IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

A Review on aging aware Reliable Multiplier

Sushant Jalindar Sawant[1], Prof. Sanjay S. Badhe[2]

M.E. Student (VLSI & Embedded Systems), E&TC Department, D.Y. Patil College of Engineering, Pune, India[1]

Department of E&TC Engg, D.Y. Patil College of Engineering, Ambi, Pune, India[2]

Abstract: Advanced Digital multipliers are the most critical arithmetic functional units. The general execution of these systems depends on the throughput of the multiplier. While negative bias temperature instability (NBTI) effects on logic gates are of major concern for the reliability of digital circuits, they become even more critical when considering the components for which even minimal parametric variations affect the lifetime of the overall circuit. In the mean time, the negative bias temperature instability effect occurs when a pMOS transistor is under negative bias (V gs=−Vdd), increasing the threshold voltage of the pMOS transistor, and decresing multiplier speed. A similar phenomenon, positive bias temperature instability, happens when an nMOS transistor is under positive bias. Both effects or impacts reduce transistor speed, and in the long term, the system may fail because of timing infringement. Therefore, it is important to design reliable high-performance multipliers. In this paper, we propose an aging-aware multiplier design with a novel adaptive hold logic (AHL) circuit. The multiplier can give higher throughput through the variable latency and can adjust the AHL circuit to mitigate performance degradation that is due to the aging effect.

Keywords: Adaptive hold logic (AHL), negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), reliable multiplier, variable latency.

I. INTRODUCTION

Multiplication is an basic arithmetic operation for common DSP applications, for example, filtering and fast Fourier transform (FFT). To accomplish high execution speed, parallel array multipliers are broadly utilized. These multipliers tend to consume most of the power in DSP calculations, and along these power-efficient multipliers are very important for the design of low-power DSP systems. If the multipliers are too slow, the execution of whole circuits will be reduced. Besides, negative bias temperature instability (NBTI) occurs when a pMOS transistor is under negative bias (V gs = −V dd). In this condition, the interaction between inversion layer holes and hydrogen-passivated Si atoms breaks he Si–H bond created during the oxidation process and producing H or H 2 molecules. At the point when these molecules diffuse away, interface traps arecleared out. The accumulated interface traps between silicon and the gate oxide interface result in increased expanded limit voltage (V th) and reducing the circuit switching speed. At the point When the biased voltage is removed, the reverse reaction happens, decreasing the NBTI impact. In any case, the reverse reaction does not avoid all the interface traps created during the stress phase, and V th is expanded in the long term. Hence, it is important to design a reliable high-performance multiplier. The corresponding impact on an nMOS transistor is positive bias temperature instability (PBTI), which happens 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 [1]–[2]. Traditional circuits use critical path delay as the overall circuit clock cycle in order to perform effectively. 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, utilizing the critical path delay as the overall cycle period will result in significant timing waste. Hence, the variable-latency design was proposed to decrease the timing waste of traditional circuits. The variable-latency design separates the circuit into two parts: 1) shorter paths and 2) longer paths. Shorter paths can execute accurately in one cycle, whereas longer paths require two cycles to execute. At the point When shorter paths are initiated much of the time, the average latency of variable-latency designs is superior than that of traditional designs. For example, several variable-latency adders were proposed using the speculation technique with error detection and recovery [3]–[4]. A short path activation function algorithm was proposed in [16] to improve the accuracy of the hold logic and to enhance the execution of the variable-latency circuit. An instruction scheduling algorithm was proposed in [5] to schedule the operations on nonuniform latency functional units and enhance the performance of Very Long Instruction Word processors. These research designs were able to decrease the timing waste of traditional circuits to improve performance, however they didn't consider the aging effect and couldn’t adjust themselves during the runtime. A variable-latency adder design that considers the aging effect was proposed in [6] and [7]. Be that as it may, no variable-latency multiplier design that considers the aging effect and can adjust progressively has been finished.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

II. LITERATURE SURVEY

1. 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.
While negative bias temperature instability (NBTI) effects on logic gates are of major concern for the reliability of digital circuits, they become even more critical when considering the components for which even minimal parametric varieties affect the lifetime of the overall circuit. PMOS header transistors utilized in power-gated architectures are one relevant example of such components. For these types of devices, an NBTI-induced current capability degradation converts into a larger IR-drop impact on the virtual-Vddrail, which genuinely influences the performance and, in this way, the dependability of allpower-gated cells. In this short, they address the issue of designing NBTI-tolerant power-gating architectures. It propose a set of efficient NBTI-aware circuit design solutions, including both staticand dynamicstrategies, that go for enhancing the lifetime stability of power-gated circuits by means of oversizing, body biasing, and stress-probability decrease while limiting the design overheads.

