AGING-AWARE RELIABLE MULITIPLIER DESIGN WITH ADAPTIVE HOLD LOGIC CIRCUIT

T.VENU GOPAL *

ABSTRACT:

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 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 (Vgs = −Vdd). 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 (Vth), 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 Vth 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 process. 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.

KEYWORDS: PBTI,NBTI,PMOS, NMOS, VLSI, SILICON AND THE GATE OXIDE INTERFACE

* Associate Professor, ECE, Vidya Jyothi Institute of Technology, Aziz Nagar,c.b.post

1. INTRODUCTION:

In many of the applications such as Fourier transform, discrete cosine transforms, and digital filtering, Digital multipliers are the most prominent arithmetic functional units are used. If the multipliers are too slow, the performance of total circuit will be reduced thus the throughput of such applications mostly depends on multipliers. When pMos transistor is under negative bias (V_{gs}=-V_{dd}) NBTI occurs. In such case, interaction between holes and hydrogen-passivated Si atoms breaks the Si–H bond generated during the oxidation process, giving H or H2 molecules. Interface traps are left when these molecules diffuse away. This interface traps between Si and gate oxide increases threshold voltage (V_{th}), which reduces the circuit switching speed. The reverse reaction occurs, when the bias voltage is removed which reduces the NBTI effect.

All the interface traps generated during stress phase does not eliminate by the reverse reaction and V_{th} is increased in the long term. So it is necessary to design a high-performance reliable multiplier. Similarly when an nMOS transistor is under positive bias, the PBTI effect occurs. On oxide or polygate transistors the PBTI effect is much smaller than the NBTI effect, and hence it is ignored. But, for high-*k*/metal-gate nMOS transistors the PBTI effect cannot be ignored. In fact, it has been shown that for 32-nm high-k/metal-gate process the PBTI effect is more significant than the NBTI effect. In order to perform correctly the Traditional circuit's uses critical path delay as the overall circuit clock cycle. The probability of the critical paths are activated is low. In many cases, the path delay is shorter than the critical path. Using the critical path delay for the non critical paths as overall cycle period will result in significant timing waste. Therefore, to reduce the timing waste of traditional circuits the variable latency design was proposed.

1.1 Aging Model:

As

The NBTI (PBTI) effect occurs when pMOS (nMOS) transistor is under negative (positive) bias voltage respectively, resulting in V_{th} drift. If bias voltage is removed the recovery process occurs which reduces the V_{th} drift. It is referred to as static NBTI (PBTI), if a pMOS (nMOS) transistor is under constant stress. Similarly, if both stress and recovery phases exist, then 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. The dc RD model should be modified to an ac RD model, if transistors are under alternative stress and recovery phases.

$$\Delta V_{th}(t) \cong K_{AC} \times t^n \cong \alpha (S, f) \times K_{DC} \times t^n$$

In the above equation α is a function of signal probability (S), and stress frequency (f). K_{DC} is a technology-dependent constant (which also depends on temperature). Here we use 32-nm high-k metal gate models in this paper. We set the temperature at 125 °C in our simulation and use the above BTI model to predict the BTI effect on the circuits. The BTI effect increased the critical path circuit delay by 13% in the above BTI model. Therefore, the increased delay may cause system failure in the long term, if the BTI effect is not considered during circuit design.

1.2 Existing System:

The Traditional circuits, takes critical path delay to perform overall circuit clock cycle correctly. The path delay is less than the critical path in many cases. For overall cycle period these noncritical paths use the critical path delay which results in significant timing waste. The existing method is fixed-latency pipelined multiplier architecture with an Array multiplier. The architecture was designed using Verilog-HDL and Synthesized in Xilinx RTL Compiler. Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Although the method is simple, as the addition is done serially as well as in parallel. However, the array multiplier which gives output easily, but has some drawbacks when compared to bypassing multipliers.

Drawbacks: In the Existing system major drawback is delay, as it if fixed latency technique where a longer clock cycle should be used to complete execution. And also if NBTI (PBTI) effect occurs the threshold voltage is increased which reduces the circuit switching speed. Here in existing method we use an array multiplier, which requires this larger delay, larger number of gates because of which area is also increased. For this reason array multiplier is less economical. It is a fastest multiplier with hardware complexity more.

