PARALLEL PROCESSING

Execution of *Concurrent Events* in the computing process to achieve faster *Computational Speed*

Levels of Parallel Processing

- Job or Program level

- Task or Procedure level

- Inter-Instruction level

- Intra-Instruction level

PARALLEL COMPUTERS

Architectural Classification

- **Flynn's classification**
  - **The taxonomy of computer systems proposed by M. J. Flynn in 1966 and is based on the notion of instruction and data streams that can be simultaneously manipulated by the machine. A stream is just a sequence of items (instruction or data).**
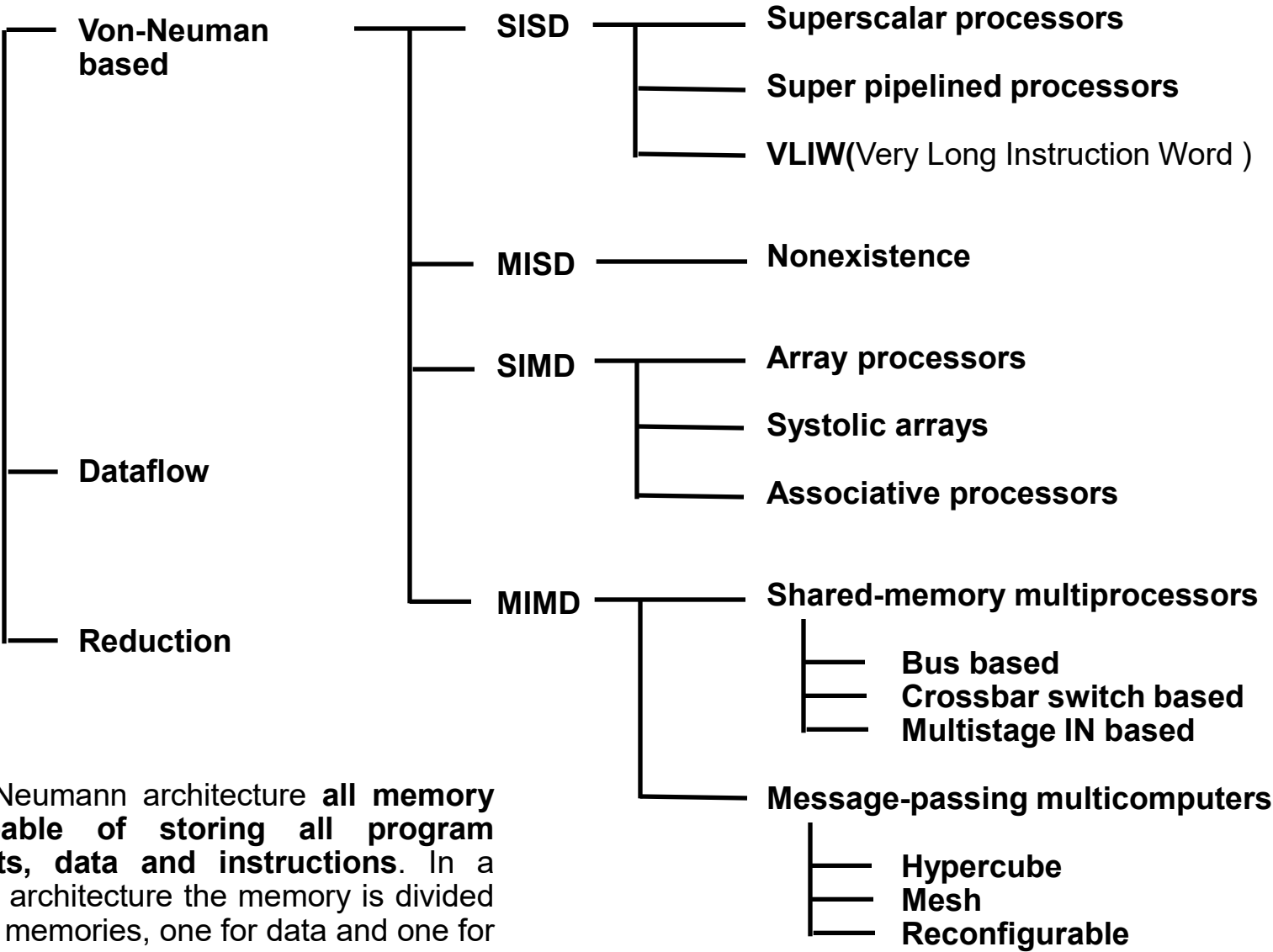 - **- Instruction Stream**
    - **Sequence of Instructions read from memory**
  - **- Data Stream**
    - **Operations performed on the data in the processor**
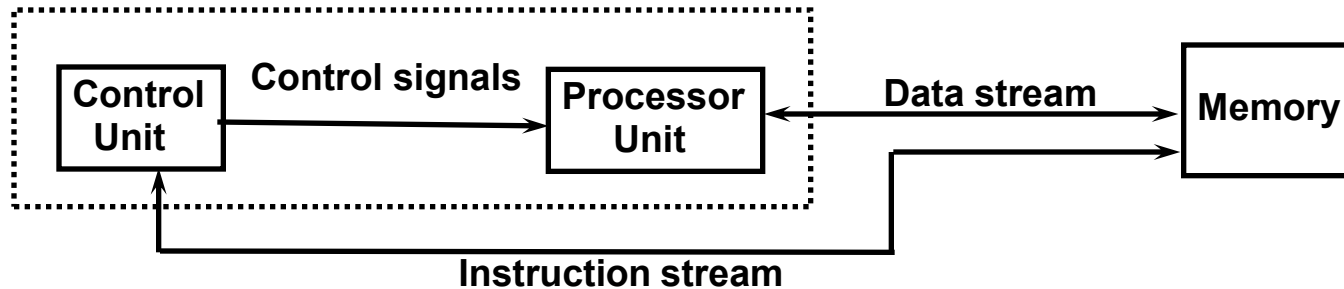
		Number of *Data Streams*	
		Single	**Multiple**
Number of *Instruction* *Streams*	**Single**	**SISD**	**SIMD**
	Multiple	**MISD**	**MIMD**

