Instruction Level Parallelism (ILP)

P. K. Roy
Asst. Professor
Siliguri Institute of Technology

Why Parallelism ?

Goal: Speed up and quality up
Higher throughput
Increasing performance & efficiency

Pre-requisite: Pipelining

How is it Possible ?

- Using multiple processors to work toward a given goal, with each processor running its own program.

- Using only a single processor to run a single program, but allowing instructions from that program to execute in parallel.

The latter is called instruction-level parallelism, or ILP.

Performance Metrics/Scalability Issues of a Parallel Processing System

1. Speedup:

Where,

$$S(k) = \frac{T(1)}{T(k)}$$

T(1) = time to execute n tasks in k-stage non-pipelined processor (serial execution time)

T(k) = time to execute n tasks in k-stage pipelined processor (parallel execution time)

cntd…

cntd…

2. Efficiency:

$$E(k) = \frac{T(1)}{k.T(k)}$$

k = no. of stages/processors

3. Throughput (Utilization):

$$H(k) = \frac{E(k)}{t}$$

t = cycle time

cntd…

Exercise 1:

Considering a 4 stage pipeline –

The time taken for 4 stages are

t1 = 40ns t2 = 60ns t3 = 90ns t4 = 50ns

Latch delay = 10ns

Total no. of tasks (n) = 200

Find ***Speed-up , Efficiency & Throughput.***

Flynn’s Classification Of Computer Architectures

- In 1966, Michael Flynn proposed a classification for computer architectures based on the number of instruction streams and data streams (Flynn’s Taxonomy).
- Flynn uses the stream concept for describing a machine's structure
- A stream simply means a sequence of items (data or instructions).

cntd…

cntd…

Flynn's Taxonomy

- SISD: Single instruction single data
  – Classical von Neumann architecture (Uniprocessor)

- SIMD: Single instruction multiple data
  – Vector Processor, Array Processor

- MISD: Multiple instructions single data
  – Non existent, just listed for completeness

- MIMD: Multiple instructions multiple data
  – Most common and general parallel machine
  – Shared Memory or Tightly-coupled system(Multiprocessor), Loosely-coupled(Multicomputer)

SISD

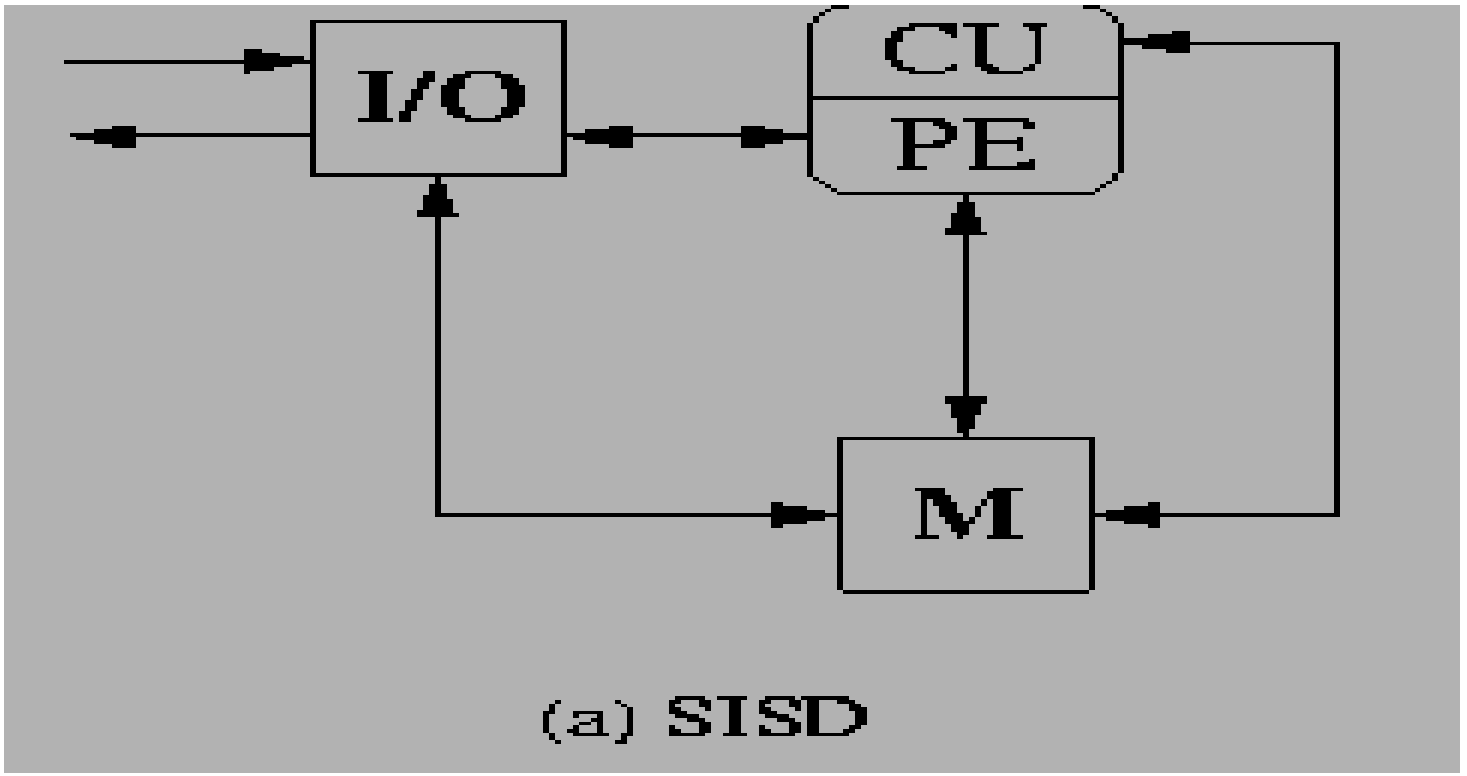
- **SISD** (**S**inge-**I**nstruction stream, **S**inge-**D**ata stream)

