Design and performance analysis of aging aware reliable multiplier with adaptive hold logic

[1]Pooja Beniwal, [2]Monika Gupta

[1]M Tech Student, [2]Assistant Professor
VLSI Design,
F.O.T.U.T.U Dehradun

***Abstract*— Multipliers are the major elements to perform arithmetic function in devices but aging effect degrades the speed of multipliers and in long term system may fail due to timing violations. . For reducing these timing violations an aging aware multiplier with adaptive hold logic is designed. Performance analysis of an aging-aware variable latency multiplier design with different adders is done in this paper. An aging-aware reliable multiplier where column/row bypassing multiplier is designed with carry save adder and another aging aware reliable multiplier where column/row bypassing multiplier is designed with Brent Kung Adder to reduce the aging induced degradation of multiplier further. The performance evaluation is done on Xilinx 14.7, coding is done using Verilog.**

Index Terms: Negative bias temperature instability (NBTI), Positive bias temperature instability (PBTI), Hot carrier injection (HCD), Electro migration (EM), Carry save adder (CSA), Brent Kung adder (BKA), Row bypassing multiplier (RBM), Column bypassing multiplier (CBM).

I. INTRODUCTION

Circuits have constantly aged however these aging effects weren't vital till the circuits enter nanotechnology area. The concurrent use of higher operating frequencies and tremendously small channel length has raised circuit aging from an academic exercise to growing and perhaps deterministic concern. Most aged systems fail because of delay problem so delay is the important issue that increases due to aging. Aging variations in circuits occurs owing to bias temperature instability (BTI), time dependent dielectric breakdown (TDDB), hot carrier injection (HCI), and electro migration (EM) causes the circuit to degrade over time. NBTI for instance is thought of as the worst aging effect nowadays. Multipliers are the principal components to perform arithmetic operation in devices however outcomes of aging effects like NBTI, PBTI, HCD and TDDB degrade transistor speed and multiplier speed is reduced in long run system might also stop functioning because of timing violations. The outcome of many applications i.e fourier transform, discrete cosine transforms, digital filtering etc relies upon on multipliers. The performance of complete circuits will decrease, if the multipliers are too slow.

Within the conventional circuit critical path delay is used as general circuit clock cycle but for non-critical path using critical path as general circuit clock cycle will result in timing wastage. Variable-latency technique[17] lessen the timing waste occurring in conventional circuits that use the critical path cycle as an general circuit clock cycle with a view to perform effectively. The basic concept of variable latency design is to perform operation that required the shorter path with the usage of shorter cycle and longer path using longer clock cycle. Considering that maximum paths execute in a cycle duration this is a lot smaller than the critical path delay i.e. probability that the critical path is activated is less consequently the average delay for variable-latency layout is smaller.

Objective of paper

In this paper, performance analysis of an aging-aware reliable multiplier design adaptive hold logic circuit with different adder is done. The AHL circuit can be adjusted to attain reliable operation under the impact of aging mechanisms. Here multiplier is based on variable latency technique. The objectives of this paper are summarized as follows. To design an aging-aware reliable multiplier where column/row bypassing multiplier will be designed with carry save adder and to design another aging aware reliable multiplier where column/row bypassing multiplier will be designed with Brent Kung Adder to reduce aging induced degradation further. Comprehensive analysis and comparison of the aging aware multiplier with carry save adder and aging aware multiplier with Brent Kung adder to show the effectiveness of our proposed architecture.

II. AGING MECHANISMS

Bias Temperature Instability

Bias Temperature Instability is a phenomenon in which threshold voltage increases gradually over a long period of time causing the drive current to decrease. Depending on the transistor state (on/off), BTI has two phases known as stress and recovery. During the stress phase the transistor is on, the reaction between inversion layer charge carrier and hydrogen passivated silicon break the Si-H bonds formed during the oxidation process, H and H_2 molecules will be generated. When these molecules diffuse away interface traps is generated (see Fig. 1). Furthermore, the generated traps become active and contribute to the increase of the transistor threshold voltage.

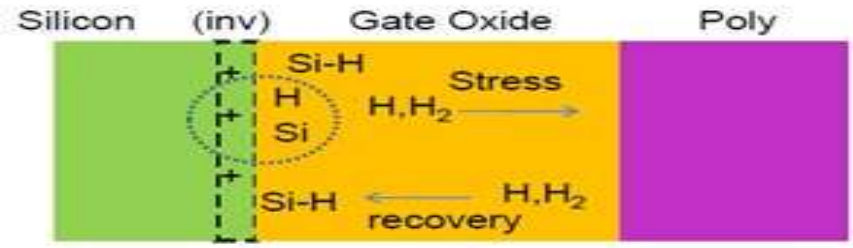

Fig-1: Stress and recovery phase of NBTI

When the transistor is off and biased voltage is removed transistor enters in the recovery phase and the escaped H atom start to go back towards the SiO_2 interface and recombine with the Si atom reforming the Si-H bonds (see Fig. 1). As a result, some of the charged traps generated during the stress phase become inactive again. However, during the recovery phase, not all the H atoms are defused back to their original positions and reform the bond, which in long term eventually results in a net V_{th} shift .
NBTI mechanism only occurs in PMOS devices when pmos transistor is provided with negative bias PBTI only occurs in NMOS devices when in nmos transistor is provided with positive bias. NBTI is the result of a combination of hole trapping ΔNht due to process related preexisting defects and generation of interface states at the channel oxide interface ΔNit PBTI result of combination of electron trapping with trap generation. PBTI only seems to be observed in high-k NMOS devices whereas NBTI is active for simple oxide gate dielectrics the impact of PBTI in high-k NMOS devices is comparable or even worse than NBTI.

Hot Carrier Injection

Some electrons and holes being accelerated by a strong electric field inside a semiconductor gain high kinetic energy and this is called hot carrier. Due to their high kinetic energy, hot carriers can get injected in prohibited region rather there intended path and get caught in prohibited region. When injected in such region they get caught or cause the generation of interface that causes the device performance to degrade. The term 'hot carrier degradation', in this way, refers to device degradation and caused by hot carrier injection.

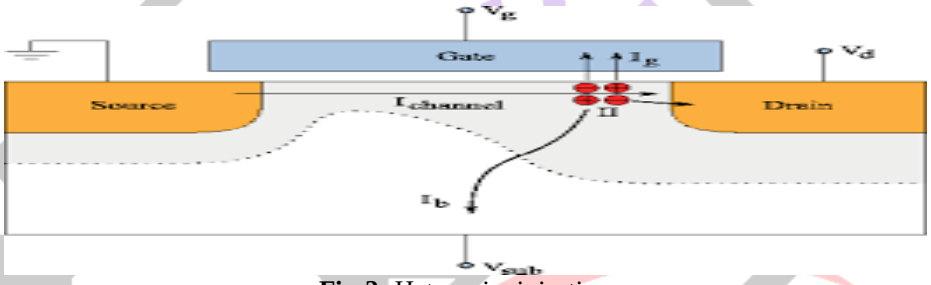

Fig-2: Hot carrier injection

Time Dependent Dielectric Breakdown

TDDB is result of high operating electric fields in the gate dielectric. The thin gate oxide and trend in supply voltage scaling results in large electric fields across the thin dielectric layer due to which traps are formed in the dielectric. The tunneling of carriers is assisted by these traps leading to the tunneling current. This step can be termed as soft oxide breakdown (SBD). Although these traps may lead to performance degradation, and cause shift in parameters such as threshold voltage shift and random gate leakage, the device will still function. Initially the generated traps are non-overlapping and thus do not conduct but as more traps are generated, they start to overlap and conducting path is formed. The breakdown is caused due the formation of conducting path through the gate oxide to substrate due to electron tunneling current.