1.3 Objective:

- It is important to design the digital circuits with high speed and also another important factor to consider is aging effect. Thus due to NBTI(PBTI) effects degrades the performance in terms of delay. This results in significant timing violations.
- Since many of the Digital multipliers are used in Fourier Transform, Discrete cosine transforms, Digital filtering. So it is necessary to speed up the multipliers performance.

- For this reason the Variable-latency Adaptive Hold Logic concept is introduced to speed up the multiplier although if aging effect occurs.
- The speed of Digital circuits also depends on Multipliers we, here we use an efficient multipliers which reduces the delay and power consumption and also area which yields a better results.
- In the introduction the importance of the multipliers in many DSP applications is discussed. The operation and importance of AHL, Variable latency design, Aging effect are discussed. Also, the objective and purpose of the proposed system is stated by comparing it with the existing system.

2.PROBLEM STATEMENT:

2.1 Reliable Multiplier Design With Adaptive Hold Logic:

Here, we are proposing a reliable multiplier design with adaptive hold logic (AHL) circuit with aging tolerance. To achieve reliable operation under the influence of NBTI and PBTI effects, we are designing a multiplier which is based on the variable-latency technique and can adjust the AHL circuit. The contributions of this paper are clearly specified below:

1) The Variable-latency multiplier is designed with an AHL circuit which is shown in Figure2.1.Whether the input patterns require one or two cycles can be decided by this AHL circuit which adjusts the judging criteria that to ensure, there is minimum performance degradation after aging occurs;

2) Under different cycle periods the comprehensive analysis and comparison of the multiplier's performance to show advantage of our proposed architecture over previous;

3) The reliable multiplier design with AHL circuit method is suitable for large multipliers. Though the experiment is performed in 16- and 32- bit multipliers, our proposed architecture can be extended easily to large designs;

4) From this Reliable Multiplier design we 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. Furthermore, our proposed architecture can achieve up to 80.17% and 69.40% performance improvement with 16 × 16 and 32 × 32 row-bypassing multipliers compared with 16× 16 and 32× 32 fixed-latency row-bypassing multipliers.

2.2 Basic Architecture Of Reliable Multiplier Design With AHL:

In Figure 2.1 our proposed aging-aware reliable multiplier architecture is shown, it has two m-bit inputs (m is a positive number), one 2m-bit output, one column- or row-bypassing multiplier, 2m 1-bit Razor flip-flop, and an AHL circuit.

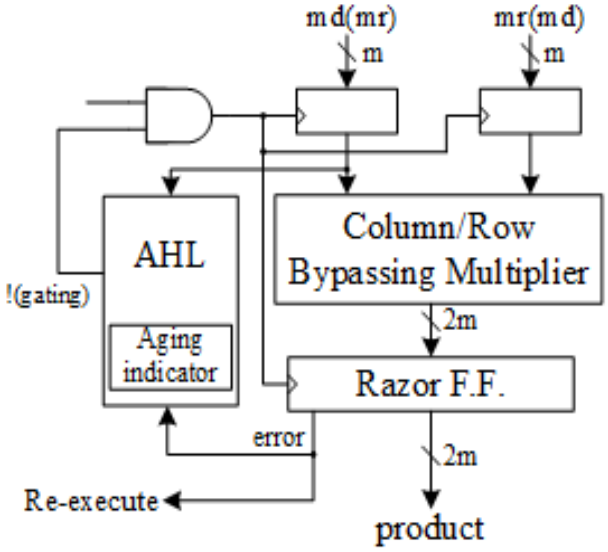

Figure 2.1 Proposed Architecture

The column- and row-bypassing multipliers are used to examine by the number of zeros in either the multiplicand or multiplicator to find whether the operation requires one cycle or two cycles to complete. Hence, using the number of 0's or 1's as the judging criteria results in similar outcomes. The two aging-aware multipliers can be designed using similar architecture, and hence input signals of AHL vary between these two bypassing multipliers. From 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, and that of the row-bypassing multiplier is multiplier bits. Razor flip-flop is used to detect whether timing violations occur before the next input pattern arrives.

2.2.1 Bypassing Multipliers:

Here, we presents low power multiplier design that inserts more number of zeros in the multiplicand or multiplier thereby reducing the number of switching activities as well as power consumption. This method is called as By-passing Technique. Bypassing multiplier shows

effective results in terms of speed and power. The unnecessary operations taken place in the corresponding adding cells are detected and reduced in this technique. There are two such multiplier designs namely:

- Row Bypass Multiplier.
- Column Bypass Multiplier.

