International Journal of Electrical and Electronic Engineering & Telecommunications

ISSN 2319 – 2518 www.ijeetc.com
Vol. 4, No. 4, October 2015

Research Paper

REALIZATION OF LOW POWER AND AREA REDUCED AGING-AWARE MULTIPLIER DESIGN

Ch Pavan Mohan[1]* and M Sumalatha[2]

*Corresponding Author: Ch Pavan Mohan, ✉ pavanmohan450@gmail.com

Low power design has been an important part in VLSI system design. Digital multipliers are most critical functional units of digital filters. The overall performance of digital filters depends on the throughput of multiplier design. Aging problem of transistors has a significant effect on performance of these systems and in long term, the system may fail due to delay problems. Aging effect can be reduced by using over-design approaches, but these approaches leads to area, power inefficiency. Moreover, timing violations occur when fixed latency designs are used. Hence to reduce timing violations and to ensure reliable operation under aging effect, low power variable latency multiplier with adaptive hold logic is used. Negative bias and positive bias temperature both effects degrade transistor speed, and in the long term, the system may fail due to timing violations. Therefore, it is important to design reliable high-performance multipliers. In this paper, we propose a Vedic multiplier design with a razor based multiplier circuit. This design is able to provide higher throughput through the area and power efficiency enhancement than the existing Column By passing multiplier. This multiplier design can be applied to digital filters so as to enhance its performance in the real time environment.

Keywords: Aging Effect, Vedic Multiplier, Razor Flipflops, AHL

INTRODUCTION

Multiplication operation is one of the area which consuming more arithmetical operations in high performance circuits. As for importance many of the researchers deal with high speed multipliers of low power design. Multiplication operation contains two basic operations, one to generate partial products and another one to generate their sum and this performed using two types of multiplication algorithms parallel and serial. Where the Serial multiplication algorithms use sequential circuits with feedbacks, whereas the inner products are sequentially produced and the computed. Parallel multiplication algorithms often use

[1] M.Tech Student, Department of ECE, Chirala Engineering College, Chirala, A.P., India.

[2] Assistant Professor, Department of ECE, Chirala Engineering College, Chirala, A.P., India.

combinational circuits and these never contain any feedback structures.

A multiplier is one of the key hardware blocks in most Digital Signal Processing (DSP) systems. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis. Many current DSP applications are targeted at portable, battery-operated systems, so that power dissipation becomes one of the primary design constraints. Since multipliers are rather complex circuits and must typically operate at a high system clock rate, reducing the delay of a multiplier is an essential part of satisfying the overall design.

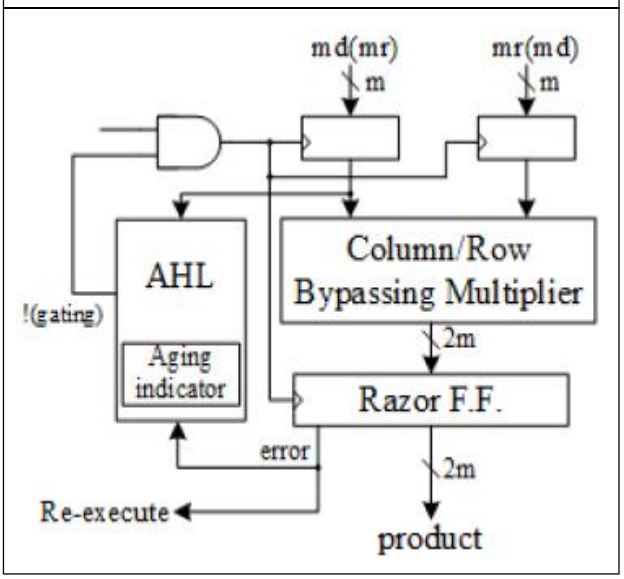
The aging behavior of digital designs is a major topic of research within the integrated circuits. There exists a large body of work that addresses aging effects at the hardware level. For example, Wu and Marculescu presented optimization techniques that allow synthesizing digital circuits that are less prone to aging degradation. In different process variations at a chip level are studied and characterized. Software techniques are presented to cope with problems that are posed by unreliable hardware. These include devising reliability-aware instruction sets, coupled with appropriate instruction scheduling using reliability aware compilation techniques. Recently, in Huang *et al.* presented an allocation framework based on a heuristic that aims at maximizing the lifetime of a System-on-Chip (SoC). Our work here follows this line of research and also deals with the allocation issues that arise in aging devices.

