National Institute of Technology Karnataka Surathkal
Department of Information Technology

IT 301 Parallel Computing

Memory Models, Flynn's Classification

Dr. Geetha V

Assistant Professor

Dept of Information Technology

NITK Surathkal

Course Outline

Course Plan: Theory:

Part A: Parallel Computer Architectures

Week 1,2,3: ***Introduction to Parallel Computer Architecture:*** Parallel Computing, Parallel architecture, bit level, instruction level , data level and task level parallelism. Instruction level parelllelism: pipelining(Data and control instructions), scalar and superscalar processors, vector processors. Parallel computers and computation.

Week 4,5: Memory Models: UMA, NUMA and COMA. Flynns classification, Cache coherence,

Week 6,7: Amdahl's Law. Performance evaluation, Designing parallel algorithms : Divide and conquer, Load balancing, Pipelining.

Week 8 -11: ***Parallel Programming techniques like Task Parallelism using TBB, TL2, Cilk++ etc. and software transactional memory techniques.***

Course Outline

Part B: OpenMP/MPI/CUDA

Week 1,2,3 : ***Shared Memory Programing Techniques:*** Introduction to OpenMP : Directives: parallel, for, sections, task, single, critical, barrier, taskwait, atomic. Clauses: private, shared, firstprivate, lastprivate, reduction, nowait, ordered, schedule, collapse, num_threads, shared, if().

Week 4,5: ***Distributed Memory programming Techniques:*** MPI: Blocking, Non-blocking.

Week 6,7 : CUDA : OpenCL, Execution models, GPU memory, GPU libraries.

Week 10,11,: ***Introduction to accelerator programming using CUDA/OpenCL and Xeon-phi. Concepts of Heterogeneous programming techniques.***

Practical:

Implementation of parallel programs using OpenMP/MPI/CUDA.

Assignment: Performance evaluation of parallel algorithms (in group of 2 or 3 members)

Index

1. Flynn's classification
2. Memory models
3. Perspective on parallel programming

1. Flynn's Classification

- In 1966, M.J. Flynn proposed a classification for the organization of a computer system by the number of instructions and data items that are manipulated simultaneously.
- Flynn uses the stream concept for describing a machine's structure.
- The sequence of instructions read from memory constitutes an **instruction stream**.
- The operations performed on the data in the processor constitute a **data stream**.

Flynn's Classification

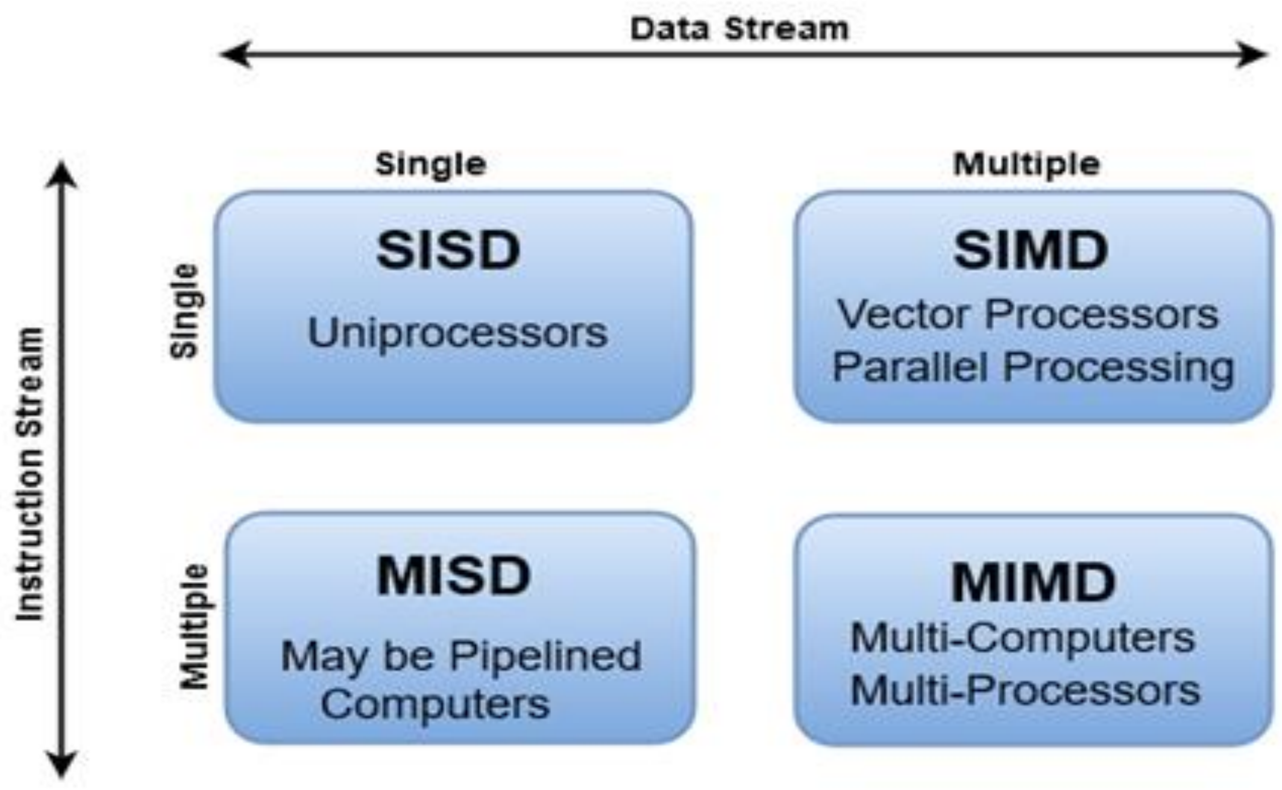
The classification of computer architectures based on the number of instruction streams and data streams (Flynn's Taxonomy)

Flynn's Classification divides computers into four major groups .

- Single instruction stream, single data stream (SISD)
- Single instruction stream, multiple data stream (SIMD)
- Multiple instruction stream, single data stream (MISD)
- Multiple instruction stream, multiple data stream (MIMD)

Flynn's Classification

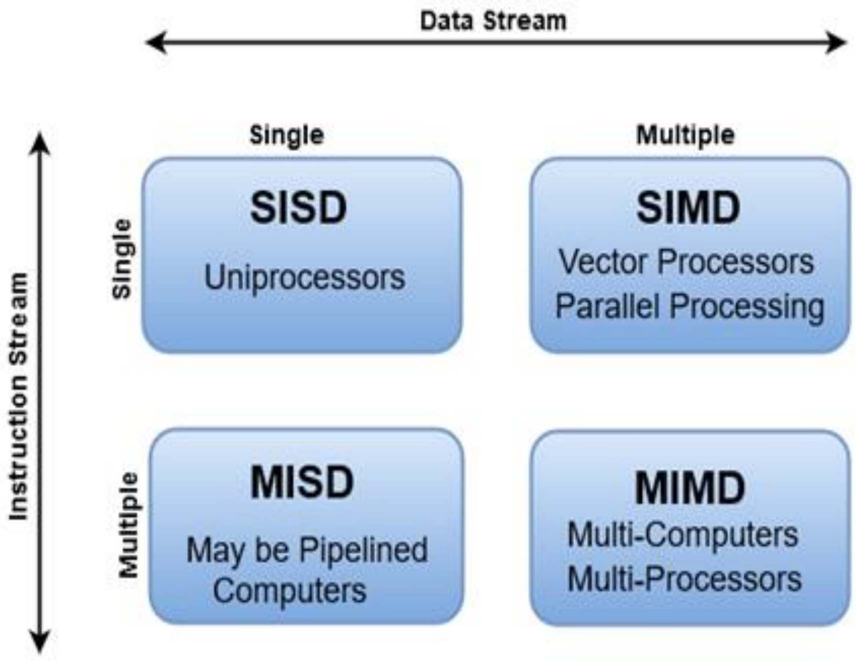
Flynn's Classification divides computers into four major groups .



