Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

AGING EFFECT TOLERANT MULTIPRECISION RAZOR-BASED MULTIPLIER

Kaviya.A[1],Darani Kumar.M[2], Rajasekar.P[3]
[1]*Assisitant professor, ECE,Kathir College of Engineering*
[2]*Assisitant professor, ECE, Karpagam University*
[3]*Assisitant professor, ECE, Kathir College of Engineering*

Abstract— Digital world multipliers are 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 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 a razor based multiplier circuit. The multiplier is able to provide higher throughput through the variable latency and can adjust the razor flip flop 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. The experimental results show that our proposed architecture with 16 × 16 and 32 × 32 column-bypassing multipliers can attain up to 62.88% and 76.28% performance improvement, respectively, compared with 16×16 and 32×32 fixed-latency column-bypassing multipliers. Furthermore, 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.

Keywords— Adaptive hold logic (AHL), Negative bias temperature instability (NBTI), Positive bias temperature instability (PBTI), Reliable multiplier, Variable latency, Razor flip flop variable-latency column-bypassing (VLCB) and variable-latency row-bypassing (VLRB) fixed-latency row-bypassing (FLRB) multiplier.

I. INTRODUCTION

Digital world multipliers are the most critical arithmetic functional units in many applications, such as the Fourier transform, discrete cosine transforms, and digital filtering. The throughput 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, gen-erating H or H_2 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.

A traditional method to mitigate the aging effect is overde-sign [5], [6], 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 [7] to guarantee the performance of the circuit during its lifetime. In [8], an NBTI-aware sleep transistor was 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 symmetries and transistor stacking effects. They also proposed an NBTI optimization method that considered path sensitization [12]. In [10] and [11], dynamic voltage scaling and body-basing techniques were proposed to reduce power or extend circuit life. These techniques, however, require circuit modification or do not provide optimization of specific circuits.

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 [13]–[15]. A short path activation function algorithm was proposed in [16] 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 [17] to schedule the operations on non uniform latency functional units and improve the performance of Very Long Instruction Word processors. In [18], a variable-latency pipelined multiplier architecture with a Booth algorithm was proposed. In [19], 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 in [20] and [21]. However, no variable-latency multiplier design that considers the aging effect and can adjust dynamically has been done.

A. Paper Explanation

In this paper, we propose an razor flip flop design with a 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 per-formance degradation after considerable aging occurs.

2) Comprehensive analysis and comparison of the multi-plier'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.

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 improve-ment as compared with 16×16 and 32×32 fixed-latency row-bypassing multipliers.

II. PRELIMINARIES

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. 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.

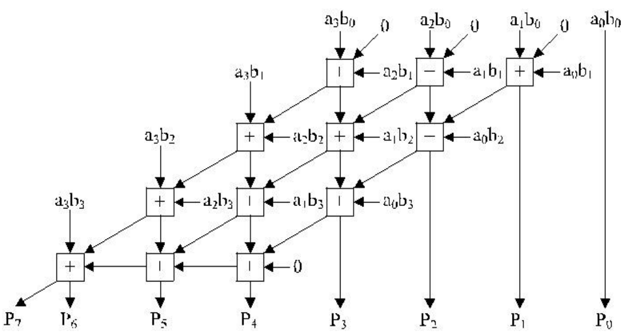

Fig. 1. 4 × 4 normal AM

The FAs in the AM are always active regardless of input states. In [22], 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. 2 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 $a_i b_i$. Therefore, the output of the adders in both diagonals is 0, and the output sum bit is simply equal to the third bit, which is the sum output of its upper FA.

Hence, the FA is modified to add two tristate gates and one multiplexer. The multiplicand bit a_i can be used as the 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 in [22].

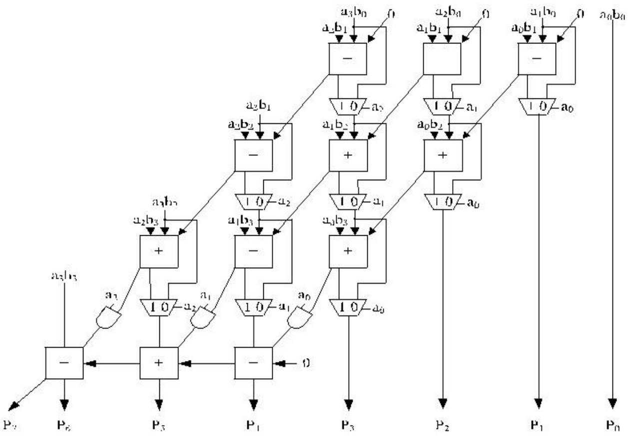

Fig. 2. 4 × 4 column-bypassing multiplier

B. Row-Bypassing Multiplier

A low-power row-bypassing multiplier [23] 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.

