ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

A Survey on Aging-Aware Reliable Multiplier Design Using Adaptive Hold Logic Techniques

Pranali G. Gahukar, Prof.Monali Chinchmalatpure

Student, Department of E&TC, Dhole Patil College of Engg.Pune, Savitribai Phule Pune University, Pune, India.

Professor, Department of E&TC Dhole Patil College of Engg.Pune, Savitribai Phule Pune University, Pune, India.

ABSTRACT: Multiplication is one of the basic functions used in digital signal processing (DSP). Hardware resources and processing time required by it are more than addition and subtraction. Digital multipliers are the most critical functional block in arithmetic operations; however its performance is depends on its throughput. The negative bias temperature instability for pMOS and positive bias temperature instability of nMOS causes the timing violation and degrades the performance of digital multipliers. This papers reviews the adaptive hold logic technique to design aging aware reliable multiplier.

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

I. INTRODUCTION

Multiplier is an essential functional block of a microprocessor because multiplication is needed to be performed repeatedly in almost all scientific calculations. The fast and low power multipliers are required in small size wireless sensor networks and many other DSP (Digital Signal Processing) applications. Different computer arithmetic techniques can be used to implement a digital multiplier. Out of these most techniques involve computing a set of partial products, and then summing the partial products together. Until the 1970s, most minicomputers did not have a multiply instruction however, Mainframe computers had multiply instructions, but they did the some sorts of shifts and adds as a "multiply routine". Early microprocessors also had no multiply instruction. Then the Motorola 6809, introduced in 1978, was one of the earliest microprocessors with a dedicated hardware multiply instruction. It did the same sorts of shifts and adds as a "multiply routine", but the difference is that implementation was done in the microcode of the MUL instruction.The through put of these digital multipliers depends on multipliers, 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 (Vgs = −Vdd). Traditional method to mitigate the aging effect is overdesign including such things as guard-banding and gate over sizing; 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 into guarantee the performance of the circuit during its lifetime.In 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 Marcules cu proposed a point 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. In 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.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

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. A traditional method to mitigate the aging effect is overdesign, including such things as guard-banding and gate oversizing; however, this approach can be very pessimistic and area and power inefficient. To avoid this problem, many NBTI-aware methodologies have been proposed. An NBTI-aware technology mapping technique was proposed in to guarantee the performance of the circuit during its lifetime. In, 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 Marcules 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. In and, 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. This paper reviews an aging-aware reliable multiplier 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. 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.

II. LITURATURE REVIEW

This sections introduces the background of the digital multipliers, column-bypassing multiplier, row-bypassing multiplier, variable-latency design, and NBTI/PBTI models.

2.1 Multiplier design techniques

A. M XM binary multiplier

There are two basic multiplication methods namely Booth multiplication algorithm and Array multiplication algorithm used for the design of multipliers". The schemes for efficient addition of partial products are Wallace tree [1]; Dadda tree [2]. The speed of multiplication (as well as power dissipation) is dominantly controlled by the propagation delay of the full / half adders used for the addition of partial products. The figure below shows the general block diagram of Vedic multiplication method described by Urdhva-tiryakbyham sutra of Vedic arithmetic

For (mxm) multipliers 4 numbers of (m/2) x (m/2) multipliers are required as shown in fig. 1 above. The first reduction layers requires 'm' full adders and second reduction stage requires a K Bit binary adders. (K is equal to [(3/2)m-1]) where m is number of bits in operand. Vedic multiplication method offers the advantage of design reuse in the sense that (n/2) x (n/2) multipliers and k- bit binary adders can be reused for design of (n x n) multiplier. The use of Full adders having equal input to output delay results in glitch free output.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