Ref:https://www.javatpoint.com/flynns-classification-of-computers

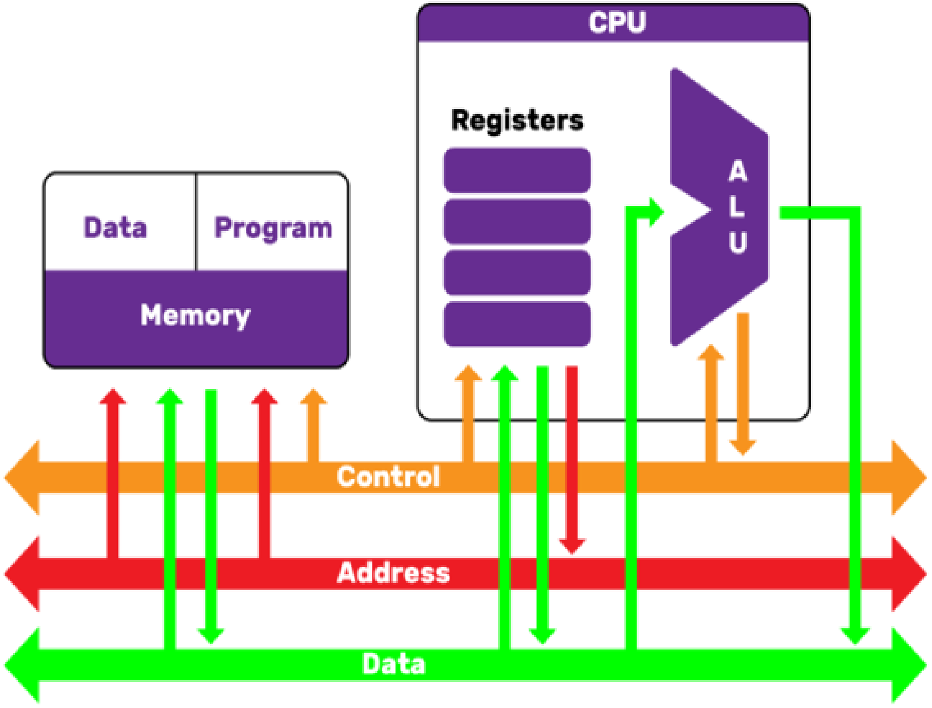
Flynn's Classification: SISD

Flynn's Classification



Ref:https://www.javatpoint.com/flynns-classification-of-computers

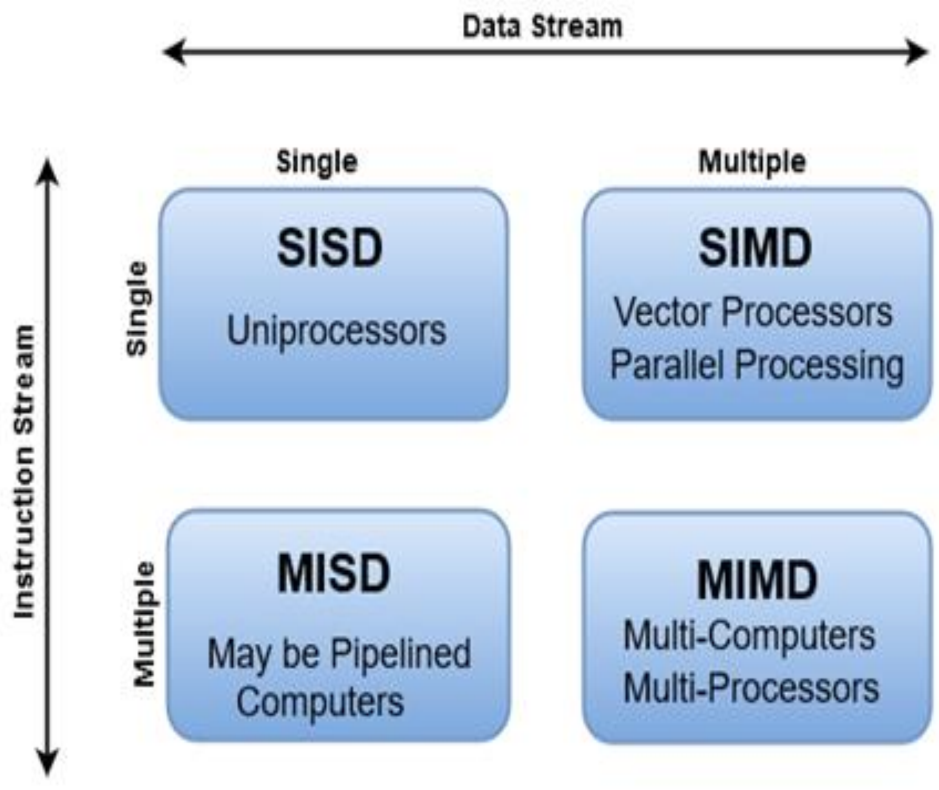
Example: Traditional Computers



Ref:https://www.futurelearn.com/courses/how-computers-work/0/steps/49283

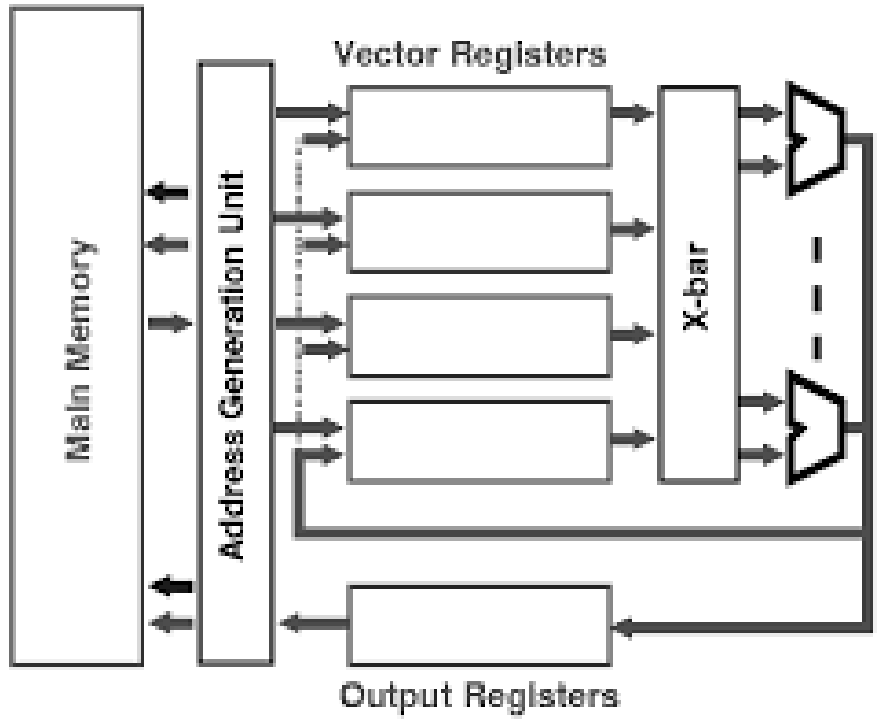
Flynn's Classification: SIMD

Flynn's Classification



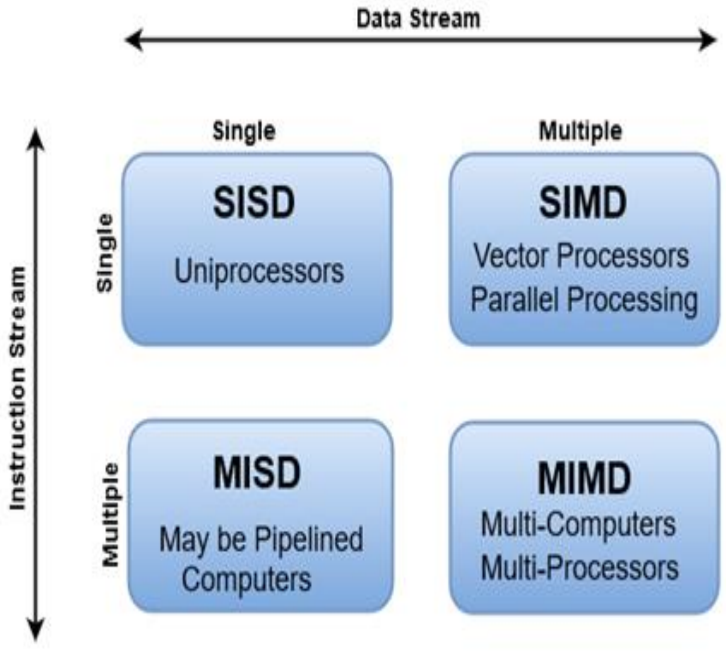
Ref:https://www.javatpoint.com/flynns-classification-of-computers

Example: Vector processor CRAY -I



