

# Reliable High Performance Multiplier with Adaptive Hold Logic for Aging Awareness

P. Shreya\* and R. Saravanan

School of Computing, Sastra University, Thanjavur – 613401, Tamil Nadu, India;  
Coolshreya1893@gmail.com, saravanan\_r@ict.sastra.edu

## Abstract

**Background:** 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 aspect with respect to the speed and power consumption. **Statistical analysis:** Multipliers are the key functional units in various applications. The transistor speed is reduced due to bias temperature instability effects. When this effect extends in distant future, the system may not work properly due to timing infraction. **Findings:** 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 ( $V_{gs} = -V_{dd}$ ), 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 decrease the transistor speed, in the long run, the system may fail due to timing violations. Usually consummation of a system depends on the throughput of the multiplier. So it is important to adopt a reliable high performance multiplier. The multiplier is able to attenuate the performance degradation due to aging. **Improvement:** Hence, the objective of the project is to design a high speed and power efficient multiplier to increase the performance of the device. Design of Multiplier circuit using Adaptive Hold Technique is proposed. By using AHT circuit we can reduce the NBTI and Positive Bias Temperature Instability (PBTI) effects, hence performance will be increased and the aging effects will be reduced. The proposed technique is done using Xilinx 14.7 tool. The code is simulated and synthesized and comparison results are made based on the performance of the multipliers with and without AHT.

**Keywords:** Aging, Attenuate, Hold Logic, Temperature Instability, Throughput

## 1. Introduction

Multipliers are the most pivotal units in logical and operative calculation. If the throughput is increased, then the conduct is also increased. If the conduct of the multiplier gradually reduces then the conduct of the entire circuits will be diminished. So it is important to design a reliable multiplier.

Pmos transistor is subjected to NBTI effects. It occurs when  $V_{GS} = -V_{DD}$  It is a result of the interaction between hydrogen and si atom breaks during oxidation, it will generate H or H<sub>2</sub> molecules. This will create compound pitfall and girth of gate oxide. Even though the bias voltage is removed it cannot phase out all the traps. This will expand the threshold voltage ( $v_{th}$ ), clipping the switching speed of the transistor.

This effect also occurs in Nmos transistor. When Nmos is under positive bias it gives rise to PBTI. BTI effects also includes the amount of time in which the transistor is stressed. PBTI effect is much minor and can be ignored. Whereas for high k-metal gate processes, PBTI effect shows more charge trapping so it cannot be blinked. A theory about BTI shows that threshold voltage shifts with the stressing time and increase in  $V_{th}$  is an important reliability factor. Device aging affects circuit performance and life time. It gives rise to reliability degradation.

To avoid this aging aware methods were proposed like guard banding gate oversizing. This will be power inefficient. Timing analysis methods was proposed to reduce aging induced performance degradation. But transistor resizing will increase power. Any way these techniques will not provide any optimisation to certain circuits.

\* Author for correspondence

Moreover, traditional circuits will use longest path delay as the cycle period. The probability that the longest path is used is very low. So timing waste in the circuit becomes large. To avoid this variable latency technique is used. This will reduce the significant timing waste in non-critical path. The idea behind variable latency is that it divides the circuit into two paths namely precise and deeper paths.

## 2. Variable Latency Design

The variable latency design reduces the timing waste in non-critical paths. The fixed latency utilizes longest path delay as the complete cycle period. But the probability that the critical path is triggered is very low.

Whereas, the VLD divides the cycle in two path 1. precise path 2. deeper path. A precise path is executed in single cycle and a deep path is executed in two cycles<sup>1</sup>.

The variable latency analytical addition proposed persistent adders<sup>2</sup> using variable latency which is faster than the conventional adders. Here a fast unreliable adder produces roughly correct result for various input combination. The same adder is implemented in VLSA with error detection. If an error is detected it will produce correct result after few cycles. This delay will be much shorter than the unreliable adder.

Design of pipelined multipliers<sup>2,3</sup> showed the performance of variable latency will be more effective than fixed latency multipliers. Variable Latency Carry Select Addition proposed a Speculative Carry Select Addition. But the design gives rise to aging.

Hold logic is applied in non-tree structures<sup>4</sup> which can result in good performance boost. But the path delay is not considered.