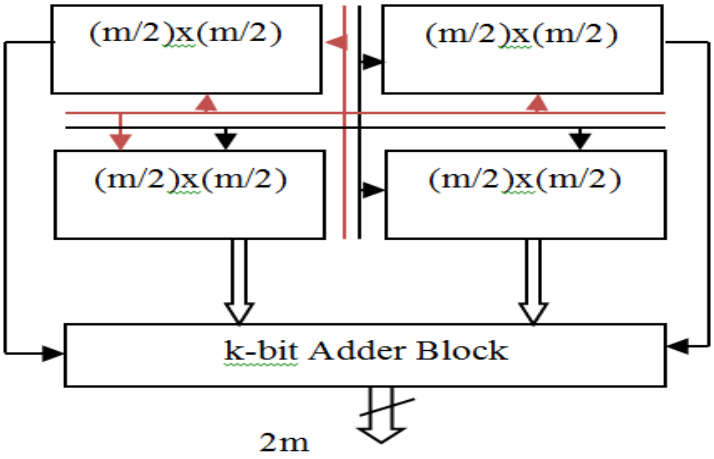

Fig. 1: mxm multiplication using UT sutra.

B. **4 × 4 normal AM.**

A low-power row-bypassing multiplier [23] is also proposedto reduce the activity power of the AM as shown in Fig.3. 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 multiplication.

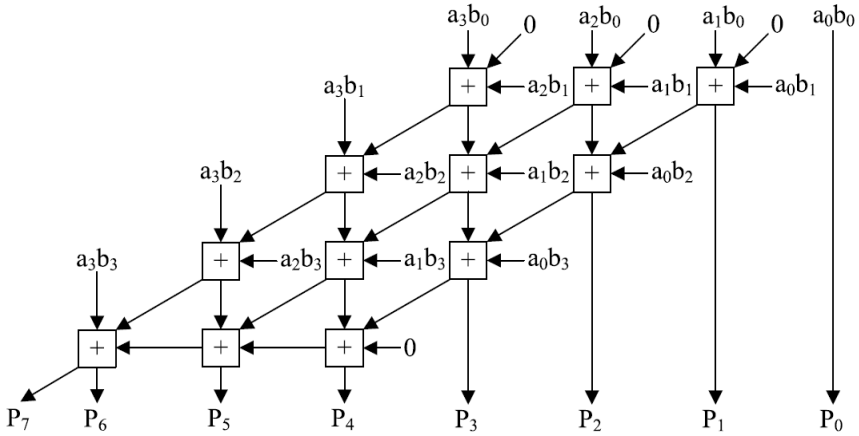

Fig. 2: 4 × 4 normal AM.

C. **Row-Bypassing Multiplier**

The Fig. 3 is a 4 ×4 row-bypassing multiplier. Each input is connected to an FA through a tristate gate. When the inputs are 11112 * 10012, the two inputs in the first and second rows are 0 for FAs. Because b1 is 0, the multiplexers in the first row select aib0 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 b2is 0, no switching activities will occur in the second-row FAs.However, the Fas must be active in the third row because the b3 is not zero selectors of the multiplexer to decide the output of the FA, and ai can also be used as the selector of the tristate gate to turn off the input path of the FA. If *ai* 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 *ai* is 1, the normal sum result is selected. More details for the column-bypassing multiplier can be found.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

