Realization of Aging Aware Multiplier using Verilog HDL

[1]Kishore T, [2]Jayamani R, [3]Jaivignesh R, [4]Ajin R
[1]Student, [2] Student, [3] Student, [4]Assistant Professor
[1,2,3,4] Department of Electronics and Communication Engineering,
[1,2,3,4] Prathyusha Engineering College, Tiruvallur, India

Abstract : Most of the digital devices use multipliers for their operation especially in digital signal applications. It helps to achieve a high data throughput in digital devices. Comparing to addition, multiplication process consumes a greater deal of time, consuming more amount of power and area and thus reduces the speed of the processor. Aging of transistors has a considerable effect on the performance of the multiplier. The effect of aging can be reduced by the use of over-design approaches, but it leads to inefficiency in area and power. Furthermore, the use of fixed latency design may lead to timing violations. To overcome this predicament, we use low power variable latency multiplier with Adaptive Hold Logic (AHL). Negative bias and positive bias temperature instability, both degrade the transistor speed, and the system may fail owing to timing violations. Hence, it becomes more important to design highly reliable multipliers with reduced area, delay and power consumption. The proposed architecture will be designed as a razor based Vedic multiplier design with novel adaptive hold logic (AHL) circuit which as high efficiency than the existing column bypass multiplier. The experimental results show that our proposed architecture with 32 x 32 aging aware multipliers is more efficient and has lesser latency.

Index Terms **- Aging Aware, Adaptive hold logic, Latency, Bypassing, Reliable.**

I. INTRODUCTION

Digital Multipliers form the basic functional unit of any processors and is widely used arithmetic unit for variety of applications such as discrete cosine transform, Fourier transform and for digital filtering purposes. The simplest among all multipliers is the Array multiplier and it is also the slowest among all multipliers [1]. There are also other high speed multipliers which includes Booth multiplier, Wallace tree multiplier, etc. [2]. The process of multiplication involves a considerable amount of time as it includes a number of steps in the computation process. The throughput of multipliers can be improved by reducing these steps so as to achieve a speed in multiplication process [3]. In order to maximize the speed of multiplication, we use Vedic algorithm [4] in our multiplication process which is a faster way of oral multiplication.

Among the various Vedic multiplication techniques, the Urdhava Tiryakbhyam sutra is considered to be more efficient in terms of speed [5]. Bypassing in multipliers is a methodology to reduce the number of unwanted computations to be performed when the multiplicand (or multiplier) bit is zero. This reduces the power and the delay caused but to these computations.

II. LITERATURE SURVEY

Multipliers are designed to be more reliable by adopting many ways. In [6], variable latency pipelined multiplier using booth algorithm is proposed. This paper proposes an implementation as a self-timed multiplier core or as a multicycle multiplier core. The proposed architecture combines a split carry save array pipelined organization with 2nd order Booth algorithm and the design is proposed for 32 bit synchronous implementation. A 64-bit carry select adder is proposed in [7] using variable latency (VL) adder technique which required a lower supply voltage than consumed by a conventional adder. The effect of negative bias temperature instability and circuit delay can be overcomed by the modification of the VL adder design. An aging aware multiplier was proposed in [8] where Array multiplier is used along with column/row bypassing to achieve a greater throughput. Also the temperature bias instability is analyzed and the design is able to resist these instabilities.

A low power multiplier design using row and column bypassing was proposed in [9] which showed that column bypassing and 2-D bypass multipliers were more efficient than row bypassing multipliers. The multiplier is designed using simplified adders and half adders instead of full adders. In [10], a Braun multiplier which is a parallel array multiplier is designed using bypassing techniques both 1-Dimensional and 2-Dimensional bypassing. This paper shows a comparative study on the basis of area, power and delay of different bypassing multipliers for various bits using Spartan – 3E FPGA. In [11], it reveals the Vedic algorithm, which is a faster way of oral calculation and for multiplication and product finding through the end results of normal multiplication. Also it portrays the current usage of Vedic algorithms and its importance in the design of computer processors to improve its speed and performance.

III. PROPOSED AGING AWARE MULTIPLIER

To overcome the effect of aging in multipliers, an aging aware multiplier using Adaptive Hold Logic (AHL) is proposed. Fig. 1 shows the architecture of the proposed aging aware multiplier which consists of an n-bit multiplicand and an n-bit multiplier as inputs, 2n-bit output, AHL circuit, column/row bypassing multiplier and 2n 1-bit Razor flip-flop.