## 3. Bypassing Technique

Different designs of multipliers<sup>5</sup> have been illustrated for low power in early years. The bypassing techniques is used to reduce the vigorous power consumption. Either the column or row of adders is bypassed. The resulting output depends upon the input bit coefficient value say 0 or 1. So it is clear that we use muxes for selecting output from the adders.

A normal array multiplier<sup>6</sup> shown in Figure 1 consists of full adders. The sum bit is obtained for each full adder and the carry bit goes to the left. Finally, using RCA the

carry bit is propagated and the resulting bit of sum and carry is obtained. The bypassing multipliers are generally a modification of array multipliers. This reduces the vigorous power consumption. The path delay is strongly trusted to the value of the either the multiplicand or multiplier.

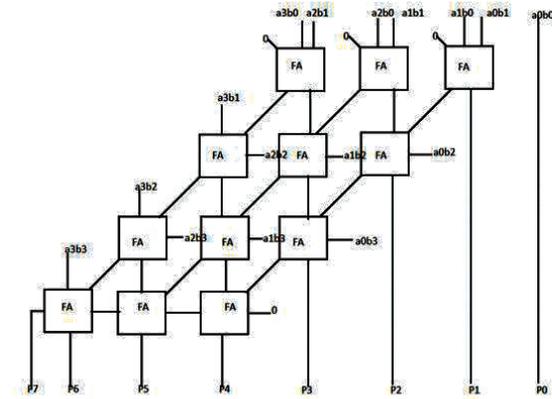


Figure 1. Array multiplier.

### 3.1 Column Bypassing Multiplier

A single unit is shown in Figure 2. A column bypassing multiplier, shown in Figure 3 consists of rows of full adder. Multiplexers are used in each column of the adder<sup>7,8</sup>. The multiplicand bit is the selection line. Example if the input bit,  $a0b1=0$  and  $a1b0=1$ , the selection line is multiplicand bit  $a0=0$  then the addition operation will be omitted. The sum bit grounds and the carry operation goes to the left side. In case if the selection line is  $a0=1$  then an addition operation will be done by the full adder. The resulting sum bit grounds and carry is passed to the left. Hence it clips power and the path delay depends on the input bit value.

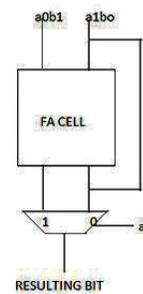


Figure 2. Column Bypass Single unit.

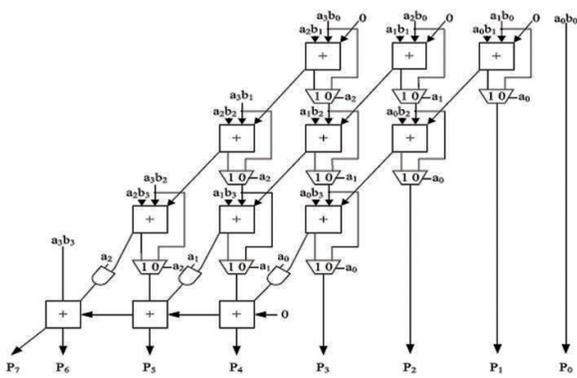


Figure 3. Column bypassing multiplier.

### 3.2 Row Bypassing Multiplier

Row bypassing multipliers shown in Figures 4 and 5, are similar to the column bypassing multipliers. When the result from the sectional product is zero, the adder operation is omitted and bypassed the inputs to outputs. In this designed each full adder is extended with two multiplexers and three tristate buffers. As the result from the right fullest adders are bypassed extra make over circuits must be added to get the proper multiplication result. Thus the power consumption is diminished by turning down the switching movement This is achieved by disarming the operation of adders by bypassing which will cease the abortive part of the circuit when not in use.

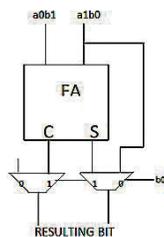


Figure 4. Row bypass single unit.

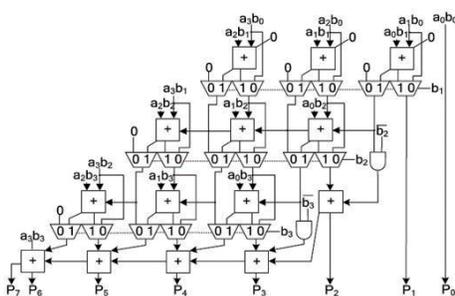


Figure 5. Row bypassing multiplier.

