IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 5, May 2016

Design and Analysis of Aging Aware and Efficient Baugh-Wooley Multiplier with Adaptive Hold Logic

Druva Kumar L[1], Parveez Shariff[2], Praveen J[3], Raghavendra Rao[4].

M.Tech Student, Dept. of ECE, Alva's Institute of Engg & Technology, Mijar, Moodbidri, Karnataka, India[1]

Sr. Assistant Professor, Dept. of ECE, Alva's Institute of Engg.& Tech, Mijar, Moodbidri, Karnataka, India[2,3,4]

Abstract: High speed, low power consumption are key requirement to any VLSI design. The power efficient multipliers play an important role. This paper presents an efficient implementation of a high speed, low power Baugh-Wooley multiplier using aging aware technique and adaptive hold logic. This study presented the design and implementation of Baugh Wooely multipliers using XILINX. In this work, Modified Baugh Wooley is having least area, power and delay. The Modified Baugh Wooley architecture with adaptive hold logic and aging awareness make this efficient and also reliable. 32 bit signed multiplication and fractional multiplication is carried out and verified with around 10000 test patterns.

Keywords: Low Power, Multiplier, Baugh-Wooley, Precision, Aging Aware, Adaptive hold.

I. INTRODUCTION

Multipliers play a pivotal role in many high performance systems such as Microprocessor, FIR filters, Digital Processors, etc. In its early stage, multiplication algorithms were proposed by Burton and Noaks in the year 1968, by Hoffman in the year 1986 and by Guilt and De Mori in the year 1969 for positive numbers. In the year of 1973 and 1979, Baugh-Wooley and Hwang proposed multiplication algorithm for numbers in two's complement form. Multiplication is hardware intensive and the main criteria of interest are higher speed, lower cost and lower power [1]. With development in technology, several researchers have tried multipliers which provide design targets such as low power consumption, increased speed, and regularity of layout or combination of them in one multiplier. This helps making them suitable for achieving compact high speed and low power implementation.

The performance of a system is generally controlled by the performance of the multiplier as the multiplier is usually the slowest element in the system. Furthermore, multiplier is normally the most area consuming element in the system. Therefore, optimizing its speed and area are vital design factors. However, area and speed are generally the conflicting constraints improving speed which results mostly in large area.

With ever increasing applications in portable equipment and mobile communications, the demand for high performance, low-power VLSI systems is gradually increasing. Digital signal processors and application specific integrated circuits depend on the efficient implementation of arithmetic circuits (adder and multiplier) to execute dedicated algorithm such as convolution, correlation and filtering [2]. A Baugh-Wooley multiplier using decomposition logic is presented here which increases speed when compared to the booth multiplier.

II. LITERATURE SURVEY

Traditional circuits use critical path delay as the overall circuit clock cycle in order to perform correctly. However, the probability that the critical paths are activated is low. In most cases, the path delay is shorter than the critical path. For these noncritical paths, using the critical path delay as the overall cycle period will result in significant timing waste. Hence, the variable-latency design was proposed to reduce the timing waste of traditional circuits.The variable-latency design divides the circuit into two parts: 1) shorter paths and 2) longer paths. Shorter paths can execute correctly in one cycle, whereas longer paths need two cycles to execute. When shorter paths are activated frequently, the average latency of variable-latency designs is better than that of traditional designs. For example, several variable-latency adders were proposed using the speculation technique with error detection and recovery [13]–[15]. A short path activation function algorithm was proposed in [16] to improve the accuracy of the hold logic and to optimize the performance of the variable-latency circuit. An instruction scheduling algorithm was proposed in [17] to schedule the operations on no uniform latency functional units and improve the performance of Very Long Instruction Word processors.

In[18], a variable-latency pipelined multiplier architecture with a Booth algorithm was proposed. In [19], process-variation tolerant architecture for arithmetic units was proposed, wherethe effect of process-variation is considered to increase thecircuit yield. In addition, the critical paths are divided into two shorter paths that could be unequal and the clock cycleis set to the delay of the longer one. These research designswere able to reduce the timing waste of traditional circuitsto improve performance, but they did not consider the agingeffect and could not adjust themselves during the runtime. Avariable-latency adder design that considers the aging effect was

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 5, May 2016

proposed in [20] and [21]. However, no variable-latency multiplier design that considers the aging effect and can adjust dynamically has been done.

