Designing of Multiplier with Improved AHL

Ankita Gupta
M-Tech Research Scholar
Deptt. of Electronics & Communication, NIIIST
Bhopal

Braj Bihari Soni
Professor, Research Guide
Deptt. of Electronics & Communication, NIIIST
Bhopal

Puran Gaur
Professor, HOD
Deptt. of Electronics & Communication, NIIIST
Bhopal

ABSTRACT

The effects aging of digital circuits are came into the focused due to observations made with several experiments and researchers has start working towards making changes for the improvements in base paper architecture. The integrated device suffers with NBTI and PBTI due to CMOS semiconductor properties and it affects the working of different logic operations and in the same context here we have taken multiplier for consideration and working to develop delay efficient multiplier with aging aware design using adaptive hold logic which is modified in this work to reduce effective delay to speedup circuit logic. The simulation of experiments are conducted in Xilinx IDE 13.1.

General Terms

Multiplier Design on Xilinx IDE 13.1

Keywords

Adaptive hold logic, row bypassing, column bypassing, Multiplier, Aging Effect, NBTI, PBTI, Delay Efficient

1. INTRODUCTION

Multipliers are key segments of many high performance systems, for example, FIR filters, microchips, DSPs, and so forth. The performance of the system is generally determined by the execution of the multiplier in light of the fact that the multiplier is basically the slowest merciful in die system. Moreover, it is generally die most area consuming. Subsequently, the major issue is to design and optimize die speed consumed area. Notwithstanding, area and speed are normally conflicting requirements so that enhancing speed comes about for large areas.

At most essential level, digital multiplication can be viewed as a progression of bit moves and bit additions,, where two numbers, where two numbers, the multiplier and the multiplicand are consolidated into the last outcome. Consider the multiplication of two numbers: the multiplier P, and multiplicand C, where P is a n-bit number with bit representation {pn-1,pn-2,…..p0 }, the most significant bit being pn-1 and the slightest huge bit being p0; C has a comparative bit representation{cn-1,cn-2,..c0 }. For unsigned multiplication, up to n moved copies of the multiplicand are added to form the outcome. The whole system is separated into three steps: partial product (PP) era, partial product diminishment, and last addition.

While some understanding can be increased through direct perception of logic structure, power dissipation originates from a few sources; strategies which reduce the power because of one of these sources can intensify the power scattering because of another.

The multiplier is able to provide higher throughput through the variable latency and can adjust the AHL circuit to mitigate performance degradation that is due to the aging effect.

1.1 Latency or Delay

Latency is the delay between inputs provided into a system to taken desired outcome; the term is seen marginally diversely in different settings and inactivity issues additionally change starting with one framework then onto the next. The latency basically, a period of time that one segment in a system is spinning its wheels waiting for another segment or component. Therefore, Latency is wasted time inside of a system. For instance, in getting to information on a disc, inactivity is characterized as the time it takes to position the correct part under the read/write head. In networking, the total taken to transfer a packet from source to destination. Together, latency and data transfer capacity characterize the speed and limit of a system. In VoIP phrasing, inactivity alludes to a delay in bundle conveyance. VoIP idleness is an administration issue that is generally in view of physical separation, bounces, or voice to information change.

In fixed latency scheme, critical path delay as the general circuit check cycle with a specific end goal to perform effectively. Be that as it may, the likelihood that the critical paths are actuated is low. For these noncritical paths, utilizing the basic path delay as the general cycle time frame will bring about noteworthy exercise in futility. Thus, the fixed latency with column by passing (FLCB) and fixed latency with row bypassing (FLRB) is better outline contrasting option to diminish the latency of CMOS circuits.

1.2 AM

Array multiplier is notable because of its consistent structure. Multiplier circuit depends on add and shift calculation. Every partial product is created by the multiplication operation of the multiplicand with one multiplier bit. The partial product is shifted by their bit orders and then added.

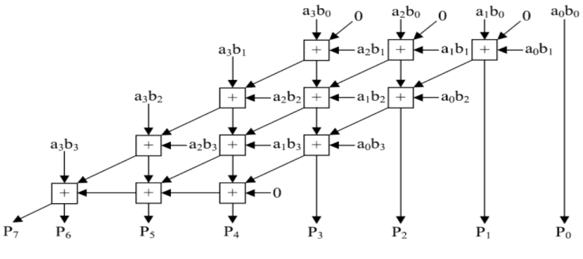


Fig. 1. 4 × 4 normal AM.

