INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

DESIGNING OF AN AGING-AWARE RELIABLE MULTIPLIER BY USING CARRY LOOK-AHEAD ADDER

A. R. Gunasekhar[1], Sk. Mahaboob Basha[2]

[1] *M.Tech Scholar, Dept. of ECE Sree Vidyanikethan Engg. College (Autonomous), Tirupati,*
argunasekhar@gmail.com
[2] *Assistant Professor, Dept. of ECE Sree Vidyanikethan Engg. College Autonomous), Tirupati,*
shaik001122@gmail.com

Abstract: - 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. In this paper, we propose an aging-aware multiplier design with novel adaptive hold logic (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 column- or row-bypassing multiplier.

Keywords: - Adaptive hold logic (AHL), Negative Bias Temperature instability (NBTI), Positive Bias Temperature Instability (PBTI),

1. Introduction

Digital multipliers are among the most critical arithmetic functional units in many applications, such as the Fourier transform, discrete cosine transforms, and digital filtering. The through put of these applications depends on multipliers, and if the multipliers are too slow, the performance of entire circuits will be reduced. Furthermore, negative bias temperature instability (NBTI) occurs when a pMOS transistor is under negative bias ($V_{gs} = -V_{dd}$). In this situation, the interaction between inversion layer holes and hydrogen-passivated 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/polygate transistors, and therefore is usually ignored. However, for high-*k*/metal-gate nMOS transistors with significant charge trapping, the PBTI effect can no longer be ignored. In fact, it has been shown that the PBTI effect is more significant than the NBTI effect on 32-nm high-*k*/metal-gate processes [2]-[4].

A traditional method to mitigate the aging effect is overdesign [5], [6], including such things as guard-banding and life. These techniques, however, require circuit modification or do not provide optimization of specific methodologies have been proposed. An NBTI-aware technology mapping technique was proposed in to symmetries and transistor stacking effects. They also proposed an NBTI optimization method that considered path sensitization. In dynamic voltage scaling and body-basing techniques guarantee the performance of the circuit during its lifetime. In [8] an NBTI-aware sleep circuits. Gate oversizing; however, this approach can be very pessimistic and area and power inefficient. To avoid this problem, many NBTI-aware transistors were designed to reduce the aging effects on pMOS sleep-transistors, and the lifetime stability of the power-gated circuits under consideration was improved. Wu and Marculescu [9] proposed a joint logic restructuring and pin reordering method, which is based on detecting functional, were proposed to reduce power or extend circuit [10].

Traditional circuits use critical path delay as the overall circuit clock cycle in order to perform correctly. However, the probability that the critical paths are activated is low. In most cases, the path delay is shorter than the critical path. For these noncritical paths, using the critical path delay as the overall cycle period will result in significant timing waste. Hence, the variable-latency design was proposed to reduce the timing waste of traditional circuits. The variable-latency design divides the circuit into two parts: 1) shorter paths and 2) longer paths. Shorter paths can execute correctly in one cycle, whereas longer paths need two cycles to execute. When shorter paths are activated frequently, the average latency of variable-latency designs is better than that of traditional designs. For example, several variable-latency adders were proposed using the speculation technique with error detection and recovery. A short path activation function algorithm was proposed in to improve the accuracy of the hold logic and to optimize the performance of the variable-latency circuit. An instruction scheduling algorithm was proposed in to schedule the operations on non-uniform latency functional units and improve the performance of Very Long Instruction Word processors. In a variable-latency pipelined multiplier architecture with a Booth algorithm was proposed. In process-variation tolerant architecture for arithmetic units was proposed, where the effect of process-variation is considered to increase the circuit yield. In addition, the critical paths are divided into two shorter paths that could be unequal and the clock cycle is set to the delay of the longer one. These research designs were able to reduce the timing waste of traditional circuits to improve performance, but they did not consider the aging effect and could not adjust themselves during the runtime. A variable-latency adder design that considers the aging effect was proposed. However, no variable-latency multiplier design that considers the aging effect and can adjust dynamically has been done.

