ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of 64 Bit Variable Latency Multiplier Using AHL

Aruna.G
M.Tech in VLSI System Design,
Indur Institute of Engineering and Technology,
JNTU, Hyderabad.

P.Renuka, M.Tech
Assistant Professor,
Indur Institute of Engineering and Technology,
JNTU, Hyderabad.

Abstract:

Digital multipliers are among the most critical arithmetic functional units. The overall performance of these systems depends on the throughput of the multiplier. Meanwhile, the negative bias temperature instability effect occurs when a pMOS transistor is under negative bias (Vgs = –Vdd), increasing the threshold voltage of the pMOS transistor, and reducing multiplier speed. A similar phenomenon, positive bias temperature instability, occurs when an nMOS transistor is under positive bias. 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 an aging-aware multiplier design with a novel adaptive hold logic (AHL) circuit. 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. Moreover, the proposed architecture can be applied to a column- or row-bypassing multiplier. The experimental results show that our proposed architecture with 16 ×16 , 32 ×32 and 64 ×64 column-bypassing multipliers and row bypassing multipliers.

Index Terms:

Adaptive hold logic (AHL), negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), reliable multiplier, variable latency.

I. INTRODUCTION:

DIGITAL multipliers are among the most critical arithmetic functional units in many applications, such as the Fourier transform, discrete cosine transforms, and digital filtering. The throughput of these applications depends on multipliers, and if the multipliers are too slow, the performance of entire circuits will be reduced Furthermore, negative bias temperature instability NBTI) occurs when a pMOS transistor is under negative bias (Vgs = –Vdd). In this situation, the interaction between inversion layer holes and hydrogen-passivated Si atoms breaks the Si–H bond generated during the oxidation process, generating H or H_2 molecules. When these molecules diffuse away, interface traps are left. The accumulated interface traps between silicon and the gate oxide interface result in increased threshold voltage (Vth), reducing the circuit switching speed. When the biased voltage is removed, the reverse reaction occurs, reducing the NBTI effect. However, the reverse reaction does not eliminate all the interface traps generated during the stress phase, and Vth is increased in the long term. Hence, it is important to design a reliable high-performance multiplier.

The corresponding effect on an nMOS transistor is positive bias temperature instability (PBTI), which occurs when an nMOS transistor is under positive bias. Compared with the NBTI effect, the PBTI effect is much smaller on oxide/polygate transistors, and therefore is usually ignored. However, for high-k/metal-gate nMOS transistors with significant charge trapping, the PBTI effect can no longer be ignored. In fact, it has been shown that the PBTI effect is more significant than the NBTI effect on 32-nm high-k/metal-gate processes.

A traditional method to mitigate the aging effect is overdesign including such things as guard-banding and gate oversizing; however, this approach can be very pessimistic and area and power inefficient. To avoid this problem, many NBTI-aware methodologies have been proposed. An NBTI-aware technology mapping technique was proposed in [7] to guarantee the performance of the circuit during its lifetime. In [8], an NBTI-aware sleep transistor was designed to reduce the aging effects on pMOS sleep-transistors, and the lifetime stability of the power-gated circuits under consideration was improved. Wu and Marculescu proposed a joint logic restructuring and pin reordering method, which is based on detecting functional symmetries and transistor stacking effects.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

They also proposed an NBTI optimization method that considered path sensitization [12]. In [10] and [11], dynamic voltage scaling and body-basing techniques were proposed to reduce power or extend circuit life. These techniques, however, require circuit modification or do not provide optimization of specific circuits. 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.

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 non-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, where the effect of process-variation is considered to increase the circuit yield. In addition, the critical paths are divided into two shorter paths that could be unequal and the clock cycle is set to the delay of the longer one. These research designs were able to reduce the timing waste of traditional circuits to improve performance, but they did not consider the aging effect and could not adjust themselves during the runtime. A variable-latency adder design that considers the aging effect. However, no variable-latency multiplier design that considers the aging effect and can adjust dynamically has been done.

