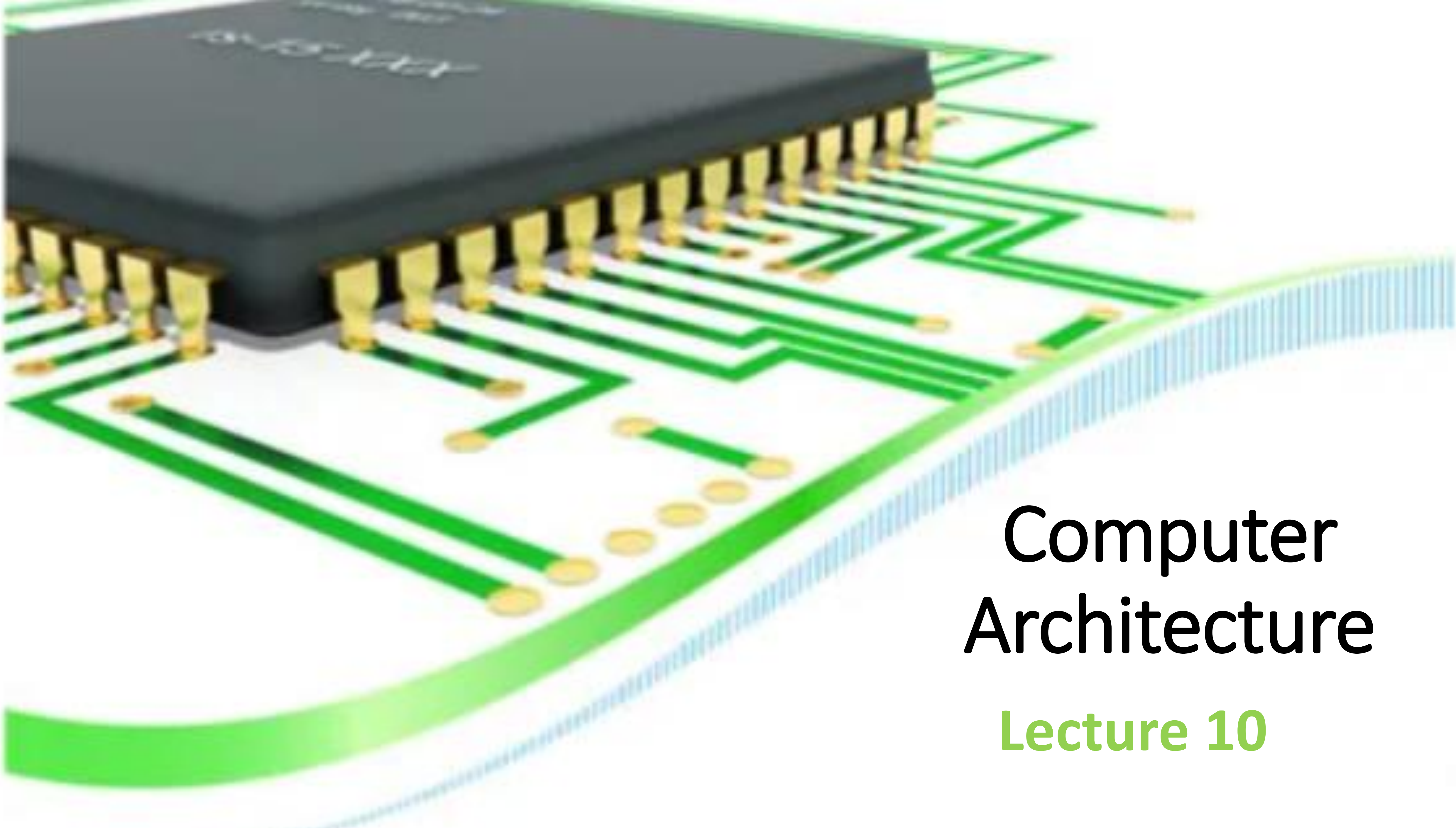


Computer Architecture

Lecture 10

Agenda

- RISC vs. CISC architectures
- Parallel processing
- Memory Organization

RISC Architecture

Reduced Instruction Set Computer (RISC) Architecture

RISC-based architectures depend on simplicity in hardware by employing an instruction set that consists of only a few basic steps. *Example: MIPS*

Characteristics of RISC:

1. It has simpler instructions and thus simple instruction decoding.
2. More general-purpose registers.
3. The instruction takes one clock cycle to get executed.
4. The instruction comes under the size of a single word.
5. Pipeline can be easily achieved.
6. Few data types.
7. Simpler addressing modes.