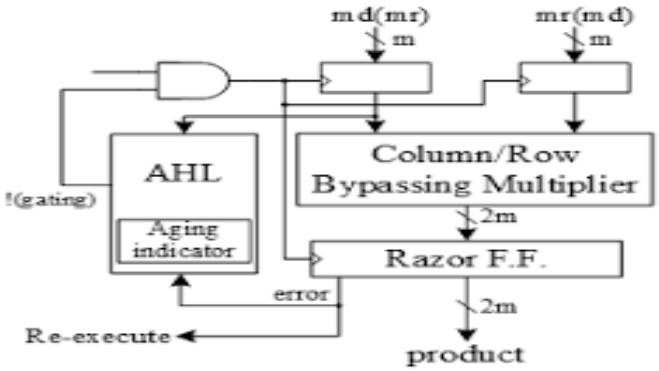

Fig.1. Aging Aware Multiplier

The variables (inputs) in the parenthesis are the inputs of row-bypassing multiplier and those outside are of column-bypassing multiplier. The number of zeros in the multiplicand or the multiplier are examined to find out whether the operation requires one or two clock cycles to complete. The input patterns are random and the number of ones and zeros in both the multiplicand and the multiplier takes a normal distribution.

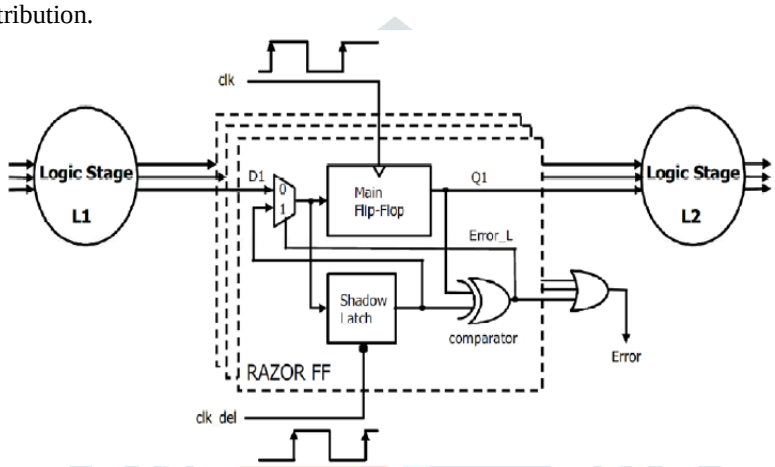

Fig.2. Razor Flip-flop

Razor flip-flops are used to detect the timing violations [12] that may occur after or prior to the input pattern activities. Fig. 2 shows the Razor flip-flop. It consists of a main flip-flop, shadow latch, multiplexer and XOR gate. The main flip-flop takes the result of execution of the combinational circuit by means of a normal clock signal whereas the shadow latch caches the result with a delayed clock signal which is slower than the clock signal used for catching the result of combinational circuit. If the latched bit of shadow latch varies from that of the main flip-flop, the path delay of the present process surpasses the cycle period and the main flip-flip will catch an improper result.

The Razor flip-flop sets the error signal as 1 on the event of occurrence of an error to notify the AHL that an error has occurred and re execution occurs. Razor flip-flops are used to sense whether the operation requires one complete its execution or requires two clock cycles for re execution. Though the re execution may seem to be costly, the overall cost is low due to only low frequency is required during re execution.

Adaptive Hold Logic

The adaptive hold logic (AHL) is the most significant component in the design of aging aware multiplier. Fig. 3 shows the block diagram of AHL circuit. It consists of two judging blocks, D flip-flop, mux, an OR gate and an aging indicator. The aging indicator is used to detect whether the circuit has undergone performance degradation due the effect of aging.

