ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design and Implementation of Aging-Aware Reliable Multiplier by Using Booth Algorithm

Thakur Vishesh Singh
M.Tech (VLSI-SD),
Dept of ECE,
TKREC, Meerpet, Hyderabad.

K. Kumara Swamy
Assistant Professor
Dept of ECE,
TKREC, Meerpet, Hyderabad.

Dr.P.Ram Mohan Rao
FIE,CE(I),MISTE,MISH,MISCEE,MASCE(I),MISNT,
Dept of ECE,
TKREC,Meerpet, Hyderabad.

ABSTRACT:

Digital multipliers are among 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 occur swhen a pMOS transistor is under negative bias (Vgs = −Vdd), increasing the threshold voltage of the pMOS transistor, and reducing multiplier speed. A similar phenomenon, positive bias temperature instability, occurs when an nMOS transistor isunder positive bias. Both effects degrade transistor speed, and in thelong term, the system may fail due to timing violations. Therefore, it is important to design reliable high-performancemultipliers. In this paper, we propose an aging-aware multiplier design with novel adaptive hold logic (AHL) circuit using booth multiplier algorithm. Themultiplier is able to provide higher throughput through the variable latency and can adjust the AHL circuit to mitigateperformance degradation that is due to the aging effect. Moreover, the proposed architecture can be applied to a column- orrow-bypassing multiplier.

I.INTRODUCTION:

Digital multipliers are among the most critical arithmetic functional units in many applications, such as the Fouriertransform, discrete cosine transforms, and digital filtering. The through put of these applications depends on multipliers,and if the multipliers are too slow, the performance of entire circuits will be reduced. Furthermore, negative biastemperature instability (NBTI) occurs when a pMOS transistor is under negative bias (Vgs = −Vdd). In this situation, theinteraction between inversion layer holes and hydrogen-passivated Si atoms breaks the Si–H bond generated during theoxidation 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 NBTIeffect. However, the reverse reaction does not eliminate all the interface traps generated during the stress phase, and Vth isincreased in the long term. Hence, it is important to design a reliable high-performance multiplier. The corresponding effecton an nMOS transistor is positive bias temperature instability (PBTI), which occurs when an nMOS transistor is underpositive bias. Compared with the NBTI effect, the PBTI effect is much smaller on oxide/polygate transistors, and thereforeis usually ignored.

However, for high-k/metal-gate nMOS transistors with significant charge trapping, the PBTI effect canno longer be ignored. In fact, it has been shown that the PBTI effect is more significant than the NBTI effect on 32-nmhigh-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-awaremethodologies have been proposed.An NBTI-aware technology mapping technique was proposed in to guarantee theperformance of the circuit during its lifetime. In an NBTI-aware sleep transistor was designed to reduce the aging effects onpMOS sleep-transistors, and the mlifetime stability of the power-gated circuits under consideration was improved. Wu and-Marculescu proposed a point logic restructuring and pin reordering method, which is based on detecting functionalsymmetries and transistor stacking effects. They also proposed an NBTI optimization method that considered pathsensitization. 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, theprobability that the critical paths are activated is low. In most cases, the path delay is shorter than the critical path. For thesenoncritical paths, using the critical path delay as the overall cycle period will result in significant timing waste. Hence, thevariable-latency design was proposed to reduce the timing waste of traditional circuits. The variable-latency design dividesthe circuit into two parts: 1) shorter paths and 2) longer paths. Shorter paths can execute correctly in one cycle, whereaslonger paths need two cycles to execute. When shorter paths are activated frequently, the average latency of variablelatencydesigns is better than that of traditional designs. For example, several variable-latency adders were proposed usingthe speculation technique with error detection and recovery. A short path activation function algorithm was proposed in toimprove the accuracy of the hold logic and to optimize the performance of the variable-latency circuit. An instructionscheduling algorithm was proposed in to schedule the operations on non uniform latency functional units and improve theperformance of Very Long Instruction Word processors. In a variable-latency pipelined multiplier architecture with a Boothalgorithm was proposed. In process-variation tolerant architecture for arithmetic units was proposed, where the effect ofprocess-variation is considered to increase the circuit yield. I

n addition, the critical paths are divided into two shorter pathsthat could be unequal and the clock cycle is set to the delay of the longer one. These research designs were able to reducethe timing waste of traditional circuits to improve performance, but they did not consider the aging effect and could notadjust themselves during the runtime. A variable-latency adder design that considers the aging effect was proposed.However, no variable-latency multiplier design that considers the aging effect and can adjust dynamically has been done.