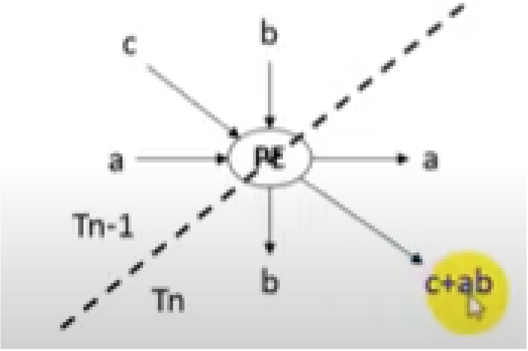
Flynn's Classification: MISD

Flynn's Classification

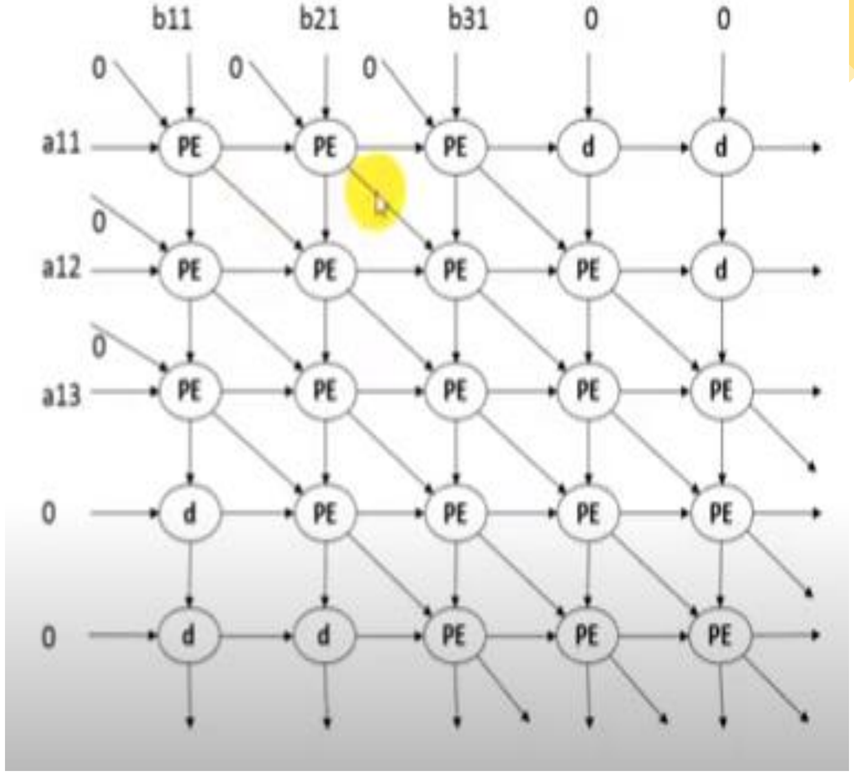


Example: Systolic arrays : eg: 3 x 3 matrix multiplication

T1 Cycle



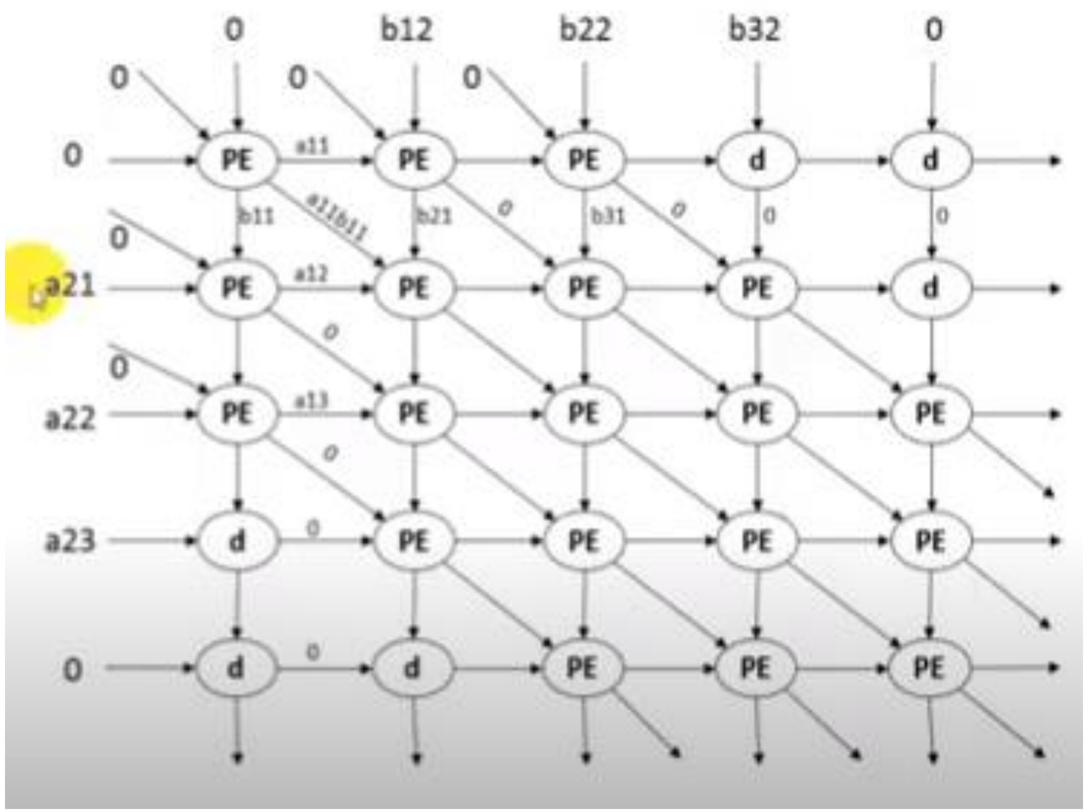
c11 = c11 + a11b11 + a12b21 + a13b31
c12 = c12 + a11b12 + a12b22 + a13b32
c13 = c13 + a11b13 + a12b23 + a13b33
c21 = c21 + a21b11 + a22b21 + a23b31
c22 = c22 + a21b12 + a22b22 + a23b32
c23 = c23 + a21b13 + a22b23 + a23b33
c31 = c31 + a31b11 + a32b21 + a33b31
c32 = c31 + a31b12 + a32b22 + a33b32
c33 = c33 + a31b13 + a32b23 + a33b33



