An efficient implementation of Online Arithmetic

Yiren Zhao
Imperial College London
London, UK, SW7 2AZ
yiren.zhao13@imperial.ac.uk

George A. Constantinides
Imperial College London
London, UK, SW7 2AZ
g.constantinides@imperial.ac.uk

ABSTRACT

An Online operator generates results starting from the most significant digit (MSD). This allows numerical precision to be controlled at run-time. In this paper, we have developed an optimized FPGA architecture for Online operators to perform efficient serial addition, multiplication and division. We compare these operators with conventional fixed-point arithmetic, over multiple precisions. Using our proposed FPGA implementation, we demonstrate a novel feature in our implementation of providing numerical results to an arbitrary precision at run-time without any increase in circuit area. Applying our approach on the Newton's Method, we have computed results with 32-digit precision with 4.3x less hardware usage, however, is 11.4x greater in latency.

1. INTRODUCTION

Traditional fixed-point arithmetic operators, both parallel and serial, require precision to be defined at design time. These operations fall into two categories; some, such as multiplication, proceed from the least significant digit(LSD) to the most significant digit(MSD), while others, such as division, computes from the MSD to LSD. As a result of this difference, fixed-point arithmetic performs number-by-number computations: it only takes a new number once finishes calculation of all digits within the previous number. Parallel fixed-point operators have a smaller latency compare with serial operators; nevertheless, they consume increasing hardware resources while precision increases.

Online arithmetic unifies all arithmetic operations in a MSD-first fashion. Parallel Online arithmetic requires precision to be confirmed at design time, however, serial Online arithmetic allows successive computations to generate answers digit-by-digit. It ensures an output digit of a number to be immediately fed into following operations without finishing generation of all digits in that number. This enables production of results with arbitrary precision: we could connect Online operators and generate results to an unlimited precision at run-time. Because of its capability of generating results with arbitrary precision, serial Online arithmetic could achieve early decision making and thus early termination of calculations. For instance, to compare a fractional number with 0, we could achieve an early decision once its MSD is generated to be greater or smaller than 0. In contrast, in order to make a valid comparison, fixed-point arithmetic has to calculate this fractional number in all precisions from LSD to MSD.

Serial Online operators contain the features mentioned above, besides, it only uses a fixed amount of circuitry. Therefore, by applying Online arithmetic, we discovered a novel way to provide arbitrary precision generation with a non-growing piece of hardware. Our implementation could track to any precision at run-time without knowing what precision is needed before running.

In later sections, an introduction to Online arithmetic and correlated algorithms are given, and optimized hardware architectures are discussed. We make following contributions in this paper:

- Novel application of Online arithmetic to generate numerical results to an arbitrary precision that is only limited by memory.
- Optimization on hardware architectures of serial Online multiplier and divider.
- Evaluation of serial Online operators based on Newton's method.

2. BACKGROUND

Online arithmetic has been broadly exploited in the fields of control and signal processing [2][5]. Recently, it has also proved its usage in over-clocking computations [7]. Online arithmetic has some unique features due to the algorithms that it relies on. Firstly, the digits are always taken from the MSD (left to right) and the resulting output always contains an initial delay, which is called the Online delay and denoted by [4]. Second feature is that Online arithmetic employs redundant number representation system. To unify the use of representations, in this paper, all implementations only involve digits in signed representation (SD) in a redundant digit set of $\{-1,0,1\}$, which we denote $\{1,0,1\}$ for the ease of presenting.

For each given n-digit operand, we could apply a left-to-right indexing system for its serial computation to the jth iteration, meanwhile, we take considerations of to make this indexing of inputs ($x[j]$ and $y[j]$) and output ($z[j]$) consistent. Since we apply radix-2 system in this paper, the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.