ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 6, Issue 1, January 2017

Efficient Multiplier Design Using Modified Booth Algorithm and Razor Flip-Flop.

Shubhangi Ramannawar, Deepak Kumar

***Abstract*—The advent of digital signal processing and various other applications made multipliers to play major role in science and technology. With recent advancement in technology, many researchers are working to design multipliers which offer either of the following design targets – high speed, low power consumption and less area. Furthermore, the negative bias temperature instability effect occurs when a pMOS transistor is under negative bias (Vgs = −Vdd), increasing the threshold voltage of the pMOS transistor, and reducing multiplier speed. Similarly, positive bias temperature instability, occurs when an nMOS transistor is under positive bias. Both the effects directly hinder the multiplier speed by degrading transistor speed, if this problem occurs for long time then the system may fail due to timing violations. Booth algorithm is used to design two 16-bit input and 32-bit output and is able to provide efficient high-performance multipliers with help of Razor flip flop to mitigate performance degradation that is due to the aging effect. Moreover, the proposed multiplier architecture can be applied to more number of inputs by adapting higher versions of the booth algorithm.**

***Index Terms*—Negative Bias Temperature Instability(NBTI), Positive Bias Temperature Instability(PBTI), Radix-4, Modified Booth Algorithm, Adaptive Hold Logic(AHL)**

I. INTRODUCTION

Digital multipliers are the most complex and critical arithmetic functional units in many applications, such as the Fourier transform, discrete cosine transforms, and digital filtering. The through put of these applications rely on multipliers, and if the multipliers are too slow, the performance of entire circuits will be reduced.

Parameters that degrade the multiplier speed are, the negative bias temperature instability(NBTI) effect which occurs when a pMOS transistor is under negative bias (Vgs = −Vdd), increasing the threshold voltage of the pMOS transistor, and reducing transistor switching speed. On the other hand, positive bias temperature instability (PBTI), occurs when an nMOS transistor is under positive bias.

NBTI effect results from a association of hole trapping in oxide defects and formation of interface states at the channel oxide interface (Schroder and Babcock 2003; Kaczer et al. 2008; Grasser and Kaczer 2009). PBTI is supposed to come from electron trapping in preexistant oxide traps, combined with a trap generation process (Crupi et al. 2005; Ioannou et al. 2009). Further, the very first research on next generation CMOS structures such as multi-gate devices (MuGFETs, FinFETs, etc.) suggests that BTI remains a problem in future CMOS technologies.

A traditional method to reduce the aging effect is overdesign, including such things as guard-banding and gate oversizing; however, this approach can be very pessimistic and area and power inefficient. To avoid this problem, many NBTI-aware methodologies have been proposed. An NBTI-aware technology mapping technique was proposed in to pledge the performance of the circuit during its lifetime. In an NBTI-aware sleep transistor was planned to decrease the aging effects on pMOS sleep-transistors, and thelifetime stability of the power-gated circuits under consideration was made better. Wu and Marculescu [9] proposed a joint logic restructuring and pin reordering method, which is based on detecting functional symmetries and transistor stacking effects.

II. EXISTING METHODOLOGY

The paper consists of an aging-aware reliable multiplier design with novel adaptive hold logic (AHL) circuit.[1] The multiplier is based on the variable-latency technique and can adjust the AHL circuit to achieve reliable operation under the influence of NBTI and PBTI effects. To be specific, the contributions of this paper are summarized as follows:

1. Novel variable-latency multiplier architecture with an AHL circuit. The AHL circuit can decide whether the input patterns require one or two cycles and can adjust the judging criteria to ensure that there is minimum performance degradation after considerable aging occurs.

2. Comprehensive analysis and comparison of the multiplier's performance under different cycle periods to show the effectiveness of our proposed architecture.

Manuscript received Dec 30, 2016.

