IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 4, Issue 8, August 2016

Design of Efficient Aging-Aware Predictable Multiplier Using Adaptive Hold Logic

Venudhara B[1], Santhosh Kumar G[2]

M.Tech in Digital Electronics Dept of ECE, EWIT, Bengaluru, India[1]

Assistant Professor, Dept of ECE, EWIT, Bengaluru, India[2]

Abstract: High speed and low power utilization is critical configuration objectives in VLSI design circuits. Digital multipliers are most important functional unit. The major constraints for delay in any VLSI circuits are latency and throughput .The negative bias temperature instability (NBTI) effect happens when a pMOS transistor is under negative bias (Vgs= -VDD) increasing the threshold voltage of pMOS transistor and decreases the speed. A comparative phenomenon is done for nMOS, positive bias temperature instability, happens when an nMOS transistor is under positive bias. Both impacts will reflect on the performance of transistor speed, and in the long term, the system may fail due to violation of timing therefore in order to maximize the power consumption and delay ,multiplier with adaptive hold logic is used ,the multiplier is able to provide high throughput through variable latency and can adjust the AHL circuit ti mitigate performance degradation that is due to the aging effect . Additionally, the proposed design can be connected to a segment or column bypassing multiplier. The exploratory results demonstrate that our proposed design with 16 × 16 and 32 × 32 section bypassing multipliers can accomplish up to 62.88% and 76.28% execution change, respectively, contrasted and 16×16 and 32×32 fixed -latency section bypassing multipliers. Besides, our proposed design with 16 × 16 and 32 × A32 column bypassing multipliers can accomplish up to 80.17% and 69.40% performance improvement as compared with 16×16 and 32 × 32 fixed-latency row-bypassing multipliers. Furthermore we removed the tristate buffer from the column -bypass multiplier. With the goal that we can decrease the gate count and enhance the efficiency and speed and reduce the power consumption.

Keywords: Adaptive hold logic (AHL), negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), variable latency.

I. INTRODUCTION

Multiplication is one fundamental arithmetic operation for basic DSP applications , such as, filtering and fast Fourier transform (FFT). To accomplish high execution speed, parallel array multipliers are generally used. These multipliers have a tendency to consume most of the power in DSP computation, and along these lines power-effective multipliers are very critical for the design of low-power DSP systems. In the event that the multipliers are too slow, the performance of entire circuits will be decreased.

Besides, negative bias temperature instability (NBTI) happens when a pMOS transistor is under negative bias (Vgs = −Vdd). In this circumstance, the collaboration between inversion layer holes and hydrogen-passivity Si molecules breaks the Si–H bond created amid the oxidation procedure, producing H or H2 atoms. At the point when these particles diffuse away, interface traps are left out. The collected interface traps between silicon and the gate oxide interface result in increased threshold voltage (Vth), decreasing the circuit switching speed. At the point when the biased voltage is removed, the reverse reaction happens, diminishing the NBTI impact. Be that as it may, the reverse response does not dispense with all the interface traps generated at the time of stress phase, and Vth is increased in the long haul. Consequently, it is essential to develop a predictable high-performance multiplier The comparing impact on a nMOS transistor is positive bias temperature instability (PBTI), which happens when a nMOS transistor is under positive bias. Contrasted and the NBTI impact, the PBTI impact is much littler on oxide/polygate transistors, and in this manner is typically disregarded. Be that as it may, for high-k/metal-entryway nMOS transistors with huge charge trapping, the PBTI impact can no longer be disregarded. Actually, it has been demonstrated that the PBTI impact is huger than the NBTI impact on 32-nm high-k/metal-gate forms

Traditional circuits use basic path delay as the generally overall circuit clock cycle keeping in mind in order to perform effectively. Be that as it may, the probability that the critical paths are actuated is low. In most cases, the path delay is shorter than the basic path. For these noncritical ways, utilizing the basic way defer as the generally speaking cycle period will bring about noteworthy timing waste. Subsequently, the variable-latency design was proposed to decrease the timing waste of conventional circuits.

