



# Optimized structures of hybrid ripple carry and hierarchical carry lookahead adders



Atef Ibrahim<sup>a,b,c,\*</sup>, Fayez Gebali<sup>b</sup>

<sup>a</sup> Sattam Bin AbdulAziz University, Kharj, Saudi Arabia

<sup>b</sup> ECE Department, University of Victoria, Victoria, BC, Canada

<sup>c</sup> Electronics Research Institute, Cairo, Egypt

## ARTICLE INFO

### Article history:

Received 3 December 2014

Received in revised form

12 June 2015

Accepted 14 June 2015

Available online 10 July 2015

### Keywords:

Hybrid adders

Hierarchical carry lookahead adders

Fast adders

Optimized adder structures

ASIC implementation

Digital VLSI design

## ABSTRACT

This paper proposes improved structures for fast adders that include carry lookahead (CLA) and hierarchical carry lookahead (HCLA). Also, it proposes optimized novel structures of hybrid ripple carry/hierarchical carry lookahead (RCA/HCLA) adders. A general methodology is presented for constructing  $M$ -bit hierarchical carry lookahead adders using  $n$ -bit modules. The only restriction on the values of  $M$  or  $n$  is  $n \leq M$ . Two algorithms are developed to efficiently construct hierarchical carry lookahead adders for the case when  $M$  is not an integer power or an integer multiple of  $n$ . The improved hierarchical levels of carry lookahead adders are integrated with the ripple carry adder to construct the novel hybrid RCA/HCLA adders. Area and time complexities of the resulting designs are reported for different values of radix  $n$  and the practical values of 32 and 64 bits of  $M$ . An ASIC implementation of the proposed structures and previously published recent designs shows that one of the proposed hybrid RCA/HCLA adders achieves 28.2–77.7% reduction in area–delay product and 40.5–75.8% reduction in energy, for  $M=64$  and  $n=8$ , over the different compared adder designs.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

Design of adders showing high performance in speed of addition, power consumption and silicon area is important for many applications such as advanced digital signal processors, crypto-processors and embedded wireless mobile devices that require strong encryption to provide the needed security for the users. In traditional very large scale integration (VLSI) design, the system designer must take into consideration the design area and power consumption [1,2]. Managing the power in a VLSI chip does not only target power reduction, but also ensures that no hotspots are present within the die [3].

Wide adders are a piece of the most crucial power-density processor modules, making thermal hotspots and severe temperature inclinations [4–6]. The existence of various arithmetic logic units (ALUs) in current superscalar processors [7,8] and different execution cores on the same chip [8–10] further worsen the problem, affecting circuit reliability and expanding cooling costs. At the same time, wide adders are also crucial for performance, and come into view inside the ALUs and floating point units (FPUs) of microprocessor datapaths. In a perfect world, a datapath adder would realize the highest performance using the minimal amount of power and has a little

layout footprint so as to reduce interconnect delays in the core [6]. These conflicting necessities constitute a challenging issue in choosing the best adder architecture and circuit implementation.

The literature gives a variety of solutions for optimizing adders using different techniques such as carry-select adders [11–13], carry save adders [14], carry lookahead adders [15–18], hybrid between carry-select and carry-lookahead adders [19–24], carry skip [25,26], and conditional-sum adders [27,28].

The main contribution in this paper is constructing  $M$ -bit hybrid ripple-carry and hierarchical carry lookahead (RCA/HCLA) adder structures using arbitrary choice of  $n$ -bit HCLA modules. Two crucial differences exist between the proposed hierarchical structures and the structure of HCLA. The first difference is that the  $n$ -bit HCLA modules at the first level of the proposed hierarchical structures are modified so that they produce the propagate and generate signals only that there is no need to generate the carry signals at this level. The second difference is that the  $n$ -bit HCLA module at the top-most hierarchy level generate the carry signals only that there is no need to produce the propagate and generate signals at this level. At the same time, the delay in the RCA section of these adder structures is only for  $n$  bits RCA module since the carry-in signal for each  $n$ -bit RCA module is obtained directly from the  $n$ -bit HCLA module in second level of the hierarchy. Therefore, these new structures achieve a significant reduction in area and power with a minimal delay penalty since it uses  $n$ -bit RCA modules.

\* Corresponding author at: Sattam Bin AbdulAziz University, Kharj, Saudi Arabia.

E-mail addresses: [atef@ece.uvic.ca](mailto:atef@ece.uvic.ca), [attif\\_ali2002@yahoo.com](mailto:attif_ali2002@yahoo.com) (A. Ibrahim), [fayez@uvic.ca](mailto:fayez@uvic.ca) (F. Gebali).

This paper is organized as follows. Section 2 explains how to model system performance to have an idea on the impact of different parameters on the different adder structures. Section 3 presents the basic modules of the RCA, CLA (carry lookahead adder) and HCLA adders. Section 4 describes constructing efficient  $M$ -bit CLA for  $M$  is not an integer multiple of  $n$ . Section 5 describes constructing efficient  $M$ -bit HCLA for  $M$  is not an integer multiple of  $n$ . Section 6 describes the proposed  $M$ -bit hybrid ripple-carry and hierarchical carry lookahead (RCA/HCLA) adder structures using arbitrary choice of  $n$ -bit HCLA modules. Section 7 shows the complexity analysis results for the different types of adders investigated. Section 8 compares the ASIC implementation results of the different types of adders investigated and previously reported efficient adders. Finally Section 9 concludes the paper.