2. 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.
In many designs, the worst-case delay of a critical path might be activated infrequently. Traditional optimization approaches assume the worst-case situation, which could lead to an inefficient resource usage. It is conceivable to enhance the throughput of such designs by presenting variable latency. The design of the hold logic in telescopic units impacts the circuit's throughput. In this paper, we demonstrate that the generally designed hold logic may be inaccurate. They utilize the short path activation conditions to obtain more accurate hold logic and improve the efficiency of telescopic units. To decrease the overhead for large circuits, we propose an efficient heuristic methodology of constructing non-exact hold logic. We also discuss how to choose the telescopic unit's timing constraint.

3. S. Zafar et al., "A comparative study of NBTI and PBTI (charge trapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates," in Proc.IEEE Symp. VLSI Technol. Dig. Tech. Papers, 2006, pp. 23–25.
Threshold voltage (V_t) of a field effect transistor (FET) is observed to shift with stressing time and this stress induced V_t shift is an important transistor reliability issue. V_t shifts that happen under negative gate bias is referred as NBTI and those that occur under positive bias is referred as PBTI or charge trapping. In this paper, we present a comparative study of NBTI and PBTI for a variety of FETs with various dielectric stacks and gate materials. The study has two parts. In part I, NBTI and PBTI measurements are performed for FUSI NiSi gated FETs with SiO_2 SiO_2/HfO_2 and SiO_2/HfSiO as gate dielectric stacks and the results are compared with those for conventional SiON/poly-Si FETs. NBTI is observed to be independent of gate material whereas PBTI is fundamentallymore regrettable for FUSI gated devices.

4. Hao-I. Yang, Shyh-Chyi Yang, Wei Hwang, and Ching-Te Chuang "Impacts of NBTI/PBTI on Timing Control Circuits and Degradation Tolerant Design in Nanoscale CMOS SRAM",IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 58, NO. 6, JUNE 2011 1239
They examine the effects of NBTI and PBTI on the stability and WM of a two-port 8T SRAM design. They then give extensive analyses on the degradations of SRAM Read access and Write performance with hierarchical bit-line and Read/Write replica timing control circuits . They show that because the Read/Write replica timing control circuits are activated in every Read/Write cycle, they exhibit distinctively different degradation behavior from the normal array access paths, resulting in degradation of timing control and performance. Further, we show that raising the Standby Virtual Ground voltage of the 8T cell's Read buffer mitigates the Read performance degradation, and data-retention power-gating techniques and dual- 8T cell can be utilized to mitigate the stability degradation of bit cells. Hierarchical Read/Write scheme can enhance the efficiency of these systems.

5. J. Ohban, V. G. Moshnyaga, and K. Inoue, "Multiplier energy reduction through bypassing of partial products," in Proc. APCCAS, 2002, pp. 13–17. For a low-power row-bypassing multiplier the operations in the j-th row can be disabled to reduce the power dissipation if the bit bj in the multiplier is 0. In the multiplier design, each modified FA in the CSA array is attached by three tri-state buffers and two 2-to-1 multiplexers. Since the addition operations of the rightmost FAs in the CSA rows are able to be bypassed, the additional correcting circuits must be added to correct the multiplication result.

III. RELATED WORK

In this paper, we propose an aging-aware reliable multiplier design with a novel adaptive hold logic (AHL) circuit. The multiplier is depends on the variable-latency procedure and can adjust the AHL circuit to accomplish reliable operation

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

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 design with an AHL circuit. The AHL circuit can choose whether the input patterns require one or two cycles and can change the judging criteria to ensure that there is minimum performance degradation after considerable aging happens;
2) comprehensive analysis and comparison of the multiplier's performance under various cycle periods to demonstrate the effectiveness of our proposed architecture;
3) an aging-aware reliable multiplier design method that is appropriate for large multipliers. Although the experiment is performed in 16- and 32-bit multipliers, our proposed architecture can be easily extended to large designs;

A) Column-Bypassing Multiplier
A column-bypassing multiplier is an improvement on the normal array multiplier (AM). The AM is a fast parallel AM and is shown in Fig.1(a). The multiplier array consists of (n − 1) rows of carry save adder (CSA), in which each row contains (n − 1) full adder (FA) cells. Each FA in the CSA array has two outputs: 1) the sum bit goes down and 2) the carry bit goes to the lower left FA. The last row is a ripple adder for carry propagation. The FAs in the AM are always active regardless of input states.