II.PAPER CONTRIBUTION:

In this paper, we propose an aging-aware reliable multiplier design with novel adaptive hold logic (AHL) circuit. Themultiplier is based on the variable-latency technique and can adjust the AHL circuit to achieve reliable operation under theinfluence of NBTI and PBTI effects.

To be specific, the contributions of this paper are summarized as follows: 1) novelvariable-latency multiplier architecture with an AHL circuit. The AHL circuit can decide whether the input patterns requireone or two cycles and can adjust the judging criteria to ensure that there is minimum performance degradation afterconsiderable aging occurs; 2) comprehensive analysis and comparison of the multiplier's performance under different cycleperiods to show the effectiveness of our proposed architecture; 3) an aging-aware reliable multiplier design method that issuitable for large multipliers. Although the experiment is performed in 16- and 32-bit multipliers, our proposed architecturecan be easily extended to large designs; 4) the experimental results show that our proposed architecture with the 16 × 16and 32 × 32 column-bypassing multipliers can attain up to 62.88% and 76.28% performance improvement compared withthe 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 improvement as comparedwith 16× 16 and 32× 32 fixed-latency row-bypassing multipliers.

III.PRELIMINARIES:

Row-Bypassing Multiplier:

A low-power row-bypassing multiplier [23] is also proposed to reduce the activity power of the AM. The operation of thelow-power row-bypassing multiplier is similar to that of the low-power column-bypassing multiplier, but the selector of themultiplexers and the tristate gates use the multiplicator.

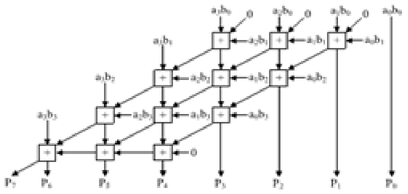

Fig: 4 × 4 normal AM

Fig. 3 is a 4 ×4 row-bypassing multiplier. Each input is connected to an FA through a tristate gate. When the inputs are11112 * 10012, the two inputs in the first and second rows are 0 for FAs. Because b1 is 0, the multiplexers in the first rowselect aibo as the sum bit and select 0 as the carry bit. The inputs are bypassed to FAs in the second rows, and the tristategates turn off the input paths to the FAs.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Therefore, no switching activities occur in the first-row FAs; in return, power consumption is reduced. Similarly, becauseb2is 0, no switching activities will occur in the second-row FAs. However, the FAs must be active in the third row becausethe b3 is not zero.

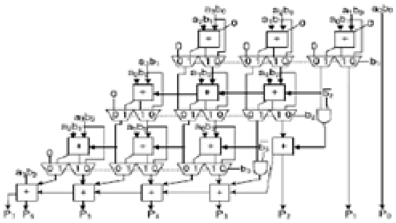

Fig: 4X4 Row by passing multiplier

selector of the multiplexer to decide the output of the FA, and aican also be used as the selector of the tristate gate to turnoff the input path of the FA. If is 0, the inputs of FA are disabled, and the sum bit of the current FA is equal to the sum bitfrom its upper FA, thus reducing the power consumption of the multiplier. If is 1, the normal sum result is selected. Moredetails for the column-bypassing multiplier can be found.

IV.PROPOSED AGING-AWARE MULTIPLIER:

The proposed aging-aware reliable multiplier design. It introduces the overall architecture and the functions of eachcomponent and also describes how to design AHL that adjusts the circuit when significant aging occurs.

Proposed Architecture:

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.

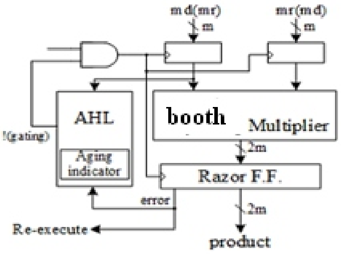

Fig:Proposed architecture (md means multiplicand; mr means multiplicator).

Booth Multiplier:

Booth’s multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two’s complement notation. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on crystallographyat Birkbeck College in Bloomsbury, London. Booth used desk calculators that were faster at shifting than adding and created the algorithm to increase their speed. Booth’s algorithm is of interest in the study of computer architecture.