- **Baugh-Wooley multiplier**

In signed multiplication the length of the partial products and the number of partial products will be very high. So an algorithm was introduced for signed multiplication called as Baugh-Wooley algorithm. The Baugh-Wooley multiplication is one amongst the cost-effective ways to handle the sign bits. This method has been developed so as to style regular multipliers, suited to 2's compliment numbers.

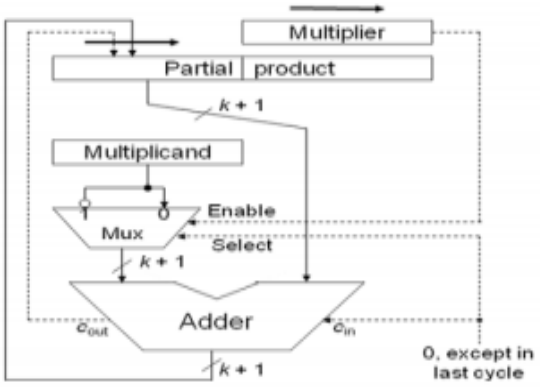

Fig: Baugh-Wooley multiplier architecture.

Baugh-Wooley Multiplier provides a high speed, signed multiplication algorithm [5]. It uses parallel products to complement multiplication and adjusts the partial products to maximize the regularity of multiplication array [6]. When number is represented in two's complement form, sign of the number is embedded in Baugh-Wooley multiplier. This algorithm has the advantage that the sign of the partial product bits are always kept positive so that array addition techniques can be directly employed [6]. In the two's complement multiplication, each partial product bit is the AND of a multiplier bit and a multiplicand bit, and the sign of the partial product bits are positive [6].

III. PROPOSED AGING AWARE MULTIPLIER

This section details the proposed aging-aware reliable multiplier design. It introduces the overall architecture and the functions of each component and also describes how to design AHL that adjusts the circuit when significant aging occurs.

A. Proposed Architecture

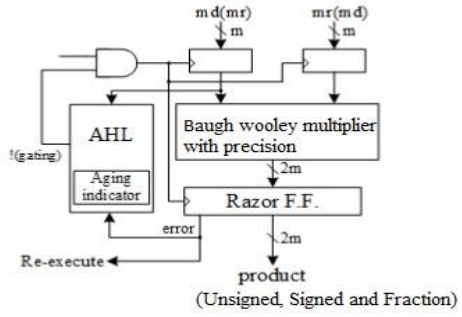

Fig 2: Proposed aging aware Baugh-Wooley multiplier.

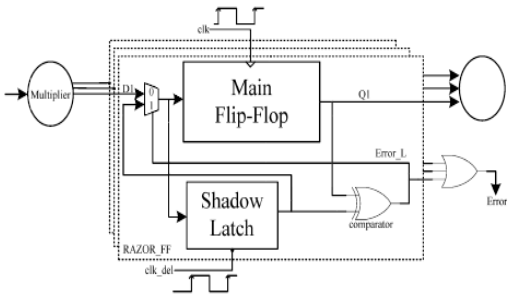

Fig. 3: Razor flip flops

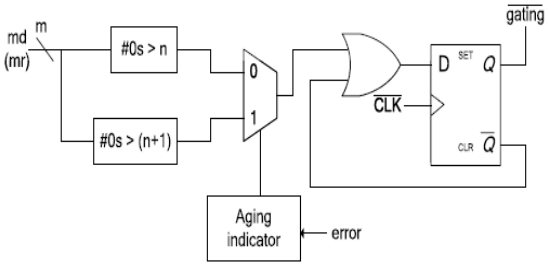

Fig 4: Diagram of AHL (md means multiplicand; mr means multiplicator).

Fig 2, shows our proposed aging-aware multiplier architecture, which includes two m-bit inputs (m is a positive number), one 2m-bit output, one Baugh-Wooley multiplier, 2m 1-bit Razor flip-flops [27], and an AHL circuit. The inputs of the row-bypassing multiplier are the symbols in the parentheses.In the proposed architecture, Baugh-Wooley multiplier multipliers can be examined by the number of zeros in either the multiplicand or multiplicator to predict whether the operation requires one cycle or two cycles to complete. When input patterns are random, the number of zeros and ones in the multiplicator and multiplicand follows a normal distribution. Therefore, using the number of zeros or ones as the judging criteria results in similar outcomes. 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 that of the row-bypassing multiplier is the multiplicator. Razor flip-flops can be used to detect whether timing violations occur before the next input pattern arrives.

