ns)	37.23	25.33	18.93	18.33	107.52	44.50
Total Dynamic Power P (W)	7.57	6.66	7.32	6.66	0.29	6.26
Delay ·Power Product (DP) (ns W)	281.88	168.77	138.60	122.13	30.66	278.53
Area•Power Product (AP) (# W)	24837.98	18656.40	24319.36	18959.57	138.89	18795.78
Area•Delay Product (AD) (# ns)	1.22E+05	7.09E+04	6.29E+04	5.22E+04	5.24E+04	1.34E+05
Area•Delay^2 Product (AD^2) (# ns^2)	4.55E+06	1.80E+06	1.19E+06	9.56E+05	5.63E+06	5.95E+06

Table.2: Performance Comparison of Multipliers

SIMULATION RESULTS



Journal of Engineering Sciences
Vol 12, Issue 06, JUNE /2021
ISSN NO:0377-9254

Fig.10: RTL of SPST

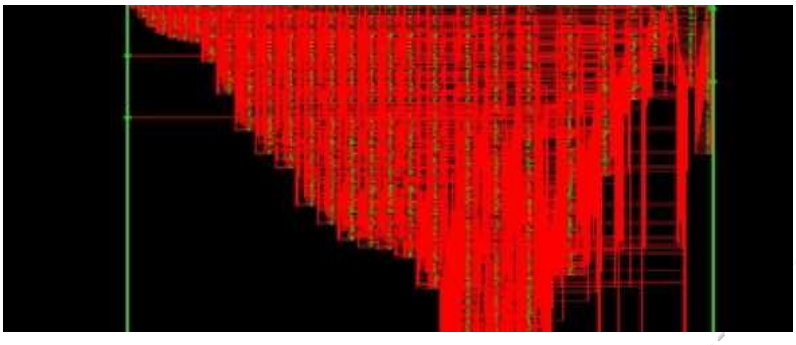


Fig.11: Technical Schematic

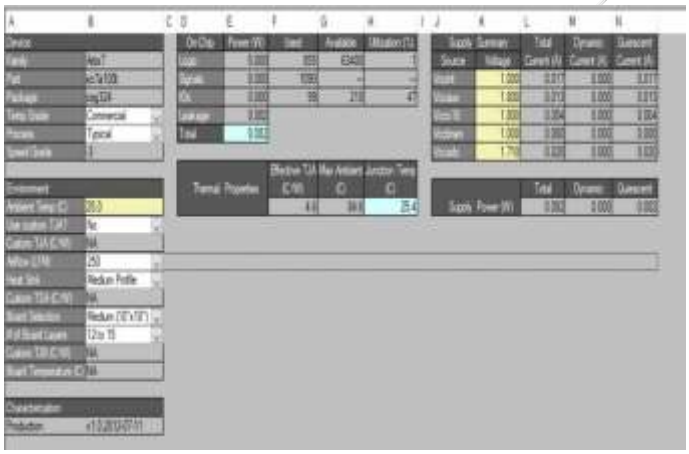


Fig.12: Output Power

Fig.13: output delay

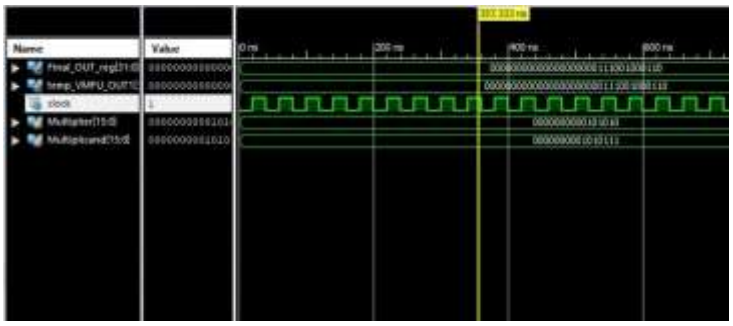


Fig.14: Output

ADVANTAGES

- The SPST adder will avoid the unwanted addition and thus minimize the switching power dissipation.
- This facilitates the robustness of SPST can attain significant speed improvement and power reduction when compared with the conventional tree multipliers.
- The modified Booth encoder will reduce the number of partial products generated by a factor two.
- Low power consumption.
- Effective at high end system.

