Announcements

- Reading for this class finish Chapter 9 and start 10
- Project due: Friday at 6:00 PM
  – use submit program for tar file of program
  – drop the writeup by my office

CM-1

- 256 x 256 grid of processors
  - each processor is one bit
  - grouped into 4096 (2^{12}) units of 16 processors each
    - connected by a 12 dimension hypercube
    - also connected to NEWS neighbors
- Processors
  - inputs a & b bits from memory and 1 flag bit
  - output A bit and new flag bit
  - ALU Operators ($2^3 = 8$)

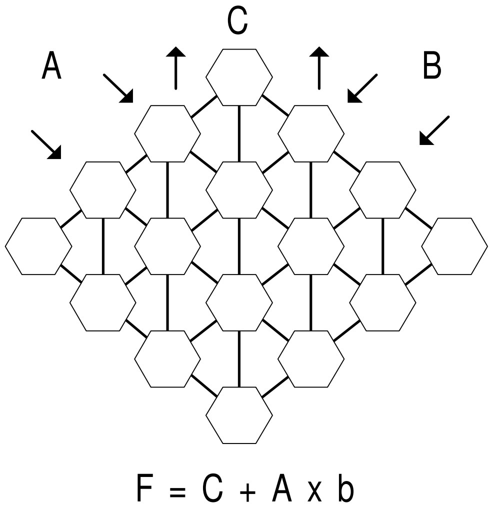
ADD	A← XOR(A,B,F)	Fd←MAJ(A,B,F)
AND	A←AND(A,B,F)	Fd←AND(A,B,F)
OR	A←OR(A,B,F)	Fd←OR(A,B,F)
MOVE	A←B	Fd←AND(XOR(A,B),F)
SWAP	A←F	Fd←B

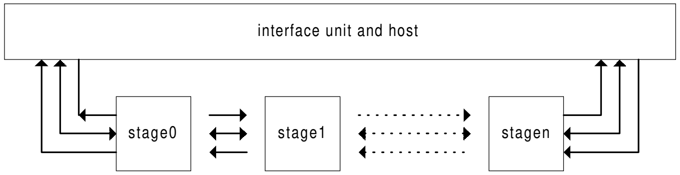
CM-1 (cont.)

- Memory
  - 4096 bits of main memory
  - 16 flag bits
- Micro-controller sends
  - 16 truth table bits
  - A & B address (12 bits each)
  - Read & Write flag (4 bits each)
  - Condition flag (4 bits)
  - Condition sense (1 bit) - should 0/1 imply execute
  - NEWS direction (2 bits)
- Routing Messages
  - store-and-forward
  - cut-through
    - users must prevent collisions
  - end-to-end time of messages is mili-seconds

Systolic Arrays

- Design Issues
  - replicate ALU/FPU
  - optimize memory bandwidth by re-using values at each unit
- Choices
  - dimensions 1, 2, … dimensions possible
  - how many units to replicate?





SIMD Issues

- Performance
  - \+ must wait for the slowest processor
  - – but SIMD slows down the slowest processor
- Design
  - \+ simpler to design processors
  - – no longer commodity processors
- Memory Efficiency
  - \+ no need to replicate text
  - – text is generally a small part of the program
- Programming and Debugging
  - \+ synchronous, so its easy to to understand
  - – must design algorithms to fit SIMD model

MIMD

- Replicate Processor
  - can use commodity parts
- Shared vs. Private Memory
  - shared memory permits any programming model
    - can share data as desired
    - can build message passing using shared buffers
  - private memory is cheaper
    - no expensive interconnect to build
    - interconnect is only to communicate shared info
- Grain Size
  - larger then SIMD machines
  - macro dataflow
    - parts of large datasets flow: $(A^{-1} \times B \times A^{T})^{-1}$
  - macro pipeline
    - cpp | compile | as | ld

Message Passing Machines

- Network of Workstations (NOW)
  - use normal LANs (TCP/IP or custom protocols)
  - message passing libraries (PVM, Express, or MPI)
  - cheap to build
  - communication bandwidth and latency are poor
- Hypercubes
  - Examples: Cosmic Cube and Intel IPSC and IPSC/860
  - original didn't have hardware message forwarding
  - recent systems have included hardware routers
  - how much OS is required
    - just enough to send messages?
    - what about resource allocation?
      - do you need/get a sub-cube?
    - how about tools: debugging performance measurement

Message Passing Machines (cont.)

- Intel Paragon
  - mesh connected machine (2-d mesh)
  - i860 processors
    - 75 Mflops
    - can issue one multiple and one add per cycle
  - communication
    - 200 MB/sec bandwidth (per link)
    - second i860 used for communication (shares memory)
  - memory
    - 16-128 MB/node
    - OS takes over 1MB/node
      - for 1,000 nodes that a giga-kernel