## 2. Performance modeling

Exploring the optimal VLSI design requires using either performance estimation using back annotation following full place and route using a specific technology. This approach gives realistic estimates but lacks any insight on the main parameters affecting system performance. For our case the main parameters are  $M$ ,  $n$  and rough estimates of the elementary gate technology parameters. Extensive implementations using different parameter settings are necessary to gain any insight. Simple analytic models for system performance give an approximate performance estimates but at least identify the important parameters that impact system performance. The assumptions employed in the next subsection are good for implementing simplified models for area and speed but they will not be realistic for modeling power. This is due to the dynamic power component of CMOS gates that depends on the switching activity factor which is a strong function in the signal statistics, inter-signal correlations, and glitching transitions. Therefore, we relayed only on the actual power simulations to study power consumption of the proposed designs.

### 2.1. Modeling area

In order to study the order-of magnitude complexity of the proposed designs, we used the standard layout results discussed in basic VLSI technology such as in [29]. We make the following assumptions for our numerical results:

1. Standard static CMOS technology is used.
2. All logic modules will be implemented in terms of NAND gates. According to the analysis given in [30], NAND gate has better area and delay over NOR gate for static CMOS technology.
3. Large-fan-in gates with number of inputs larger than 2 are implemented using basic 2-input NAND gates in order to limit power consumption and maintain symmetric rise and fall times with reasonable transistor sizing.
4. We normalize all areas relative to the area of a 2-input NAND gate.
5. We shall ignore the inverter areas since their number is very small relative to the total number of other gates. This assumption was based on the observation that the basic module structures of

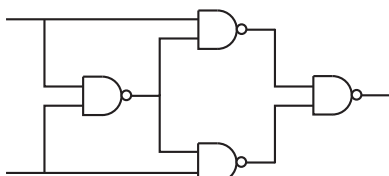


Fig. 1. Constructing 2-input XOR gate using 2-input NAND gates [29].

adders have AND-gate level followed by OR-gate level (AND–OR structure). We converted the AND–OR structure of these modules to the corresponding NAND–NAND structure. Therefore, when implementing  $n$ -input NAND gates with 2-input ones, the inverters required by the  $n$ -input NAND-gates of the first level will be offset by the inverters required by the  $n$ -input NAND gates of the second level. Also, the remaining very few inverters will not be on the critical path of the modules and thus they will not have any effect on the delay. For these reasons, inverters have been ignored from most of the adder architectures mentioned in this paper.

Based on the above assumptions, the normalized area of an  $i$ -input NAND gate is give by  $A_i = i - 1$  normalized relative to the area of a 2-input NAND gate. A 2-input XOR gate can be implemented using four 2-input NAND gates, as shown in Fig. 1 [29]. Consequently the normalized 2-input XOR gate area is  $A_X = 4$ .

### 2.2. Modeling delay

Similar to gate areas, we normalize a gate delay relative to the delay of a 2-input NAND gate driving a similar minimum-area 2-input NAND gate. The normalized delay of an  $i$ -input NAND gate is given by  $T_i = \lceil \log_2 i \rceil$  normalized relative to the delay of a 2-input NAND gate. According to Fig. 1, a 2-input XOR gate would have a normalized delay of  $T_X = 3$ .

## 3. RCA, CLA and HCLA basic modules

We provide in this section detailed analysis of the basic modules used in constructing  $M$ -bit RCA, CLA and HCLA adders.

### 3.1. Ripple-carry adder module (RCA)

We start this section by mentioning the standard 1-bit ripple-carry adder (RCA) module construction shown in Fig. 2. When two  $M$ -bit numbers are to be added, the addend  $a$  and augend  $b$  are supplied to the  $M$ -bit RCA. The RCA is composed of two blocks: the bit-parallel  $P$  &  $G$  block and the bit-serial  $S$  &  $C$  block. The  $P$  &  $G$  block is used extensively in CLA as well as HCLA structures, as will be discussed below. On the other hand, the  $S$  &  $C$  block operates on the input data serially. Each sum output bit  $s_i$  is produced after the carry out bit  $c_{i-1}$  of the previous stage is produced.

With reference to Fig. 2, the normalized area of the RCA module is estimated as  $2A_X + 3$ , where  $A_X$  is the normalized area of the 2-input XOR gate. The normalized delay of the RCA module is taken as the delay of the carry-out signal which is estimated as  $T_X + 2$ , where  $T_X$  is the normalized delay of the 2-input XOR gate. This estimate takes into account that an RCA delay is bound by the carry propagate signal, as opposed to the sum delay.

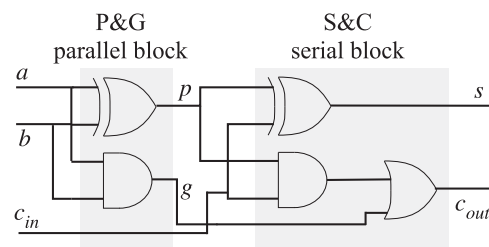


Fig. 2. A ripple-carry adder (RCA) is composed of a parallel  $P$  &  $G$  part and a serial  $S$  &  $C$  part.

Download English Version:

<https://daneshyari.com/en/article/543168>

Download Persian Version:

<https://daneshyari.com/article/543168>

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