International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 04, April 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

SURVEY ON RELIABLE HIGH PERFORMANCE SUPER MULTIPLIER WITH ADAPTIVE HOLD LOGIC FOR AGING AWARENESS

Shital Tatyaso Atole[1], Dr. Doshi N. A.[2]

[1]*Electronics, SVPM COE Malegaon Pune University, (India)*

[2]*Professor at SVPM COE Malegaon Pune University, (India)*

ABSTRACT

In many digital systems we are using digital multipliers. The overall performance of digital systems depends on the throughput of the multiplier. But there are two effects that degrade the transistors throughput which are 1) Negative Bias Temperature Instability (NBTI) effect, 2) Positive Bias Temperature Instability (PBTI) effect. Negative Bias Temperature Instability (NBTI) effect means when a pMOS transistor is under negative bias (Vgs = −Vdd), increasing the threshold voltage (V_{th}) of the pMOS transistor and consequent decrease in drain current, and reducing multiplier speed. A same phenomenon, Positive Bias Temperature Instability (PBTI), occurs when an nMOS transistor is under positive bias. So it degrades the speed of transistors and after a long time, system may fail due to timing violations. Therefore it is important to design reliable high-performance multipliers. This paper explores the survey on aging-aware super column multiplier design with Adaptive Hold Logic (AHL) circuit. The multiplier is able to provide higher throughput through the variable latency and can adjust the AHL circuit to reduce 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

In most of the digital applications, such as the Fourier transform, discrete cosine transforms, and digital filtering. The throughput of these applications depends on the multipliers, if the multipliers are too slow, the performance of the whole circuit will be reduced so speed of the circuit will be reduced. Also, Negative Bias Temperature Instability (NBTI) occurs when a pMOS transistor is under negative bias (Vgs = −Vdd). NBTI manifests as an increase in the threshold voltage and consequent decrease in drain current. In this situation, first, interface traps are generated. Those traps cannot be recovered over a reasonable time of operation. Some authors refer to them as permanent traps. Those traps are the same as the one created by channel hot carrier. In the case of NBTI, it is believed that the electric field is able to break Si-H bonds located at the silicon-oxide interface during the oxidation process. H or is released in the substrate where it migrates 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

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

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 processes [1]–[3].

A traditional method to mitigate the aging effect is overdesign [4], [5], 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 [6] to guarantee the performance of the circuit during its lifetime.

In [7], 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 [8] 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 [11].

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

II. PRELIMINARIES

II.I. Column-Bypassing Multiplier

A column-bypassing multiplier is an improvement on the normal array multiplier (AM). The AM is a fast parallel AM. 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 [21], 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 shows a 4×4 column-bypassing multiplier. Supposing the inputs are 10102 * 11112, it can be seen that for the FAs in the first

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 04, April 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

and third diagonals, two of the three input bits are 0: the carry bit from its upper right FA and the partial product *aibi* . 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.

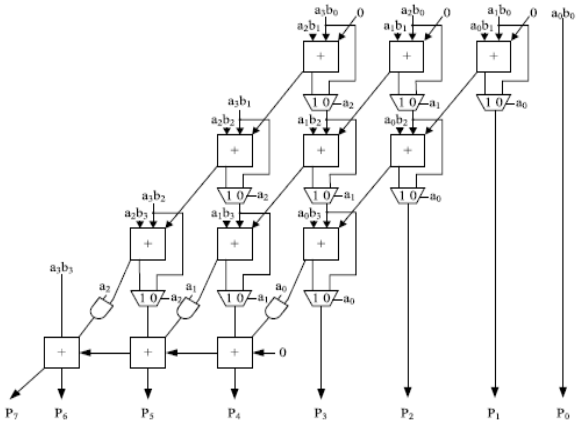

Fig.1: 4×4 Column-bypassing multiplier

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.

II.II. Variable-Latency Design

Variable-latency design was proposed to reduce 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. *A*8–*A*1, *B*8–*B*1 are 8-bit inputs, and *S*8–*S*1 are the outputs. Supposing the delay for each FA is one, and the maximum delay for the adder is 8. Through simulation, it can be determined that the possibility of the carry propagation delay being longer than 5 is low. Hence, the cycle period is set to 5, and hold logic is added to notify the system whether the adder can complete the operation within a cycle period. Fig. 4 also shows the hold logic that is used in this circuit. The function of the hold logic is (*A*4 XOR *B*4)(*A*5 XOR *B*5) as shown in fig. 2. If the output of the hold logic is 0, i.e., *A*4 = *B*4 or *A*5 = *B*5, either the fourth or the fifth adder will not produce a carryout. Hence, the maximum delay will be less than one cycle period. When the hold logic output is 1, this means that the input can activate paths longer than 5, so the hold logic notifies the system that the current operation requires two cycles to complete. Two cycles are sufficient for the longest path to complete (5 * 2 is larger than 8). All multipliers execute operations on a fixed cycle period. The maximum path delay is 1.32 ns for the AM, 1.88 ns for the column-bypassing multiplier, and 1.82 ns for the row-bypassing multiplier. It can be seen that for the AM, more than 98% of the paths have a delay of <0.7 ns. Moreover, more than 93% and 98% of the paths in the FLCB and row-bypassing multipliers present a delay of <0.9 ns, respectively. Hence, using the maximum path delay for all

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 04, April 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

paths will cause significant timing waste for shorter paths, and redesigning the multiplier with variable latency can improve their performance.

