

## Lecture 28: October 25, 2021

Computer Architecture and Organization-II

Biplab K Sikdar

### Parallel Computers -I

Today's processing need is the high speed computation.

There are applications where computer should process  $10^{16}$  operations/sec or more.

Example: Weather forecasting, modeling fusion reactors, bio-medical analysis, Speech recognition, visual image processing, VLSI circuit design and simulation....

Today's supercomputer speed is of the order of  $10^{16}$  operations/sec (exa FLOPs).

We need a system with multiple processing units - capable of parallel computation.

It is to satisfy the requirement of very high speed processing

- involving low cost computers.

Different multi-processor computer architectures are introduced in real design.

Classification of such computers can be done based on their structure or behavior.

Popular classification relies on number of instructions as well as operand sets that can be processed simultaneously by a computer.

Other alternatives - internal organization of processors, interprocessor connection structure, flow of instructions and data through the system, etc.

Flynn's Classification of parallel computers follow first parameter

- number of instructions/operand sets that can be processed simultaneously.

The basis of Flynn's classification is the number of instruction streams and data streams that can be processed simultaneously by a computer architecture.

# 0.1 Flynn’s Classification

Michael J Flynn, in 1966, roughly divided world of computers into four categories.

- 1. SISD (Single Instruction stream Single Data stream) machine,
- 2. SIMD (Single Instruction stream Multiple Data streams) machine,
- 3. MISD (Multiple Instruction streams Single Data stream) machine, and
- 4. MIMD (Multiple Instruction streams Multiple Data streams) machine.

# 0.2 SISD

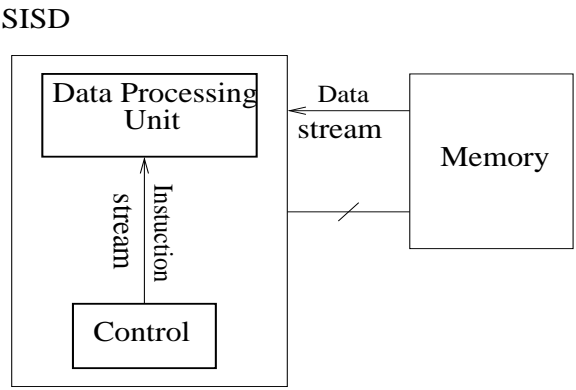


Figure 1: SISD machine

SISD computers process single instruction stream on single data stream (Figure 1).

Its structure follows von Neumann architecture.

It consists of a single processing unit that receives instruction and data streams.

At each step, control unit issues one instruction.

Algorithm runs on SISD computer is sequential. It cannot realize parallelism.

### 0.3 SIMD Computers

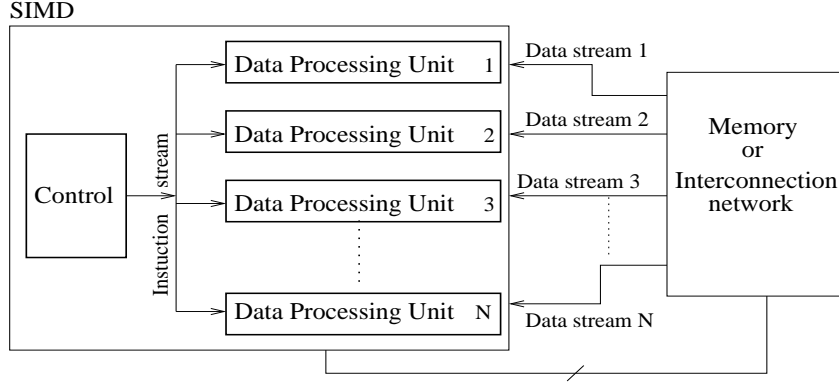


Figure 2: SIMD machine

An SIMD computer consists of  $N$  identical processors and these operate synchronously.

Processors execute single instruction stream issued by a control unit (CU).

However, an instruction in processors operates on different set of operands (Fig. 2).