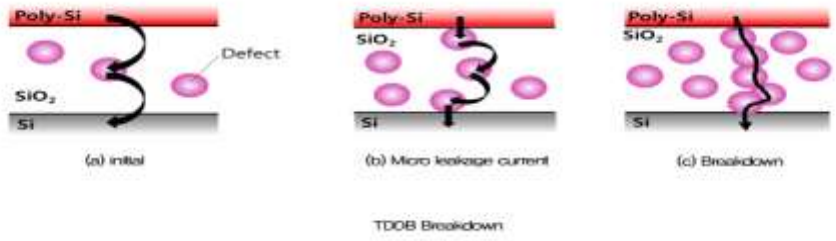

Fig- 3: Time Dependent Dielectric Breakdown

Once the conduction path is formed, more trap generation causes wider conduction path and hence more current flows which leads to a higher temperature. This thermal runaway condition leads to a catastrophic failure known as the hard oxide breakdown (HBD). After HBD, the device does not function properly.

Electro-migration (EM)

Electro-Migration (EM) is an aging mechanism which occurs in interconnects wires, contacts and vias in an integrated circuit. Electro-migration occurs when a surge of current knocks metal atom free and makes them to drift along with the flow of electron. This depletes the metal of some of its atom upstream or downstream. If depletion of metal is upstream this causes buildup of

metal downstream. This upstream thinning of metal increases the resistance of the connection sometimes to the point that it can turn into an open circuit. While downstream deposition isn't similarly catastrophic it can cause metal to buldge out of its assign tracks.

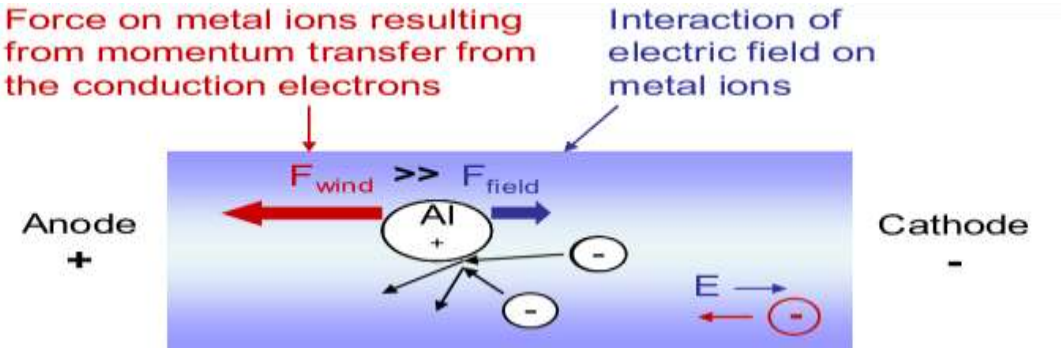

Fig-4: Two forces acting on metal electro migration is the result of the dominant force.

Current flow through a conductor produces two forces i.e. electrostatic force F_{field} and wind force F_{wind} to which the individual metal ions in the conductor are exposed. The first is an electrostatic force F_{field} caused by the electric field in the metallic interconnects. Since the positive metal ions are protected to some extent by the negative electrons in the conductor, this force can be ignored in most cases. The second force Fwind is generated by the energy exchange between conduction electrons and metal ions. This force acts in the direction of the current flow and is the main cause of electro migration (see Fig. 4).

III. AGING AWARE MULTIPLIER

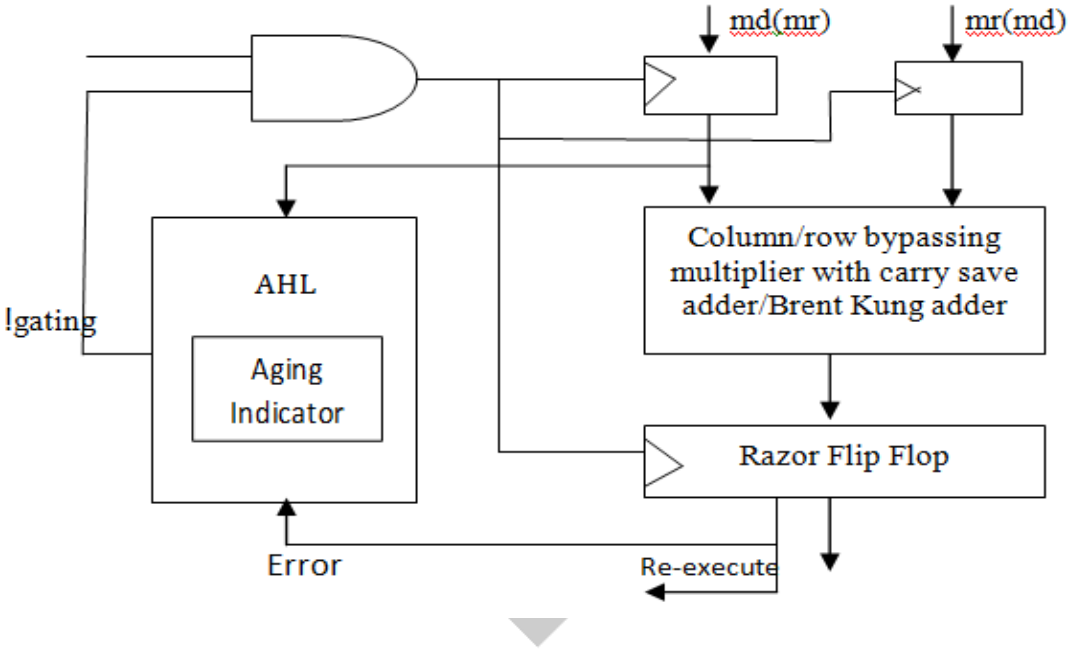

Fig-5: Aging aware multiplier with CSA and BKA

Row Bypassing Multiplier

In row bypassing multiplier the corresponding multiplier bit is the enable input to the three state gates and multiplexers. If this multiplier bit is 0 then the tri-state gates goes into high impedance state and thus inputs are not given to the full adder. The previous sum is taken as the present sum. If this bit is 1 then the tri-state gates get enabled and the inputs are given to the full adder. Thus the sum is generated and this is taken as the present sum.

