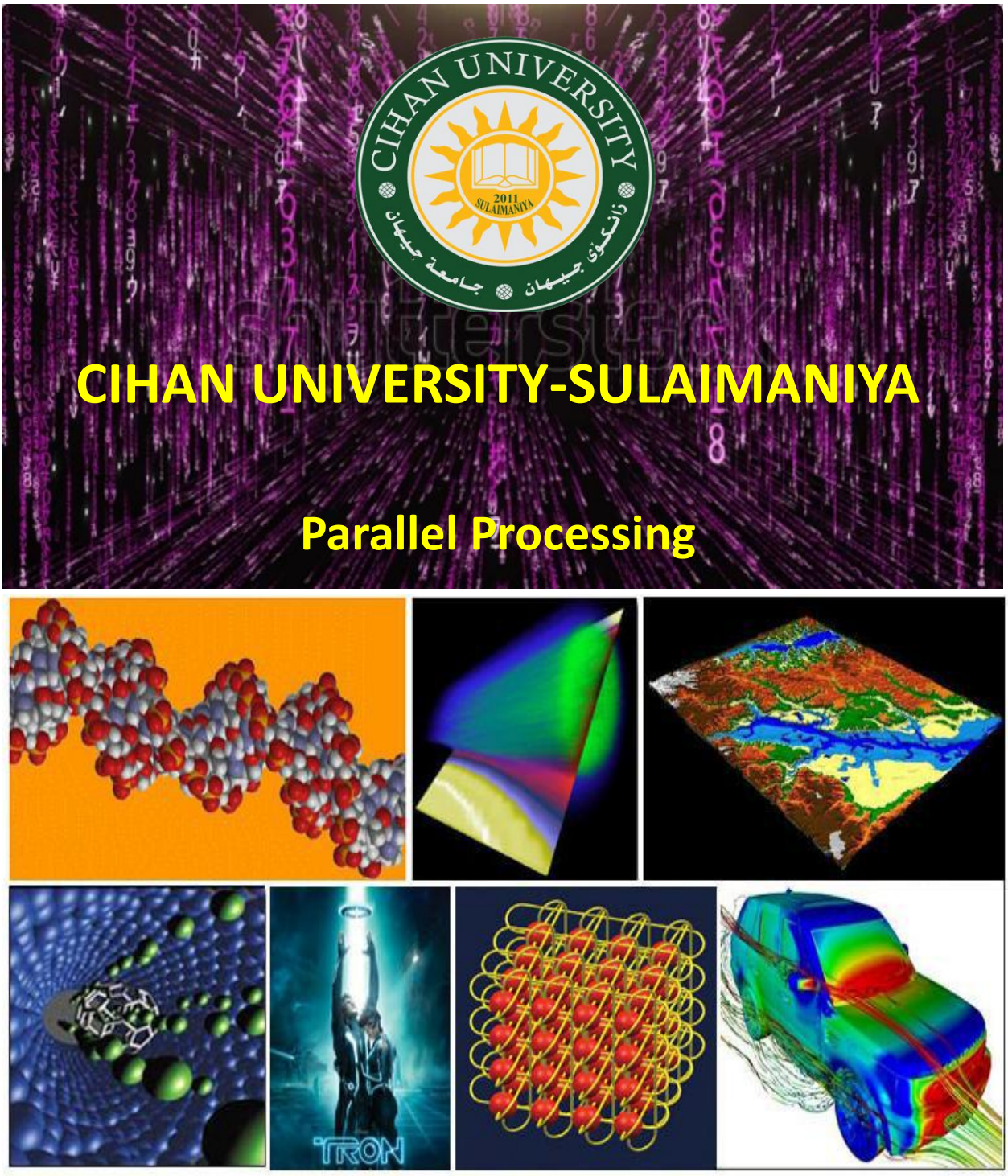


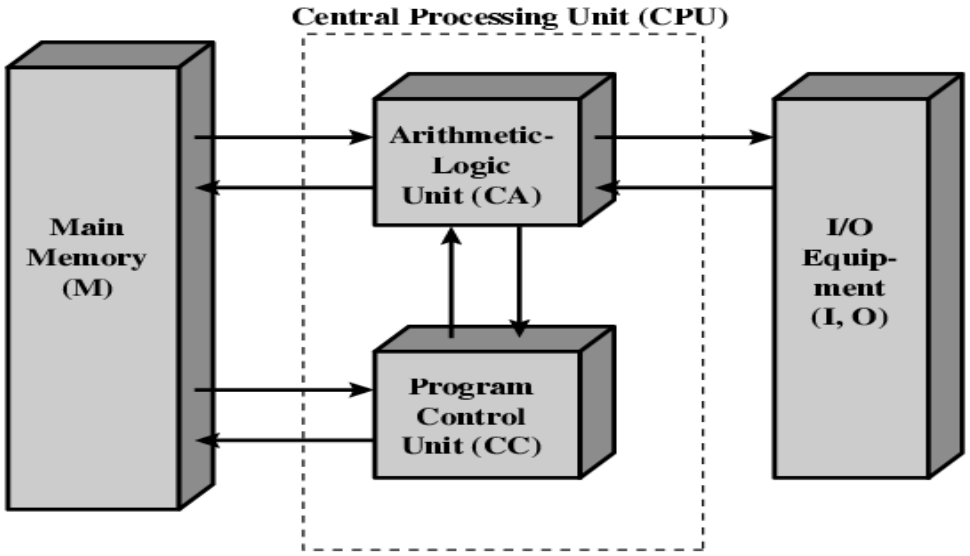
Lecturer: Dr. Kusay Faisal Abdulrazak

Computer Science Department

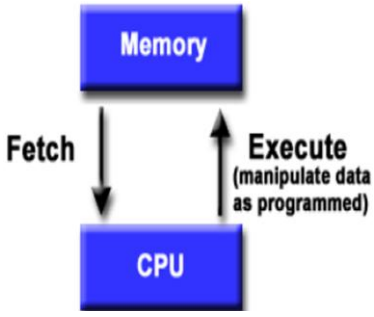
Introduction to Parallel Processing

1.1 Basic Computer

First computer architecture model was invented by Hungarian mathematician John von Neumann from late 1945 until 1951. The Von Neumann machine was the first electronic computer to be built at the **Institute for Advanced Study** (IAS).



- **Basic design**: memory is used to store both: program and data instructions.
  Program instructions are coded data which tell the computer to do something (ex: add+, subtract-, compare > <, etc).
  Program → many tasks → many Instructions.
  Data is simply information to be used by the program.
  A central processing unit (CPU) gets instructions and/or data from memory, decodes the instructions and then sequentially performs them.



The main concepts:

1. Stored program concept.
2. Main memory storing program and data.
3. ALU operating on binary data.
4. Controls unite interpreting instructions from memory and executing.
5. Input and output equipment operated by control unit.
6. Institute for Advanced Study (IAS).

Parallel Processing

The problem:

Uniprocessors must stop getting faster due to limit of high speed achieved, 2.5GHz, 3.3 GHz. Increasing more speed, results more power consuming and higher temperature of the device. Need faster speed and better performance. What do we do?

Parallel Processing is the solution.

1.2 Parallel Processing Definition

It is a collection of processing elements that cooperate and communicate to solve large problems fast by solving more than one problem at a time.

Example: multiprocessors (multicore) computer.

Goals of parallel computing are:

1. **Improve performance**: Execution time or task throughput
2. **Reduce power consumption** (4N units at freq F/4) consume less power than (N units at freq F).
3. **Improve cost efficiency and reduce complexity**.
4. **Improve dependability**: Redundant execution in space.

Level of Parallelism:

1. **Bit level parallelism**:
   - 1970~1985 → 4 bits, 8 bit, 16 bit, 32 bit microprocessors
   - 2004 → 64-bit.
2. **Instruction level parallelism (ILP):**
   - ~1985 through today
   - Pipelining
3. **Data level parallelism:**
   - Different piece of data can be operated on in parallel.
   - Systolic arrays.
4. **Process Level or Thread level parallelism:**
   - Desktop dual processor (PC).

1.3 Applications & Importance

- Earthquake and simulation (help to predict and show earthquake scenario).
- Galaxy formation, planetary movement, weather, climate changing, rush hour traffic,
- Ocean circulation simulation (help to show seasonal variation of ocean temperature)
- Databases, data mining
- Oil exploration
- Web search engines, web based business services
- Medical imaging and diagnosis.

Who is using Parallel computing:

1. Industrial and communication.
2. Science.

1. 4 Some General Parallel Terminology

- **Task:** A logically discrete section of computational work. A task is typically has set of instructions that is executed by a processor.
- **Parallel Task:** A task that can be executed by multiple processors safely (yields correct results).
- **Serial Execution:** Execution of a task sequentially, one statement at a time. In the simplest sense, this is what happens on a one processor machine.
- **Parallel Execution:** Execution of a program by more than one task, with each task being able to execute the same or different statement at the same moment in time.
- **Granularity**: In parallel computing, granularity is a qualitative measure of the ratio of computation to communication. Types:
  1- **Coarse:** relatively large amounts of computational work are done between communication events.
  2- **Fine:** relatively small amounts of computational work are done between communication events

HOMEWORK 1 Submit by next week **(3 Marks)**

1. What is granularity and what are its types? Explain with drawing.
2. Define the following: Task, Instruction, parallel processing.
3. What is the main purpose of the parallel computers?
4. Processor executions are two types? What are they? and what is the difference between them illustrated the answer with diagrams.
5. List 10 applications require parallel computer.
6. What are the levels of parallelism?

Architecture Classification

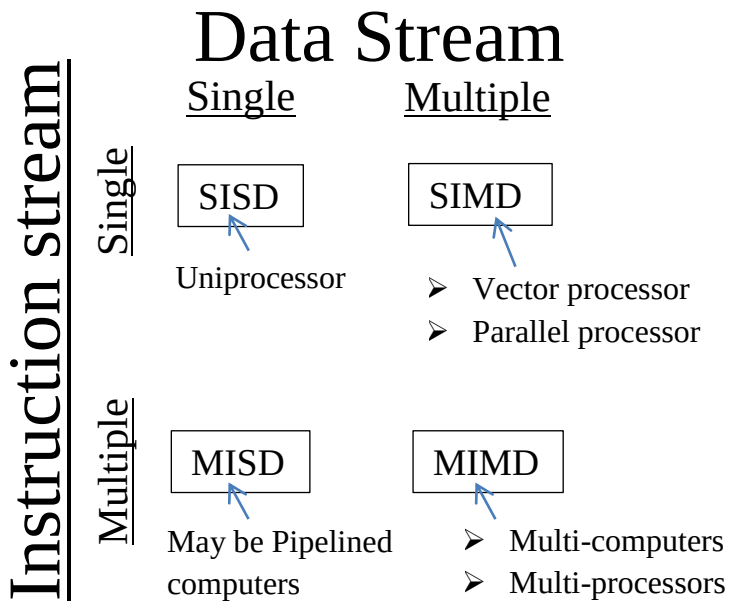
2.1 Flynn's Classification

This classification proposed by Michael J. Flynn in 1966.

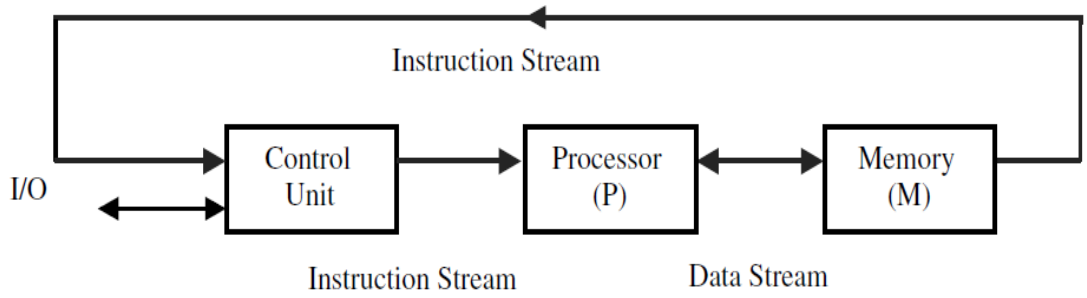
The classification depends on instruction and data stream:

- **Instruction stream**: Is the sequence of instructions which coming from memory to be read by the processor.
- **Data stream**: Is the operations performed on the data in the processor to be stored in the memory.

The following diagram summarizes Flynn's classification:

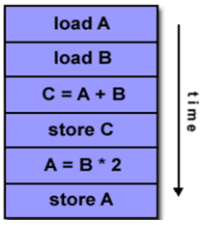


1. **SISD: Single Instruction Single Data.**
   A. One Control Unit.
   B. One Processing Unit.
   C. One Memory unit.

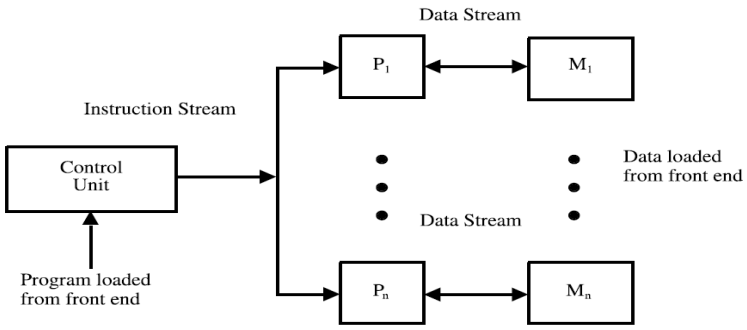


Parallel Processing

- A serial (non-parallel) computer
- **Single instruction (SI)**: only one instruction stream is being acted on by the CPU during any one clock cycle.
- **Single data (SD):** only one data stream is being used as input during any one clock cycle.
- This is the oldest and until recently, the most prevalent form of computer. Examples: most PCs, single CPU workstations.

