Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 4 (2011) 878–887

Procedia
Computer Science

International Conference on Computational Science, ICCS 2011

GRAPE-MP: An SIMD Accelerator Board for Multi-precision Arithmetic

Hiroshi Daisaka[a,1], Naohito Nakasato[b], Junichiro Makino[c], Fukuko Yuasa[d], Tadashi Ishikawa[d]

[a] *Hitotsubashi University*
[b] *University of Aizu*
[c] *National Astronomical Observatory of Japan*
[d] *High Energy Accelerator Research Organization*

Abstract

We describe the design and performance of the GRAPE-MP board, an SIMD accelerator board for quadruple-precision arithmetic operations. A GRAPE-MP board houses one GRAPE-MP processor chip and an FPGA chip which handles the communication with the host computer. A GRAPE-MP chip has 6 processing elements (PE) and operates with 100MHz clock cycle. Each PE can perform one addition and one multiplication in every clock cycle. The architecture of the GRAPE-MP is similar to that of the GRAPE-DR. It is implemented using the structured ASIC chip from eASIC corp. A GRAPE-MP processor board has the theoretical peak quadruple-precision performance of 1.2 Gflops. As a preliminary result, we present the performance of the GRAPE-MP board for two target applications. The performance of the numerical integration of Feynman loop is 0.53 Gflops. The performance of a N-body simulation with the second order leapfrog schema is 0.505 Gflops for $N = 1984$, which is more than 10 times faster than the performance of the host computer.

Keywords: SIMD, Quadruple-precision, GRAPE

1. Introduction

Most of numerical simulations are currently done using double-precision operations. The reason is that almost all present-day CPUs have double-precision floating-point operation units, and their accuracy is sufficient in most cases. However, there are a lot of problems which can not be solved with double-precision operations.

An example of such problems is the evaluation of the Feynman loop integrals. One of such integrals is a one-loop integral expressed as

$$\begin{aligned} I &= \int_0^1 dx \int_0^{1-x} dy \int_0^{1-x-y} dz F(x,y,z), \\ F(x,y,z) &= D(x,y,z)^{-2} \\ D(x,y,z) &= -xys - tz(1-x-y-z) + (x+y)\lambda^2 \end{aligned}$$

[1] Corresponding author

Selection and/or peer-review under responsibility of Prof. Mitsuhisa Sato and Prof. Satoshi Matsuoka
doi:10.1016/j.procs.2011.04.093

$$+(1-x-y-z)(1-x-y)m_e^2$$
$$+z(1-x-y)m_f^2. \tag{1}$$

Here, s and t are kinematic parameters, m_e and m_f are physical constants, And λ is a fictitious photon mass. We need to evaluate this ingegral for the limit of $\lambda \to 0$. The accurate numerical evaluation of this integral is a hard problem due to its divergent nature [1]. Yuasa et al. [1] reported that the numerical evaluation is numerically unstable with double-precision operations. A solution to this difficulty is to compute the integral in higher precision. Furthermore, Yuasa et al. [1] reported that the use of an extrapolation scheme for λ [2] was necessary.

If we adopt quadruple-precision operations with emulation schemes proposed in [3] and [4], one quadruple-precision variable is expressed by a pair of two double-precision variables. With the scheme in [4], one quadruple-precision addition and one multiplication require 20 and 23 double-precision operations, respectively. Thus, a quadruple-precision operation is at least 20 times slower than a double-precision operation.

To overcome this difficulty, Nakasato [5] has implemented the quadruple-precision emulation scheme on many-core accelerators and reported that GPU and GRAPE-DR [6] were effective in accelerating the numerical evaluation of the Feynman loop integral. However, the performance of quadruple-precision operations on a many-core accelerators is more than 20 times slower than the performance of double-precision operations on the same processors.

In this paper, we describe an alternative approach to improve the performance of quadruple-precision operations. We have designed a special floating-point unit for quadruple-precision operations and implemented it in a structured ASCI chip. Our project, called GRAPE-MP, is a pilot study to investigate the feasibility of a hardware implementaion of a quadruple-precision floating-point unit.

In the following, we describe the design and implementation of the GRAPE-MP system and present performance numbers on selected target applications.

2. GRAPE-MP

In this section, we overview the architecture of GRAPE-MP. The GRAPE-MP is similar to that of GRAPE-DR [6] and its internal structure is simple: (1) GRAPE-MP has programmable processing elements (PE) and memory components. (2) PEs form a Single-Instruction-Multiple-Data (SIMD) processor. (3) Instructions for PEs are supplied from a separate control processor. (4) the control processor, also sends data to GRAPE-MP, receives data from GRAPE-MP, and handles the communication with a host computer.

The GRAPE-MP processor board (Figure 4) consists of one GRAPE-MP chip and one FPGA chip which implementes the contorol processor. This separation of functionalities between GRAPE-MP and FPGA is a key design. In this way, we can maximize the quadruple-precision operation units on a chip. In addition, we can simplify the design of the custom chip, since complex control logic units such as PCIe interface are all in the FPGA chip. In the following, we describe each component of the GRAPE-MP board.

2.1. Numerical Format

First, we describe our numerical representation for a quadruple-precision value. Hereafter, we call our representation eXtended-Double format: XD format. We use 128 bit to express a value in XD format with 1 bit for sign, 11 bits for exponents and 116 bits for mantissa. The size of the exponent in the XD format is the same as the size of the exponent in the double-precision format (IEEE 754) and is smaller than the exponent defined in the binary128 format (11 vs. 15 bits). However, the size of the mantissa in the XD format is larger (116 vs. 112 bits). A drawback of the XD format is that we have no compatibility with the binary128 format. In the current software library for GRAPE-MP, we have implemented conversion functions between the XD format and Double-Double (DD) and Quad-Double (QD) formats proposed in [7].

2.2. Processor Element

Figure 1 shows the block diagram of a PE. Each PE consists of a floating-point multiply unit, an add unit, a special functional unit (RSQ unit), two general purpose register files (GRF1 and GRF2) and an auxiliary register (Treg). The multiply and add units can operate independently and handle quadruple-precision values with the throughput of one operation per clock cycle. The RSQ unit is designed to compute an approximation of the inverse of square root in