CISC Architecture

Complex Instruction Set Computer (CISC) Architecture

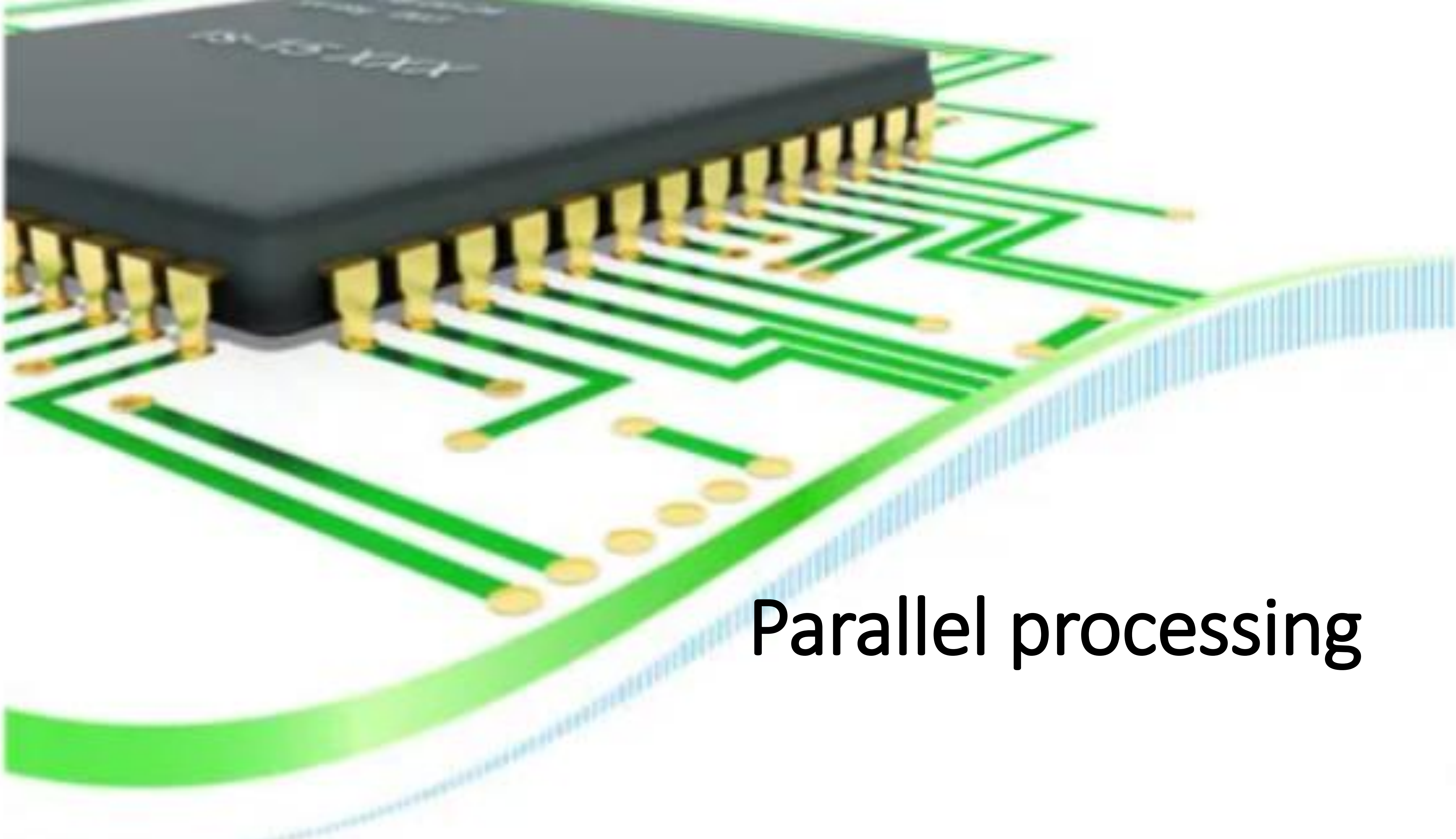
CISC-based architectures are based on having complex instructions where a single instruction requires several clock cycles for execution. *Example:* Intel x86

Characteristics of CISC:

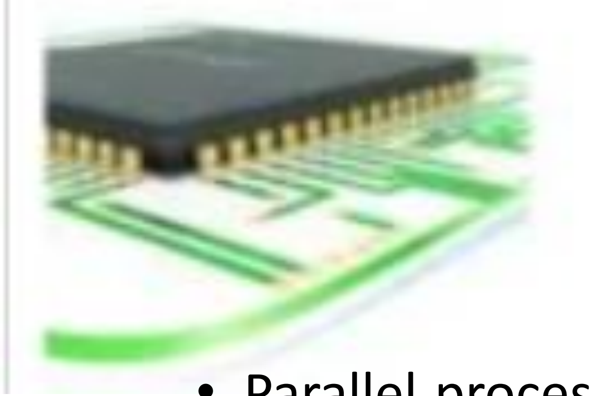
1. Instructions are complex, and thus it has complex instruction decoding.
2. The instructions may take more than one clock cycle to get executed.
3. The instruction is larger than one-word size.
4. Lesser general-purpose registers since the operations get performed only in the memory.
5. More data types.
6. Complex addressing modes.

