IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 6, Ver. III (Nov - Dec .2015), PP 27-33
www.iosrjournals.org

Implementation of a High Speed and Power Efficient Reliable Multiplier Using Adaptive Hold Technique

Padala.Abhishek.T.S.[1], Dr. Shaik.Mastan Vali, M.E, Ph.D[2]
[1]*M.Tech.Student, Department of ECE, MVGR College of Engineering, Vizianagaram, India.*
[2] *Professor, Department of ECE, MVGR College of Engineering, Vizianagaram, India*.

Abstract: *Digital multipliers, the most important part which is used to implement most of the digital processing and arithmetic applications such as Filters, FFT's, etc. As the rapid developments in technology required, many researchers are going to design multipliers which offers an efficient design aspects with respect to the speed and power consumption. Digital multipliers are one of the most essential arithmetic functional units. The overall performance of these systems depends on the multiplier's throughput. The negative bias temperature instability (NBTI) effect occurs when a pMOS transistor is under negative bias (Vgs = −Vdd), which results the increase in the threshold voltage of the pMOS transistor, hence the delay in the multiplier is increased. In the same way, positive bias temperature instability, occurs when an nMOS transistor is under positive bias. Both the temperature effects decreases the transistor speed, in the long run, the system may fail due to timing violations. Hence, there is a need to design a high speed and power efficient multiplier to increase the performance of the device. In this paper Design of Multiplier circuit using Adaptive Hold Technique is proposed. By using AHT circuit we can reduce the NBTI & PBTI effects, hence performance in terms of delay will be increased and the aging effects will be reduced.*

Keywords*: Adaptive hold Technique (AHT), Metal oxide semiconductor (MOS), Negative bias temperature instability (NBTI), positive bias temperature instability (PBTI).*

I. Introduction

A multiplier is one of the most important unit which plays a vital role among most of the critical arithmetic functional units in many applications such as Discrete cosine transforms (DCTs), digital filtering, Fourier transform (FTs), The rate of production of these application devices depends on multipliers. If the multipliers have high delay then the overall performance of the device will be reduced.

Moreover, there are some temperature instability effects such as NBTI (Negative bias temperature instability) and PBTI (Positive bias temperature instability) occurs at the harsh environment in some applications like aerospace applications, chemical industries and in various nuclear applications. The NBTI occurs when a pMOS transistor is under negative bias (*V*gs=−*V*dd). 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 get diffused, there will be incorporate traps will be left. These traps between silicon and the gate oxide lead to increase in threshold voltage (*V*th) and automatically show an impact on reducing the speed in circuit switching. When the biased voltage is removed, the reverse reaction occurs, and then NBTI effect will be reduced. However, the reverse reaction does not eliminate all the interface traps generated during the stress phase, and *V*th is increased in the long term. Hence, it is important to design a more reliable high-performance multiplier which must have reduced threshold voltage so, that NBTI can be reduced.

The similar effect on an nMOS transistor is 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. Whereas, for high-*k*/metal-gate nMOS transistors with significant charge trapping, the PBTI effect can no longer be ignored. In open literature it is shown that the PBTI effect is more significant than the NBTI effect on 32-nm high-*k*/metal-gate processes [2].

A traditional method to mitigate the aging effect is overdesign [3], including such things as guard-banding and gate over sizing; however, this approach are area and power inefficient. To avoid this problem, many NBTI-aware methodologies have been proposed. An NBTI-aware technology mapping technique was proposed by Yang etal., Calmera.A etal,.[3-6] to guarantee the performance of the circuit during its lifetime. B.C.Paul, etal designed an NBTI-aware sleep transistor [7] to reduce the aging effects on pMOS sleep-transistors, and the lifetime stability of the power-gated circuits under consideration was improved. Y Y. Chen et al [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. Y.Lee etal., M.Basoglu etal., [9, 10] proposed dynamic voltage scaling and body-basing

techniques to reduce power or extend circuit life. However, these techniques require circuit modifications or optimization in specific circuits.

