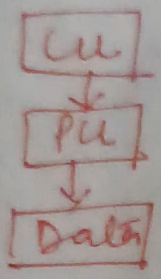
Flynn's Taxonomy for classification of computers

SISD: (Single instruction, Single Data)

> A traditional sequential computer architecture where a single processor executes a single instruction stream on a single data stream.

> Characteristics:

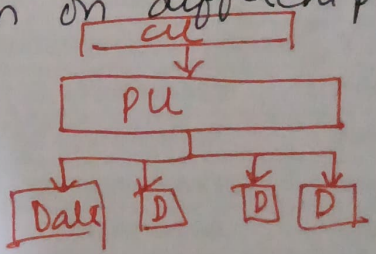


One CU
One Processing unit
Executes one instruction at a time
Process one data at a time

> eg:- classical uniprocessor systems
> use?? Embedded systems, simple tasks.

SIMD:

> A parallel architecture where multiple processing units execute same instruction on different pieces of data simultaneously.



> Characteristics:

One CU
Multiple PU
Execute one inst on many data pts.
Vector & matrix operations

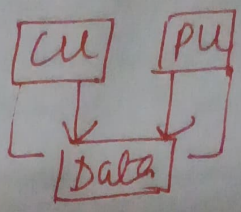
> eg: GPU
Intel AVX

> Image processing
ML Training

> A model considered ...

MISD:

> An uncommon architecture where multiple processors execute different instructions on same data stream.



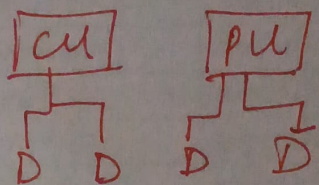
> Chr:
- Multiple control units
- Multiple PU
- Limited applicability

> eg: fault Tolerant systems.

> Real Time systems
Safety critical systems (Avionics)

MIMD:

> A highly flexible architecture where multiple processors execute different instructions on different data independently.



> Chr:
- X CU
- X PU
- asynchronous parallelism
- Shared/ Distributed memory based.

> eg: Multicore CPU.
cluster/ cloud.
Modern supercomputers.

> web servers
Databases.

PRAM (Parallel Random Access Machines)

> A model considered for most parallel algorithms
> Multiple processors are attached to

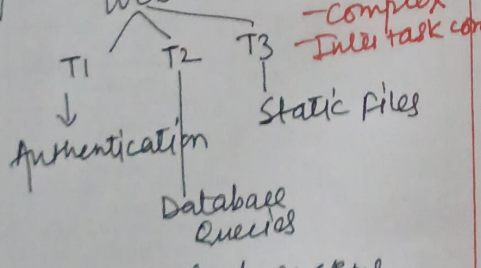
	SISD	SIMD	MISD	MIMD
Instn stream	Single	Single	Multiple	Multiple
Data stream	Single	M	S	M
Parallelism	None	Data P	Instn Redundancy	Task P.
Usage	Legacy sys Embedded sys.	-GPU -SC.C. -ML-Training	-fault Tolerant systems	-Multicore CPUs, -clusters

Task Parallelism

> Involves different tasks / fns / ~~tasks~~ threads across multiple processors.

> Features:
* Parallel execution of different tasks.
* Each task may be independent or may req. co-ord.
* Heterogeneous workloads.
  - Diverse workload
  - Eff ↑, Flexible
  - complex
  - Inter task comm.

> eg:



> Multithreaded systems
OS / CC / clusters
RTS

MCCPU's

Data Parallelism

> Involves distributing same task across multiple processors.
> Each processor works on different subset of data.

> Features:
* Parallel execution of same task.
* Efficient for large datasets.
* Homogeneous workloads.

> eg: Image Processing
Each processor applies same filter to diff. part of image.

> ML-Training
Graphics.

GPU's

- Scalable
- Performance ↑
- simple

→ comm. overhead
→ limited use.

PRAM (Parallel Random Access Machines)

> A model considered for most parallel algorithms
> Multiple processors are attached to single block of a memory through (MAU - Memory ...

Bit Level Parallelism

> This is about parallelism inside a processor
> word size of a computer ↑
> 8 → 16 → 32 → 64.
> Benefits:
  * Faster arithmatic operations
  * Reduces no. of instructions required.
  * common in modern processors.

Instruction Level Parallelism:

> Executing multiple instructions simultaneously within a processor.
> sequential → concurrent
  = parallelism.
> Benefits: * Better CPU utilization.
  * ↑ Performance
> Techniques:
  * Pipelining (Fetch → Decode → Execute)
    eg: Assembly line
  * Superscalar Execution
    Multiple Execution units
  * Out of order execution
    ✓ As resources are available
    X not strictly in programm order
  * Speculative execution & Branch Prediction
    ✓ execute instruction ahead of time

HPC systems aims to solve large, complex problems by using multiple computing resources in parallel. The key concepts used for this are:-

1). concurrency
2). Decomposition

Note C+D foundation of parallel pgm model

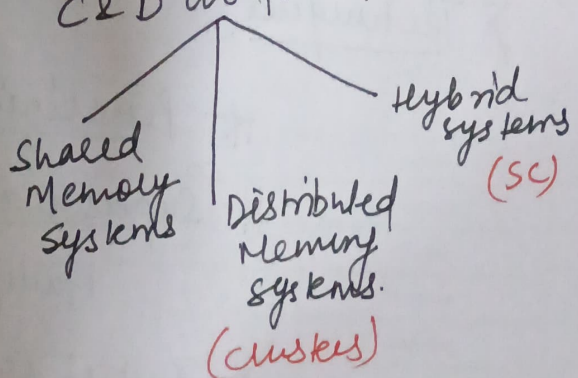
Concurrency

> Multiple tasks are in progress at the same time.
> concurrency → Parallelism
> Degree of conc ↑↑.
> Levels:
  * ILP (Instruction)
  * TLP (Thread)
  * PL Concurrency (Process)
  * DLP (Data)
  * Task level concurrency
> Conc. Computing
  ↓
  Threading
  +
  Asynchrony
  +
  Preemptive Multitasking

Decomposition

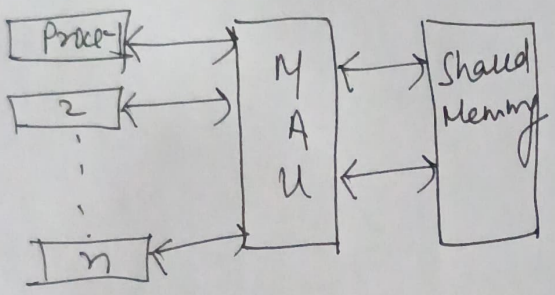
> Process of breaking a large problem into smaller subproblems.
  ↓
  concurrency.
> Types:
  * Domain D (Data)
  * Functional D (Task)
  * Hybrid D.

C & D in Architectures



PRAM (Parallel Random Access Machines)

- A model considered for most parallel algorithms
- Multiple processors are attached to single block of a memory through (MAU) - Memory access unit.
- A PRAM model contains
  - * A set of similar type of processors.
  - * common shared memory unit.
  - * MAU.



n number of processors can perform independent operations on n no. of data. which may result in simultaneous access of same memory location. To solve we have following constraints.

EREW
ERCW
CREW
CRCW

Methods to implement PRAM

1. Shared Memory model
2. Message passing model
3. Data parallel Model