LPVM: Low-Power Variation-Mitigant Adder Architecture Using Carry Expedition

Alireza Namazi
Computer Engineering department
Tehran university
Tehran, Iran
a.namazi@ut.ac.ir

Meisam Abdollahi
Computer Engineering department
Tehran university
Tehran, Iran
meisam.abdolahi@ut.ac.ir

Abstract— **Addition is one of the most crucial operation in microprocessors which must be performed within a predefined deadline (critical path). Variation is a phenomenon which negatively affects the performance of this operation. This paper proposes a new Low-Power Variation-Mitigant (LPVM) adder design using intrinsic behavior of addition operation. The LPVM approach drastically decrease the probability of deadline violation in addition circuit. The basic idea of this paper is to expedite carry propagation in adder circuits for vulnerable inputs. The LPVM is an input oriented approach which adds a simple logic to the adder architecture that only affects the vulnerable inputs. This approach is applicable for all presented types of adders and improves all high level approaches which tend to overcome the variation issue. Results show that this approach decrease the percentage of violated RCA, CLA and CSA about 70.3%, 59.7% and 67.6% respectively. The LPVM approach not only reduces variation effects on the adder operation from the view point of performance but also it has a very negligible impact on the adder power consumption. The average power consumption overhead of the LPVM approach for RCA, CLA and CSA is about 7.3%, 2.1% and 3.1% for RCA, CLA and CSA, respectively.**

Keywords— Addition, Process Variation, Low Power

I. INTRODUCTION

Addition is one of the most useful and important arithmetic units [1] in microprocessors. Due to its critical role in almost all processing elements, there exist several architectures with the same functionality and different characteristics. Researchers have been investigated adders from different views such as performance[2], power consumption [3][4] and reliability [5]. Earliest investigations were focused on the performance [3] and power consumption improvement [1] of the adders.

In recent years, due to drastically decrease in feature sizes of digital system designs, process variation has become the major obstacle for system designers and the researchers has shown massive interest to address the variation effects with techniques from device to system level aspects [6]. Process variation is the concept of the deviation of manufactures component from nominal designed component. The variation impacts on the performance of a system and makes disorders and violations in their operation. All synchronous systems have a rigid timing constraints and all units must perform with predefined delay constraints. Variation is an issue which modifies the delay of operating units stochastically.

There exists many efforts in the literature to overcome the effects of the variation in digital circuits. Previous works can be divided into two major categories. The first one includes high level approaches which try to overcome the variation issue such as Razor Logic (RL) [7], Telescopic Unit (TU), PaceLine Approach [8] and DynaTune Approach (DA) [9]. The RU and TU both add new logics at the end of each pipeline stage and use their own error detection and correction techniques to overcome the variation issue. For example, RL [7] uses multiple copy of the output logic and compare them to find out if there exists any error in results. The PaceLine uses a novel duplication technique based on overclocking feature of processors. The DynaTune proposes a circuit level optimization technique to improve circuit behavior by probabilistic analysis of critical gates of the circuits. These techniques solve the variation problem globally for the worst case scenario which may drastically degrade performance.

All above mentioned techniques are considered to be general, hence our proposed approach is specially designed for the adder circuits considering their behavior. All circuit-level techniques should handle variation effects of their internal combinational segments, therefore having more tolerable components leads to better performance in these techniques. The LPVM imposes significantly less overhead to the system using intrinsic characteristics of the adder circuits. It can be used along with above mentioned high-level techniques and also can increases their efficiency.

The second category includes statistical approaches [10]. The [10] proposes a high level approach for presenting a variation-aware binding a component selection to maximize the yield. It uses rebinding and Statistical Static Timing Analysis (SSTA) to evaluate and maximize the performance. These techniques are also general and do not consider the intrinsic behavior of the circuit in their calculations. To the best of our knowledge, the proposed LPVM adder is the first variation mitigating approach which considers the behavior of the adder in order to remove the effects of variation on the operation of adder considering predefined clock period.