Traditional circuits use critical path delay as the overall circuit clock cycle in order to perform correctly. However, the probability that the critical paths are activated is low. In most cases, the path delay is shorter than the critical path. For these noncritical paths, using the critical path delay as the overall cycle period will result in significant timing waste. Hence, the proposed adaptive hold multiplier is designed to reduce the delay as well as to eliminate the temperature instabilities.

II. Existing Methods

2.1. Array Multiplier

The Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length.

				A3	A2	A1	A0	Inputs
			x	B3	B2	B1	B0	
			C	B0 x A3	B0 x A2	B0 x A1	B0 x A0	
		+	B1 x A3	B1 x A2	B1 x A1	B1 x A0		
		C	sum	sum	sum	sum		
	+	B2 x A3	B2 x A2	B2 x A1	B2 x A0			Internal Signals
	C	sum	sum	sum	sum			
+	B3 x A3	B3 x A2	B3 x A1	B3 x A0				
C	sum	sum	sum	sum				
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Outputs

Fig.1 Multiplication Process

The AM is a fast parallel multiplier and the multiplication process is as shown in Fig. 1, and Fig.2 shows the block diagram of Array Multiplier. It 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. In the results 16bit, 32bit array multipliers is designed and compared.

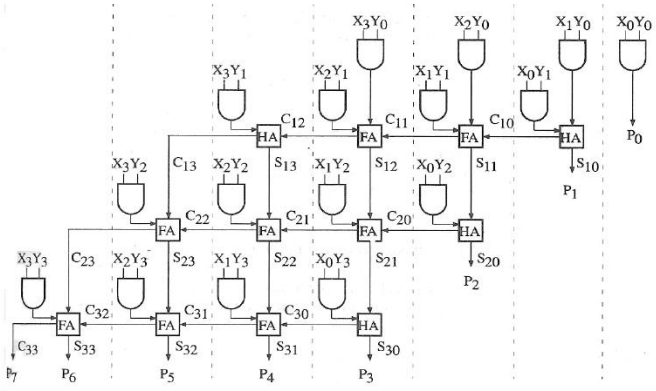

Fig.2 Array Multiplier

2.2 Column Bypassing Multiplier

A column-bypassing multiplier is an advanced multiplier when compared to the traditional array multiplier (AM). A low-power column-bypassing multiplier design is proposed to reduce power and delay as well. According to Column Bypasing Multiplier the FA operations are disabled with the corresponding bit in the multiplicand is 0. Fig.3 shows the architecture of 4×4 column-bypassing multiplier. In open literature the column-bypassing multiplier is available [11] it is given by M.C.Wen, etal.

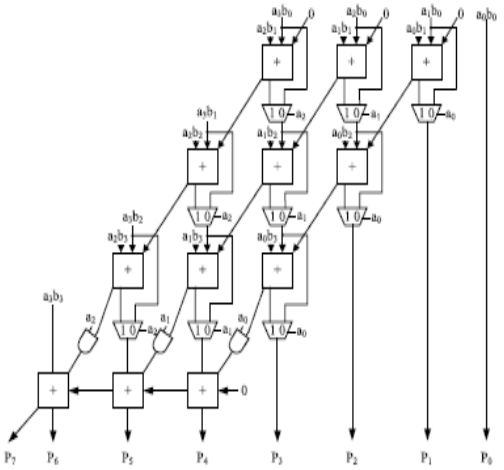

Fig.3 Column Bypassing Multiplier

2.3 Row-Bypassing Multiplier

A low-power row-bypassing multiplier [13] is also proposed to reduce the activity power of the AM. The internal Architecture of the Row bypassing multiplier is as shown in the Fig.4. The operation of the low-power row-bypassing multiplier is nearer as that of the low-power column-bypassing multiplier, but the difference is the selector of the multiplexers and the tristate gates. 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 *b*2 is 0, no switching activities will occur in the second-row FAs. However, the FAs must be active in the third row because the *b*3 is not zero.

