Evaluation of Optimum Arithmetic Operators For Approximate Datapath Design

Kan Shi and George A. Constantinides
Department of Electrical and Electronic Engineering
Imperial College London
London, UK
Email: k.shi11, g.contantinides@imperial.ac.uk

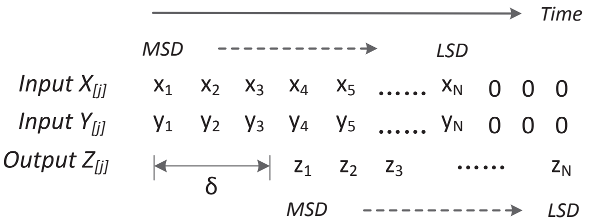


Fig. 1. Dataflow in digit-serial online arithmetic, in which both inputs and outputs are processed from the MSD to the LSD. δ denotes the online delay.

***Abstract*—The abstract goes here.**

***Keywords*—approximate computing; datapath; computer arithmetic;**

I. Introduction

II. Background

A. Approximate Datapath Design

B. Online Arithmetic

Online arithmetic has been found using in numerous applications such as signal processing and control algorithms [1], [2]. Online arithmetic was originally designed for digit-serial operation, as illustrated in Fig 1. It can be seen that in order to generate the first output digit, δ digits of inputs are required, where δ is called the "online delay". δ is normally a small constant, which is independent of the precision. For ease of discussion, for the rest of this paper, the input data is assumed to be fixed point numbers in the range $(-1, 1)$. Based on this premise, the online representation of N-digit operands and result at iteration j are given by (1), where $j \in [-\delta, N-1]$ and r denotes the radix [3].

$$X_{[j]} = \sum_{i=1}^{j+\delta} x_i r^{-i},\ Y_{[j]} = \sum_{i=1}^{j+\delta} y_i r^{-i},\ Z_{[j]} = \sum_{i=1}^{j} z_i r^{-i} \tag{1}$$

MSD-first operation is possible only if a redundant number system is used. Normally there are two most commonly used redundant number representations: carry-save (CS) [4] and signed-digit (SD) [5]. With SD representation, each digit is represented using a redundant digit set $\{-a, \cdots, -1, 0, 1, \cdots, a\}$, where $a \in [r/2, r-1]$. In comparison, the standard non-redundant representation only uses a digit set $\{0, \cdots, r-1\}$. Thus a standard number corresponds to several possible redundant representations. For example, the binary number 0.011 can be represented in SD form as $0.1\bar{1}1$, $0.10\bar{1}$ or 0.011 among many other possible representations.

Due to the redundancy, the MSDs of the result can be calculated using partial information from both inputs. Then the value of the number can be revised using the subsequent digits, because each number has multiple representations.

III. Comparison of Different Adder Structures

A. Ripple Carry Adder

Adders serve as a key building block for arithmetic operations. In general, the ripple carry adder (RCA) is the most straightforward and widely used adder structure. As such, in our previous work we proposed probabilistic models of overclocking errors for RCA.

For an n-bit RCA, it is composed of n serial-connected full adders (FA). Typically the maximum frequency of RCA is determined by the longest carry propagation. Under the assumption that the carry propagation delay of each FA is a constant value μ, the critical path of the RCA is: $\mu_{RCA} = n\mu$. For the sampling period T_S, it follows that if $T_S \geqslant \mu_{RCA}$, correct result will always be sampled; otherwise if $T_S < \mu_{RCA}$, intermediate be sampled and potentially generating errors. In the previous work, the mean value of overclocking error is modeled as given in (2), where coefficient b is given in (6) and it determines the maximum length of error-free carry propagation in bits.

$$E_O = \begin{cases} 2^{-b} - 2^{-n-1}, & \text{if } b \leq n \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$b := \left\lceil \frac{T_S}{\mu} \right\rceil = \left\lceil \frac{1}{\mu \cdot f_S} \right\rceil \tag{3}$$

On the other hand, a specific timing target can be met by truncating the word-length of RCA while timing violation is not permitted. This will result in truncation error for most data. The mean value of truncation error is also modeled as given in (4) [xxx], where parameters k and n denote the word-length of RCA before and after truncation, respectively.

$$E_O = \begin{cases} 2^{-n} - 2^{-k}, & \text{if } n < k \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In this paper we consider two design scenarios for RCA: one is the overclocking scenario where timing violation is