Fig. 3 shows the details of Razor flip-flops. 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 reexecute the operation and notify the AHL circuit that an error has occurred. We use Razor

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 4, Issue 5, May 2016

flip-flops to detect whether an operation that is considered to be a one-cycle pattern can really finish in a cycle. If not, the operation is reexecuted with two cycles.

Although the reexecution may seem costly, the overall cost is low because the reexecution frequency is low. More details for the Razor flip-flop can be found in [27]. The AHL circuit is the key component in the aging-ware variable-latency multiplier. Fig. 12 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. If errors happen frequently and exceed a predefined threshold, it meansthe circuit has suffered significant timing degradation due to the aging effect, and the aging indicator will output signal 1; otherwise, it will output 0 to indicate the aging effect is still not significant, and no actions are needed.

The first judging block in the AHL circuit will output 1 if the number of zeros in the multiplicand (multiplicator for the Baugh-Wooley multiplier) is larger than n (n is a positive number, which will be discussed in Section IV), and the second judging block in the AHL circuit will output 1 if the number of zeros in the multiplicand (multiplicator) is larger than n + 1. They are both employed to decide whether an input pattern requires one or two cycles, but only one of them will be chosen at a time. In the beginning, the aging effect is not significant, and the aging indicator produces 0, so the first judging block is used. After a period of time when the aging effect becomes significant, the second judging block is chosen. 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 $\bar{Q}$ 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 !(gating) signal will become 1, and the input flip flops will latch new data in the next cycle. On the other hand, when the output of the multiplexer is 0, which means the input pattern requires two cycles to complete, the OR gate will output 0 to the D flip-flop. Therefore, the !(gating) signal will be 0 to disable the clock signal of the input flip-flops in the next cycle. Note that only a cycle of the input flip-flop will be disabled because the D flip-flop will latch 1 in the next cycle.

IV.RESULTS

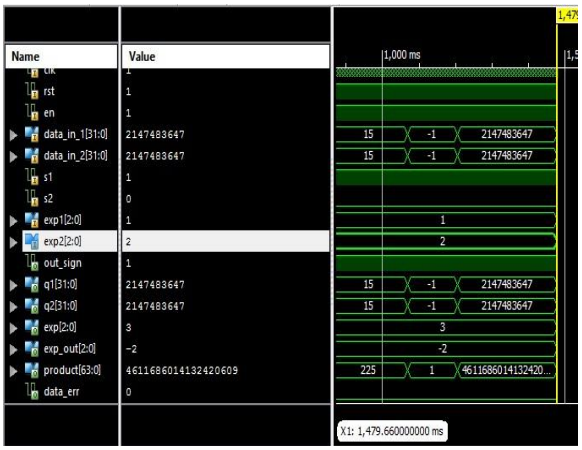

Fig 5: Simulation result

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	1625	204000	1
Signals	0.000	1919	---	---
IOs	0.000	208	600	35
Leakage	0.117			
Total	0.117			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	1.4	84.8	25.2

Fig 6: power estimation

Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	99	408,000	1%	
Number used as Flip Flops	99			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	1,625	204,000	1%	
Number used as logic	1,620	204,000	1%	
Number using O6 output only	1,547			
Number using O5 output only	0			
Number using O5 and O6	73			
Number used as ROM	0			
Number used as Memory	0	70,200	0%	
Number used exclusively as route-thrus	5			
Number with same-slice register load	5			
Number with same-slice carry load	0			
Number with other load	0			
Number of occupied Slices	712	51,000	1%	
Number of LUT Flip Flop pairs used	1,649			
Number with an unused Flip Flop	1,557	1,649	94%	
Number with an unused LUT	24	1,649	1%	
Number of fully used LUT-FF pairs	68	1,649	4%	
Number of unique control sets	3			
Number of slice register sites lost to control set restrictions	13	408,000	1%	
Number of bonded IOBs	208	600	34%	
IOB Flip Flops	67			

Fig 7: Area Estimation

Our experiments are conducted in a windows operating system. We adopt a 32-nm high-k predictive technology model [1] to estimate the BTI degradation for seven years.