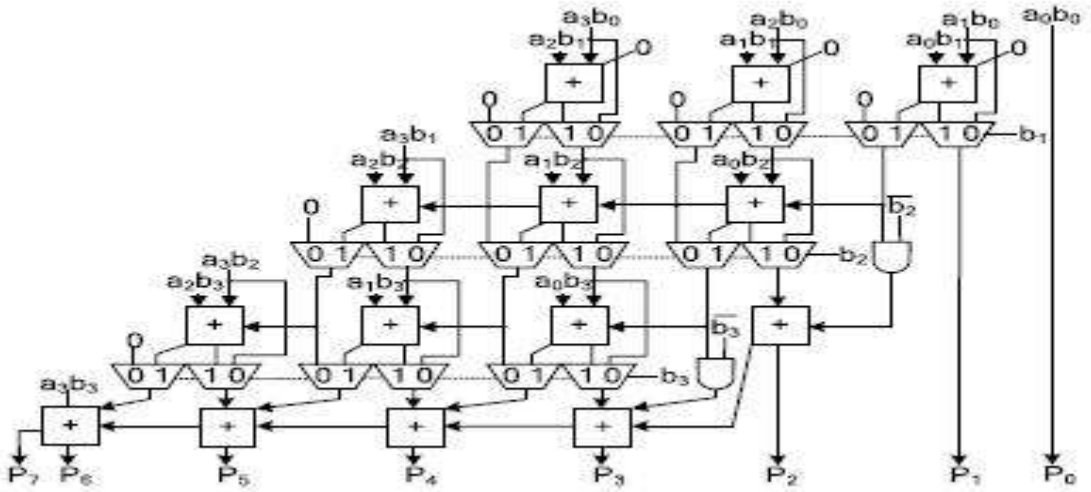

Fig-6: Row Bypassing Multiplier

Let us take an example of 1110 x 1001. Here, in the first and the second positions, multiplier bit consists of zero. During multiplication, adders of first and second row get disabled and the previous sum is taken as the current sum.

Column Bypassing Multipliers

In column bypassing multiplier the corresponding multiplicand bit is the enable input to the tri state gates and multiplexers. If this multiplicand bit is 0 then the tri-state gates goes into high impedance state and thus inputs are not given to the full adder. The previous sum is taken as the present sum. If this bit is 1 then the tri-state gates get enabled and the inputs are given to the full adder. Thus the sum is generated and this is taken as the present sum.

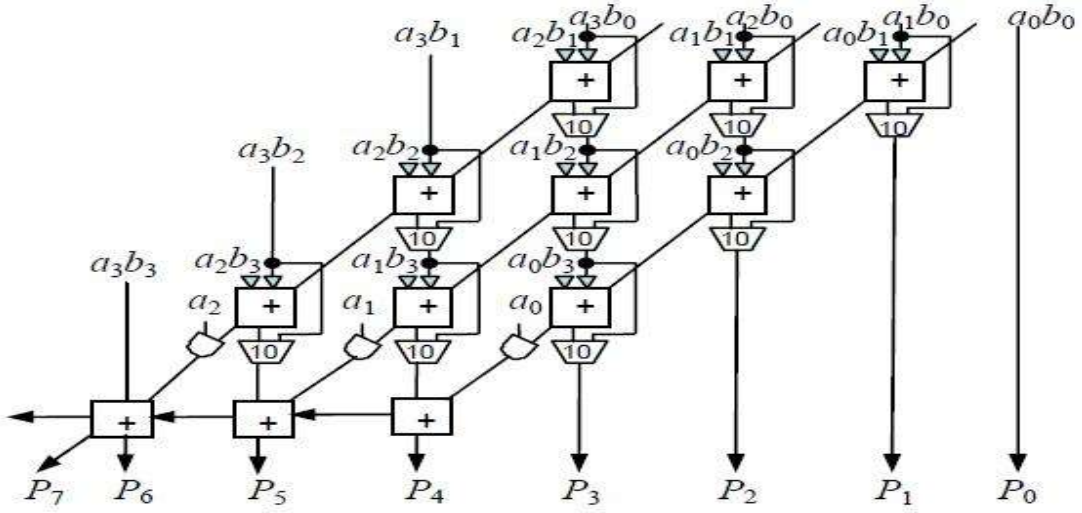

Fig-7: Column Bypassing Multiplier

Let us take the multiplication of 1001 x 1000. Since the multiplicand bit consists of zeros in the first and second position, so, corresponding columns will get disabled. As the multiplicand bit is 0, their inputs in the i^{th} column will be disabled and carry out in the column must be set to zero for producing the correct output. Hence, the process can be corrected by adding an AND gate at the output of the last row of CSA.

Carry save adder

Carry save adder [20] used to perform 3 bit addition at once. In carry save addition 3 input bit i.e. A, B, C is provided and in 1st stage we get 2 bit as an output. At initial stage result carry isn't propagated. So to generate carry, we've got to implement ripple carry adder at the last stage.

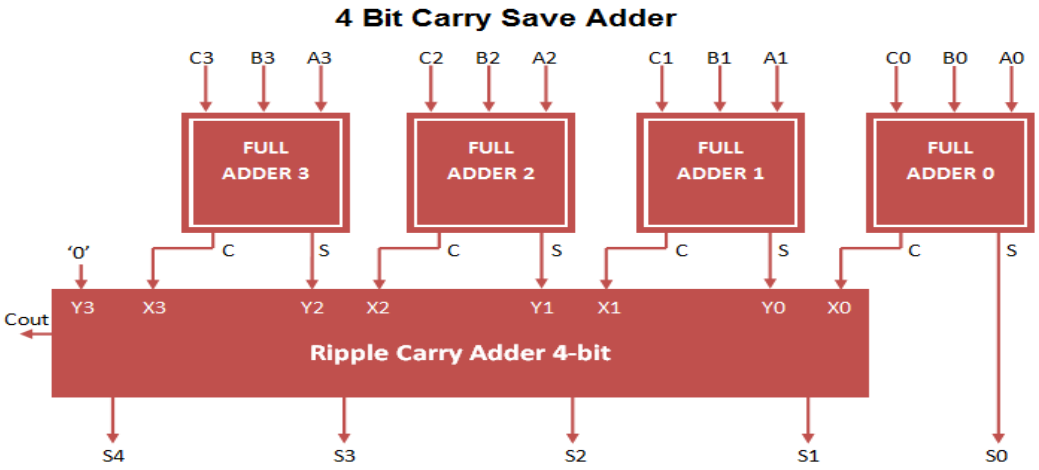

Fig-8: 4-bit Carry save adder

For instance: Let X, Y and Z be the information bit to a three-operand CSA, and let C and S be the carry and sum output bit vectors. Then it can be written as:

$$X+Y+Z=S+2C \quad (1)$$

In CSA, the main adder block consist carry save adder and ripple carry adder adders are used in the final row of adder.