1.1 Paper Contribution

In this paper, we propose an aging-aware reliable multiplier design with novel adaptive hold logic (AHL) circuit. 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; 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.

2. Preliminaries

2.1 Row Bypassing Multiplier

A low-power row-bypassing multiplier [11] is also proposed to reduce the activity power of the AM. The operation of the low-power row-bypassing multiplier is similar to that of the low-power column-bypassing multiplier, but the selector of the multiplexers and the tristate gates use the multiplicator.

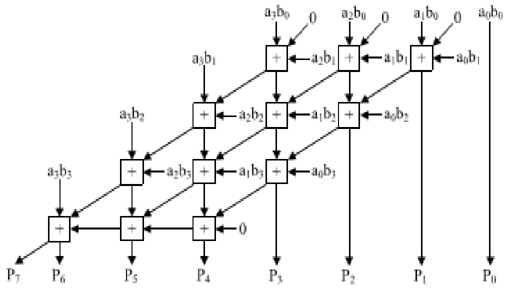


Figure. 1. 4 × 4 normal AM.

Fig. is a 4 ×4 normal Array multiplier. Each input is connected to an FA through a tristate gate. When the inputs are 1111_2 * 1001_2, the two inputs in the first and second rows are 0 for FAs. Because b_1 is 0, the multiplexers in the first row select a_ib_0 as the sum bit and select 0 as the carry bit. The inputs are bypassed to FAs in the second rows, and the tristate gates turn off the input paths to the FAs.

Therefore, no switching activities occur in the first-row FAs; in return, power consumption is reduced. Similarly, because $b_2$is 0, no switching activities will
Occur in the second-row FAs. However, the FAs must be active in the third row because the b_3 is not zero.

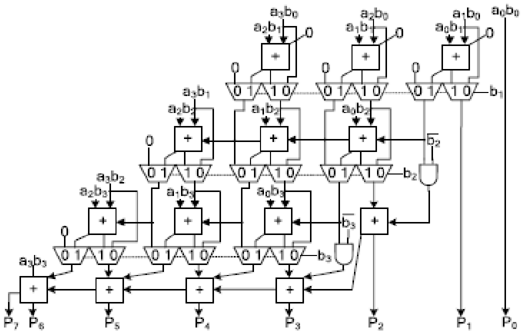


Figure. 2. 4 X 4 Row by passing multiplier

Selector of the multiplexer to decide the output of the FA, and a_i can also be used as the selector of the tristate gate to turn off the input path of the FA. If a_i is 0, the inputs of FA are disabled, and the sum bit of the current FA is equal to the sum bit from its upper FA, thus reducing the power consumption of the multiplier. If a_i is 1, the normal sum result is selected. More details for the column-bypassing multiplier can be found.

2.2 Variable - Latency Design

In ripple carry adders, the carry propagation time is the major speed limiting factor as seen in the previous lesson.

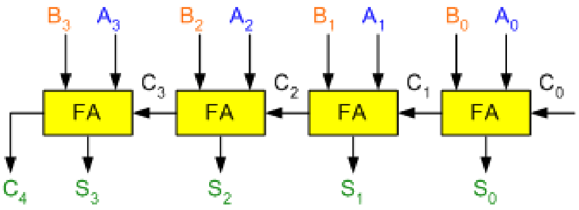


Figure. 3. 4-Bit Ripple Carry Adder

Most other arithmetic operations, e.g. multiplication and division are implemented using several add/subtract steps. Thus, improving the speed of addition will improve the speed of all other arithmetic operations.

Accordingly, reducing the carry propagation delay of adders is of great importance. Different logic design approaches have been employed to overcome the carry propagation problem.

This type of adder circuit is called as carry look-ahead adder (CLA adder).It is based on the fact that a carry signal will be generated in two cases:

(1) When both bits A_i and B_i are 1, or
(2) When one of the two bits is 1 and the carry-in (carry of the previous stage) is 1.