RISC vs CISC Architecture

RISC	CISC
It is a Reduced Instruction Set Computer.	It is a Complex Instruction Set Computer.
It emphasizes on software to optimize the instruction set.	It emphasizes on hardware to optimize the instruction set.
RISC has simple decoding of instruction.	CISC has complex decoding of instruction.
Uses of the pipeline are simple in RISC.	Uses of the pipeline are difficult in CISC.
It uses a limited number of instruction that requires less time to execute the instructions.	It uses a large number of instruction that requires more time to execute the instructions.
The execution time of RISC is very short.	The execution time of CISC is longer.
It has fixed format instruction.	It has variable format instruction.
The program written for RISC architecture needs to take more space in memory.	Program written for CISC architecture tends to take less space in memory.



Parallel processing



Parallel Processing

- Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data-processing tasks to increase the computational speed of a computer system.

- Instead of processing each instruction sequentially as in a conventional computer, a parallel processing system is able to perform concurrent data processing to achieve faster execution time.

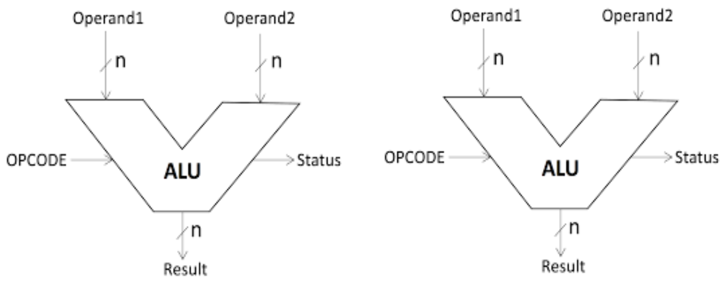
Parallel Processing

For example:

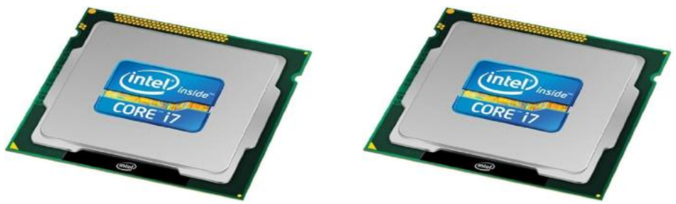
- While an instruction is being executed in the ALU, the next instruction can be read from memory.

EX1	WB1
ID2	EX2

- The system may have two or more ALUs and be able to execute two or more instructions at the same time.



- Furthermore, the system may have two or more processors operating concurrently.



Parallel Processing

- The purpose of parallel processing is to speed up the computer processing capability and increase its *throughput*.

- **Throughput** is, the amount of processing that can be accomplished during a given interval of time.

Classification of Parallel Processing

- One classification introduced by M. J. Flynn considers the organization of a computer system by the number of instructions and data items that are manipulated simultaneously.
- The sequence of instructions read from memory constitutes an instruction stream.
- The operations performed on the data in the processor constitutes a data stream.
- Parallel processing may occur in the instruction stream, in the data stream, or in both.

Classification of Parallel Processing

- Flynn's classification divides computers into four major groups as follows:

Flynn's Classification

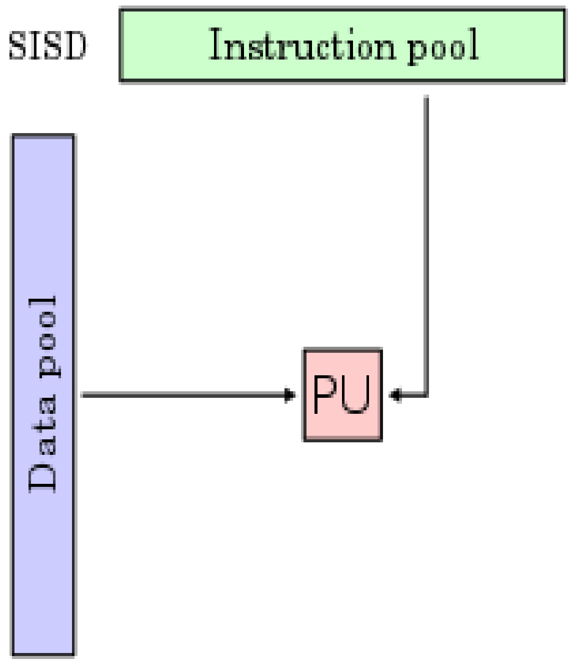
- SISD
  Single instruction stream, single data stream
- SIMD
  Single instruction stream, multiple data stream
- MISD
  multiple instruction stream, single data stream
- MIMD
  multiple instruction stream, multiple data stream

SISD

- SISD represents the organization of a single computer containing a control unit, a processor unit, and a memory unit.