The variable-latency design partitions the circuit into two parts: 1) shorter paths and 2) longer paths. Shorter paths can execute effectively in one cycle, whereas longer paths need two cycles to execute. At the point when shorter path activated regularly, the normal average latency of

DOI 10.17148/IJIREEICE.2016.4824

variable-latency design is superior to that of conventional designs. For example, a few variable-latency adders were proposed utilizing the hypothesis method with error detection and recovery. A short path activation functional calculation was proposed to improve the precision of the hold logic and to advance the execution of the variable-latency circuit. A guideline planning calculation was proposed to plan the operations on nonuniform latency functional units and progress the performance of Very Long Instruction Word processors. In, a variable-inactivity pipelined multiplier design with a Booth calculation was proposed. In process-variety tolerant design for arithmetic units was proposed, where the impact of procedure variety is considered to expand the circuit .What's more, the basic ways are isolated into two shorter ways that could be unequal and the clock cycle is set to the postponement of the more one. These examination plans could lessen the planning misuse of conventional circuits to enhance execution, yet they didn't consider the maturing impact and couldn't change themselves amid the runtime. A variable-latency adder design that considers the aging effect was proposed. In any case, no variable-latency multiplier design that considers the aging effect and can adjust powerfully has been done.

II. EFFECT OF NBTI ON THE PERFORMANCE DEGRADATION OF DIGITAL CIRCUITS

.

NBTI is a noteworthy symptom on the lifetime dependability of integrated circuits. With the persistent scaling of transistor dimensions, the degradation quality debasement of circuits has turned into a vital issue. Because of an increasing electric field across the thin oxide, the generation of interface traps under negative bias temperature instability (NBTI) in pMOS transistors has become to be a one of the most basic reliable quality issues that decide the lifetime of CMOS device. due to NBTI, the threshold voltage of the transistor increase with time bringing about the decrease in drive current, which in thusly brings a temporal performance degradation of circuits.

Vth Degradation Model: NBTI is the result of trap era at Si/SiO interface in adversely based PMOS transistors at lifted temperatures. The association of reversal layer gaps with hydrogen passivity Si molecules can break the SiH bonds, making an interface trap and one H particle that can diffuse far from the interface or can strengthen a current trap.

Gate Delay Degradation Model: Delay of gate relies on upon threshold voltage esteem. Hence, by checking the threshold voltage degradation, the adjustment in gate delay can be effortlessly evaluated with a high level of precision From the above examination, plainly circuit delay relies on upon threshold voltage variation and henceforth performance degradation will happen. Different techniques used to moderate this impact are Vdd tuning, PMOS sizing, Tuning of gate length and Tuning of switching frequency. By this strategies we can lessen NBTI to a more noteworthy extend but area, power inefficiency is a major issue.

III PRELIMINARIES

A) column Bypassing Multiplier

The column bypassing multiplier is a change on the ordinary array multiplier (AM). The AM is a quick parallel AM as shown in Fig. 1. The multiplier exhibit comprises of (n−1) rows of carry save adder (CSA), in which every rows contains (n − 1) full adder (FA) cells. Every 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 propogation. The FAs in the AM are constantly active regardless of input states. In a low-power column bypassing multiplier configuration is proposed in which the FA operations are disabled on the off chance that the relating bit in the multiplicand is 0. Fig. 2 shows a 4×4 segment bypassing multiplier. Assuming the inputs are 10102 * 1111, it can be seen that for the FAs in the first and third diagonals, two of the three info bits are 0: the convey bit from its upper right FA and the incomplete item aibi . In this manner, the yield of the adders in both diagonals is 0, and the output sum bit is just equivalent to the third bit, which is the sum output of its upper FA.

Henceforth, the FA is changed to add two tristate gates and one multiplexer. The multiplicand bit ai can be utilized as the multiplicand bit ai can be utilized as the selector of the multiplexer to choose the output of the full adder, and ai can likewise be utilized as the selector of the tristate gate to turnoff the input path the full adder. On the off chance that ai is 0, the inputs of full adder are disabled, and the sum bit of the current full adder is equivalent to the sum bit from its upper full adder, accordingly lessening the power utilization of the multiplier. On the off chance that ai to 1, the normal sum result is chosen.