More detailed information for the row-bypassing multiplier is given in the open literature [12] by J.Ohban etal.

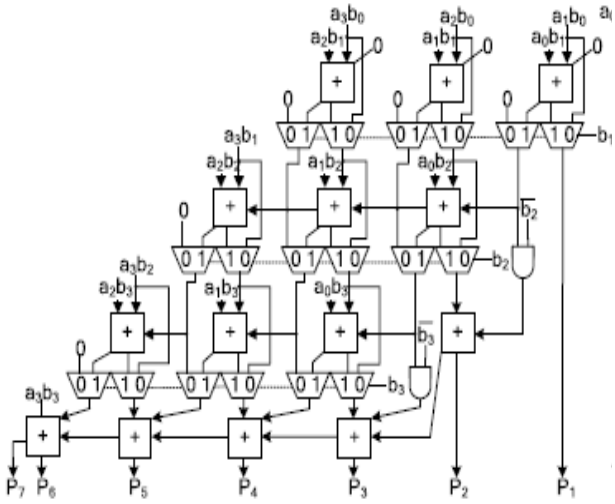

Fig.3 Row-bypassing multiplier.

III. Proposed Technique

The aging-aware reliable multiplier is designed by interlinking the Adaptive Hold Technique to the either Row-bypassing or Column- bypassing multipliers. The proposed AHT Architecture consists of different blocks such as of 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 AHT circuit. The overall architecture of the Aging Multiplier is as shown in Fig.4.

As 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 AHT. According to the bypassing selection in the column or row-bypassing multiplier, the input signal of the AHT in the architecture with the column-bypassing multiplier is the multiplicand (Md), whereas that of the row-bypassing multiplier is the multiplicator

(mr). Razor flip-flops can be used to detect whether timing violations occur before the next input pattern arrives. The functioning of the each module in the proposed multiplier is illustrated as follows:

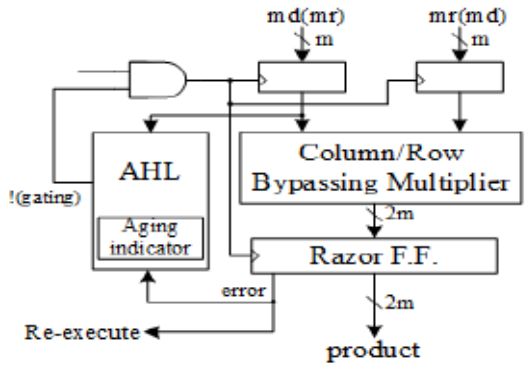

Fig.4 Architecture of Aging Aware Multiplier

3.1 Razor flip flop

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 multiplexer. 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 reexecute the operation and notify the AHL circuit that an error has occurred.

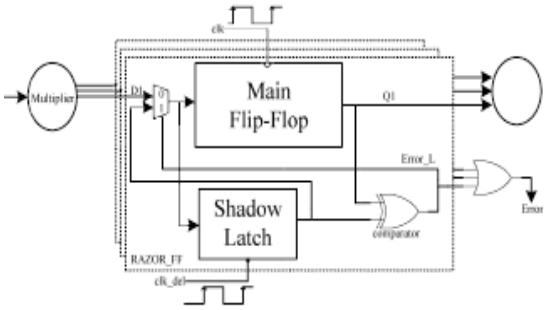

Fig.5 Razor Flip-flop

3.2 Adaptive hold Technique

The Adaptive Hold Technique circuit is the key component of variable-latency multiplier. The AHL circuit contains 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.