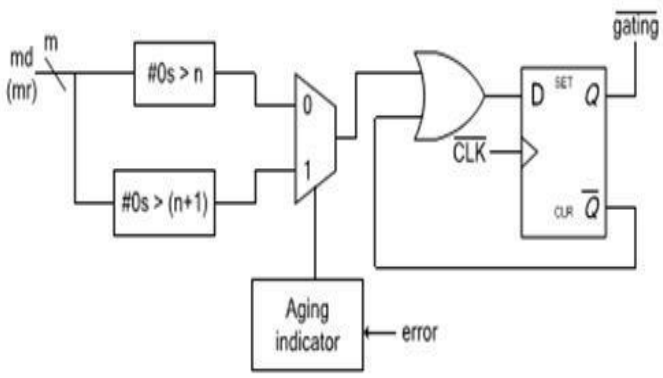

Fig.3. Block Diagram of AHL

The aging indicator is realized using a simple counter which counts the number of errors occurring over a certain number of operations and is reset to zero at the completion of those operations. If the cycle period is too small, the column or row bypassing multiplier will not be able to successfully complete the operations and will lead to timing violations. These are caught by the Razor flip-flops and will generate error signals. The aging indicator will generate the output signal as 1 if the error occurs frequently and exceeds the threshold level and denotes that the circuit has undergone a noteworthy amount of timing degradation due to aging effect. If the aging effect is not that much significant, then the output of the aging indicator is 0, implying it requires no action.

In column bypassing technique, if multiplicand bit is zero, then the corresponding FA operations are disabled. Similar conditions prevails to row by passing multiplier except that multiplicand is replaced with multiplier. For column bypassing multiplier, the multiplicand acts as the input signal for bypass selection whereas for row bypassing, it is the multiplier. Bypassing eliminates the number of computations performed when the multiplicand (multiplier in the case of row bypassing) bit is zero which in turn reduces the power required and the delay it produces.

Working:

The AHL block consists of two judging blocks which produces the output based on the input patterns that arrive. If the number of zeros contained in the multiplicand (multiplier) is greater than n, the judging block 1 will produce an output 1. The output will be 1 in the second judging block when the number of zeros in the multiplicand (multiplier) in the input pattern is greater than n+1. Any one of the judging blocks will be executed at any point of time. Based on the output from these judging blocks, the circuits will decide whether the operation requires one or two clock cycle to execute the operation. Both these results are passed to the multiplexer which selects any one of the judging block output based on the output of the aging indicator. At the initial stage, the output of the aging indicator is zero as there is no significant aging effect and the first judging block is selected by the multiplexer. The aging effect becomes significant over a period of time and the second judging block is chosen for execution.

The output of the multiplexer is fed to an OR gate along with the Q' signal which would determine the input to the D flip-flop. The output of AHL is 1 for normal executions and the output will be 0 when the input pattern requires two cycles to complete its operations to disable the clock signal of the flip-flops. When the output of the multiplexer is 1, the !(gating) signal is 1 and a new data is latched to the input flip-flop in the next cycle. Furthermore, when the multiplexer outputs zero, the !(gating) signal is 0 which would disable the clock signal to the input flip-flop in the next cycle. When the multiplier completes its operation, its result is passed to the Razor flip-flop which is check the violations in path delay timings. If path delay violations occurs, it signifies that the output of the multiplier is incorrect as the cycle period is not long enough. So, the Razor flip-flop will send an error signal to the AHL stating that two cycles are required for re-execution of the current operation to ensure higher accuracy in the results of the multiplier.

IV. SIMULATION RESULTS

Fig. 4 shows the module diagram of 64-bit Aging Aware Vedic Multiplier with Adaptive Hold Logic.

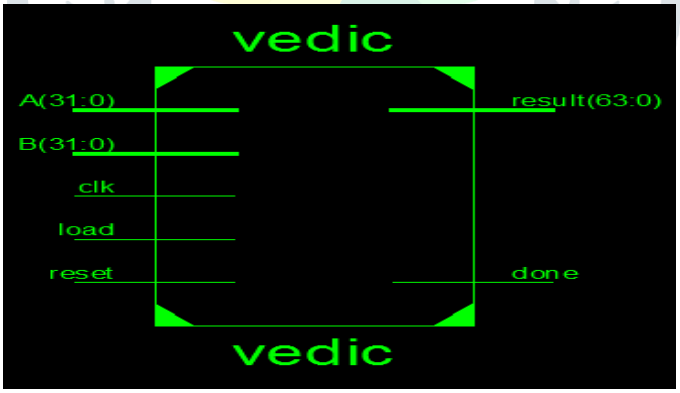

Fig.4. Module of 64 bit Vedic Multiplier.