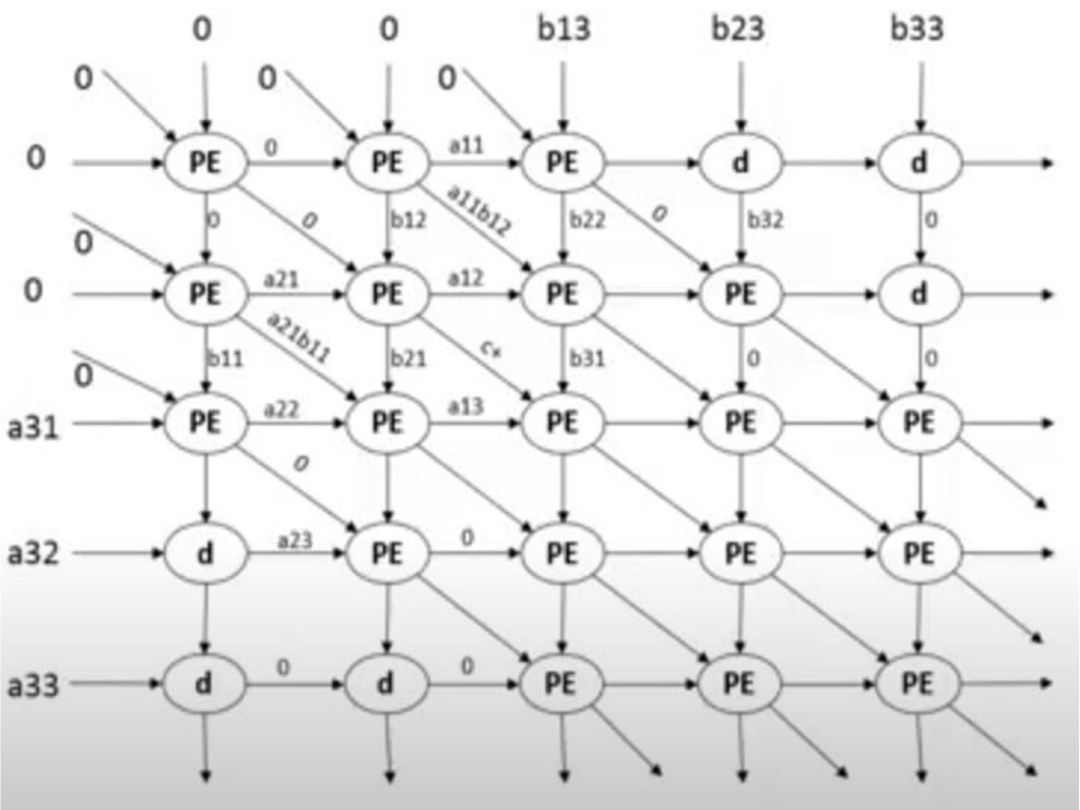
Ref:https://www.javatpoint.com/flynns-classification-of-computers