In this paper, we propose a novel low power architecture for adders to overcome the effects of process variation. This architecture can be used along with all high level approaches proposed in the literature and can increase their efficiency because it drastically decreases the adder malfunction probability. This reduces their overheads to the system. The basic idea behind this approach is to expedite carry propagation for vulnerable inputs which may violate the working clock period.

The rest of this paper is organized as follows. Section II describes the process variation. Section III describes the motivation of the paper. The proposed approach is presented in Section IV. Experimental results also presented in Section VII and finally Section VI concludes the paper.

II. PROCESS VARIATION

There exist many types and sources for variation in deep sub-micron digital circuits. Two major sources are known for variation: 1- manufacturing variations, 2- operation-induced variations [12]. This paper has concentrated on the first category.

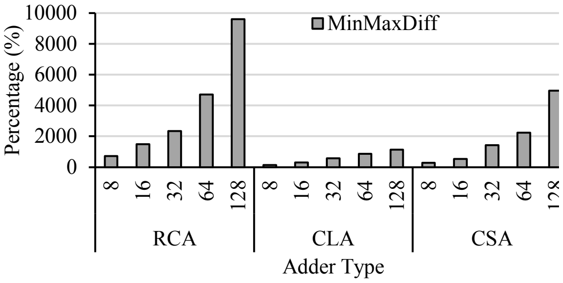

Fig. 1. The percentage of difference between Minimum and Maximum delay in adders for different bit widths based on input pairs

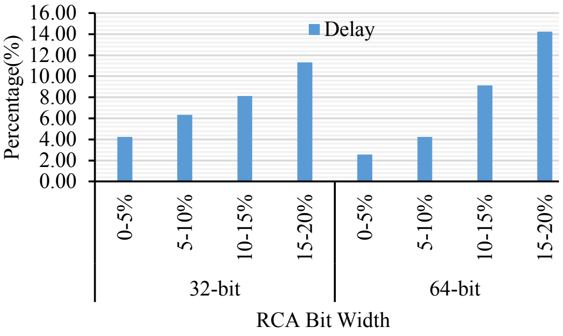

Fig. 2. Input pair percentages of RCA with 64 and 128 bit widths categorized base on their calculation time in compare of the worst case delay

Nanoscale IC manufacturing imperfections lead to variation in design parameters such as length (L), width (W), oxide thickness (T_{ox}) and threshold voltage(V_{th}) [13]. These fluctuations in design of Nano-scale (<<90nm) circuits results in many side-effects on the operation of designs in comparison with nominal design parameters. This paper focuses on the voltage threshold fluctuations because it is the most affecting parameter [14] and it changes the expected performance and power consumption of the systems. However, the proposed approach is applicable for all sources of variation.

III. Motivation

The addition performance completely depends on its inputs. Carry chain is consisted of multiple consecutive bit positions in an adder architecture which their carry-out depends on their input carry. Considering simulation results which is depicted in Fig. 1, it can be concluded that the adder can calculate the addition result of its inputs with the delay less than its critical path considering their inputs. Results show that inputs in Ripple Carry Adder (RCA), Carry Look-ahead (CLA) and Carry Select Adder (CSA) with 8, 16, 32, 64 and 128 bits, directly affect the calculation delay of the adder. The longest delay relates to inputs with longest carry chain. As adder circuits have different addition delay based on their inputs, different input pairs are categorized based on their calculation delay in comparison with the longest addition delay. Results depicted in Fig. 2 show that for 32-bit RCA adder delay of about 30% of input pairs is up to 20% less than the longest delay. Besides, about 4.1% of input pairs have about 10% of the longest delay.