A.Paper Contribution:

In this paper, we propose an aging-aware reliable multiplier design with a novel adaptive hold logic (AHL) circuit. The multiplier is based on the variable-latency technique and can adjust the AHL circuit to achieve reliable operation under the influence of NBTI and PBTI effects. To be specific, the contributions of this paper are summarized as follows:

1) novel variable-latency multiplier architecture with an AHL circuit. The AHL circuit can decide whether the input patterns require one or two cycles and can adjust the judging criteria to ensure that there is minimum performance degradation after considerable aging occurs;
2) comprehensive analysis and comparison of the multiplier's performance under different cycle periods to show the effectiveness of our proposed architecture;
3) an aging-aware reliable multiplier design method that is suitable for large multipliers. Although the experiment is performed in 16- and 32-bit multipliers, our proposed architecture can be easily extended to large designs;
4) the experimental results show that our proposed architecture with the 16×16 and 32×32 column-bypassing multipliers can attain up to 62.88% and 76.28% performance improvement compared with the 16 × 16 and 32 × 32 fixed-latency column-bypassing (FLCB) multipliers. In addition, our proposed architecture with 16 × 16 and 32 × 32 row-bypassing multipliers can achieve up to 80.17% and 69.40% performance improvement as compared with 16×16 and 32×32 fixed-latency row-bypassing multipliers.

The paper is organized as follows. Section II introduces the background of the column-bypassing multiplier row-bypassing multiplier, variable-latency design, and NBTI/PBTI models. Section III details the aging-aware variable-latency multiplier based on the column- or row-by-passing multiplier. The experimental setup and results are presented in Section IV. Section V concludes this paper.

II. PRELIMINARIES:

A. Column-Bypassing Multiplier:

A column-bypassing multiplier is an improvement on the normal array multiplier (AM). The AM is a fast parallel AM and is shown in Fig. 1.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The multiplier array consists of (n–1) rows of carry save adder (CSA), in which each row contains (n – 1) full adder (FA) cells. Each FA in the CSA array has two outputs: 1) the sum bit goes down and 2) the carry bit goes to the lower left FA. The last row is a ripple adder for carry propagation.The FAs in the AM are always active regardless of input states. In [22], 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. Fig. 1 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 sum bit is simply equal to the third bit, which is the sum output of its upper FA.

Hence, the FA is modified to add two tri-state 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 ai can also be used as the selector of the tristate gate to turn off the input path of the FA. If ai 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 ai is 1, the normal sum result is selected.

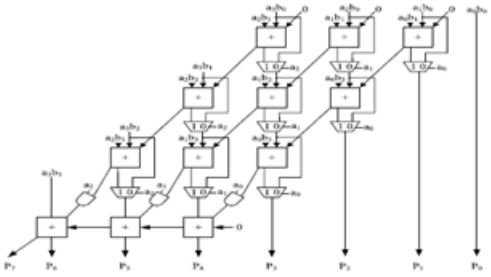

Fig. 1. 4 × 4 column-bypassing multiplier

B. Row-Bypassing Multiplier:

A low-power row-bypassing multiplier is also proposed to reduce the activity power of the AM. The operation of the low-power row-bypassing multiplier is similar to that of the low-power column-bypassing multiplier, but the selector of the multiplexers and the tristate gates use the multiplicator. Fig. 2 is a 4 × 4 row-bypassing multiplier. Each input is connected to an FA through a tristate gate. When the inputs are 11112 * 10012, the two inputs in the first and second rows are 0 for FAs.

Because b1 is 0, the multiplexers in the first row select aibo as the sum bit and select 0 as the carry bit. The inputs are bypassed to FAs in the second rows, and the tristate gates turn off the input paths to the FAs. Therefore, no switching activities occur in the first-row FAs; in return, power consumption is reduced. Similarly, because b2 is 0, no switching activities will occur in the second-row FAs. However, the FAs must be active in the third row because the b3 is not zero.

C. Variable-Latency Design:

The basic concept is to execute a shorter path using a shorter cycle and longer path using two cycles. Since most paths execute in a cycle period that is much smaller than the critical path delay, the variable-latency design has smaller

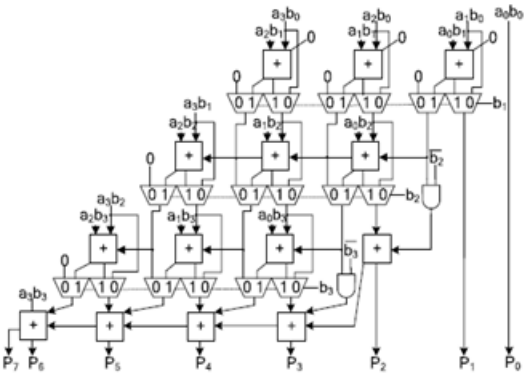

Fig. 2. 4 × 4 row-bypassing multiplier.