The overall flow of the proposed architecture is as follows: when input pattern is arrived to the column or row bypassing multiplier, the AHT circuit execute simultaneously. According to the number of zeros in the multiplicand or multiplicator, the AHT circuit will decides whether the input patterns require one or two cycles. If the input pattern requires two cycles to complete, the AHT will output 0 to disable the clock signal of the flip-flops. Otherwise, the AHT will output 1 for normal operations. When the column or row bypassing multiplier finishes the operation, the result will be passed to the Razor flip-flops. The Razor flip-flops check whether there is the path delay timing violation. If timing violations occur, it means the cycle period is not long enough for the current operation to complete and that the execution result of the multiplier is incorrect. Thus, the Razor flip-flops will output an error to inform the system that the current operation needs to be 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, the proposed AHT 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 AHT circuit judges incorrectly. In this case, the extra reexecution cycles did not produce significant timing degradation.

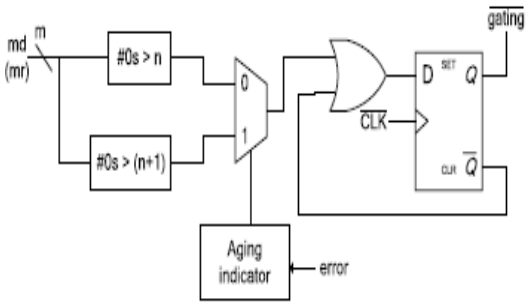

Fig.6 Internal Architecture of AHT circuit

IV. Results

The traditional multipliers and proposed multiplier with 16bit, 32bit are designed in Verilog using Xilinx tool 14.1 version and the simulations are observed with ISE simulator. Xilinx Synthesizer is used to analyse the delay and the Xpower analyzer is used to analyse the Static and Dynamic power. Fig.7- Fig.11 are the simulation results for 16 bit multipliers, Fig.12 - Fig.16 are simulation results for 32 bit multipliers. Table.1 and Table.2 are the synthesis results of 16 and 32bit multipliers respectively.