Process variation results in delay changes in adder circuits. According to results depicted in Fig. 3, it can be concluded that variation effects completely depend on the architecture and input pair characteristics. This figure shows the minimum and maximum deviation percentage from nominal delay considering all possible input pairs in different adders with various bit-widths. Results are extracted using Hspice simulations and the variation of the threshold voltage (V_{th}) is selected as the main impacting parameter with maximum deviation range of 20% through Monte Carlo simulations. Simulated results are two folded as presented below:

- Input pairs have different calculation delay based on their carry propagation pattern.
- Variation changes the calculation delay of each input pair and also may change the worst case delay of the adder. These changes may violate the deadline which is predefined for the adder.

IV. Proposed Approach

Considering results gathered from input-based variation simulations on different types of adders, it is obvious that although variation impacts the delay characteristics of the adder, but it does not have influence on the result of all inputs. This paper proposes a simple and low overhead technique to overcome the effects of process variation on delay characteristics of each adder. The Low Power Variation Mitigating (LPVM) approach tries to overcome the process variation in adder circuits by simply taking care of inputs with calculation time near the critical path of nominal adder. The LPVM design approach schematic diagram is presented in Fig. 4. Simulation results showed that inputs with longer carry propagation chains are more susceptible to process variation. The LPVM design approach inserts simple combinational logics to the adder architecture in order to break the long carry propagation chain. The inserted blocks decrease the calculation time of the adder only for selected inputs and is called Carry Chain Breaker (CCB) block. The proposed approach has five steps which is described as follows:

A. *Carry chain Determination (Step 1)*

The first step is to determine Carry Chains (CC) of each input pair which is called $CC(S,F)$. Parameters S and F show start and finish bits of the CC, respectively. This input pair has three CCs. All carry chains of an input pair (A,B) reside in a set called Chain Set (a.k.a $CS(A,B)$).

B. *Susceptibility Analysis (Step 2)*

The second step divides input pairs into non-overlapped categories considering their calculation delay. Each input pair has a weight ($W(A,B)$) based on its carry propagation pattern. The weight of each input pair only depends to the weight of its longest CC and is calculated using (1).

$$W\left(CS(A,B)\right) = \underset{\forall cc \in CS}{MAX}\left(\left|F_{cc} - S_{cc} + 1\right|\right) \qquad (1)$$

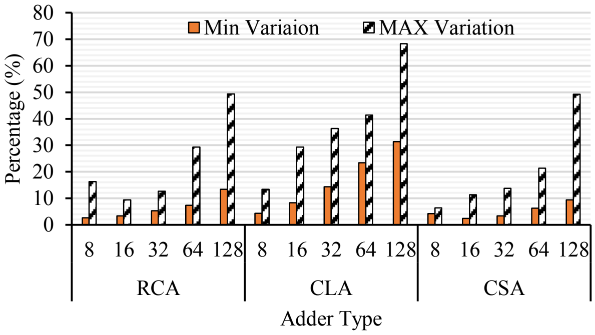

Fig. 3. Minimum and Maximum delay deviation percentage of calculation delay for input pairs in comparison with nominal delay

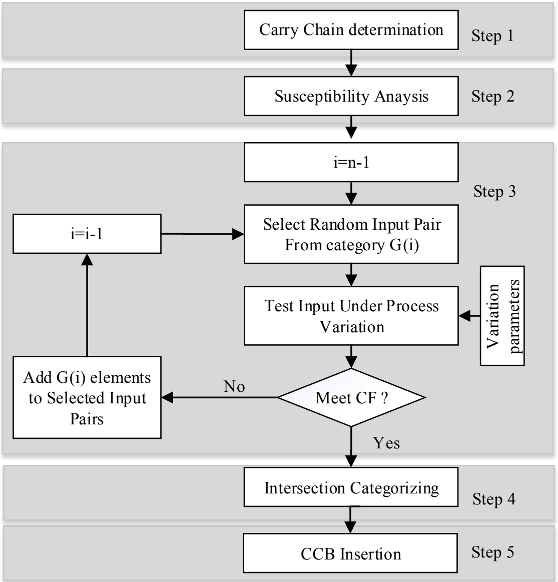
Carry Chain determination
Step 1
Susceptibility Anaysis
Step 2
i=n-1
Step 3
i=i-1
Select Random Input Pair From category G(i)
Test Input Under Process Variation
Variation parameters
Add G(i) elements to Selected Input Pairs
No
Meet CF ?
Yes
Intersection Categorizing
Step 4
CCB Insertion
Step 5