COMPUTER ARCHITECTURES FOR PARALLEL PROCESSING

- **Von-Neuman based**
  - **SISD**
    - **Superscalar processors**
    - **Super pipelined processors**
    - **VLIW(**Very Long Instruction Word)
  - **MISD** —— **Nonexistence**
  - **SIMD**
    - **Array processors**
    - **Systolic arrays**
    - **Associative processors**
  - **MIMD**
    - **Shared-memory multiprocessors**
      - **Bus based**
      - **Crossbar switch based**
      - **Multistage IN based**
    - **Message-passing multicomputers**
      - **Hypercube**
      - **Mesh**
      - **Reconfigurable**
- **Dataflow**
- **Reduction**

A Von Neumann architecture **all memory is capable of storing all program elements, data and instructions**. In a Harvard architecture the memory is divided into two memories, one for data and one for instructions

SISD COMPUTER SYSTEMS

```
┌·····································┐
: ┌──────────┐  Control signals  ┌──────────┐ :    Data stream    ┌──────────┐
: │ Control  │ ────────────────▶ │Processor │ :◀────────────────▶│  Memory  │
: │   Unit   │                   │   Unit   │ :                   │          │
: └──────────┘                   └──────────┘ : ─────────────────▶└──────────┘
└·····································┘
        ▲
        └──────────────────────────────────────┘
              Instruction stream
```

Characteristics

- **Standard von Neumann machine**
- **Instructions and data are stored in memory**
- **One operation at a time**

Limitations

> **Von Neumann bottleneck**

Maximum speed of the system is limited by the *Memory Bandwidth* (bits/sec or bytes/sec)

