Computers and Electrical Engineering 63 (2017) 18–29

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

An optimized reconfigurable architecture for hardware implementation of decimal arithmetic☆

Samaneh Emami, Mehdi Sedighi*

Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 19 February 2017
Revised 20 August 2017
Accepted 21 August 2017

Keywords:
Computer arithmetic
Decimal arithmetic
Coarse-grain architecture
Reconfigurable hardware

ABSTRACT

Hardware and software implementations of decimal arithmetic have resurfaced in recent years to overcome the limitations of binary arithmetic. Traditionally, decimal arithmetic units have been designed as application-specific hardware modules. But there is an emerging trend towards the design and implementation of decimal arithmetic operations on reconfigurable structures. This paper contributes to this trend by proposing a reconfigurable architecture, namely DARA, for high performance implementation of decimal arithmetic operations. Some basic decimal arithmetic operations were implemented on DARA and synthesized subsequently. The results show that DARA has a delay overhead of 26% and area overhead of 54% on average compared to an ASIC implementation of the same operations. At the same time, if those basic operations had been implemented on a modern commercial FPGA, DARA would have outperformed the commercial device in terms of delay and area by a factor of almost 4 and 9, respectively.

1. Introduction

Humans have traditionally used decimal arithmetic because of having ten fingers. Early computers such as ENIAC [1] and IBM 650 [2] used decimal arithmetic as well. Later generations of computers adopted binary arithmetic due to the relative simplicity and high performance of binary circuits. But decimal arithmetic has seen a renewed interest in recent years as researchers have focused on more efficient hardware implementation of decimal arithmetic in computers.

One of the primary reasons for using decimal arithmetic is to overcome a major problem of binary arithmetic, namely its inaccuracy in representing some non-integer decimal numbers such as 0.1 that can cause unacceptable errors in financial and commercial applications. For instance, it has been reported that in a large telephone billing system, using binary arithmetic instead of decimal can result in an estimated annual loss of up to five million dollars [3]. The importance of decimal arithmetic in recent years has increased to the extent that the IEEE 754-2008 standard for floating-point arithmetic [4] includes specifications for Decimal Floating-Point (DFP) arithmetic. In this standard, two decimal number formats for both software and hardware implementations of decimal arithmetic are presented:

- Binary Integer Decimal (BID) is proposed for software implementation of decimal arithmetic and is used in IBM decFloats modules [5] and Intel DFP Math Library [6]. These implementations can eliminate the inaccurate representation errors, but they are usually slow and inefficient [7].

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. H. Sarbazi-Azad.

* Corresponding author.
E-mail address: msedighi@aut.ac.ir (M. Sedighi).

http://dx.doi.org/10.1016/j.compeleceng.2017.08.018
0045-7906/

- Densely Packed Decimal (DPD) is recommended for hardware implementation of decimal arithmetic and is used in machines that have dedicated decimal hardware units, like IBM Power and System z processors [8,9] and Fujitsu SPARC processor family [10].

Since the transistor density on semiconductor chips has increased dramatically in recent years, decimal hardware implementations typically do not incur a significant cost while providing superior accuracy and higher performance [7]. As a result, decimal hardware solutions are gaining prominence in both academia and industry.

Various hardware implementations have been proposed to support decimal arithmetic. These implementations are usually special-purpose hardware units that their usage is limited to the specific application they have been designed for. Nevertheless, there are reports of using reconfigurable devices, such as Field-Programmable Gate Arrays (FPGAs), for decimal arithmetic as well [11–13]. The motivation for these works is often the remarkable flexibility and high performance of today's modern FPGAs along with the inclusion of various embedded binary arithmetic units (such as embedded multipliers) on these devices [14]. As a result, rigid Application-Specific Integrated Circuit (ASIC) realizations may no longer be very attractive choices. However, when it comes to decimal arithmetic, the current FPGA technology fails to provide efficient solutions for decimal arithmetic and very long integers used in financial or accounting applications [11]. Furthermore, the architectural characteristics of existing FPGAs make it difficult to implement many familiar decimal arithmetic algorithms, such as decimal tree adders, on them [12]. Therefore, there is a general trend toward flexible designs among hardware and system designers. This trend is mainly driven by the ever-changing market demands and design specifications. Therefore, to cope with these circumstances, designers often try to embed some form of flexibility or reconfigurability in their designs. As a result, it would be plausible to envisage a reconfigurable architecture designed and optimized for high performance realization of decimal arithmetic operations. Such architecture would provide system designers with a considerable flexibility without a significant compromise on the performance. This paper presents an architecture that is designed to achieve this goal.

The remainder of this paper is organized as follows: Section 2 presents an overview of various hardware implementations of decimal arithmetic operations. Section 3 discusses the proposed reconfigurable architectures in detail. Analysis and synthesis results are provided in Section 4. The paper is concluded in Section 5.

2. Previous works

There are numerous studies on various implementations of basic arithmetic operations such as addition, subtraction, and multiplication. Among these studies, some focus on decimal hardware implementations [15–20]. These works often present application-specific designs with good performance but no flexibility.

Nevertheless, some papers that present decimal arithmetic units with some degrees of reconfigurability can be found in the literature [21–23]. A close look at these papers shows that the term "reconfigurability" seems to have different definitions among them. In some of the works that involve hardware implementation of decimal arithmetic units, reconfigurability refers to performing different computations on the same inputs, or execution of a single operation on inputs with different sizes or different representation formats. For example, [21] and [22] have proposed combined binary/decimal adder/subtractor units. These units can operate on different input formats, but not different sizes. On the other hand, [23] has proposed a reconfigurable parallel prefix Ling adder that can perform addition on operands with different input sizes (16, 8 or 4-bit operands).

Another category of works has focused on efficient implementation of decimal operations on general-purpose reconfigurable devices such as FPGAs. Nannarelli has studied FPGA-based acceleration of decimal arithmetic operations [11]. This study shows that applications requiring decimal operations can be expedited by an arithmetic processor implemented on an FPGA board that connects to a computer. This processor ran a telephone billing computation that included 3 additions and 3 multiplications. The author achieved a speed-up of around 10 over its execution on the CPU of the host computer. This achievement is mainly due to more flexibility in the FPGA implementation compared to an ASIC design. Vázquez and Dinechin [12] have also presented a new method for fast implementation of multi-operand decimal addition in current FPGAs. This method is based on pre- and post-corrections of the binary sum. With the pre-corrections, the hexadecimal carries correctly serve as decimal carries, which provides the opportunity to utilize built-in carry chain in FPGAs. As a result, the authors have reported that their implementation on a Virtex-6 FPGA device halves the area and latency of previous reconfigurable implementations [13].

Overall, there is a whole body of evidence in the literature that moving arithmetic units to reconfigurable structures is a promising trend [11–13,21–24]. However, as mentioned earlier, the existing general purpose reconfigurable devices (such as common commercial FPGAs) have some architectural limitations (e.g., lack of optimized resources) that lower their performance in arithmetic applications. For example, [12] has shown that the common implementation of decimal addition on a Virtex-6 FPGA incurs a significant delay.

On the other hand, in other realms, there have been efforts to design a fully reconfigurable architecture specialized for efficient implementation of specific applications in that realm. For instance, an array of reconfigurable hardware units, called Coarse Grain Reconfigurable Architectures (CGRAs), has been devised for parallel and distributed digital signal processing applications [24]. This architecture has been optimized for wireless systems computations by taking advantage of CGRAs to gain a higher performance than existing FPGAs.