

A Modified Architecture for Radix-4 Booth Multiplier with Adaptive Hold Logic

S.Suvarna^{#1}, K. Rajesh^{*2}, T. Radhu^{^3}

[#]PG Scholar (VLSI Design), Department of ECE, Knowledge Institute of Technology, Salem
¹ssuvarnabe@gmail.com

^{**}Assistant Professor, Department of ECE, Knowledge Institute of Technology, Salem

²krece@kiot.ac.in

³trece@kiot.ac.in

Abstract- High speed digital multipliers are most efficiently used in many applications such as Fourier transform, discrete cosine transforms, and digital filtering. The throughput of the multipliers is based on speed of the multiplier, and then the entire performance of the circuit depends on it. The pMOS transistor in negative bias cause negative bias temperature instability (NBTI), which increases the threshold voltage of the transistor and reduces the multiplier speed. Similarly, the nMOS transistor in positive bias cause positive bias temperature instability (PBTI). These effects reduce the transistor speed and the system may fail due to timing violations. So here a new multiplier was designed with novel adaptive hold logic (AHL) using Radix-4 Modified Booth Multiplier. By using Radix-4 Modified Booth Encoding (MBE), we can reduce the number of partial products by half. Modified booth multiplier helps to provide higher throughput with low power consumption. This can adjust the AHL circuit to reduce the performance degradation. The expected result will be reduce threshold voltage, increase throughput and speed and also reduce power. This modified multiplier design is coded by Verilog and simulated using Xilinx ISE 12.1 and implemented in Spartan 3E FPGA kit.

Keywords: Negative Bias Temperature Instability (NBTI), Positive Bias Temperature Instability (PBTI), Radix-4, Modified Booth Encoding, Adaptive Hold Logic (AHL), Xilinx ISE.

I. INTRODUCTION

Digital Multipliers used in Digital Signal Processing (DSP) like Fast Fourier Transform (FFT), Finite Impulse Response (FIR) filter, discrete cosine transforms (DCT) etc....required high speed and low power consumption. The throughput of these applications depends on multipliers and if the multipliers are too slow, the performance of entire circuits will be reduced. The negative bias temperature instability (NBTI) [7] occurs when a pMOS transistor is under negative bias ($V_{gs} = -V_{dd}$). In this situation, the interaction between inversion layer holes and hydrogen-passivized Si atoms breaks the Si-H bond generated during the oxidation process, generating H or H₂ molecules. When these molecules diffuse away interface traps are left. The accumulated interface traps between silicon and the gate oxide interface result in increased threshold voltage (V_{th}), reducing the circuit switching speed. When the biased voltage is removed, the reverse reaction occurs, reducing the NBTI effect. However, the reverse reaction does not eliminate all the interface traps generated during the stress phase and V_{th} is increased in the long term. Hence, it is important to design a reliable high-performance multiplier. The corresponding effect on an nMOS transistor is positive bias temperature instability (PBTI), which occurs when an nMOS transistor is under positive bias. Compared with the NBTI effect, the PBTI effect is much smaller on oxide or

poly-gate transistors and therefore is usually ignored. However, for high metal-gate nMOS transistors with significant charge trapping, the PBTI effect can no longer be ignored.

An efficient multiplier should have following characteristics [2]: Accuracy- A good multiplier should give correct result. Speed- Multiplier should perform operation at high speed. Area- A multiplier should occupy less number of slices and LUT's. Power- Multiplier should consume less power. For high-speed multiplication, the modified Booth algorithm (MBA) [5] is commonly used. For partial product reduction different types of adders are used. Booth multiplication allows for the smaller, faster multiplication circuits through encoding the signed bits to 2's complement which is also the standard technique in chip design. Although the partial products are further reduced by using higher radix (2, 4, 8, 16, and 32), Booth Encoder increases complexity and improves the performance. By increasing radix value, partial products will be reduced. Hence it will increase the speed. Low power consumption is also an important issue in multiplier design. To reduce significant power consumption it is good to reduce the number of operation thereby reducing dynamic power which is a major part of total power consumption. So the need of high speed and low power multiplier has increased.