In [8], a low-power column-bypassing multiplier design is proposed in which the FA operations are disabled if the corresponding bit in the multiplicand is 0. Fig.1(b) shows a 4 × 4 column-bypassing multiplier. Supposing the inputs are 1010 2 * 1111 2 , it can be seen that for the FAs in the first and third diagonals, two of the three input bits are 0: the carry bit from its upper right FA and the partial product a i b i . In this way, the output of the adders in both diagonals is 0, and the output sum bit is basically equal to the third bit, which is the sum output of its upper 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,

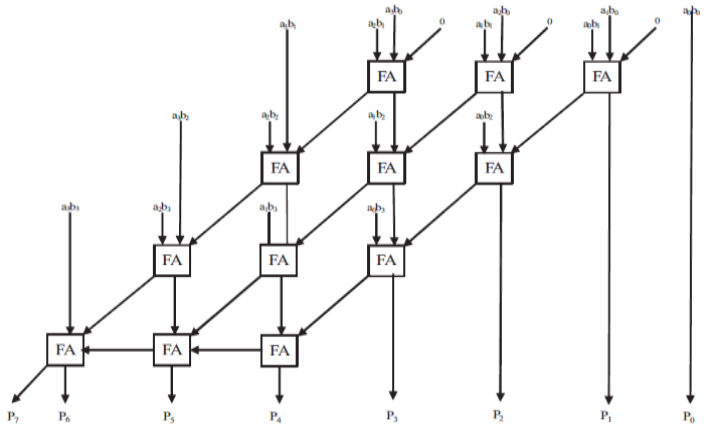

Fig1. (a) 4 × 4 normal AM

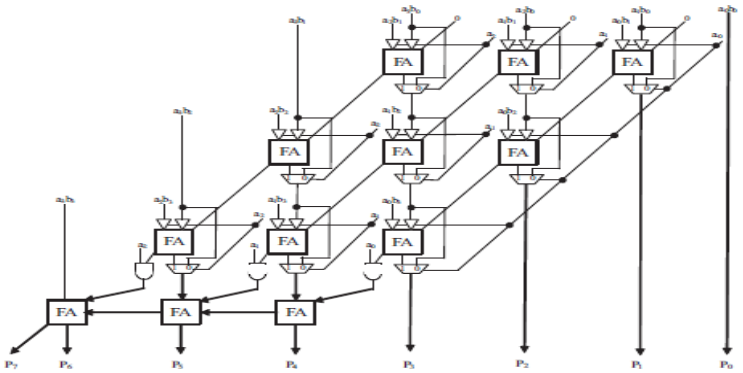

Fig1.(b)
Fig1. (b) 4 × 4 column-bypassing multiplier.

In this way lessening the power utilization of the multiplier. If a i is 1, the normal sum result is selected. More points of interest for the column-bypassing multiplier can be found in [8].