- **Limitation on *Memory Bandwidth***
- **Memory is shared by CPU and I/O**

PERFORMANCE IMPROVEMENTS

- **Multiprogramming**

- **Spooling**

- **Multifunction processor**

- **Pipelining**

- **Exploiting instruction-level parallelism**

    - **Superscalar**
    - **Superpipelining**
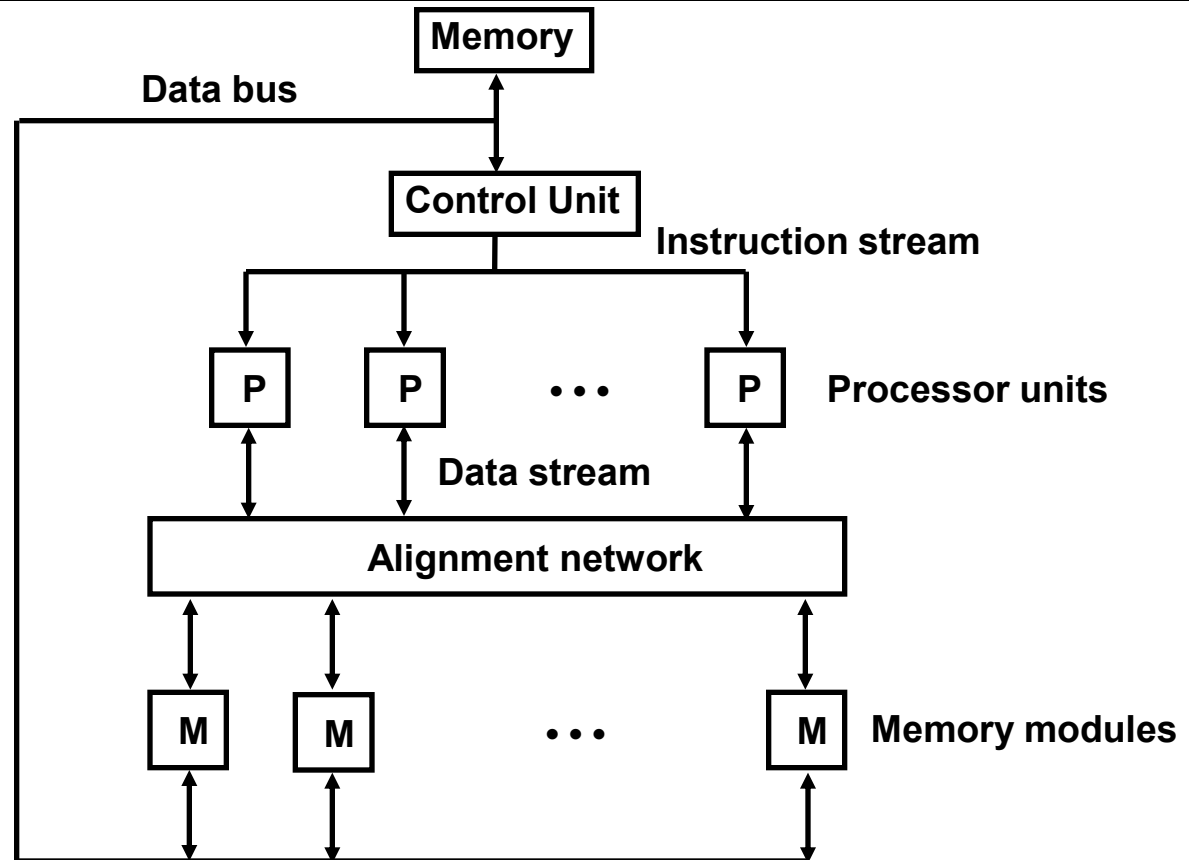   - **VLIW (Very Long Instruction Word)**