2.2.2 Row Bypass Multiplier:

The Row bypassing scheme disables the operation in some rows in order to save switching power consumption. To understand the row bypassing technique, let take an example of an unsigned 4×4 multiplication. In fig 2, if bit is 0, all products in row j (for 0 i n-1) are 0. As a result, the addition in corresponding row can be bypassed. For example, let b1 of fig 3 be 0. For this case, the output from the first CSA row can be fed directly to the third CSA row and the second CSA row is disabled, thus by disabling the second CSA the switching activities are reduced row which results in low power dissipation. Since the rightmost FA in the second row is disabled, so it does not execute the addition and because of that output is not correct. In order to remove this problem, extra circuit must be added in the row bypassing technique. A low-power row-bypassing multiplier 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 tri-state gates use the multiplicator. Fig. 2.2 is a 4×4 row-bypassing multiplier. Each input is connected to an FA through a tri-state gate. When the inputs are 11112*10012, 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 *aib*0 as the sum bit and select 0 as the carry bit.

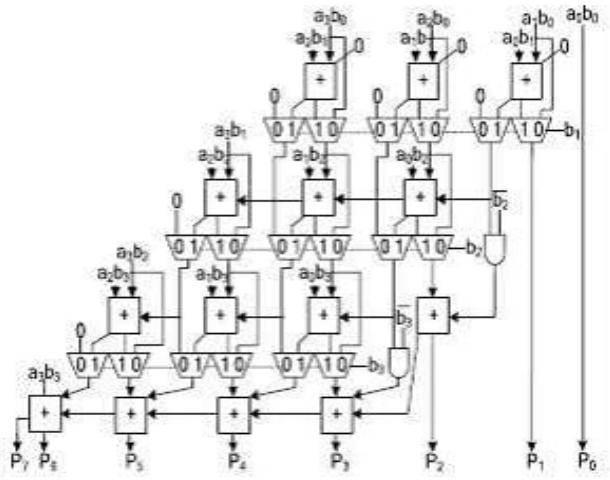

Figure 2.2 4x4 Row bypass multiplier

2.2.3 Column Bypass Multiplier:

In this technique, if the corresponding bit in the multiplicand is 0, the operations in a column can be disabled. There are two advantages of this technique. Firstly, it eliminates the extra correcting circuit. Second, the modified FA is simpler than that used in the row-bypassing multiplier. To understand the column bypassing technique, let take an example of 4×4 multiplication as shown in Figure 3.3, which executes 1010×1111. A column-bypassing multiplier is a modified array multiplier. The column bypassing multiplier is an excellent candidate for the variable design since we can simply examine the number of zeros in the multiplicand to predict whether the operation requires one cycle or two cycles to complete. A column-bypassing multiplier is designed in such a way that, the FA operations are disabled if the corresponding bit in the 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 tri-state 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 tri-state gate to turnoff 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.

