Computers and Electrical Engineering 50 (2016) 39–53

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Conditional speculative mixed decimal/binary adders via binary-coded-chiliad encoding [☆]

M. Dorrigiv, G. Jaberipur[*]

Department of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran

ARTICLE INFO

Article history:
Received 30 May 2015
Revised 3 December 2015
Accepted 3 December 2015
Available online 31 December 2015

Keywords:
Decimal computer arithmetic
Speculative decimal addition
Mixed decimal/binary adder
Binary coded chiliad encoding
Densely packed decimal encoding

ABSTRACT

Decimal arithmetic circuits, based on IEEE-754-2008 standard, commonly use 10-bit densely-packed-decimal (DPD) encoding of three binary-coded-decimal (BCD) digits. Binary-coded-chiliad (BCC) encoding, as storage (arithmetic) efficient as DPD (BCD), equivalently packs three BCD digits. No unpacking/packing to/from BCD (entailing extra delay/power) per each arithmetic operation (required in case of DPD), are necessary for BCC. Therefore, while abiding to DPD standard, we are motivated to design decimal arithmetic operators that accept BCC operands and produce BCC results. As such, DPD data from memory or input devices are converted to BCC, manipulated in BCC and stored in the BCC register file, during multi-operation decimal computations, and converted back to DPD only on reporting results to memory or output devices. In this paper, following a previous simple mixed BCC/binary adder, we design and synthesize more efficient ones, and compare them with previous relevant BCD and BCC adders to show advantages in area, and power.

1. Introduction

Research on hardware realization of decimal arithmetic has been revived in the past two decades (e.g., [1–7]), in response to new demands for high performance decimal computations. Some recently commercialized general-purpose digital processors include hardware decimal units and decimal arithmetic instruction subsets. For example, IBM included a decimal floating-point hardware unit in the chips of z-196 [8] servers and has doubled these units in the newest z13 servers [1]. Also, Fujitsu announced the new Sparc64 X processor which include an accelerator, called “software on chip (SWoC)”, to speed up operations in cryptography and decimal calculations [5]. The IEEE 754-2008 standard for floating-point decimal arithmetic [9] includes two encodings of decimal numbers. The one that is commonly used for storage of decimal operands and hardware realization of decimal arithmetic operators (e.g., [1,8,10]) is called densely packed decimal (DPD), which packs each three binary coded decimal (BCD) digits as a 10-bit DPD encoding. For example, the 16-digit BCD significand of an IEEE single precision radix-10 floating-point number is packed as five DPD declets, while the most significant BCD digit is encoded together with the exponent in the combination field of the *decimal64* representation. Such representation cannot be directly carried out by decimal arithmetic operators, and is subject to unpacking to 16 BCD digits and 10-bit exponent. The IEEE 754-2008 discussions [11] include a proposal for 10-bit radix-1000 packing of three BCD digits in $[0, 999]$ (e.g., $B\,C\,D = 100B + 10C + D$), which was named as binary coded millennium. This encoding, which was not approved, is as storage efficient as DPD and can be as arithmetic efficient as BCD, since it can be directly manipulated by radix-1000 arithmetic operators, with no prior unpacking to three BCD digits and post packing, as

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. C-H Hsu.
[*] Corresponding author. Tel.: +98 2129904165; fax: +98 2122431804.
E-mail address: jaberipur@sbu.ac.ir (G. Jaberipur).

http://dx.doi.org/10.1016/j.compeleceng.2015.12.002
0045-7906/© 2015 Elsevier Ltd. All rights reserved.

i	6	5	4	3	2	1	0
X_i		0 0 1 1	0 1 1 0	0 1 0 1	0 0 0 0	0 0 0 1	0 0 1 0
Y_i		0 0 1 0	0 1 0 0	0 1 1 1	1 0 0 1	1 0 0 0	0 1 1 0
$X_i + 6$		1 0 0 1	1 1 0 0	1 0 1 1	0 1 1 0	0 1 1 1	1 0 0 0
Y_i		0 0 1 0	0 1 0 0	0 1 1 1	1 0 0 1	1 0 0 0	0 1 1 0
c_i	0	1	1	0	0	0	0
$(X_i + 6) + Y_i + c_i$		1 1 0 0	0 0 0 1	0 0 1 0	1 1 1 1	1 1 1 1	1 1 1 0
$+\lvert 10\overline{c_{i+1}}\rvert_{16}$		1 0 1 0	0 0 0 0	0 0 0 0	1 0 1 0	1 0 1 0	1 0 1 0
S_i		0 1 1 0	0 0 0 1	0 0 1 0	1 0 0 1	1 0 0 1	1 0 0 0

Fig. 1. Speculative BCD addition of $365,012 + 247,986 = 612,998$ via 24-bit binary adder.

is required in the DPD case. However, this binary-coded-chiliad[1] encoding (BCC as is called in [12]) has been rarely used for realization of decimal arithmetic (e.g., [12,13]).