In this paper, we mainly consider those hard errors that are permanent once they manifest, such as Time Dependent Dielectric Breakdown (TDDB), Electron Migration (EM), and Negative Bias Temperature Instability (NBTI).While the fundamental causes of the above hard intrinsic failure shave been studied for decades, it has recently re-attracted lots of research interests, due to their increasingly adverse effects with technology scaling. Srinivasan *et al.* proposed an application aware architecture level model RAMP to evaluate a processor's lifetime reliability. Shin et al. defined reference circuits and presented a structure-aware lifetime reliability estimation framework. The above models focus on analyzing single-core processor's life time reliability.

AGING-AWARE MULTIPLIER

The aging-aware multiplier architecture, which includes two m-bit inputs (m is a positive number), one 2 m-bit output, one column-or row-bypassing multiplier, 2 m 1-bit Razor flip-flops, and an AHL circuit.

Figure 1: Aging Aware Architecture (md Means Multiplicand; mr Means Multiplicator)



Hence, 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 AHL. According to the bypassing selection in the column or row-bypassing multiplier, the input signal of the AHL in the architecture with the column-bypassing multiplier is the multiplicand, whereas of the row-bypassing multiplier is the multiplicator.

Figure 2: Razor Flip Flop Architecture

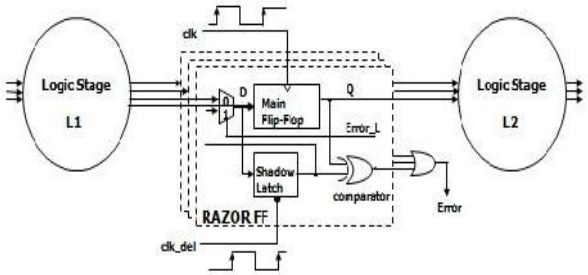
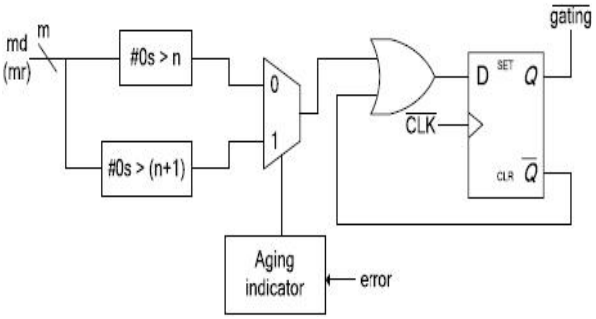


Figure 3: Diagram of AHL (md Means Multiplicand; mr Means Multiplicator)



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 mux. 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 re execute the operation and notify the AHL circuit that an error has occurred.

If not, the operation is re executed with two cycles. Although the re execution may seem costly, the overall cost is low because the re execution frequency is low. The AHL circuit is the key component in the aging-ware variable-latency multiplier. Figure 3 shows the details of the AHL circuit. The AHL circuit contains an aging indicator, two judging blocks, one mux, and one D flip-flop. The aging indicator indicates whether the circuit has suffered significant performance degradation due to the aging effect. The aging indicator is implemented in a simple counter that counts the number of errors over a certain amount of operations and is reset to zero at the end of those operations. If the cycle period is too short, the column- or row-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.

Compared with the first judging block, the second judging block allows a smaller number of patterns to become one-cycle patterns because it requires more zeros in the multiplicand (multiplicator). The details of the operation of the AHL circuit are as follows: when an input pattern arrives, both judging blocks will decide whether the pattern requires one cycle or two cycles to complete and pass

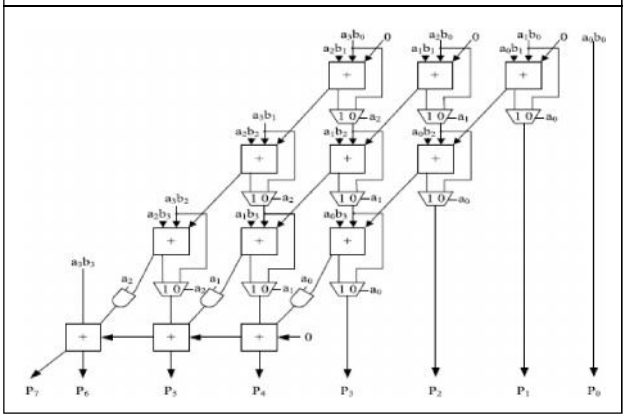
both results to the multiplexer. The multiplexer selects one of either result based on the output of the aging indicator. Then an OR operation is performed between the result of the multiplexer, and the Q_signal is used to determine the input of the D flip-flop. When the pattern requires one cycle, the output of the multiplexer is 1.