https://www.youtube.com/watch?v=vADVh1ogNo0

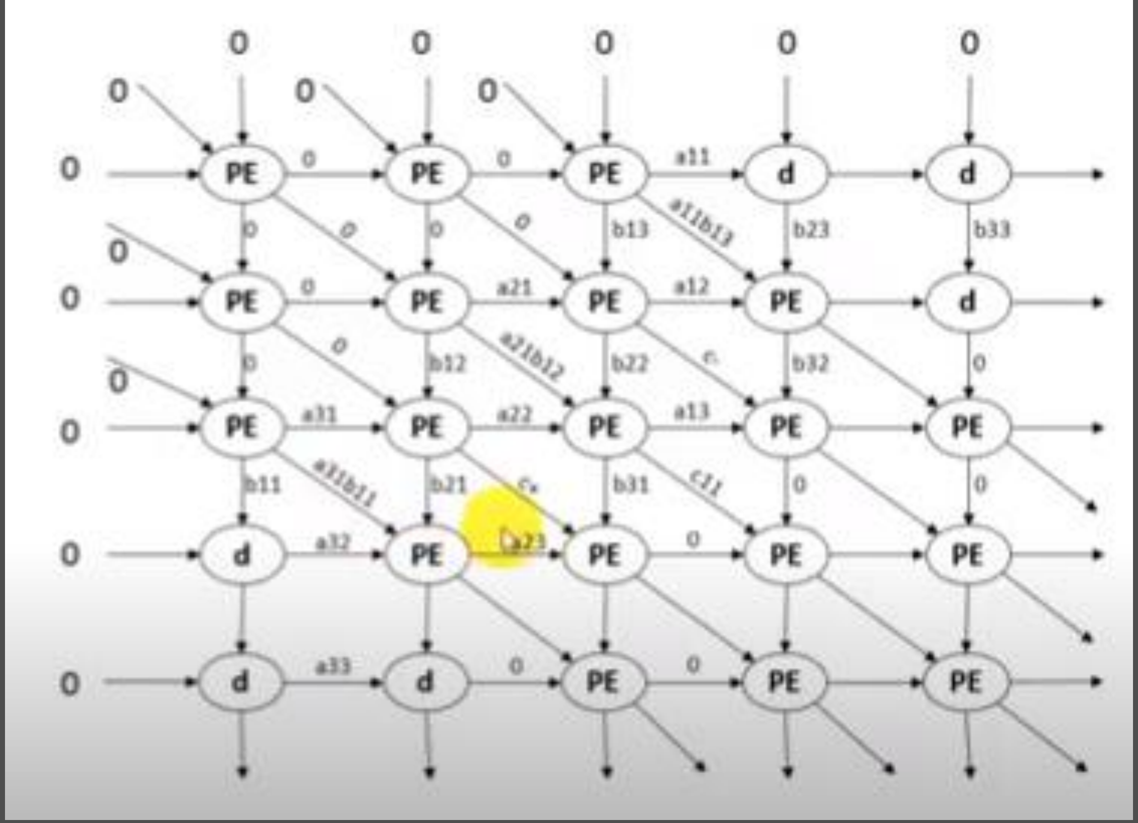
T2 Cycle



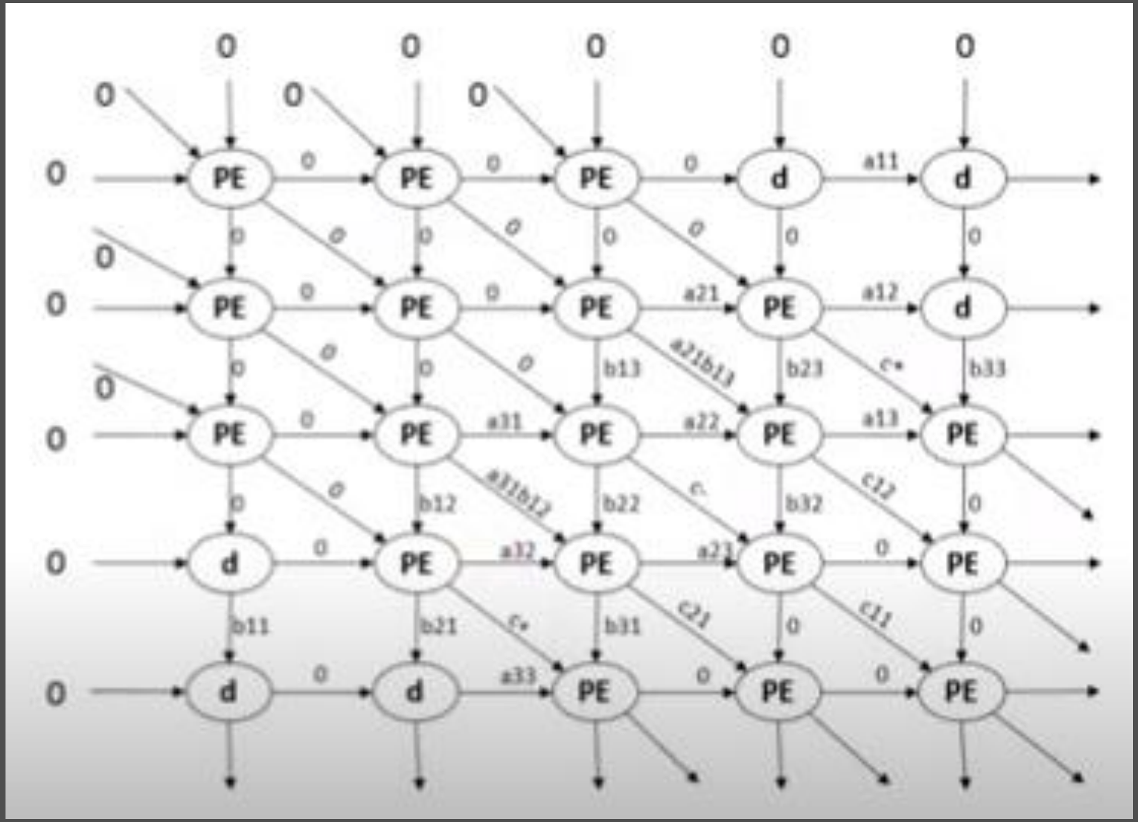
T3 Cycle



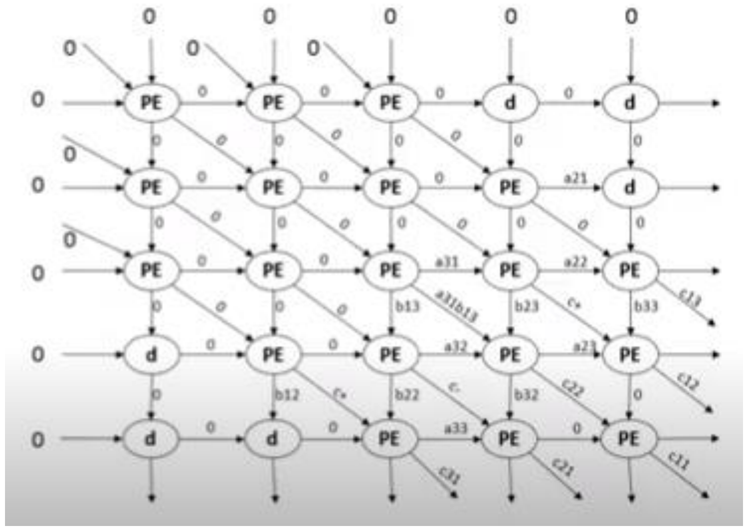
T4 Cycle



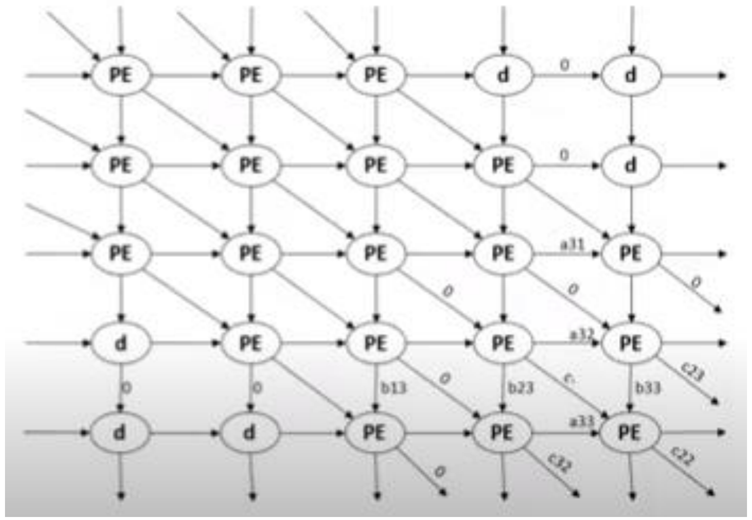
T5 Cycle



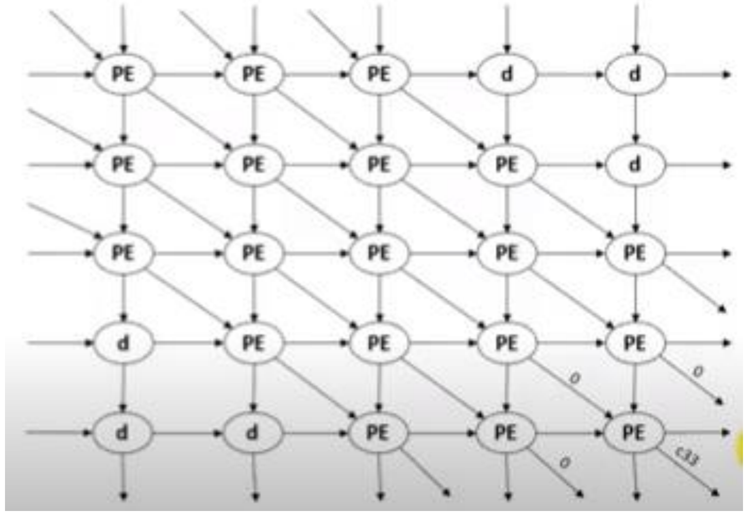
T6 Cycle



T7 Cycle

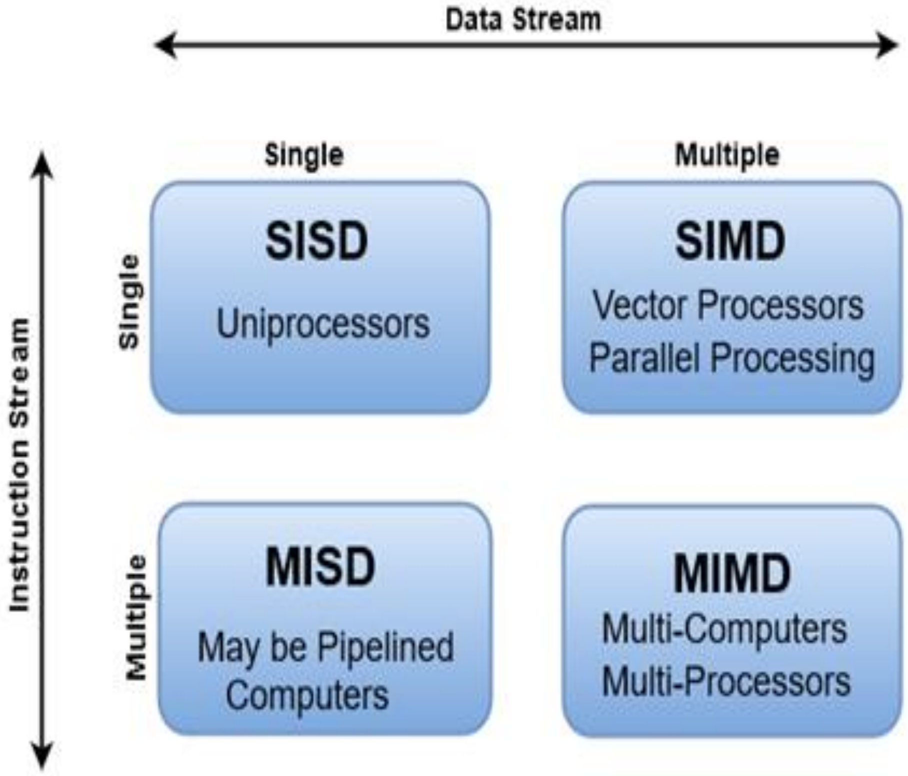


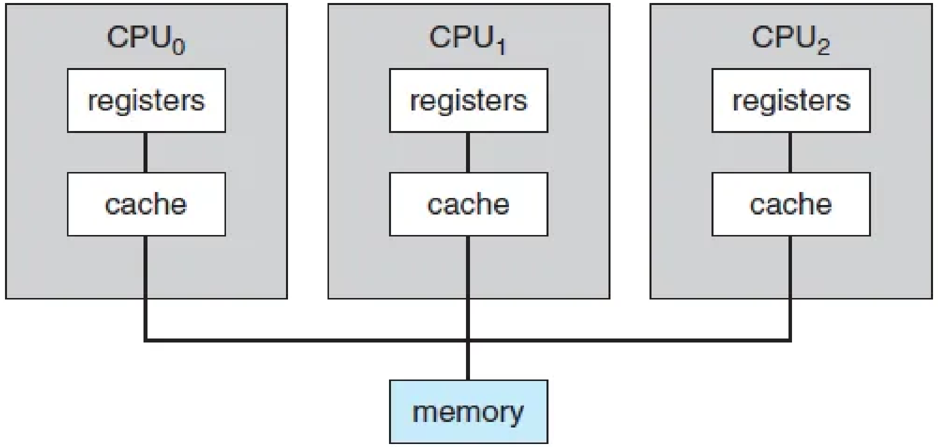
T8 Cycle

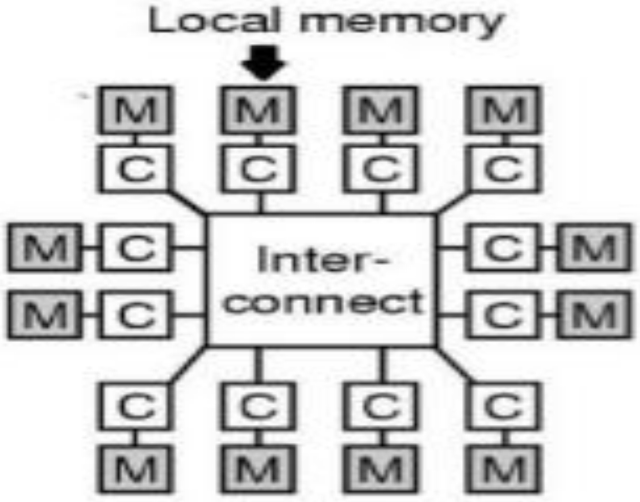