Fig. 3 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. 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. More details for the row-bypassing multiplier can also be found in [23].

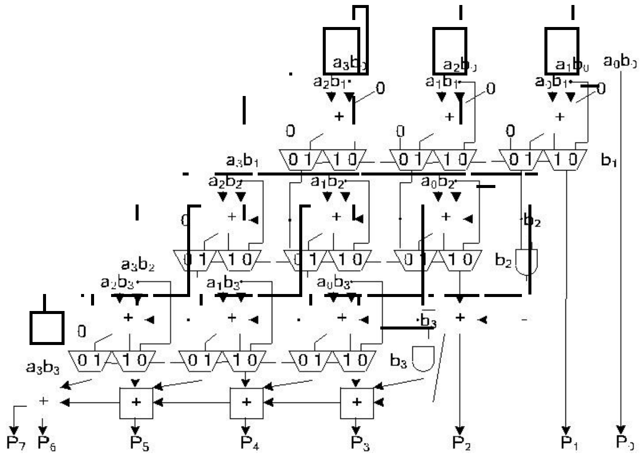

Fig. 3. 4 × 4 row-bypassing multiplier.

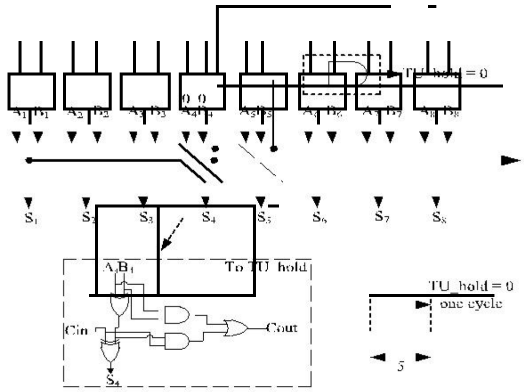

Fig. 4. 8-bit RCA with a hold logic circuit.

C. Aging Model

As mentioned in Section I, the NBTI (PBTI) effect occurs when a pMOS (nMOS) transistor is under negative (positive) bias voltage, resulting in V_{th} drift. When the bias voltage is removed, the recovery process occurs, reducing the V_{th} drift.

If a pMOS (nMOS) transistor is under constant stress, this is referred to as static NBTI (PBTI). If both stress and recovery phases exist, it is referred to as dynamic NBTI (PBTI). The V_{th} drift of pMOS (nMOS) transistor due to the static NBTI (PBTI) effect can be described by dc reaction-diffusion (RD) framework. If transistors are under alternative stress and recovery phases, the dc RD model should be modified to an ac RD model [24], [25] where α is a function of stress frequency (f) and signal probability (S). Since the impact of frequency is relatively insignificant, the effect of signal frequency is ignored. K_{DC} is a technology-dependent constant where A is a constant, and T_{OX} is the oxide thickness. E_{OX} is the gate electric field, which is $(V_{GS} - V_{th}) / T_{OX}$; k is the Boltzmann constant, and T is the temperature. E_0 and E_a are technology-independent characteristics of the reaction that are equal to 1.9–2.0 MV/cm and 0.12 eV, respectively.

In this paper, we use 32-nm high-k metal gate models. We set the temperature at 125 °C in our simulation and use the above BTI model to predict the BTI effect on the circuits. Fig. 7 shows the simulated delays of the 16 × 16 column-and row-bypassing multipliers under a seven-year NBTI/PBTI effect. From this figure, it can be seen that the BTI effect increased the critical path circuit delay by 13%. Hence, if the BTI effect is not considered during circuit design, the increased delay may cause system failure in the long term.

III. PROPOSED AGING-AWARE MULTIPLIER

This section details the proposed aging-aware reliable razor flip flop 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.

A. Proposed Architecture

Fig. 5 shows our proposed aging-aware multiplier archi-tecture, which includes 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 [27], and an AHL circuit.