Fig. 4. LPVM approach

The weight of each input pair shows the calculation delay of the input pair in comparison with the longest pairs. All input pairs with the same weight reside in the same category. The category $G(i)$ contains all possible (existing) input pairs of the first step which their weight is equal to *i*. Step 1 and Step 2 are performed based on the *LPVM categorization* algorithm which is depicted in Algorithm 1.

C. Weight Selection (Step 3)

The third step is to find out vulnerable weighted categories. It is an iterative approach as depicted in Fig. 4. This step starts from the category with highest weight. It randomly selects an input pair. The selected input is examined on the adder architecture to find out its behavior under occurrence of process variation. In this step, the input is evaluated under different variation conditions based on designer parameters. Although, this paper selects threshold voltage as the main affecting parameter, other affecting parameters can be used to evaluate the variation effect. Afterwards, addition delay in all executed samples are gathered and checked if they have met the deadline or not. At this point, a new affecting parameter will be inserted by the designer which is called Certainty Factor (CF). This CF shows the acceptable percentage of violations in calculated results. For example if a designer selects 100% for the CF, this means that the G(i) is acceptable only if all its simulated samples from selected input pair meets the deadline. Otherwise the G(i) is not acceptable and should be added to the vulnerable inputs. When the algorithm reaches to a category which meets the defined CF, it does not evaluate the rest of categories because their calculation latency is absolutely less than evaluated category. By reaching this point, Step 3 iterations stops.

D. Intersection Categorizing (Step 4)

The third step is to divide CCs of selected input pairs into non-overlapped categories. All CCs which have any intersection with other CCs will be put in a same category. The fourth step is to find most covering subsets from CCs reside in each category. This step finds the most overlapping chain between different vulnerable input pairs. Therefore, the overhead of the CCB block insertion reduces.

E. CCB Block Insertion (Step 5)

The last step is inserting the CCB units into the adder architecture. This simple block breaks all or a segment of carry chains in susceptible input pairs to decrease or overcome the effects of variation on the adder circuit operation. The internal structure of the CCB for detecting continues propagation pattern is shown in Fig. 4. The depicted CCB block is designed for $CC(i, j)$. This block is consisted of parallel operating 2-input XOR gates which are connected to the adder inputs to detect the propagation pattern. This block connects the carry of the adder in *i*th bit to the carry in the *j*th position. When propagation pattern is detected, a 2-to-1 multiplexer selects C_i and replaces it with C_{j-1} carry. The CCB block has very low overhead because the XOR gates already resides in the basic architecture of the adders. The CCBs operate in parallel with the adder and does not increase its latency. The number of CCBs and their length completely depends on selected input vectors. The CCB architecture has overlapped with full adders. The XOR section of the CCB is generated in all adder circuits which reduces the overhead of the CCBs.

```
Algorithm 1: LPVM Categorization
Finds the best calculation threshold
Inputs : n (adder bit width)
For All Possible Input Pair
    $A = Convert\_binary(i);$                          }
    $B = Convert\_binary(j);$                          } Step1
    $CS = Carry\_Chain\_Extraction(A, B);$             }
    $CCL = Longest\_Chain(CS);$                        } Step2
    $Put\ pair\ (A, B)\ in\ G(i)\ where\ i = CCL$      }
End
```

V. Experimental Results

The experimental results consists of two different phases. The first phase uses the LEON3 processor (32-bit). The second phase relates to variation investigation of adder circuit. Simulations of this phase are performed with Hspice simulator and the technology size in considered as 32nm. Some applications of Mibench benchmark suit are executed on the LEON3 processor and the input entries of the adder unit is gathered and evaluated. The weight distribution of input pairs is depicted in Fig. 6. Results show that in real application executions, we may not have all possible input operands. Therefore exhaustive exploration of input pairs for the LPVM approach is no longer necessary.

