­

Micro-Processor and Assembly Language

FA20-BCS-025

Idrees Ahmed Ghazi

Assignment # 1

**PREPARED FOR:** Sir Taimoor Shehzad

**PREPARED BY:**

# **Question#01**

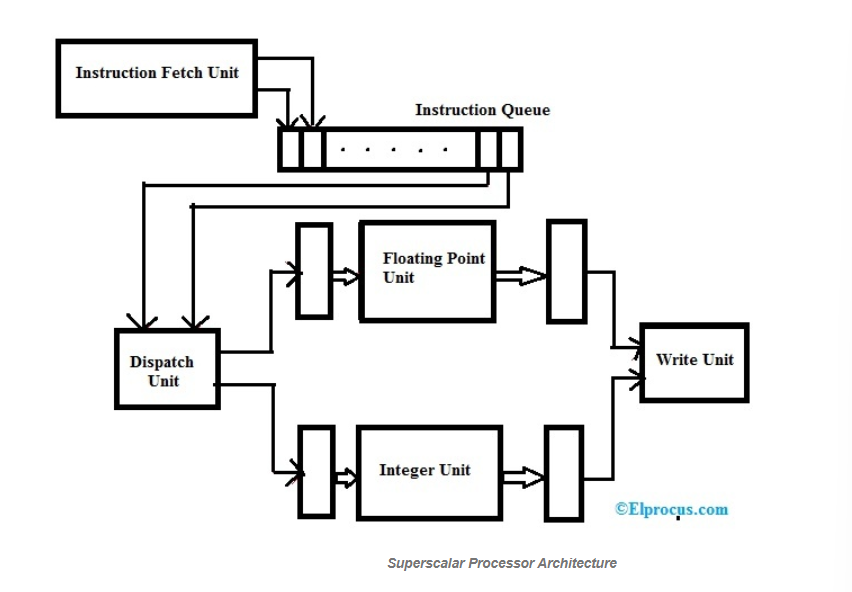
## Superscalar Architecture:

The Superscalar architecture is a type of architecture implemented in the CPU which involves executing instructions in a parallel way. Superscalar architecture follows instruction-level parallelism in a single processor to execute more than one instruction in a Clock cycle. A scalar processor runs only one instruction at a time while superscalar architecture can execute multiple instructions at a time.

### Features:

The following are the features of superscalar architecture:

* In this architecture, the CPU manages several instruction pipelines to perform multiple instructions simultaneously during a single Clock cycle.
* Superscalar Architecture includes all the pipelining features although multiple types of instructions are running simultaneously within the same pipeline.
* The architecture usually has parallel register renaming and parallel instruction decoding, speculative execution and out-of-order execution. So, these methods are normally used with complementing design methods like caching, pipelining, branch prediction and multi-core in microprocessor designs.



## Superscalar Architecture can be added to up to current Processor Markets:

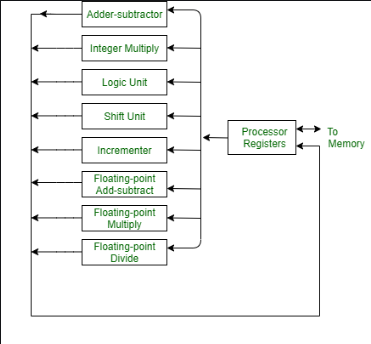
Superscalar architecture can be added to the markets to be sold because they offer efficient and potentially increased performance without the need of redesigning the processor’s architecture. By adding more execution units and improving the execution of instructions, superscalar architecture can increase the number of instructions in a single clock cycle. This can lead to significant performance improvement in applications, such as those which are highly parallelizable or the ones which require more computation.

# **Question#02**

## Parallel Processing:

Parallel Processing is the execution of multiple tasks or instructions simultaneously. This requires the use of multiple processors or execution units. Parallel processing increases the performance of the system and execution time is faster because the workload is distributed across multiple processing units.