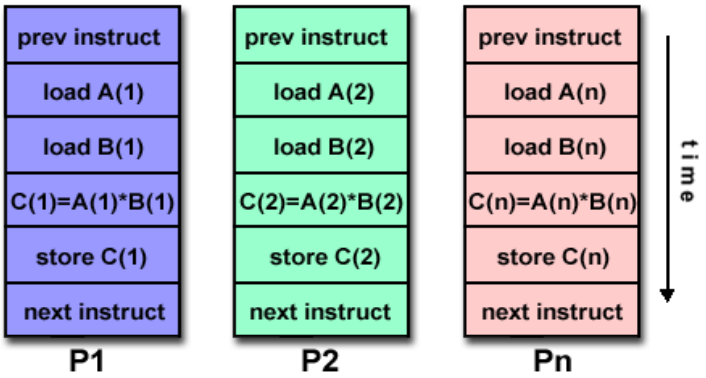


2. **SIMD: Single Instruction Multiple Data.**
   A. Many processes unit under supervision of one control unit.
   B. All Processors (PE) receive same instruction from Control Unit (CU) but operate on different item of data.
   C. Memory must have ability to connect with all processors simultaneously

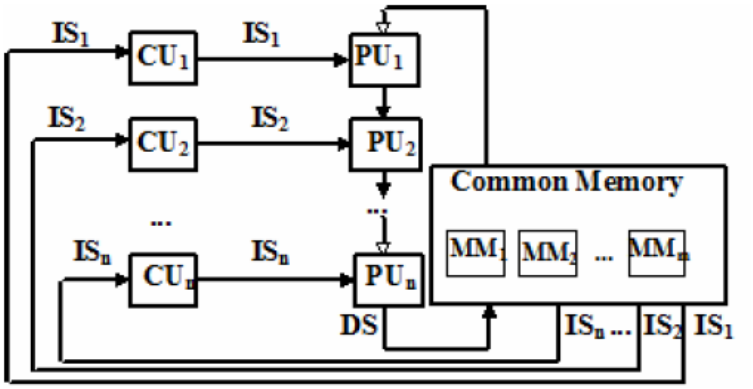


- A type of parallel computer.
- Single instruction: All processing units execute the same instruction at any given clock cycle.
- Multiple data: Each processing unit can operate on a different data element .

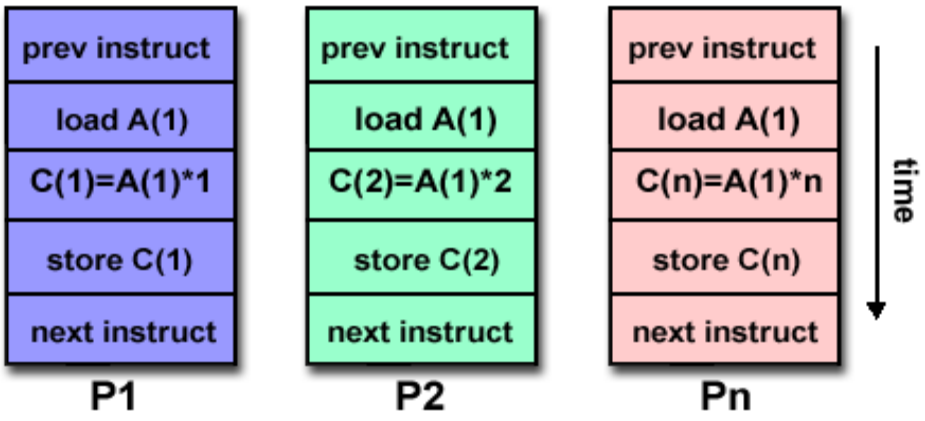
Examples: Vector Pipelines: IBM 9000, Cray C90, Fujitsu VP, NEC SX-2, Hitachi S820.



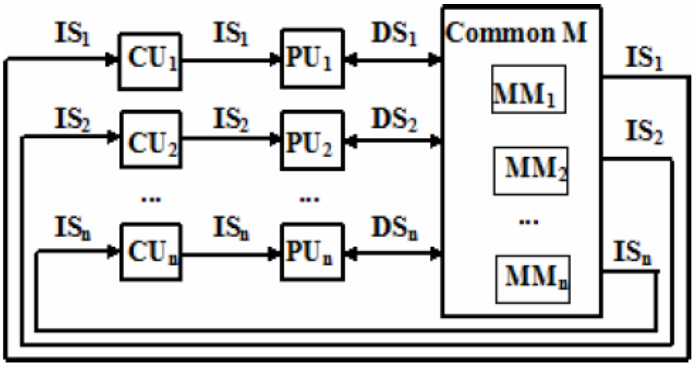
3. **MISD: Multiple Instruction Single Data.**
   A. Multiple CU.
   B. Multiple PU.
   C. Data are taken from memory serially.
   D. All the PU execute the same single Datum.



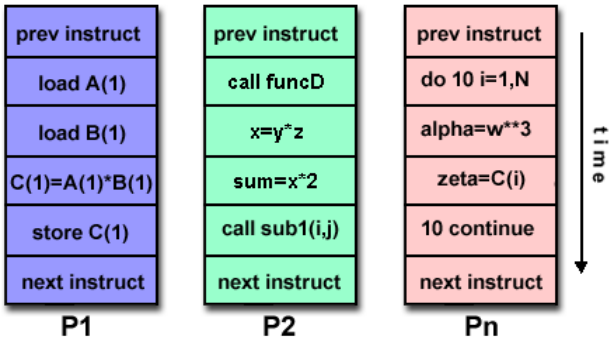
- A single data stream is fed into multiple processing units.
- Each processing unit operates on the data independently via independent instruction streams.
- Few examples of this class of parallel computer have ever existed, such as experimental Carnegie-Mellon C.mmp computer (1971).
- Some conceivable uses might be:
  1. Multiple frequency **filters operating** on a signal stream.
  2. Multiple **cryptography** algorithms attempting to crack a single coded message.

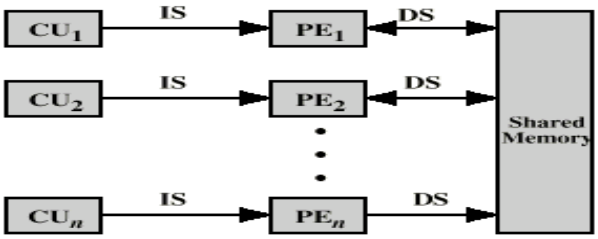


4. **MIMD: Multiple Instruction Multiple Data.**
   A. Multiple CU.
   B. Multiple PU.
   C. Different data are fetched into each CU and PU.
   D. Each PU executes different instruction.

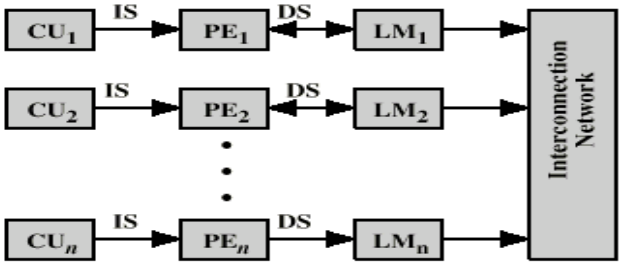


- Currently, the most common type of parallel computer. Most modern computers fall into this category.
- Multiple Instruction: every processor may be executing a different instruction stream.
  Multiple Data: every processor may be working with a different data stream.
  Examples: most current supercomputers, networked parallel computer "grids".

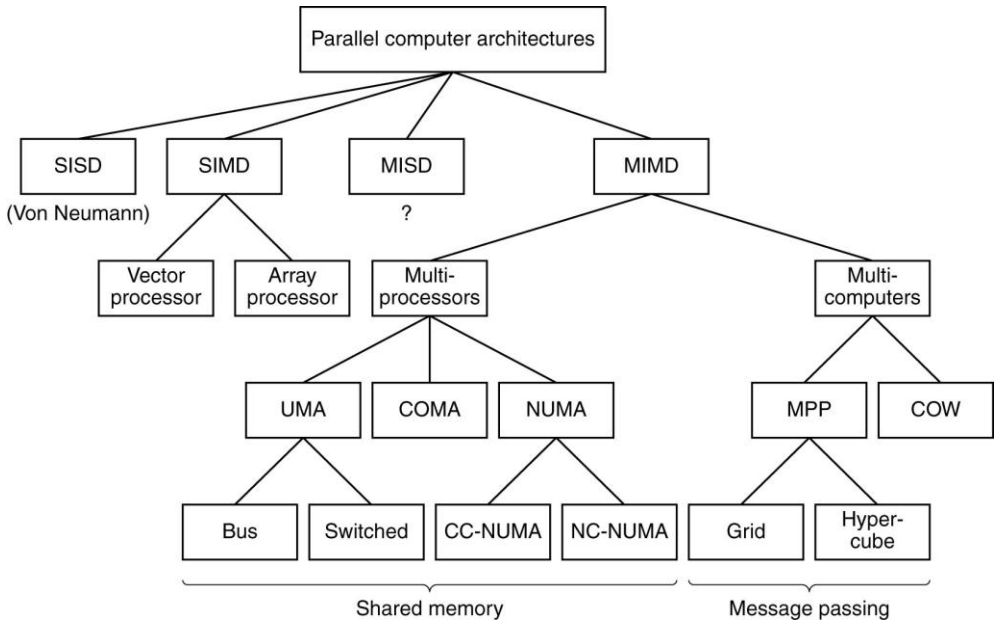




"MIMD Share Memory"



MIMD Message Passing

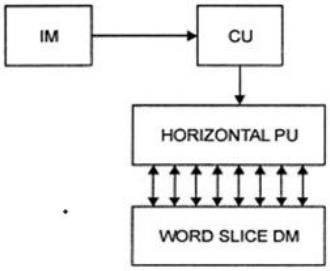


2.2 Shore's Classification

Classification is based on how the computer is organized from its constituent parts:

CU = Control Unit.
PU = Processing Unit.
IM = Instruction Memory.
DM = Data Memory.
Accordingly, 6-kind of machines were recognized by a designer:

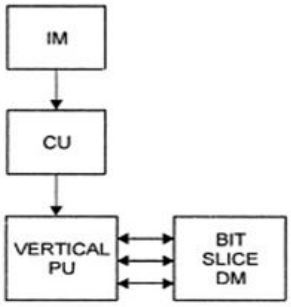
1. **Machine 1:** has IM, CU, PU and DM. PU accesses the data horizontally (bit slice per single word memory).



(a) Machine 1

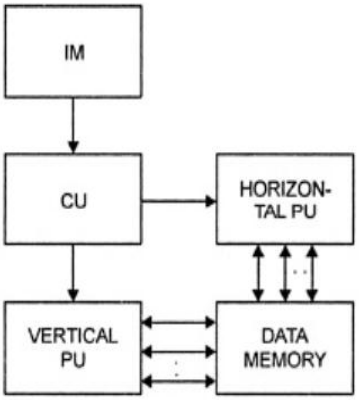
2. **Machine 2:** has IM, CU, PU and DM. PU accesses the data vertically (bit slice per multi-word memory).

Parallel Processing



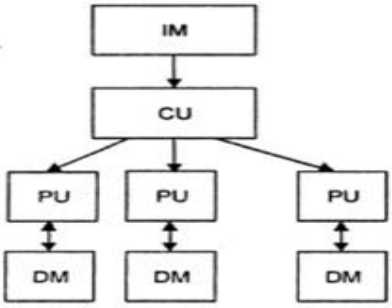
(b) Machine 2

3. **Machine 3:** It is a combination of machine 1 and 2.
   - It has IM,CU, PU and DM.
   - PU Accessing the Data is done by both:
     1. Horizontally: bit slice per single word memory.
     2. Vertically: bit slice per mult-word memory.



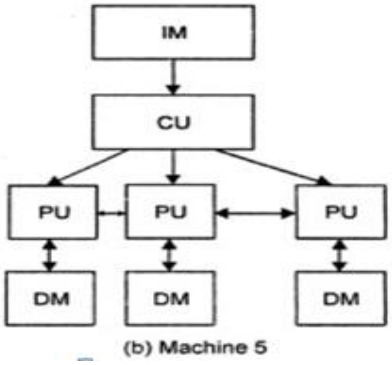
Machine 3

4. **Machine 4:** Multi- PU & DM under one CU. Without neighbor connection among PUs.



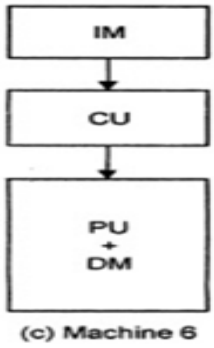
(a) Machine 4

5. **Machine 5**: Multi PU & DM under one CU. With neighbor connection among PUs.



(b) Machine 5

6. Machine 6: named Logic-in-memory array (LIMA), here , PU are distributed throughout the Memory.



(c) Machine 6

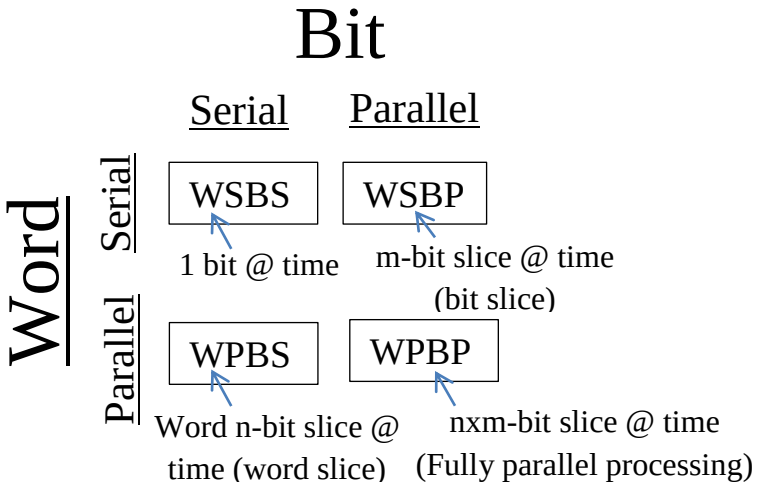
2.3 Feng's Classification

- It proposed by Tseyun Feng in 1972, suggested the use of degree of parallelism to classify various computer architectures.
- It is based on sequential and parallel operations at a bit and word level, word length (n) and bit length (m).