DISADVANTAGES

- 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., the16-bit adder/subtractor in this design.
- 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., the16-bit adder/subtractor in this design example) remains the same.

CONCLUSION

This study proposes a multiplier adopting the new SPST implementing approach. The simulation results show that the power reduction of the new approach, i.e. a 40% saving, is very close to that of the former approach. Besides, the new approach leads to a 40% speed improvement when compared with the former one. When implemented in a 0.18-μm CMOS technology, the proposed SPST-equipped multiplier dissipates only 0.0121 mW per MHz in H.264 texture coding. In addition, the performance of the proposed design under the conditions of different bit-width input data is explored. The results also show that the new SPST approach not only owns equivalent low-power performance but also leads to a higher maximum speed when compared with the former SPST approach. Moreover, the proposed SPST-equipped multiplier also has better power efficiency when compared with the existing modern multipliers. 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. IJOART International Journal of Advancements in Research & Technology, Volume 4, Issue 6, June -2015 35 ISSN 2278-7763 Copyright © 2015 SciResPub. 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.

FUTURE SCOPE

In this, the Booth multiplier has the highest operational speed and less hardware count as compared to other circuits. This algorithm is competitive with other more commonly used algorithms when used for high performance implementations. Considering different technique or design of MAC unit, parallel and pipelined booth multiplications give good performance in terms of speed and SPST and block enabling technique are better in low power consumption and area. Using higher radix MBA and partial product reduction technique by the hybrid carry save adder tree can give good results in terms of speed.

REFERENCES

1 O. Chen, S. Wang, and Y. W. Wu, “Minimization of switching activities of partial products for designing low-power multipliers,” IEEE Trans. VLSI, vol. 11, no. 3, pp. 418-433, June 2003.

2 Z. Huang, and M.D. Ercegovac, High-performance low-power left-to-right array multiplier design," IEEE Trans. on Computers, vol. 54, no. 3, pp. 272-283, Mar. 2005.

3 M. C. Wen, S. J. Wang; Y. N. Lin, "Low-power parallel multiplier with column bypassing," Electronic Letters, vol. 41, no.12, pp. 581-583, May 2005.

4 H. Lee, "A power-aware scalable pipelined Booth multiplier," IEEE Int. SOC Conf., pp.123-126, Sep. 2004.

5 Y. Liao, and D.B. Roberts, "A high-performance and low-power 32-bit multiply-accumulate unit with single-instruction- multiple-data (SIMD) feature," IEEE J. of Solid-State Circuits, vol. 37, no. 7, pp. 926-931, July 2002.

6 A. Danysh, and D. Tan, "Architecture and implementation of a vector/SIMD multiply-accumulate unit," IEEE Trans. on Computers, vol. 54, no. 3, pp. 284-293, Mar. 2005.

7 J. S. Wang, C. N. Kuo, and T. H. Yang, "Low-power fixed-width array multipliers," IEEE Symp. Low Power Electronics and Design, pp. 307-312, 9-11, Aug. 2004.

8 W. C Yeh, C. W. Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. Computer, vol. 49, no. 7, pp.692-701, July 2000.

9 S. K. Hsu, S. K. Mathew, M. A. Anders, B. R. Zeydel, V. G. Oklobdzija, R. K. Krishnamurthy, and

10 S. Y. Borkar, "A 110 GOPS/W 16-bit multiplier and reconfigurable PLA loop in 90-nm CMOS," IEEE

11 J. Solid-State Circuits, vol. 41, no. 1, pp. 256-264, Jan. 2006.

12 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," IEEE Int. Symp. Low Power Electronics Design, San Diego, California, Aug. 8-10 2005.

13 K. H. Chen, Y. M. Chen, and Y. S. Chu, "A Versatile Multimedia Functional Unit Design Using the Spurious Power Suppression Technique," IEEE Asian Solid-State Circuits Conf., Hangzhou, China, Nov. 2006.

14 Y. W. Lin, H. Y. Liu, and C. Y. Lee, "A dynamic scaling FFT processor for DVB-T applications," IEEE J. Solid-State Circuits, vol. 39, no. 11, pp. 2005-2013, Nov. 2004.

Journal of Engineering Sciences
Vol 12, Issue 06, JUNE /2021
ISSN NO:0377-9254