A Floating Point Convolution Systolic Cell

F. Larochelle, J.F. Côté, A.S. Malowany

Computer Vision and Robotics Laboratory
Department of Electrical Engineering
McGill University
Montréal, Québec, Canada

Abstract

This paper describes the design of a pipeline architecture double precision floating point systolic cell for convolution. Both the pixel intensities and the partial sum values are loaded four bits at the time which implies that a set of operands is loaded every 16 clock cycles. The arithmetic operations are distributed into three pipeline stages which enables the cell to process each set of operands at the same rate as they are loaded. While offering the same precision obtained on standard computers, our systolic cell reduces the convolution time expenditure by as much as three orders of magnitude.

KEYWORDS: Convolution, Floating Point Arithmetic, Pipeline Architecture, Systolic Cell.

I. Introduction

Computer vision and image processing enables robots to guide and validate their activities by supplying them with information about their environment [1]. Unfortunately, algorithms used to extract information out of images generally involve a great amount of computation. Two dimensional convolution is a low-level image processing operation that is necessary in several algorithms [2]. It is easily implemented in software but such implementation normally requires too much execution time for real time robotic applications.

In an effort to reduce the convolution processing time, several VLSI implementations of convolution processor have been proposed [3,4,5]. In our laboratory, a CMOS systolic chip [6] as well as a special architecture board [7] that uses this chip were developed and can be used to perform high speed convolution of images. However, like every other versions proposed in the literature, it uses only integer arithmetic with eight bit precision for the input pixels and sixteen for the resulting sum . We are presently investigating the design of a fast and reliable method of performing curvature estimates of range data [8]. Such a process requires performing six convolutions on a single image in order to compute the first and second derivatives at each pixel. Noise or step edges can produce very large variations making the use of integer arithmetic convolution processors referred above very difficult without generating overflows. It is easily realized that floating point arithmetic convolution is required in such applications.

To enable the construction of a floating point convolution processor, we designed a VLSI double precision floating point systolic cell. Simulation at the architectural level was performed and the results predict that its implementation could reduce the convolution time expenditure by as much as three orders of magnitude while preserving the double precision floating point accuracy that is possible when the convolution is performed in software.

II. Architecture Overview

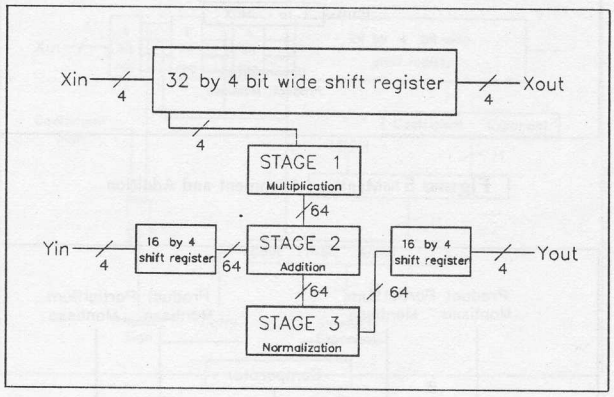

Figure 1 Architecture Block Diagram

The systolic cell is designed to multiply a pixel intensity by a given coefficient and add this product to a partial sum. The coefficient is constant for an entire image and is initially loaded into the cell. During convolution, both the pixel intensities and the partial sum values are serially loaded four bits at the time. A double precision floating point number is represented using eight bytes. It therefore requires sixteen clock cycles to load a set of operands. The cell is organized into three pipeline stages and enables operands to be processed in sixteen clock cycles as well. Figure 1 outlines the basic organization of the pipeline structure and specifies the respective task of each stage. In order to allow these cells to be chained into a systolic array, one extra 16 by 4 shift register has been incorporated inside the chip to delay the flow of intensities and ensure that each meets with the proper partial sum in the next cell. Overflows and underflows are detected and transmitted from one cell to the other. Under the IEEE