A new reliable low power multiplier design by adopting algorithmic noise tolerant (ANT) architecture described by [3] with the fixed width multiplier to build the reduced precision replica redundancy block (RPR). The proposed ANT architecture can meet the demand of high precision, low power consumption, and area efficiency. We design the fixed width RPR with error compensation circuit via analyzing of probability and statistics. Using the partial product terms of input correction vector and minor input correction vector to lower the truncation errors, the hardware complexity of error compensation circuit can be simplified. In a 12×12 bit ANT multiplier, circuit area in our fixed width RPR can be lowered by 44.55% and power consumption in our ANT design can be saved by 23% as compared with the state-of-art ANT design.

An aging-aware multiplier design with novel adaptive hold logic (AHL) circuit [4]. Digital multipliers are among the most critical arithmetic functional units. The overall performance of these systems depends on the throughput of the multiplier. Meanwhile, the negative bias temperature instability effect occurs when a pMOS transistor is under negative bias ($V_{gs} = -V_{dd}$), increasing the threshold voltage of the pMOS transistor, and reducing multiplier speed. A similar phenomenon, positive bias temperature instability, occurs when an nMOS transistor is under positive bias. Both effects

degrade transistor speed, and in the long term, the system may fail due to timing violations. Therefore, it is important to design reliable high-performance multipliers. The multiplier is able to provide higher throughput through the variable latency and can adjust the AHL circuit to mitigate performance degradation that is due to the aging effect. Moreover, the proposed architecture can be applied to a column- or row-bypassing multiplier.

Approximate multiplier with an approximate 2bit adder [5] is deliberately designed for calculating the sum of 1 and 2 of a binary number. This adder requires a small area, a low power and a short critical path delay. Subsequently, the 2bit adder is employed to implement the less significant section of a recoding adder for generating the triple multiplicand with no carry propagation. In the pursuit of a tradeoff between accuracy and power consumption, two signed 16 bit approximate radix8 Booth multipliers are designed using the approximate recoding adder with and without the truncation of a number of less significant bits in the partial products. The proposed approximate multipliers are faster and more power efficient than the accurate Booth multiplier. Moreover, the multiplier with 15bit truncation achieves the best overall performance in terms of hardware and accuracy when compared to other approximate Booth multiplier designs. Finally, the approximate multipliers are applied to the design of a low pass FIR filter and they show better performance than other approximate Booth multipliers.

More, T.V [7] et al described a low power Column bypass multiplier design methodology that inserts more number of zeros in the multiplicand thereby reducing the number of switching activities as well as power consumption. It is well known that multipliers consume most of the power in DSP computations. Hence, it is very important for modern DSP systems to develop low power multipliers to reduce the power dissipation. In this paper, we presents low power Column bypass multiplier design methodology that inserts more number of zeros in the multiplicand thereby reducing the number of switching activities as well as power consumption. The switching activity of the component used in the design depends on the input bit coefficient. This means if the input bit coefficient is zero, corresponding row or column of adders need not be activated. If multiplicand contains more zeros, higher power reduction can be achieved. To reduce the switching activity is to shut down the idle part of the circuit, which is not in operating condition. Use of look up table is an added feature to this design.

A modified low power booth multiplier for high speed and low power applications described [8]. The design of normal multiplier consumes most of the power in DSP processors. In order to reduce the power consumption of multiplier, the low power Booth recoding methodology is implemented by recoding technique. This booth decoder will increase number of zeros in multiplicand. Booth multiplier has booth decoder to recode the given input to booth equivalent. Hence the number of switching activity will be reduced so the power consumption of the design can be reduced. The input bit coefficient will determine the switching activity of the component that is when the input coefficient is zero corresponding rows or column of the adder should be deactivated. When multiplicand contains more number of

zeros the higher power reduction can takes place. So in booth multiplier high power reductions will be achieved.

High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers [9]. This paper presents the design and implementation of signed unsigned Modified Booth Encoding (SUMBE) multiplier. The present Modified Booth Encoding (MBE) multiplier and the Baugh Wooley multiplier perform multiplication operation on signed numbers only. The array multiplier and Braun array multipliers perform multiplication operation on unsigned numbers only. Thus, the requirement of the modern computer system is a dedicated and very high speed unique multiplier unit for signed and unsigned numbers. Therefore, this paper presents the design and implementation of SUMBE multiplier. The modified Booth Encoder circuit generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the SUMBE multiplier is obtained. The Carry Save Adder (CSA) tree and the final Carry Look ahead (CLA) adder used to speed up the multiplier operation. Since signed and unsigned multiplication operation is performed by the same multiplier unit the required hardware and the chip area reduces and this in turn reduces power dissipation and cost of a system.