The proposed multiplier is designed in Verilog. In the variable-latency design, the average latency is affected by both the percentage of one-cycle patterns and the cycle period. If more patterns only require one cycle, the average latency is reduced. Similarly, if the cycle period is reduced, the average latency is also reduced. However, the cycle period cannot be too small. If the cycle period is too small, large amounts of timing violations will be detected by the Razor flip-flops, and the average latency will increase. Hence, it is important to analyse the tradeoffs between the percentage of one-cycle patterns and the cycle period.

V. CONCLUSION

This paper proposed an aging-aware variable-latency multiplierdesign with the AHL. The multiplier is able to adjustthe AHL to mitigate performance degradation due to increaseddelay. The experimental results show that our proposed architecturewith 16×16 and 32×32 column-bypassing multiplierscan attain up to 62.88% and 76.28% performance improvementcompared with the 16 × 16 and 32 × 32 other multipliers like column bypass and row bypass.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the department of Electronics and Communication Engineering of Alva's institute of engineering and technology for their continuous support during our study and research, for their patience, motivation and for sharing their vast knowledge. We will always remember their positive attitude and understanding which help us to shape our professional career. Their insightful directions helped us throughout our research work. With their supportand help we are able to prepare this paper. We shall bevery thankful for their warm support and guidance.

REFERENCES

[1] Y. Cao. (2013). Predictive Technology Model (PTM) andNBTI Model[Online]. Available http://www.eas.asu.edu/~ptm.

[2] S. Zafar et al., "A comparative study of NBTI and PBTI (chargetrapping) in SiO2/HfO2 stacks with FUSI, TiN, Re gates," in Proc.IEEE Symp. VLSI Technol. Dig. Tech. Papers, 2006, pp. 23–25.

[3] S. Zafar, A. Kumar, E. Gusev, and E. Cartier, "Threshold voltageinstabilities in high-k gate dielectric stacks," IEEE Trans. Device Mater.Rel., vol. 5, no. 1, pp. 45–64, Mar. 2005.

[4] H.-I. Yang, S.-C. Yang, W. Hwang, and C.-T. Chuang, "Impacts ofNBTI/PBTI on timing control circuits and degradation tolerant designin nanoscale CMOS SRAM," IEEE Trans. Circuit Syst., vol. 58, no. 6,pp. 1239–1251, Jun. 2011.

[5] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and mimization ofpMOS NBTI effect for robust naometer design," in Proc. ACM/IEEEDAC, Jun. 2004, pp. 1047–1052.

[6] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, "NBTI-awareflip-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 synthesisof digital circuits," in Proc. ACM/IEEE DAC, Jun. 2007, pp. 370–375.

[8] A. Calimera, E. Macii, and M. Poncino, "Design techniqures for NBTItolerantpower-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 reorderingagainst NBTI-induced performance degradation," in Proc. DATE, 2009, pp. 75–80.

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

[11] F. Najm, "Transition density, a stochastic measure of activityin digital circuits," in Proc. 28th Design Automation Conf.,pp. 644-649, June 1991.

[12] I. S. Abu-Khater, A. Bellaouar, and M. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," IEEE J. Solid-State Circuits, vol. 31, pp. 1535–1546, Oct. 1996

[13] C. R. Baugh and B. A.Wooley, "A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., vol. C-22, pp. 1045–1047, Dec. 1973.

[14] A. Wu, "High performance adder cell for low power pipelinedmultiplier," in Proc. IEEE Int. Symp. on Circuits and Systems, vol. 4 , pp. 57-60, May 1996.

[15] C. P. Lerouge, P. Girard, and J. Colardelle, "A fast 16-bit NMOS parallel multiplier," IEEE J. Solid-State Circuits, vol. SC-19, pp. 338–342, Mar. 1984.

[16] S. Mahant-Shetti, P. Balsara, and C. Lemonds, "High Performance Low Power Array Multiplier Using Temporal Tiling," IEEE Trans. VLSI Systems. pp. 121-124, Mar. 1999.

[17] A. A. Fayed and M. A. Bayoumi. "A novel architecture for low-power design of parallel multipliers," in Proc. IEEE Workshop on VLSI, pp. 149–154, 2001.