## 4. Aging Model

When the PMOS (NMOS) is under negative (PMOS) bias results in negative (positive) temperature bias instability. If uniform force is observed in the transistors, it is referred as static BTI. If both force and restoration phase occurs, then it is called dynamic BTI. The increase in threshold voltage creates aging effect. In this paper we propose 16\*16 column bypassing and row bypassing multipliers with efficient hold logic to reduce the aging effect. The multiplier is capable to adjust the execution derogation with adaptive hold logic and variable latency. The block diagram is shown in Figure 6.

## 5. Proposed Technique

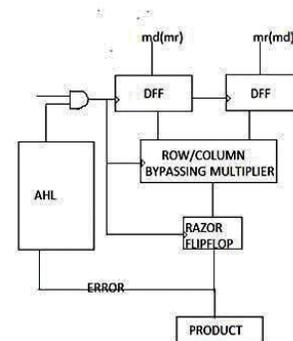


Figure 6. Block diagram.

### 5.1 Razor Flipflop

The razor flip flop in Figure 7 consists of a d flipflop and a shadow latch. The d flipflop works in normal clock operation and the shadow latch is given a clock with a delay compared to the normal clock. If the resulting bit from the d flipflop differs from the shadow latch, then there exists a timing transgression. This error signal is obtained by a comparator and the error from the razor flip flop is advised to AHL so that the error is corrected in a few cycles.

### 5.2 Adaptive Hold Logic

The AHL shown in Figure 8, consists of a mux, D flip flop and aging indicator<sup>9</sup>. When the input pattern arrives the AHL ensures whether the operation can be performed in

one or two cycle. If the output from the multiplexer is 1, then no error has occurred and the flipflop will latch new data in next cycle. If the output from the multiplexer is 0, then an error is caught from the d flipflop. Then the or operation is performed is between the complement of q signal and the result from the mux. The clock signal will be gated disabling the operation for next cycle. The aging indicator<sup>9</sup> is used to count the number of errors in a particular cycle.

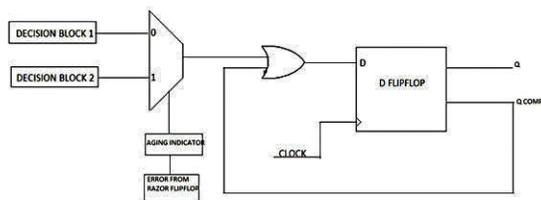


Figure 8. Adaptive hold logic.

The overall procedure of the architecture, in Figure 6 is as follows:

First the AHL and the multipliers start working simultaneously. The decision block is the input bit coefficient from the multipliers. The decision block<sup>9</sup> counts the number of zeroes whether the operation can be executed in one or two cycles. The multiplier passes the result to the razor flip flop. The razor flip flop<sup>10</sup> detects the path delay violation and return the error to AHL.

The AHL will gate the next input pattern. Now the error will be corrected just after a few cycles. Thus the proper product is obtained. Mostly the AHL will properly ensure the one or two cycle operations.

## 6. Results and Discussion

The Verilog code is simulated and synthesized using Xilinx 14.7 tool. 16\*16 row (Figure 9) and column architecture is designed. Then AHL circuit is implemented to the multipliers. The decision blocks are used to identify the sum of 0's in the multiplicand. The razor flipflop is used for error disclosure in Figure 10 and the final result is shown in Figure 11. The results are synthesized using Xilinx. Finally, the comparison.

Table 1 shows that the delay is reduced while implemented with ahl. Performance improvement is achieved more in row bypassing multiplier.

Table 1. Delay comparison

MULTIPLIER(16*16)	Without AHL	With AHL
ARRAY	19.54	16.25
COLUMN	15.35	12.08
ROW	13.95	9.89