There are 4 types of methods under Feng's classification:

1. **Word Serial and Bit Serial (WSBS):** Has been called bit serial processing because one bit is processed at a time. (slow), example: (1,1).
2. **Word Parallel and Bit Serial (WPBS):** Has been called ***bit slice*** processing because m-bit slice is processes at a time. Example: (*,1).
3. **Word Serial and Bit Parallel (WSBP):** ***Word Slice*** processing because one word of n-bit processed at a time. Example: (1,*).
4. **Word Parallel and Bit Parallel (WPBP):** Is known as fully parallel processing in which an array on $n \times m$ bits is processes at one time. Example: (*,*).

The classification could be summarizing as following diagram:



2.4 Handler's Classification

It is an elaborate notation for expressing the pipelining and parallelism of computers. Handler's classification addresses the 3 distinct levels:

1. Processor Control Unit PCU → Processor.
2. Arithmetic Logic Unit ALU→ Functional unit and process element.
3. Bit-level Circuit (BLC) → Logic circuit needed to perform one bit operations in the ALU.

Handler's classification uses the following three pairs of integers to describe a computer:

Computer= (p * p', a * a', b * b').

p = Number of PCUs.
p' = Number of PCUs that can be pipelined.
a = Number of ALUs controlled by PCU.
a' = Number of ALUs that can be pipelined.
b = Number of bits in ALU or Processing element (PE) word.
b' = Number of Pipeline segments on all ALUs or in a single PE.

Some notation for Handler's Classification:

- The '*' operator is used to indicate that the units are pipelined or macro-pipelined with a stream of data running through all the units.
- The '+' operator is used to indicate that the units are not pipelined but work on independent streams of data.
- The 'v' operator is used to indicate that the computer hardware can work in one of several modes.
- The '~' symbol is used to indicate a range of values for any one of the parameters.
- Peripheral processors are shown before the main processor using another three pairs.

Example-1: Instrument's Advanced Scientific Computer (ASC) has one controller coordinating four arithmetic units. Each arithmetic unit is an eight stage pipeline with 64-bit words. Thus we have:

Answer: Handler's classification of ASC = (1, 4, 64 * 8)

Example-2: The Cray-1 is a 64-bit single processor computer whose ALU has twelve functional units, eight of which can be chained together to from a pipeline. Different functional units have from 1 to 14 segments, which can also be pipelined. Handler's description of the Cray-1 is:

Answer: Handler's classification of Cray-1 = (1, 12 * 8, 64 * (1 ~ 14))

Parallel Processing

Summary:

Architecture Classifications:

1. Flynn's Classification: based on Information (Instruction & Data) stream, has four types: SISD, SIMS, MISD, MIMD.
2. Shore's Classification: based on the computer constituent parts, it has 6 types.
3. Feng's Classification: based on the degree of the parallelism, it has 4 types: WSBS, WSBP, WPBS, WPBP.
4. Handler's Classification: based on a specific expression that describes the computer parts, **Computer= (p*p', a* a', b*b').**

Exercises:

1. According to Flynn's Taxonomy, write a simple program show the branches.
2. According to Flynn's Taxonomy, draw a diagram depicting MISD.
3. Flynn Classification is based on _____ , while Shores 'one is based on _____ .
4. Explain Feng's classification briefly?.
5. Multi-core or mult-processor, under which branch of flynn 's classification.
6. Explain and draw all shores 's classification?
7. The CDC 6600 has a single main processor supported by 10 I/O processors. One control unit coordinates one ALU with a 60-bit word length. The ALU has 10 functional units which can be formed into a pipeline. The 10 peripheral I/O processors may work in parallel with each other and with the CPU. Each I/O processor contains one 12-bit ALU. What is the handler's classification for both:
   1- CDC 6600 Main Processor.
   2- CDC 6600 I/O processor.
   3- Overall the system of CDC 6600.

Parallel Architecture:

3.1 Introduction

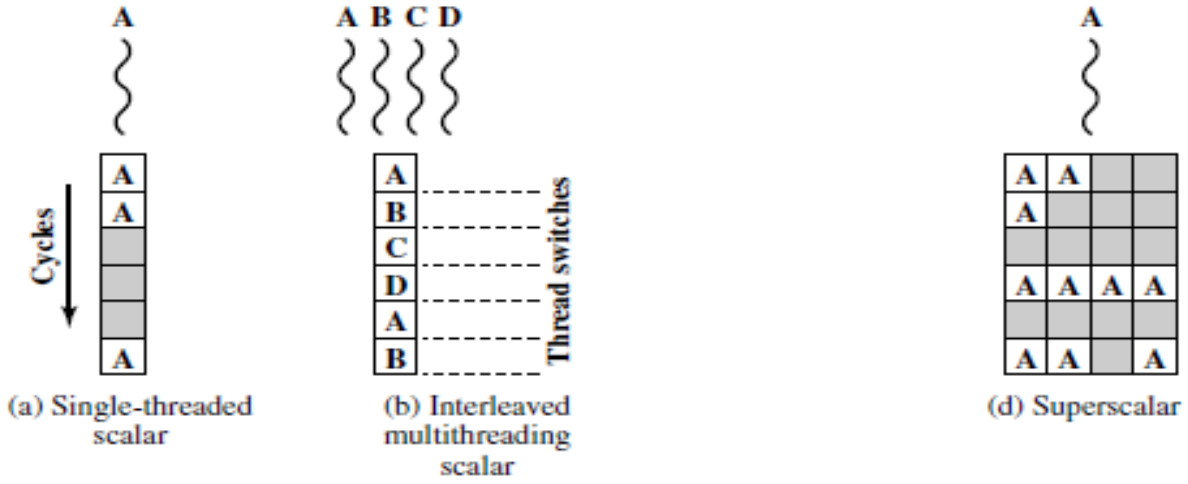
Parallel Computers: refer to architectures in which many processors are running in parallel to implement certain applications.

Parallel Computer can be organized in very different ways, depending on several key parameters:

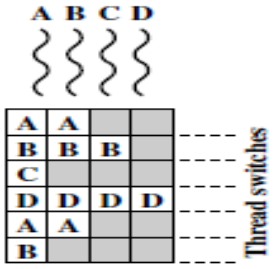
1. Number and complexity of individual CPUs.
2. Availability of common (shared) memory.
3. Interconnection technology and topology.
4. Performance of Interconnection network.
5. I/O devices.

3.2 Multithreading and Chip Multiprocessors

- **Program:** Sequence of instruction to achieve a function.
- **Process:** A program under CPU execution.
- **Thread (Task):** A displaceable unit of work within a process. Many thread = one process
- **Instruction:** Group of bits, tells CPU what to do either arithmetic or logic operations.
- **Single-thread scalar**: One Pipeline executes one thread (A) as figure (a).
- **Interleaved multithreading scalar**: Several threads are being switched in a scalar, (b).
- **Superscalar:** Multi-pipeline execute a thread (A), as in figure (d).

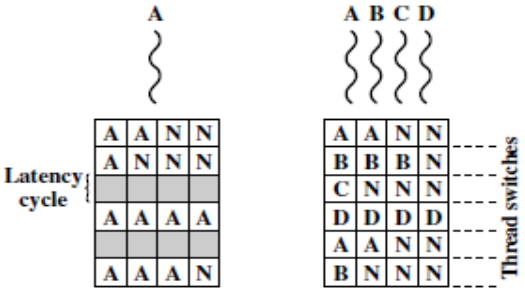


- **Multithreading superscalar:** several threads in multi-pipeline, at each cycle executes same thread, (c).



(e) Interleaved multithreading superscalar

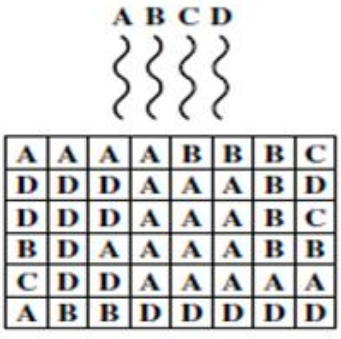
- **VLIW (Very Long Instruction Word):** Same as superscalar but in one word has more than one thread (compiler must arrange), if there is no operation. Put (no-op=N), (g).
- **Multithreading VLIW:** Same as multithreading superscalar but has longer word, if there is no thread, but N=no-op, (h).



(g) VLIW

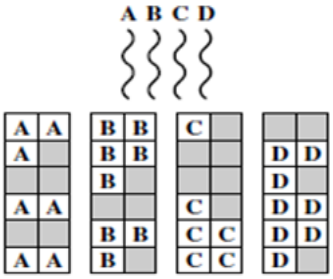
(h) Interleaved multithreading VLIW

- **Simultaneous Multithreading Processor (SMT):** Same as VLIW with ability to execute multithreading either same or different in same cycle, as in figure (j).



(j) Simultaneous multithreading (SMT)

- **Chip Multiprocessor (Multicore) SMP:** In figure (k), a chip containing 4 Physical processors, each of which has a two-issue Superscalar processor. Each processor is assigned a thread, from which it can issue up to two instructions per cycle.



(k) Chip multiprocessor (multicore)

Example: Pentium 4 uses a **multithreading** technique that the Intel literature refers to as **hyperthreading** by using SMT with support for 2 threads. How many logical and physical processors does it contain?

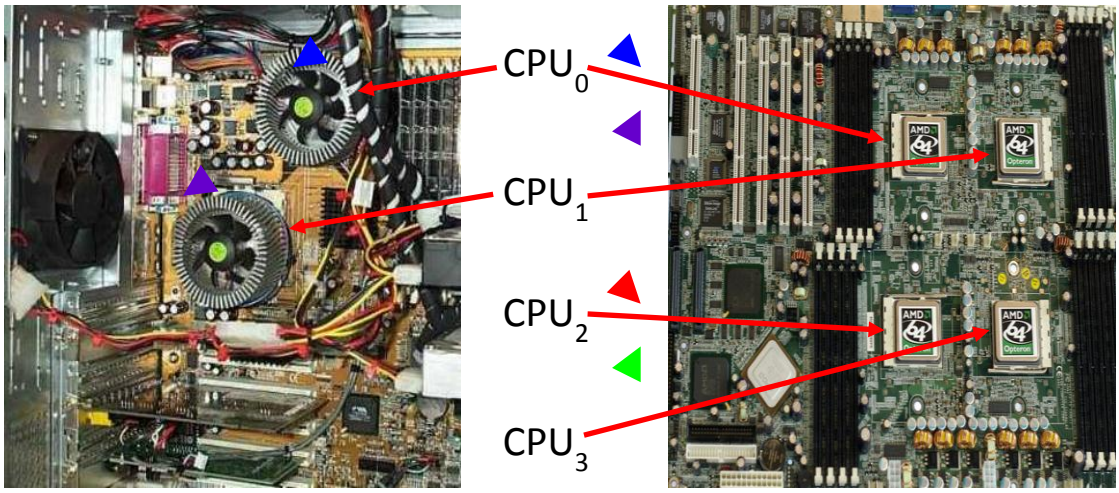
Solution:

1. Physically: one CPU.
2. Logically: Single multithreaded processor is two CPUs.

3.3 Multiprocessors

- Synchronous Multiprocessors (Array processors).
- Asynchronous (Conventional) Multiprocessors.

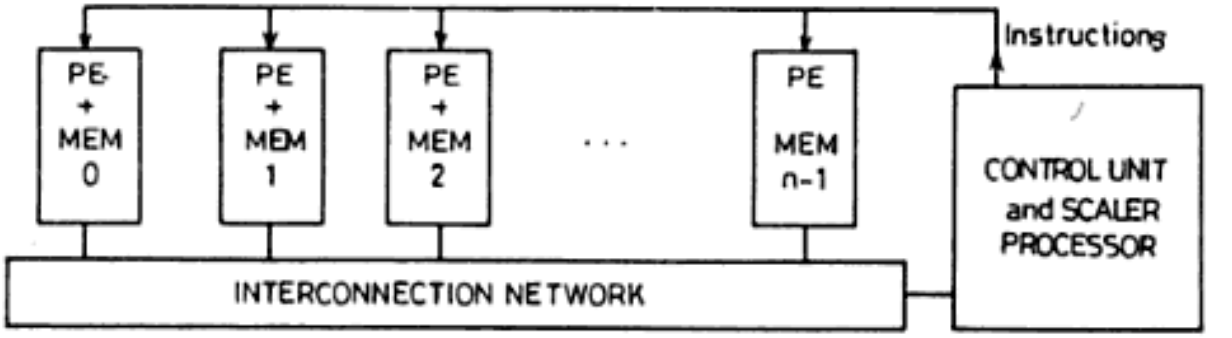
  Symmetric Multi-Processor (SMP).



Parallel Processing

3.2.1 Synchronous Multiprocessors (Array processors).

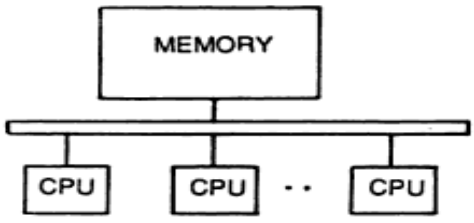
- This machine model consists of several processing elements (PEs), all of which execute the same instruction on different data.
- The instruction is usually fetched and broadcast to all (PEs) by a comment control unit.
- According to Flynn's classification, It is SIMD type.



