Design of Algorithmic Array Processors and its Applications

By S. Peng

Faculty of Computer and Information Sciences
Hosei University

Tokyo, Japan

Introduction

- Design of algorithm-oriented array processors has been a popular research topic for more than two decades.
- The famous paper “Systolic arrays (for VLSI)” by H.T. Kung and C.E. Leiserson in 1978, and VLSI technology opened the era for the research on this area.
- The book “VLSI Array Processors” by S.Y. Kung provides major results on this area up to 1988.

Introduction (Continue)

- In 1990s, the research on the area was extended to more advanced issues such as systematic designing tools, asynchronous arrays, programmable arrays, reconfigurable arrays, and fault-tolerance mechanisms etc., and became one of the fruitful branches in parallel and distributed systems.

Introduction (Continue)

- Algorithmic array processors may derive a maximal concurrency by using pipelining and parallel processing. The key question is: How to map application algorithms onto array structures such that the inherent concurrency is fully achieved?
- In this talk, we will focus on this key issue and provide two systematic methods for designing efficient array processors.

Array Processors

- The term "array processors" refers to a class of special-purpose parallel computers (see the book "Computer Architectures and Parallel Processing" by K. Hwang and F. Briggs in 1984)
- The term "algorithmic array processors" in this talk refers to a set of simple, locally interconnected processor elements (PEs). The PE's functionality and the communication structures among PEs depend on the algorithm.

Levels of Design

- For the design of computer systems, there are at least five levels of designs: algorithmic level, architectural level, logic level, circuit level and layout level.
- In this talk, we will focus on the levels 1 – 3 of design.
- In order to describe the design accurately, we need to define some terminologies.

Terminologies

- Algorithm expression is a basic tool for a proper description of an algorithm for parallel and pipeline processing.
- A single assignment code is a form where every variable is assigned one value only during the execution of the algorithm.
- Recursive algorithms are those algorithms expressed by recursive equations with space-time indices (one index for time and the other indices for space).

Terminologies (Continue)

- A dependency graph (DG) is a graph that shows the dependence of the computations that occur in an algorithm. A DG is the graphical representation of a single assignment algorithm.
- An algorithm is computable if and only if its DG contains no cycles.
- A locally recursive algorithm is an algorithm whose DG has only local dependencies.
- A DG is shift-invariant if the dependence arcs in the index space are independent of their positions.

Mapping Algorithm to DG

- Although many methods have been proposed to construct a DG from sequential code, a formal and automatic methodology remains a major open research problem.
- Finding alternative ways for the ordering of the single assignment code of an algorithm such that the corresponding DGs are suitable for constructing efficient array processors is the key for successful design of array processors.

Re-indexing of DG

- A useful technique for modifying the DG is by the re-indexing scheme. For example, when there is no permissible linear schedule or systolic schedule for the original DG, it is often desirable to modify the DG so that such a desired schedule may be obtained.
- In this talk, we will show how to find a re-indexing scheme and construct optimal array processors for division-free Gaussian elimination algorithm using a top-down approach.

A Top-down Design Method: Projection of DG

- In a linear projection, nodes of the DG in a certain straight line are projected to a PE in the processor array along a projection direction.
- To guarantee that each PE executes at most one computation, there are conditions for a admissible projections.
- We use a software tool (S4CAD) for creating all admissible array processors of a given DG automatically.
- This method is suitable for a locally recursive algorithm with 2-D or 3-D DG.

The Top-down Design Method (Continue)

- The design begins with investigating the DG of a locally recursive algorithm, and then uses re-indexing techniques if necessary to re-organize the DG for the higher degree of concurrence.
- The second part of the design is done through a software tool S4CAD (System of Systolic Structures Synthesis). The S4CAD automatically produces all admissible array processors, among which the designer can choose the optimal one.
- The website of S4CAD:

  http://gemini.u-aizu.ac.jp/HPCC/S4CAD

A Bottom-up Design Method for Multi-dimensional Arrays

- For many multi-dimensional signal/image transforms (e.g., DFT, WHT, DCT), the kernels are separable. That is, a *k*-D transform can be done by performing consecutive 1-D transforms *k* times, each along distinct dimension.
- For such applications, we proposed a bottom-up design method using different types of 1-D array processors as building blocks.

The Idea

- Three types of linear array processors with distinct I/O format are created as building blocks:
  - Type I: Input data are fed into the array in a pipelined fashion and output data are generated and stored inside each PE.
  - Type II:Input data are from local registers and output data are delivered to the next PE.
  - Type III: Both input and output data are fed/stored in the local registers.

The Method

- Step 1: Find a 2-D localized DG of the 1-D signal/image transform.
- Step 2: Find the three types of linear array processors through projection and pipelining techniques applying on the 2-D DG.
- Step 3: Refine the design to reduce I/O overhead through analyzing the kernel of the transform.
- Step 4: Build an k-D array processor ($k = 2$ or 3) for the k-D transform as follows: The first and the last dimensions use type I and type II structures, and the intermediate dimension uses type III structure.

An Array Processor for Video Applications

- The bottom-up design method is also called the dimensional-splitting method which can be used to construct array processors for multi-dimensional image/signal transforms with separable kernel.
- The construction of the efficient linear array processors is the key issue for this approach.
- We will show, as an example, how to design array processors for 2-D or 3-D DCT which is practically useful for many video applications.

Concluding Remarks

- Because of changing technologies and applications in the last few years, there are many interesting research topics on this area. We list some possible directions of research below:
  - Design array processors for the new application algorithms: computer vision, 3-D computer graphics, multimedia, virtual and cyber worlds.
  - Design automatic design/analysis software tools (e.g., using java graphical interfaces) for array processors.
  - Investigate array processor designs that are created by the proposed methods in circuit or layout levels.