Table 1: Comparison between 32 bit and 64 bit multiplier

Parameters	32 bit Multiplier	64 bit Multiplier
Maximum Frequency	156.018 MHz	148.922 MHz
Minimum Period	6.41 ns	6.715 ns
Minimum input arrival time before clock	5.308 ns	5.741 ns
Maximum output required time after clock	6.767 ns	7.125 ns

Maximum combinational path delay	No path Delay	No path Delay

Table 1 shows the comparison of frequency and period required for 32-bit and 64-bit Aging-Aware Vedic combinational path delay existence between the input and the output of the multiplier. Fig. 5 shows the simulated output waveform of 64-bit Vedic Multiplier

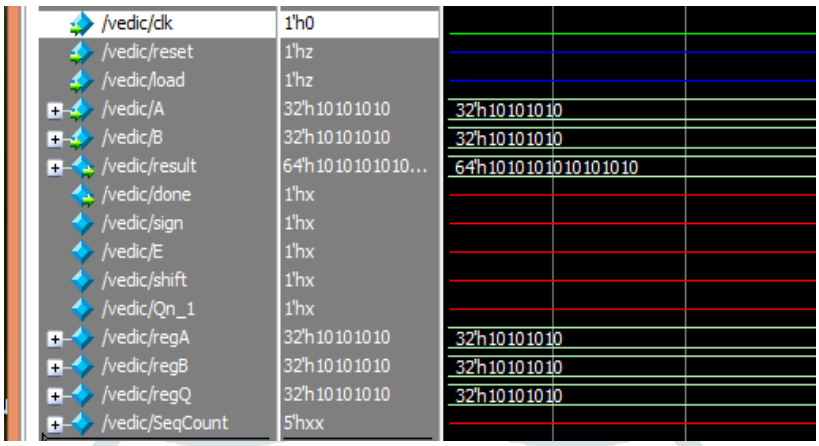

Fig.5. Simulation Waveform of 64-bit Multiplier

In the above simulation diagram, clock is represented by clk, multiplicand is represented by A, multiplier by B, reset represents reset signal and the final product is represented as result.

V. CONCLUSION

This paper proposes a low power variable latency multiplier design using Adaptive Hold Logic to overcome the timing violations that would occur by fixed latency design. The multiplier has the ability to resist to the performance degradation that would otherwise occur due to the effect of aging. This Razor based multiplier design with AHL proves to have higher efficiency with reduced latency. The frequency and period for 32 bit and 64 bit multipliers are tabulated in this paper. The synthesis and simulation of the proposed architecture of 64 bit multiplier was carried out using Xilinx and ModelSim respectively and Verilog is used for coding.

REFERENCES

[1] N. H. E. Weste and D. M. Harris. 2010. Integrated Circuit Design, Pearson.
[2] Deepak Purohit, Himanshu Joshi. 2014. "Comparative Study & Analysis of Fast Multipliers", International Journal of Engineering & Technical Research (IJETR), 2(7).
[3] D. Pal and A. G. Naik. 2018. "Delay estimation, chip-power analyses and comparison of single-level and multi-level recursive Vedic algorithm with conventional algorithms for digital multiplier," 2018 International Conference on IC Design & Technology (ICICDT), Otranto, pp. 29-32.
[4] Swami Bharathi Krishna Tirthaji Maharaja. 1965. "Vedic Mathematics", Motilal Banarsidass Publishers.
[5] D. K. Kahar and H. Mehta. 2017. "High speed Vedic multiplier used Vedic mathematics," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, pp. 356-359.
[6] M. Olivieri. 2001. "Design of synchronous and asynchronous variable-latency pipelined multipliers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 9(2): 365-376.
[7] Y. Chen et al. 2010. "Variable-Latency Adder (VL-Adder) Designs for Low Power and NBTI Tolerance," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18(11): 1621-1624.
[8] I. Lin, Y. Cho and Y. Yang. 2015. "Aging-Aware Reliable Multiplier Design with Adaptive Hold Logic," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(3): 544-556.
[9] J. Yan and Z. Chen. 2009. "Low-power multiplier design with row and column bypassing," 2009 IEEE International SOC Conference (SOCC), Belfast, pp. 227-230.
[10] Ahuja, M., and S. Bajaj. 2013. "Design and Analysis of Bypassing Multiplier." Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom).
[11] Mathur, M. and Aarnav. 2017. "Demystification of Vedic Multiplication Algorithm", American Journal of Computational Mathematics, Vol. 7, pp.94-101.
[12] D. Ernst et al. 2003. "Razor: a low-power pipeline based on circuit-level timing speculation," Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-36., San Diego, CA, USA, pp. 7-18.