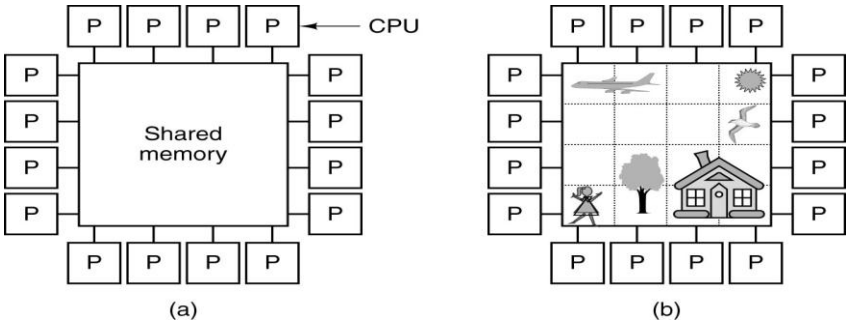
3.2.2 Asynchronous (Conventional) Multiprocessors.

Symmetric Multi-processor (SMP).

- This machine model consists of several CPUs and memory banks connected through either a bus or connection network.



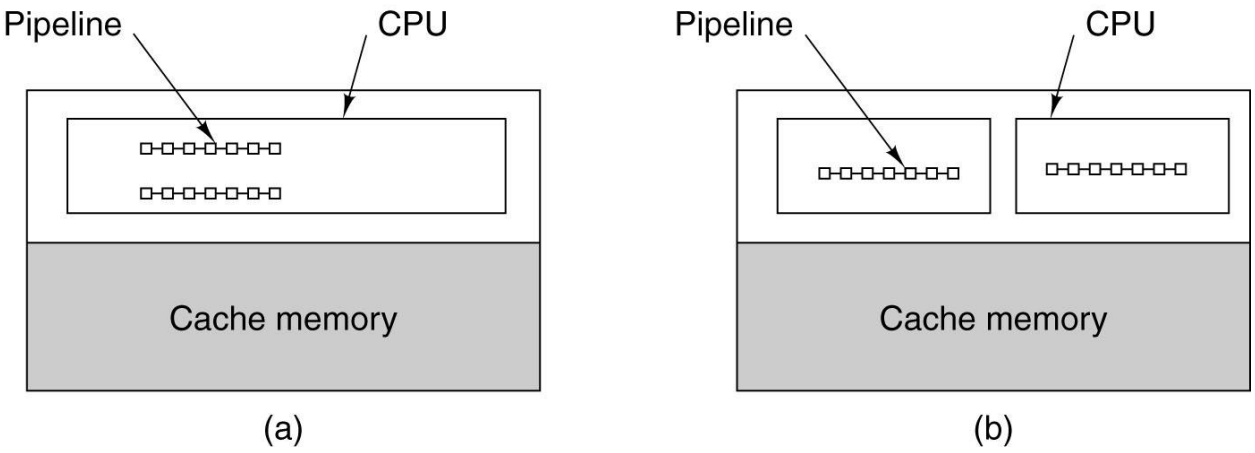
Common Bus Architecture



a) A multiprocessor with 16 CPUs sharing a common memory.
b) An image partitioned into 16 sections, each being analyzed by a different CPU.

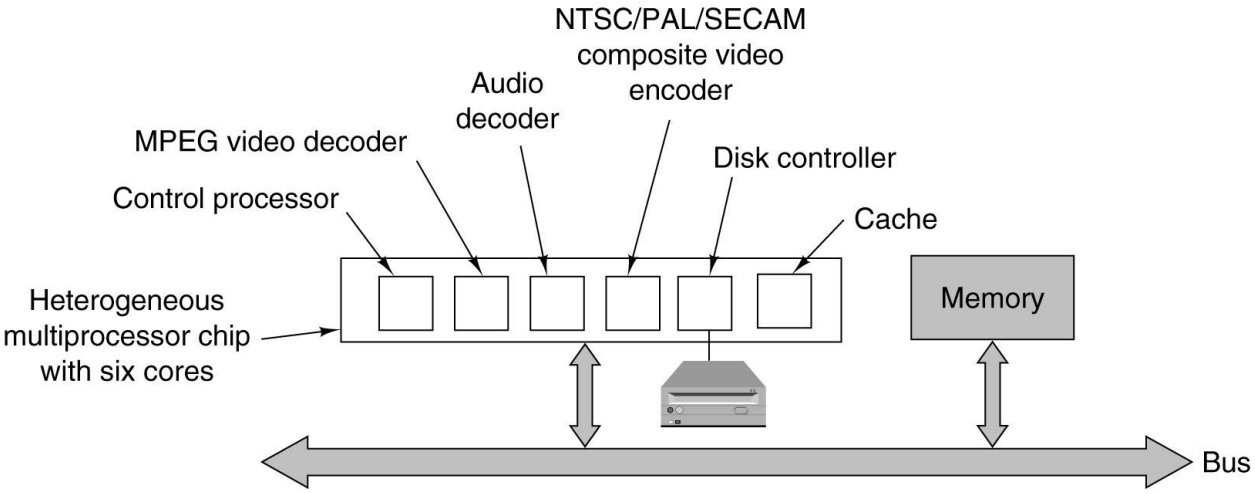
Parallel Processing

Multiprocessor (CPUs) on single chip has two types: Homogeneous (with identical processors) and Heterogeneous (different processors).



(a) A dual-pipeline chip. (b) A chip with two cores.

Single-chip homogeneous multiprocessors.



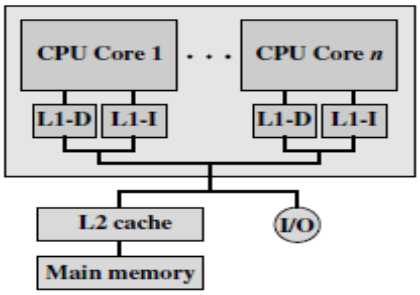
The logical structure of a simple DVD player contains a **heterogeneous** multiprocessor containing multiple cores for different functions.

Parallel Processing

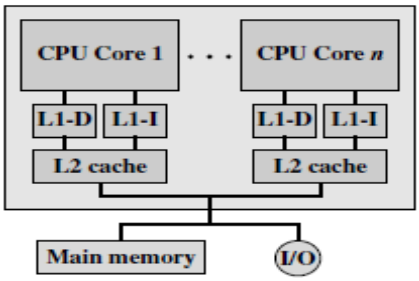
Multiprocessors (Multi-core processors) (multi-cpu):

- **Each Core has the following components:**
  1. **Independent processor (CPU):**
     - **Registers.**
     - **ALU: (Arithmetic Logic Units).**
     - **Pipeline hardware.**
     - **Control unit.**
  2. **L1 instruction and data caches:** Cache is small enough to provide a one- or two-cycle access time (size in KB "about 384KB").
  3. **L2 cache and in some cases, L3 cache:** L2 cache is built from SRAM but is larger than L1 (size in MB "about 3MB"), and therefore slower, than the L1 cache. While L3is bigger (size about 32MB) and slower.

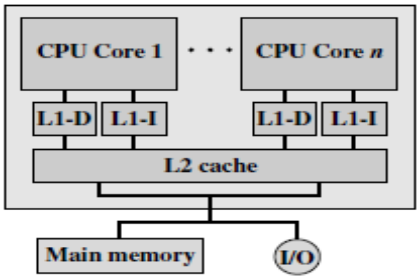
- **Types of multi-core processors organizations (multi-cpu):**
  1. **Dedicated L1 Cache.**
  2. **Dedicated L2 Cache.**
  3. **Shared L2 Cache.**
  4. **Shared L3 Cache.**



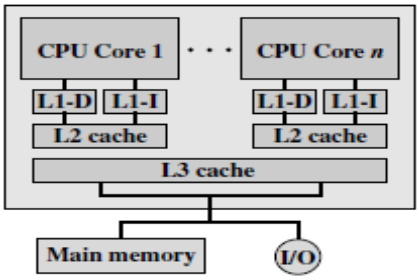
(a) Dedicated L1 cache



(b) Dedicated L2 cache

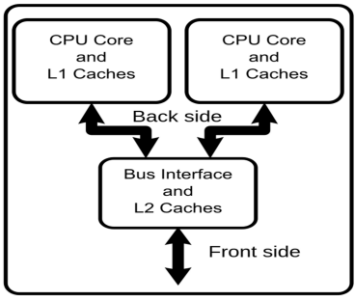


(c) Shared L2 cache



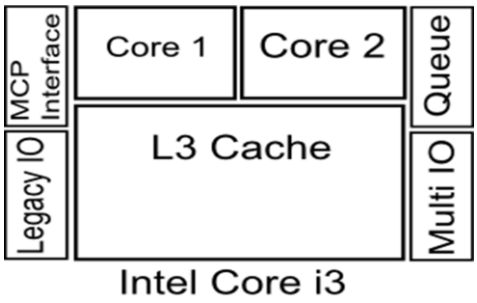
(d) Shared L3 cache

- Multicore on chip Core 2 due architecture. Tightly coupled.



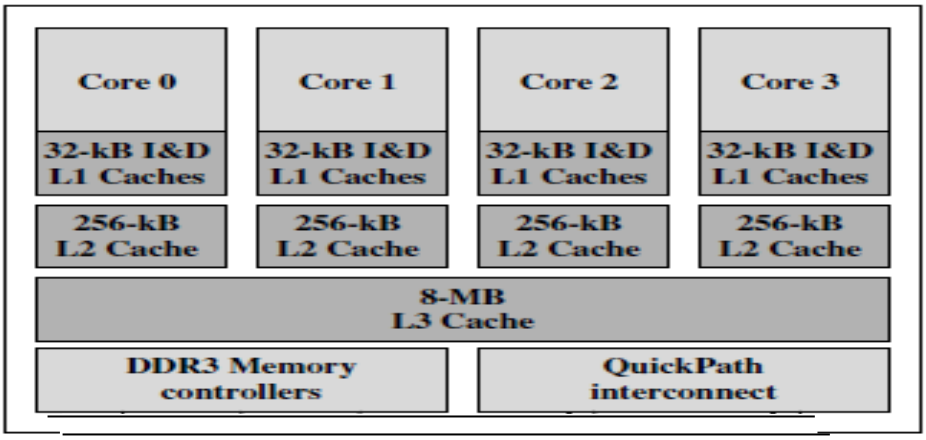
Core 2 due Architecture

- Multicore on chip Core i3 architecture. Tightly coupled.



Question: What is the difference between core 2-Due and core i3?
Sol:
Both Core 2 Due and Core i3 have two CPUs, but the former has Shared Cache L2 and the latter has Shared Cache L3.

- Multicore on chip, Core i7 architecture. Tightly coupled (The Lattice Double Data Rate (DDR3)).
  - I: instruction.
  - D: Data.

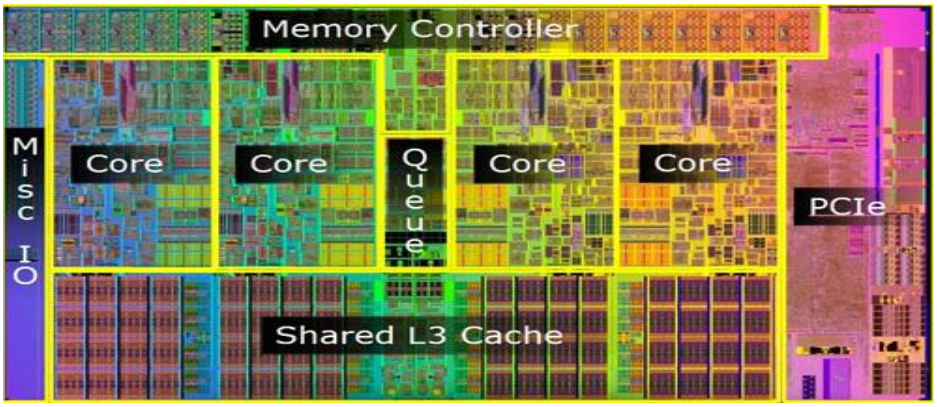


Intel Core i7 Block Diagram

Exercise: How Does Shared Cache communicate with dedicated cache?
Solution: By **MESI** Protocol (**M**odified **E**xclusive **S**hared **I**nvalided).

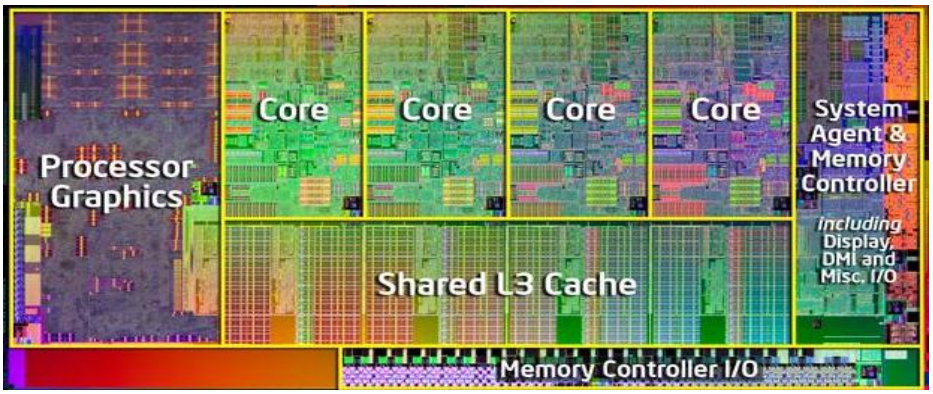
Assignment-1: Write a report on MESI.

- Multi-core on chip, Core i5 & Corei7 architecture. Tightly coupled.