Let consider computation of vector addition  $C = A + B$ , where

$A(a_0, a_1, a_2, \dots, a_i, \dots, a_{n-1})$ ,

$B(b_0, b_1, b_2, \dots, b_i, \dots, b_{n-1})$  and

$C(c_0, c_1, c_2, \dots, c_i, \dots, c_{n-1})$  are vectors.

$$c_0 = a_0 + b_0, c_1 = a_1 + b_1, \dots, c_i = a_i + b_i, \dots, c_{n-1} = a_{n-1} + b_{n-1}.$$

It requires  $n$  additions. An SISD computer requires  $n$ -time step to find  $C$ .

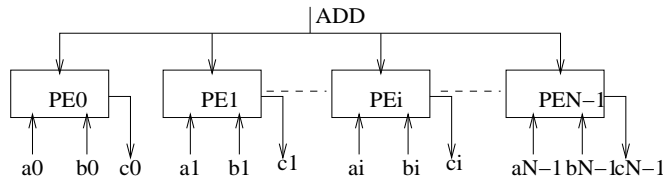


Figure 3: Vector addition in one unit of time

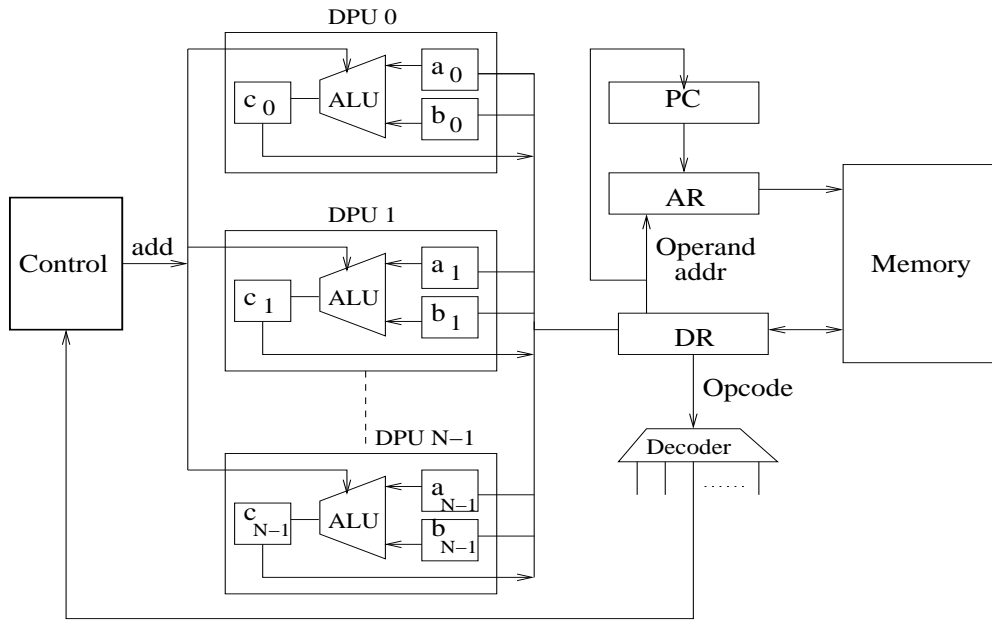


Figure 4: Vector addition in SIMD computer

Computation of  $C$ , shown in Figure 4, is in an SIMD computer.

Single *add* instruction is executed at all DPUs ( $\text{DPU}_i$  computes  $c_i = a_i + b_i, \forall_i$ ).

Number of processing units ( $N$ ).

Number of components ( $n = N$ ) of each vectors  $A$ ,  $B$ , and  $C$  (for this example).

Computation of  $C$  is in one unit of time once operands  $a_i$ 's/ $b_i$ 's are placed in DPUs.

Computing power of a multiprocessor system with  $N$  processors is, therefore - depends on efficiency of distribution of tasks of a job that can be run simultaneously.

It is desirable that processors should be able to communicate among themselves.

Conventionally, such communication is realized either by

- i) Communication through a shared memory, or
- ii) Communication through an intermediate network.