Srinivas K.B [10] et al described a bypassing logic will reduce Dynamic power dissipation as well as signal propagation delay. The demand for electronic portable devices is gaining more attention in recent decades. Portable devices are demanding for low power. Multiplier is the critical part of any arithmetic operation in many DSP applications. So it is essential to design multipliers that utilize less power and provide high speed of operation. One main aspect of low power design is to minimize switching activities to reduce dynamic power dissipation. Row and column bypass multiplier is a new design which reduces the switching activities with architecture optimization. The switching activity should not occur unnecessarily and it should be avoided by bypassing. The adders corresponding to those rows or columns which are required to be bypassed, need not be activated and signal get bypassed to the further stage. With the help of tri-state buffer as a control gating element, unnecessary signal propagation can be stopped. Thus, the unwanted switching activity can be reduced. The proposed multiplier design is efficient in terms of power by 20% or more when probability of occurrence of zeros is more. These features make the proposed design more suitable for DSP applications like filtering, Discrete Cosine Transform (DCT) and Fast Fourier Transform (FFT).

A new multiplier using modified Radix4 booth algorithm with redundant binary adder for low energy applications [11]. Multipliers are the building blocks of high performance systems like FIR filters, Digital signal processors, etc in which speed is the dominating factor. There are many multiplier architectures developed to increase the speed of algebra. Booth algorithm is the most effective algorithm used for fast performances. This works by introducing a high performance multiplier using Modified Radix4 booth algorithm with Redundant Binary Adder to get high speed. A comparative study of different booth algorithms in terms of power consumption, delay, and area, energy and energy delay product is also discussed in this work. All the circuits are simulated in the Cadence simulation tool using 180nm

technology. The experimental results show that the proposed booth multiplier shows high speed, low energy and low energy delay product compared to the existing booth multipliers.

II. BLOCK DIAGRAM

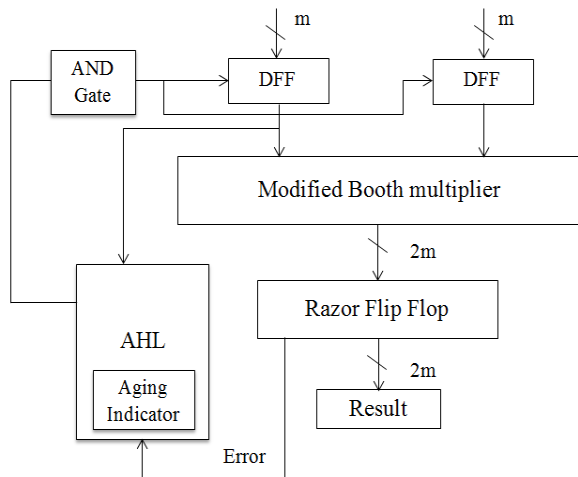


Figure 1: Block Diagram with AHL

III. DESCRIPTION

A new multiplier was designed with a novel adaptive hold logic (AHL) circuit as shown in figure 1. This multiplier includes two m -bit inputs (m is a positive number), one $2m$ -bit output, one Modified Booth Multiplier, $2m$ 1-bit Razor flip-flops, and an AHL circuit. The multiplier is able to provide higher throughput through the variable latency and can adjust the AHL circuit to mitigate performance degradation that is due to the aging effect. Moreover, the proposed architecture can be applied to a Modified Booth multiplier. 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. This architecture is performed in 16-bit and 32-bit multipliers and also be easily extended to large designs.

A Modified Booth Multiplier

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. 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 is performed by radix-4 booth encoder.