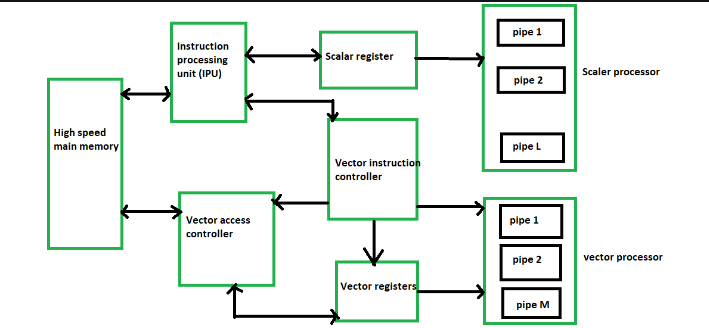
* Instructions are executed in the ALU, and the instructions are read from memory.
* The system can have multiple ALUs and be able to execute multiple executions at a time.
* A system can have multiple processors operating concurrently.



Parallel processing is derived from multiple levels of complexity. Shift registers work on one bit at a time while parallel register work simultaneously with all bits which increases the performance of the processor. By distributing data to multiple functional units, parallel processing is achieved.

## Vector Processing:

Vector processor are a type of parallel processing architecture which is specifically designed to handle mathematical and scientific computations which require a large amount of data. The used technique is called vectorization which involves the processing of multiple types of data elements at the same time. This results in faster execution time.



## Types of Parallel Processing:

There are many types of parallel processing.

### SIMD:

Single instruction and multiple data is used for the execution same instruction on multiple pieces of data simultaneously. SIMD is used in vector processors and graphical processing units (GPU).

### MIMD:

Multiple instructions and multiple data are used for the execution of multiple instructions on multiple pieces of data simultaneously. SIMD is used in multi-core processors and distributed systems.

### SPMD:

Single program multiple data is used for running a single program with multiple types of data. It is commonly used for high-performance in computing and supercomputing applications.

### Data Parallelism:

This involves splitting a large data set into smaller chunks and processing them in parallel architecture. It is commonly used in scientific and engineering applications that involve large amount of data sets.

### Task Parallelism:

Task Parallelism involves dividing a larger task into smaller sub-tasks that can be executed in parallel. This is commonly used in distributed systems and cloud computing.

Overall, parallel processing allows for faster and more efficient execution of complex tasks, by distributing the workload across multiple processors or execution units.

# **Question#03**

The organization of memory in a computer system can be approached in different ways, two of which are the segmented memory model and the flat memory model.

## Segmented Memory Model:

The segmented memory model was popularly used in older computer systems such as the Intel 8086 processor used in the early IBM PCs. This processor had a segmented memory model that consisted of four different segments: code, data, stack, and extra segments. This approach allowed for efficient use of memory as each segment could be optimized for a specific type of data or operation.

However, programming with a segmented memory model was more complicated and difficult as it required explicit management of memory segments and pointers. It also made data sharing between different parts of a program challenging since each segment had its address space and access permissions.

The segmented memory model divides memory into multiple segments or sections, each with its unique starting address and length. Each segment can have different access permissions and can store various types of data. These segments are usually defined by the operating system or the application and can be swapped in and out of memory as required.

## Flat Memory Model:

In the flat memory model, the entire memory space is considered as one contiguous block of memory. As a result, any part of memory can be accessed directly with a single memory address. This approach is commonly used in modern operating systems and applications due to its simplicity and ease of use.

The flat memory model enables the entire memory space to be viewed as a single linear, sequential and contiguous block, making it easy to access any part of the memory with a single memory address. Although the flat memory model may require memory management and address translation to enable certain operating system functions, protect resources, enable multitasking or expand memory capacity beyond the processor's physical address space, its primary advantage is its simplicity and ease of use. By treating memory as a single block, the flat memory model allows for efficient memory access and management, making it a common choice for modern operating systems and applications.

Currently, the flat memory model is preferred in modern computer systems due to its ease of use. However, specialized applications such as real-time operating systems or embedded systems may still use a segmented memory model for specific performance or security reasons.