Brent Kung adder

Parallel prefix adders [19] are unique class of adders that are based on the utilization of generate and propagate signals and Brent Kung adder is one of the parallel prefix adders. The parallel prefix adder involves three stages:

Pre-processing stage:

Generate and propagate signals are computed in pre processing stage and value of these signals is given by the following equations:

$$Pi=Ai \text{ xor } Bi \quad (2)$$
$$Gi=Ai \text{ and } Bi \quad (3)$$

Carry generation network:

We compute carries equivalent to each bit in carry generation network stage. These operations are carried out in parallel. Carry propagate and generate are utilized as an intermediate signals which are given by the logic Eq. 4& 5:

Carry generation and propagation

$$CPi:j=Pi:k+1 \text{ and } Pk:j \quad (4)$$
$$CGi:j=Gi:k+1 \text{or}(Pi:k+1 \text{and} Gk:j \quad (5)$$

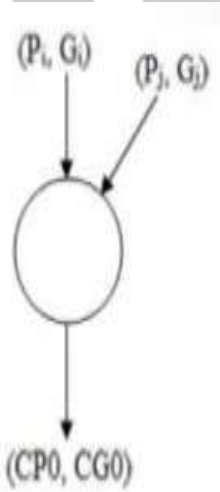

Fig-9: Carry network

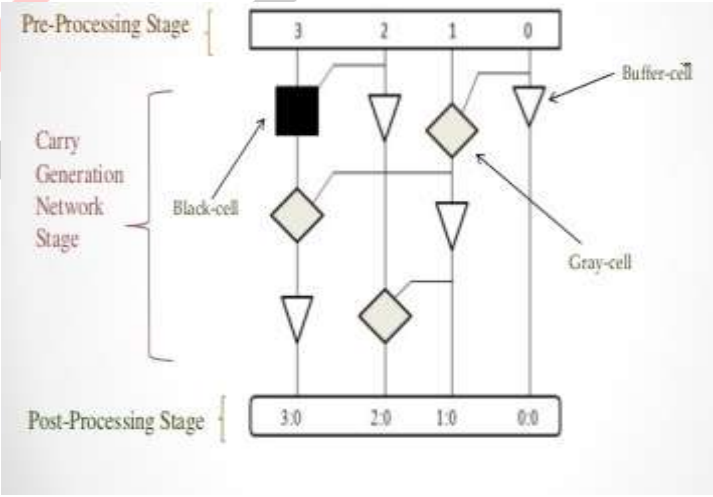

Fig-10: 4-bit Brent Kung adder

From fig.10

$$CP0=Pi \text{ and } Pj \quad (6)$$
$$CG0=Gi \text{ or } (Pi \text{ and } Gj) \quad (7)$$

Post processing Stage

This is the last step to compute the summation of input bits.

$$Cout=Gi \text{ or } (Pi \text{ and } Ci) \quad (8)$$

Si=Pi xor Cout (9)

Brent Kung adder computes the prefixes for 2 bit groups. These prefixes are utilized to discover the prefixes for the 4 bit groups, which in turn are utilized to discover the prefixes for 8 bit groups and so on. These prefixes are then utilized to compute the carry out of the particular bit stage. These carries will be used along with the Group Propagate of next stage to compute the Sum bit of that stage.

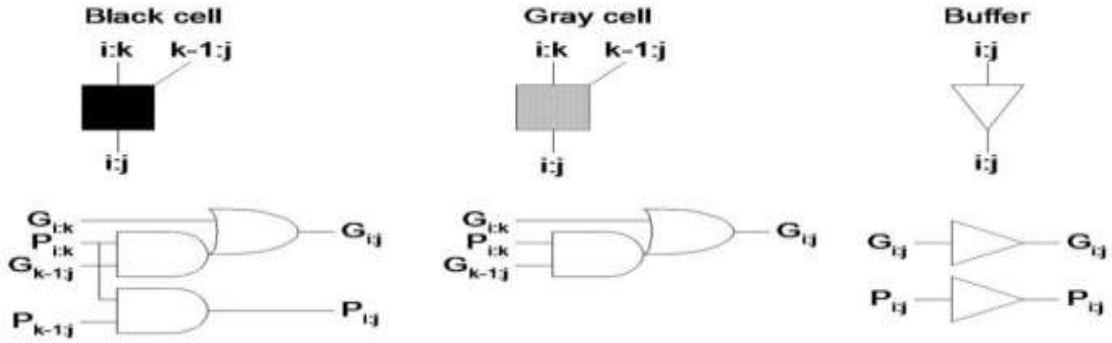

Fig-11: (i) Black cell (ii) Grey cell (iii) Buffer

Two pairs of generate and propagate signals (gi, pi) and (gj, pj) is taken as input in black cell. It results in a pair of generate and propagate signals (g, p) as output The gray cell takes two pairs of generate and propagate signals (Gi, Pi) and (Gj, Pj) as inputs and generate signal "G"

Adaptive hold logic

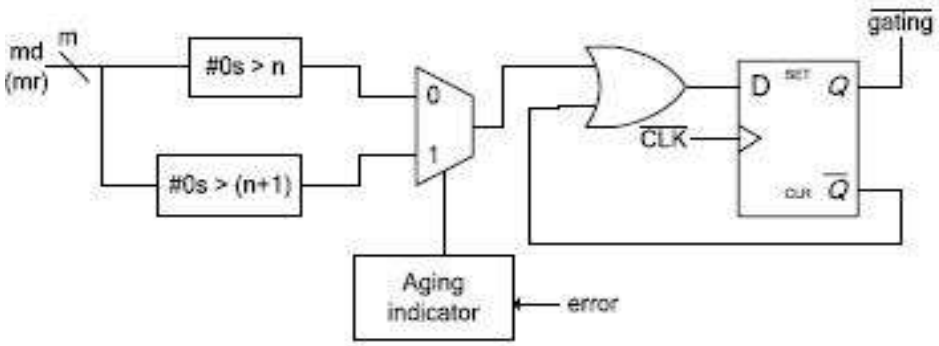

Fig-12: Adaptive hold logic circuit

Aging indicator:

Due to the aging mechanisms timing violation occurs and these timing violations will be caught by razor flip flop and the error is generated. Main function of aging indicator is to count errors over certain time of operation therefore it is implemented in form of counter that counts errors and at the end of operations this counter is reset to zero. The aging indicator in the AHL circuit point out whether the circuit has suffered notable performance degradation due to the aging mechanisms or not. The aging indicator will provide output 1 to the select line of the multiplexer if significant aging degradation has occurred otherwise, it will output 0 to indicate that no actions are needed the aging effect is still not significant.

Judging block:

Aging indicator will produces zero in the beginning as the aging effect is not crucial, so the first judging block is selected. If the number of zeros in the multiplicand (multiplier) is larger than n, the first judging block in the adaptive hold logic circuit will output 1, here n is defined by user, and if the number of zeros in the multiplicand (multiplier) is smaller than n it will output 0 . After some time when the aging effect becomes notable, the second judging block is chosen. Both of these blocks are utilized to decide whether an input pattern requires one or two cycles, but only one of them will be chosen at a time.