Fig. 1.1 Array Multiplier Design

1.3 FLCB

Column passing multiplier dispenses with the additional revising circuit to avoid the full adder cell and furthermore expends lesser power higher frequency of operation. This multiplier comprises of rows of carry save adders. The significant concentration of this multiplier is to diminish the switching moves required to play out the calculations.

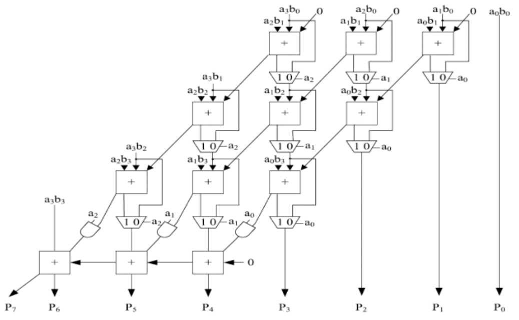


Fig. 2. 4 × 4 column-bypassing multiplier.

Fig. 1.2 Column Bypassing Multiplier Design

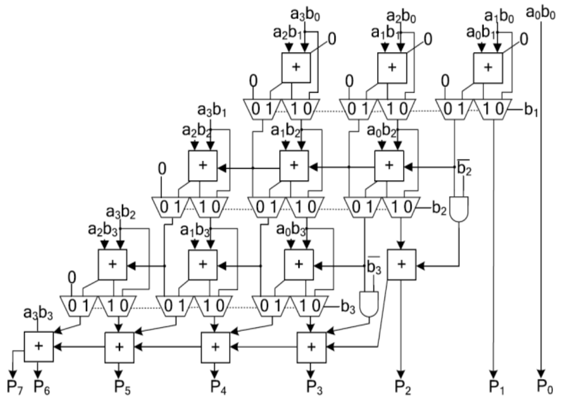


Fig. 1.3 Row Bypassing Multiplier Design

1.4 FLRB

Row bypassing multiplier expends lesser power than higher frequency of operation. It comprises of the rows of the ripple carry based full adder cells. Tri-state cradles at the contribution of the adder cells are embedded for reducing the switching transitions, if these cells are bypassed. While, multiplexer is inserted to select the Sum yield under no bypassing condition or when the bypassing is utilized.

1.5 AHL

A novel architecture of an Adaptive Hold Logic (AHL) circuit which will decrease the aging impacts. The Adaptive Hold Logic (AHL) circuit can choose whether the pattern of input require one or two cycles and can Adjust the judging criteria to guarantee that there is least execution debasement after significant aging occurs. The Adaptive Hold Logic (AHL) circuit is as appeared in fig. 1.4. Accept the AHL circuit has a m bit input. The Adaptive Hold Logic (AHL) circuit comprises of the accompanying squares. a. Judging Blocks b. D Flip-Flop c. One Multiplexer d. Aging Indicator.

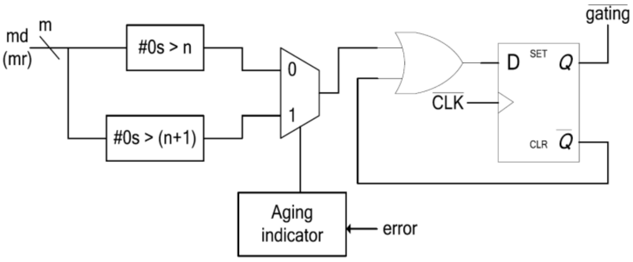


Fig. 1.4 Adaptive Hold Logic

2. SYSTEM MODEL

The interest for frameworks with elite has expanded hugely, driving the move of CMOS innovation towards all the more profoundly scaled nanometer include sizes. While more noteworthy on-chip gadget incorporation inside similar chip zone offers higher processing abilities, highlight measure scaling likewise brings about a corresponding lessening in the probability that the created chip, at the postsilicon arrange, meets the details created in the outline stream before manufacture, at the presilicon organize[6].

This deviation is essentially owing to the impacts of varieties (process, natural, and maturing), which have become bigger with contracting highlight sizes. This represents a significant test to accomplish a concurrent conclusion on the triple measurements of execution: solid registering, high throughput, and low power[9].

Similarly, maturing varieties in circuits because of predisposition temperature shakiness (BTI), hot transporter infusion (HCI), and time subordinate dielectric breakdown (TDDB), cause the circuit postponement to corrupt after some time. For a 32nm Predictive Technology Model (PTM) [11] based design, the degradation in this delay is about 24% for des, and causes functional failures early in lifetime (by violating the clock period, Tclk) without adequate delay guard bands. The effects of aging, as with those of process variations, are more significant with more deeply scaled CMOS technology nodes[8].

