

Thinking Machines Corporation's Connection Machine (CM-2)

- Thinking Machines Corp., founded 1983
 - CM-1, 1986 (1000 MIPS, 4K mem / proc)
 - CM-2, 1987 (2500 MIPS, 64K...)
- Program are on a front-end, which issues instructions to the processor array
 - Sequencer receives instructions, and broadcasts them to all processors in its section of the array
- Parallel data structures are spread across processing elements, one element in each PE's memory
 - For more than 64K elements, operated in virtual processor mode, pretending to have more PEs with less memory

1

Fall 1999, Lecture 03

CM-2 Processor

- Processors (PEs) implemented as processor chip, memory chip, 2 floating-point accelerator chips
- Processor chip
 - Contains ALU, flag registers, NEWS interface, router interface, I/O interface for 16 processors
 - 16 processors are connected by 4x4 mesh allowing N, E, W, S communication
- RAM memory
 - 64 Kbits, bit addressable
- FP acceleration
 - 2 chips serve 32 processors

2

Fall 1999, Lecture 03

CM-2 Processor Chip

- 16 flag registers
 - 8 bits for general purpose use
 - 8 bits predefined
 - NEWS flag
 - Flags for message router data movement and handshaking
 - Memory parity flag
 - Flag for daisy-chaining processors
 - Zero flag (hardcoded)
 - Diagnostic flags
- Instruction
 - All processor chips receive same instruction from the sequencer
 - Individual processors may be masked using a flag bit for processor activation
 - Produces outputs based on memory / flag lookup table (256 possible functions)
 - Arithmetic is bit-serial

3

Fall 1999, Lecture 03

Hypercube

- The 64K processors are linked by a 12-dimensional hypercube
 - Data-parallel message passing
 - Message-combining hardware: receive bitwise OR, numerically largest, or sum
- Simple communication
 - Front-end system to all PEs (although I/O system is better for large amount of data)
- Parallel communications
 - NEWS grid — shift fixed amount
 - Send, Get — arbitrary processors
 - Spread — spread data through PEs
 - Sort
 - Reduce and broadcast — sum, etc.
 - Scan — collect cumulative results over sequence of processors

4

Fall 1999, Lecture 03

Communication

- Router transmits data between processors
 - Messages move across each of 12 dimensions in sequence
 - If no conflicts, a message will reach its destination within 1 cycle of the sequence
 - Every processor can send a message (of any length), all messages are sent and delivered at same time
 - Throughput depends on message length and access patterns
- NEWS grid
 - Nearest-neighbor communication of multiple dimensions: 256x256, 1024x64, 8x8192, 64x32x32, 16x16x16x16, 8x8x4x8x8x4
 - Regular pattern avoids overhead of explicitly specifying destination address

5

Fall 1999, Lecture 03

Nexus

- Nexus is a 4x4 crosspoint switch that connects up to 4 front-end computers to up to 4 sequencers
 - CM can be configured as up to 4 sections, each used separately
 - Any front-end can be connected to any section or combination of sections
 - Example: 64K processors, four 16K sections, 1 to one FE, 1 to another FE, 2 to third FE, fourth FE used for other tasks
- Each section connects to one of 8 I/O channels (graphics display frame buffer, I/O controller)
 - Transfers initiated by front-end computers
 - Data goes into buffers, when buffers are full it goes to a data vault (each has 39 disk drives, total capacity 10GB)

6

Fall 1999, Lecture 03

Software

- System software is based on OS used in front-end computers
 - Use familiar OS, languages
 - Front end handles all flow of control, including storing and executing program, and interaction with user and programmer
 - Languages: Paris, *LISP, CM-LISP, C*
- Paris (parallel instruction set)
 - Inter-processor communication, vector summation, matrix multiplication, sorting
 - Front-end processor sends Paris instructions to processor sequencers
 - Functions & subroutines (direct actions of processors, router, I/O, etc., including scan and spread operations), global variables (find out how many processors are available, etc.)
 - Sequencer produces lower-level instructs.

7

Fall 1999, Lecture 03

BBN Butterfly

- Processor node = processor + memory
 - System can have 1 to 256 processor nodes, as necessary for application
 - Each usually has 1 MB of memory, and can execute 0.5 MIPS
 - Resilient to failures
- Butterfly switch
 - Collective memory of all processor nodes is the shared memory for the machine
 - All processor nodes connect to the butterfly switch, which allows each processor to access memory on remote nodes (slower to access remote memory than local memory)
 - Data goes through switch in packet-switching fashion to other processor nodes

8

Fall 1999, Lecture 03