D flip flop:

The multiplexer chooses one of either result on the basis of the output of aging indicator. When output of multiplexer is 1 means pattern requires one cycle. The !(gating) will progress toward becoming 1, and the input flip flop will lock new information in the following cycle. At the point when the result of the multiplexer is 0, which implies the input information requires two cycles to finish it operation the OR gate will provide input 0 to the D flip-flop. Hence, the !(gating) will progress toward becoming 0 and disable the clock cycle of the input flip-flop in the next cycle. Just a cycle of the input flip-flop will be disabled because the D flip-flop will latch 1 in the next cycle

Razor flip-flop.

Razor flip-flop [20] is utilized to recognize whether timing infringement happen before the next input arrives. The main flip-flop gets the perform operation for the circuit utilizing an original clock cycle, and the shadow latch gets the perform operations

utilizing a delayed clock cycle. If the latched bit of the shadow latch is distinct from that of the main flip-flop, this implies the path delay of the present operation surpasses the cycle period, and the fundamental flip-flop gets an inaccurate outcome If timing violation happen, the Razor flip-flop will set the error signal and inform the system to re-execute the operation and tell the AHL circuit that a mistake has happened.

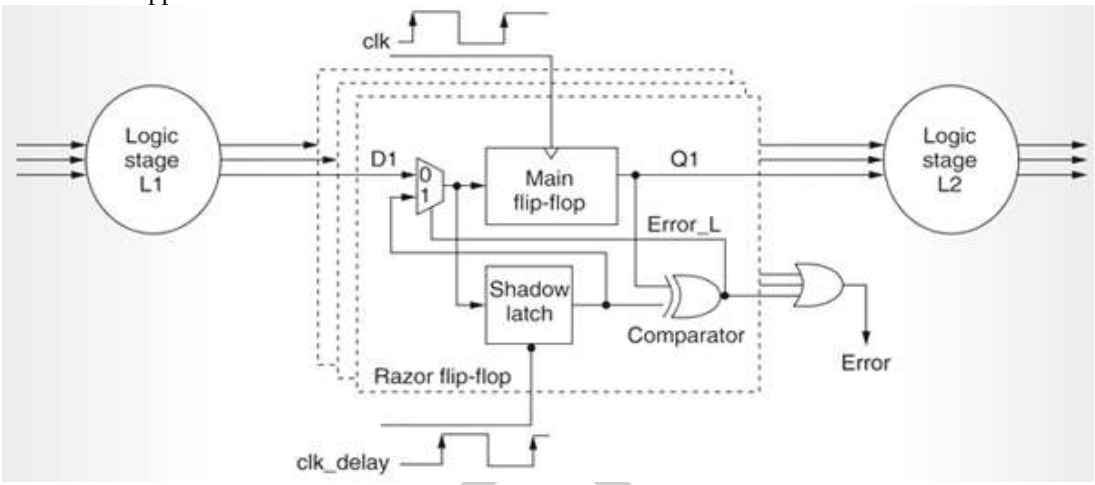

Fig-13: Razor flips flop

Razor flip-flop is used to recognize whether an operation that is believed to be a one-cycle can really complete an operation in a cycle. If not, the operation is re-executed with two cycles. Despite the way that the re-execution may seem, by all accounts, to be costly, the general cost is low in light of the fact that the re-execution frequency is low.

Working

The general flow of aging aware multiplier is as follows: when input pattern arrive, the row/column bypassing multiplier and the AHL circuit execute at the same time. As demonstrated by the number of zeros in the multiplicand (multiplier), the AHL circuit picks if the input pattern require perhaps one or more than one cycles. If the data input information requires two cycles to finish the operation, the AHL will provide output 0 to disable the clock cycle of the input flip-flop and if the data input pattern requires one cycle the AHL will provide 1 for general operations. Right when the row/column bypassing multiplier finishes the operation, the outcome of column/row bypassing multiplier will be passed to the Razor flip-flop The Razor flip-flops check whether there is the timing violations or not. In the case of timing violation happens, it implies the cycle period is not sufficiently long for the present operation to finish and that the execution consequence of the multiplier is erroneous. In the event that the, the Razor flip-flops will output an error implies that the present operation should be re-executed utilizing two cycles to guarantee the operation is right. In this situation, the additional re-execution cycles caused by timing violation acquires penalty to general average latency. Be that as it may, AHL circuit can precisely anticipate whether the information designs require one or more clock cycle to finish operation completely When AHL circuit judges incorrectly then only input patterns cause a timing violation therefore number errors will be less.

IV. RESULTS

The proposed aging aware multipliers are implemented using Xilinx ISE 14.7 tools. The results are simulated for different input vectors using Xilinx ISim Simulator. Synthesis report describes the actual hardware utilization, delay, area etc. Simulation results described the behavioral functionality of the design. The results shown below are for 4*4 and 16*16 Column/row bypassing multiplier with Carry Save Adder and Brent Kung Adder. Language used for coding is Verilog HDL.

Aging aware (4*4) multiplier

In Fig. 14 and 15 clk signal represents clock, a[3:0] represents multiplicand, b[3:0] represents multiplier which we are applying as inputs to the design . Here clock signal is generated for the positive edge. The output products get the multiplicity value of the applied inputs a and b and output in simulation waveform is represented by op. Here er waveform is generated by razor flip flop when timing violation occurs and aging indicator is counting the er here aging indicator is implemented as counter. When aging indicator will reach the predefined threshold the aging result signal will switch to high.

*Aging aware (4*4) row bypassing multiplier with CSA and BKA*

From Fig.14 also it can be seen when number of zeroes in the input b [3:0] is greater than n (defined by user here it is 2) the output time period is reduced. Here the number of zeroes depends on the error count. It is a row bypassing multiplier so when multiplier bit i.e. 'b' has number of zero more than the n then only output period will be reduced.