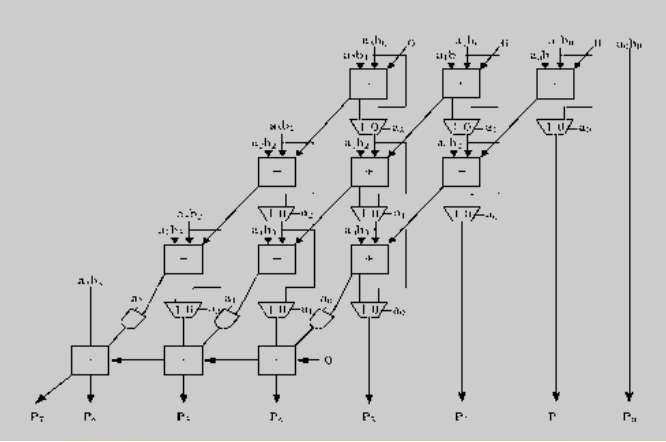

Figure 2.3 4x4 Column bypass multiplier

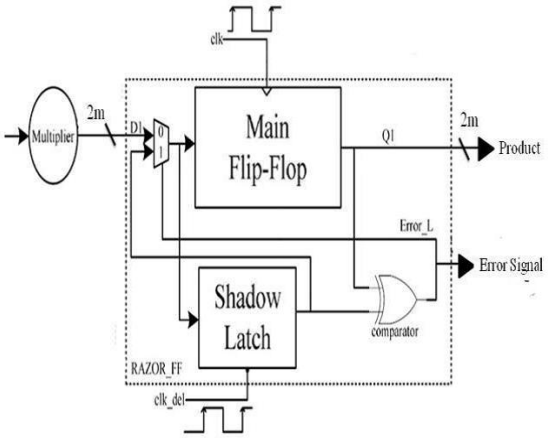

Figure 2.4 Razor Flip flop

2.2.4 Razor Flip-flop:

Fig.2.4 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 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.

2.2.5 Adaptive Hold Logic:

The AHL circuit is the key component in the aging-ware variable-latency multiplier. Fig. 3.5 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 1; otherwise, it will output 0 to indicate the aging effect is still not significant, and no actions are needed.

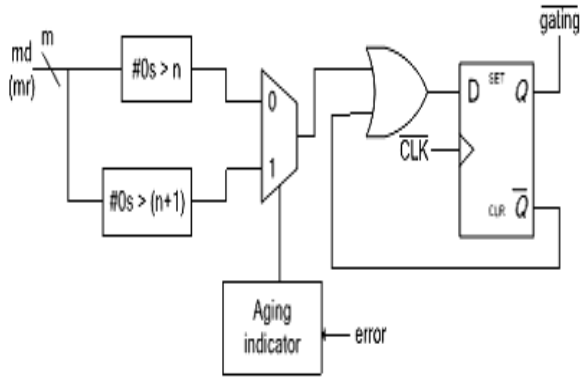

Figure 2.5 Adaptive Hold Logic Diagram

2.3 Proposed System:

In our proposed system the design for Aging aware reliable multiplier design is implemented with adaptive hold logic. The multiplier is designed using Bypassing Technique. Here we have implemented two bypass multipliers. They are

- Row Bypass Multiplier
- Column Bypass Multiplier

These Bypass Multipliers are used for high reliable digital system applications. Both of them have advantages and disadvantages compared to one another. When we referred to variable latency design then we can say these two multipliers can use for Reliable multiplier design for Adaptive hold logic with aging indicator. The proposed design is targeted towards Xilinx Spartan3E XC3S1600E. Also, the AHL circuit along with fixed-latency array.

3.TEST PLAN AND RESULTS:

3.1 Test Cases:

3.1.1 Control test case:

Control test case is mainly targeted for the usb section of the bridge. Its purpose is to check whether the device is responding to the standard request or not and to check the control endpoints of the device. It generates random standard requests to the usb devices and look for the

response for the device. It also checks the operation of the endpoint 0 controller in the device. It uses the sequences like set_address_command, set_confioguration_command, get_configuration_command, get_status_command, set_feature command and clear_feature_comand.

3.1.2 Register Write Test Case:

This test case is used to verify the registers in the i2c section of the bridge. There are several registers on the i2c side like control register, i2c target address register, counter register, clock prescale register etc all these registers are programmable by the user. The goal is to write different values in these registres and retrive them. This test case uses standard sequences like set_address_command, set_configuration_command and register write sequence from the sequence library.

3.1.3 Register Read Test case:

Register read test case works in tandom with the register write test case. It asks the usb device to give information about all the registers in the i2c side. The dut is written such that it will give all the register values as a group but the individual registers. The test case first configures the device properly, applies register writeing test case and then applies this test case to see contents of the registers. It uses standard sequences and register read sequences to read from the device.

3.1.4 Data Transmit Test Case:

This test case is to verify the master transmitter of the i2c section, but any way this test case should make sure some data in master transmit FIFO. So the test case will first calls standard commands to set the device up , then it has to write some data on the transmit FIFO and finally it will start the master transmitter state machine by configuring the registers on the i2c side. That means it uses standar sequences, data write sequences, register write sequences from the sequence library.

3.1.5 Data Receive Test Case:

This test case is used to verify the functionality of the master receiver of the i2c device. This test case first configures the device for use by using standard requests, issues the command to receive the data from the external i2c device and finally retrives the data from the master receiver FIFO by using data read commands. It uses standard sequences, write register sequence, data read sequence from the sequence library.