3. PROPOSED ARCHITECTURE

In the proposed aging – aware reliable multiplier design. Presented the overall architecture and functioning. the architecture is simulated on result has taken from the Xilinx platform.

The description of proposed architecture is given below.

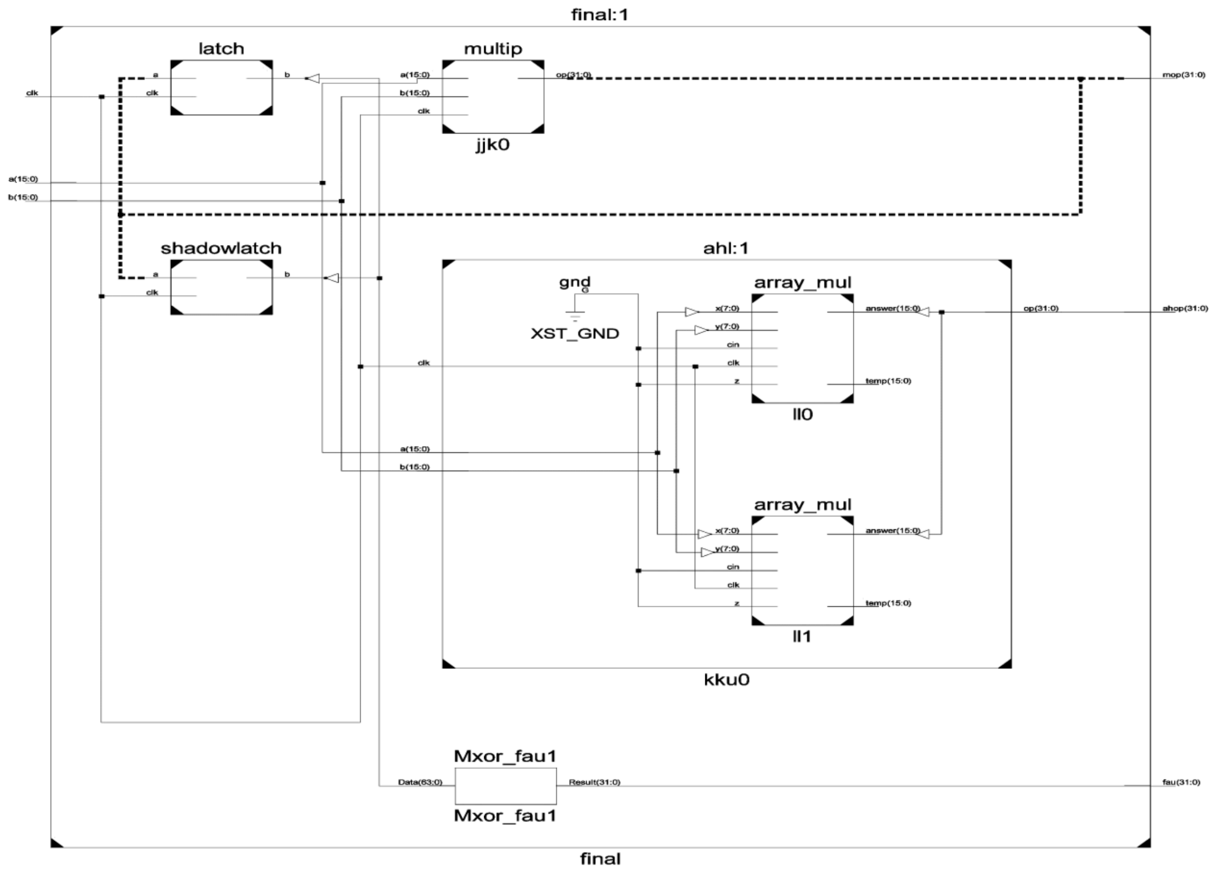


Figure: 3.1 RTL Schematic of Proposed Architecture

Figure 3.1 shows the proposed RTL schematic of architecture reliable aging- aware multiplier hold logic (AHL).which includes two inputs of m bit (m is a positive integer) and two out puts , in the proposed architecture as illustrated in figure 3.1 the RTL schematic diagram .

There are 4 sections in a proposed architecture having a latch, a shadow latch, a flip flop and two arrays of multipliers 0 and 1 correspondingly. The outcome has taken from a hop(31.0).,a(15.0) and b(15.0)are the inputs.