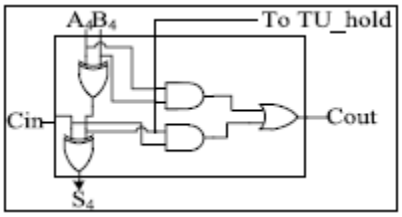

Fig. 2: Hold Logic Circuit

Another key observation is that the path delay for an operation is strongly tied to the number of zeros in the multiplicands in the column-bypassing multiplier. Three thousand randomly selected patterns are used in each experiment. It can be seen as the number of zeros in the multiplicands increases, delay distribution is left shifted, and average delay is reduced. The reason for this is the multiplicand is used as the select line for column-bypassing multipliers, and if more zeros exist in the multiplicand, more FAs will be skipped, and the sum bit from the upper FA is passed to the lower FA, reducing the path delay.

II.III. AGING MODEL

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 [22]

$$\Delta v_{th}(t) \cong K_{AC} \times t^{n} \cong \alpha(S, f) \times K_{DC} \times t^{n} \quad (1)$$

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

$$K_{DC} = A \times T_{OX} \times \sqrt{C_o \times (v_{GS} - v_{th})} \times [1 - v_{DS} / \alpha(v_{GS} - v_{th})] \times \exp(E_{OX} / E_O) \times \exp\left(-\frac{E_a}{KT}\right) \quad (2)$$

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 *Ea* are technology-independent characteristics of the reaction that are equal to 1.9–2.0 MV/cm and 0.12 eV, respectively.

III. PROPOSED AGING-AWARE MULTIPLIER

This is 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.

III.I. Proposed Architecture

Fig. 3 shows our 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 [23], and an AHL circuit. The inputs of the row-bypassing multiplier are the symbols in the parentheses.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 04, April 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

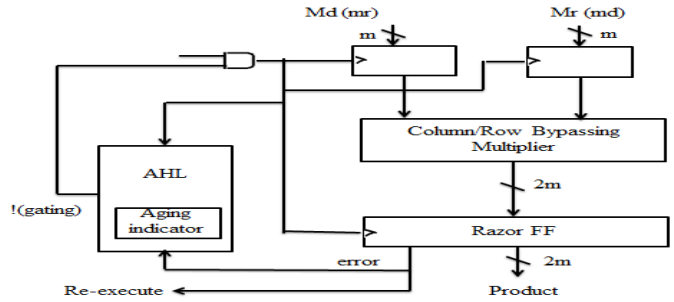

Fig.3: Proposed architecture (md means multiplicand; mr means multiplicator).

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. 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. 4 shows the details of Razor flip-flops.

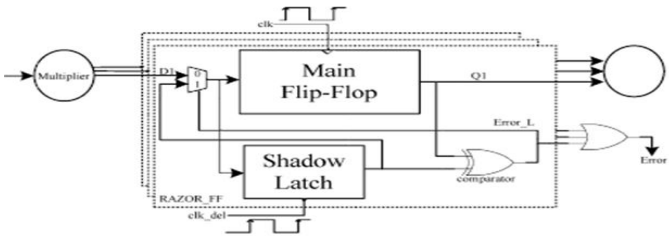

Fig.4: Razor Flip Flop [23]

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 reexecuted with two cycles. Although the reexecution may seem costly, the overall cost is low because the reexecution frequency is low. The AHL circuit is the key component in the aging-ware variable latency multiplier. Fig. 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.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 04, April 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

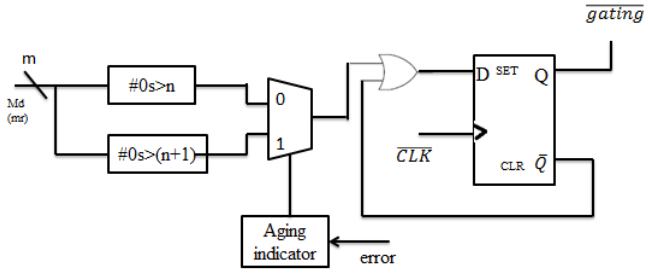

Fig.5: Adaptive Hold Logic (AHL)

If the cycle period is too short, the column-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 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 details of the operation of the AHL circuit are as follows: when an input pattern arrives, both 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 the output of the aging indicator. Then an OR operation is performed between the result of the multiplexer, and the .*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, which 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

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

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 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. EXPERIMENTAL RESULT

Our experiments are conducted in a Linux operating system. We adopt a 32-nm high-*k* predictive technology model [2] 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.