- SISD corresponds to the traditional mono-processor (von Neumann computer). A single data stream is being processed by one instruction stream

OR

- A single-processor computer (uni-processor) in which a single stream of instructions is generated from the program.

SISD

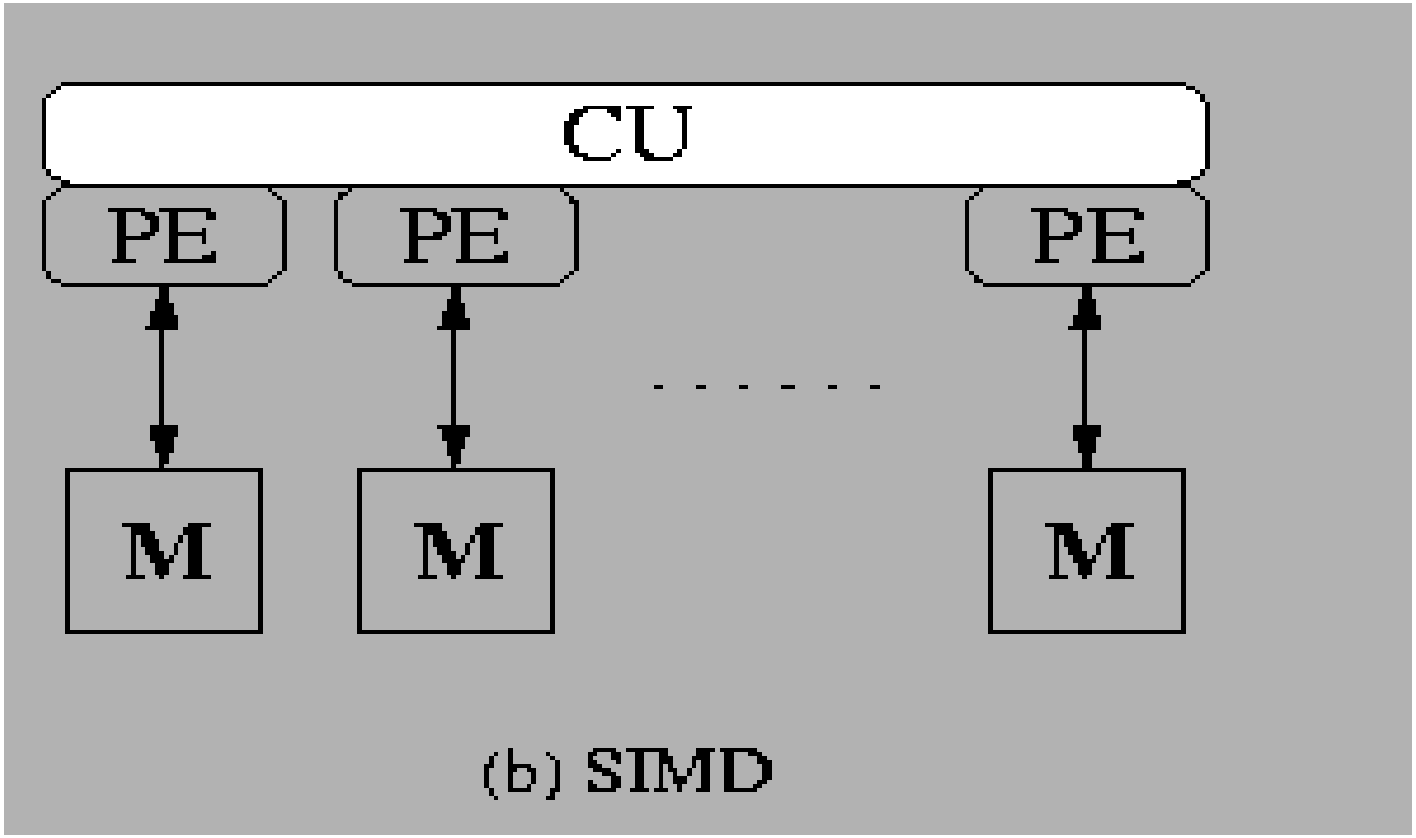


where CU= Control Unit, PE= Processing Element,
M= Memory

SIMD

- **SIMD** (**S**ingle-**I**nstruction stream, **M**ultiple-**D**ata streams)
- Each instruction is executed on a different set of data by different processors i.e multiple processing units of the same type process on multiple-data streams.
- This group is dedicated to array processing machines.
- Sometimes, vector processors can also be seen as a part of this group.