MISD COMPUTER SYSTEMS
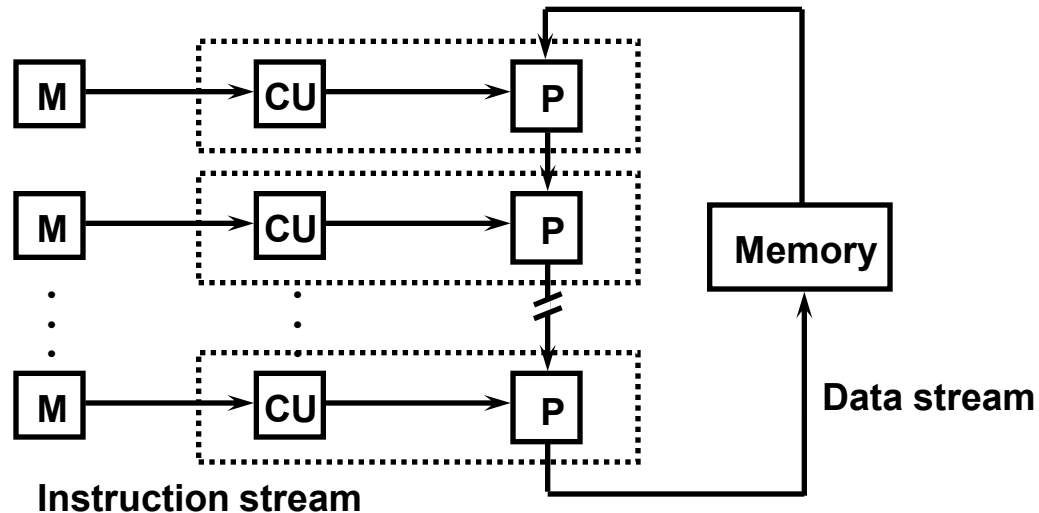
Characteristics

   **- There is no computer at present that can be
     classified as MISD**

SIMD COMPUTER SYSTEMS

Characteristics

- Only one copy of the program exists
- A single controller executes one instruction at a time

TYPES OF SIMD COMPUTERS

Array Processors

   **- The control unit broadcasts instructions to all PEs, and all active PEs
execute the same instructions**
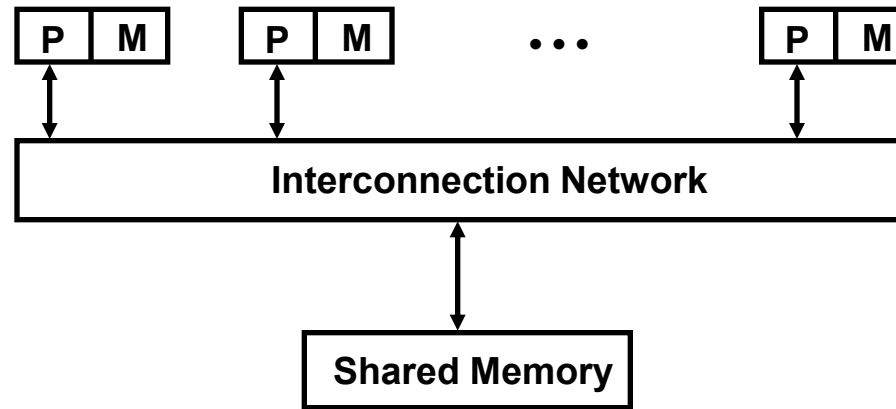   **- ILLIAC IV, GF-11, Connection Machine, DAP, MPP**

Systolic Arrays

   **- Regular arrangement of a large number of very simple processors
constructed on VLSI circuits**
   - CMU Warp, Purdue CHiP

Associative Processors

   - Content addressing
   **- Data transformation operations over many sets of arguments with a
single instruction**
   - STARAN, PEPE