The overall flow of the architecture is as follows: when input patterns arrive, the column- or row-bypassing multiplier, and the AHL circuit execute simultaneously. According to the number of zeros in the multiplicand (multiplicator), the AHL circuit decides if the input patterns require one or two cycles. If the input pattern requires two cycles to complete, the AHL will output 0 to disable the clock signal of the flip-flops. Otherwise, the AHL 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.

Column-Bypassing Multiplier

The FAs in the AM are always active regardless of input states. In, a low-power column-bypassing multiplier design is proposed in which the FA operations are disabled if the corresponding bit in the multiplicand is 0. Figure 4 shows a 4 × 4 column-bypassing multiplier. Supposing the inputs are 10102 * 11112, it can be seen that for the FAs in the first and third diagonals, two of the three input bits are 0: the carry bit from its upper right FA and the partial product aibi. Therefore, the output of the adders in both diagonals is 0, and the output sumbit is simply equal to the third bit, which is the sum output of its upper FA. Hence, the FA is modified to add two tristate gates and one multiplexer. The multiplicand bit ai can be used as the selector of the multiplexer to decide the output of the FA, and aican also be used as the selector of the tristate gate to turn off the input path of the FA. If α_i is 0, the inputs of FA are disabled, and the sum bit of the current FA is equal to the sum bit from its upper FA, thus reducing the power consumption of the multiplier. If α_i is 1, the normal sum result is selected.

Figure 4: 4 × 4 Column-Bypassing Multiplier



Adaptive Hold Logic

The most critical concern in sub threshold circuits is to achieve high level of performance with very tight power constraints. This is evident in the development of mobile phones: in last one decade talk-time per gram of battery has improved by 60x. Challenges that prevent sub-threshold circuits from being widely used are their performances dependency on different Process Voltage and Temperature (PVT) conditions.

That is why the classical guard band methodology for “worst-case” is no more

efficient, so some adaptive performance control techniques are required. Initially, the most critical paths of the circuits were replicated to track the correct functionality.Represents an application of adaptive performance control with replica circuit but, original critical path circuit and its replica part can't be identical from manufacturing point of view.To address these issues, different adaptive techniques were proposed.

VEDIC DESIGN USING REVERSIBLE ARCHITECTUE

In 1965, according to 'Moore's law', stated by Gordon Moore, Intel Co-founder, the performance of integrated circuits improve at an exponential rate with the performance per unit cost increasing by a factor of 2 every 18 months. This resulted in shrinking the dimensions on integrated structures to make it possible to operate at higher speed for the same power per unit area. One should not forget that there is a minimum of quantum energy associated with elementary events which puts a fundamental limit on the miniaturization. So the question is, will Moore's law going to end? Using current technology more and more components are getting packed onto the chip and at the same time the power dissipation in the present day computer is very high. So, one of the major current research trends is towards saving of the power. Later Bennett, in 1973, showed that in order to avoid KT ln2 joules of energy dissipation in a circuit it must be built from reversible circuits. Reversible logic ensures zero information loss and low power dissipation.

Thus reversibility will become an essential property in future circuit design. Synthesis of reversible logic circuits differs from the combinational one in many ways. Firstly, in reversible circuit there should be no fan-out, that is, each output will be used only once. Secondly, for each input pattern there should be a unique output pattern. Finally, the resulting circuit must be acyclic. Any reversible gate performs the permutation of its input patterns only and realizes the functions that are reversible. If a reversible gate has k inputs, and therefore k outputs, then we call it a k×k reversible gate. Any reversible circuit design includes only the gates that are reversible. In a reversible circuit, the outputs that are not used as primary outputs are called garbage and the input lines that are set to constants are termed as constant inputs. An efficient design should keep the number of garbage outputs to minimum.

Reversible Logic Gates

A reversible logic gate is an n-input n-output logic device with one-to-one mapping. This helps to determine the outputs from the inputs and also the inputs can be uniquely recovered from the outputs. Also in the synthesis of reversible circuits direct fan-out is not allowed as one–to-many concept is not reversible. However fanout in reversible circuits is achieved using additional gates. A reversible circuit should be designed using minimum number of reversible logic gates.

Peres Gate: This gate contains 3 inputs and 3 outputs which are nothing but A, B, C and P, Q, R.