### In shared memory

For communication from processor  $P_i$  to processor  $P_j$

(say,  $P_j$  needs a data item  $x$  from  $P_i$ )

Following steps are to be executed (Figure 5(a)):

1. Processor  $P_i$  writes ( $x$ ) to memory at location L,
2. Processor  $P_j$  reads the item ( $x$ ) written by  $P_i$  from the memory location L.

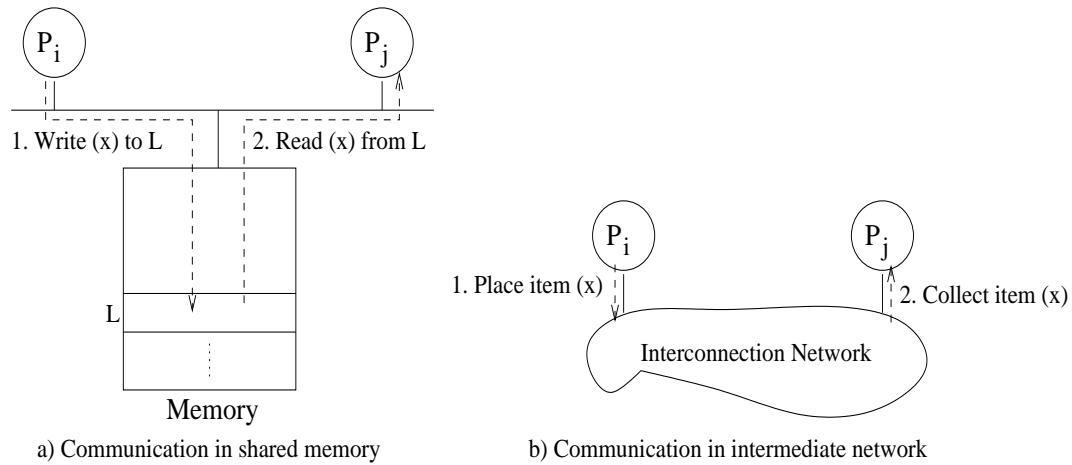


Figure 5: Communication between processors

### Other option

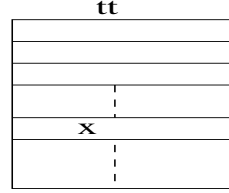
Figure 5(b): communication between  $P_i$  and  $P_j$  through interconnection network.

Based on mode of communication, SIMD computer can, therefore, be realized as

- i) Shared memory (SM) SIMD computers, or
- ii) Interconnection network SIMD computers.

### 0.3.1 Shared memory (SM) SIMD computers

Let consider searching of an element  $x$  in a table  $tt$  consisting of  $n$  elements.



$x$	

Figure 6: Table  $tt$

Searching time for  $x$  is  $n$  units if executed in an SISD computer.

In an SIMD computer, table search can be distributed among the  $N$  DPUs/PEs.

All DPUs simultaneously compute *compare*.  $DPU_i$  compares  $x$  and *element<sub>i</sub>* of  $tt$ .

Total time for completion of the job depends on following tasks

1. Place (broadcast)  $x$  in all DPUs,
2. Place *element<sub>i</sub>* in  $DPU_i$ ,  $\forall_i$ ,
3. Compute *compare* instruction at all the DPUs, and
4. Store outcome of search (say,  $DPU_i$  stores true in location  $l_i$  if *element<sub>i</sub>* =  $x$ ).

Time taken for task 3 -that is, for *compare*, is  $O(1)$  in an SIMD computer

(as all DPUs execute *compare* simultaneously).

Delays in task 1, 2 and 4 are dependent on shared memory organization.

For better efficiency, SM SIMD computer is designed with parallel RAM (PRAM).

Basic PRAM model allows DPUs/PEs to access shared memory simultaneously  
if memory locations accessed (read from or write into) are different.

Shared memory SIMD machines are classified into four groups

-EREW, ERCW, CREW, and CRCW.

It is based on the feature of simultaneous access (read-write) to the same memory location by more than one PE.