The algorithm:

Booth’s algorithm examines adjacent pairs of bits of the N-bit multiplier Y in signed two’s complement representation, including an implicit bit below the least significant bit, y-1 = 0. For each bit yi, for i running from 0 to N-1, the bits yi and yi-1are considered. Where these two bits are equal, the product accumulator P is left unchanged. Where yi = 0 and yi-1 = 1, the multiplicand times 2i is added to P; and where yi = 1 and yi-1 = 0, the multiplicand times 2i is subtracted from P. The final value of P is the signed product.The multiplicand and product are not specified; typically, these are both also in two’s complement representation, like the multiplier, but any number system that supports addition and subtraction will work as well. As stated here, the order of the steps is not determined.

Typically, it proceeds incremental shifting of the P accumulator to the right between steps; low bits can be shifted out, and subsequent additions and subtractions can then be done just on the highest N bits of P.[1] There are many variations and optimizations on these details. The algorithm is often described as converting strings of 1’s in the multiplier to a high-order +1 and a low-order –1 at the ends of the string. When a string runs through the MSB, there is no high-order +1, and the net effect is interpretation as a negative of the appropriate value. Hence, the two aging-aware multipliers can be implemented using similar architecture, and the difference between the twobypassing multipliers lies in the input signals of the AHL. According to the bypassing selection in the column or rowbypassingmultiplier, the input signal of the AHL in the architecture with the column-bypassing multiplier is themultiplicand, whereas of the row-bypassing multiplier is the multiplicator.

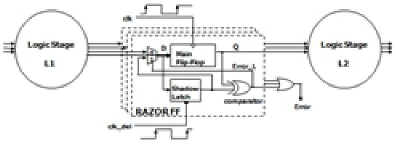

Fig:Razor flip flops.

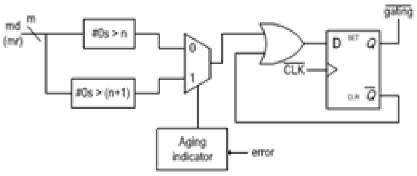

Fig:Diagram of AHL (md means multiplicand; mr means multiplicator).

Razor flip-flops can be used to detect whether timing violations occur before the next input pattern arrives.A 1-bit Razor flip-flop contains a main flip-flop, shadow latch, XOR gate, and mux. The main flip-flop catches theexecution result for the combination circuit using a normal clock signal, and the shadow latch catches the execution resultusing a delayed clock signal, which is slower than the normal clock signal. If the latched bit of the shadow latch is differentfrom that of the main flip-flop, this means the path delay of the current operation exceeds the cycle period, and the mainflip-flop catches an incorrect result. If errors occur, the Razor flip-flop will set the error signal to 1 to notify the system toreexecute the operation and notify the AHL circuit that an error has occurred. We use Razor flip-flops to detect whether anoperation that is considered to be a one-cycle pattern can really finish in a cycle. If not, the operation is reexecuted with twocycles. Although the reexecution may seem costly, the overall cost is low because the reexecution frequency is low.

Moredetails for the Razor flip-flop can be found. The AHL circuit is the key component in the aging-ware variable-latencymultiplier. Fig. shows the details of the AHL circuit. The AHL circuit contains an aging indicator, two judging blocks, onemux, and one D flip-flop. The aging indicator indicates whether the circuit has suffered significant performancedegradation due to the aging effect. The aging indicator is implemented in a simple counter that counts the number of errorsover a certain amount of operations and is reset to zero at the end of those operations. If the cycle period is too short, thecolumn- or row-bypassing multiplier is not able to complete these operations successfully, causing timing violations.

Thesetiming violations will be caught by the Razor flip-flops, which generate error signals. If errors happen frequently andexceed a predefined threshold, it means the circuit has suffered significant timing degradation due to the aging effect, andthe aging indicator will output signal 1; otherwise, it will output 0 to indicate the aging effect is still not significant, and noactions 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 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 ofthem will be chosen at a time. In the beginning, the aging effect is not significant, and the aging indicator produces 0, so thefirst judging block is used.