Figure 5: Block Diagram of Peres Gate

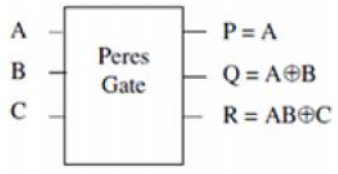
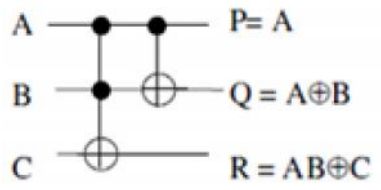


Figure 6: Symbol of Peres Gate



CNOT Gate

CNOT gate is also known as controlled-not gate. It is a 2*2 reversible gate. The CNOT gate can be described as:

Iv = (A, B); Ov = (P = A, Q = AB)

Figure 7: Block Diagram of CNOT Gate

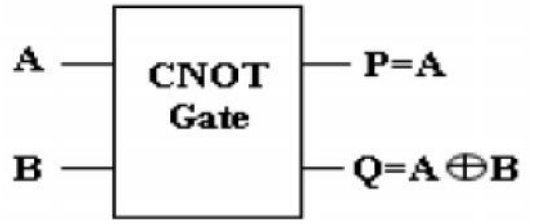
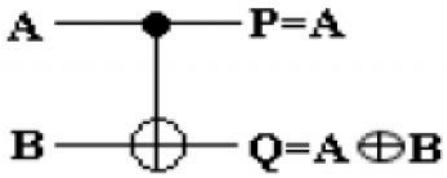


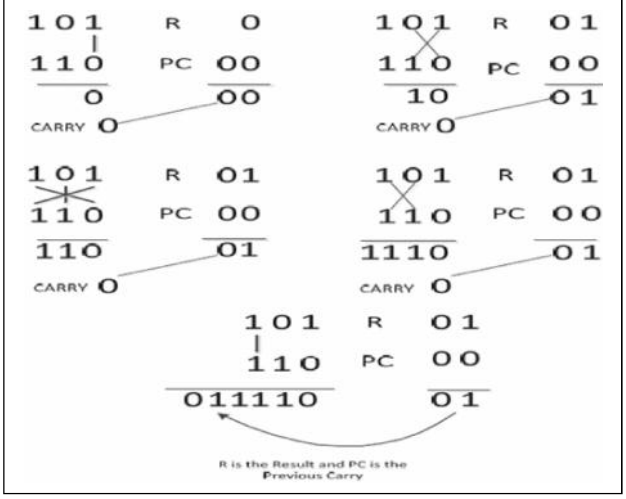
Figure 8: Symbol of CNOT Gate



Urdhva Tiryakbhayam Multiplication Algorithm

Urdhva Tiryakbhayam (UT) is a multiplier based on Vedic mathematical algorithms deviced by ancient Indian Vedic mathematicians. Urdhva Tiryakbhayam sutra can be applied to all cases of multiplications viz. Binary, Hex and also Decimals. It is based on the concept that generation of all partial products can be done and then concurrent addition of these partial products is performed. The parallelism in generation of partial products and their summation is obtained using Urdhva Tiryakbhayam. Unlike other multipliers with the increase in the number of bits of multiplicand and/or multiplier the time delay in computation of the product does not increase proportionately. Because of this fact the time of computation is independent of clock frequency of the processor. Hence one can limit the clock frequency to a lower value. Also, since processors using lower clock frequency dissipate lower energy, it is economical in terms of power factor to use low frequency processors employing fast algorithms like the above mentioned. The Multiplier based on this sutra has the advantage that as the number of bits increases, gate delay and area increases at a slow pace as compared to other conventional multipliers.

Figure 9: Urdhva Tiryakbhayam Procedure for Multiplication



Design of 4 x 4 Urdhva Tiryakbhayam Multiplier

The Reversible 4 x 4 Urdhva Tiryakbhayam Multiplier design emanates from the 2 x 2 multiplier. The block diagram of the 4 x 4 Vedic Multiplier is presented in the Figure 12. It consists of four 2 x 2 multipliers each of which

procures four bits as inputs; two bits from the multiplicand and two bits from the multiplier. The lower two bits of the output of the first 2 x 2 multiplier are entrapped as the lowest two bits of the final result of multiplication.

Two zeros are concatenated with the upper two bits and given as input to the four bit ripple carry adder. The other four input bits for the ripple carry adder are obtained from the second 2 x 2 multiplier. Likewise the outputs of the third and the terminal 2 x 2 multipliers are given as inputs to the second four bit ripple carry adder. The outputs of these four bit ripple carry adders are in turn 5 bits each which need to be summed up. This is done by a five bit ripple carry add which generates a six bit output. These six bits from the upper bits of the final result.