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

B. Row-Bypassing Multiplier

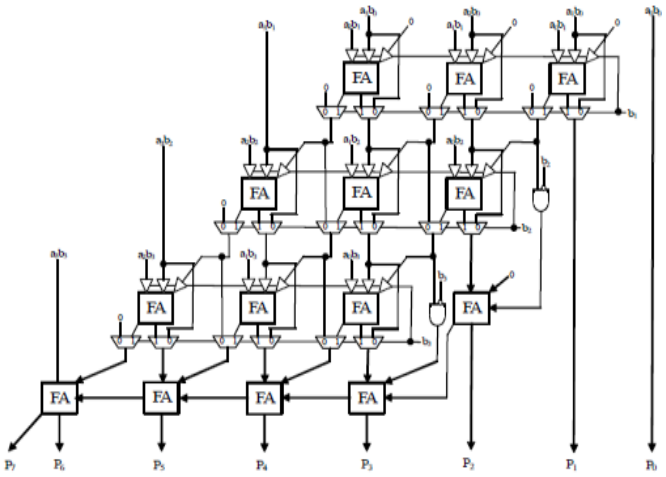

Fig 2. 4 × 4 row-bypassing multiplier

A low-power row-bypassing multiplier [9] is also proposed to decrease the activity power of the AM. The operation of the low-power row-bypassing multiplier is like that of the low-power column-bypassing multiplier, yet the selector of the multiplexers and the tristate gates utilize the multiplicator. Fig. 2 is a 4 × 4 row-bypassing 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 i b 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. In this way, no switching activities occur in the first-row Fas; consequently, power utilization is decreased. Similarly, because b 2 is 0, no switching activities will happen in the second-row Fas.

C. Variable-Latency Design

The variable-latency design was proposed to decrease the timing waste occurring in traditional circuits that use the critical path cycle as an execution cycle period. The basic concept is to execute a shorter path using a shorter cycle and longer path using two cycles. Since most paths execute in a cycle period that is much smaller than the critical path delay, the variable-latency design has smaller average latency.

IV. PROPOSED AGING -AWARE MULTIPLIER

This section details the proposed aging-aware reliable multiplier design. It presents the overall architecture and the functions of each component and also describes how to design AHL that adjusts the circuit when significant aging occurs. Fig. 3 shows our proposed aging-aware multiplier architecture, which consists of two m-bit inputs (m is a positive number), one 2m-bit output, one column- or row-bypassing multiplier, 2m 1-bit Razor flip-flops [10], and an AHL circuit. The inputs of the row-bypassing multiplier are the symbols in the parentheses.

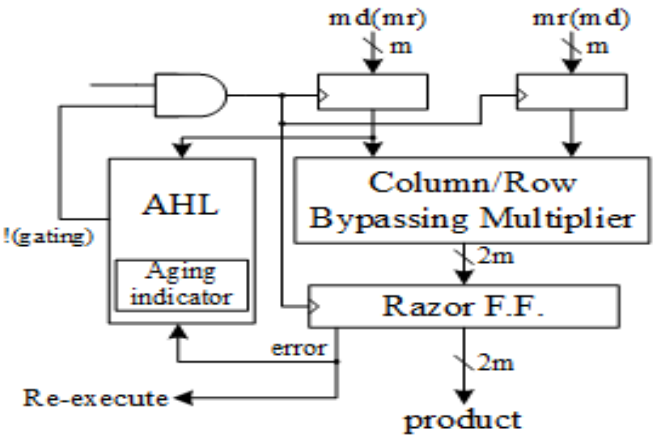

Fig.3 Proposed architecture

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

In the proposed architecture, the column- and row-bypassing multipliers can be examined by the number of zeros in either the multiplicand or multiplicator to predict whether the operation requires one cycle or two cycles to complete. At the point When input patterns are irregular, the number of zeros and ones in the multiplicator and multiplicand follows a normal distribution.

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 that of the row-bypassing multiplier is the multiplicator. Razor flip-flops can be used to detect whether timing violations happens before the next input pattern arrives.

Fig.4 shows 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 reexecute 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 reexecuted with two cycles. Although the reexecution may seem costly, the overall cost is low because the reexecution frequency is low. More details for the Razor flip-flop can be found in [10].

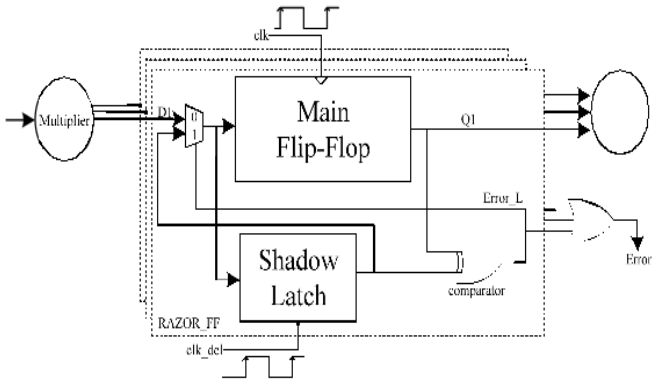

Fig.4. Razor flip-flops

The aging indicator indicates whether the circuit has suffered significant execution degradation due to the aging effect. The aging indicator is implemented in a basic 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. 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 demonstrate the aging effect is still not significant, and no actions are required.