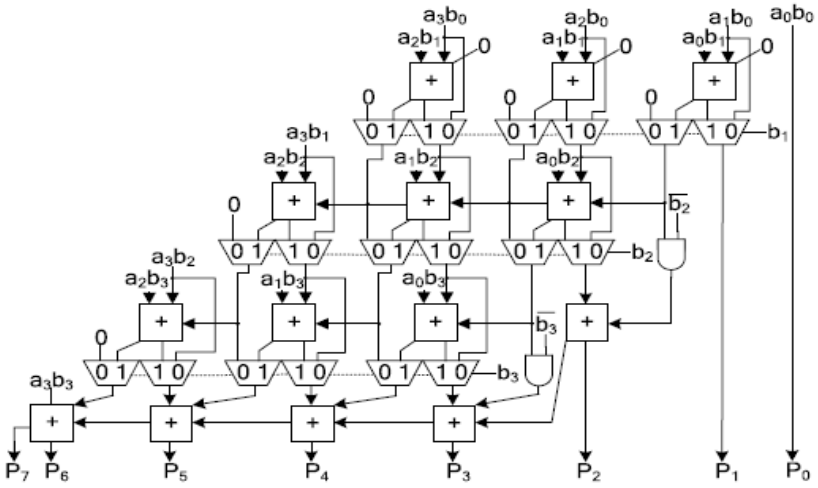

Fig. 3 : Row Bypassing multiplier

D. Column-Bypassing Multiplier

A column-bypassing multiplier is an improvement on the normal array multiplier (AM) as shown in Fig.5. The AM is a fast parallel AM and is shown in Fig. 3. 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 [2], a low-power columnbypassing multiplier design is proposed in which the FA operations are disabled if the corresponding bit in the multiplicand is 0. Fig. 4 shows a4×4 column-bypassing multiplier. Supposing the inputs are 10102 * 11112, 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 aibi.

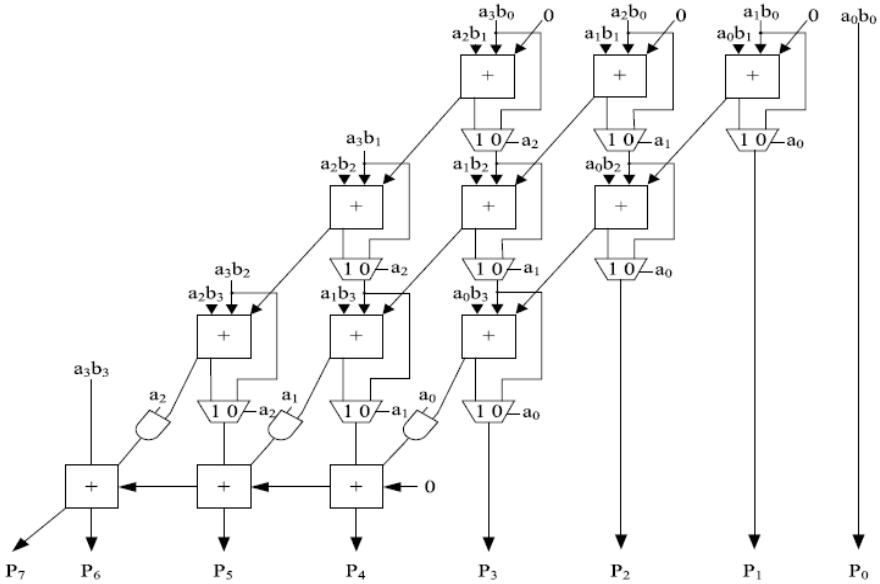

Fig. 4: Column-Bypassing Multiplier

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

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 ai can be used as the selector of the multiplexer to decide the output of the FA, and ai can also be used as the selector of the tristate gate to turn off the input path of the FA. If ai 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 ai is 1, the normal sum result is selected. More details for the columnbypassing multiplier can be found.

E. **Variable-Latency Technique**

Variable latency units (VLUs) allow for improved throughput by allowing one clock cycle for some computations, and two clock cycles for others, using hold logic to differentiate between the two cases.In ripple carry adders, the carry propagation time is the major speed limiting factor as seen in the previous lesson. 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 as shown in Fig.5. 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. One widely used approach employs the principle of carry look ahead solves this problem by calculating the carry signals in advance, based on the input signals. 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 Ai and Bi 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:

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$$

$$= G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 C_3$$

$$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

In general, the i^{th} carry output is expressed in the form Ci = F i (P's, G's , C0).