Ripple Carry Adder

The design shown in consists of only HNG gates. The number of HNG gates is 4 if the ripple carry adder is used in the second stage or five if the ripple carry adder is used in the last stage of the 4 x 4 Urdhva Tiryakbhayam Multiplier. The ripple carry adder can be modified as under. Since for any ripple carry adder the input carry for the first full adder is zero, this implicitly means the first adder is a half adder. Thus a Peres gate can efficiently replace a HNG. This cut down the quantum cost by two for any ripple carry adder and the garbage output by one. The Constant inputs and the gate count remain unchanged.

Figure 10: Modified 5 Bit Ripple Carry Adder Design

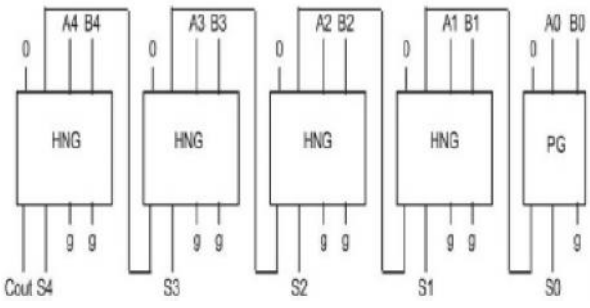


Figure 11: Modified 4 Bit Ripple Carry Adder Design

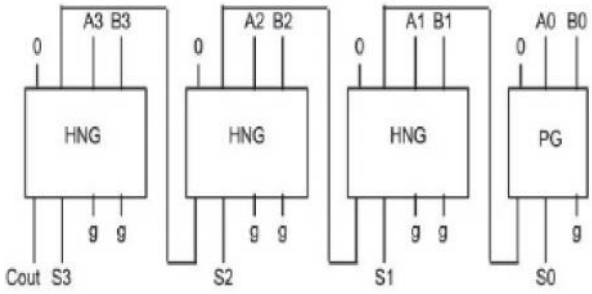
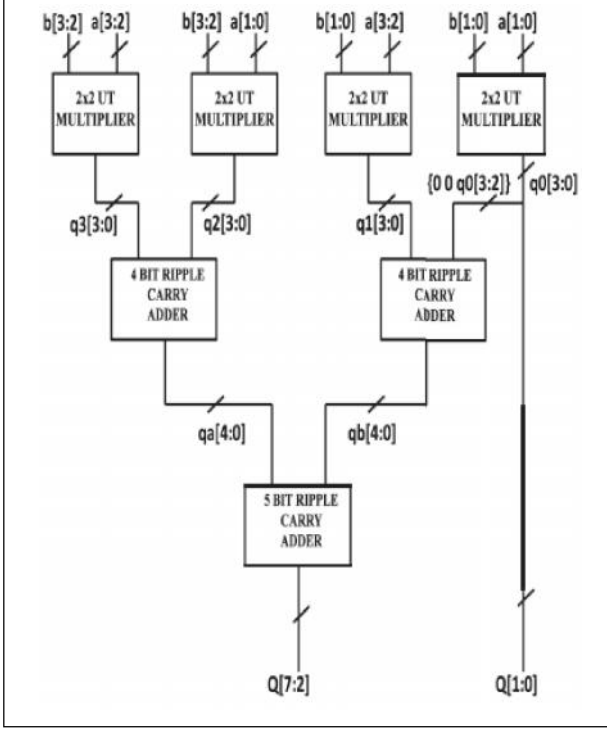