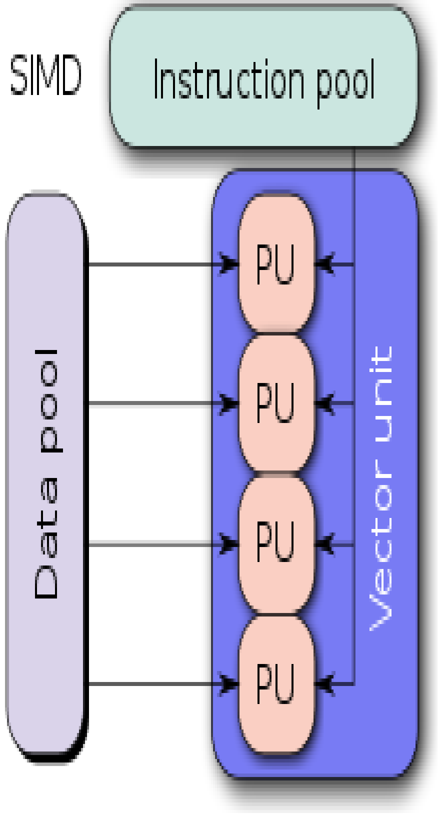
- Instructions are executed sequentially, and the system may or may not have internal parallel processing capabilities.

- Parallel processing in this case may be achieved using multiple functional units or by pipeline processing



SIMD

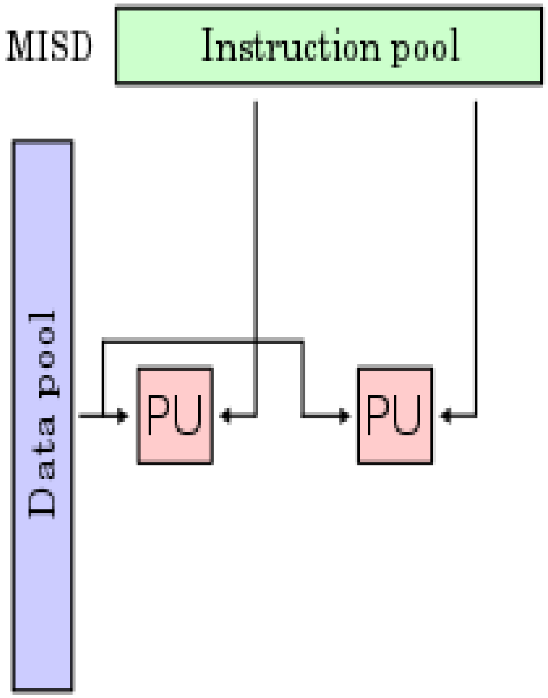
- SIMD represents an organization that includes many processing units under the supervision of a common control unit.
- All processors receive the same instruction from the control unit but operate on different items of data.
- The shared memory unit must contain multiple modules so that it can communicate with all the processors simultaneously.



MISD

- MISD is a type of parallel computing architecture where many functional units perform different operations on the same data.

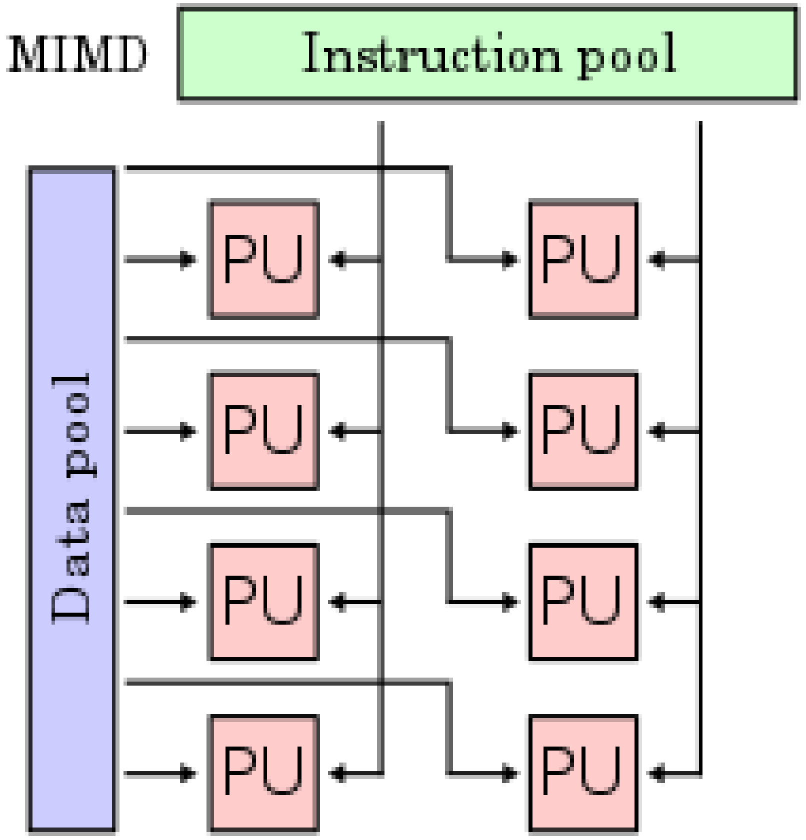
- Usually used in systems for Fault tolerance and task replication