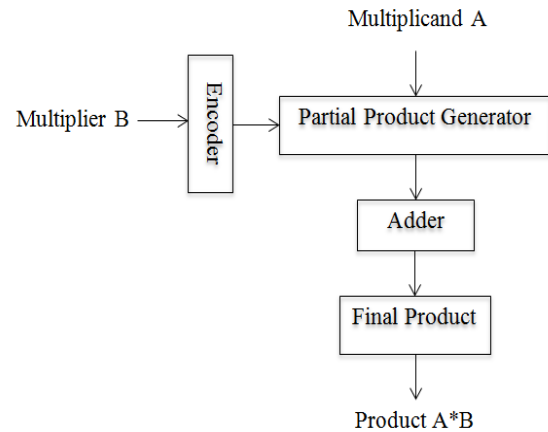


Figure 2: Flow Chart for Modified Booth Multiplier

B. Radix-4 Modified Booth Encoder

Radix-4 Modified Booth Encoder performs the process of encoding the multiplicand based on multiplier bits. It will compare 3 bits at a time with overlapping technique. Grouping starts from the LSB, and the first block only uses two bits of the multiplier and assumes a zero for the third bit. The detection unit has one of the two operands as its input to decide whether the Booth encoder calculates redundant computations. The functional operation of Radix-4 booth encoder is shown in the Table 1.

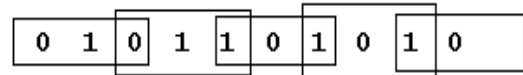


Figure 3: Grouping of bits from multiplier term

In Radix-4 Modified Booth algorithm, the number of partial products reduced by half. For multiplication of 2's complement numbers, the two bit encoding using this algorithm scans a triplet of bits. To Booth recode the multiplier term, 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 show the grouping of bits from the multiplier term in modified booth encoding.

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 1: RADIX-4 BOOTH ENCODING

Block	Recoded Digit	Operation on X
000	0	0X
001	+1	+1X
010	+1	+1X
011	+2	+2X
100	-2	-2X
101	-1	-1X
110	-1	-1X
111	0	0X

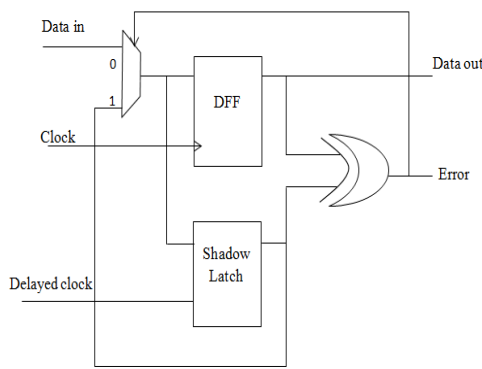


Figure 4: Razor Flip Flop

C. Razor Flip Flop

A 1-bit Razor flip-flop contains a main flip-flop, shadow latch, XOR gate, and multiplier as shown in figure 4. 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. Although the re-execution may seem costly, the overall cost is low because the re-execution frequency is low.

D. Adaptive Hold Logic(AHL)

The AHL circuit contains an aging indicator, two judging blocks, one mux, and one D flip-flop as shown in figure 5. The aging indicator indicates whether the circuit has suffered significant performance degradation due to the aging effect. The aging indicator is implemented in a simple counter that counts the number of errors over a certain amount of operations and is reset to zero at the end of those operations. If the cycle period is too short, Modified Booth Multiplier is not able to complete these operations successfully, causing timing violations. These timing violations will be caught by the Razor flip-flops, which generate error signals.

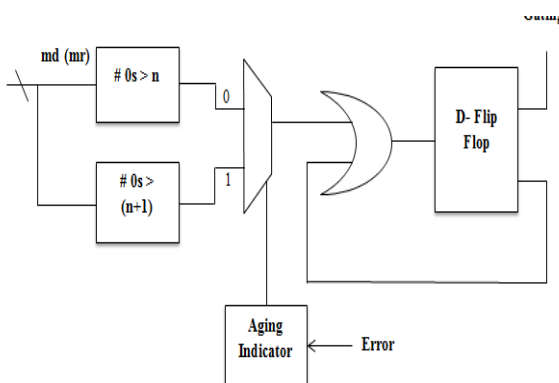


Figure 5: Adaptive Hold Logic (AHL)

If errors happen frequently and exceed a predefined threshold, it means the circuit has suffered significant timing degradation due to the aging effect, and the aging indicator will output signal 1, otherwise it will output 0 to indicate the aging effect is still not significant, and no actions are needed.