Applying LPVM approach on the adder types presented in Section III shows that variation phenomena based on threshold voltage (in the range of 20%) different adder architectures demonstrate different operation violations. Fig. 7.a shows the percentage of operation violations in different adder architectures for various threshold variation ranges.

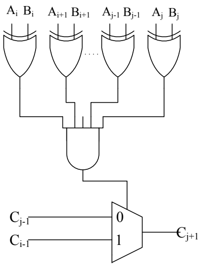

Fig. 5 General CCB Architecture

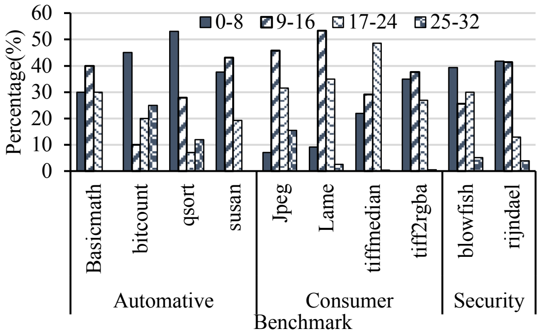

Fig. 6. Weight distribution of input pairs in 32-bit adder based on Mibench benchmarks

Results show that for selected benchmarks, the variation impacts the performance of the adder and results in deadline violation in addition operation. As the variation impact increase, the violated percentage of addition also increases. Results show that the CLA has the worst behavior in front of variation in comparison with other adder types. Applying the LPVM approach drastically reduces the deadline violations. This happens because CCB blocks which are inserted in the adder architecture break the carry chain of vulnerable input pairs.

The LPVM reduces the variation effect of the adder on the system behavior by decreasing the deadline violation percentage of the adder. Our proposed approach reduces the variation violation in all adder architectures. The effect of the proposed approach is different and relates to the adder architecture. According to results presented in Fig. 7.b, the LPVM decrease the violation percentage of RCA, CLA and CSA about 70.3%, 59.7% and 67.6%, respectively. In the fourth step of the LPVM approach, the intersection of carry chains of vulnerable input pairs are selected to decrsease the overhead of the proposed approach. Therefore, in this step, a systematic trade-off appears between power consumption overhead and variation mitigation. Results show that the LVMP approach acceptably reduces the variation effects and power consumption overhead.

Results show the LPVM approach reduces the violation percentage of the adder architecture in front of variation. It also imposes very low power dissipation and area overhead to the system. The average power consumption overhead of LPVM adders are respectively 7.3%, 2.1% and 3.1% for RCA, CLA and CSA architectures.

VI. Conclusion

The LVPM approach proposes a new approach to design variation tolerant adder circuits based on their intrinsic behavior. This approach reduces carry chain of vulnerable input pairs. This drastically reduces the effect of variation. The proposed approach can reduce the malfunction percentage of the adder up to 70%. The other advantage of this approach is that is imposes very low power consumption overhead to the adder (up to 7.3%).

VII. Future Work

The LPVM approach should be extended to design a variation mitigate ALU to overcome the variation with low power consumption overheads.

