Flynn's Taxonomy

Flynn's Taxonomy

- Best-known classification scheme for parallel computers
- A computer's category depends upon the parallelism it exhibits in
  - Instruction stream: process may be executing a sequence of instructions
  - Data stream: process may be manipulating a sequence of operands

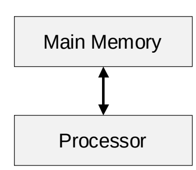
Models of Computation

- SISD: Single Instruction Single Data stream
- SIMD: Single Instruction Multiple Data Stream
- MISD: Multiple Instruction Single Data Stream
- MIMD: Multiple Instruction Multiple Data Stream

Sequential Computers

A conventional computer consists of a processor executing a program stored in a main memory

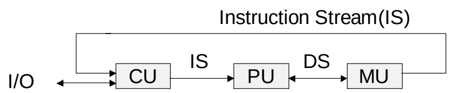
Instructions to Processor



Data to and from Processor

Each main memory location in the memory is located By a number called its *address*. Addresses start from 0 to $2^n - 1$ *n* Binary bits

SISD (Sequential computer):Single Instruction Stream over Single Data



CU: Control Unit
MU : Memory Unit
PU: Processor Unit
DS: Data Stream

Von Neumann Model

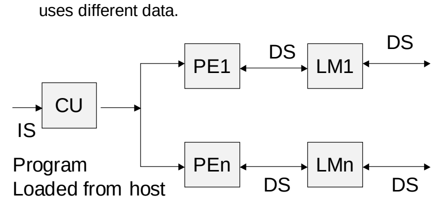
- John von Neumann invented SISD computer
- An algorithm for a computer in this class is called sequential (serial)

However

- Modern uniprocessor may still exhibit some concurrency of execution
- Superscalar architectures:
  - Dynamic identification and selection of multiple independent operations that may be executed simultaneously.
- But, according to Flynn these are examples of concurrency of *processing* rather than concurrency of *execution*.

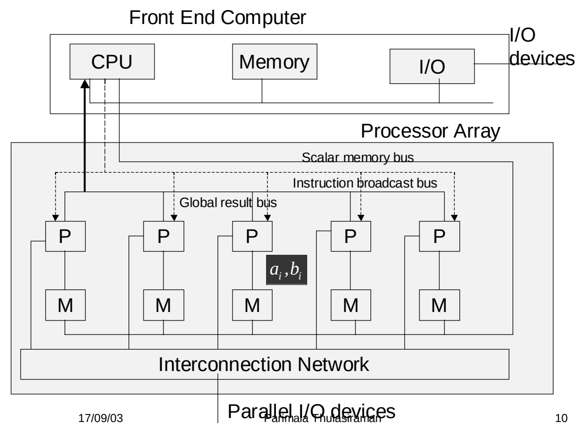
SIMD Single Instruction Multiple Data

- Single IS from a single program, but multiple data streams
- Instructions from program are broadcast to many processors
- Each processor executes same instruction in **synchronism** but uses different data.



SIMD

- In this category are:
  - pipelined vector processors: streams vectors from memory to CPU, where pipelined arithmetic units manipulate them (Cray 1)
  - processor arrays: a vector computer (the instruction set in a vector computer include operations on scalars and vectors) implemented as a sequential computer connected to a set of identical, synchronized elements capable of simultaneously performing the same operation on different data.
    - Popular
    - Data parallel applications



Example

- Adding two vectors: A and B
- A[0] and B[0] given to PE1;
- A[1] and B[1] given to PE2;
- A[2] and B[2] given to PE3;
- Etc.
- Each PE performs the same instruction add

Example

- If there are 1024 PEs
- Length of vector <= 1024:
  - Time to add two vectors <= 1024 is the same time required to add two vectors of length 1024.
  - This is independent of the number of processors that are active
- Length of vectors > 1024; say 10000
  - # of processors > length of vector
  - 784 processors get 10 elements, 240 processors get 9 elements (784*10+240*9 = 10000)

Processor array Performance

- Performance is a metric indicating the amount of work accomplished per time unit.
- Performance of processor array depends on the utilization of the processors
- The size of the data structure directly affects performance.
- Performance is highest when all processors are active.
- Performance in terms of operations/sec.

Question

- Suppose a processor array contains 1024 processors. Each processor is capable of adding a pair of integers in

  What is the performance of this processor array adding two integer vectors of length 1024, assuming each vector is allocated to the processors in a balanced fashion?

Solution

- The number of integer operations performed in 1024. Each processor performs one integer addition, requiring

Performance = 1024 operations/
=

Question

- Suppose a processor array contains 512 processors. Each processor is capable of adding a pair of integers in

What is the performance of this processor array adding two integer vectors of length 600, assuming each vector is allocated to the processors in a balanced fashion?

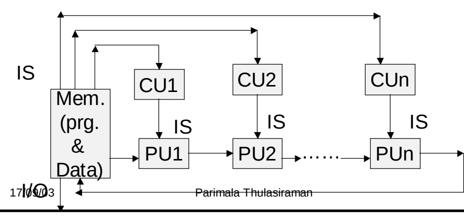
Solution

- The number of integer operations performed is 600. First 512 processors are given 1 element each. So, 600-512 = 88 elements left and given to 88 processors. So, 424 processors add only a single pair of integers. They sit while the other 88 processors add their second integer pair.
- Performance = 600 operations/2 usec

  =

MISD(Multiple Instruction Single Data)

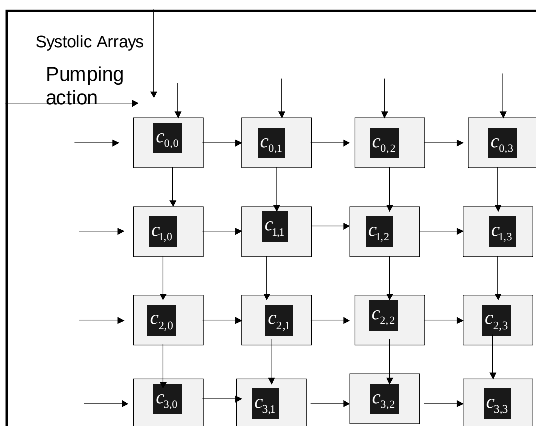
- Same data stream flows through a linear array of processors
- Executes different IS
- Pipelined execution of specific algorithm



Systolic Array

- *Systolic Arrays another name:* name borrowed from the medical field (systole)
- As the heart pumps blood, data is pumped in various directions at regular intervals.
- Below example, info. is pumped left to right and from top to bottom.
- Information meets at internal nodes where the processing occurs.
- The same info. passes onward (left to right or downward).

Systolic Arrays



/* receive from left */

/* receive from top */

/* accumulate value from ci,j */

/* send to right */

/* send downwards */

17/09/03 Parimala Thulasiraman 21

MIMD (Multiple Instruction Multiple Data)

- Multicomputers (distributed memory) and multiprocessors (shared memory).
  - Multiple CPUs can simultaneously execute different instruction streams manipulating different data streams.
- Most contemporary computers fall into MIMD model
- SPMD: Single Instruction Multiple Data
  - Same program on every processor
  - Asynchronous/Independent Execution

17/09/03 Parimala Thulasiraman 22