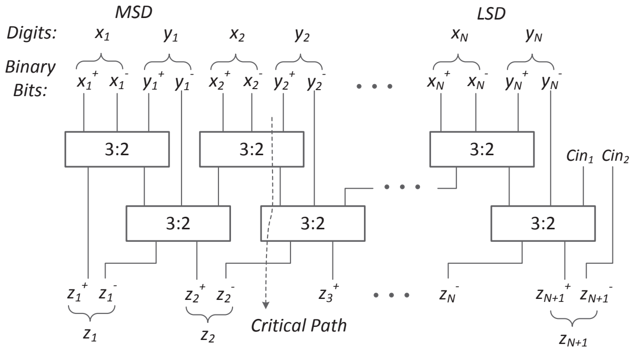

MSD
LSD
Digits:
x_1 y_1 x_2 y_2 x_N y_N
Binary Bits:
x_1^+ x_1^- y_1^+ y_1^- x_2^+ x_2^- y_2^+ y_2^- x_N^+ x_N^- y_N^+ y_N^-
3:2
Cin_1 Cin_2
z_1^+ z_1^- z_2^+ z_2^- z_3^+ z_N^- z_{N+1}^+ z_{N+1}^-
z_1 z_2 z_{N+1}
Critical Path

Fig. 2. An N-digit binary digit-parallel online adder. Both inputs and outputs are represented using SD representation. “3:2” denotes a 3:2 compressor.

allowed to happen while maintaining the original word-length; the other is the trad scenario by truncating RCA word-length to meet timing. For a given f_S, in the first design scenario we have $n = k$. In the second design scenario we have $n = b - 1$ to prevent timing violation while minimizing the value of E_O. Therefore the comparison between the two scenarios in mean error value is given in (5) by updating (2) and (4) respectively. As seen in (5), in theory the mean value of error in the overclocking scenario (E_{oc}) is two times smaller than that in the traditional scenario (E_{trad}).

$$\frac{E_{oc}}{E_{trad}} = \frac{2^{-b} - 2^{-n-1}|_{n=k}}{2^{-n} - 2^{-k}|_{n=b-1}} = \frac{1}{2} \tag{5}$$

B. Carry Select Adder

C. Online Adder

1) Algorithm: The structure of a digit-parallel online adder where all signals represented with SD numbers of digit set $\{-1, 0, 1\}$ is shown in Fig. 2. The module “3:2” denotes a 3:2 compressor, which takes three inputs and generates two outputs, and is logically equivalent to a full adder (FA). A major advantage of the redundant number system over the standard ripple-carry based arithmetic is that the propagation of carry is eliminated, resulting in a precision-independent computation time for addition. As labelled in Fig. 2, ideally the computation delay of this adder is only two FA delays for any operand word-length, with the cost of one extra FA for each digit of operands. This makes the online adder suitable for building up more complex arithmetic operators such as multipliers to accelerate the sum of partial products [6].

$$b := \left\lceil \frac{T_S}{\mu} \right\rceil = \left\lceil \frac{1}{\mu \cdot f_S} \right\rceil \tag{6}$$

IV. Comparison of Different Multiplier Structures

A. Standard Binary Multiplier

B. Online Multiplier

1) Algorithm: Similar to online addition, online multiplication can also be implemented in a unrolled digit-parallel manner, of which the optimized algorithm is shown in Algorithm 1 [xxx]. In this algorithm, both inputs and outputs are N-digit numbers as given in (1). For a given iteration j, the product digit z_j is generated MSD-first through a selection function $sel()$. For any radix r and chosen digit set, there exits an appropriate selection method and a value of δ which ensure convergence. As the binary radix is used most commonly in computer arithmetic, we keep $r = 2$ throughout this paper with the corresponding redundant digit set $\{\overline{1}, 0, 1\}$. In this case $sel()$ is given by (7) [7].

Algorithm 1 Digit Parallel Online Multiplication

```
Initialization: $P_{[0]} = 0$
for $j = 1, 2, \cdots, N$ do
    $Xy_j \leftarrow X \cdot y_j$
    $W_{[j]} \leftarrow P_{[j-1]} + Xy_j$
    $z_j \leftarrow sel(W_{[j]})$
    $P_{[j+1]} \leftarrow r\left(W_{[j]} - Z_{[j]}\right)$
end for
$Z_{[N+1:2N]} \leftarrow frac(W_{[N]})$
```