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

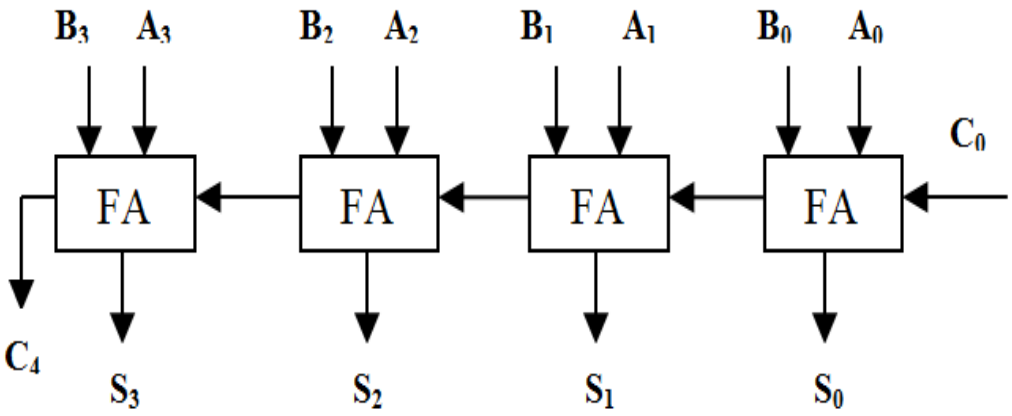

Fig. 5: variable-latency technique.

In other words, each carry signal is expressed as a direct SOP function of C0 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

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. Itgenerates the carry signals (C1 , C2 , C3 , and C4) as definedby the above expressions. Output signals of this level (C1 , C2 , C3 , and C4) will be valid after 3τ.
Third Level*:* Four XOR gates which generate the sum signals(S i) (S i = P i⊕Ci). 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.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

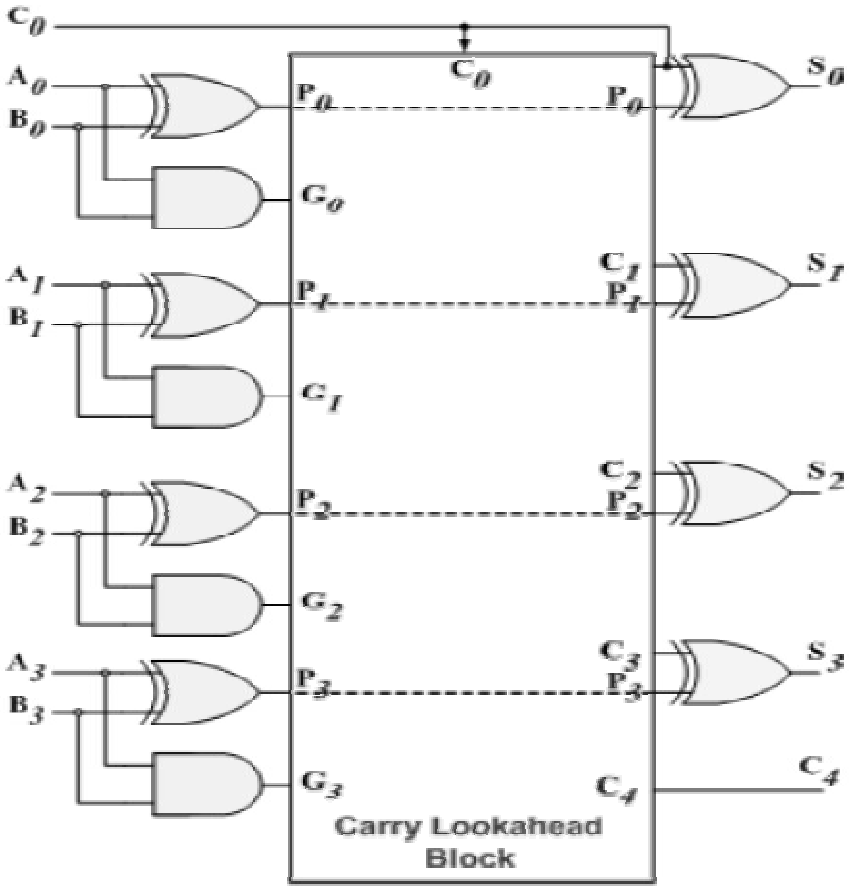

Fig. 6: Multiplication logic for carry look-ahead

III. PROPOSED AGING AWARE MULTPLIER

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.