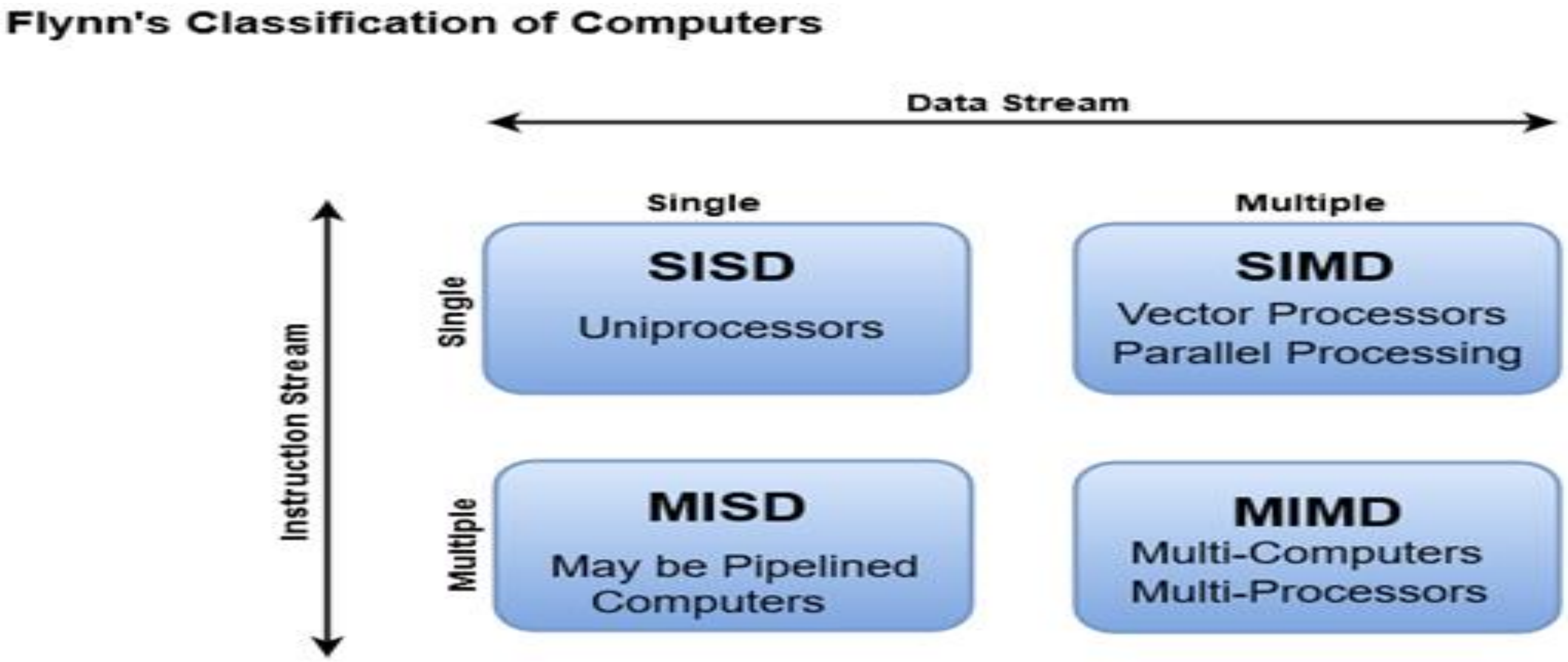
MIMD

- MIMD organization refers to a computer system capable of processing several programs at the same time.

- Most multiprocessor and multicomputer systems can be classified in this category.



Flynn’s Classification



Parallel processing

Parallel processing could be achieved by either:

1. Pipeline processing
2. Vector processing
3. Array processors

- Pipeline processing is an implementation technique where arithmetic suboperations or the phases of a computer instruction cycle overlap in execution.
- Vector processing deals with computations involving large vectors and matrices.
- Array processors perform computations on large arrays of data.

Pipeline Processing

- There are two areas of computer design where the pipeline organization is applicable.

- An arithmetic pipeline divides an arithmetic operation into suboperations for execution in the pipeline segments.