4. COMPARISON OF PROPOSED ARCHITECTURE

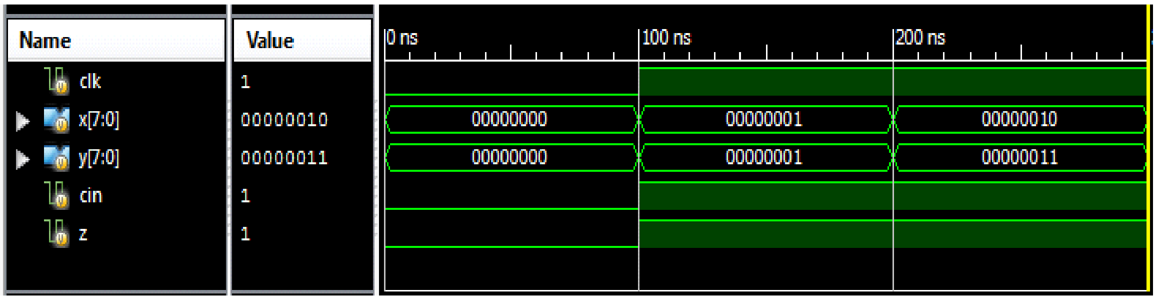
Proposed architecture for delay efficient design has 0.717ns, 0.986ns and 0.524ns is faster than the Base paper architectures which has 1.32ns, 1.82ns and 1.88ns respectively for AM, FLRB and FLCB.

Table 4.1: Performance Comparison of Proposed Architecture with Base paper Architecture for AM, FLRB and FLCB

Parameters	Base Paper Architecture (Using AHL)	**Proposed Architecture (Improved AHL)**
AM	1.32 ns	**0.717 ns**
FLRB	1.82 ns	**0.986 ns**
FLCB	1.88 ns	**0.524 ns**

The performance evolutions and comparison is shown in table 4.1 with proposed architecture to architecture Base paper 4.1 shows comparison of AM, FLRB and FLCB .

(See Fig. 4.1) to (Fig. 4.4) the test bench waveforms of proposed architecture.