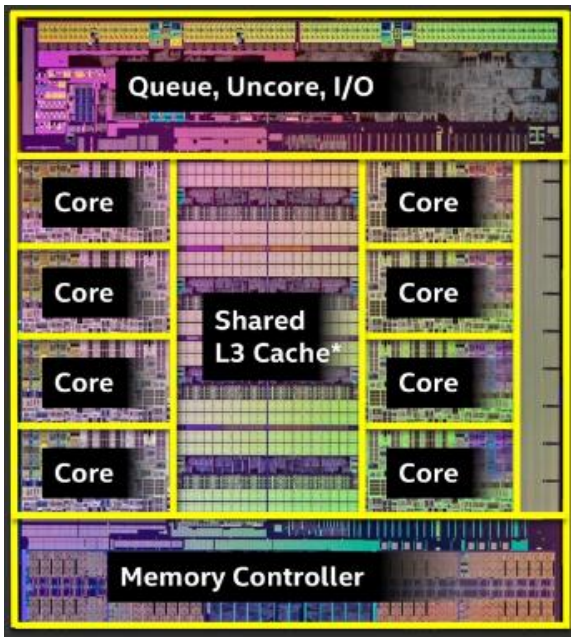
PCIe (peripheral component interconnect express) is an interface standard

Intel Corei5-750



Intel Corei7-2600K

- Multicore on chip, Core i7 architecture., Tightly coupled.



core-i7-5960x

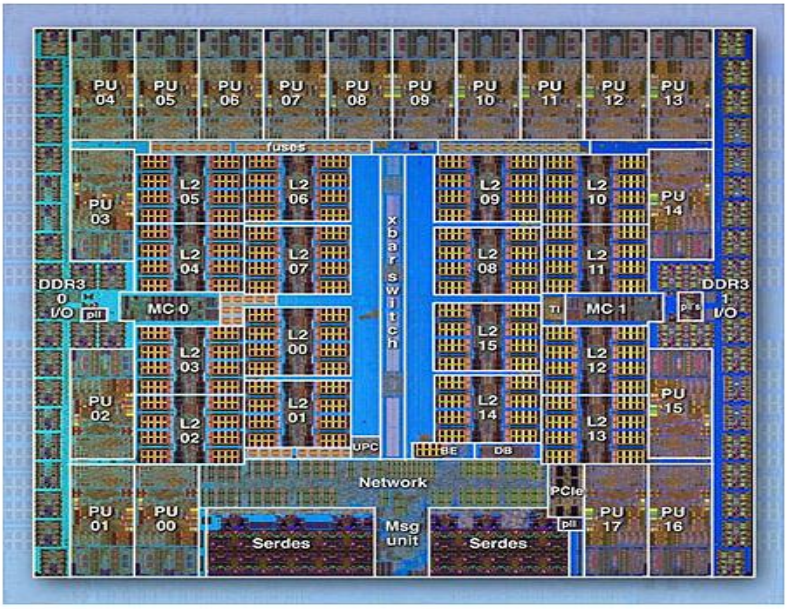
- Comparison between Core-i3, Corei5 and Corei7

	Core i3	Core i5	Core i7
1	Entry level processor.	Mid range processor.	High end processor.
2	2-4 Cores	2-4 Cores	4 Cores
3	4 Threads	4 Threads	8 Threads
4	Hyper-Threading (efficient use of processor resources)	Hyper-Threading (efficient use of processor resources)	Hyper-Threading (efficient use of processor resources)
5	3-4 MB Cache	3-8 MB Cache	4-8 MB Cache
6	32 nm Silicon (less heat and energy)	32-45 nm Silicon (less heat and energy)	32-45 nm Silicon (less heat and energy)
7		Turbo Mode (turn off core if not used)	Turbo Mode (turn off core if not used)

- Multi-core on chip types

Code name	Key products	Cores	Threads	Last-level cache size
Bloomfield	Core i7	4	8	8 MB
Lynnfield	Core i5, i7	4	8	8 MB
Gulftown	Core i7-970, 990X	6	12	12 MB
Sandy Bridge	Core i5, i7	4	8	8 MB
Sandy Bridge-E	Core-i7-39xx	8	16	20 MB
Deneb	Phenom II	4	4	6 MB
Thuban	Phenom II X6	6	6	6 MB
Orochi/Zambezi	FX	8	8	8MB

- IBM 18 Multicore on chip. Tightly coupled.



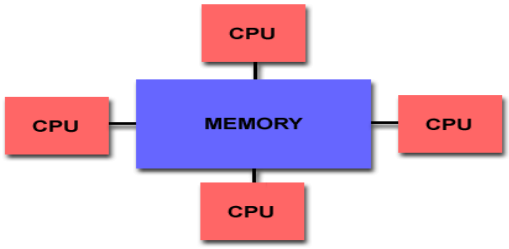
IBM BG/Q Compute Chip with 18 cores (PU) and 16 L2 Cache units (L2)

3.4 Memory architectures:

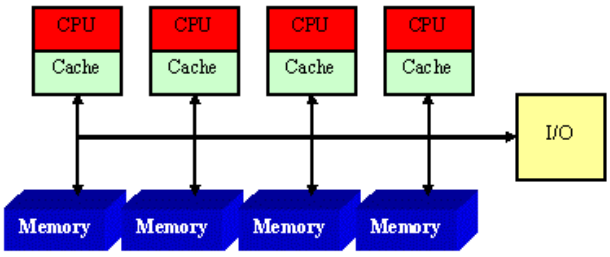
3.4.1 Shared Memory (tightly coupled). Called multi core- multi threading (traditional multiprocessing)

A. Uniform Memory Access (UMA).
B. Non-uniform Memory Access (NUMA).

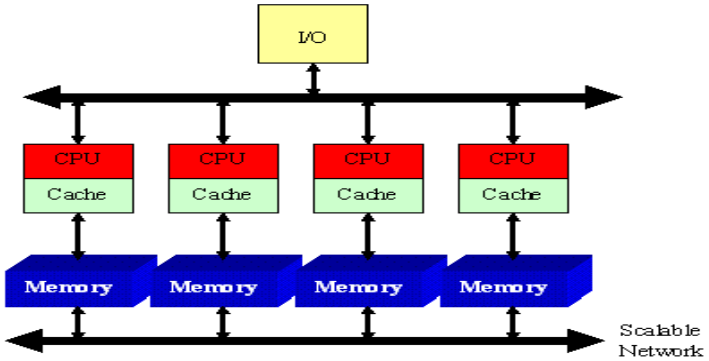
- Multiple processors can operate independently but share the **same memory resource**.
- CPU communicates with shared memory by **Load** & **Store**.
- Shared memory machines can be divided into two main classes based upon memory access times: ***UMA*** and ***NUMA***.
- Shared global memory address space.



1. **Uniform Memory Access (UMA):**
    1. **same latency** to access memory.
    2. Lack of scalability between memory and CPUs.



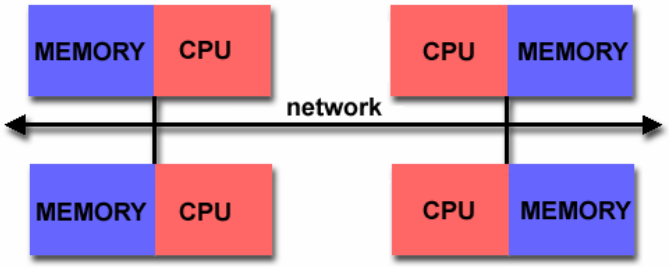
2. **Non-Uniform Memory Access (NUMA):**
    1. Each processor has its own local memory. Different latency to access foreign memory section.
    2. Scalability is better than UMA.



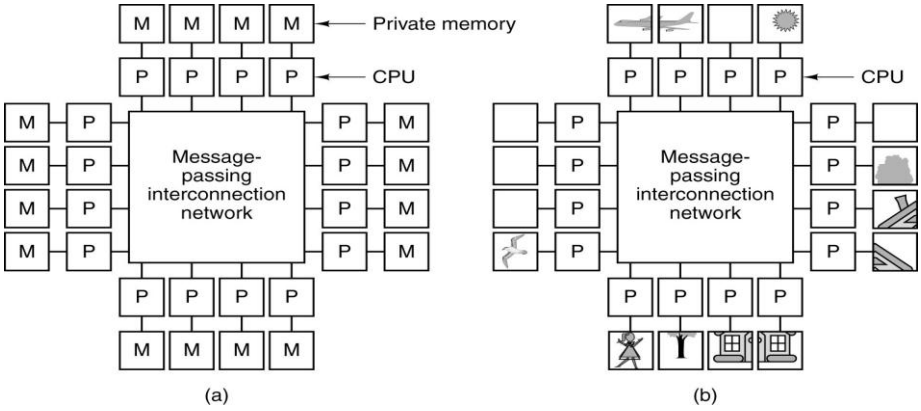
3.4.2 Distributed Memory (loosely coupled).

A. Grid.
B. Cluster.

- Distributed memory systems require a communication network to connect inter-processor memory.
- No shared global memory address space.
- Multicomputer network.
- Usually programmed via message passing: **send** & **receive** for communication.
- Advantage: Memory is scalable with number of processors.

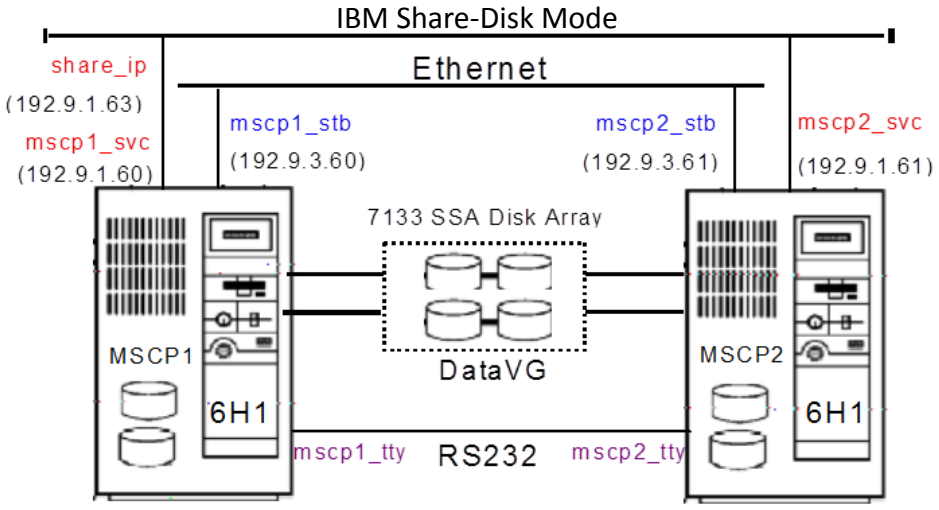


Parallel Processing



A multicomputer with 16 CPUs, each with its own private memory.

1-Cluster: Computers connected over high-bandwidth local area network (Ethernet) used as a parallel computer.

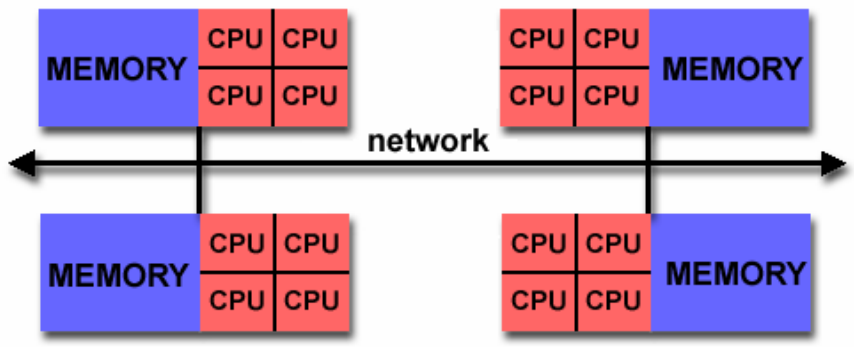


2- Grid: Computers connected over wide area network (Grid computing is the most distributed form of parallel computing). It makes use of computers communicating over the **Internet** to work on a given problem such as cloud computing.

Grid Computers

3.4.3 **Hybride** shared and distributed Memory.

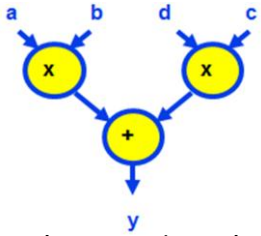
- The largest and fastest computers in the world today employ both shared and distributed memory architectures.
- It is a combination of both shared and distributed memory architecture.
- **Advantage**: Increased scalability.
- **Disadvantage**: Increased complexity in terms of programming.



3.5 Data Flow Computer Architecture.

- Data flow architecture is represented by both: **nodes** and **arrows**.
  A. Node refers to processing operations.
  B. Arrows refer to data flow.

Example-1: From the following data flow architecture, what are the operations that are going to be happenned?



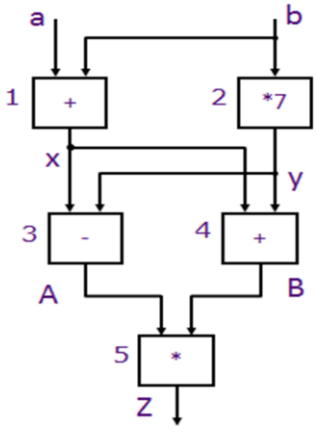
Sol: Process Operation:

y = a*b + c*d

Example-2: From the following data flow architecture, what are the operations that are going to be happenned?