4.1 Simulations results

Fig.7 16x16 Array Multiplier

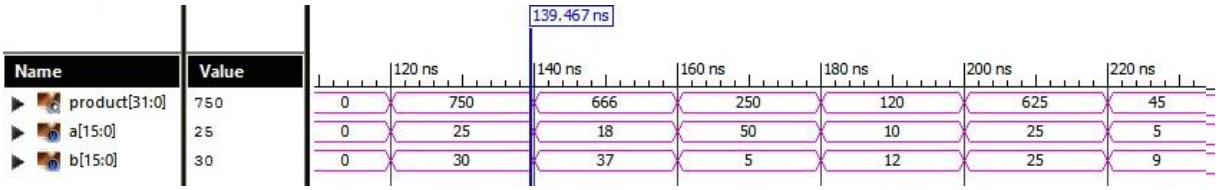

Fig.8 16x16 Row bypassing

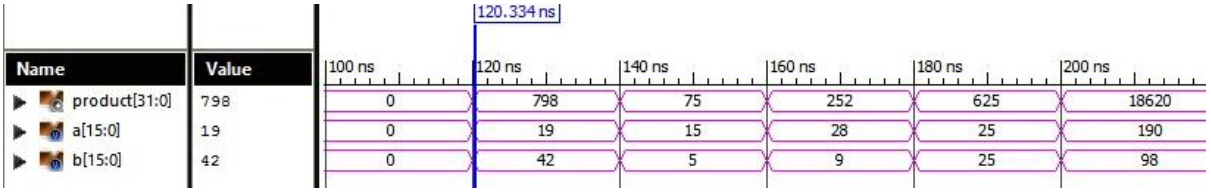

Fig.9 16x16 Column bypassing

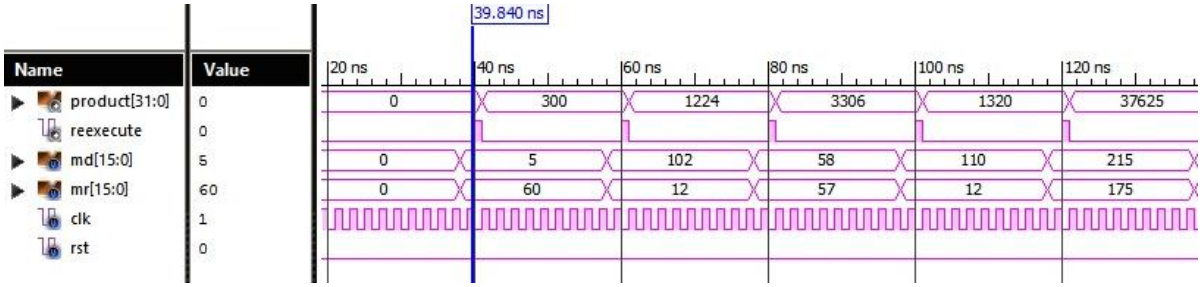

Fig.10 16x16 AHT using row bypassing

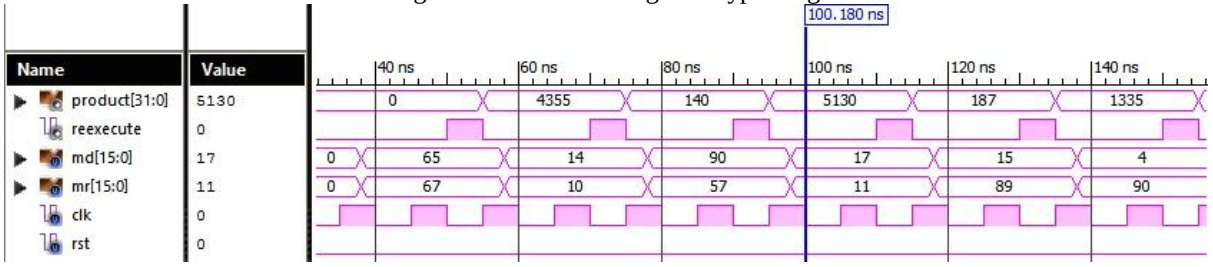

Fig.11 16x16 AHT using column bypassing

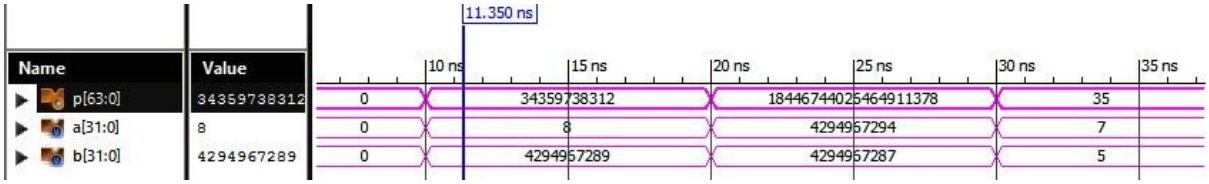

Fig.12 32x32 Array Multiplier

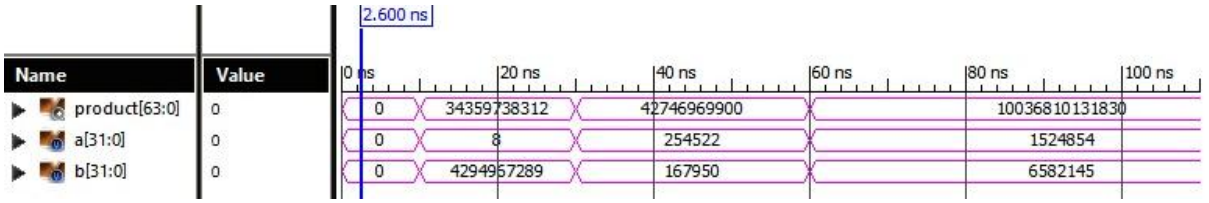

Fig.13 32*32 Row bypassing

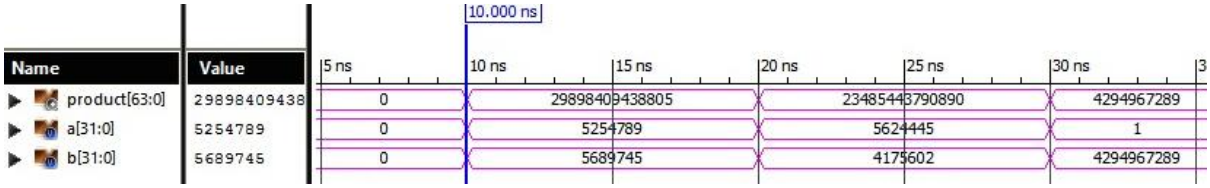

Fig.14 32*32 Column bypassing

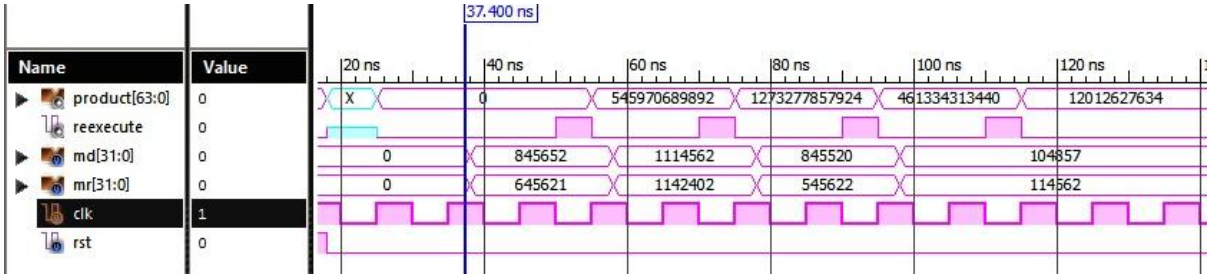

Fig.15 32*32 AHT using row bypassing

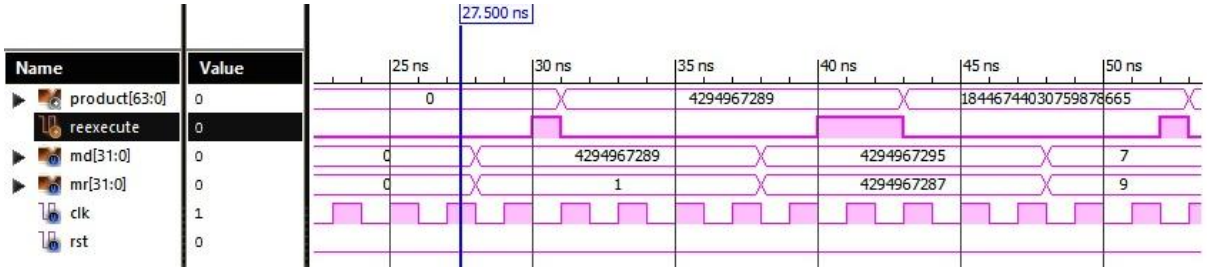

Fig.16 32*32 AHT using column bypassing.

4.2 Synthesis results

Table.1 Comparison table of 16*16 multipliers

NAME	DELAY in ns	POWER (W)			AREA		
		Quiescent Power (W)	Dynamic Power (W)	Total Power (W)	LUTs	FFs	SLICES
Array	41.821	0.081	0.12	0.201	701	-	389
Row Bypassing	32.162	0.081	0.061	0.142	857	-	476
Column Bypassing	30.872	0.081	0.04	0.121	726	-	409
AHT Row	6.420	0.081	0.008	0.089	1375	130	712
AHT Column	6.420	0.081	0.005	0.085	1199	132	626

Table.2 Comparison table of 32*32 Multipliers

NAME	DELAY in ns	POWER (W)			AREA		
		Quiescent Power (W)	Dynamic Power (W)	Total Power (W)	LUTs	FFs	SLICES
Array	73.619	0.081	0.142	0.223	2963	-	1647
Row Bypassing	53.467	0.081	0.088	0.169	3640	-	2018
Column Bypassing	51.973	0.081	0.075	0.156	3063	-	1720
AHT Row	6.463	0.081	0.009	0.09	5153	259	2653
AHT Column	6.466	0.081	0.005	0.086	4852	271	2676