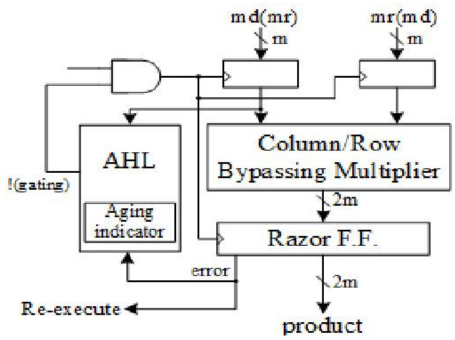

Fig. 5. Architecture of AHL (md means multiplicand; mr means multipli-cator).

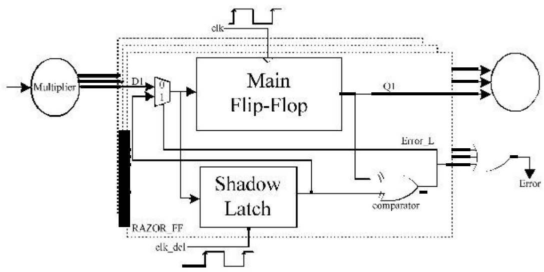

Fig. 6. Razor flip flops.

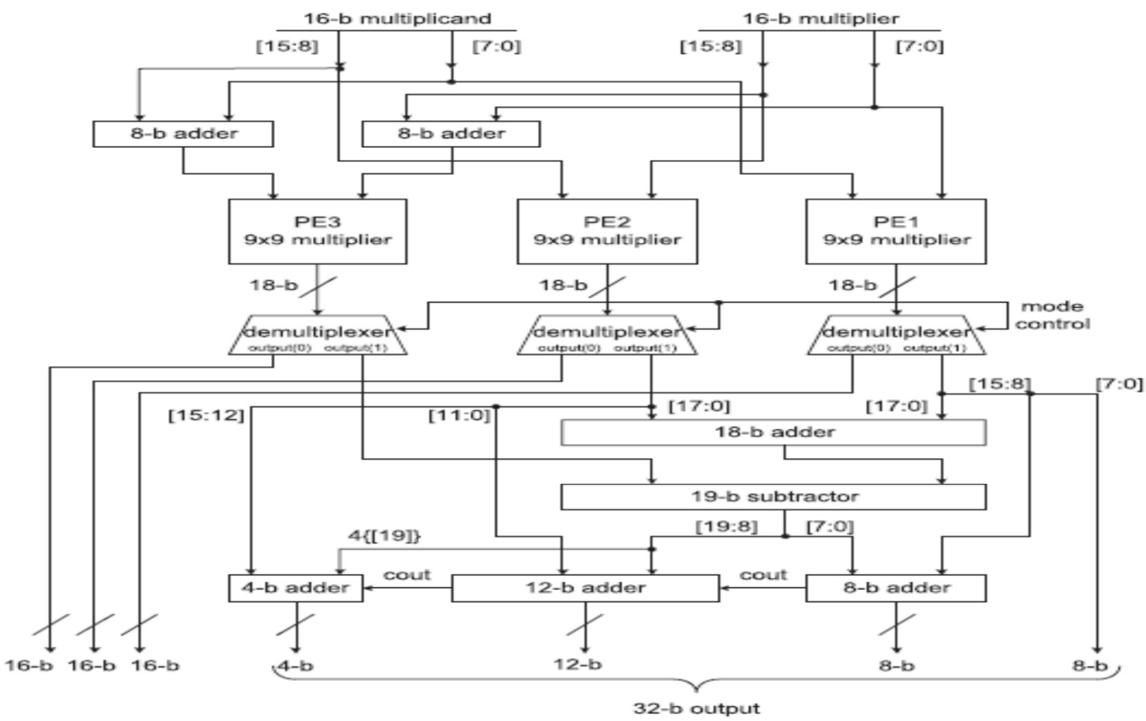

Fig.7 Proposed architecture of razor flip flop with AHL

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. When input patterns are random, the number of zeros and ones in the multiplicator and multiplicand follows a normal distribution, as shown in Figs. 5 and 6. Therefore, using the number of zeros or ones as the judging criteria results in similar outcomes.

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 occur before the next input pattern arrives.

Fig. 6 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 [27].

The AHL circuit is the key component in the aging-ware variable-latency multiplier. Fig. 12 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. 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.