***Shubhangi Ramannawar, M.Tech. Scholar,** Department of Electronics & Communication Engineering, Vidhyapeeth Institute of Science & Technology Bhopal, (RGPV,Bhopal) India.*

***Deepak Kumar**, Assistant Professor, Department of Electronics & Communication Engineering, Vidhyapeeth Institute of Science & Technology Bhopal, (RGPV,Bhopal) India.*

3. An aging-aware reliable multiplier design method that is suitable for large multipliers. Although the experiment is performed in 16- and 32-bit multipliers, our proposed architecture can be easily extended to large designs.

4. The experimental results show that our proposed architecture with the 16×16 and 32×32 column-bypassing multipliers can attain up to 62.88% and 76.28% performance improvement compared with the 16 × 16 and 32 × 32 fixed-latency column-bypassing (FLCB) multipliers. In addition, our proposed architecture with 16 × 16 and 32 × 32 row-bypassing multipliers can achieve up to 80.17% and 69.40% performance improvement as compared with 16×16 and 32×32 fixed-latency row-bypassing multipliers.

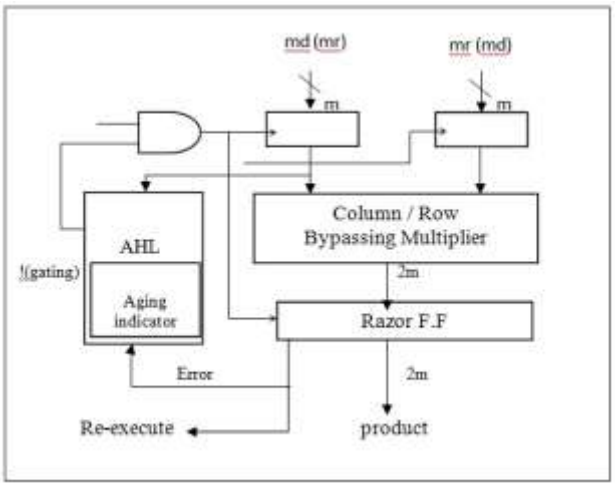

Fig 1. Existing Architecture (md means multiplicand and mr means multiplicator)

Problem Identification:

Submit The Existing architecture consists of Column/Row Multiplier along with Novel Adaptive Hold Logic and Razor flip flop to avoid timing violations. The row/ column by pass multipliers are bulky in design and consume more space.

The row/column bypass multipliers use large number of Full adders hence giving rise to more delay.

More prone to error when dealing with negative numbers.

As complexity in input increases, circuit performance degrades.

DSP application use Fast Fourier Transform (FFT), Finite Impulse Response (FIR) filter, discrete cosine transforms (DCT) etc, which use more complex circuits out of which 80% of the application consists of multipliers. So it become of at most importance to reduce delays incurred in multiplier. In base paper they used column by pass multiplier which itself is more complex in design and internally it consists of more number of full adders and if column multiplier is employed for more number of input/output combination then the multiplier architecture become more complex. Due to which there are more chances of delay and we have to compromise on speed. Instead of column by pass multiplier we can employ Advance Booth multiplier which helps in reducing partial products and more over it do not require any additional circuit to deal with negative numbers.

III. Proposed Methodology

In proposed model, we employ a modified radix-4 16x16bit Booth multiplier in place of row/column by-pass multipliers to increase throughput of multipliers. Modified Booth's algorithm employs both addition and subtraction and also treats positive and negative operands uniformly. No special actions are required for negative numbers. Multipliers are key components of many high performance systems such as FIR filters, Microprocessor, digital signal processors, etc. Signed multiplication is a careful process. With unsigned multiplication there is no need to take sign of number into consideration. Booth multiplication algorithm or Booth algorithm was named after the inventor Andrew Donald Booth. It can be defined as an algorithm or method of multiplying binary numbers in two's complement notation. It is a simple method to multiply binary numbers in which multiplication is performed with repeated addition operations by following the booth algorithm. Again this booth algorithm for multiplication operation is further modified and hence, named as modified booth algorithm.

