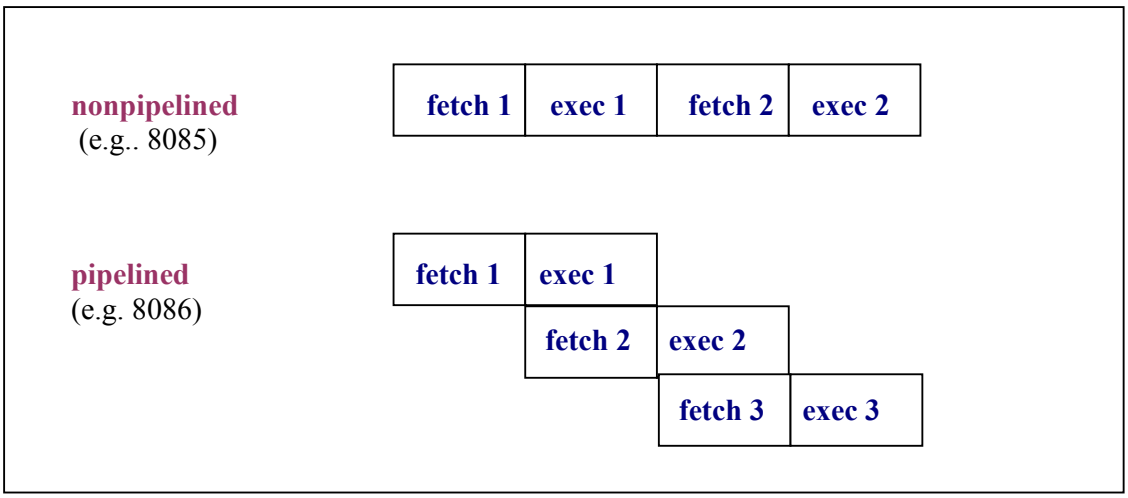
Pipelining

- In the 8085 microprocessor, the CPU could either fetch or execute at a given time. CPU had to fetch an instruction from the memory, then execute it, then fetch again and execute it and so on..
- Pipelining is the simplest form to allow the CPU to ***fetch*** and ***execute*** at the same time. Note that the fetch and execute times can be different.

Pipelined vs. Nonpipelined Execution



- Intel implemented the concept of pipelining by splitting the internal structure of 8088/86 into two sections.
  - the ***execution unit*** (EU)
  - the ***bus interface unit*** (BIU)
  - These two sections work simultaneously. BIU accesses memory and peripherals while the EU executes the instructions previously fetched.
  - It only works if BIU keeps ahead of EU. Thus BIU has a buffer of ***queue***. (8088 has 4 byte, and 8088 has 6 bytes).
  - If the execution of any instruction takes to long, the BIU is filled to its maximum capacity and busses will stay idle. It starts to fetch again whenever there is 2-byte room in the queue.
  - When there is a jump instruction, the microprocessor must flush out the queue. When a jump instruction is executed BIU starts to fetch information from the new location in the memory. In this situation EU must wait until the BIU starts to fetch the new instruction. This is known as ***branch penalty***.