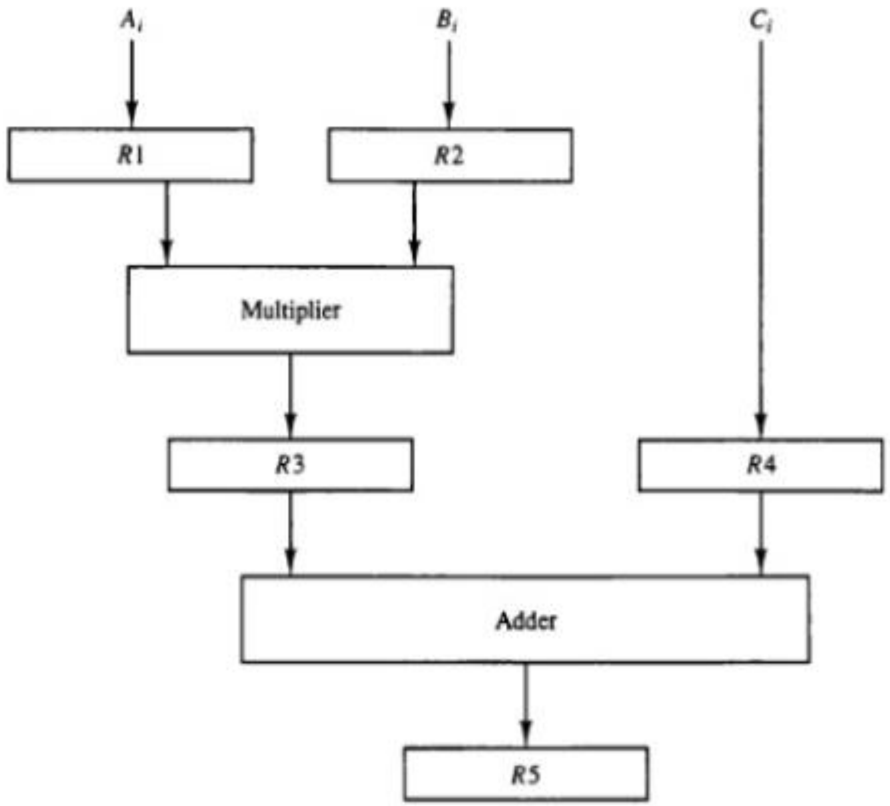
- An instruction pipeline operates on a stream of instructions by overlapping the fetch, decode, and execute phases of the instruction cycle.

Pipeline Processing

Instruction pipeline

	1	2	3	4	5	6	7
Instruction1	IF1	ID1	MEM1	EX1	WB1		
Instruction2		IF2	ID2	MEM2	EX2	WB2	
Instruction3			IF3	ID3	MEM3	EX3	WB3

Arithmetic pipeline



Vector processing

- There is a class of computational problems that are beyond the capabilities of a conventional computer.

- These problems are characterized by the fact that they require a vast number of computations that will take a conventional computer days or even weeks to complete.

- In many science and engineering applications, the problems can be formulated in terms of vectors and matrices that lend themselves to vector processing

Vector processing

- Application areas where vector processing is of the utmost importance:
  - Long-range weather forecasting
  - Petroleum explorations
  - Medical diagnosis
  - Aerodynamics and space flight simulations
  - Artificial intelligence and expert systems
  - Mapping the human genome
  - Image processing

Advantages of Vector Processors

- **High Performance:** Vector processors can process multiple operations simultaneously, increasing the speed of calculations.
- **Highly Parallel:** Vector processors are able to handle multiple operations in parallel, allowing for faster computations.
- **High Memory Bandwidth:** Vector processors are able to access large amounts of data at once, increasing the speed of computations.
- **Low Power Consumption:** Vector processors are much more efficient than traditional processors, reducing the amount of power needed to operate them.
- **Reduced Software Overhead:** Vector processors can reduce the amount of software code needed to complete tasks, saving time and resources.
- **Improved Accuracy:** Vector processors are more accurate than scalar processors, making them ideal for applications that require precision.

