

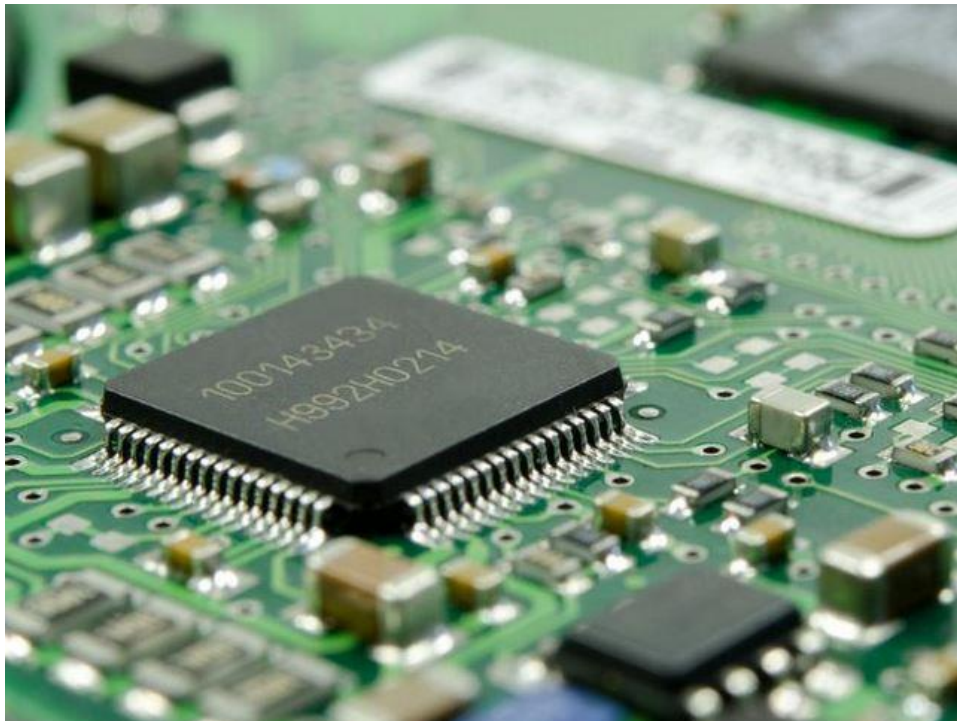


جامعة القادسية
كلية التربية



Lecture 25

Computer Architecture



Prepared by:

Firas Abdulrahman Yosif

Pipelining

Pipelining organizes the execution of the multiple instructions simultaneously. Pipelining improves the throughput of the system. In pipelining the instruction is divided into the subtasks. Each subtask performs the dedicated task.

An instruction in a process is divided into 5 subtasks likely,

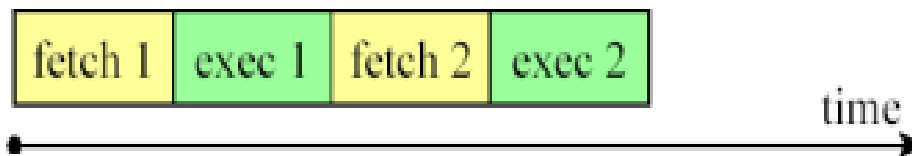
| | | | | |
|----------------------|-----------------------|------------------|------------------------|------------------|
| Instruction Fetch | Instruction Decode | Operand Fetch | Instruction Execute | Operand Store |
|----------------------|-----------------------|------------------|------------------------|------------------|

The instruction steps in process consist of:

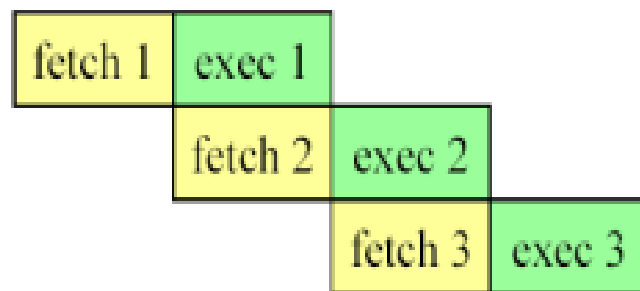
1. In the first subtask, the instruction is fetched.
2. The fetched instruction is decoded in the second stage.
3. In the third stage, the operands of the instruction are fetched.
4. In the fourth, arithmetic and logical operation are performed on the operands to execute the instruction.
5. In the fifth stage, the result is stored in memory.

Pipelining is a process that allows the CPU (Microprocessor) to fetch and execute instructions at the same time. Intel Co. implemented the concept of pipelining by splitting the internal architecture of the 8088/8086 μ P into two units that work simultaneously:

non-pipelined 8085



pipelined 8086



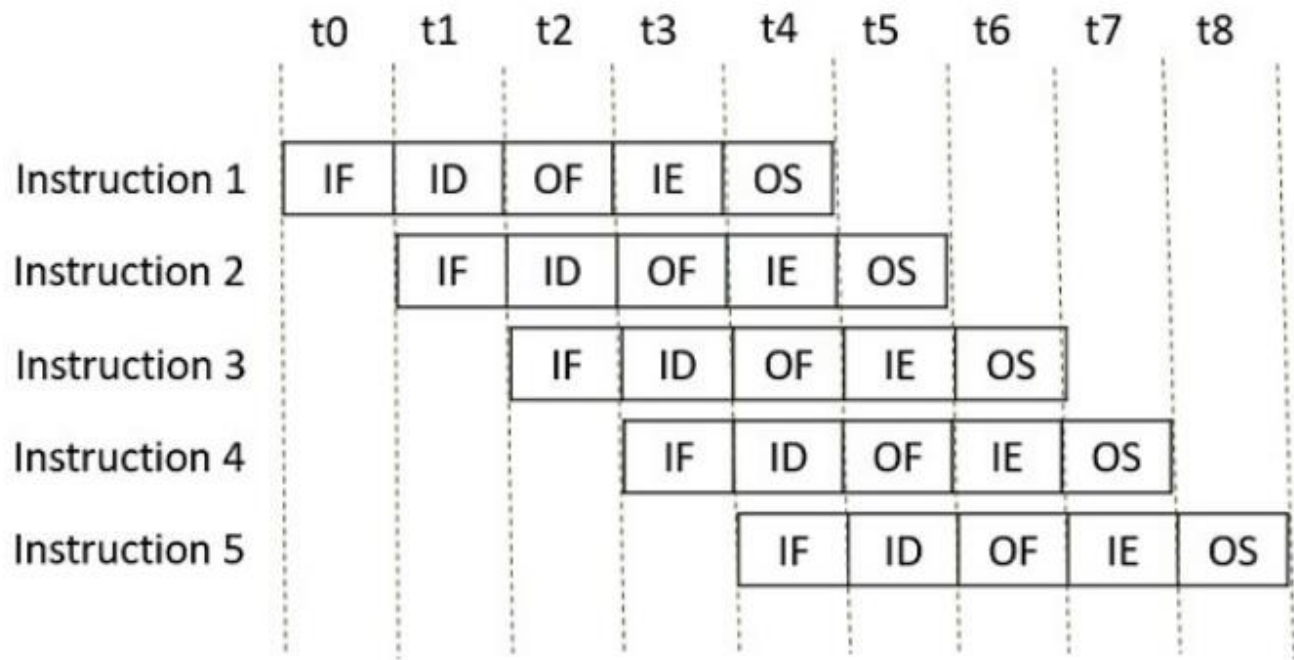
Q: Explain the operations of instructions queue in BIU.

Ans.)

The instruction queue is 6-bytes in length, operates on FIFO (first-in first out) basis. It receives the instruction codes from memory. BIU fetches the instructions for the instructions queue from memory.

- Cycle time of pipelining process

Look at the figure below the 5 instructions are pipelined. The first instruction gets completed in 5 clock cycle. After the completion of first instruction, in every new clock cycle, a new instruction completes its execution.



Pipelining of 5 Instructions

Observe that when the Instruction fetch operation of the first instruction is completed in the next clock cycle the instruction fetch of second instruction gets started. This way the hardware never sits idle it is always busy in performing some or other operation. But, no two instructions can **execute** their **same stage** at the **same clock cycle**.

Types of Pipelining:

In 1977 Handler classified pipeline processors depending on their functionality.

1. Arithmetic Pipelining

It is designed to perform high-speed floating-point addition, multiplication and division. Here, the multiple arithmetic logic units are built in the system to perform the parallel arithmetic computation in various data format. Examples of the arithmetic pipelined processor are Star-100, TI-ASC, Cray-1, Cyber-205.

2. Instruction Pipelining

Here, the number of instruction are pipelined and the execution of current instruction is overlapped by the execution of the subsequent instruction. It is also called **instruction look ahead**.

3. Processor Pipelining

Here, the processors are pipelined to process the **same data stream**. The data stream is processed by the first processor and the result is stored in the memory block. The result in the memory block is accessed by the second processor. The second processor reprocesses the result obtained by the first processor and the passes the refined result to the third processor and so on.

4. Unifunction Vs. Multifunction Pipelining

The pipeline performing the precise function every time is unifunctional pipeline. On the other hand, the pipeline performing multiple functions at a different time or multiple functions at the same time is multifunction pipeline.

5. Static vs Dynamic Pipelining

The static pipeline performs a fixed-function each time. The static pipeline is unifunctional. The static pipeline executes the same type of instructions continuously. Frequent change in the type of instruction may vary the performance of the pipelining.

Dynamic pipeline performs several functions simultaneously. It is a multifunction pipelining.

6. Scalar vs Vector Pipelining

Scalar pipelining processes the instructions with scalar operands. The vector pipeline processes the instruction with vector operands.

Advantages

1. Pipelining improves the throughput of the system.
2. In every clock cycle, a new instruction finishes its execution.
3. Allow multiple instructions to be executed concurrently.

Disadvantages

- Pipelining divides the instruction in 5 stages instruction fetch, instruction decode, operand fetch, instruction execution and operand store.
- The pipeline allows the execution of multiple instructions concurrently with the limitation that no two instructions would be executed at the **same stage** in the **same clock cycle**.
- All the stages must process at equal speed else the slowest stage would become the bottleneck.
- Whenever a pipeline has to stall for any reason it is a pipeline hazard.

• Pipeline Latency

Pipeline latency is defined as the number of clock cycles between an interrupt signal being asserted and the execution of the first instruction at the exception vector. It can vary widely, depending on the type of memory the processor is executing from and the impact of other host ports in your hardware. Theoretically, this time could be infinite if an ill-behaved host port blocks the processor from accessing memory, freezing the processor.

يتم تعريف استجابة انسيابية الاوامر على أنه عدد دورات الساعة بين إشارة المقاطعة التي يتم تأكيدها وتنفيذ التعليمات الأولى في متجه الاستثناء. يمكن أن يختلف بشكل كبير ، اعتماداً على نوع الذاكرة التي ينفذها المعالج وتأثير منافذ المضيف الأخرى في جهازك. من الناحية النظرية ، يمكن أن تكون هذه المرة لانهاية إذا قام منفذ مضيف سيئ السلوك بمنع المعالج من الوصول إلى الذاكرة ، مما يؤدي إلى تجميد المعالج.