B). Row-Bypassing Multiplier

A low-power row-bypassing multiplier is also proposedto reduce the activity power of the AM. The operation of the low-power row-bypassing multiplier is similar to that of the low-power column-bypassing multiplier, but the selector of the multiplexers and the tristate gates use the multiplicator. Each input is connected to an FA through a tristate gate. When the inputs are 1111 * 1001, the two inputs in the first and second rows are 0 for FAs. Because b1 is 0, the multiplexers in the first row select ai b0 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 b2 is 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.

DOI 10.17148/IJIREEICE.2016.4824

A low-power row-bypassing multiplier is additionally proposed to decrease the action power of the AM. The operation of the low-power row bypassing multiplier is like that of the low-power section bypassing multiplier, yet the selector of the multiplexers and the tristate gates utilize the multiplicator

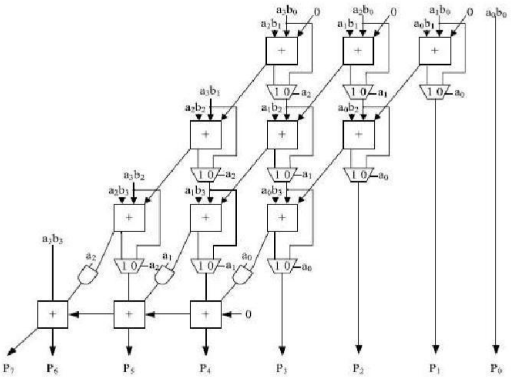


Fig.1: 4 × 4 column-bypassing multiplier.

.

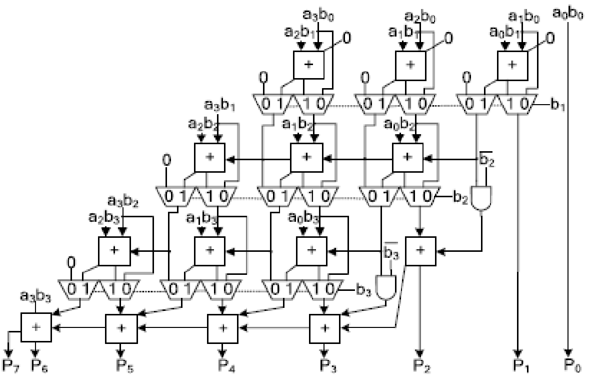


FIG 2: 4 × 4 row bypassing multiplier

Fig. 2 is a row 4 × 4 bypassing multiplier. Every information is associated with a FA through a restate gates. At the point when the inputs are 1111 * 1001, the two inputs in the first and second lines are 0 for FAs. Since b1 is 0, the multiplexers in the main column select aib0 as the total piece and select 0 as the convey bit. The inputs are skirted to FAs in the second lines; what's more, the tristate gates turn off the info ways to the FAs.

Hence, no exchanging exercises happen in the main line FAs; consequently, control utilization is diminished. Thus, in light of the fact that b2 is 0, no exchanging exercises will happen in the second-row FAs. Be that as it may, the FAs must be dynamic in the third line in light of the fact that the b3 is not zero. More points of interest for the column bypassing multiplier can likewise be found in [3].

C) Variable Latency Design

The 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

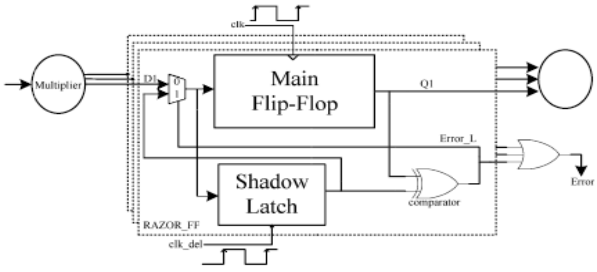


Fig 3: razor flip-flop

D).Razor Flip Flop:

Razor flip-flops is a 1-bit Razor flip-flop contains a fundamental flip-flop, shadow lock, XOR gate and mux. The fundamental flip-flop gets the execution result for the interconnected circuit utilizing a typical clock signal, and the shadow latch gets the execution result using the delayed clock signal, which is slower than the ordinary clock signal. In the event that the latched bit of the shadow latch is unique in relation to that of the principle flip-flop, this implies the path delay of the present operation surpasses the cycle period, and the primary flip-flop gets an inaccurate result. On the off chance that mistakes happen, the Razor flip-flop will set the error signal to 1 to tell the system to re execute the operation and inform the AHL circuit that a error has happened. We utilize Razor flip-flop to distinguish whether an operation that is thought to be a one-cycle path which can be finished in one cycle. If not, the operation is re executed with two cycles. Despite the fact that the re execution may appear to be much costlier, the general expense is low because of the fact that there execution recurrence is low.

E).Adaptive Hold Logic

The Adaptive Hold Logic circuit is the key component of variable-latency multiplier. The AHL circuit contains many blocks that will have a specific assignment of work such as decision block, MUX and a D flip-flop. 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, it means the circuit has suffered significant timing degradation due to the aging effect

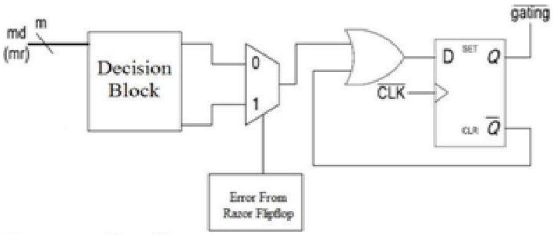


Fig.4 Diagram of AHL

DOI 10.17148/IJIREEICE.2016.4824

VI. 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.

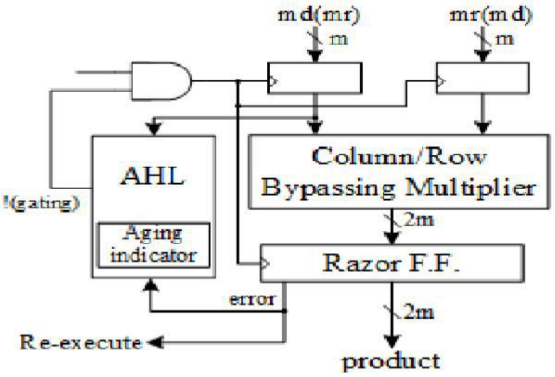


Fig 5 Proposed aging-aware multiplier architecture

The Fig. 5 shows our proposed aging-aware multiplier architecture, which includes two m-bit inputs , one 2m-bit output, one column- or row-bypassing multiplier, 2m 1-bit Razor flip-flops and an AHL circuit At the point when input patterns arrives at the input terminal, the row- or column bypassing multiplier, and the AHL circuit execute the input patterns parallely by examining number of zeros in the multiplicand (multiplicator), the AHL circuit chooses if the input patterns require maybe a cone or two cycles ti finish the cycle. In the pattern of the input requires two cycles to finish the cycle, the AHL will output 0 to shuts the clock sign of the flip-flops. Something else, the AHL will output 1 in general for simple operations. At the point when the column or row bypassing multiplier completes the operation, the outcome will be passed to the Razor flip-flops. The Razor flip-flops check whether there is the path violate the timing criteria. In the case if the criteria fails, it implies the cycle period is not sufficiently longer for the execution operation to complete and that execution result of the multiplier is erroneous. In this way, the Razor flip-flop will examine an error to advise the sytem that the executed operation should be reexecuted which may utilizing two cycles to guarantee the operation is error free. In this circumstance, the additional reexecution cycles brought about by timing violation put an penalty for the overall latency . In any case, our proposed AHL circuit can predict with an acuuracy whether the pattern required 1 or 2 cycles in some case. Just or very few input pattern will have the violation of timing criteria calculated by AHL circuit mistakenly. For this situation, the additional re execution cycles did not create a timing degradation.