Modified Booth Algorithm: Booth multiplication algorithm consists of three major steps as shown in the structure of booth algorithm figure that includes generation of partial product called as recoding, reducing the partial product in two rows, and addition that gives final product. For a better understanding of modified booth algorithm & for multiplication, we must know about each block of booth algorithm for multiplication process.

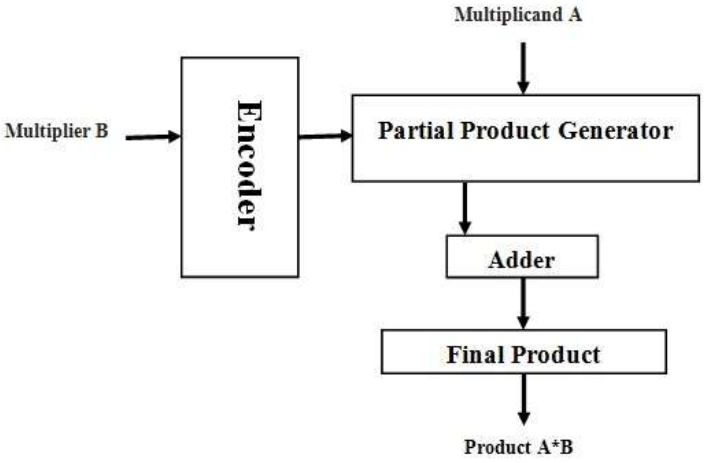

Fig 2: Modified Booth Algorithm

Brief Working Principle of Booth Algorithm: This modified booth multiplier is used to perform high-speed multiplications using modified booth algorithm. this modified booth multiplier's computation time and the logarithm of the word length of operands are proportional to each other. we can reduce half the number of partial product. radix-4 booth algorithm used here increases the speed of multiplier and reduces the area of multiplier circuit. In this algorithm, every second column is taken and multiplied by 0 or +1 or +2 or -1 or -2 instead of multiplying with 0 or 1 after shifting and adding of every column of the booth multiplier. Thus, half of the partial product can be reduced using this booth algorithm. Based on the multiplier bits, the process of encoding the multiplicand (M) is performed by radix-4 booth encoder.

The overlapping is used for comparing three bits at a time. This grouping is started from least significant bit (LSB), in which only two bits of the booth multiplier are used by the first block and a zero is assumed as third bit as shown in the figure.

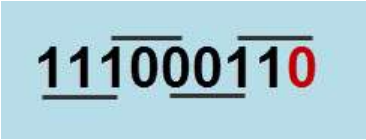

Fig 3. Bit Pairing as per Booth Recoding

The figure shows the functional operation of the radix-4 booth encoder that consists of eight different types of states. The outcomes or multiplication of multiplicand with 0, -1, and -2 are consecutively obtained during these eight states.

Hence, to design n-bit parallel multipliers only n2 partial products are generated by using booth algorithm

Table 1. Booth Recoding Table for Radix-4

Multiplier Bits Block			Recoded 1-bit pair		2 bit booth	
i+1	i	i-1	i+1	i	Multiplier Value	Partial Product
0	0	0	0	0	0	Mx0
0	0	1	0	1	1	Mx1
0	1	0	1	-1	1	Mx1
0	1	0	1	0	2	Mx2
1	0	0	-1	0	-2	Mx-2
1	0	1	-1	1	-1	Mx-1
1	1	0	0	-1	-1	Mx-1
1	1	0	0	0	0	Mx0

Now the partial products generated as part of booth multiplier are added. Simultaneously the AHL circuit is computing the number of cycles required by the multiplier and if the multiplication process exceeds two cycles then an error is generated at razor flip flop and again the multiplication process is carried out. As we are using modified booth algorithm which is more faster than array multipliers chances of timing violations to occur are nearly nill even then to keep an eye on the behaviour of the multiplier circuit we have employed AHL circuit with razor flip flop.Which makes sure that timing violations does not exist.

Adaptive Hold Logic:

The operation of the AHL circuit is as follows: when an input pattern arrives, both judging blocks will decide whether the pattern requires one cycle or morecycles tocomplete and pass both results to the multiplexer. The multiplexer selects one of either result based on the output of the aging indicator. Then an OR operation is performed between the result of the multiplexer, and the Q signal is used to determine the input of the D flip-flop.