Figure 12: Design of 4 x 4 Urdhva Tiryakbhayam Multiplier

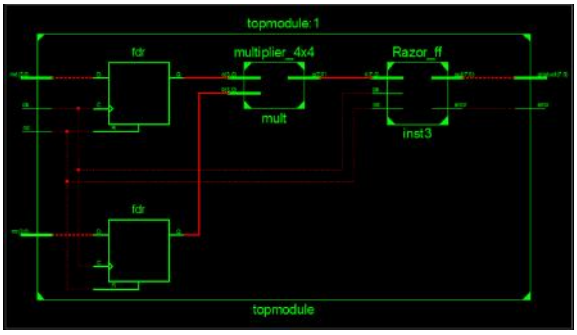


RESULTS AND DISCUSSION

RTL Schematic for the Top Module

The RTL SCHEMATIC gives the information

Figure 13: RTL Schematic for the Top Module

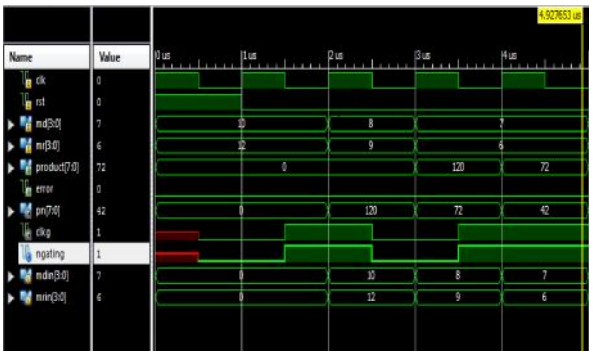


about the user view of the design. The internal blocks contains the basic gate representaion of the logic. These basic gate realization is purely depend upon the corresponding FPGA selection and the internal database information.

Output Wave Form for the Top Module

In the waveform which is shown above, clk signal represents clock, rst signal represents reset, md represents multiplicand, mr represents multiplier which we are applying as inputs to the design. Similarly product is the output signal for the design. Here clock signal is generated for the positive edge. Initially the reset signal should be force to logic 1 and after one clock cycle made it to logic 0 for performing the corresponding functional operation. To obtain the required outputs force the inputs logic with the required values. The output product get the multiplicity value of the applied inputs md and mr.

Figure 14: Output Wave Form for the Top Module



From the above table the power and area is less for the proposed Aging aware Vedic multiplier than the existing Aging aware column multiplier.

Table 1: Comparision Between Existing Column Multiplier and Proposed Vedic Multiplier Aging Aware Design

Method	Area (in Terms of LUT's)	Power (mw)
Aging aware with column	72	0.58701
Aging aware with Vedic multiplier	61	0.49732

CONCLUSION

Aging problem of transistors has a significant effect on performance of these systems and in long term, the system may fail due to delay problems, which can be reduced by using over-design approaches. This paper proposed an aging-aware variable-latency multiplier design with the AHL. The multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. The experimental results show that our proposed architecture with 4 x 4 multiplication using vedic mathematics is the area and power efficient design compared to the existing column bypass multiplier. The Verilog language is used for coding. The synthesis and simulation is carried out using Xilinx ISE 12.3i.

REFERENCES

1. Abrishami H, Hatami S, Amelifard B and Pedram M (2008), "NBTI-Aware Flip-Flop

Characterization and Design", in *Proc. 44th ACM GLSVLSI*, pp. 29-34.

2. Calimera A, Macii E and Poncino M (2012), "Design Techniqures for NBTI Tolerant Power-Gating Architecture", *IEEE Trans. Circuits Syst., Exp. Briefs*, Vol. 59, No. 4, pp. 249-253.
3. Cao Y (2013), *Predictive Technology Model (PTM) and NBTI Model* [Online], available: http://www.eas.asu.edu/<"ptm
4. Kumar S V, Kim C H and Sapatnekar S S (2007), "NBTI-Aware Synthesis of Digital Circuits", in *Proc. ACM/IEEE DAC*, June, pp. 370-375.
5. Lee Y and Kim T (2011), "A Fine-Grained Technique of NBTI-Aware Voltage Scaling and Body Biasing for Standard Cell Based Designs", in *Proc. ASPDAC*, pp. 603-608.
6. Vattikonda R, Wang W and Cao Y (2004), "Modeling and Miimization of pMOS NBTI Effect for Robust Naometer Design", in *Proc. ACM/IEEEDAC*, June, pp. 1047-1052.
7. Wu K-C and Marculescu D (2009), "Joint Logic Restructuring and Pin Reordering Against NBTI-Induced Performance Degradation", in *Proc. DATE*, pp. 75-80.
8. Yang H-I, Yang S-C, Hwang W and Chuang C-T (2011), "Impacts of NBTI/PBTI on Timing Control Circuits and Degradation Tolerant Design in Nanoscale CMOS SRAM", *IEEE Trans. Circuit Syst.*, Vol. 58, No. 6, pp. 1239-1251.
9. Zafar S *et al.* (2006), "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.
10. Zafar S, Kumar A, Gusev E and Cartier E (2005), "Threshold Voltage Instabilities in High-k Gate Dielectric Stacks", *IEEE Trans. Device Mater. Rel.*, Vol. 5, No. 1, pp. 45-64.