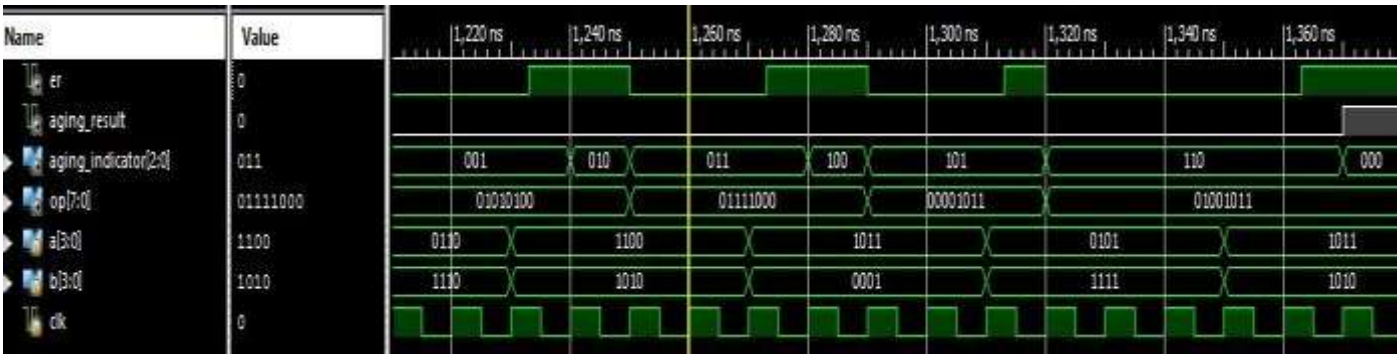

Fig-14: Simulation waveform of (4*4) row bypassing multiplier

*Aging aware (4*4) column bypassing multiplier CSA and BKA*

From Fig. 15 also it can be seen when number of zeroes in the input a [3:0] is greater than n (defined by user here it is 2) the output time period is reduced. Here the number of zeroes depends on the error count. It is a column bypassing multiplier so when multiplicand i.e. 'a' has number of zero more than the n then only output period will be reduced

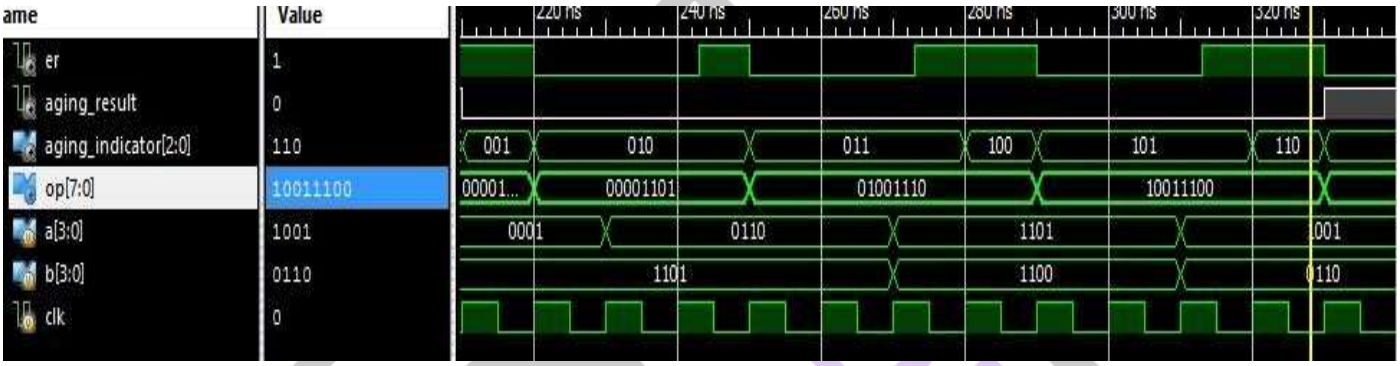

Fig-15: Simulation waveform aging aware (4*4) column bypassing multiplier

Performance analysis of aging aware (4*4) multipliers

Here performance analysis is done according to parameters like number of LUT, delay power, minimum period and maximum frequency. The delay of multiplier's with Brent Kung adder is less then that that of delay of multiplier's with carry save adder.

Table 1: Comparison of different parameters of (4*4) aging aware multiplier's

Parameters	Aging aware RBM (CSA)	Aging aware RBM (BKA)	Aging aware CBM (CSA)	Aging aware CBM (BKA)
Delay	6.061ns	5.813ns	5.568ns	5.436ns
Maximum frequency	124.626 MHz	132.888 MHz	141.335 MHz	146.801 MHz
Minimum Period	8.063 ns	7.565 ns	7.075 ns	6.812 ns
Area (Total LUT)	73	64	57	60

When we compare aging aware row bypassing multiplier and aging aware column bypassing multiplier the area, delay and minimum period of row bypassing multiplier is more than that of aging aware column bypassing multiplier this is due to additional equipment in row bypassing multiplier.

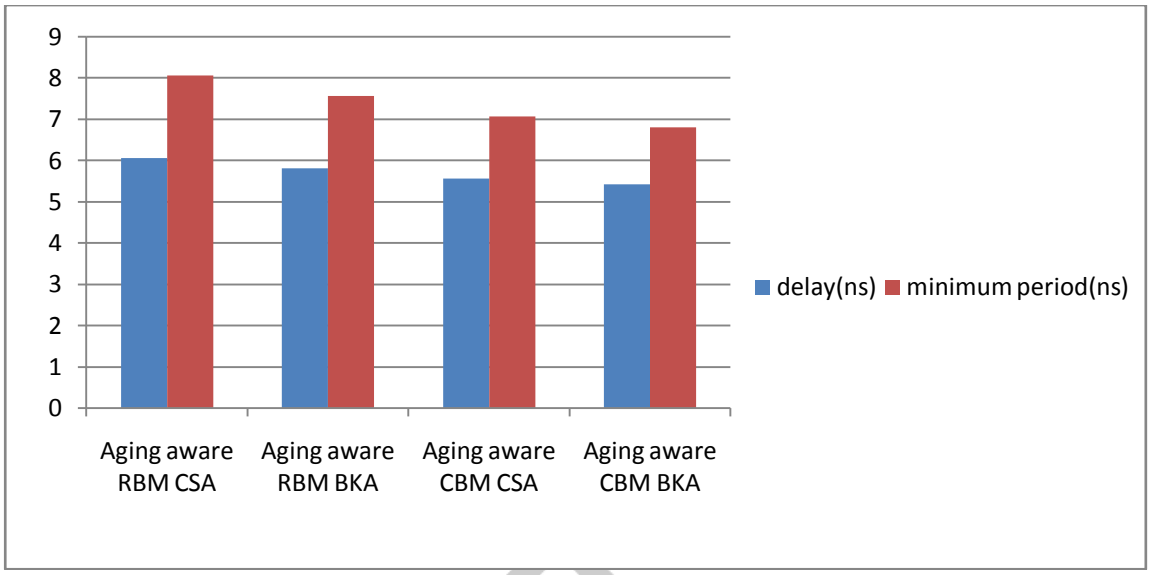

Fig-16: Comparison of delay and minimum period of (4*4) aging aware multiplier's

So from all this analysis we can say than the column by passing multiplier gives better performance than that of row bypassing and multiplier with Brent Kung adder gives better performance than multiplier with carry save adder. The best performance in all 4 types of multiplier is of aging aware column bypassing multiplier with BKA.

Aging aware 16*16 Multiplier

In the waveform which is shown in Fig. 17 and 18 signal represents clock, a[15:0] represents multiplicand, b[15:0] represents multiplier which we are applying as inputs to the design and. Here clock signal is generated for the positive edge. To obtain the required outputs force the inputs logic with the required values. The output products get the multiplicity value of the applied inputs a and b and output in simulation waveform is represented by op. Here er waveform is generated by razor flip flop when timing violation occurs and aging indicator is counting the er here aging indicator is implemented as counter. When aging indicator will reach the predefined threshold the aging result signal will switch to high as shown in Fig 17 and 18.