Flynn's Classification MIMD

Flynn's Classification







Ref:https://www.javatpoint.com/flynns-classification-of-computers

2.Memory Models

Parallel Computers Architectural Model/ Physical Model

Distinguished by having-

1. Shared Common Memory:

Three Shared-Memory Multiprocessor Models are:

i. UMA (Uniform-Memory Access)

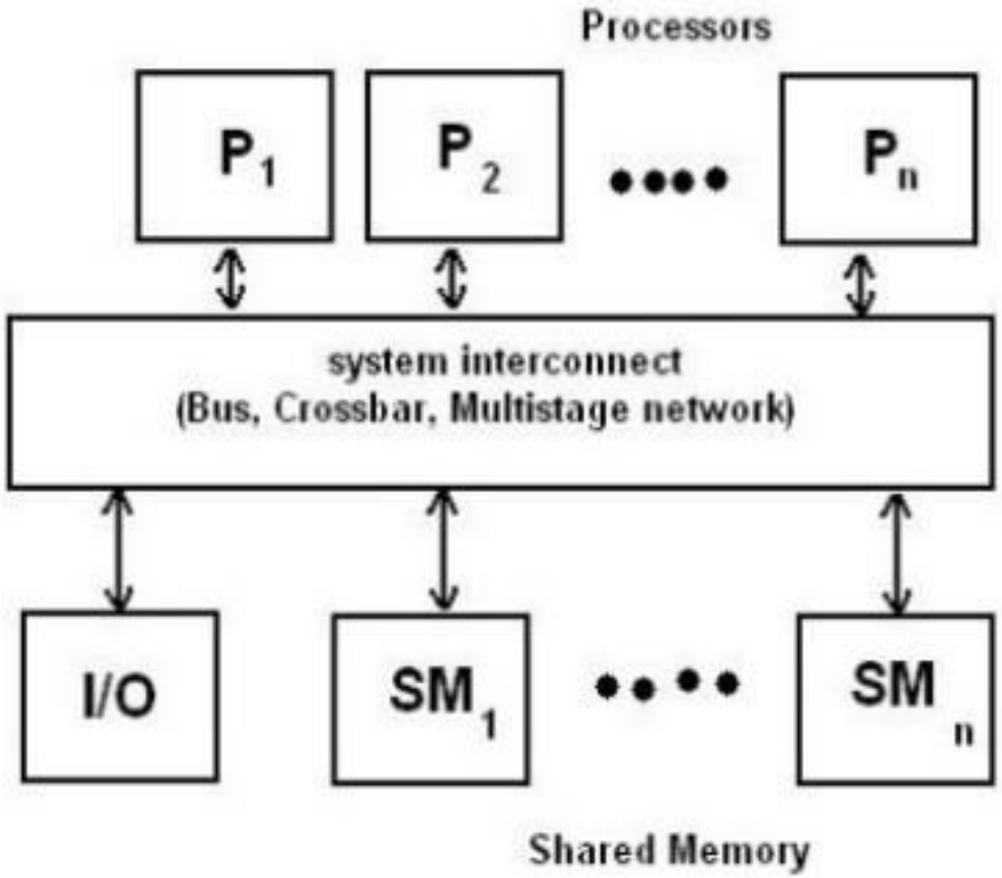
ii. NUMA (Non-Uniform-Memory Access)

iii. COMA (Cache-Only Memory Architecture)