The first judging block in the AHL circuit will output 1 if the number of zeros in the multiplicand (multiplicator for the row-bypassing multiplier) is larger than n (n is a positive number, which will be discussed in Section IV), and the second judging block in the AHL circuit will output 1 if the number of zeros in the multiplicand (multiplicator) 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 (multiplicator).

The overall flow of our proposed architecture is as follows: 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 decides if the input patterns require one or two cycles. If the input pattern requires two cycles to complete, the AHL will output 0 to disable the clock signal of the flip-flops. Otherwise, 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 accurately predict whether the input patterns require one or two cycles in most cases. Only a few input patterns may cause a timing variation when the AHL circuit judges incorrectly. In this case, the extra reexecution cycles did not produce significant timing degradation.

In 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 reexecute 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.

IV. EXPERIMENTAL RESULT

Our experiments are conducted in a Linux operating system. We adopt a 32-nm high-*k* predictive technology model [1] to estimate the BTI degradation for seven years. The proposed multiplier is designed in Verilog and converted to SPICE files using SpringSoft Laker. Then Synposys Nanosim is used to analyze the delay and power of the circuit. The V_{th} drift caused by BTI is estimated using the BTI model proposed in Section II-D and is added into the SPICE files during simulation.

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 require one cycle, the average latency is reduced. 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. Hence, 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.

	16x16 VLCB	16x16 VLRB
Skip-7	73.58%	77.39%
Skip-8	53.78%	59.89%
Skip-9	33.22%	40.20%,

TABLE 1 ONE-CYCLE PATTERN RATIO IN 16 × 16 MULTIPLIER

	32x32 VLCB	32x32 VLRB
Skip-15	66.46%	66.99%
Skip-16	52.68%	52.74%
Skip-17	38.18%	38.42%

TABLE 2 ONE-CYCLE PATTERN RATIO IN 32 × 32 MULTIPLIER

V. CONCLUSION

This paper proposed THE multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. The multiplier is based on the variable-latency with razor flip flop technique. The AHL circuit to achieve reliable operation under the influence of NBTI and PBTI effects. Our proposed architecture with the 16×16 column-bypassing multipliers and row-bypassing multipliers. The efficiency of the proposed architecture will be increased. our proposed variable latency multipliers have less performance degradation because variable latency multipliers have less timing waste.

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 reorder-ing 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] M. Basoglu, M. Orshansky, and M. Erez, "NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime," in Proc. ACM/IEEE ISLPED, Aug. 2010, pp. 253–258.
[12] K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and opti-mization considering path sensitization," in Proc. DATE, 2011, pp. 1–6.
[13] K. Du, P. Varman, and K. Mohanram, "High performance reli-able variable latency carry select addition," in Proc. DATE, 2012,pp. 1257–1262.
[14] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. DATE, 2008, pp. 1250–1255.
[15] D. Baneres, J. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in Proc. DATE, 2009, pp. 1704–1709.
[16] Y.-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Per-formance" optimization using variable-latency design style," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 10, pp. 1874–1883, Oct. 2011.
[17] M. Olivieri, "Design of synchronous and asynchronous variable-latency pipelined multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 9, no. 4, pp. 365–376, Aug. 2001.
[18] D. Mohapatra, G. Karakonstantis, and K. Roy, "Low-power process-variation tolerant arithmetic units using input-based elastic clocking," in Proc. ACM/IEEE ISLPED, Aug. 2007, pp. 74–79.
[19] 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.
[20] 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.
[21] M.-C. Wen, S.-J. Wang, and Y.-N. Lin, "Low power parallel mul-tiplier with column bypassing," in Proc. IEEE ISCAS, May 2005, pp1638–1641.
[22] J. Ohban, V. G. Moshnyaga, and K. Inoue, "Multiplier energy reduc-tion through bypassing of partial products," in Proc. APCCAS, 2002, pp.13–17
[23] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits," IEEE Electron Device Lett., vol. 26, no. 8, pp. 560–562, Aug. 2005.
[24] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Neg-ative bias temperature instability: Estimation and design for improved reliability of nanoscale circuit," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 26, no. 4, pp. 743–751, Apr. 2007.
[25] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pMOS NBTI effect for robust nanometer design," in Proc. 43rd ACM/IEEE DAC, Aug. 2006, pp. 1047–1052.
pp. 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.