V. Conclusion

In this paper the proposed Technique for multiplier design has three important features. First, its Delay is very less when compared to the Traditional Multipliers. 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. Last but not least, the AHT architecture is power efficient and it can also adjust the percentage of one-cycle patterns to minimize performance degradations due to the aging effect. When the circuit is aged, and many errors occur, the AHT circuit uses the second judging block to decide if an input is one cycle or two cycles and hence the timing errors can also be eliminated and can perform the error free operations. As the AHT Technique can be implemented by using both Row and Column bypassing multipliers, the 32bit AHT row bypassing multiplier can perform its task with decrease of 47.004ns delay and 79milli-watts power dissipation when compared to normal row multiplier. Similarly, the AHT column by passing multipliers can perform its task with decrease of 45.507ns delay and 70milli-watts power dissipation when compared to normal column multiplier. The proposed AHT architecture has a disadvantage in terms of size as the number of LUTs and Slices are more in addition the Flip-flops are also used in order to increase the speed. But the AHT Multipliers has a great advantage in the both Power and Delay as well but and hence, it can be stated as the reliable multiplier technique which can be used in harsh environment mostly in aerospace applications etc.

References

[1] Lin.I.C., Cho.Y.H. and Yang.Y.M., Aging-Aware Reliable Multiplier Design With Adaptive Hold Logic, IEEE Transactions on Very Large Scale Integration(VLSI) Systems, 23(3), 2015, 544-556.

[2] S.Zafar, A. Kumar, E. Gusev, and E. Cartier, Threshold voltage instabilities in high-k gate dielectric stack,IEEE Trans. Device Mater, 5(1), 2005, 45–64.

[3] Yang, Hao.I, Shyh.Chyiyang, Wei Hwang and Ching Te Chuang, Impacts of NBTI/PBTI on Timing Control Circuits and Degradation Tolerant Design in Nanoscale CMOS SRAM, IEEE Transactions on Circuits and Systems I, 58(6), 2011, 1239-1251.

[4] Calimera.A, Macii.E. and Poncino.M., Design Techniques for NBTI-Tolerant Power-Gating Architectures, IEEE Transactions on Circuits and Systems II, 59(4),2012,249- 253.

[5] Paul.B.C., Kunhyuk Kang, Kufluoglu.H., Alam.M.A. and Roy.K., Impact of NBTI on the temporal performance degradation of digital circuits, IEEE Transactions on Electron Device Letters, 26(8), 2005, 560-562.

[6] 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 Transaction on Computer Aided Design and Integrated Circuits Systems, 26(4), 2007, 743–751.

[7] Olivieri.N, Design of synchronous and asynchronous variable-latency pipelined multipliers, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 9(2), 2001, 365-376.

[8] Y.Y.Chen, Variable-latency adder (VL-Adder) designs for low power and NBTI tolerance, IEEE Transaction on Very Large Scale Integration (VLSI) Systems, 18(11), 2010, 1621–1624.

[9] M.Basoglu, M.Orshansky and M.Erez, NBTI-aware DVFS, A new approach to saving energy and increasing processor lifetime: Proc.ACM/IEEE ISLPED, 2010, 253–258.

[10] D.Mohapatra, G.Karakonstantis and K.Roy, Low-power process variation tolerant arithmetic units using input-based elastic clocking: Proc.ACM/IEEE ISLPED, 2007, 74–79.

[11] M.C.Wen, S.J.Wang and Y.N.Lin, Low power parallel multiplier with column bypassing: Proc. of IEEE ISCAS, 2005,1638–1641.

[12] J.Ohban, V.G.Moshnyaga and K.Inoue, Multiplier energy reduction through bypassing of partial products: Proc. APCCAS, 2002.