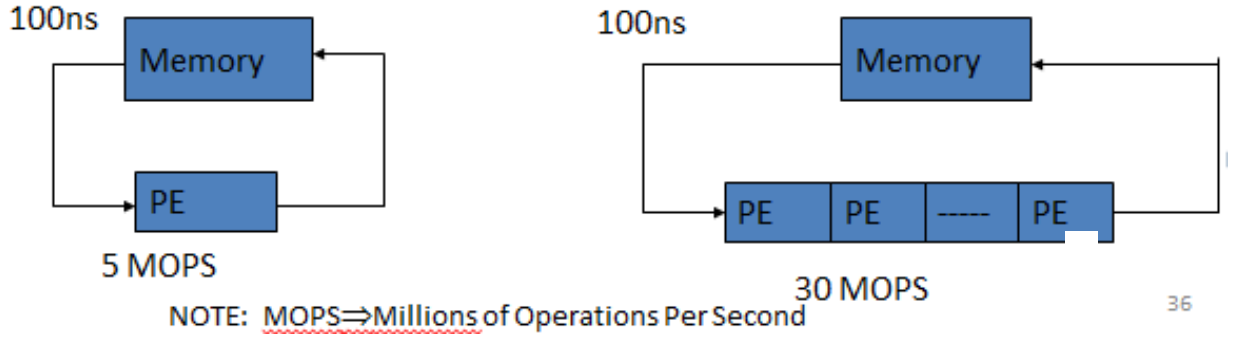
Sol:

{
x = a + b;
y = b * 7
A= x-y
B= x+y
Z= A * B
}



3.6 Systolic Computer Architecture

- The word systolic refers to the "pumping" action of a heart.
- In parallel computer architecture, a systolic array is a homogeneous network of tightly coupled Data Processing Units called cells or Nodes.
- Each Node independently computes a partial result as a function of the data received from its upstream neighbors, stores the result within itself and passes it downstream.



Exercises

1. Draw the following architecture (basic computer, systolic architecture, dataflow architecture).

2. Computer memory connection has many types. List them with drawing each one as a diagram illustrating the processor (CPU) and memory arrangement.

3. Computer wants to add two integer numbers each consisting of 16-bit, the CPU processes only 8-bit word per instruction. How many instruction does CPU need to add these two numbers?

4. In question (3), if the processor has 16-bit data bus processing. How many instructions does the CPU need to add?

5. What is the difference between uniform and non-uniform memory architecture with diagrams.

6. Give two examples of loosely couples and two examples of tightly couples memory.

7. In figure 1:
   A-What kind of computer architecture?
   B- What is the final function form mathematically?

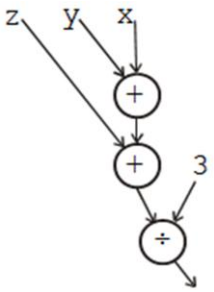


Figure 1.

8. Depict a diagram for the following:
   A- Processor Core 2 due.
   B- Processor Core i7.

9. Define:
   A- Grid Parallel Computer.
   B- Cluster Parallel Computer.

10. CPU chip has 4 multithreading superscalar with two CPUs, how many physical and logical CPUs does this chip have?

11. Pentium 4 has two Multithreading, How many physical and logical CPU does it has?

12. List all the types of multi-core processor organizations with their diagrams.

13. IBM POWER5 chip, which is used in high-end PowerPC products, combines chip multiprocessing with SMT. The chip has two separate processors, each of which is a multithreaded processor capable of supporting two threads concurrently using SMT.
    A- How many physical CPU?
    B- How many logical CPU?
    C- How many MAXIMUM thread can this chip process in the same time?
    D- Draw a diagram explaining the superscalar for this chip.

Performance of Parallel Processing

4.1 Speedup & Efficiency.
4.2 Amdahl's Law.
4.3 Minsky's Conjecture.
4.4 Gustafson's Law.

Objectives:

- How to measures the performance of parallel architectures.
- Understanding various Laws and formulas for analyzing the performance.

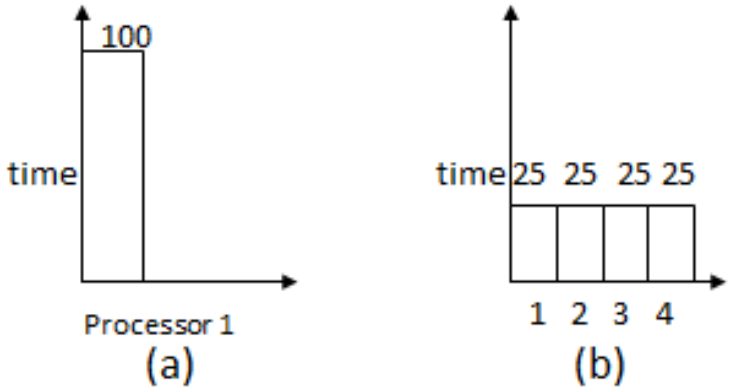
4.1 Speedup & Efficiency:

Performance = Output of the system

- Three ways to improve the performance:
  1. Work harder: → Using faster hardware.
  2. Work smarter: → Doing things more efficiently (algorithms and computational techniques).
  3. Get help: → Using multiple computers to solve a particular task.

Computer Performance: $MIPC = f * IPC$
where: f= frequency.
IPC= Instruction per cycle.

- **Speedup Factor *S(n):*** It is the ratio between the time taken by a single processor to execute a task to the time taken by a parallel system consisting of (*n)* processors to execute the same task.



$S_p = \frac{100}{25} = 0.4$

Perfect parallelization

Parallel Processing

- **What is the best Speedup?**

  Linear speedup:

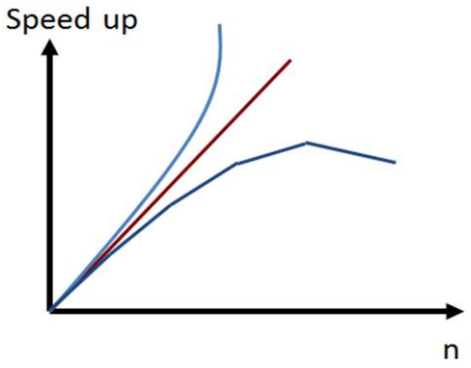
  $S_p = n$

  Superlinear speedup:

  $S_p > n$

  Sub-linear speedup:

  $S_p < n$



- **Speedup Factor S(n) =** $\frac{Sequantion\ running\ time}{Parallel\ running\ time}$

  - *Ts*: Time for Single processor.
  - *Tm*: Time for multi-processor.
  - Tc: Overhead time (time needed for the communicated processors among each other to execute their sub-tasks).

  Sol :

$$S(n) = \text{speedup factor with communication overhead}$$
$$= \frac{t_s}{t_m} = \frac{t_s}{\frac{t_s}{n} + t_c} = \frac{n}{1 + n \times \frac{t_c}{t_s}}$$

  - **Efficiency *(E)*:** it is the achieved speedup of the parallel system divided by the number of the processors (*n*). Efficiency = $\frac{Speedup}{Processor\ (n)}$
  - It is formulated as follows: $\zeta = \frac{1}{1+n \times \frac{t_c}{t_s}}$

Example-1: Consider a Parallel Computer system has 5 processors working together, the time required to execute a task by one processor is 50 ns. Compute the following:

1. Time required to implement same task by 5 processors, consider Tc=0 ns.
2. Parallel system speedup.
3. Efficiency of this system.

Sol:

1. Tm=10 ns.
2. S=5.
3. E= 5/5=1.

$$\text{Efficiency} = \frac{Sequantial\ running\ time}{Processor \times Parallel\ running\ time} = \frac{Speedup}{Processors}$$

- **Factors that limit the achievable Speedup**:
  1. Communication cost.
  2. Load balancing of processors.
  3. Costs of creating and scheduling processes.
  4. I/O operations.

4.2 Amdahl's Law.

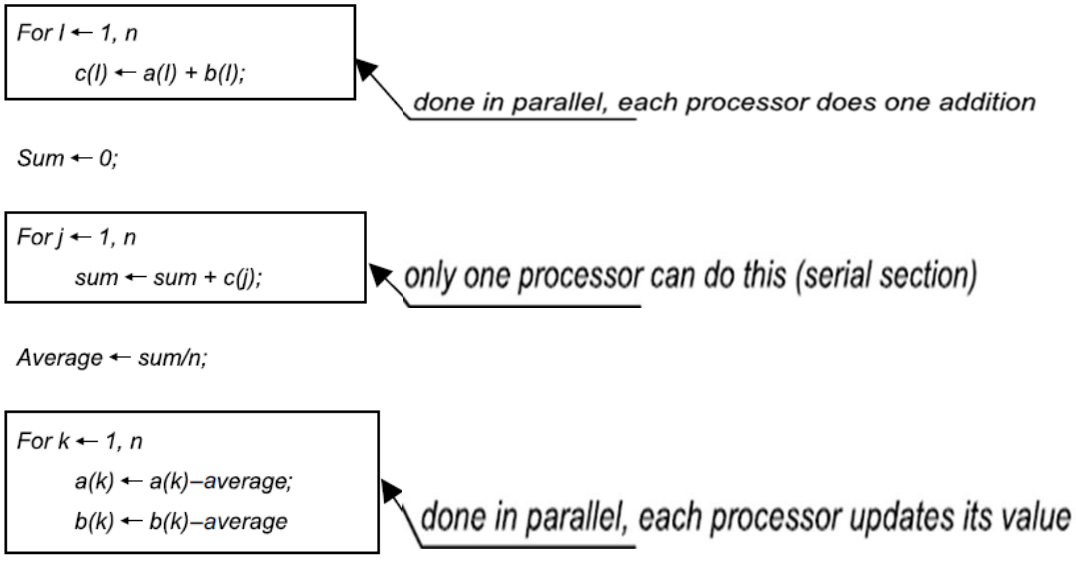
- Gene Amdahl, chief architect of IBM's.
- It is used to measure the system performance improvement.
- Amdahl's Law involved in parallel processing to measure the improved speedup *S(n)* in terms of number of processors (*n*).

Law Definition:

- f= is the fraction of a calculation that is sequential.
- (1-f) = is the fraction that can be parallelized (dividable into concurrent subtask).
- Amdahl's Law Speedup is:

$$S(n) = \frac{t_s}{f t_s + (1-f)\frac{t_s}{n}} = \frac{n}{1 + (n-1) f}$$

What is *f* ?



Example 1: A program code has 90% of a calculation considered as parallelized (i.e. 10% is sequential). What is the maximum speed-up which can be achieved on 5 processors?

Sol:

$$S_5 = \frac{n}{1 + (n-1)\, f} = \frac{5}{1 + (5-1)\times 0.1} = 3.75$$

That means: the program can theoretically run 3.6 times faster on five processors than on one.

Example 2: A program code has 90% of a calculation considered as parallelized (i.e. 10% is sequential). What is the maximum speed-up which can be achieved on 10 processors?

Sol:

$$S_{10} = \frac{n}{1 + (n-1)\, f} = \frac{10}{1 + (10-1)\times 0.1} = 5.3$$

That means: Investing twice as much hardware speeds the calculation up by about 50%.

Example 3: A program code has 90% of a calculation considered as parallelized (i.e. 10% is sequential). What is the maximum speed-up which can be achieved on 20 processors?

Sol:

$$S_{20} = \frac{n}{1 + (n-1)\, f} = \frac{20}{1 + (20-1)\times 0.1} = 6.9$$

That means: doubling the hardware again speeds up the calculation by only 30%.

Example 4: A program code has 90% of a calculation considered as parallelized (i.e. 10% is sequential). What is the maximum speed-up which can be achieved on 1000 processors?.

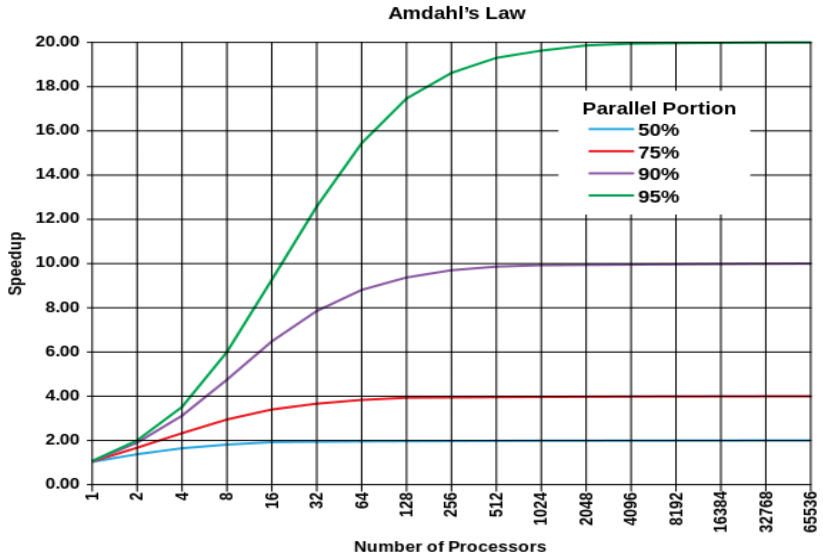
Sol:

$$S_{1000} = \frac{n}{1 + (n-1)\, f} = \frac{1000}{1 + (1000-1)\times 0.1} = 9.9$$

That means: throwing an absurd amount of hardware at the calculation results in a maximum theoretical speed-up of 9.9.

Amdahl 's Law Curve Characteristics

The point, which Amdahl was trying to make, was that using lots of parallel processors was not a viable way of achieving the sort of speed-ups that people were looking for.



4.3 Minsky's Conjecture

Minsky's conjecture: Speedup tends to be proportional to log *n:*

$$S\,(n) = \log_2 (n)$$

4.4 Gustafson's Law

- Speed up according to Gustafson's Law measured by scaling the problem to the processor number.