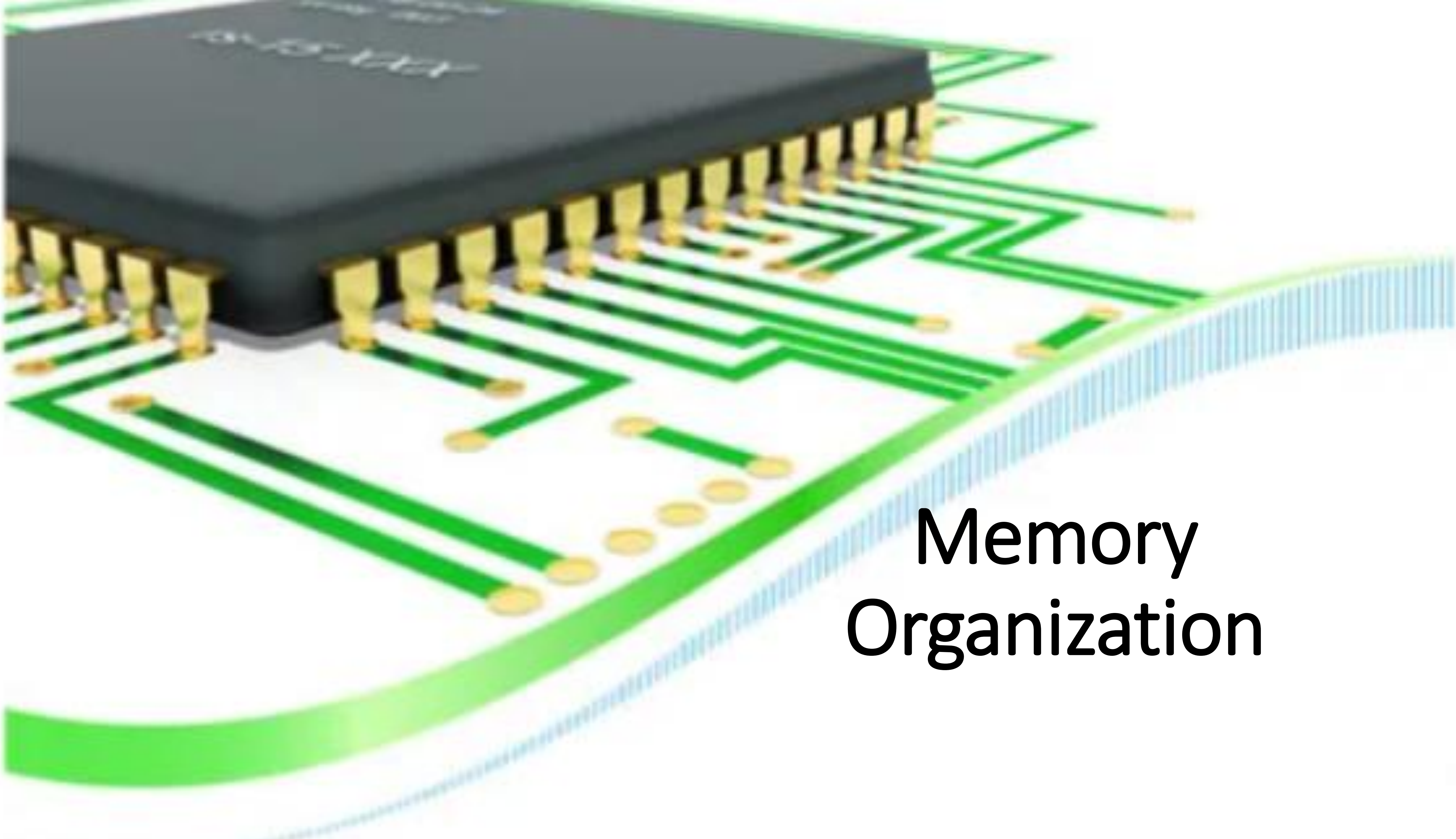
Limitations of Vector Processor

- **Speed Limitation:** Vector processors are limited by the speed at which they can execute instructions.
- **Memory Limitation:** Vector processors are limited by the amount of memory available for storing data.
- **Instruction Limitation:** Vector processors can only execute certain instructions and often require instruction modifications for more complex tasks.
- **Cost Limitation:** Vector processors are often more expensive than scalar processors due to their advanced technology.

Array processing vs. Vector processing

- The distinction between array processing and vector processing is that while vector processing uses a single processor to execute the same operation on numerous data items concurrently.

- Array processing uses several processors to work on individual array elements.