A. Proposed Architecure

Proposed aging-aware multiplier architecture, 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, and an AHL circuit as shown in Fig.7.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

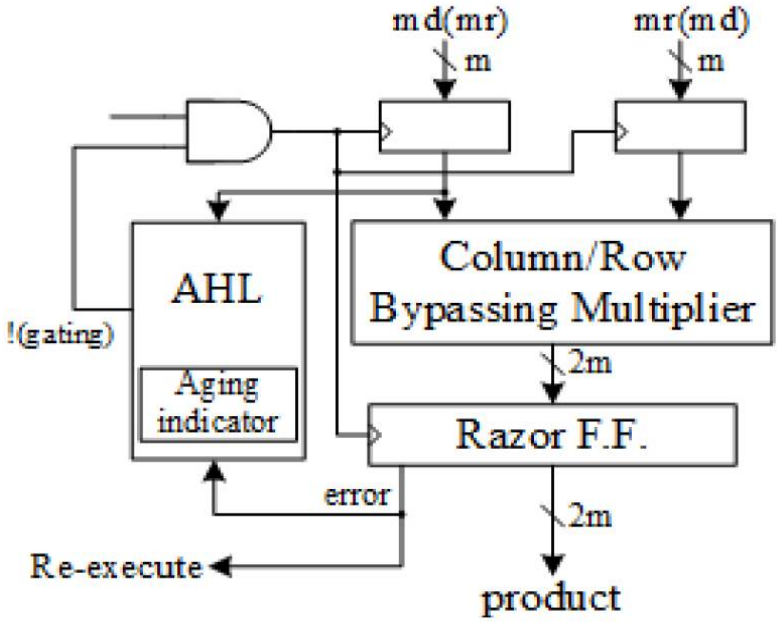

Fig. 7 : Proposed architecture (md means multiplicand; mr means multiplicator).

Proposed architecture (md means multiplicand; mr meansmultiplicator). Hence, the two aging-aware multipliers can beimplemented using similar architecture, and the differencebetween the two bypassing multipliers lies in the input signalsof 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.