## 7. Conclusion

The proposed architecture is applied to 16\*16 array/row/

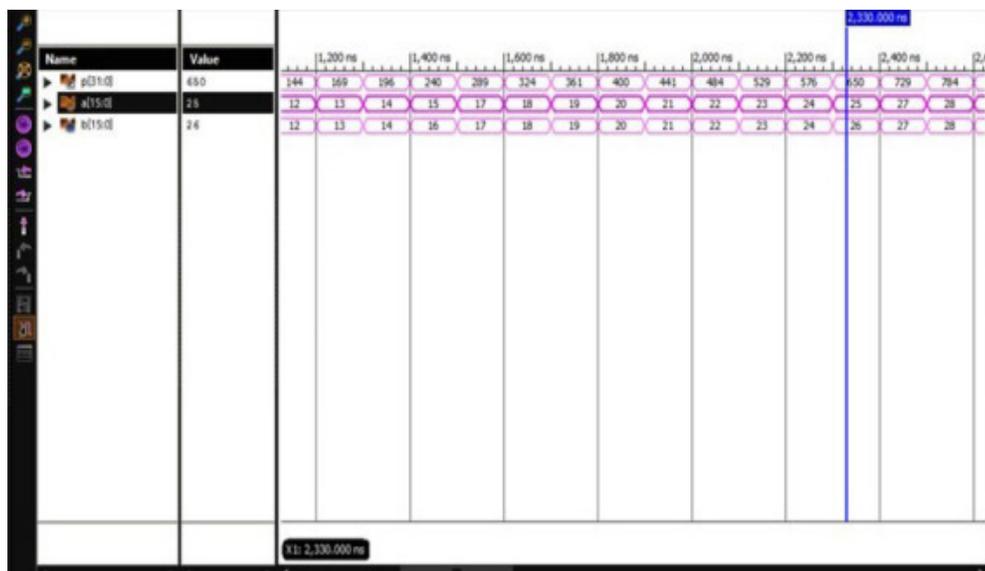


Figure 9. Row bypassing multiplier (16\*16).



Figure 10. Razor flipflop\_32.



Figure 11. Final result.

column bypassing multiplier. Aging effect is reduced using AHL circuit. Performance improvement is achieved due to the latency design. Overall delay is reduced in the multiplier design caused due to aging. In future the same technique can be applied to 32,64,128 bit multipliers and adders.

## 8. References

1. Zafar S, Kim YH, Narayanan V, Cabral C, Paruchuri V, Doris B, Stathis J, Callegari A, Chudzik M. A comparative study of NBTI and PBTI (charge trapping) in SiO<sub>2</sub>/HfO<sub>2</sub> Stacks with FUSI, TiN, Re Gates. 2006 Symposium on VLSI Technology, 2006. Digest of Technical Papers; 2006. p. 23–5.
2. Kai-Chiang Wu, Marculescu D. Aging-aware timing analysis and optimization considering path sensitization. Design, Automation and Test in Europe Conference and Exhibition (DATE); 2011 Mar. p. 1–6 .
3. Ajay K, Brisk P, Jenne P. Variable latency speculative addition: A new paradigm for arithmetic circuit design. Design, Automation and Test in Europe Conference and Exhibition (DATE 08), Munich: Germany; 2008 Mar. p.1250–55.
4. Olivieri M. Design of synchronous and asynchronous vari-

- able-latency pipelined multipliers. *IEEE Transactions on Very Large Scale Integration (VLSI) systems*. 2001 Apr; 9(2):365–76.
5. Praveena R, Nirmala S. Realization of efficient multiplier for low power biomedical signal processing system-on-chip design for portable ECG monitoring systems. *Indian Journal of Science and Technology*. 2015 Sep; 8(24):1–7. DOI: 10.17485/ijst/2015/v8i24/80211.
  6. Khan PAI, Mishra RS. Comparative analysis of different algorithm for design of high-speed Multiplier Accumulator Unit (MAC). *Indian Journal of Science and Technology*. 2016 Feb; 9(8):1–5. DOI: 10.17485/ijst/2016/v9i8/83614.
  7. Yan J-T, Chen Z-W. Low-power multiplier design with row and column bypassing. *SOC Conference, IEEE International*; 2009 Sep. p. 227–30.
  8. Rani JS, Ramadevi D, Kumar BS, Reddy KJ. Design of low power column bypass Multiplier using FPGA. *IOSR Journal of VLSI and Signal Processing*. 2011 Apr; 3:431–35.
  9. Lin I-C, Cho Y-H, Yang Y-M. Aging-aware reliable multiplier design with adaptive hold logic. *Transactions on Very Large Scale Integration (VLSI) systems*. 2015 Mar; 23(3):544–56.
  10. Erns D. Razor: circuit-level correction of timing errors for low-power operation. *IEEE Micro*. 2004 Nov-Dec; 24(6):7–18.