Scaled speed up SS(n)
f= sequential part of the program.
P= parallel part of the program.

$$\text{SS (n)} = \frac{S + P \times n}{S + P} = \text{n} + (1 - \text{n}) \times f$$

Example: A program code has 90% of a calculation considered as parallelized (i.e. 10% is sequential). What is the maximum speed-up which can be achieved on 20 processors by using Gustafson's Law:

Solution:

f= sequential part of the program= 0.1
N= number of the parallel processors=20.

$$\text{SS (n)} = \text{n} + (1 - \text{n}) \times f = 20 + (1 - 20) \times 0.1 = 18.1$$

It means: if computer used 20 parallel processors, the speedup will be 18.1 times faster than using one processor. As compared with Amdahl's Law, the speedup is 6.9.

Summary:

- Performance factors have been discussed.
- Speed up
- Efficiency.
- Amdahl ‘s Law
- Minsky’s law
- Gostafson Law

Exercises:

1. Given a (scaled) speed up of 20 on 32 processors, what is the serial fraction if this speedup measured by using:
    a. Amdahl’s law.
    b. Gustafson’s Law.

2. An oceanographer gives you a serial program and asks you how much faster it might run on 8 processors. You can only find one function amenable to a parallel solution. Benchmarking on a single processor reveals 80% of the execution time is spent inside this function. What is the best speedup a parallel version is likely to achieve on 8 processors?

3. 95% of a program’s execution time occurs inside a loop that can be executed in parallel. What is the maximum speedup we should expect from a parallel version of the program executing on 8 CPUs by using:
    A. Amdahl's Law?
    B. Minsky’s conjecture.
    C. Gustafson’s law.

4. **List** all the types of speed up as well as **draw** their graph characteristic. Also, which one is the best type in terms of performance?

Pipeline Processing

5.1 Introduction

Pipeline: Is a technique of decomposing a sequential process into sub-processes, with each sub-process being executed in a special dedicated segment that operates concurrently with all other segments.

5.2 What is Pipelining concept in computer architecture?

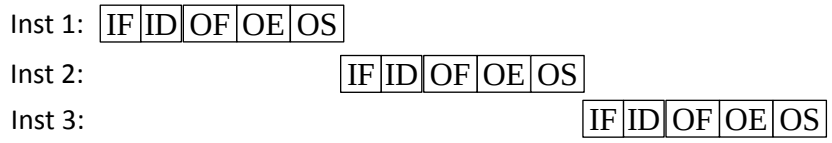
- Pipelining is a technique where multiple instructions are overlapped during execution.
- It is divided into 5 stages and these stages are connected with one another to form a pipe like structure. It allows storing and executing instructions in an orderly process.
- In pipeline system, each segment consists of an input register followed by a combinational circuit.
- Pipelining increases the overall instruction throughput.

5.3 What are the 5 stages of Pipelining with short explanation?

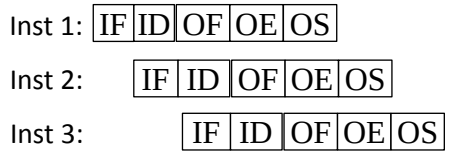
1. **Interaction Fetch** (Read instruction from memory)
2. **Interaction decode**: (after fetching instruction from memory, the processor will decode the instruction, to get update instruction).
3. **Operand fetch:** Get idea about number of operands (data required for instruction to be executed).
4. **Instruction execute:** To produce the result.
5. **Operand store:** Store result (write the result in memory).

5.4 What is the different between Non-Pipelining and Pipelining explain your answer by giving example with 3 instructions

In Non-Pipelining, it will not start new instruction unless finish the five cycles of previous instruction (Instruction wise interleave).



In Pipelining, it will start new instruction as soon as finish the first cycle of the previous instruction (phase wise interleave).



For three instructions it required 15 cycles in Non-Pipelining, where it requires 7cycles in Pipelining which saved time.

5.5 What are the 4 parameters important in Pipelining?

Speed up ratio, Frequency, Efficiency, Throughput

5.6 What are the advantages and disadvantage of Pipelining?

Advantages:

1. The time cycle of the processor is reduced.
2. It increases the throughput of the system
3. It makes the system reliable.

Disadvantages:

1 The design of pipelined processor is complex and costly to manufacture.
2 The instruction latency is more.

5.7 Pipeline Performance

Pipeline Speed up (S) = $\frac{nt_n}{(k+n-1)t_p}$

Where: t_n = Non-pipeline time, t_p = Pipeline time, t = Clock cycle time, k = Segment, and n= Number of instruction

$t_n = k * t_p$

Question: Prove mathematically that the pipeline performance (S) equals to the number of segment (K), $S = K$.

Sol: (S) = $\frac{nt_n}{(k+n-1)t_p}$

As the number of tasks increases, n becomes much larger than $k - 1$, and $k + n - 1$ approaches the value of n. Under this condition, the speedup becomes

$$S = \frac{t_n}{t_p}$$

If we assume that the time it takes to process a task is the same in the pipeline and nonpipeline circuits, we will have $t_n = kt_p$. Including this assumption, the speedup reduces to

$$S = \frac{kt_p}{t_p} = k$$

This shows that the theoretical maximum speedup that a pipeline can provide is k, where k is the number of segments in the pipeline.

Example:

A computer system uses pipeline technique to process computer instructions. Assume the time it takes to process an operation in each segment is equal to t_p=20 ns. Assume this pipeline has k=4 segments and executes n=100 instructions in sequence. Answer the following:

1- How long time to execute 100 instructions does this computer take with this pipeline?
2- How long time does it take with non-pipeline, (Tnp)?
3- What is the speed up.

Sol:

1- $T_p = (n + k - 1)t = (100 + 4 - 1)*20$ ns = 2060 ns.
2- Tnp = k * t_p = 4 * 20= 80 ns.
3- Non- pipleine system time = n k t_p = 100 * 4 * 20= 8000 to complete 100 instruction.
4- Speed up= Tnp/tp= 8000/2060= 3.88.

Interconnection Network

6.1 Definition and Goal

Interconnection networks carry data between processors and to memory.

- Interconnections are made of:
  1- Switches.
  2- Links: Wires & Fiber.

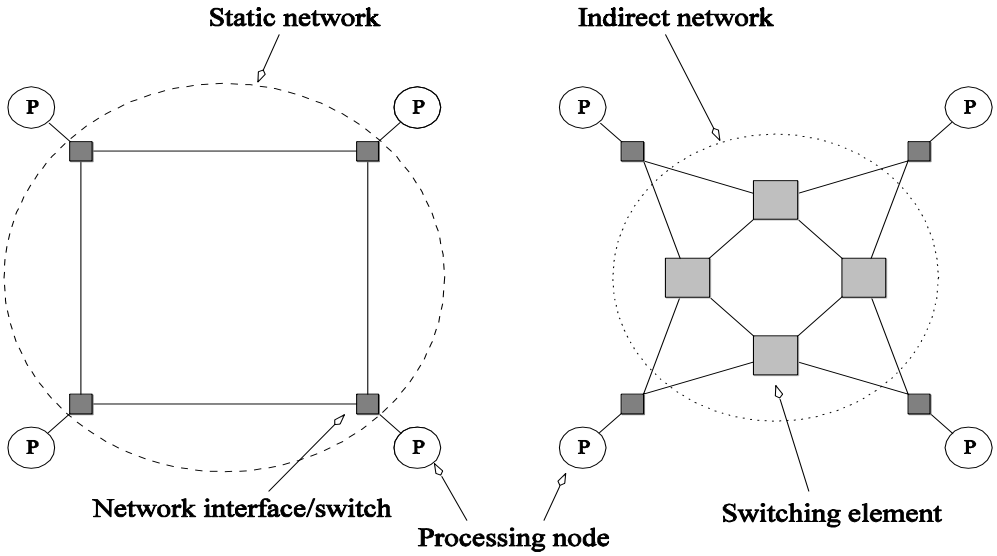
- A network is characterized by:
  1- **Topology:** physical interconnection structure of the network graph.
  2- **Routing algorithm:** route of messages may follow through the network graph.
  3- **Switching strategy:** How the data in a message traverses its route A- Circuit switch, 2- packet switches.
  4- **Flow control mechanism:** determines when the message, or portions of it, move along its route.

- Terminology:
  1- ***Diameter:*** The distance between the farthest two nodes in the network.
     - The diameter of a linear array is $p - 1$,
     - a mesh is $2(sqrt(p) - 1)$,
     - a tree and hypercube is $log\ p$,
     - a completely connected network is $O(1)$.
  2- ***Bisection Width:*** The minimum number of wires you must cut to divide the network into **two equal parts**. The bisection width of:
     - a linear array and tree is 1,
     - a mesh is sqrt(p),
     - a hypercube is $p/2$,
     - a completely connected network is $p^2/4$.
  3- ***Cost:*** The number of links or switches is measured of the cost.

6.2 Types of Interconnection networks

- Interconnects are classified as static or dynamic.

1. Static Networks consist of point-to-point communication links among processing nodes and are also referred to as direct networks.
2. Dynamic Networks are built using switches and communication links. Dynamic networks are also referred to as indirect networks.



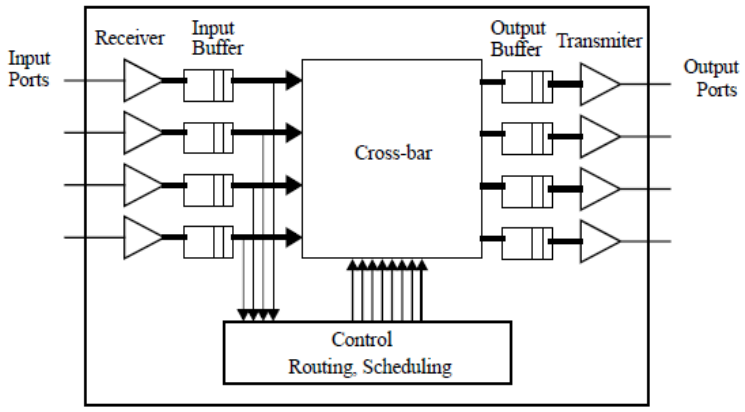
Classification of interconnection networks: a static network (left) and Dynamic network (right).

6.3 Switch and its types

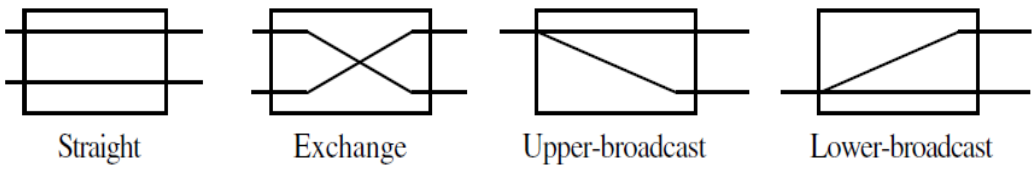
- Switches: map a fixed number of inputs to outputs.
- The total number of ports on a switch is the degree of the switch

Switch Consists of:

1. Input ports.
2. Receiver.
3. Input Buffer.
4. Cross-bar.
5. Output buffer.
6. Transmitter.
7. Output ports.
8. Control Routing.



Switch types are the following:



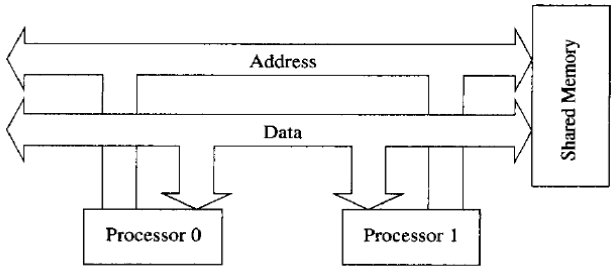
6.4 Network Topologies

- A variety of network topologies have been proposed and implemented.
- These topologies tradeoff performance for cost.
- Commercial machines often implement hybrids of multiple topologies for reasons of packaging, cost, and available components.

Parallel Processing

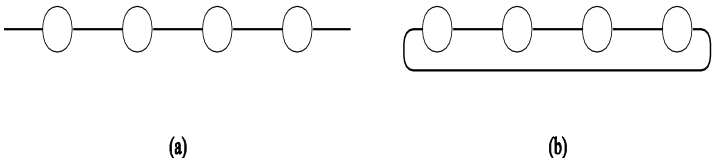
Topology: Network shape or type of node connections.

1. **Buses:** Since much of the data accessed by processors is local to the processor, a local memory can improve the performance of bus-based machines.



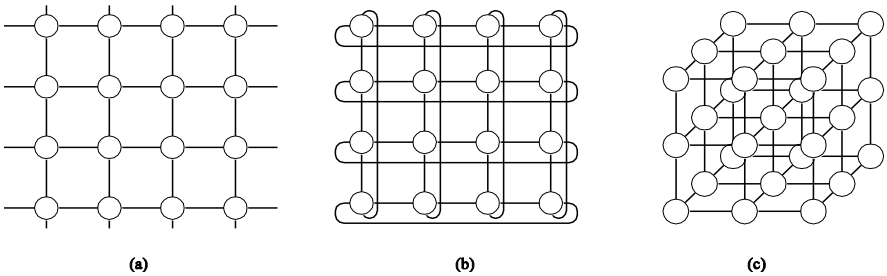
- Some of the simplest and earliest parallel machines used buses.
- All processors access a common bus for exchanging data.
- The distance between any two nodes is O(1) in a bus. The bus also provides a convenient broadcast media.
- The bandwidth of the shared bus is a major bottleneck.

2. Linear Arrays:
   a. Every node has two neighbors (except terminal nodes).
   b. Every node has exactly two neighbors.