*Aging aware (16*16) row bypassing multipler with CSA and BKA*

From the simulation waveform also it can be seen when number of zeroes in the input b [15:0] is greater than n (defined by user here it is 8) the output time period is reduced. It is a row bypassing multiplier so when multiplier bit 'b' has number of zero more than the n then only output period will be reduced. Here the number of zeroes depends on the error count.

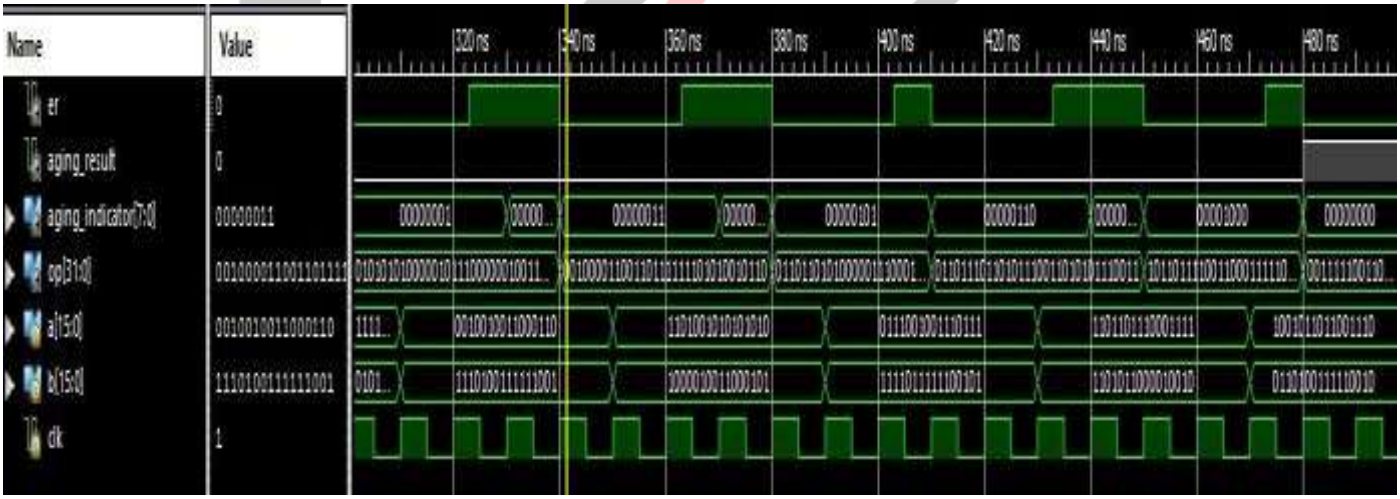

Fig-17: Simulation waveform of aging aware (16*16) row bypassing multipler

*Aging aware (16*16) column bypassing multipler with CSA and BKA*

From the simulation waveform also it can be seen when number of zeroes in input a[15:0] is greater than n (defined by user here it is 8) the output time period is reduced. It is a column bypassing multiplier so when multiplicand i.e. 'a' has number of zero more than the n then only output period will be reduced. Here the number of zeroes depends on the error count.

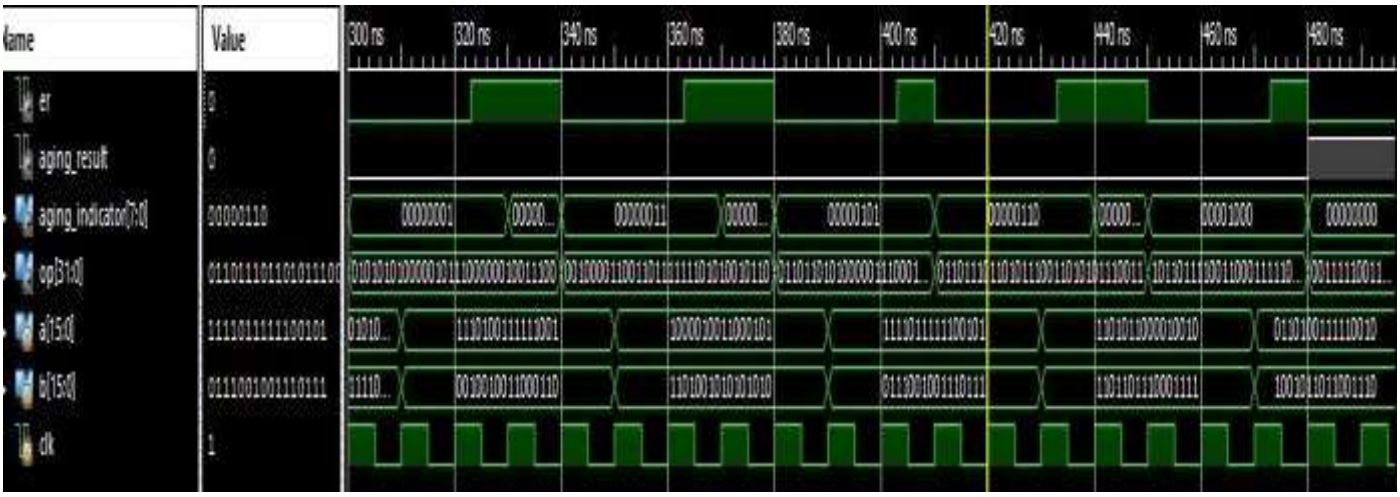

Figure 19: Simulation waveform aging aware (16*16) column bypass multipler

Performance analysis of aging aware (16*16) multipliers

In table 2 comparison of different aging aware (16*16) multiplier according to various parameters is done. Here performance analysis is done according to parameters like delay, area, power, minimum period, number of LUT. The delay and minimum period of multiplier's with Brent Kung adder is less then that that of delay and minimum period of multiplier's with carry save adder. Also the maximum frequency of multiplier's with Brent Kung adder is less than that of the multiplier with carry save adder

Table 2: Comparison of different parameters of (16*16) aging aware multiplier's

Parameters	Aging aware RBM (CSA)	Aging aware RBM (BKA)	Aging aware CBM (CSA)	Aging aware CBM (BKA)
Delay	22.801ns	21.632ns	22.251ns	17.447ns
Maximum Frequency	28.050 MHZ	30.900 MHZ	29.761 MHZ	41.679 MHZ
Minimum Period	34.716 ns	32.263 ns	33.601 ns	23.993 ns
Area (Total LUT)	1179	1360	709	772