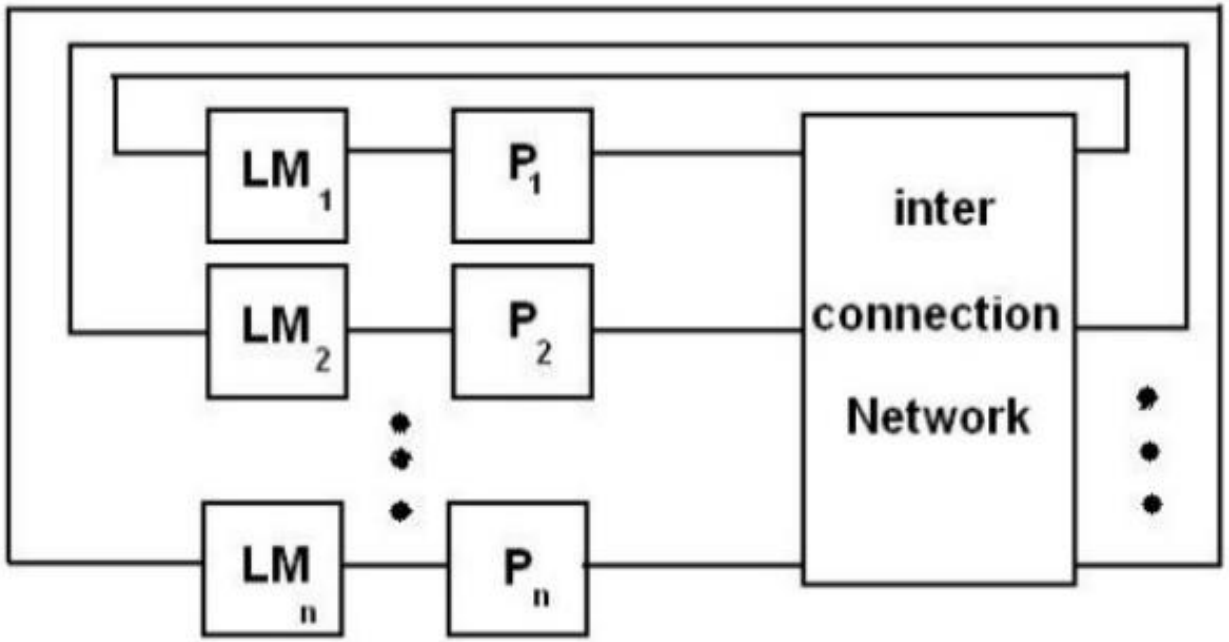
2. Unshared Distributed Memory

i. CC-NUMA (Cache-Coherent -NUMA)

UMA Multiprocessor Model



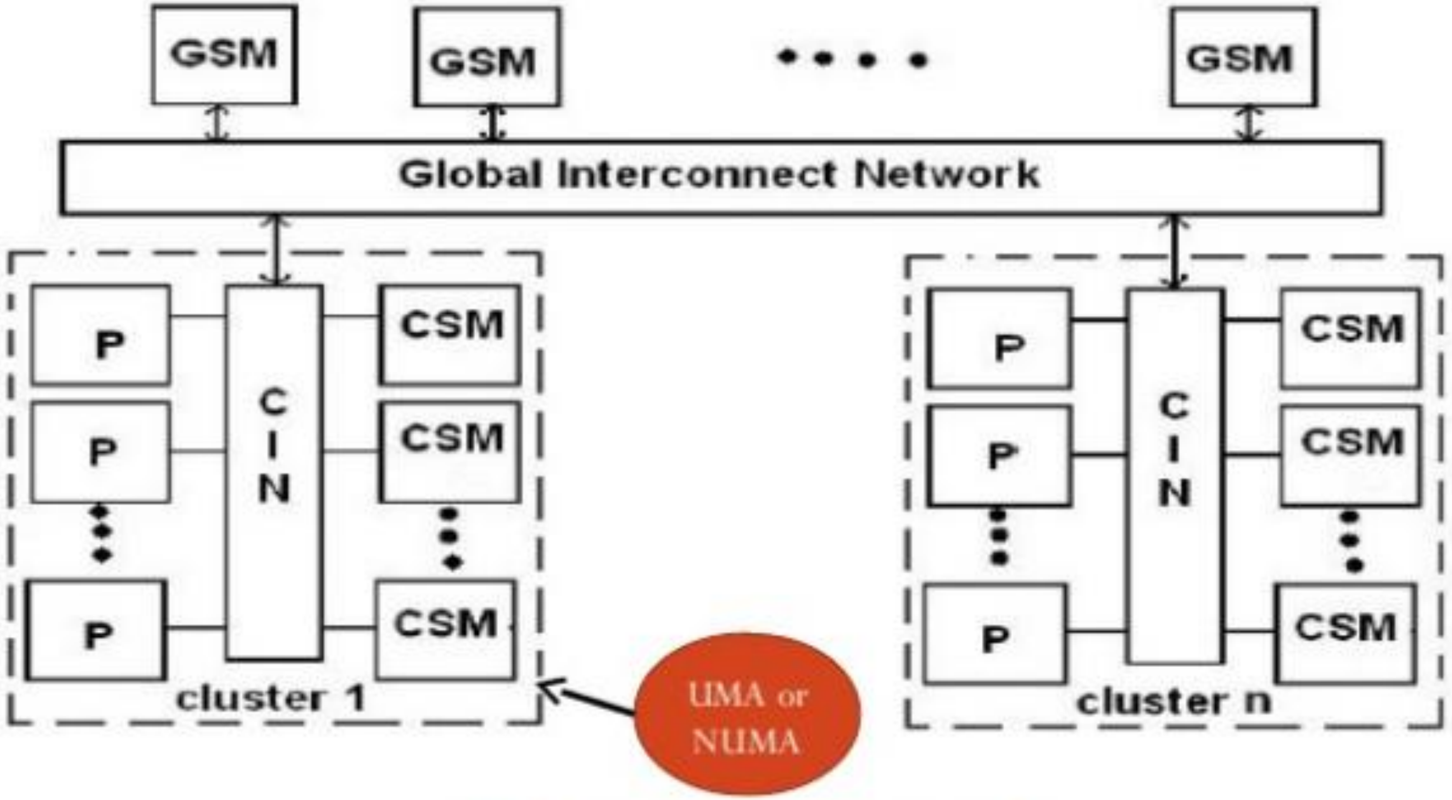
NUMA - Memory Models



(a) Shared local memories

LM – Local Memory
P - Local Processor

NUMA - Memory Models

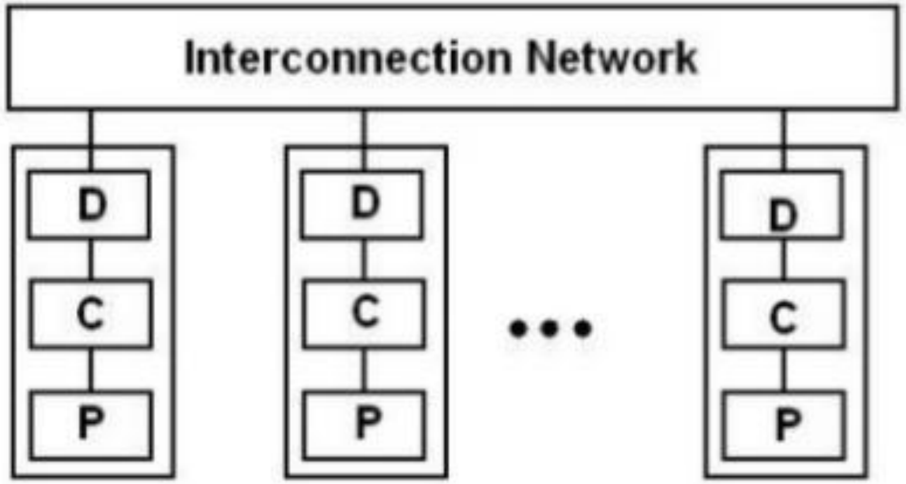


(b) A hierarchical cluster model

(Access of Remote Memory)

P – Processor
CSM – Cluster Shared Memory
CIN – Cluster Interconnection Network
GSM – Global Shared Memory