The Boolean expression of the carry outputs of various stages can be written as follows:

$C_1 = G_0 + P_0C_0$
$C_2 = G_1 + P_1C_1 = G_1 + P_1 (G_0 + P_0C_0)$
$= G_1 + P_1G_0 + P_1P_0C_0$
$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$
$C_4 = G_3 + P_3C_3$
$= G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$

In general, the $i^{th.}$ carry output is expressed in the form $\mathbf{C_i} = F_i$ (P's, G's, $\mathbf{C_0}$).
In other words, each carry signal is expressed as a direct SOP function of $\mathbf{C_0}$ rather than its preceding carry signal.
Since the Boolean expression for each output carry is expressed in SOP form, it can be implemented in two-level circuits.

The 2-level implementation of the carry signals has a propagation delay of 2 gates, i.e., 2τ.
The 4-bit carry look-ahead (CLA) adder consists of 3 levels of logic: First

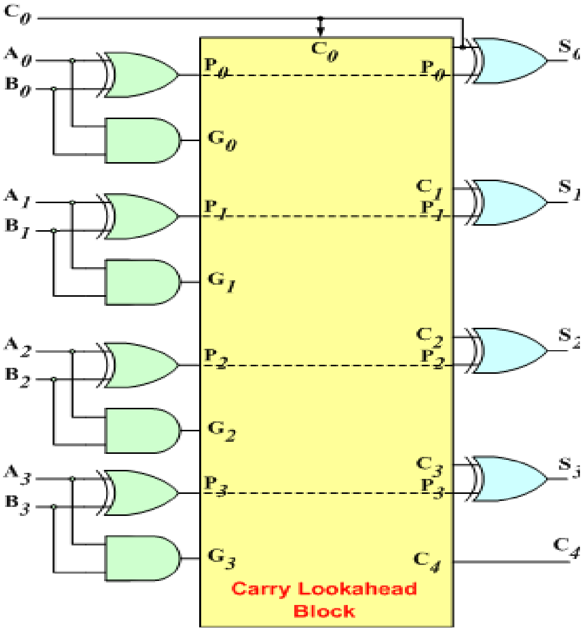


Figure. 4. 4- Bit Carry Look Ahead Adder

First level: Generates all the P & G signals. Four sets of P & G logic (each consists of an XOR gate and an AND gate). Output signals of this level (P's & G's) will be valid after 1τ.

Second level: The Carry Look-Ahead (CLA) logic block which consists of four 2-level implementation logic circuits. It generates the carry signals (C_1, C_2, C_3, and C_4) as defined by the above expressions. Output signals of this level (C_1, C_2, C_3, and C_4) will be valid after 3τ.
Third level: Four XOR gates which generate the sum signals (S_i) ($S_i = P_i \oplus C_i$). Output signals of this level (S_0, S_1, S_2, and S_3) will be valid after 4τ.

Thus, the 4 Sum signals (S_0, S_1, S_2 & S_3) will all be valid after a total delay of 4τ compared to a delay of (2n+1)□τ for Ripple Carry adders.
For a 4-bit adder (n = 4), the Ripple Carry adder delay is 9τ.
The disadvantage of the CLA adders is that the carry expressions (and hence logic) become quite complex for more than 4 bits.

3. Proposed Aging-Aware Multiplier

The proposed aging-aware reliable multiplier design. It introduces the overall architecture and the functions of each component and also describes how to design AHL that adjusts the circuit when significant aging occurs.

3.1 Proposed Architecture

proposed aging-aware multiplier architecture, which includes two *m*-bit inputs (*m* is a positive number), one 2*m*-bit output, one column- or row-bypassing multiplier, 2*m* 1-bit Razor flip-flops[12], and an AHL circuit.

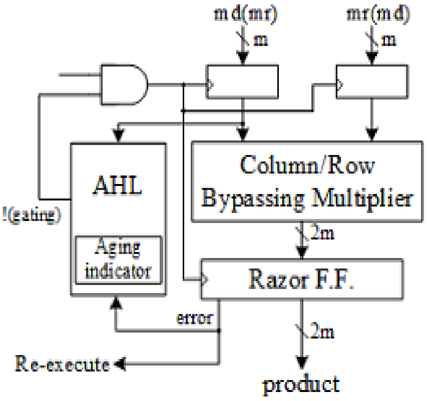


Figure. 5. Proposed architecture (md means multiplicand; mr means multiplicator).