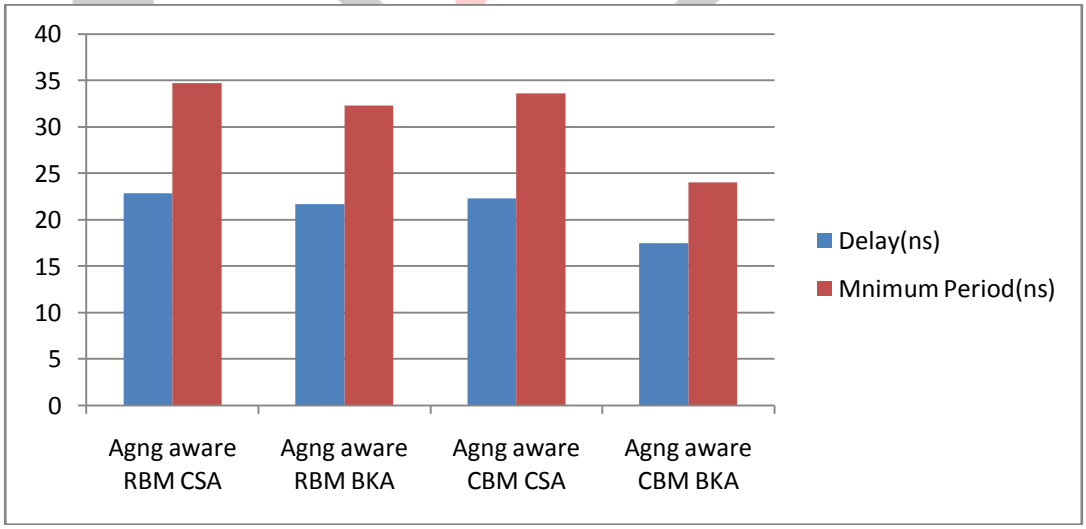

Figure 20: Comparison of delay and minimum period of (16*16) aging aware multiplier's

When we compare aging aware row bypassing multiplier and aging aware column bypassing multiplier the area, delay and minimum period of row bypassing multiplier is more than that of aging aware column bypassing multiplier. So from all this analysis we can say than the column by passing multiplier gives better performance than that of row bypassing and multiplier

with Brent Kung adder gives better performance than multiplier with carry save adder. Experimental results show the best performance in all 4 types of multiplier is of aging aware column bypassing multiplier with BKA.

V. CONCLUSION

Performance analysis of an aging aware multiplier with different adder is done. The multiplier is designed with adaptive hold circuit mitigate performance degradation due to increased delay. An aging aware multiplier has less performance degradation because it is based on variable latency technique which diminishes the timing wastage. Therefore aging aware multipliers have less timing waste, but conventional multipliers need to consider the degradation caused by aging mechanisms and use the critical path delay as the cycle period. An aging-aware reliable multiplier where column/row bypassing multiplier is designed with carry save adder and another aging aware reliable multiplier where column/row bypassing multiplier is designed with Brent Kung Adder to mitigate the aging induced degradation of multiplier further. The experimental results show proposed architecture with 4*4 and 16* 16 multiplications with Brent Kung adder decrease the delay, decreases minimum period and work on high frequency in comparison to the aging aware multiplier with carry save adder. From the performance comparison tables it can conclude that Delay of the circuit with Carry Save Adder is the higher, while that with Brent-Kung Adder is the lower. Hence, it can be concluded that the Brent Kung Adder is a better option for Aging aware Reliable Multiplier.

VI. FUTURE SCOPE

In future we can implement a multiplier which can perform both row and column bypassing technique in a single architecture We can design another full adder cell to reduce the delay further. In future the same technique can be applied to 32, 64,128 bit multipliers and adder

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 VLSI Technology Digest of Technical Paper, 2006, pp.23-25.
[2]S. Zafar, A. Kumar, E. Gusev, and E.Cartier, "Threshold voltage instabilities in high-k gate dielectric stacks" in Device Mater. Rel., 2005, pp. 45-64.
[3] MOORE, G. E. “Cramming More Components onto Integrated Circuit”. Electronics 38, pp 114–117, Apr. 1965.
[4] Y. H. Cho, I. C. Lin and Y. M. Yang, "Aging-aware reliable multiplier design," IEEE International SOC Conference, Niagara Falls, NY, 2012, pp. 322-327.
[5]R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design" in DAC, 2006, pp. 1047-1052
[6] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, "NBTI-Aware Synthesis of Digital Circuits," in DAC, 2007, pp. 370-375
[7] Andrea Calimera, Enrico Macii, Massimo Poncino “NBTI-Aware Sleep Transistor Design for Reliable Power-Gating”, Research Gate Conference Paper, January 2009.
[8] K.-C.Wu and D. Marculescu, "Joint logic restructuring and pin reordering against NBTI-induced performance degradation" IEEE, in 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 ASP-DAC, 2011, pp. 603-608
[10]K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization" in DATE, 2011, pp. 1-6.
[11] Venudhara , Santhosh Kumar “ Design of Efficient Aging-Aware Predictable Multiplier Using Adaptive Hold Logic” International Journal of Innovative Research in Electrical, Electronics, Vol. 4, Issue 8, August 2016 .
[12] Yu-Shih Su, Da-Chung Wang, Shih-Chieh Chang, Member, IEEE, and Malgorzata Marek-Sadowska, “Performance Optimization Using Variable-Latency Design Style”, IEEE transactions on very large scale integration (VLSI) systems, Volume 19, Issue 10, Oct. 2011, pp. 1874
[13] Supriya Juluganti, T. Muni Reddy “Realization of Aging-Aware Reliable Multiplier Design using Verilog HDL” International Journal of VLSI System Design And Communication System, Vol.03, Issue.04, July-2015, Pages:0403-0408.
[14]Bao Liu, Lu Wang and Juan Portillo “Variable latency VLSI design based on timing analysis, delay ATPG, and completion prediction” ECE Department, University of Texas, San Antonio, TX, 78249-0669.
[15] Shubhangi Ramannawar, Deepak Kumar “Efficient Multiplier Design Using Modified Booth Algorithm and Razor Flip-Flop” International Journal of Science, Engineering and Technology Research (IJSETR), Volume 6, Issue 1, January 2017.
[16] Yiran Chen, Hai Li, Jing Li , and Cheng-Kok Koh, "Variable-latency Adder (VL-Adder) : New Arithmetic Circuit Design Practice to Overcome NBTI" in ISLPED, 2007, pp. 195-200
[17] Ming-Chen Wen, Sying-Jyan Wang, and Yen-Nan Lin, "Low Power Parallel Multiplier with Column Bypassing" in ISCAS, 2005, pp.1638-1641.
[18] A. Krishnan, C. Chancellor, S. Chakravarthi, P. Nicollian, V. Reddy, A. Varghese, R. Khamankar, and S. Krishnan, “ Material dependence of hydrogen diffusion: implications for NBTI degradation”, Electron Devices Meeting, IEDM Technical Digest. IEEE International, Dec 2005.
[19] Pallavi Saxena, “Design of Low Power and High Speed Carry select adder using Brent Kung adder” International conference on VLSI systems Architecture, Technology and Applications, IEEE 2015.
[20] R. P. Brent and H. T. Kung, "Regular layout for parallel adders," IEEE Transactions on Computers, Mar 1982, vol. C-31, pp. 260-264