3.1.6 Slave Data Transmit Test Case:

This test case is to verify the slave transmitter of the i2c section, but any way this test case should make sure some data in slave transmit FIFO. So the test case will first calls standard commands to set the device up , then it has to write some data on the slave transmit FIFO and finally slave is ready to transmit the data. The transmission happens when external master initiates the transfer. It uses standard sequences, data write sequences, register write sequences from the sequence library.

3.1.7 Slave Data Receive Test Case:

This test case is used to verify the functionality of the slave receiver of the i2c device. This test case first configures the device for use by using standard requests, slave receives the data from the external i2c device and stores it in the slave receiver FIFO.Then agent reads the FIFO by using data read s commands. It uses standard sequences, write register sequence, data read sequence from the sequence library.

4. SIMULATION RESULTS:

The below figures shows the simulation results of test cases applied to the DUT. The response of the device for the control test case at the usb interface. The master transmitter sending random data to the external slave device.The experimental results show that our proposed architecture with the 32×32 column and row bypassing multipliers performance improvement.

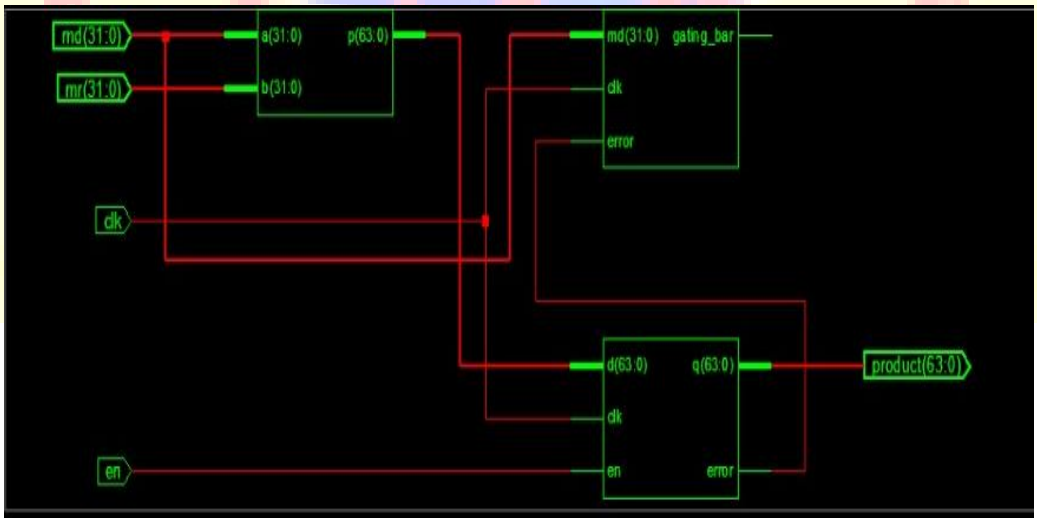

Figure 1: 32X32 Column Bypass Multiplier

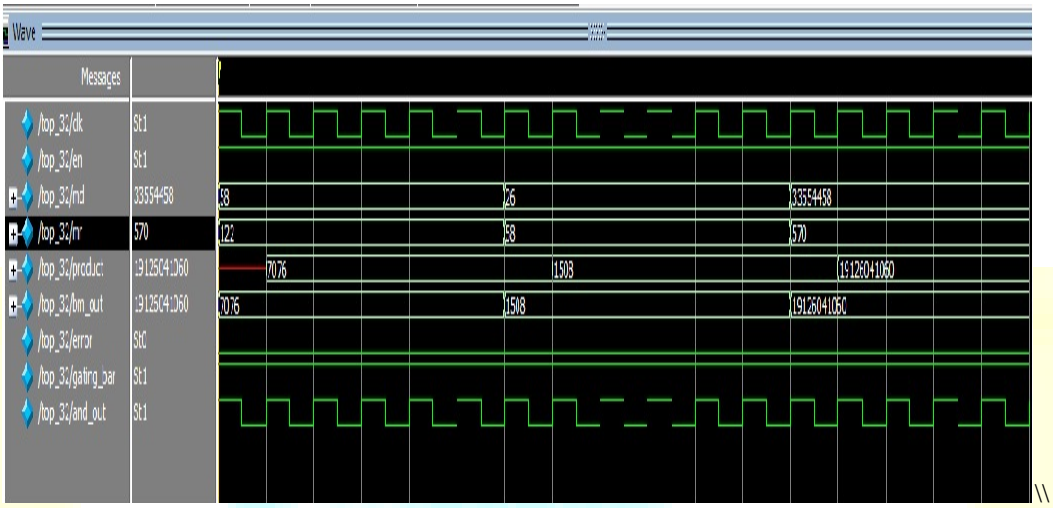

Figure 2: Simulation result of 32x32 Column bypass multiplier

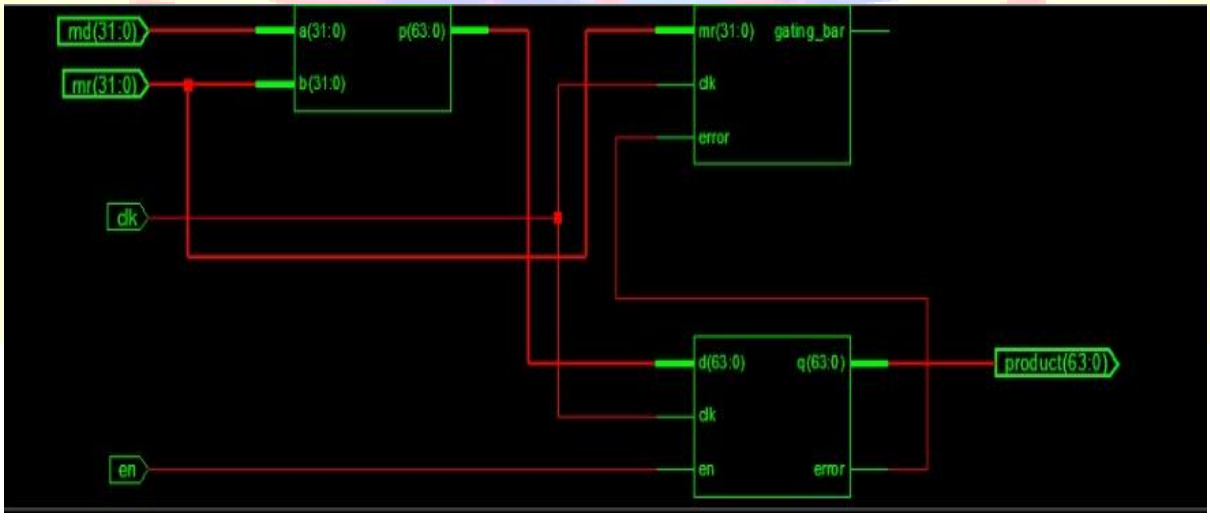

Figure 3: 32X32 Row Bypass Multiplier

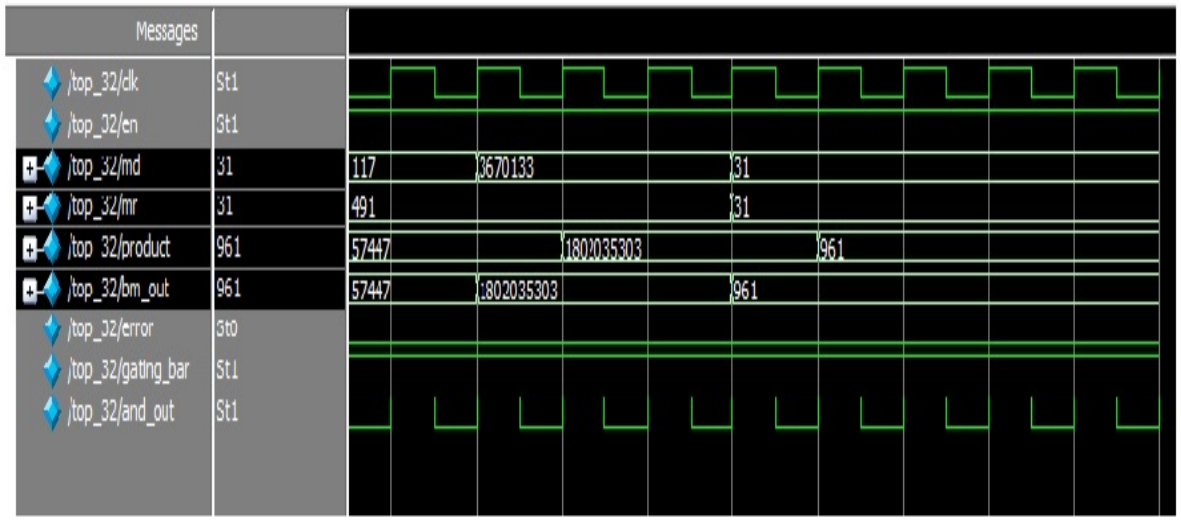

