



# Fault-tolerant carry look-ahead adder architectures robust to multiple simultaneous errors



Mojtaba Valinataj

School of Electrical and Computer Engineering, Babol University of Technology, Shariati Street, Babol, P.B. 484, Iran

## ARTICLE INFO

### Article history:

Received 24 April 2015

Received in revised form 23 August 2015

Accepted 25 August 2015

Available online 12 September 2015

### Keywords:

Carry look-ahead adder

Fault-tolerance

Error detection

Error correction

Triple modular redundancy

## ABSTRACT

Currently, the demand for reliable and high performance computing is increasing due to the enlarging susceptibility of computing circuits to different environmental effects, and the advent of diverse computation-based applications. Arithmetic operators, as the main building blocks of the processing units in computing systems, are exposed to different types of single or multiple errors incurred by different faults which can seriously damage the whole system. In this paper, a new approach is presented to achieve fault-tolerant carry look-ahead adder architectures, much more efficient than the conventional methods, with the characteristic of robustness against multiple simultaneous errors. The proposed method is based on revising the carry generation block to achieve multiple error correction capability, and utilizing a modified parity prediction-based method that in combination with the proposed error correction scheme leads to multiple error detection capability. Verified by simulations, analytical assessments show that, as well as correcting or detecting all single permanent or transient errors, multiple simultaneous errors in the new carry look-ahead adders are corrected or at least detected with a high probability independent of the number of errors. Apart from having more reliability against multiple errors, these adders require lower area overheads compared to the state of the art designs as well as conventional fault-tolerant methods.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

The processing elements inside different types of processors especially chip multi-processors (CMP) are being more vulnerable to the variety of external effects due to the shrinking transistor feature sizes, power supply voltages and the capacitance associated to the circuit nodes. Thus, an external effect such as strike of a high energy particle can simply make a transient pulse in the combinational logic as a Single Event Transient (SET) [1] in the systems used in both the ground and the space environments such as nuclear facilities and satellites. Moreover, many transient or soft errors might appear at the same time in the form of multiple scattered errors or single event multiple bit errors [2] in addition to permanent errors caused by permanent faults. Thus, the fault-tolerance characteristic is of great importance not only for safety- or mission-critical systems but also for the general computing systems.

Arithmetic operators as the main parts of the processing elements are also susceptible to different environmental effects. However, since multiple errors occurrence is highly probable because of multiple simultaneous faults, a different design approach should be taken to attain robust operators. So far, many fault-tolerant arithmetic operators have been designed to perform addition [3–9], multiplication [10–16] and division [17–19]. However, the incorporated techniques can be divided

into error detection and error correction techniques. Most designs that usually include self-checking designs use error detection techniques because of their lower overheads, and the others exploit error correction. Concurrent error detection (CED) techniques are mainly based on arithmetic residue codes, duplication with comparison, and the parity prediction-based schemes. Arithmetic residue codes cannot detect all single errors in a circuit in addition to requiring complex checkers [3]. A self-checking arithmetic logic unit (ALU) based on duplication with comparison is introduced in [5]. Another method described in [20] uses duplication with comparison in combination with parity check codes in order to detect and correct errors in combinational and sequential logic circuits. The methods proposed in [5,20] require more than twice as much area. Among different error handling techniques for arithmetic operations, the parity prediction-based schemes [3,4,10] are more popular since they lead to lower area overheads.

Different error correction techniques can also be used for arithmetic operations. A well-known technique is triple modular redundancy (TMR) that can be used to mask a permanent or transient error and produce the correct result. In this manner, a fault-tolerant component is achieved with concurrent error correction but with a high hardware cost due to the triplication of the primary component. Besides, TMR-based designs incur single point of failure due to the use of a majority voter. Another technique includes error correcting codes (ECC) as an information redundancy which has been used for arithmetic operations [21–23] in addition to memory elements. Other class of error correcting

E-mail address: [m.valinataj@nit.ac.ir](mailto:m.valinataj@nit.ac.ir).

designs includes time redundancy in which the main overhead is a higher delay in providing the results as well as power and area overheads. This method is usually used in combination with other techniques such as the designs proposed in [24–27]. In [24] an error correction approach is proposed in which, at first, error detection is performed by parity prediction and then, the corrected output is reached by inverting the inputs and obtaining the reversed output. While this method is suitable to deal with single stuck-at faults, it requires large overheads. Another method [25] utilized to handle the permanent faults in adders uses a parity-based technique to detect a fault and then, faulty digit is localized using a RE-computation with shifted operands (RESO), and at last, error correction is made. In [26] a similar method based on re-computing with shifted operands is presented, while its extension [27] utilizes some extra adders along with time redundancy. However, these methods require large overheads.

In this paper, we propose a new technique to design fault-tolerant carry look-ahead adders based on two main ideas: (1) revising the carry generation block to attain single error correction in each bit of the adder and as a result, multiple error correction capability in overall, and (2) exploiting the parity prediction as an error detection method combined with the error correction scheme to reach multiple error detection capability. Accordingly, the main goal is to achieve robustness against multiple simultaneous errors in the form of error correction or error detection. Thus, our aim is to detect multiple simultaneous errors if they cannot be corrected.