average latency. For example, Fig. 3 is an 8-bit variable-latency ripple carry adder (RCA). A8–A1, B8–B1 are 8-bit inputs, and S8–S1 are the outputs. Supposing the delay for each FA is one, and the maximum delay for the adder is 8.Through simulation, it can be determined that the possibility of the carry propagation delay being longer than 5 is low. Hence, the cycle period is set to 5, and hold logic is added to notify the system whether the adder can complete the operation within a cycle period.Fig. 4 also shows the hold logic that is used in this circuit. The function of the hold logic is (A4 XOR B4)(A5 XOR B5). If the output of the hold logic is 0, i.e., A4 = B4 or A5 = B5, either the fourth or the fifth adder will not produce a carryout. Hence, the maximum delay will be less than one cycle period. When the hold logic output is 1, this means that the input can activate paths longer than 5, so the hold logic notifies the system that the current operation requires two cycles to complete. Two cycles are sufficient for the longest path to complete (5 * 2 is larger than 8).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

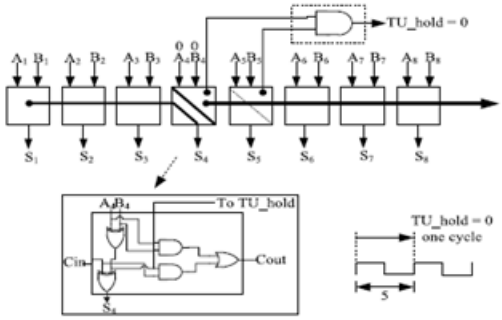

Fig. 3. 8-bit RCA with a hold logic circuit.

D. Aging Model:

As mentioned in Section I, the NBTI (PBTI) effect occurs when a pMOS (nMOS) transistor is under negative (positive) bias voltage, resulting in Vth drift. When the bias voltage is removed, the recovery process occurs, reducing the Vth drift.If a pMOS (nMOS) transistor is under constant stress, this is referred to as static NBTI (PBTI).

If both stress and recovery phases exist, it is referred to as dynamic NBTI (PBTI). The Vth drift of pMOS (nMOS) transistor due to the static NBTI (PBTI) effect can be described by dc reaction-diffusion (RD) framework. If transistors are under alternative stress and recovery phases, the dc RD model should be modified to an ac RD model.

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. 4 shows our proposed aging-aware multiplier architecture, which includes two m-bit inputs (m is a positive number), one 2m-bit output, one column- or row-bypassing multiplier, 2m 1-bit Razor flip-flops, and an AHL circuit.The inputs of the row-bypassing multiplier are the symbols in the parentheses.

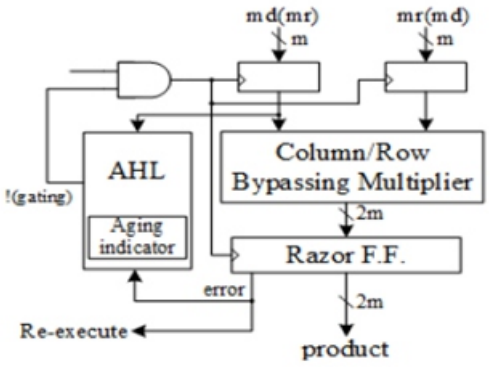

Fig. 4. Proposed architecture

In the proposed architecture, the column- and row-bypassing 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, as shown in Figs. 5 and 6. Therefore, using the number of zeros or ones as the judging criteria results in similar outcomes.

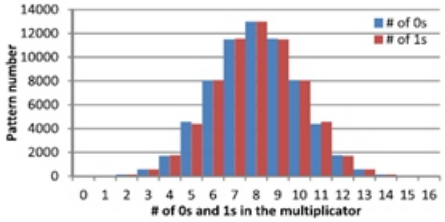

Fig. 5. Pattern number distribution based multiplicator.

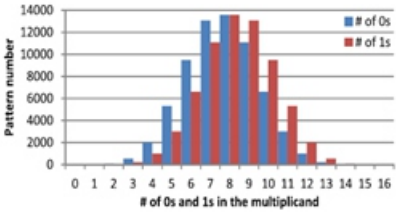

Fig. 6. Pattern number distribution based multiplicand.

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 r row-bypassing multiplier, the input signal of the AHL in the architecture with the column-bypassing multiplier is the multiplicand, whereas

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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. 7 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 re-execute the operation and notify the AHL circuit that an error has occurred. We use Razor 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 re-executed with two cycles. Although the re-execution may seem costly, the overall cost is low because the re-execution frequency is low.