Figure 4 : Simulation result of 32x32 Row bypass multiplier

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 than our proposed architecture with16*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 .Furthermore ,our proposed architecture with the 16*16 and 32*32 row by passing multipliers can achieve up to80.17% and 69.40% performance improvement compared with the 16*16 and 32*32 FLRB multipliers .In addition ,the variable-latency bypassing multipliers exhibited the lowest average EDP and achieve up to 10.45% EDP reduction in 32*32 VLCB multipliers .Note that in addition to the BTI effect increase4s transistor delay ,interconnect also has its aging issue, which is called electro-migration. Electro-migration occurs when the current density is high enough to cause the drift of metal ions along the direction of electron flow .

The metal atoms will be gradually displace after a period of time ,and the geometry of the wires will change. If a wire becomes narrower, the resistance can delay of the wire will be increased. and in the end ,electro-migration may lead to open circuits. This issue is also more

serious in advanced process technology because metal wires are narrower, and changes in aging effects caused by the BTI effect and electro-migration are considered together, the delay and performance degradation will be more significant. Fortunately, our proposed variable latency multipliers can be used under the influence of both BTI effect and electro-migration .In addition, our proposed 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

6. FUTURE SCOPE:

Future Extension:

- We propose a technique called Recovery Boosting that allows both pMOS circuit to be put into the recovery mode by slightly modifying to the design.