Hence, the two aging-aware multipliers can be implemented using similar architecture, and the difference between the two bypassing multipliers lies in the input signals of the AHL. According to the bypassing selection in the column or row-bypassing multiplier, the input signal of the AHL in the architecture with the column-bypassing multiplier is the multiplicand, whereas of the row-bypassing multiplier is the multiplicator. Razor flip-flops can be used to detect.

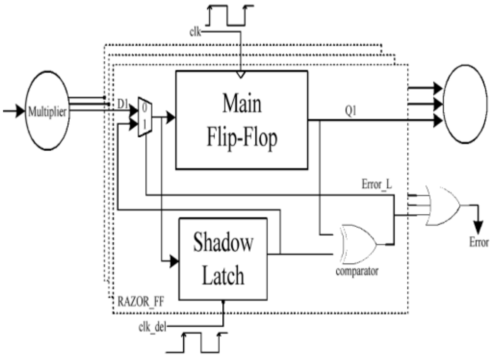


Figure. 6. Internal Block Diagram of Razor flip flop.

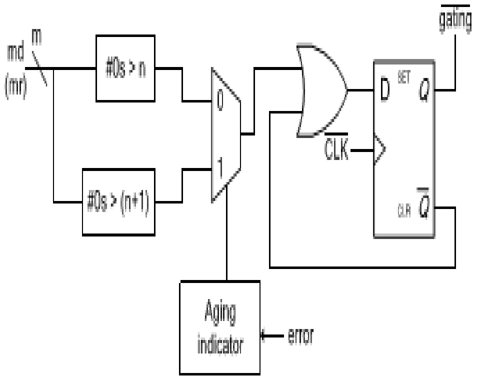


Figure. 7. Diagram of AHL

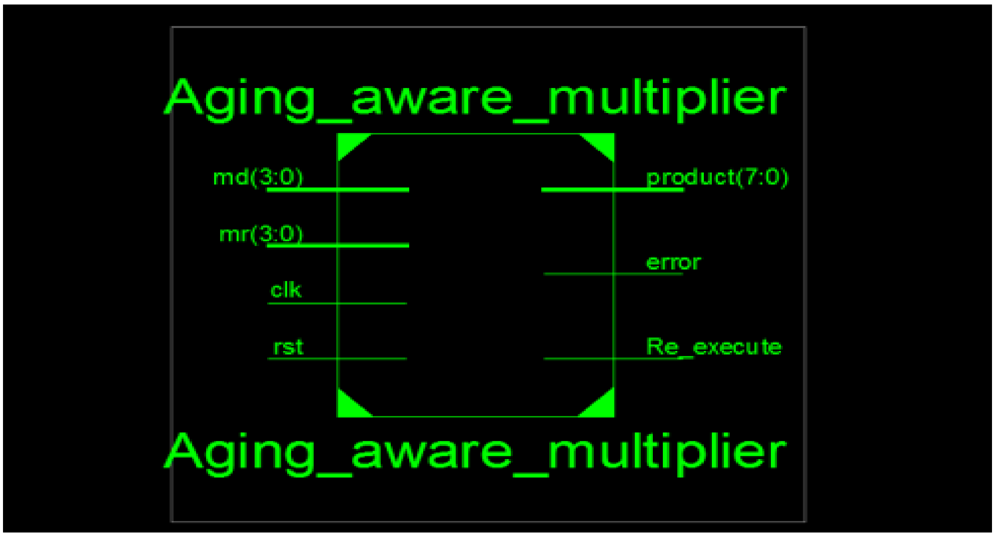
Whether timing violations occur before the next input pattern arrives.

The details of Razor flip-flops. 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.