When the pattern requires one cycle, the output of the multiplexer is 1. The !(gating) signal will become 1, and the input flip flops will latch new data in the next cycle. On the other hand, when the output of the multiplexer is 0, which means the input pattern requires more than 1 cycles to complete, the OR gate will output 0 to the D flip-flop.

Therefore, the !(gating)signal will be 0 to disable the clock signal of the input flip-flops in the next cycle. Note that only a cycle of the input flip-flop will be disabled because the D flip-flop will latch 1 in the next cycle.

RAZOR FLIP FLOP:

A 1-bit Razor flip-flop contains a main flip-flop, shadow latch, XOR gate, and mux. The main flip-flop catches the execution result for the combination circuit using a normal clock signal, and the shadow latch catches the execution result using a delayed clock signal, which is slower than the normal clock signal. If the latched bit of the shadow latch is different from that of the main flip-flop, this means the path delay of the current operation exceeds the cycle period, and the main flip-flop catches an incorrect result.

If errors occur, the Razor flip-flop will set the error signal to 1 to notify the system to re-execute the operation and notify the AHL circuit that an error has occurred. We use Razor flip-flops to detect whether an operation that is considered to be a one-cycle pattern can really finish in a cycle. If not, the operation is re-executed with two cycles.

IV. Results & Summary

Our proposed multiplier design has three key features. First, it is a variable-latency design that minimizes the timing waste of the noncritical paths. Second, it can provide reliable operations even after the aging effect occurs. The Razor flip-flops detect the timing violations and re-execute the operations using two cycles. Finally, our architecture can

adjust the percentage of one-cycle patterns to minimize performance degradation due to the aging effect. When the circuit is aged, and many errors occur, the AHL circuit uses the second judging block to decide if an input is one cycle or two cycles.

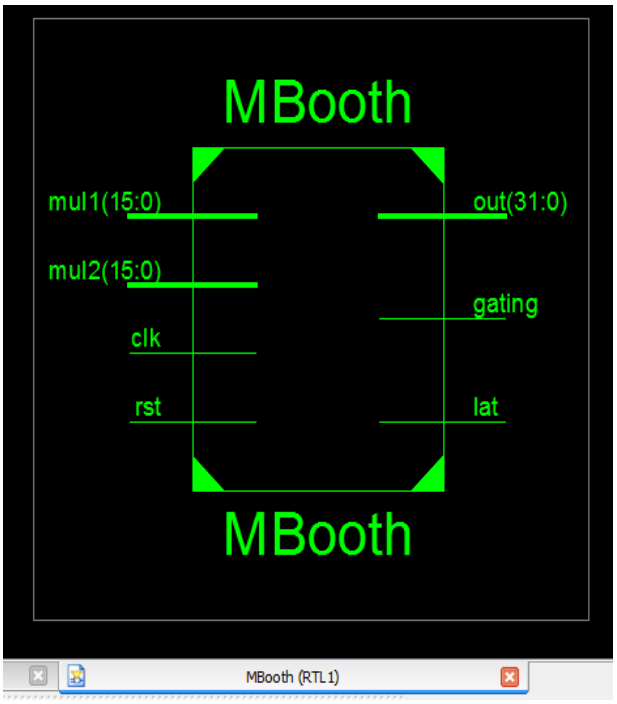

Fig 4 : RTL Schematic of proposed model.

```
Device utilization summary:
---------------------------

Selected Device : 3s100ecp132-5

 Number of Slices:                      415  out of    960    43%
 Number of Slice Flip Flops:            277  out of   1920    14%
 Number of 4 input LUTs:                686  out of   1920    35%
    Number used as logic:               679
    Number used as Shift registers:       7
 Number of IOs:                         68
 Number of bonded IOBs:                 68  out of     83    81%
 Number of GCLKs:                        1  out of     24     4%

---------------------------
```

Fig 5: Device Utilization Summary Report using Modified Booth Algorithm.