The first judging block in the AHL circuit will output 1 if the number of zeros in the multiplicand (multiplier for the row-bypassing multiplier) is larger than n (n is a positive number), and the second judging block in the AHL circuit will output 1 if the number of zeros in the multiplicand (multiplier) is larger than $n + 1$. They are both employed to decide whether an input pattern requires one or two cycles, but only one of them will be chosen at a time. In the beginning, the aging effect is not significant, and the aging indicator produces 0, so the first judging block is used. After a period of time when the aging effect becomes significant, the second judging block is chosen.

Compared with the first judging block, the second judging block allows a smaller number of patterns to become one-cycle patterns because it requires more zeros in the multiplicand. When the modified booth multiplier finishes the operation, the result will be passed to the Razor flip-flops. The Razor flip-flops check whether there is the path delay timing violation. If timing violations occur, it means the cycle period is not long enough for the current operation to complete and that the execution result of the multiplier is incorrect.

IV. RESULT AND DISCUSSION

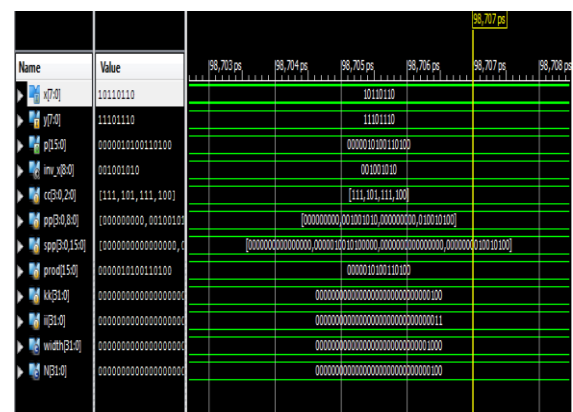


Figure 6: Simulation result for 8*8 booth multiplier

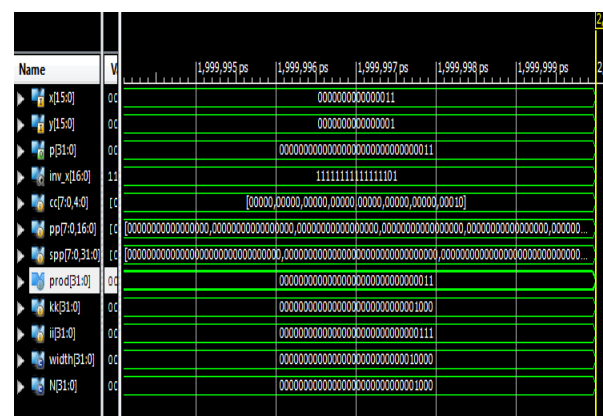


Figure 7: Simulation result for 16*16 booth multiplier

The simulation results for 8-bit and 16-bit booth multiplier using Radix-4 Modified Booth algorithm shown in figure 6

and 7. By using Radix-4 Booth algorithm the partial products are reduced by $n/2$. From figure 6, 4 rows of partial products only produced. From figure 7, 8 rows of partial products only produced. Also delay has been analyzed. Then these partial products are added and produced the 16 bit multiplied output. From figure 6 to 9, results have been obtained with the help of Xilinx 12.1 design suite



Figure 8: Simulation result for Razor Flip Flop

Figure 8 shows the output of razor Flip Flop using D-Flip Flop and Shadow Latch simulated using Xilinx 12.1. 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. Although the re-execution may seem costly, the overall cost is low because the re-execution frequency is low.

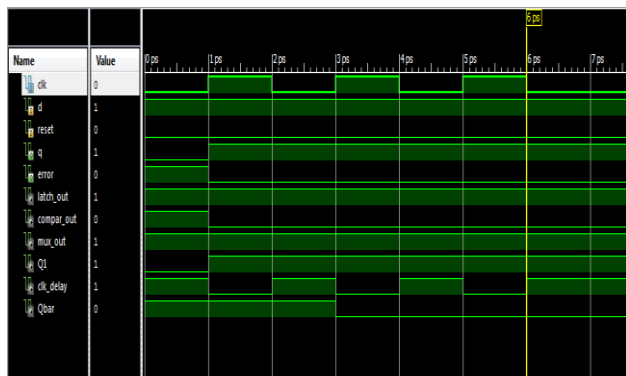


Figure 9: Simulation result for Adaptive Hold Logic

Figure 9 shows the output of Adaptive Hold Logic using D-Flip Flop simulated using Xilinx 12.1. The details of the operation of the AHL circuit are as follows: when an input pattern arrives, both judging blocks will decide whether the pattern requires one cycle or two cycles to complete 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.

