SIMD Parallel Processing

Michael Sung
6.911 Class Presentation
February 15, 2000

Introduction

- SIMD – Single Instruction, Multiple Data
- Processor Array Machines: Array of fine-grained computational units
- Good for applications with massive data parallelism – “crunch”
- Typical applications for SIMD arrays include low-level vision and image processing, discrete particle simulation, database searches, and genetic sequence matching.

SIMD Issues

- Basic trade-off between processor simplicity and cost: simple control, logic savings
- Communications/network topology: SIMD works best when there is little data exchange between processors
- Programming model: parallel programming tools, high-level languages

Basic Architecture

- Processor Array
- Control processor – issues instructions to processor array
- Synchronous lockstep operation – no synchronization needed
- Individual PE's can be disabled for conditional execution

History

- ILLIAC IV in 1962
  - First large-scale multiprocessor: 64 64-bit processors
  - Infamous supercomputer projects: Est. cost from $8 million to $31 million, Est. performance 1000 MFLOPS to actual 15 MFLOPS
  - Slowed investigation of SIMD architectures

- Connection Machine in 1985
  - Danny Hillis resurrected SIMD: 65,536 1-bit processors
  - After Thinking Machines and MasPar, SIMD has fallen by the wayside in the arena of commercial general-purpose computing

SIMD Machines

- U. of I., Burroughs, DARPA: ILLIAC IV
- Alenia Quadrics: APE
- ICL: DAP
- Penn State: MGAP
- Adaptive Solutions: CNAPS
- NASA: MPP
- Thinking Machines: CM-2
- MasPar: MP-1, MP-2
- MIT AI Lab: Abacus

Connection Machine

- 65,536 bit processors
- 16 processors per die in a 12-cube network topology
  - scalable
  - alternative routes to destination based on node ID
  - packet or circuit-based communication

Abacus

- 1024 Processors, 2D mesh, 8 ns. cycle
- Has DP register for background off-chip data transfers
- Uses configurable bit PEs and RBP algorithms instead of traditional BS

  - each PE emulates logic for part of arithmetic circuit

  - saves silicon area to implement arithmetic

  - incurs reconfiguration overhead/no pipelining

CAM-8

- Cellular Automata Machine – mimics basic spatial locality of physics
- 3D Mesh, lookup-table computation
- CA applications: microscopic physics/lattice gas, statistical mechanics, data visualization/image processing

Problems with SIMD

- Too inflexible; rules out classes of important problems
- Architecture doesn’t scale down in competitive fashion.
- Can’t leverage low-cost advances in microprocessor technology.

Why is SIMD Still Alive?

- Provides more cost performance for special applications
- Still very good for certain classes of problems, e.g. CA
- Small scale commercial variants for multimedia/graphics, etc:

  - Intel’s MMX and SSE extensions

  - AMD’s 3D-Now

  - Motorola’s AltiVec