V. Conclusion And Future Work

Use A modified radix-4 Booth multiplier design is to yield less number of partial products at output of multiplier. Apart from this, Booth algorithm considers the two's complement of given input number making multiplication of signed/ negative number as simple as positive one. Due to these advantages, there is considerable reduction in amount of area taken by multiplier circuit in the system making system compact, less delay and maximizing throughput. We can extend this work by employing Radix-8 Booth algorithm for partial products generation. Expected outcome is less number of partial products, reduced area & reduced delay.

Booth algorithm considers the two's complement of given input number making multiplication of signed/ negative number as simple as positive one. We can extend this work by employing Radix-8 Booth algorithm for partial products generation. Future work can be carried out to reduce electro migration effect and multiplier design can be extended to any number of input/output combinations. And eexpected outcome is less number of partial products, reduced area & reduced delay.

REFERENCES

[1] Ing-Chao Lin, Member, IEEE, Yu-Hung Cho, and Yi-Ming Yang, "Aging-aware reliable multiplier design with adaptive hold logic", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, vol. 23, no. 3, Mar. 2015

[2] Y. Cao. (2013). Predictive Technology Model (PTM) and NBTI Model [Online]. Available: http://www.eas. asu.edu/ptm

[3] S. Zafaret al., "A comparative study of NBTI and PBTI (charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates," in Proc.IEEESymp. VLSI Technol. Dig. Tech. Papers, 2006, pp. 23–25.

[4] S. Zafar, A. Kumar, E. Gusev, and E. Cartier, "Threshold voltage instabilities in high-k gate dielectric stacks," IEEE Trans. Device Mater.Rel., vol. 5, no. 1, pp. 45–64, Mar. 2005.

[5] H.-I. Yang, S.-C.Yang, W. Hwang, and C.-T. Chuang, "Impacts of NBTI/PBTI on timing control circuits and degradation tolerant designin nanoscale CMOS SRAM," IEEE Trans.Circuit Syst., vol. 58, no. 6, pp. 1239–1251, Jun. 2011.

[6] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pMOS NBTI effect for robust naometer design," in Proc. ACM/IEEEDAC, Jun. 2004, pp. 1047– 1052.

[7] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTIaware synthesis of digital circuits," in Proc. ACM/IEEE DAC, Jun. 2007, pp. 370–375.

[8] A. Calimera, E. Macii, and M. Poncino, "Design techniqures for NBTItolerant power-gating architecture," IEEE Trans. Circuits Syst., Exp.Briefs, vol. 59, no. 4, pp. 249–253, Apr. 2012.

[9] K.-C. Wu and D. Marculescu, "Joint logic restructuring and pin reordering against NBTI-induced performance.

[10] Y.-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance" optimization using variable-latency design style," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 10, pp. 1874–1883, Oct. 2011.

[11] Y. Chen et al., "Variable-latency adder (VL- Adder) designs for low power and NBTI tolerance," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 11, pp. 1621–1624, Nov. 2010

Shubhangi Ramannawar, is pursuing M.Tech. (VLSI) from Department of Electronics & Communication Engineering, (VIST, Bhopal) Under Rajeev Gandhi Prodyogiki Visvavidhyalaya Bhopal, Madhya Pradesh, India. Her research interests include Low power and high performance VLSI design.

Deepak Kumar received the B.E. degree in Electronics and Communication Engineering from the Rajeev Gandhi Prodyogiki Visvavidhyalaya Bhopal, Madhya Pradesh, India in 2010, M.Tech. from Dept. of Electronics Engineering, School of Engineering and Technology, Pondicherry University, Puducherry, India. He is currently working as Assistant Professor(VIST, Bhopal) Under Rajeev Gandhi Prodyogiki Visvavidhyalaya Bhopal, Madhya Pradesh, India. His research interests include Low power and high performance VLSI design, Integrated Circuits & Embedded Systems with Hardware DSP Implementation.