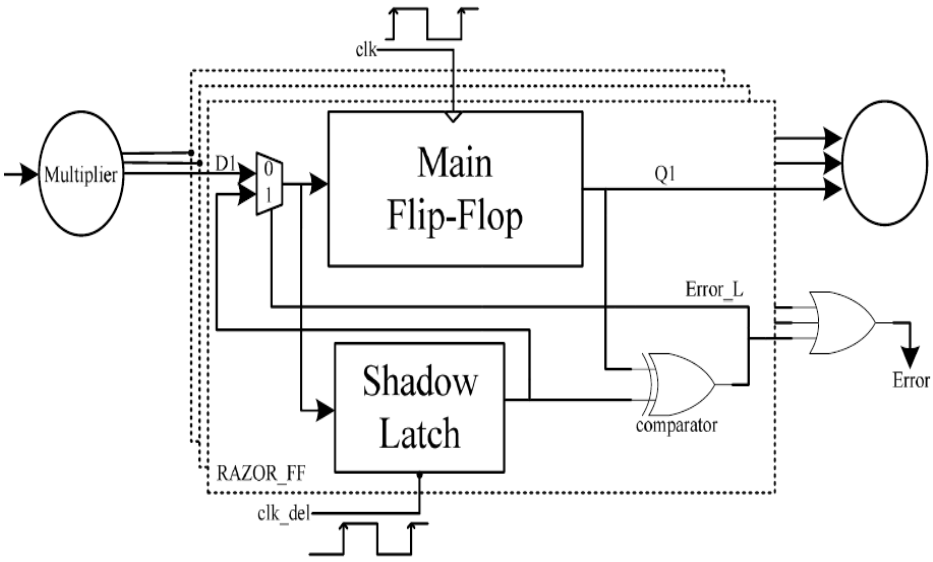

Fig. 8: Razor flip flops.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

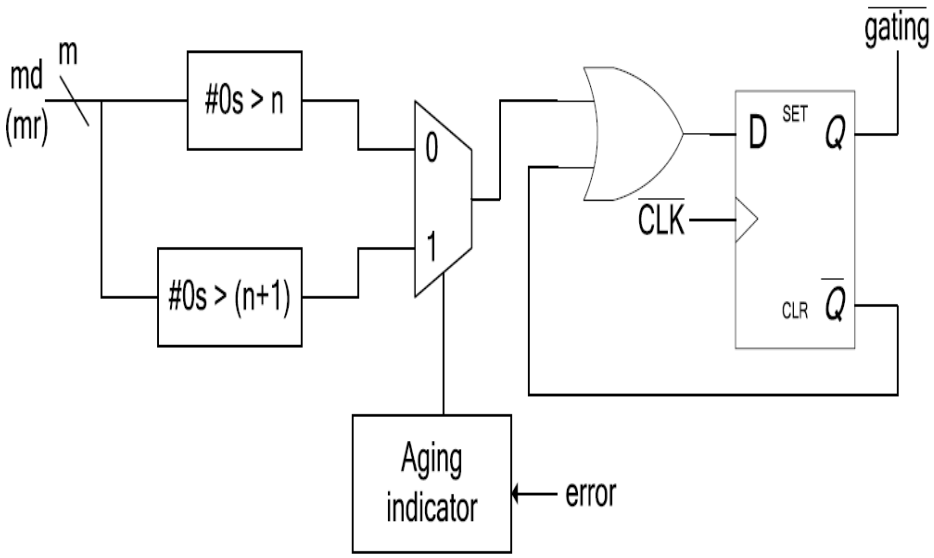

Fig.9: Diagram of AHL (md means multiplicand; mrmeans multiplicator).

Razor flip-flops can be used to detect whether timing violations occur before the next input pattern arrives as shown in Fig.9. 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 n details for the Razor flip-flop can be found. 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 as shown in Fig.10. 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 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 (multiplicator for the row-bypassing multiplier) is larger than n (n is a positive number, which will be discussed inSection 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 details of the operation of the AHL circuit are as follows: when an input pattern arrives, both

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

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 theoutput of the aging indicator. Then an OR operation is performed between the result of the multiplexer, and the $\bar{Q}$ signal is used to determine the input of the D flip-flop. When the pattern requires one cycle, the output of the multiplexer is 1. The !(gating) signal will become 1, and the input flip flops will latch new data in the next cycle. On the other hand, when the output of the multiplexer is 0, hich means the input pattern requires two cycles to complete, the OR gate will output 0 to the D flip-flop. Therefore, the (gating) signal will be 0 to disable the clock signal of the input flip-flops in the next cycle. Note that only a cycle of the input flip-flop will be disabled because the D flip-flop will latch 1 in the next cycle. 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 re-executed using two cycles to ensure the operation is correct. In this situation, the extra re-execution 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 re-execution 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 re-execute the operations using two cycles. Finally, our architecture can adjust the percentage of one-cycle patterns to minimize performance degradation due to the aging effect. When the circuit is aged, and many errors occur, the AHL circuit uses the second judging block to decide if an input is one cycle or two cycles.

IV. PROPOSED IMPLEMNETTAION

The performance analysis of proposed multiplier can be analyze by using Xilinx ISE with Verilog HDL. The target device Spartan 3 FPGA will show the device utilization in terms of LUT and Slices. Depend on the timing analysis the throughput of this multiplier will be calculated. We have detailed the system analysis of this multiplier to compare its performance and behavioral simulation with other existing methods such as Variable-latency pipelined multiplier architecture with a Booth algorithm.

V. CONCLUSION

This paper is a review on 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 proposed variable latency multipliers can be used under the influence of both the BTI effect and electro migration. In addition, it has less performance degradation because variable latency multipliers have less timing waste, but traditional multipliers need to consider the degradation caused by both the BTI effect and electro migration and use the worst case delay as the cycle period. We will analyze the behavioral simulation of this proposed multiplier by using Xilinx ISE simulator using Verilog HDL language.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 3, March 2017

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 SiO2/HfO2 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 minimization 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 NBTItolerant power-gating architecture," IEEE Trans. Circuits Syst., Exp. Briefs, vol. 59, no. 4, pp. 249–253, Apr. 2012.