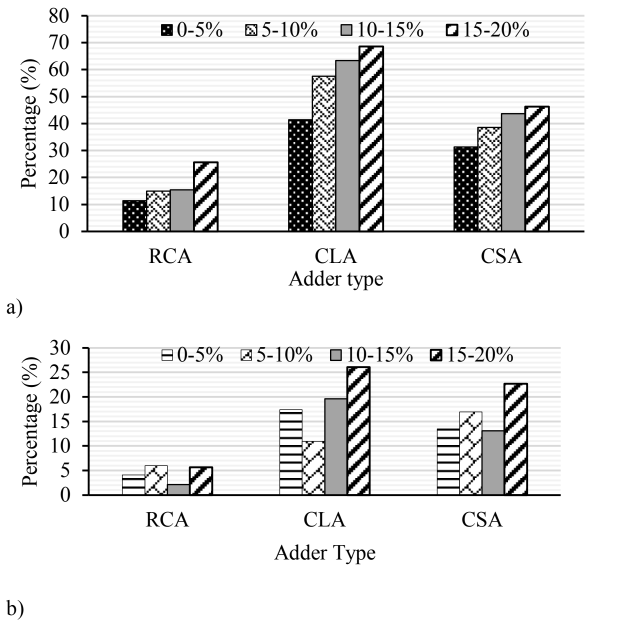

Fig. 7. Deadline violation percentage in different adder architecture a) basic architecture, b) LPVM architecture

VIII. References

[1] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy-Delay Optimization of 64-Bit Carry-Lookahead Adders With a 240 ps 90 nm CMOS Design Example," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, pp. 569–583, Feb. 2009.

[2] P. Varman and K. Mohanram, "High performance reliable variable latency carry select addition," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2012, pp. 1257–1262.

[3] P. Saxena, "Design of low power and high speed Carry Select Adder using Brent Kung adder," in *2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, 2015, pp. 1–6.

[4] V. Pudi and K. Sridharan, "Low Complexity Design of Ripple Carry and Brent–Kung Adders in QCA," *IEEE Trans. Nanotechnol.*, vol. 11, no. 1, pp. 105–119, Jan. 2012.

[5] S. Wei, "Residue checker using optimal signed-digit adder tree for error detection of arithmetic circuits," in *TENCON 2014 - 2014 IEEE Region 10 Conference*, 2014, pp. 1–6.

[6] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical Timing Analysis: From Basic Principles to State of the Art," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 27, no. 4, pp. 589–607, Apr. 2008.

[7] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, Nam Sung Kim, and K. Flautner, "Razor: circuit-level correction of timing errors for low-power operation," *IEEE Micro*, vol. 24, no. 6, pp. 10–20, Nov. 2004.

[8] Brian Greskamp and Josep Torrellas, "Paceline: Improving Single-Thread Performance in Nanoscale CMPs through Core Overclocking," in *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*, 2007, pp. 213–224.

[9] L. Wan and D. Chen, "DynaTune," in *Proceedings of the 2009 International Conference on Computer-Aided Design - ICCAD '09*, 2009, p. 172.

[10] G. Lucas, S. Cromar, and D. Chen, "FastYield: variation-aware, layout-driven simultaneous binding and module selection for performance yield optimization," pp. 61–66, Jan. 2009.

[11] J. A. Kumar and S. Vasudevan, "Variation-Conscious Formal Timing Verification in RTL," in *2011 24th Internatioal Conference on VLSI Design*, 2011, pp. 58–63.

[12] M. Kamal, A. Afzali-Kusha, S. Safari, and M. Pedram, "Impact of Process Variations on Speedup and Maximum Achievable Frequency of Extensible Processors," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 3, pp. 1–25, Apr. 2014.

[13] N. Banerjee, S. Chandra, S. Ghosh, S. Dey, A. Raghunathan, and K. Roy, "Coping with variations through system-level design," *Proc. 22nd Int. Conf. VLSI Des. - Held Jointly with 7th Int. Conf. Embed. Syst.*, pp. 581–586, 2009.

[14] M. Kamal, A. Afzali-Kusha, S. Safari, and M. Pedram, "An architecture-level approach for mitigating the impact of process variations on extensible processors," in *DATE '12 Proceedings of the Conference on Design, Automation and Test in Europe*, 2012, pp. 467–472.