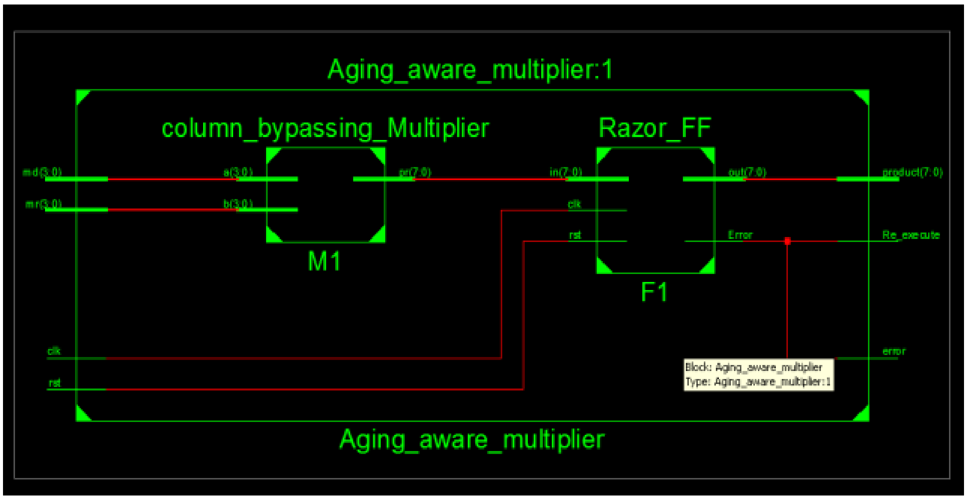
To notify the system to re-execute the operation and notify the AHL circuit that an error has occurred. We use Razor flip-flops to detect whether an operation that is considered to be a one-cycle pattern can really finish in a cycle. If not, the operation is re-executed with two cycles. Although the re execution may seem costly, the overall cost is low because the re execution frequency is low. More details for the Razor flip-flop can be found [12]. The AHL circuit is the key component in the aging-ware variable-latency multiplier. Fig. shows the details of the AHL circuit. The AHL circuit contains an aging indicator, two judging blocks, one mux, and one D flip-flop. 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, the column- or row-bypassing 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.