This paper includes the following contributions:

- Some modifications inside the carry generation block in the form of hardware redundancy are led to single error correction (error masking) in the carry generation logic of each bit of the adder. This way, at most  $n$  simultaneous errors can be masked or corrected inside an  $n$ -bit carry look-ahead adder.
- A parity prediction-based error detection method is used for other parts of the adder, combined with the proposed error correction technique. It will be analytically shown that the proposed combination of single error detection and single error correction techniques leads to multiple error detection capability. This analysis reveals that as well as having 100% overall error detection/correction against single errors in the main proposed architecture, all multiple errors are either corrected or detected with a high total probability which is around 90% for 64-bit adders. The proposed multiple error detection/correction is applicable to both transient and permanent errors. However, the correction of permanent errors is performed in the form of error masking.
- The proposed error correction technique uses a voter inside each bit of the carry look-ahead adder. However, unlike TMR-based methods, these voters are protected by the parity prediction scheme with no extra overhead, and as a result, single point of failure is prevented.

The rest of the paper is organized as follows. In Section 2, the related works and in Section 3 the preliminaries about the carry look-ahead adder and parity prediction scheme are described. In Section 4, the proposed carry look-ahead adder architectures robust to multiple simultaneous errors and their reliability analysis are explained. The area overhead and reliability evaluations of the proposed architectures are discussed in Section 5. Finally, some conclusions are drawn in Section 6.

## 2. Related works

So far many fault-tolerant arithmetic operators have been designed in which a variety of designs is dedicated to error detection/correction in the carry look-ahead adder, one of the high speed adders. Some works related to the fault-tolerant carry look-ahead adders are based on the reversible logic such as [28–30] in order to enhance the design of quantum computers and to perform complex computing such as quantum computing and nano computing. In [28] a 2-bit fault-tolerant

carry look-ahead adder is proposed by utilizing new reversible gates. It achieves complete single error detection with around 60% cost overhead just in the adder circuit compared to the non-fault-tolerant design. In [29] another design is presented based on a new low-cost fault-tolerant full adder to reduce the overall cost. In [30] a fault-tolerant carry look-ahead adder is proposed based on a new arrangement of the existing reversible gates. This adder has reached lower cost and better design parameters compared to the design proposed in [28]. It is worth mentioning that the fault-tolerance capability of the reversible adders proposed in [28–30] is based on utilizing parity-preserving reversible gates. In each parity-preserving reversible gate, the inputs' parity equals the outputs' parity, and this way it has single error detection capability. Thus, the obtained parity-preserving adders have this property, as well. However, these adders only have error detection capability and cannot correct any error. Besides, they require extra overhead to implement the parity computation and parity comparison circuits that has not been reported in [28–30].

In the irreversible or ordinary logic, one of the first carry look-ahead adders adopting error detection is proposed in [31] based on arithmetic residue codes, which has the problems stated before. Some error detection techniques used for carry look-ahead adders, utilize the parity prediction scheme or duplication with comparison method along with a two-rail checker to achieve an efficient design [3,32]. In [3] a new self-checking architecture for carry look-ahead and ripple carry adders is proposed in which two-rail checkers are used to detect errors in the internal carries and the parity prediction is used to detect errors in the output. In [32] the parity prediction is used to detect errors in input operands while two-rail checkers are used for the output error detection in which output sum bits are duplicated based on duplication with comparison method. In [33] another parity prediction-based self-checking adder structure is proposed in which a portion of the design is based on the carry look-ahead logic. These designs need less hardware cost compared to the basic duplication with comparison method. However, all the adders in [3,32,33] are augmented to detect single errors and cannot correct any number of errors.

In [34] a carry look-ahead adder architecture is presented with the aim of fault repair after fault detection and identification. This adder corrects all single faults. However, despite the fact that it can correct two concurrent faults with a high probability, a high amount of hardware overhead is required in some cases. In addition, due to the fact that this adder architecture uses RE-computation with rotated operands (RERO) as a type of time redundancy, it incurs a high delay overhead. In [35] a carry look-ahead adder with a multiple error detection/correction scheme is proposed. However, the time needed for output preparation increases when the number of errors increases. In [36] two new fault-tolerant carry look-ahead adder designs are proposed in which one is based on a modified duplication with comparison to detect permanent or transient single errors and another is based on a modified TMR specifically designed for the carry look-ahead adder to correct single errors. Although these adders have less overhead compared to their counterparts, they use unprotected checkers and voters thus incur single point of failure. In [37] another method is proposed to handle permanent faults in a 32-bit ALU including carry look-ahead adder. This method detects and isolates at most two erroneous 8-bit sub-blocks, online, in such a way that the main ALU can still work after deactivating two sub-blocks. However, this method is only useful for permanent faults. In [38] a carry look-ahead adder with multiple error detection/correction is proposed as an extension of [35]. However, in this paper, we propose different architectures with lower area overhead and higher fault-tolerance capability.

## 3. Preliminaries

### 3.1. Carry look-ahead adders

The main idea behind the carry look-ahead addition is an attempt to generate all internal carries in parallel in order to eliminate the carry

Download English Version:

<https://daneshyari.com/en/article/544712>

Download Persian Version:

<https://daneshyari.com/article/544712>

[Daneshyari.com](https://daneshyari.com)