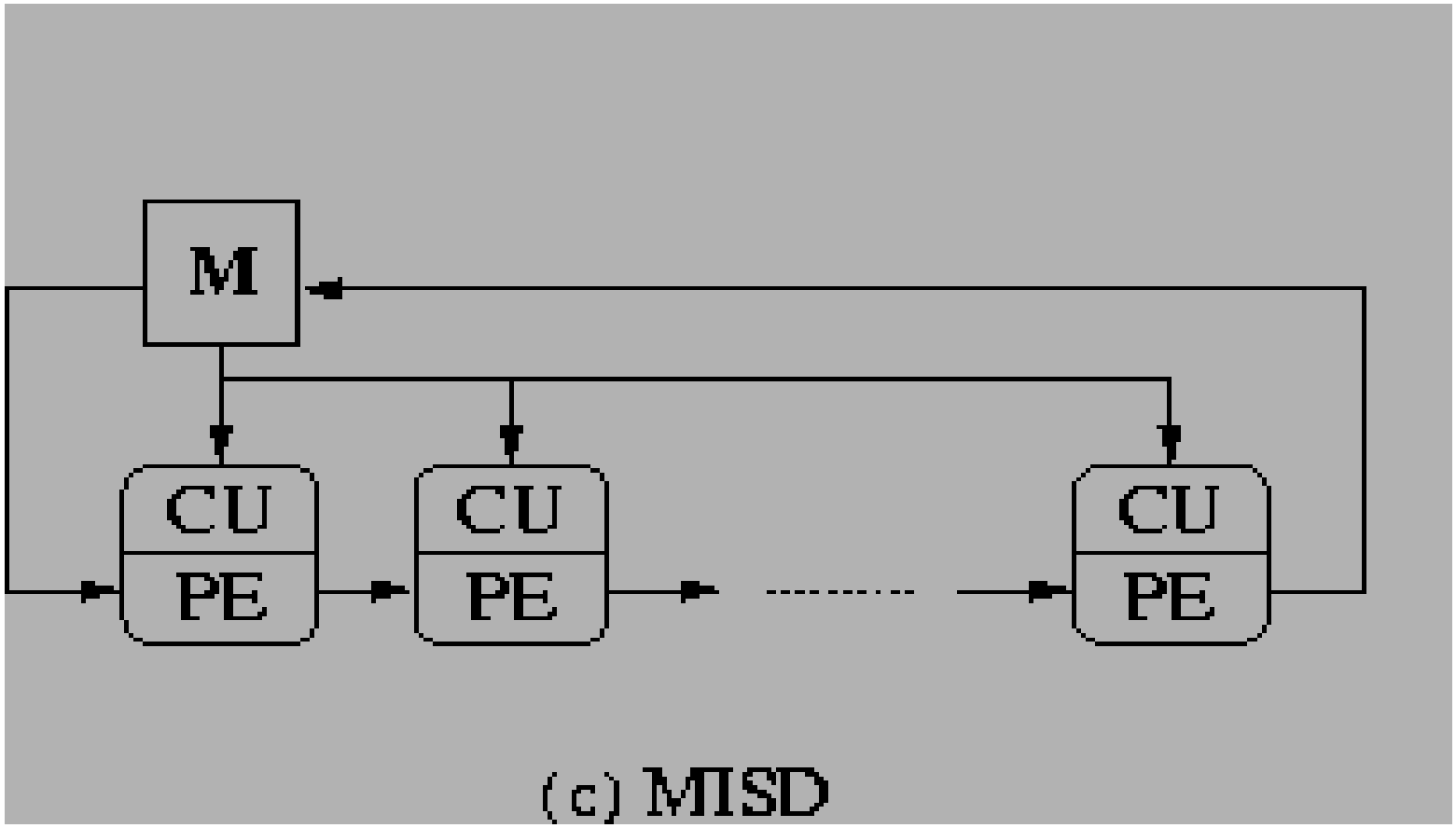
SIMD



MISD

- **MISD** (**M**ultiple-**I**nstruction streams, **S**inge-**D**ata stream)
- Each processor executes a different sequence of instructions.
- In case of MISD computers, multiple processing units operate on one single-data stream .
- In practice, this kind of organization has never been used

