Optimal Scheduling for Fast Systolic Array Implementations

I. Ozimek
Institute Jozef Stefan
Jamova 39, 1000 Ljubljana
Slovenia
E-mail: igor.ozimek@ijs.si

R. Verlic
Institute Jozef Stefan
Jamova 39, 1000 Ljubljana
Slovenia

J. Tasic
University of Ljubljana
Faculty of Electrical Engineering
Trzaska 25, 1000 Ljubljana
Slovenia

Abstract

Certain real-time applications (e.g. signal filtering and processing in a digital communication system) require the use of a special massively parallel computing structure, called the systolic array structure, to achieve acceptable performance. To implement an algorithm this way, we need a mapping procedure to map a set of equations, which describe the algorithm, to the systolic array. This mapping consists of scheduling (i.e. time mapping, mapping of each DG node to a particular time instant) and space mapping (mapping of each DG node to a systolic array cell). In the paper we propose a new approach to scheduling of complicated algorithms (that are described by a set of equations, fulfilling the requirement of regularity, i.e. constant dependence vectors). It takes into account the exact computational requirements of the basic arithmetic operations used, and yields near optimal scheduling from the viewpoint of execution speed of the resulting implementation.

1. Introduction

DG (*Dependence Graph*) is one of the basic tools in the systolic array mapping process, [1]. A simple matrix-vector multiplication $\mathbf{c} = \mathbf{A}\mathbf{b}$ can be represented by the following system of recursive equations:

$$\begin{aligned} b_{i+1,j} &= b_{i,j} \\ c_{i,j+1} &= c_{i,j} + a_{i,j}\, b_{i,j} \end{aligned} \qquad (1)$$

The corresponding DG for 3×3 matrix **A** is shown in Fig.1(a). Five (external, or macro) cycles are required for complete computation. If (for simplicity) we assume that multiplication and addition within a systolic cell each require one (micro) cycle, 10 microcycles are required for a complete computation.

2. Better scheduling

By looking at the DG entirely in the terms of microcycles, a better scheduling can be found which requires only 8 microcycles (Fig.1(b)). By using pipelinening, even faster execution is possible.

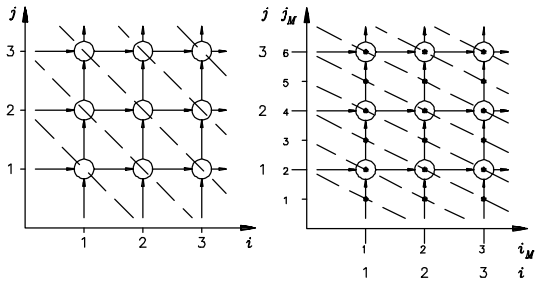

Fig.1 (a) DG and (b) μDG

The speed of computation is limited by loops in RDG (*Reduced Dependence Graph*), [2]. We developed a procedure which finds near optimal scheduling for a systolic array implementation of an algorithm from the viewpoint of execution speed of the resulting implementation, [3].

3. Conclusions

As the need for sophisticated signal processing hardware is increasing along with the introduction of more and more complex communication systems, and the VLSI technology is becoming capable of implementing complex computing hardware on chip, procedures like the one described here are becoming increasingly important for designing of specialised VLSI circuits.

References

[1] S. Y. Kung, *VLSI Array Processors*, Prentice-Hall, 1988.
[2] S. K. Rao, *Regular Iterative Algorithms and Their Implementations on Processor Arrays*, Ph.D. 1986, Stanford University.
[3] I. Ozimek, J. Tasič, "New approach to parallel systolic algorithm implementations - RLSL example", *Signal processing*, Vol. 42, No.1, February 1995, pp. 103-110.

ED&TC '97 on CD-ROM