After a period of time when the aging effect becomes significant, the second judging block ischosen. Compared with the first judging block, the second judging block allows a smaller number of patterns to becomeone-cycle patterns because it requires more zeros in the multiplicand (multiplicator). The details of the operation of theAHL circuit are as follows: when an input pattern arrives, both judging blocks will decide whether the pattern requires onecycle or two cycles to complete and pass both results to the multiplexer. The multiplexer selects one of either result basedon 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 is1. The !(gating) signal will become 1, and the input flip flops will latch new data in the next cycle. On the other hand, whenthe output of the multiplexer is 0, hich means the input pattern requires two cycles to complete, the OR gate will output 0to 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. Theoverall 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 AHLcircuit decides if the input patterns require one or two cycles. If the input pattern requires two cycles to complete, the AHLwill output 0 to disable the clock signal of the flip-flops. Otherwise, the AHL will output 1 for normal operations. When thecolumn- or row-bypassing multiplier finishes the operation, the result will be passed to the Razor flip-flops. The Razor flipflopscheck whether there is the path delay timing violation. If timing violations occur, it means the cycle period is not longenough for the current operation to complete and that the execution result of the multiplier is incorrect. Thus, the Razorflip-flops will output an error to inform the system that the current operation needs to be reexecuted using two cycles toensure the operation is correct. In this situation, the extra reexecution cycles caused by timing violation incurs a penalty tooverall average latency.

However, our proposed AHL circuit can accurately predict whether the input patterns require oneor two cycles in most cases. Only a few input patterns may cause a timing variation when the AHL circuit judgesincorrectly. In this case, the extra reexecution cycles did not produce significant timing degradation. In summary, ourproposed multiplier design has three key features. First, it is a variable-latency design that minimizes the timing waste ofthe noncritical paths. Second, it can provide reliable operations even after the aging effect occurs. The Razor flip-flopsdetect the timing violations and reexecute the operations using two cycles. Finally, our architecture can adjust thepercentage of one-cycle patterns to minimize performance degradation due to the aging effect. When the circuit is aged, andmany errors occur, the AHL circuit uses the second judging block to decide if an input is one cycle or two cycles.

V.SIMULATION RESULTS:

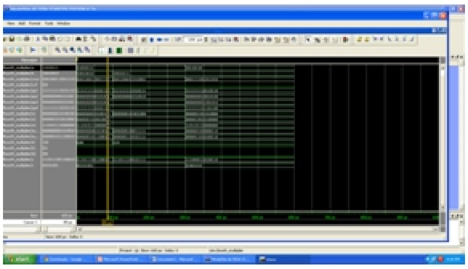

Fig: Simulation result of booth multiplier

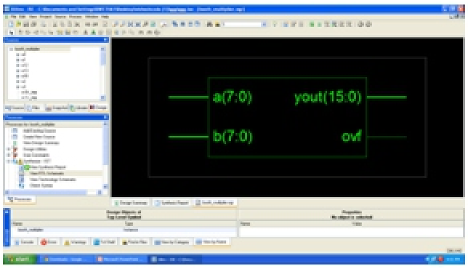

Fig:RTL schematic

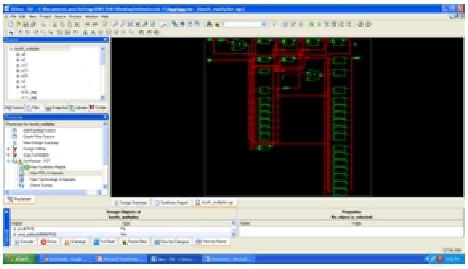

Fig: Internal schematic diagram

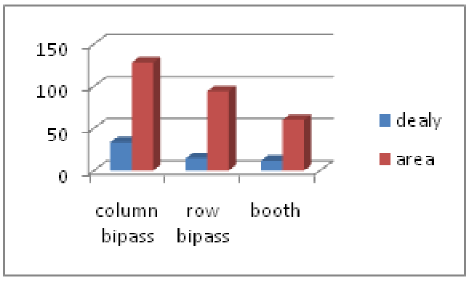

Fig: Performance graph

VI.CONCLUSION:

This paper proposed an aging-aware variable-latency multiplier design with the AHL. The multiplier is able to adjust theAHL to mitigate performance degradation due to increased delay. The experimental results show that our proposedarchitecture with 4x4 multiplication with Booth as last stage instead of Normal RCA adder it will decrease the delay andimprove the performance compared with previous designs.

REFERENCES:

[1] Y. Cao. (2013). Predictive Technology Model (PTM) and NBTI Model [Online]. Available: http://www.eas.asu.edu/ptm

[2] S. Zafaret 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 designin 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/IEEEDAC, 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.

[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.