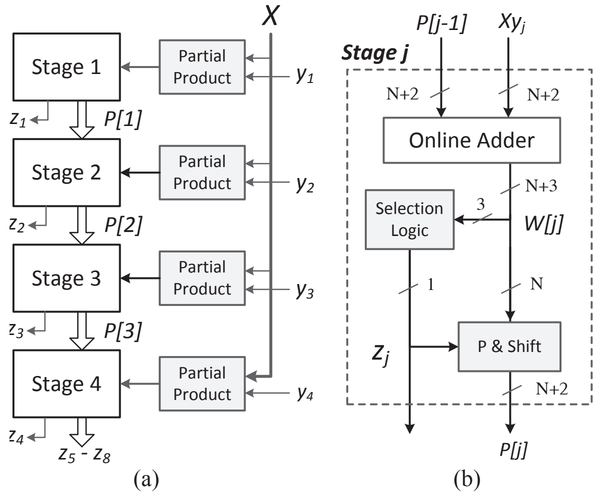


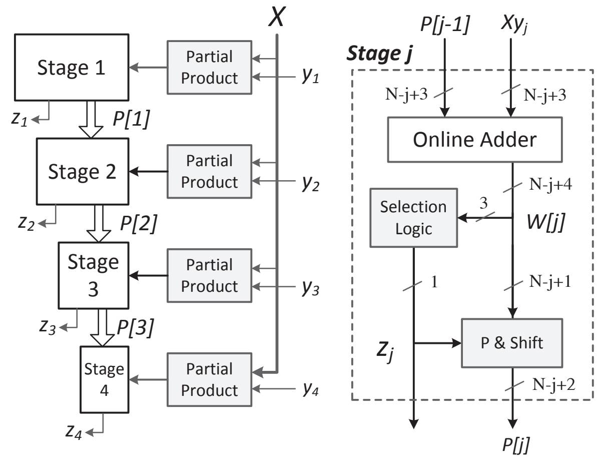
Fig. 3. (a) Structure of a 4-digit online multiplier. (b) Structure of one stage. The word-length of all signals are labeled in digits, N is the word-length of input signals.

$$sel(W_{[j]}) = \begin{cases} 1 & \text{if } W_{[j]} \geqslant \frac{1}{2} \\ 0 & \text{if } -\frac{1}{2} \leqslant W_{[j]} < \frac{1}{2} \\ \overline{1} & \text{if } W_{[j]} < -\frac{1}{2} \end{cases} \tag{7}$$

In our previous work, an area efficient implementation

2) Structure Optimization for Half Precision Results: In actual applications, normally the multiplier is connected with other arithmetic operators. If the multiplication result is utilized for subsequent operations, and a consistent word-length is used throughout the system, then only the most significant half of the product is required. In a conventional multiplier with standard binary arithmetic, this is achieved by either truncating or rounding the least significant half of the products. However, both the computation time and the structure remains unchanged, because results are generated from LSDs.

In comparison, OM offers the flexibility to simplify the structure corresponding to the required precision. This is possible because in an OM, the product digits are generated



(a) (b)

Fig. 4. (a) A 4-digit online multiplier with half precision results. (b) Structure of stage j, with the word-length of all signals labeled.

initially from the MSD, and there is no carry propagation from the LSD to the MSD with the employment of the redundant number system. This will potentially lead to a more area efficient design. For example, the modified structure diagram of a 4-digit OM is illustrated in Fig. 4(a). The word-length of signals within each stage can be correspondingly reduced, as shown in Fig. 4(b). Notice that instead of keeping identical word-length throughout the stages (in Fig. 3), in the modified structure the signal word-lengths depend on the stage number j and hence fewer bits of signals are needed for LSD stages.

V. Conclusion

Acknowledgment

References

[1] R. Galli and A. F. Tenca, "Design and evaluation of online arithmetic for signal processing applications on FPGAs," in *Proc. SPIE Advanced Signal Processing Algorithms, Architectures and Implementations*, Aug 2001, pp. 134–144.

[2] M. Dimmler, A. Tisserand, U. Holmbeg, and R. Longchamp, "On-line arithmetic for real-time control of microsystems," *IEEE/ASME Trans. on Mechatronics*, vol. 4, no. 2, pp. 213–217, Jun 1999.

[3] M. D. Ercegovac and T. Lang, *Digital arithmetic*. Morgan Kaufmann, 2003.

[4] T. Kim, W. Jao, and S. Tjiang, "Circuit optimization using carry-save-adder cells," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 10, pp. 974–984, 1998.

[5] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. on Electronic Computers*, vol. EC-10, no. 3, pp. 389–400, 1961.

[6] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, and N. Takagi, "A high-speed multiplier using a redundant binary adder tree," *IEEE Jourl. of Solid-State Circuits*, vol. 22, no. 1, pp. 28–34, 1987.

[7] K. S. Trivedi and M. Ercegovac, "On-line algorithms for division and multiplication," *IEEE Trans. on Computers*, vol. 26, pp. 161–167, 1975.