Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 166 (2005) 291–298
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

An algorithm for multiple-precision floating-point multiplication

Daisuke Takahashi

Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

Abstract

We present an algorithm for multiple-precision floating-point multiplication. The conventional algorithms based on the fast Fourier transform (FFT) multiply two n-bit numbers to obtain a $2n$-bit result. In multiple-precision floating-point multiplication, we need only the returned result whose precision is equal to the multiple-precision floating-point number. We show that the overall arithmetic operations for FFT-based multiple-precision floating-point multiplication are reduced by decomposition of the full-length multiplication into shorter-length multiplication.

Keywords: Program derivation; Multiplication; Multiple-precision arithmetic; Fast Fourier transform

1. Introduction

Many multiple-precision multiplication algorithms have been well studied [1–6]. Multiple-precision multiplication of n-bit numbers requires $O(n^2)$ bit

E-mail address: daisuke@is.tsukuba.ac.jp

0096-3003/$ - see front matter
doi:10.1016/j.amc.2004.04.034

operations using ordinary multiplication algorithm [1]. Karatsuba's algorithm [2] is known to reduce the number of operations to $O(n^{\log_2 3})$.

Multiple-precision multiplication of n-bit numbers can be performed in $O(n \log n \log\log n)$ bit operations by using the Schönhage–Strassen algorithm [3], which is based on the fast Fourier transform (FFT) [7]. However, the Schönhage–Strassen algorithm may not be able to take advantage of computers with fast floating-point hardware, and it needs binary-to-decimal radix conversion for the final result. Bailey used the discrete Fourier transform (DFT) with three prime modulo computations followed by reconstruction through the Chinese Remainder Theorem for his π calculation to 29 million decimal digits [8].

In addition, a multiple-precision multiplication algorithm using floating-point real FFT is known as another fast multiplication algorithm [9,10].

These conventional FFT-based multiplication algorithms multiply two n-bit numbers to obtain a $2n$-bit result. In multiple-precision floating-point multiplication, we need only the returned result whose precision is equal to the multiple-precision floating-point number. We will call the returned result the "short product" here. In [4,6], algorithms for multiple-precision floating-point multiplication are shown. They used the ordinary $O(n^2)$ multiplication algorithm or Karatsuba's $O(n^{\log_2 3})$ algorithm. However, in multiple-precision multiplication of several thousand decimal bits or more, FFT-based multiplication is the fastest. We show that the overall arithmetic operations for FFT-based multiple-precision floating-point multiplication are reduced by decomposition of the full-length multiplication into shorter-length multiplication.

For simplicity, in this paper, we use the short product which does not provide exact rounding. However, it is not hard to extend multiple-precision floating-point multiplication with exact rounding [4,6]. We also restrict ourselves to computing a mantissa of floating-point numbers in this paper.

2. Multiple-precision multiplication based on FFT

We deal here with real FFT-based multiple-precision multiplication.

Let us consider the product Z of two integers X and Y with length n and base B.

$$X = \sum_{i=0}^{2n-1} x_i B^i, \quad Y = \sum_{i=0}^{2n-1} y_i B^i,$$

where $0 \leqslant x_i < B$, $0 \leqslant y_i < B$, for $0 \leqslant i \leqslant n-1$, and $x_i = y_i = 0$ for $i \geqslant n$.

Then,

$$Z = X \cdot Y = \left(\sum_{i=0}^{2n-1} x_i B^i\right) \cdot \left(\sum_{j=0}^{2n-1} y_j B^j\right) = \sum_{k=0}^{2n-1} \left(\sum_{j=0}^{2n-1} x_j y_{k-j}\right) B^k = \sum_{k=0}^{2n-1} z_k B^k.$$