4. Experimental Results

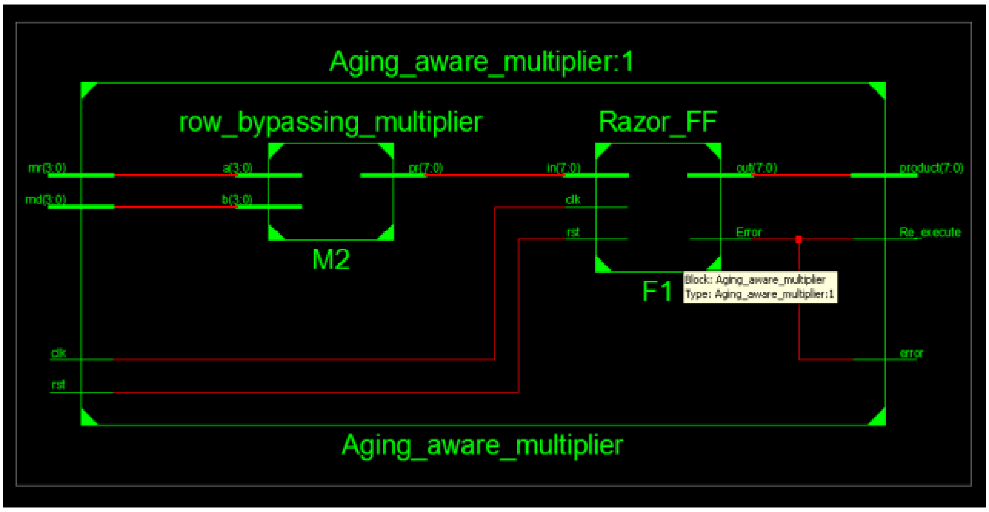
4.1. Block Diagram



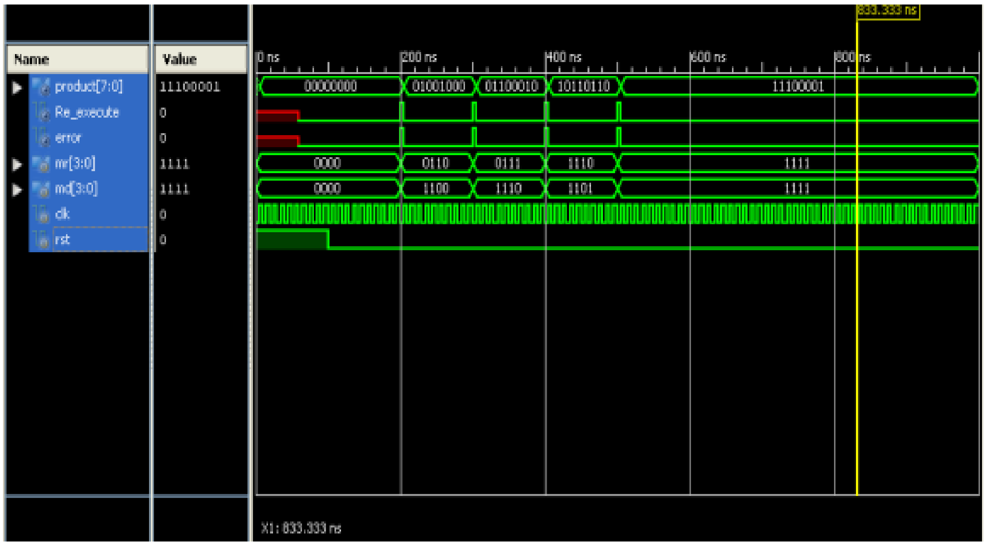
4.2. Rtl Schematic Diagram



4.3. Technology Schematic



4.4. Simulation Output Waveform



5. Conclusion

This paper proposed an aging-aware variable-latency multiplier design with the AHL. The multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. The experimental results show that our proposed architecture with 4x4 multiplications with CLA as last stage instead of Normal RCA adder it will decrease the delay and improve the performance compared with previous designs.

REFERENCES

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

[2] S. Zafar *et al.*, “A comparative study of NBTI and PBTI (charge trapping) in SiO_2/HfO_2 stacks with FUSI, TiN, Re gates,” in *Proc. IEEE Symp. VLSI Technol. Dig. Tech. Papers*, 2006, pp. 23–25.

[3] 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.

[4] H.-I. Yang, S.-C. Yang, W. Hwang, and C.-T. Chuang, “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, Jun. 2011.

[5] R. Vattikonda, W. Wang, and Y. Cao, “Modeling and miimization of pMOS NBTI effect for robust naometer design,” in *Proc. ACM/IEEE DAC*, Jun. 2004, pp. 1047–1052.

[6] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, “NBTI-aware flip-flop characterization and design,” in *Proc. 44th ACM GLSVLSI*, 2008, pp. 29–34

[7] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, “NBTI-aware 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 NBTI-tolerant 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 degradation,” in *Proc. DATE*, 2009, pp. 75–80.

[10] Y. Lee and T. Kim, “A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs,” in *Proc. ASP-DAC*, 2011, pp. 603–608.

[11] J. Ohban, V. G. Moshnyaga, and K. Inoue, “Multiplier energy reduction through bypassing of partial products,” in *Proc. APCCAS*, 2002, 13–17.

[12] D. Ernst *et al.*, “Razor: A low-power pipeline based on circuit-level timing speculation,” in *Proc. 36th Annu. IEEE/ACM MICRO*, Dec. 2003, 7–18

A R Gunasekhar – received the B.Tech. Degree in 2012 in Electronics and Communication Engineering from SITAMS College, Chittoor. He is currently pursuing his M.tech in the specialization of VLSI in ECE Department of Sree Vidyanikethan Engineering College (Autonomous), Tirupathi and would graduate in the year 2015. His research interests include Digital Logic Design, VLSI and FPGA.

Sk. Mahaboob Basha – is currently working as an Assistant Professor in the ECE Department of Sree Vidyanikethan Engineering college (Autonomous), Tirupathi. He has completed his M.Tech in VLSI Design, in Satyabhama University. His research area of interest includes Digital System Design and VLSI Signal Processing.