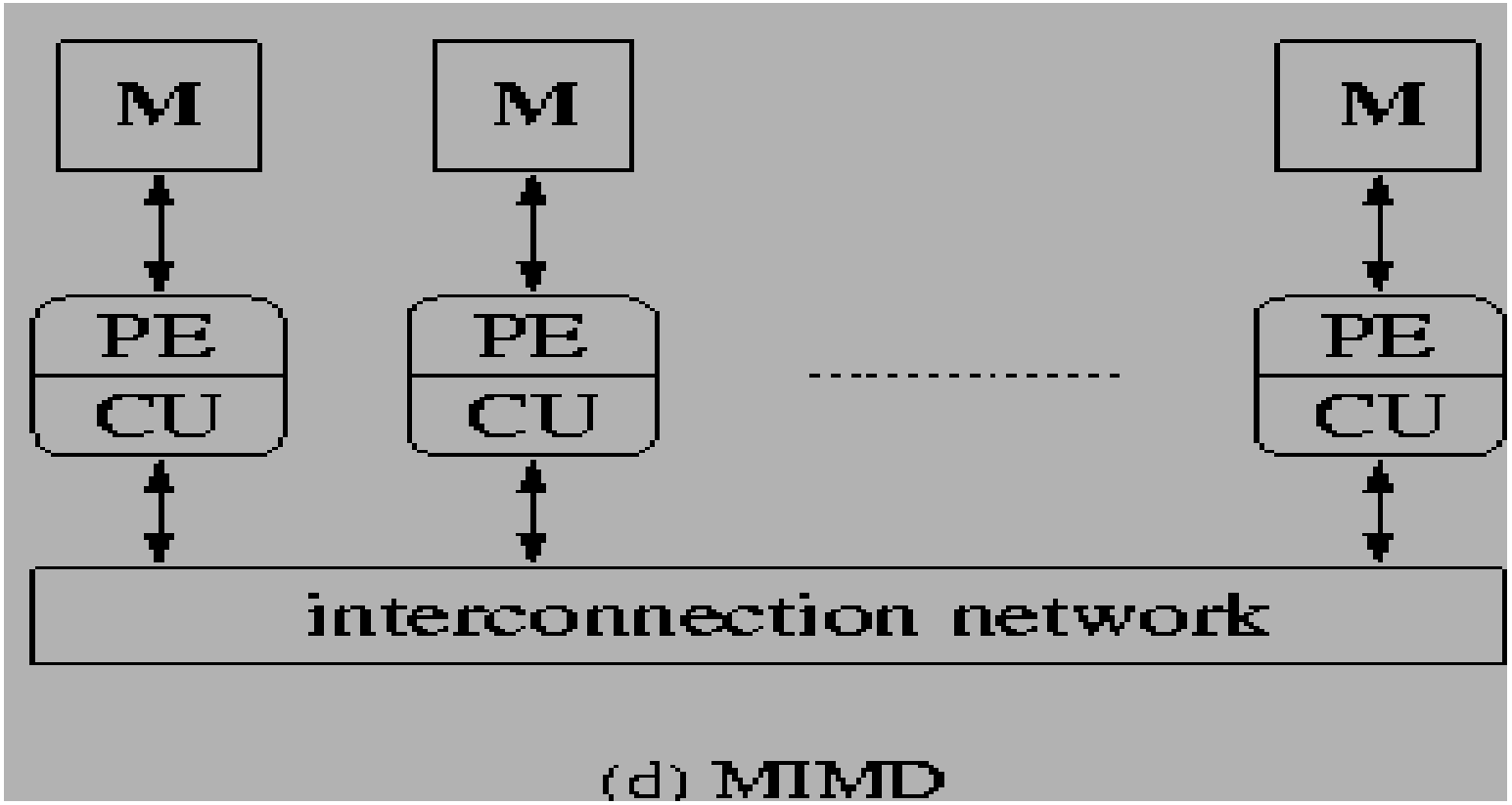
MISD



MIMD

- **MIMD** (**M**ultiple-**I**nstruction streams, **M**ultiple-**D**ata streams)
- Each processor has a separate program.
- An instruction stream is generated from each program.
- Each instruction operates on different data.
- This last machine type builds the group for the traditional multi-processors. Several processing units operate on multiple-data streams.

MIMD



ILP

- More parallelism (or more work per pipeline stage):fewer clocks/instruction [more instructions/cycle]
- Launching multiple instructions per stage allows the instruction execution rate, CPI, to be less than 1
- Increase the depth of the pipeline to increase the clock rate – super-pipelining
- ILP is a measure of the number of instructions that can be performed during a single clock cycle.
- Parallel instructions are a set of instructions that do not depend on each other to be executed.

ILP Hierarchy

- Bit level Parallelism
  - 16 bit add on 8 bit processor
- Instruction level Parallelism
- Loop level Parallelism
  - for (i=1; i<=1000; i= i+1)
    x[i] = x[i] + y[i];
- Thread level Parallelism (SMT, multi-core computers)

Making Computers Think Parallel

- Human
  - Write code yourself, directly controlling each processor
- Compiler
  - Let the compiler convert your sequential code into parallel instructions.
- Hardware

Implementations of ILP

- Pipelining
  -Super pipelining

- Superscalar Architecture
  - Dependency checking on chip.
  - Multiple Processing Elements eg. ALU, Shift

- VLIW (Very Long Instruction Word Architecture)
  - Simple hardware, Complex Compiler

- Multi processor computers

References

1. Advanced Computer Architecture – Kai Hwang
2. Computer Architecture & Organization – John P. Hayes
3. Computer System Architecture – M. Morris Mano
4. Computer Organization & Architecture – T. K. Ghosh
5. Computer Organization & Architecture – Xpress Learning