The overall flow of our proposed architecture is as per the following: when input patterns arrive, the column- or row-bypassing multiplier, and the AHL circuit execute simultaneously. According to the number of zeros in the multiplicand (multiplicator), the AHL circuit chooses if the input patterns require one or two cycles. If the input pattern needs two cycles to complete, the AHL will output 0 to disable the clock signal of the flip-flops. Something else, the AHL will output 1 for normal operations. When the column- or row-bypassing 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. Thus, the Razor flip-flops will output an error to inform the system that the current operation needs to be reexecuted using two cycles to ensure the operation is correct. In this situation, the extra reexecution cycles caused by timing violation incurs a penalty to overall average latency. However, our proposed AHL circuit can precisely predict whether the input patterns require one or two cycles in most cases.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

V. EXPERIMENTAL RESULT

A. Average Latency Comparison

In the variable-latency design, the average latency is affected by both the percentage of one-cycle patterns and the cycle period. If more patterns only need one cycle, the average latency is decreased. Similarly, if the cycle period is reduced, the average latency is also reduced. However, the cycle period cannot be too small. If the cycle period is too small, large amounts of timing violations will be detected by the Razor flip-flops, and the average latency will increase. Therefore, it is important to analyze the tradeoff between the percentage of one-cycle patterns and the cycle period. To achieve this, we analyze three scenarios for both 16 × 16 and 32 × 32 variable latency column-bypassing (VLCB) and variable-latency row bypassing (VLRB) multipliers. We also compare the results with the AM, a FLCB multiplier, and a fixed-latency row bypassing (FLRB) multiplier.

We first compare the average latency of the AM, FLCB, FLRB, VLCB, and VLRB multipliers in both 16 × 16 and 32 × 32 multipliers. In the case of the 16 × 16 variable-latency bypassing multiplier, there are three scenarios: 1) Skip-7; 2) Skip-8; and 3) Skip-9. For VLCB multipliers, Skip-7 means the patterns that have seven or more zeros in the multiplicand are regarded as one-cycle patterns, and Skip-8 means the patterns that have eight or more zeros in the multiplicand are regarded as one cycle patterns, and so on. Similarly, for VLRB multipliers, Skip-7 means the patterns that have seven or more zeros in the multiplicator are regarded as one-cycle patterns, and Skip-8 means the patterns that have eight or more zeros in the multiplicator are regarded as one-cycle patterns, and so on.

Fig. 5 compares the average latency between the AM, FLCB, FLRB, adaptive variable-latency column bypassing (A-VLCB), and adaptive variable-latency row bypassing (A-VLRB) multipliers without aging in the 16 × 16 multiplier. To simplify the comparison, the results are divided into three parts: 1) Skip-7; 2) Skip-8; and 3) Skip-9. The latency of the AM, FLRB, and FLCB is shown to be 1.32, 1.82, and 1.88, respectively.

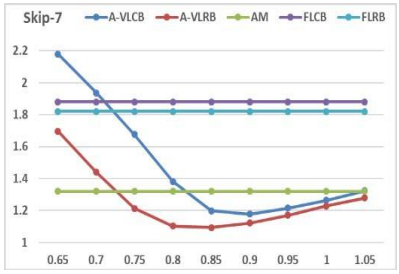

Fig (a)

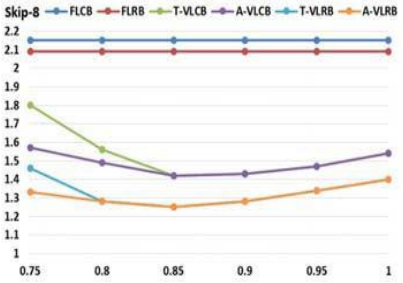

Fig (b)

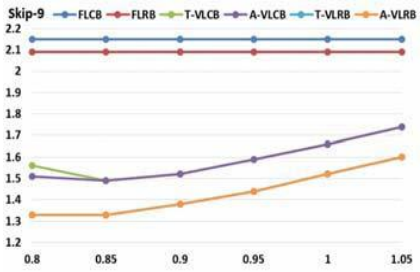

Fig (c)

Fig. 5 compares the average latency between the AM, FLCB, FLRB, adaptive variable-latency column bypassing (A-VLCB), and adaptive variable-latency row bypassing (A-VLRB) multipliers without aging in the 16 × 16 multiplier.