Since packing and unpacking of DPD declets entails considerable delay and power dissipation overhead, which is not the case for BCC representation of decimal numbers [14], we are motivated to design and implement BCC arithmetic operators.

The only work on BCC addition that we have encountered [12] extends the conditional speculative mixed BCD/binary addition scheme of [15,16] to radix-1000 operands. The conditional speculation is done via manipulating the seven most significant bits of equally weighted digits of addition operands in order to decide on $+24$ $(= 2^{10} - 10^3)$ speculation.

In this paper, we study all other possible speculation options (i.e., five cases based on 2-6 bits, besides the 7-bit case of [12]) to provide a spectrum of area-delay-power tradeoff for mixed BCC/binary adders. However, for fair comparison, since no simple speculative mixed BCC/binary adder has been reported, we were also motivated to design such adder based on the mixed BCD/binary simple speculative adder of [15].

The rest of this work is organized as follows. Section 2 provides a background on speculative and conditional speculative mixed decimal/binary and the only instance of mixed BCC/binary adders. Details of six practically useful conditional speculation options and corresponding mixed BCC/binary adders are discussed in Section 3. Implementation details are provided in Section 4, where the best speculation option and the corresponding adder design is selected. Comparison with previous mixed BCD/binary and BCC/binary adders, based on synthesis results, is covered in Section 5. Finally, Section 6 draws our conclusions.

2. Background

A ripple-carry BCD adder is structurally the same as a ripple-carry binary adder, where the main difference is that decimal full adders (DFA) are conceptually used in place of binary full adders (FA). The functionality of such n-digit BCD adder is described as $[10c_{i+1} + S_i = X_i + Y_i + c_i,\ (c_0 = 0),\ 0 \le i < n]$, where X_i and Y_i, c_i, S_i, and c_{i+1} stand for input digits, decimal carry-in bits, sum digits, and carry-out bits, respectively. There are various decimal adder implementations in the relevant literature. For example, ripple-carry decimal adder implementations [17], are based on 4-bit binary adders that produce interim sum-digits $W'_i = X_i + Y_i + c_i$, which in case of $W'_i \ge 10$, should be corrected to $S_i = W'_i - 10$ (or equivalently $|W'_i + 6|_{16}$). As in binary adders, this slow ripple-carry scheme can be speeded up via carry look-ahead logic [18]. Another speed-up technique is through early $W_i = W'_i + 6$ speculations and possible final $W_i - 6\overline{c_{i+1}}$ corrections [19]. This can be described by $S_i = (X_i + Y_i + 6) + c_i - 6\overline{c_{i+1}}$, or $S_i = (X_i + 6) + Y_i + c_i - 6\overline{c_{i+1}}$ as in [20], where both parenthesized expressions can be performed in parallel for all decimal digit positions. For example, Fig. 1 illustrates the required steps in adding two 6-digit BCD numbers, where the $-6\overline{c_{i+1}}$ operations are actually taken place as $|10\overline{c_{i+1}}|_{16} = |-6\overline{c_{i+1}}|_{16}$. With the speculative $(X_i + 6)$ operation, the hexadecimal carries correctly serve as decimal carries, which brings about the opportunity of utilizing any accelerative word-wide binary addition scheme (e.g., parallel prefix [21]) that leads to reduction of overall latency of the n-digit decimal adder. Furthermore, the same adder can be used for $(4n)$-bit binary addition.

2.1. Conditional speculation

All the $-6\overline{c_{i+1}}$ corrections, in the parallel prefix realizations of [16,19], and other similar decimal adders (e.g., [20]), take place within the critical delay path. For example, the correction logic of this simple speculative decimal addition scheme travels through two gates within the overall 23 gates (i.e., $2/23 = 9\%$) in the 16-digit decimal adder of [16]. However, Vázquez et al. have managed to put the correction logic off the critical delay path via conditional $+6$ speculation, where reduction of correction cost and some power saving are gained as by-products [15]. This technique, which has been welcome in the design of several arithmetic units (e.g., [22–24]), is described via Algorithm 1 (reproduced from [12] and [15]) and Fig. 2. The heart of this algorithm

[1] Chiliad is aptly named because it demonstrates a group that contains 1000 elements.