V. CONCLUSION

High speed and low power multiplier design with Adaptive Hold Logic has been successfully simulated using Xilinx ISE Design 12.1. By using Radix-4 Modified Booth Algorithm, the number of partial products has been reduced by half, which improves the speed of the operation. In this architecture we can improve the performance of the multiplier and also to improve the speed of the multiplier. By using Adaptive Hold Logic (AHL), the timing violation will be reduced and power

consumption will also be reduced. So the overall performance will be improved.

REFERENCES

- [1] A. Calimera, E. Macii, and M. Poncino, (2012) "Design techniques for NBTI-tolerant power-gating architecture," IEEE Trans. Circuits Syst., Exp. Briefs, vol. 59, no. 4, pp. 249–253.
- [2] Cui, x.; Liu, W.; Chen, X.; Swartzlander, E.; Lombardi, F., (2015) "A Modified Partial Product Generator for Redundant Binary Multipliers," in Computers, IEEE Transactions on , vol. PP, no. 99, pp. 11
- [3] Ichyn Wey; ChienChang Peng; FengYu Liao, (2015) "Reliable Low Power Multiplier Design Using Fixed Width Replica Redundancy Block," in Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol. 23, no. 1, pp. 7887
- [4] Ing-Chao Lin, Yu-Hung Cho, Yi-Ming Yang, (2014) "Aging-Aware Reliable Multiplier Design With Adaptive Hold Logic" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS.
- [5] Jiang, H.; Han, J.; Qiao, F.; Lombardi, F., (2015) "Approximate Radix8 Booth Multipliers for Low Power and High performance Operation," in Computers, IEEE Transactions on , vol. PP, no. 99, pp. 11
- [6] Jiun-Ping Wang, Shiann-Rong Kuang, and Shish-Chang Liang, (2011) "High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 19, NO. 1.
- [7] More, T.V.; Kshirsagar, R.V., (2011) "Design of low power column bypass multiplier using FPGA," in Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol. 3, no., pp. 431435, 810.
- [8] Prabhu, A.S.; Elakya, V., (2012) "Design of modified low power booth multiplier," in Computing, Communication and Applications (ICCCA), 2012 International Conference on , vol., no., pp. 16, 2224.
- [9] Rajput, R.P.; Swamy, M.N.S., (2012) "High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers," in Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on , vol., no., pp. 649654, 2830.
- [10] Srinivas, K.B.; Aneesh, Y.M., (2014) "Low power and high speed row and column bypass multiplier," in Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on , vol., no., pp. 14, 1820.
- [11] Surendran, E.K.L.; Antony, P.R., (2014) "Implementation of fast multiplier using modified Radix4 booth algorithm with redundant binary adder for low energy applications," in Computational Systems and Communications (ICCSC), 2014 First International Conference on , vol., no., pp. 266271, 1718.
- [12] H.-I. Yang, S.-C. Yang, W. Hwang, and C.-T. Chuang, (2011) "Impacts of NBTI/PBTI on timing control circuits and degradation tolerant design in nanoscale CMOS SRAM," IEEE Trans. Circuit Syst., vol. 58, no. 6, pp. 1239–1251.
- [13] Yongho Lee, Taewhan Kim, (2011) "A Fine-Grained Technique of NBTI-Aware Voltage Scaling and Body Biasing for Standard Cell Based Designs" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS.
- [14] Yu-Shih Su, Da-Chung Wang, Shih-Chieh Chang, and Malgorzata Marek-Sadowska, (2011) "Performance Optimization Using Variable-Latency Design Style" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 19, NO. 10.
- [15] Yiran Chen, Hai Li, Cheng-Kok Koh, Guangyu Sun, Jing Li, Yuan Xie, and Kaushik Roy, (2010) "Variable-Latency Adder (VL-Adder) Designs for Low Power and NBTI Tolerance" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 18, NO. 11.