In Fig. 5(a), when the skip number is 7, the latency of the A-VLCB is 37.3% less than the FLCB and 10.7% less than the AM when the cycle period is 0.9. The latency of the A-VLRB is 39.9% less than the FLRB and 17.2% less than the AM when the cycle period is 0.85. In Fig. 5(b), when the skip number is 8, the latency of the A-VLCB is 32.2% less

than the FLCB and 3.4% less than the AM when the cycle period is 0.8. The latency of the A-VLRB is 35.5% less than the FLRB and 11.1% less than the AM when the cycle period is 0.8. In Fig. 5(c), when the skip number is 9, the latency of the A-VLCB is 28.8% less than the FLCB, but 1.3% higher than the AM when the cycle period is 0.8. The latency of the A-VLRB is 32.0% less than the FLRB and 6.3% less than the AM when the cycle period is 0.7.
At the point the system clock period is smaller than the preferred range, the average latency is expanded because more timing violations happens, and more operations are reexecuted in smaller cycle periods. Also, when the system clock period is bigger than the preferred range, the average latency is increased because more timing waste happens. To achieve an average latency lower than that of fixed-latency multipliers, it is essential to coordinate the system cycle period with the multiplier's preferred cycle period. If both do not match, techniques, such as transistor using another skip number, can be utilized to adjust the multiplier's cycle period.

B) Average Latency Improved by AHL
Our proposed architecture design transforms one-cycle patterns that cause timing infringement to two-cycle patterns by choosing another judging block in the AHL circuit. At the point When the frequency of errors exceeds a threshold (10% in our experiment, that is, 10 errors for each 100 operations), the aging indicator outputs 1 to choose another multiplexer input. In this condition, assuming initially, a multiplicand in the column-bypassing multiplier with n zeros is a one-cycle pattern, a multiplicand must have n + 1 zeros to become a one-cycle pattern, reducing the number of one-cycle patterns whose delay is close to the cycle period. For the row-bypassing multiplier, also, the criteria of one-cycle patterns must change from n zeros to n + 1 zeros in the multiplicator. It can be seen the latency of variable latency bypassing with adaptive hold is equivalent or superior than that of the variable latency bypassing multiplier. The improved ratio is large when the cycle period is short since all the more timing infringement occur in shorter cycle periods.

VI. CONCLUSION

This paper proposed an aging-aware variable-latency multiplier design with the AHL. The multiplier is able to adjust the AHL to alleviate performance degradation due to increased delay. The experimental results demonstrate that our proposed architecture with 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 FLCB multipliers, respectively. Moreover, our proposed architecture with the 16 × 16 and 32 × 32 row-bypassing multipliers can accomplish up to 80.17% and 69.40% performance improvement compared with the 16 × 16 and 32 × 32 FLRB multipliers. Also, the variable-latency bypassing multipliers exhibited the lowest average EDP and accomplished up to 10.45% EDP reduction in 32 × 32 VLCB multipliers. Electromigration happens when the current density is high enough to cause the drift of metal ions along the direction of electron flow. Since, our proposed variable latency multipliers should be used under the influence of both the BTI impact and electromigration. Also, our proposed variable latency multipliers have less performance degradation because variable latency multipliers have less timing waste, but traditional multipliers required to consider the degradation caused by both the BTI impact and electromigration and utilize the worst case delay as the cycle period.

REFERENCES

[1] 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.
[2] 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.
[3] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. DATE, 2012, pp. 1257–1262.
[4] D. Baneres, J. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in Proc. DATE, 2009, pp. 1704–1709.
[5] 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.
[6] N. V. Mujadiya, "Instruction scheduling on variable latency functional units of VLIW processors," in Proc. ACM/IEEE ISED, Dec. 2011, pp. 307–312
[7] Y. Chen, H. Li, J. Li, and C.-K. Koh, "Variable-latency adder (VL-Adder): New arithmetic circuit design practice to overcome NBTI," in Proc. ACM/IEEE ISLPED, Aug. 2007, pp. 195–200.
[8] 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.

BIOGRAPHY

Sushant Jalindar Sawant received the B.E degree in Electronics & Tele-Communication Engineering from D.Y. Patil College of Engineering in 2012.He is completing his M.E in VLSI & Embedded systems from Dr. D. Y. Patil College of Engineering, Ambi.