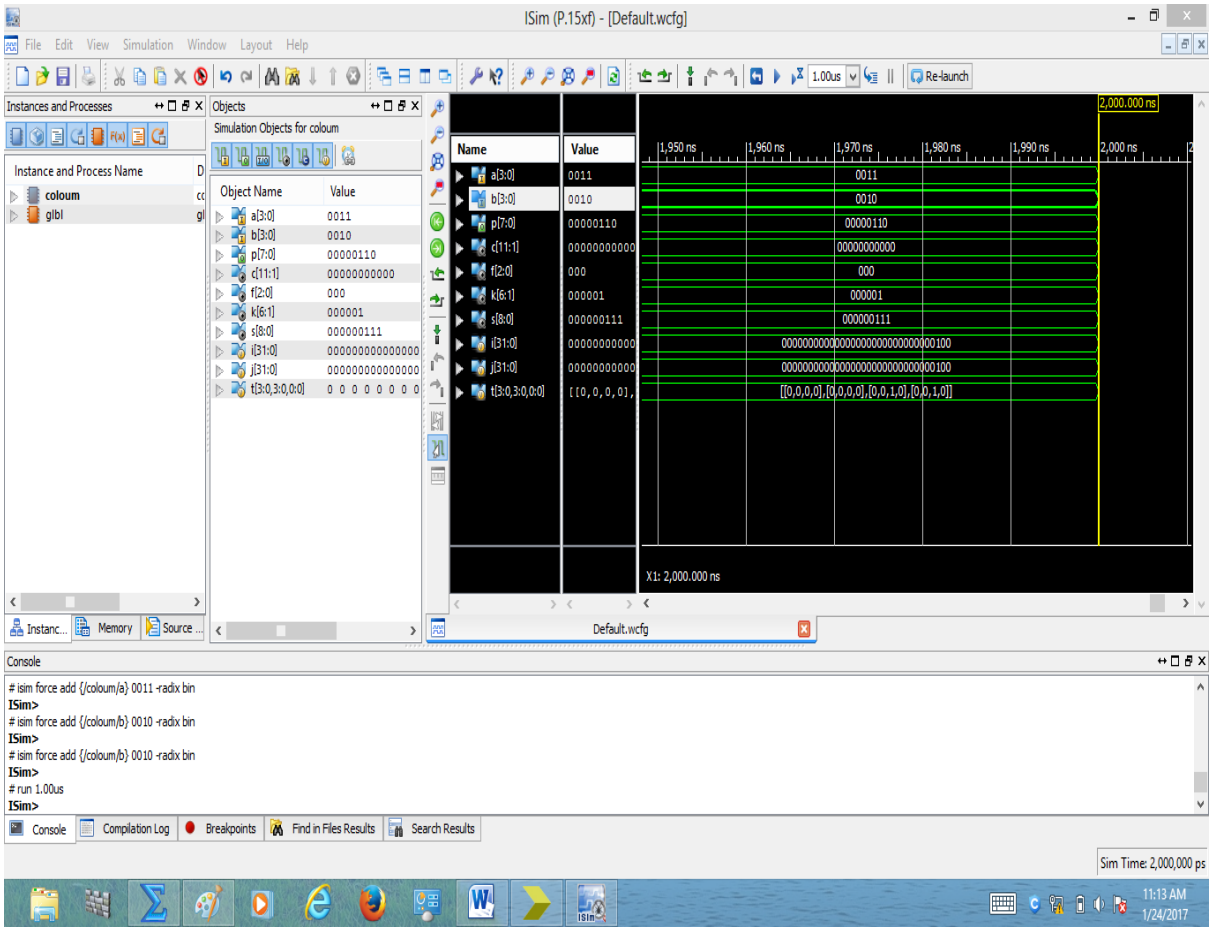

Fig.6: Simulation result of 4*4 Column Multiplier.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

V. CONCLUSION

The multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. The experimental results show that our proposed architecture with 16×16 column bypassing multipliers can attain up to 63.88% performance improvement. Note that in addition to the BTI effect that increases transistor delay, interconnect also has its aging issue, which is called electromigration. Electromigration occurs when the current density is high enough to cause the drift of metal ions along the direction of electron flow.

ACKNOWLEDGEMENTS

The authors wish to thanks all for knowledge sharing and advice. The authors wish to thank SVPM's College of Engineering and faculty of Electronics and Telecommunications Engineering for the facilities provided. The author is grateful to Dr. Neeta Doshi (Senior Faculty Member) for her constant and valuable suggestions while doing the project work.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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

[6] 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.

[7] 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.

[8] K.-C. Wu and D. Marculescu, "Joint logic restructuring and pin reordering against NBTI-induced performance degradation," in Proc. DATE, 2009, pp. 75 80.

[9] 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.

[10] 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.

[11] K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," in Proc. DATE, 2011, pp. 1–6.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

[12] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. DATE, 2012, pp. 1257–1262.

[13] 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.

[14] D. Baneres, J. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in Proc. DATE, 2009, pp. 1704–1709.

[15] 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.

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

[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 multiplier with column bypassing," in Proc. IEEE ISCAS, May 2005, pp. 1638–1641.

[22] 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.

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