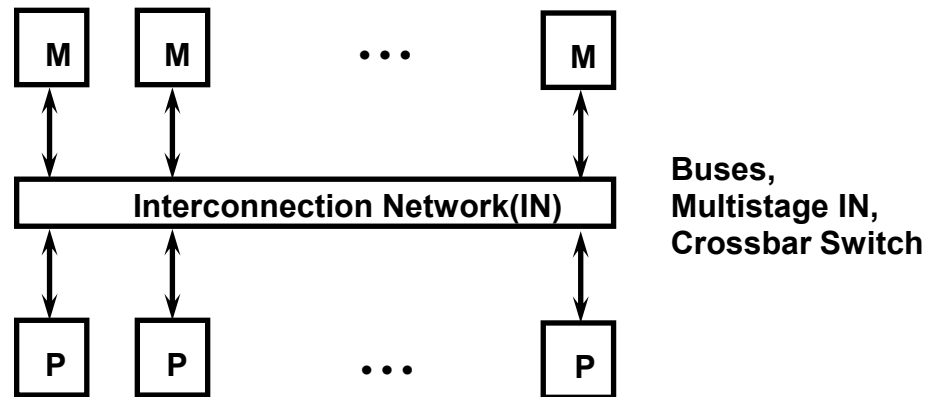# MIMD COMPUTER SYSTEMS

Characteristics

    - Multiple processing units

    - Execution of multiple instructions on multiple data

Types of MIMD computer systems

    - Shared memory multiprocessors

    - Message-passing multicomputers

SHARED MEMORY MULTIPROCESSORS

```
┌───┐ ┌───┐          ┌───┐
│ M │ │ M │   • • •  │ M │
└───┘ └───┘          └───┘
  ↕     ↕              ↕
┌─────────────────────────────┐      Buses,
│ Interconnection Network(IN)  │      Multistage IN,
└─────────────────────────────┘      Crossbar Switch
  ↕     ↕              ↕
┌───┐ ┌───┐          ┌───┐
│ P │ │ P │   • • •  │ P │
└───┘ └───┘          └───┘
```

Characteristics
> **All processors have equally direct access to**
>> **one large memory address space**

Example systems
> **Bus and cache-based systems**
>> **- Sequent Balance, Encore Multimax**
>
> **Multistage IN-based systems**
>> **- Ultracomputer, Butterfly, RP3, HEP**
>
> **Crossbar switch-based systems**
>> **- C.mmp, Alliant FX/8**

Limitations
> **Memory access latency**
> **Hot spot problem**

MESSAGE-PASSING MULTICOMPUTER

```
┌────────────────────────────────────┐
│        Message-Passing Network     │      Point-to-point connections
└────────────────────────────────────┘
   ↕        ↕                    ↕
 ┌───┐    ┌───┐               ┌───┐
 │ P │    │ P │    . . .      │ P │
 └───┘    └───┘               └───┘
   ↕        ↕                    ↕
 ┌───┐    ┌───┐               ┌───┐
 │ M │    │ M │    . . .      │ M │
 └───┘    └───┘               └───┘
```
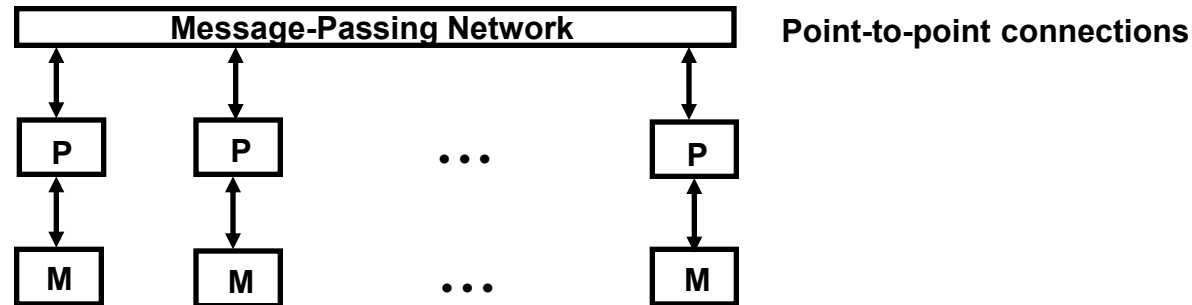
Characteristics

- **Interconnected computers**
- **Each processor has its own memory, and communicate via message-passing**

Example systems

- **Tree structure: Teradata, DADO**
- **Mesh-connected: Rediflow, Series 2010, J-Machine**
- **Hypercube: Cosmic Cube, iPSC, NCUBE, FPS T Series, Mark III**

Limitations

- **Communication overhead**
- **Hard to programming**