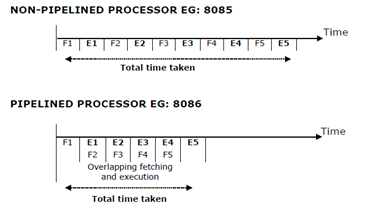
COURSE CODE: CSC222  
Unit: Assembly Language and Microprocessor.  
Name : Raha Habib  
REG No. COM/0501/21  
Number : 20

**Explain instruction prefetch (pipeling) in 8086 microprocessor.**

**Pipelining**

Pipelining is the reason processors are so **fast**. There was a time when processors took hours to do a simple calculation and now the same calculations can be done in seconds. Pipelining is the feature of fetching the next instruction while executing the current instruction.

Instructions are stored in memory, therefore, it has to be fetched, decoded and then executed. Pipelining boost performance as fetching and executing is done alongside at the same time.



**fig. Pipelined vs Non Pipelined processor**

In 8085, which has a **non-pipelined processor**for fetching and executing 5 instructions, 10-time cycles are required. While in 8086, for fetching and executing 5 instructions 6 times cycle is required. You can see the immense impact pipelining makes. In the pipeline, while one instruction is being executed side by side next **instruction** is being fetched.

Pipelining has become possible because of 6 bytes **pre-fetch queue**. It is a 6 byte FIFO RAM used to implement pipelining. BIU (Bus Interface Unit) fetches the 6 instruction byte from the code segment and stores them into the queue. 8086 has a 16-bit data bus thus code is refilled in the queue when at least 2 bytes are empty.

**Drawbacks:**

1. Data dependency- A situation where instruction depends on a result of a sequentially previous instruction before it can complete execution.

2. Branching- Sometimes it may happen that the program is not executing sequentially because of loops and conditional statements. But with the help of the pipeline, we have fetched the next instruction. In that case, we discard that instruction and jump. It creates a waste of effort.

**Advantages:**

1. Pipelining boost performance.  
2. Fetching and executing is done alongside at the same time.  
3. Saves time.  
4. Fewer instruction cycles are required.