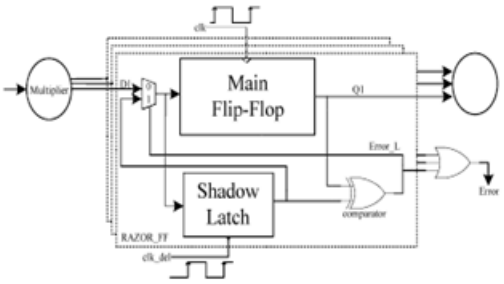

Fig. 7. Razor flip flops.

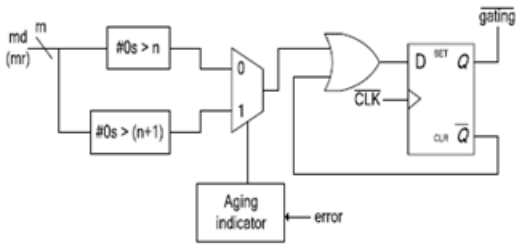

Fig. 8. Diagram of AHL

The AHL circuit is the key component in the aging-ware variable-latency multiplier. Fig. 8 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 means the 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 row-bypassing 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).

IV. RESULT ANALYSIS:

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 analyze the tradeoff between the percentage of one-cycle patterns and the cycle period. To achieve this, we analyze three scenarios for both 16×16 and 32×32 variable latency column-bypassing (VLCB) and variable-latency row-by-passing (VLRB) multipliers. We also compare the results with the AM, a FLCB multiplier, and a fixed-latency row-by-passing (FLRB) multiplier.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A.Design Summary Report:

row_bypass_multiplier_16 Project Status			
Project File:	AHL2.xise	Parser Errors:	No Errors
Module Name:	Top_Column_AHL16	Implementation State:	Synthesized
Target Device:	xc3s1600e-4fg320	• Errors:	
Product Version:	ISE 13.2	• Warnings:	
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	357	14752	2%
Number of Slice Flip Flops	128	29504	0%
Number of 4 input LUTs	615	29504	2%
Number of bonded IOBs	67	250	26%
Number of GCLKs	2	24	8%

Fig.9: design summary report for AHL using column bypass multiplier

B . RTL Schematic:

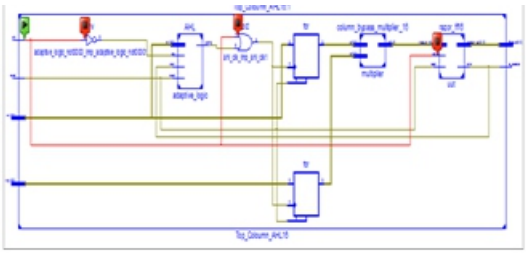

Fig.10.RTL Schematic for AHL using column bypass multiplier

C.Simulation Results:

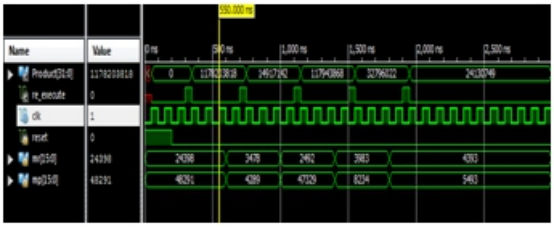

Fig.11.Simulation result for using column bypass multiplier.

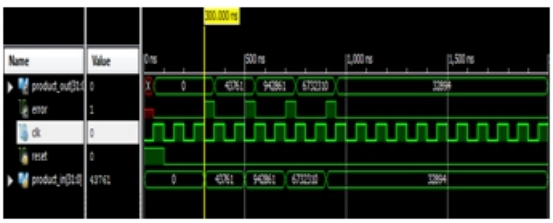

Fig.12.Simulation result for razor flip flop

V. SUMMARY AND CONCLUSION:

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 16x16, 32 x32 and 64 x64 multiplication. It will decrease the delay and improve the performance compared with previous designs.The efficiency of adiabatic circuit largely depends on the efficiency of the power-clock generation. The various techniques reported in literature use resonant drivers to generate power-clock supply. The energy efficiency of these circuits is very poor and therefore the efficiency of the entire adiabatic circuit diminishes. This is an upcoming area and has lot of research potential.

REFERENCES:

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

[2].S. Zafar et al., "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, 2006, pp. 23–25.

[3].S. Zafar, A. Kumar, E. Gusev, and E. Cartier, "Threshold voltage instabilities 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 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, Jun. 2011.

[5].R. Vattikonda, W. Wang, and Y. Cao, "Modeling and miimization of pMOS NBTI effect for robust naometer design," in Proc. ACM/IEEE DAC, Jun. 2004, pp. 1047–1052.

[6].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.

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

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

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