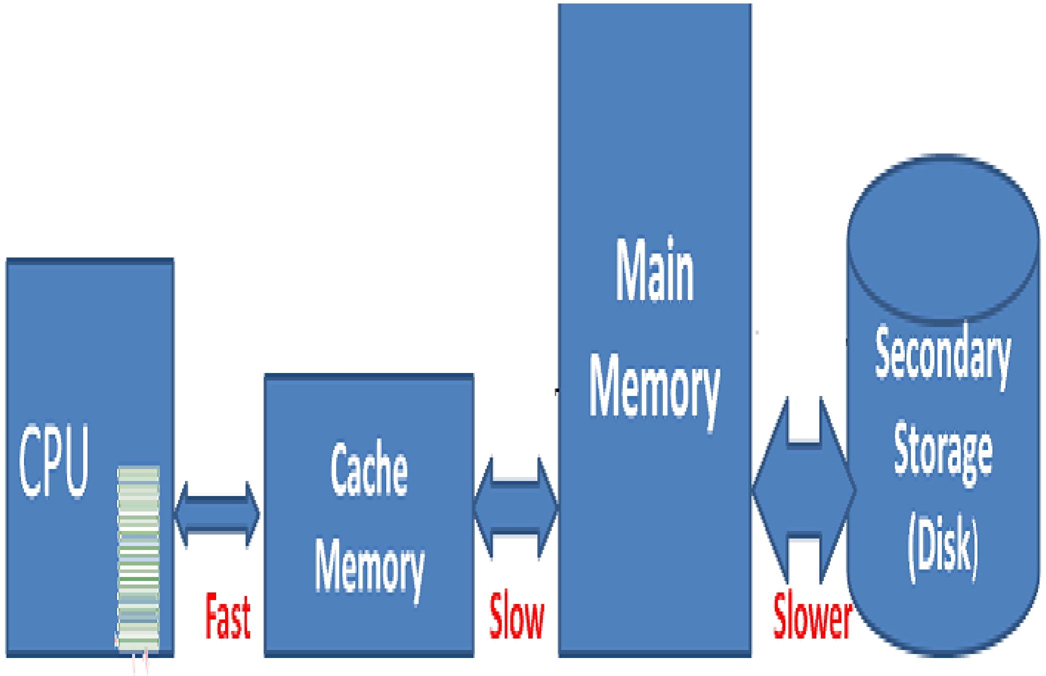
Memory Organization

Cache Memory

- Cache Memory is a special very high-speed memory. The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations.

- There are various different independent caches in a CPU, which store instructions and data.

- The most important use of cache memory is that it is used to reduce the average time to access data from the main memory.



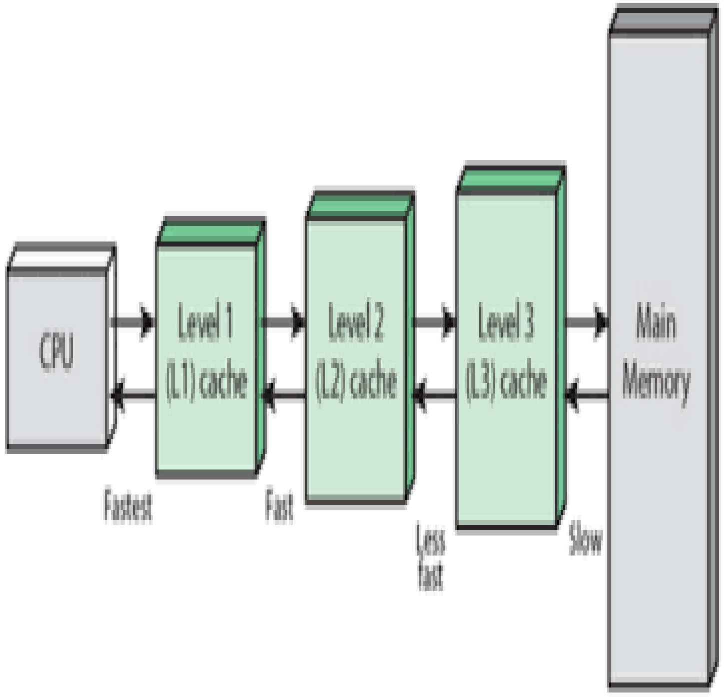
Cache Memory

Characteristics of Cache Memory

- Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU.
- Cache Memory holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
- Cache memory is more expensive than main memory or disk memory but more economical than CPU registers.
- Cache Memory is used to speed up and synchronize with a high-speed CPU.

Levels of Memory

- **Level 1 or Register:** It is a type of memory in which data is stored and accepted that are immediately stored in the CPU. The most commonly used register is Accumulator, Program counter, Address Register, etc.
- **Level 2 or Cache memory:** It is the fastest memory that has faster access time where data is temporarily stored for faster access.
- **Level 3 or Main Memory:** It is the memory on which the computer works currently. It is small in size and once power is off data no longer stays in this memory.
- **Level 4 or Secondary Memory:** It is external memory that is not as fast as the main memory but data stays permanently in this memory.



Cache Performance

- When the processor needs to read or write a location in the main memory, it first checks for a corresponding entry in the cache.
- If the processor finds that the memory location is in the cache, a **cache hit** has occurred and data is read from the cache.
- If the processor does not find the memory location in the cache, a **cache miss** has occurred. For a cache miss, the cache allocates a new entry and copies in data from the main memory, then the request is fulfilled from the contents of the cache.
- The performance of cache memory is frequently measured in terms of a quantity called **Hit ratio**.

Cache Performance

```
Hit Ratio(H) = hit / (hit + miss) = no. of hits/total accesses

Miss Ratio = miss / (hit + miss)
           = no. of miss/total accesses = 1 - hit ratio(H)
```

Thank You