V. RESULTS

In this section the experimental results and simulated outputs of the proposed circuit ,This AHL Multiplier is built by Verilog Code and simulated by using Modelsim and synthersized by Xilinx

A).simulated results:

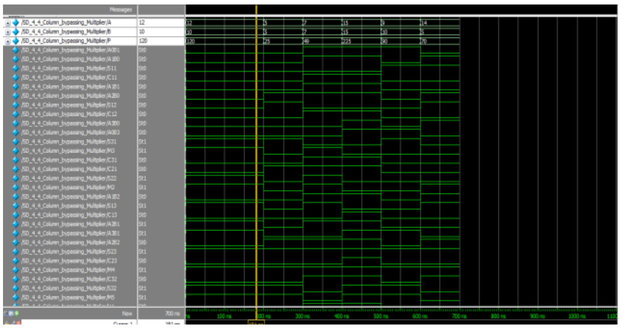

Messages
/SD_4_4_Column_bypassing_Multiplier/A
12
/SD_4_4_Column_bypassing_Multiplier/B
10
/SD_4_4_Column_bypassing_Multiplier/P
120

Fig-6: row-bypassing multiplier simulated output

This simulated output as shown in fig-6 is obtained for 4*4 input 4 bit row bypass multiplier and the output is F0-F3 which has input X0-X3,it has a wired connection of A0-A3 and B0-B3 the connection is simulated and obtained results using modelsim software,the programme is written using verilog programme in order to simulate the programme it must be assigned all the inputs and outputs by the instruction After completing the assigning inputs compiled using compiler and run the simulation, the respective results is obtained for the input 1001 and 1001

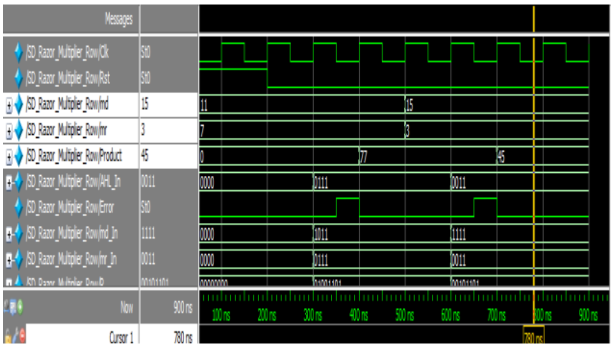
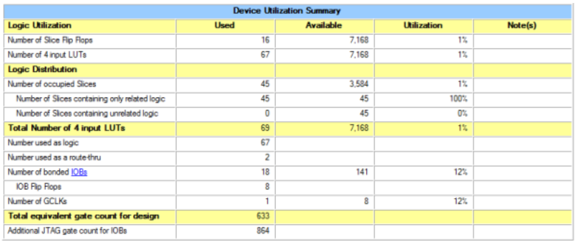

Messages
/SD_Razor_Multiplier_Row/Clk
St0
/SD_Razor_Multiplier_Row/Rst
St0
/SD_Razor_Multiplier_Row/md
15
11
15
/SD_Razor_Multiplier_Row/mr
3
7
3
/SD_Razor_Multiplier_Row/Product
45
0
77
45
/SD_Razor_Multiplier_Row/AHL_In
0011
0000
0111
0011
/SD_Razor_Multiplier_Row/Error
St0
/SD_Razor_Multiplier_Row/md_In
1111
0000
1011
1111
/SD_Razor_Multiplier_Row/mr_In
0011
0000
0111
0011
Now
900 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
Cursor 1
780 ns
780 ns

Fig-7: multiplier row with razor flip-flop

This is the simulated output for the combination of row multiplier with the razor flip-flop, in this result the input for the multiplier is given 15 and 5 ,in decimal format the output for the combined is shown in the wave format as shown in the above figure

B). Device Utilization Summary:



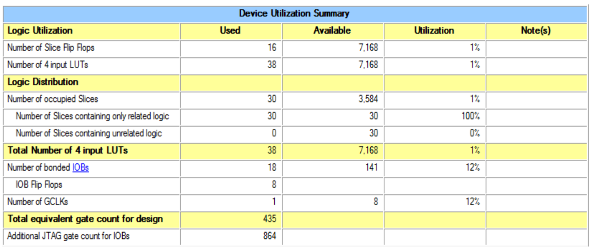
Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	16	7,168	1%	
Number of 4 input LUTs	67	7,168	1%	
Logic Distribution				
Number of occupied Slices	45	3,584	1%	
Number of Slices containing only related logic	45	45	100%	
Number of Slices containing unrelated logic	0	45	0%	
Total Number of 4 input LUTs	69	7,168	1%	
Number used as logic	67			
Number used as a route-thru	2			
Number of bonded IOBs	18	141	12%	
IOB Flip Flops	8			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	633			
Additional JTAG gate count for IOBs	864			

Fig 8: Row based multiplier utilization summary

DOI 10.17148/IJIREEICE.2016.4824

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

The above shown Fig-8 is obtained from Xilinx for Row based aging-aware multiplier design. The device utilization summary shows that the number of flip-flops ,LUTs and the gate count used for the operation of the circuit ,it shows clearly that the usage of the flip-flops and LUTs in the logic operation will be less as compared to the traditional method



Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	16	7,168	1%	
Number of 4 input LUTs	38	7,168	1%	
Logic Distribution				
Number of occupied Slices	30	3,584	1%	
Number of Slices containing only related logic	30	30	100%	
Number of Slices containing unrelated logic	0	30	0%	
Total Number of 4 input LUTs	38	7,168	1%	
Number of bonded IOBs	18	141	12%	
IOB Flip Flops	8			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	435			
Additional JTAG gate count for IOBs	864			

Fig 9: column based multiplier utilization summary

The above Fig-9 is obtained from Xilinx for Column based aging-aware multiplier design utilization . The device utilization summary shows that the number of flip-flops ,LUTs and the gate count used for the operation of the circuit ,which give the overall device utilization .

C).Area delay power comparison

This table-1 will provide the area delay power comparison between proposed and existing multiplier We can clearly have a look, that the proposed model has many advantage than the existing model, in the area and power as see in this table the gate count ,LUTs ,slices and delay ,the proposed model has a drastic change in the delay ,delay will be reduced by half as compared to existing model, our proposed model has high overall efficiency as per the result .

METHOD NAME	AREA IN NUMBER OF LUT			DELAY		
Multiplier Design	LUT	GATE COUNT	SLICES	DELAY	GATE OR LOGIC DELAY	PATH OR ROUTE DELAY
Existing Multiplier	30	186	15	18.176ns	10.131ns	8.045ns
Proposed Aging Aware Multiplier Row	69	633	16	7.367ns	6.364ns	1.003ns
Proposed Aging Aware Multiplier Column	38	435	16	7.285ns	6.364ns	0.921ns

Table 1: Area delay power comparison

REFERENCES

[1] A.K. Verma, P. Brisk, and P. Ienne, “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” in Proc.,pp. 1250–1255,2008 .

[2] 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.

[3] Baneres, J. Cortadella, and M. Kishinevsky, “Variable-latency design by function speculation,” in Proc., pp. 1704–1709,2009.

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

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

[6] Mohapatra, G. Karakonstantis, and K. Roy, “Low-power processvariation tolerant arithmetic units using input-based elastic clocking,” in Proc. ACM/IEEE ISLPED, pp. 74–79 ,Aug. 2007.

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

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

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

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

[11] 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.

[12] 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, pp. 253–258 Aug. 2010.

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

[14] R. Vattikonda, W. Wang, and Y. Cao, “Modeling and mimization of pMOS NBTI effect for robust naometer design,” in Proc.ACM/IEEEDAC, pp.1047–105 ,Jun.2004.

[15] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov.1999.

[16] 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 ,pp. 23–25 2006.

[17] 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 pp. 603–608, 2011.

DOI 10.17148/IJIREEICE.2016.4824