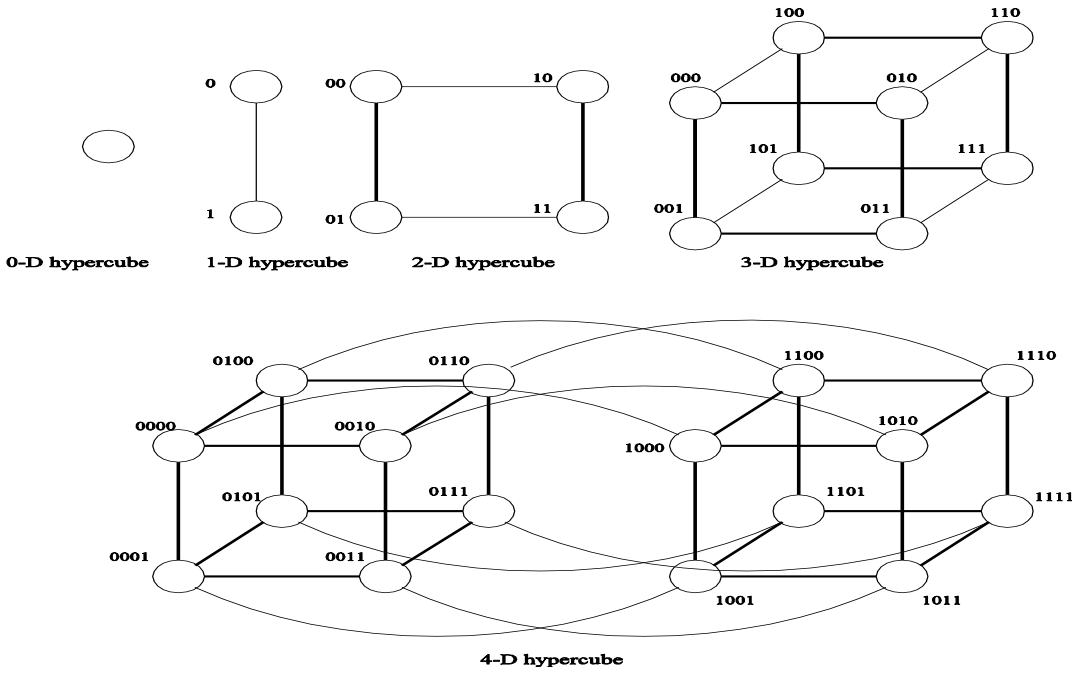
Linear arrays: (a) with no wraparound links; (b) with wraparound link (Ring).

3. Two and Three Dimensional Meshes (MESH + TORUS:)



Two and three dimensional meshes: (a) 2-D mesh with **no** wraparound; (b) 2-D mesh with wraparound link (2-D torus); and (c) a 3-D mesh with no wraparound.

4. Hypercubic (0,1,2,3,4-Dimenrional):



Construction of hypercubes from hypercubes of lower dimension.

Hypercubes: Properties:

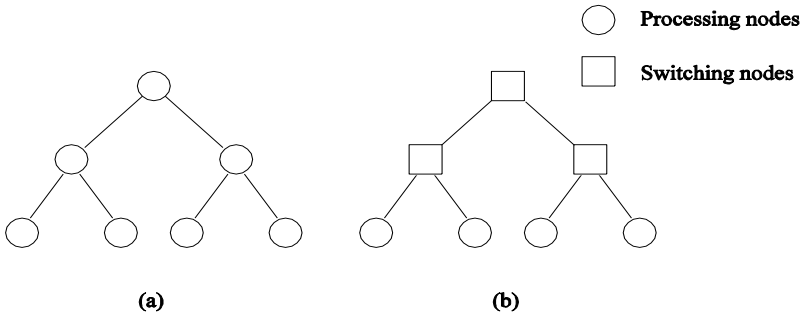
- *Dimension (d) is: d = log p, where p is the total number of nodes.*
- *The distance between any two nodes is at most log p.*
- *Each node has log p neighbors.*

Example: Find number of dimensions of a hypercube interconnection network that has 16 Processor and draw it.

Solution: !!!!

5. Tree:

- Links higher up the tree potentially carry more traffic than those at the lower levels.
- Trees can be laid out in 2D with no wire crossings. This is an attractive property of trees.



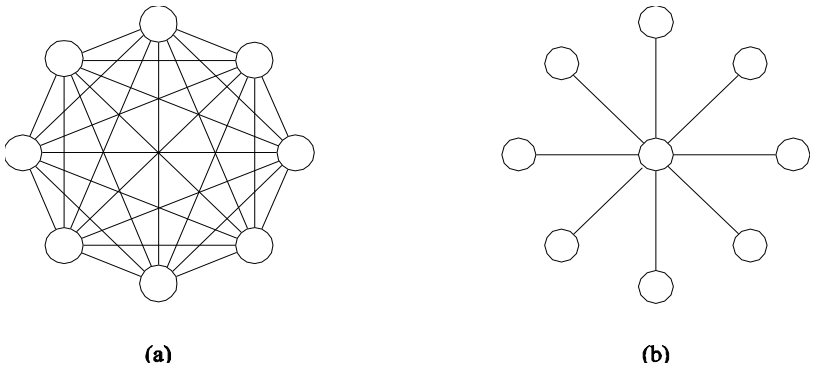
Complete binary tree networks: (a) a static tree network; and (b) a dynamic tree network.

6. Completely Connected
   - Each processor is connected to every other processor.
   - The number of links in the network scales as *O(n(n-1)/2).*
   - While the performance scales very well, the hardware complexity is not realizable for large values of *p*.

   Example of an 8-node completely connected network.

7. Star Connected Networks (b).
   - Every node is connected only to a common node at the center.
   - Distance between any pair of nodes is *O(1).* However, the central node becomes a bottleneck.

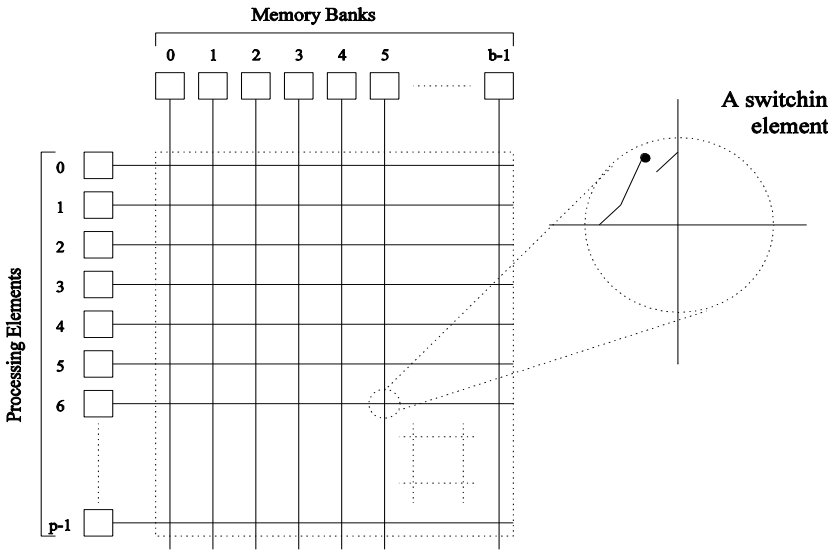
   Example: 9-nodes as star network.



(a) A completely-connected network of eight nodes; (b) a star connected network of nine nodes.

8. Crossbars:

   A crossbar network uses an $p \times m$ grid of switches to connect p inputs to m outputs.



A completely crossbar network connecting *p* processors to b memory banks.

- The cost of a crossbar of *p* processors grows as $O(p^2)$.
- Crossbars have excellent performance scalability but poor cost scalability.
- Buses have excellent cost scalability, but poor performance scalability.
- Multistage interconnects strike a compromise between these extremes.

9. Multistage Networks: Omega network

- A multistage network connects a number of processors to a number of memory banks, via a number of switches organized in layers such as **Omega, Butterfly**, or Banyan networks etc.
- One of the most commonly used multistage interconnects is the Omega network.
- This network consists of *log p* stages, where *p* is the number of inputs/outputs.
- Omega network has P/2 * log(P) switches, so the cost of this network is lower than the crossbar network.

  No. Of switches= p/2 log2 (p).

  No. of stages:= log2(p).

  No of Switch in each node = p/2.

  Note : Connection of each stage is the same as next stage.

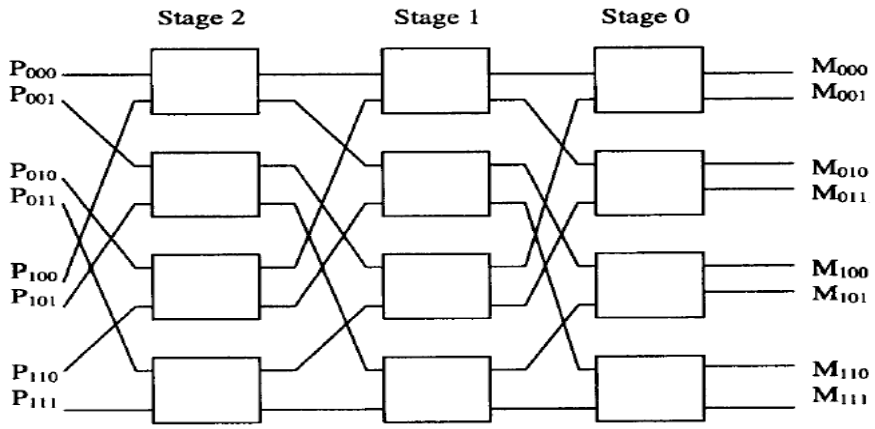
Example: Construct a dynamic network consisting of 8 processors to be connected with 8 memories, by using Multistage Omega network construction.

Solution: Omega network connecting 8 processors to 8 memory.

No. of stages: Log 8 = 3 stages

p/2= 8/2 = 4 switches in each stage.

Totally = p/2 *log p = 8/2 log 8= 12, as: How to test the drawing ?



Way of connections:

Step 1- Input with stages:

- MSB if 0 → Zero switch port.
- MSB if 1 → One switch port.

Step 2- Between Stages:

- Divides in two parts (upper & lower): Upper part is connected to 0
- port of the switch. Lower part is connected to 1 switch port.

- Tests: 0=0

1=1

.

.

.

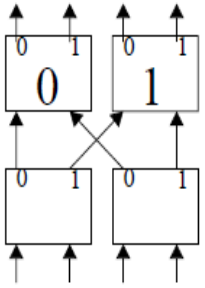
7=7

10. Butterfly Network:

It is a multistage Interconnection network, building-block of the butterfly is simply obtained by crossing one of each pair of edges, as illustrated in the figure below:

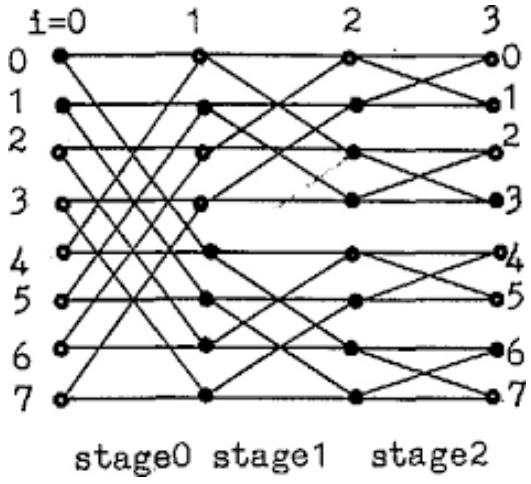
Butterfly characteristics:

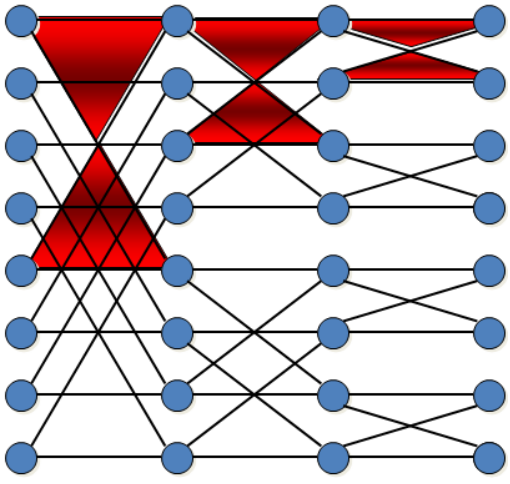
- No of Switches= N (log N +1)
- No of stages= log N +1
- No Switches per Stage=N



Basic Butterfly Building-block

Example:





6.5 Networks Evaluation.

Static Interconnection Networks

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Completely-connected	1	$p^2/4$	$p-1$	$p(p-1)/2$
Star	2	1	1	$p-1$
Complete binary tree	$2\log((p+1)/2)$	1	1	$p-1$
Linear array	$p-1$	1	1	$p-1$
2-D mesh, no wraparound	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	$2(p-\sqrt{p})$
2-D wraparound mesh	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	$2p$
Hypercube	$\log p$	$p/2$	$\log p$	$(p\log p)/2$
Wraparound k-ary d-cube	$d\lfloor k/2\rfloor$	$2k^{d-1}$	$2d$	dp

Evaluating Dynamic Interconnection Networks

Network	Diameter	Bisection Width	Arc Connectivity	Cost (No. of links)
Crossbar	1	p	1	p^2
Omega Network	$\log p$	$p/2$	2	$p/2$
Dynamic Tree	$2\log p$	1	2	$p-1$

Exercises

1. Draw the following interconnection networks topologies using 4 processors in the drawing.
   a. bus.
   b. hypercube.
   c. mesh.
   d. fully connected.
   e. crossbar switch.
2. You are as a computer expert working in Intel company asked to connect a 8 CPUs with 8 Memories as Tightly coupled connection type. Draw the diagram with illustrating, using omega IN.

3. How many switches are there in a crossbar network that connect 5 processors to 5 memories module?

4. Construct a network for connecting 4 processors to 4 memory banks by using omega switching network.

5. Design a network for connecting 8 processor to 8 memory banks by using butterfly switching network. Then, answer the following:
    a. Compute number of stages.
    b. Compute the number of switches in each stage.
    c. Copmute total number of switches in the network.
    d. Draw this network connection.