COMA Multiprocessor Model



C – Cache
P – Processor
D - Directories

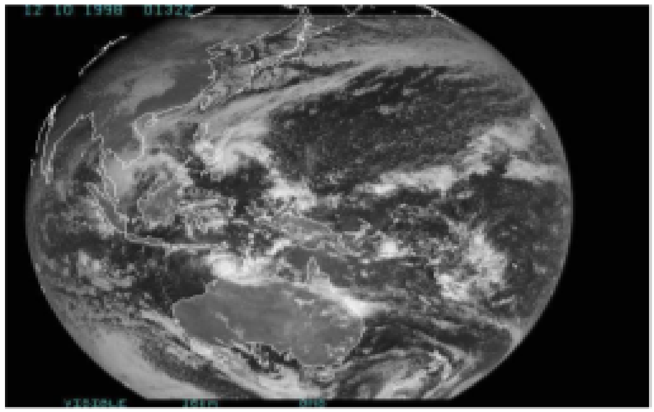
- Distributed Main Memory converted to Cache
- Cache form Global Address Space
- Remote Cache access by – Distributed cache Directories

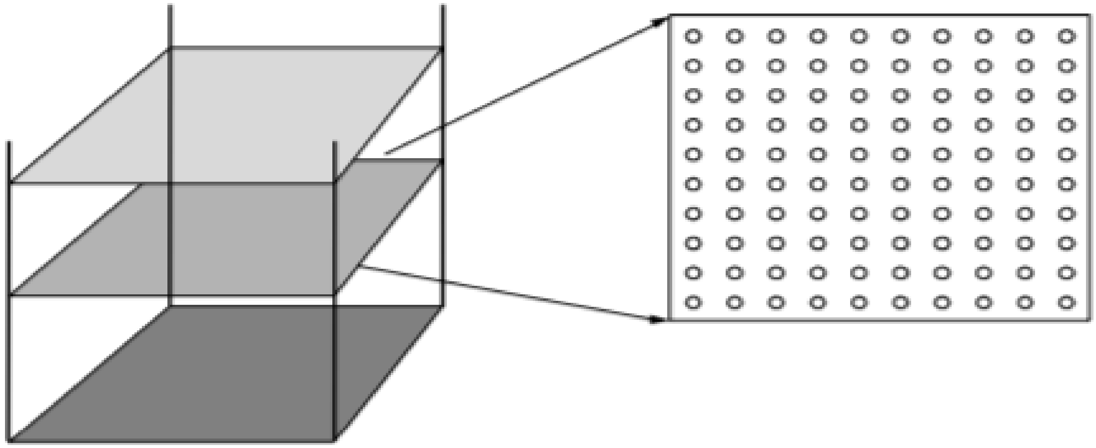
3. Perspective on parallel programming

- Motivating Problems
- Process of creating a parallel program

3. Perspective on parallel programming

- Motivating Problems : Simulating Ocean Currents
  - Model as two dimensional grid :
    - Discretize in space and time
    - Finer and temporal resolution => greater accuracy
  - Many different computations per time step





(a) Cross sections

(b) Spatial discretization of a cross section

3. Perspective on parallel programming

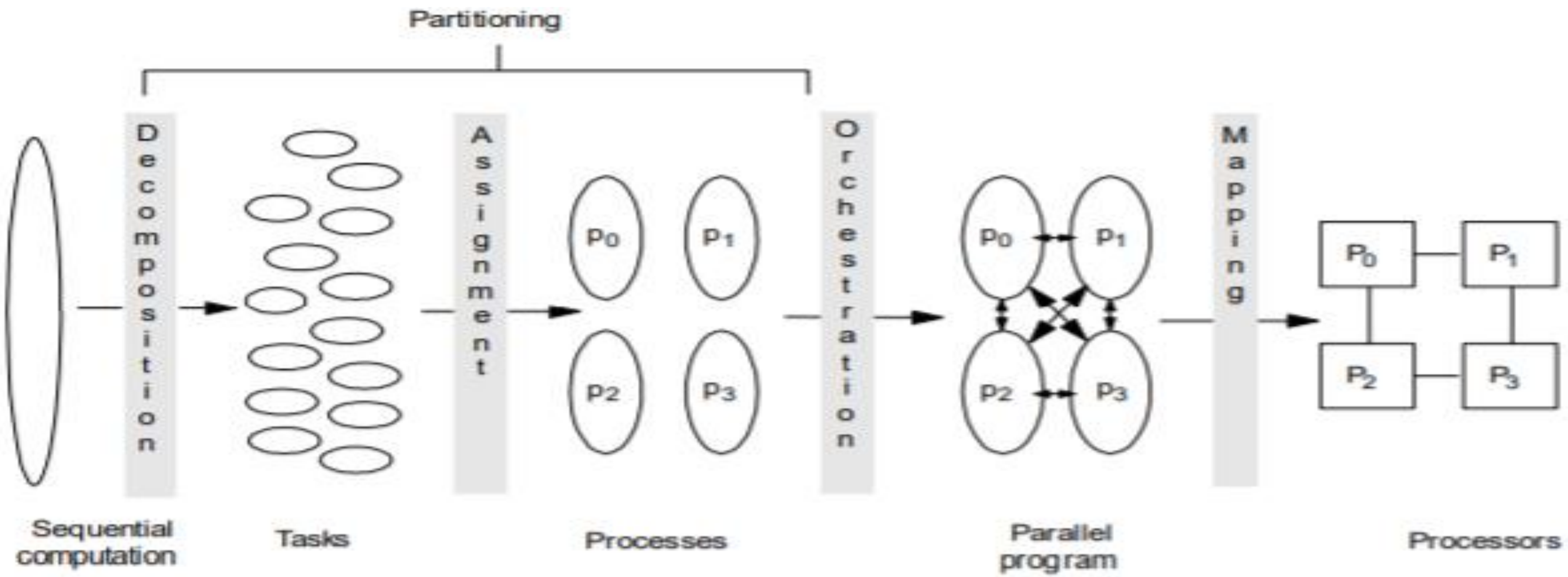
- Motivating Problems : Simulating interactions of many stars evolving over time.
    - Computing forces is expensive

    Eg. Stars on which forces of other elements need to be computed
    - Many time steps, plenty of concurrency across stars within each step.

3. Perspective on parallel programming

- Creating a parallel program
  - Identify the work that can be done in parallel : computation, data access and I/O
  - Partition of work and perhaps data among processes
  - Manage data access, communication and synchronization

3. Perspective on parallel programming



- **Decomposition** of computation in tasks
- **Assignment** of tasks to processes
- **Orchestration** of data access, comm, synch.
- **Mapping** processes to processors

Reference

Text Books and/or Reference Books:

1. Professional CUDA C Programming – John Cheng, Max Grossman, Ty McKercher, 2014
2. B.Wilkinson, M.Allen, ”Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers”, Pearson Education, 1999
3. I.Foster, ”Designing and building parallel programs”, 2003
4. Parallel Programming in C using OpenMP and MPI – Micheal J Quinn, 2004
5. Introduction to Parallel Programming – Peter S Pacheco, Morgan Kaufmann Publishers, 2011
6. Advanced Computer Architectures: A design approach, Dezso Sima, Terence Fountain, Peter Kacsuk, 2002
7. Parallel Computer Architecture : A hardware/Software Approach, David E Culler, Jaswinder Pal Singh Anoop Gupta, 2011 8. Introduction to Parallel Computing, Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Pearson, 2011

Thank You