Name	Value
clk	1
x[7:0]	00000010
y[7:0]	00000011
cin	1
z	1

Fig.4.1 Test bench waveforms of Array Multiplier

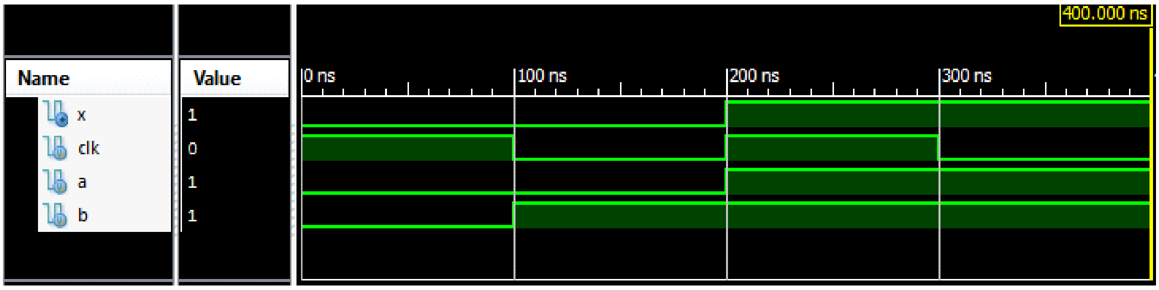


Fig. 4.2 Test bench waveforms of Full Adder

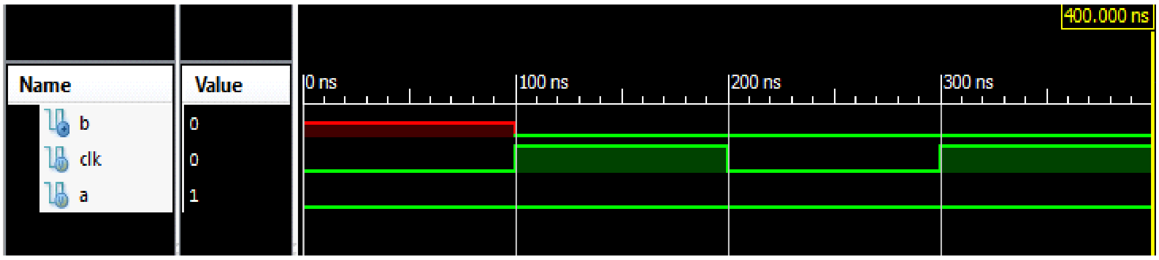


Fig. 4.3 Test bench waveforms of Latch

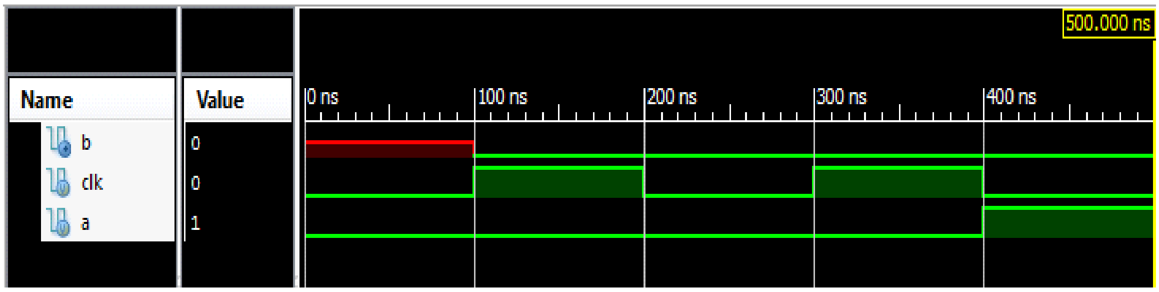


Fig. 4.4 Test bench waveforms of Shadow Latch

5. CONCLUSION AND FUTURE SCOPE

delay efficient multiplier design is explained and synthesized in this work. Simulation results shown that the delay outcomes of the proposed architecture are better than the base paper results. Synthesis of proposed architecture taken with the outcomes in terms of latency, which is explained and compared in the table 4.1. In the synthesis outcomes it is clear that the modified approach for delay efficient design has 0.717ns, 0.986ns and 0.524ns is faster than the previous architectures which has 1.32ns, 1.82ns and 1.88ns respectively for AM, FLRB and FLCB. The comparative analysis of synthesis results clearly concludes that the proposed 16x16 architecture is about 50% faster than the previous architecture and it is better prone to NBTI and PBTI effects which slow down the calculations of digital circuits, Which will help to calculate transforms faster e.g. Fourier Transform, Discrete Cosine Transform and various digital filtering techniques. So that for the future up gradation in logic circuits proposed multiplier design will be useful to speed up the calculations and save time. This architecture can also be upgraded using logic and architecture level optimization approaches for future evolution.

6. REFERENCES

[1] I. C. Lin, Y. H. Cho and Y. M. Yang, "Aging-Aware Reliable Multiplier Design With Adaptive Hold Logic," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 3, pp. 544-556, March 2015.

[2] S. Zafar et al., "A Comparative Study of NBTI and PBTI (Charge Trapping) in SiO2/HfO2 Stacks with FUSI, TiN, Re Gates," 2006 Symposium on VLSI Technology, 2006. Digest of Technical Papers., Honolulu, HI, 2006, pp. 23-25.

[3] S. Zafar, A. Kumar, E. Gusev and E. Cartier, "Threshold voltage instabilities in high-κ gate dielectric stacks," in IEEE Transactions on Device and Materials Reliability, vol. 5, no. 1, pp. 45-64, March 2005.

[4] H. I. Yang, S. C. Yang, W. Hwang and C. T. Chuang, "Impacts of NBTI/PBTI on Timing Control Circuits and Degradation Tolerant Design in Nanoscale CMOS SRAM," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 58, no. 6, pp. 1239-1251, June 2011.

[5] R. Vattikonda, Wenping Wang and Yu Cao, "Modeling and minimization of PMOS NBTI effect for robust nanometer design," 2006 43rd ACM/IEEE Design Automation Conference, San Francisco, CA, 2006, pp. 1047-1052.

[6] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, “NBTI-aware flip-flop characterization and design,” in Proc. 44th ACM GLSVLSI, 2008, pp. 29–34

[7] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, “NBTI-aware synthesis of digital circuits,” in Proc. ACM/IEEE DAC, Jun. 2007, pp. 370–375.

[8] A. Calimera, E. Macii, and M. Poncino, “Design techniqures for NBTItolerant power-gating architecture,”

IEEE Trans. Circuits Syst., Exp. Briefs, vol. 59, no. 4, pp. 249–253, Apr. 2012.

[9] K.-C. Wu and D. Marculescu, "Joint logic restructuring and pin reordering against NBTI-induced performance degradation," in Proc. DATE, 2009, pp. 75–80.

[10] Y. Lee and T. Kim, "A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs," in Proc. ASPDAC, 2011, pp. 603–608.

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

[12] K.-C. Wu and D. Marculescu, "Aging-aware timing analysis and optimization considering path sensitization," in Proc. DATE, 2011, pp. 1–6.

[13] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," in Proc. DATE, 2012, pp. 1257–1262.

[14] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in Proc. DATE, 2008, pp. 1250–1255.

[15] D. Baneres, J. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in Proc. DATE, 2009, pp. 1704–1709.

[16] Y.-S. Su, D.-C.Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance" optimization using variable-latency design style," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 10, pp. 1874–1883, Oct. 2011.