Block Diagram:

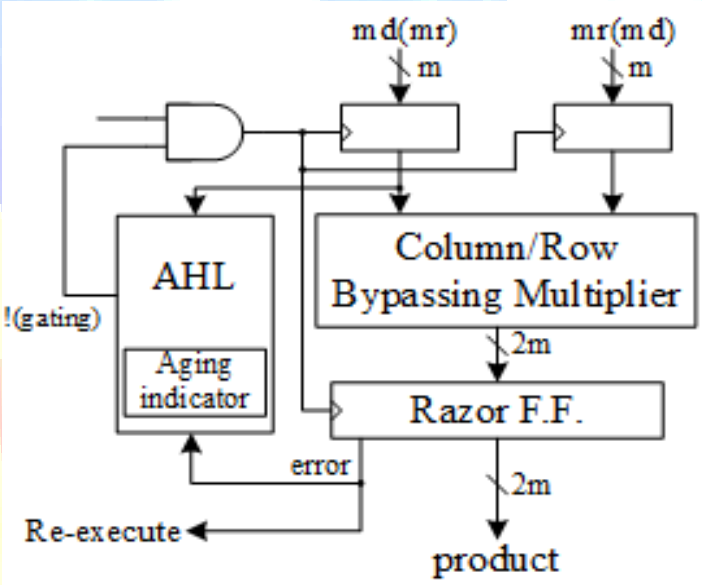

Figure 5: Future Extension Block Diagram

7.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 (chargetrapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates," in *Proc.IEEESymp. 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 -05

[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 nanometer 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.

[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. ASPDAC*, 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 optimization considering path sensitization," in *Proc. DATE*, 2011, pp. 1–6.

[13] K. Du, P. Varman, and K. Mohanram, "High performance reliable 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, "Performance" optimization using variable-latency design style," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1874–1883, Oct. 2011.

[17] N. V. Mujadiya, "Instruction scheduling on variable latency functional units of VLIW processors," in *Proc. ACM/IEEE ISED*, Dec. 2011, pp. 307–312.

[19] 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.

[20] Y. Chen, H. Li, J. Li, and C.-K. Koh, "Variable-latency adder (VL-Adder): New arithmetic circuit design to overcome NBTI," in *Proc. ACM/IEEE ISLPED*, Aug.07, pp. 195–200. [21] 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.

[22] M.-C. Wen, S.-J. Wang, and Y.-N. Lin, "Low power parallel multiplier with column bypassing," in *Proc. IEEE ISCAS*, May 2005, pp. 1638–1641.

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

[24] 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.

[25] B. C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Negative 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.