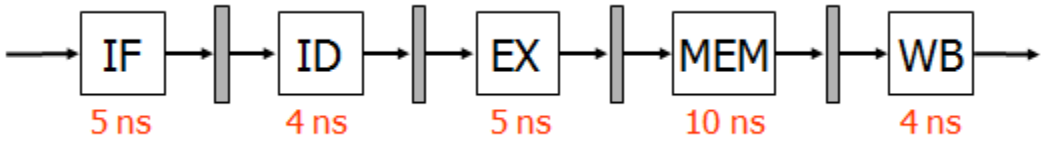
Part 7

Pipelining

* *Pipelining* is an implementation technique used to build fast processors. It allows the execution of multiple instructions to overlap in time.

Pipeline Stages



The computer pipeline divide the instruction processing into stages. Each stage completes a part of an instruction and loads a new part in parallel.

We can divide the execution of an instruction into the following 5 "classic" stages:

IF: Instruction Fetch

ID: Instruction Decode, register fetch

EX: Execution

MEM: Memory Access

WB: Register write Back

Example: Compute the time needed to process three instructions in four-stage pipelining versus sequential technique.

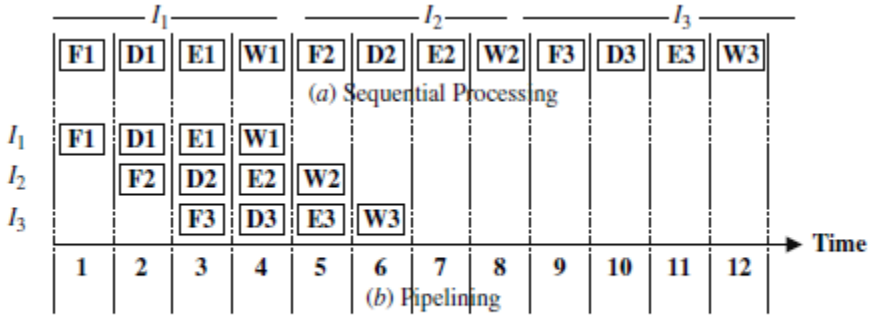


Figure 9.1 Pipelining versus sequential processing

It is clear from the figure that the total time required to process three instructions (I1, I2, I3) is only **six** time units if four-stage pipelining is used as compared to **12** time units if sequential processing is used.

Advantages/Disadvantages

Advantages:

- More efficient use of processor
- Quicker time of execution of large number of instructions

Disadvantages:

- Pipelining involves adding hardware to the chip
- Inability to continuously run the pipeline at full speed because of pipeline hazards which disrupt the smooth execution of the pipeline.

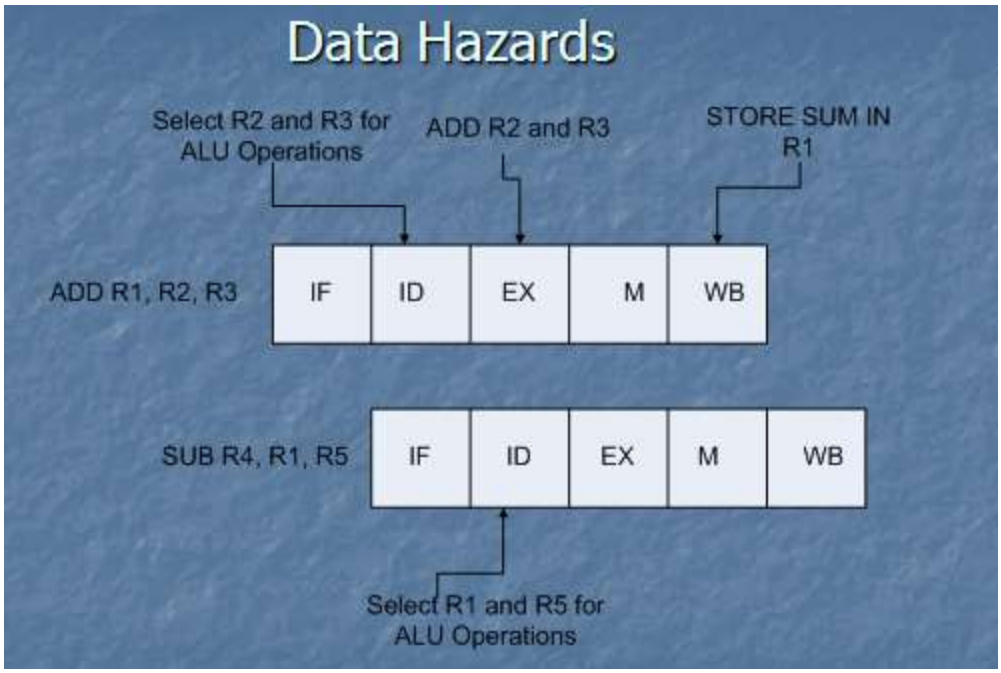
Pipeline Throughput and Latency:

Pipeline throughput: instructions completed per second

Pipeline latency: how long does it take to execute single instruction in the pipeline.

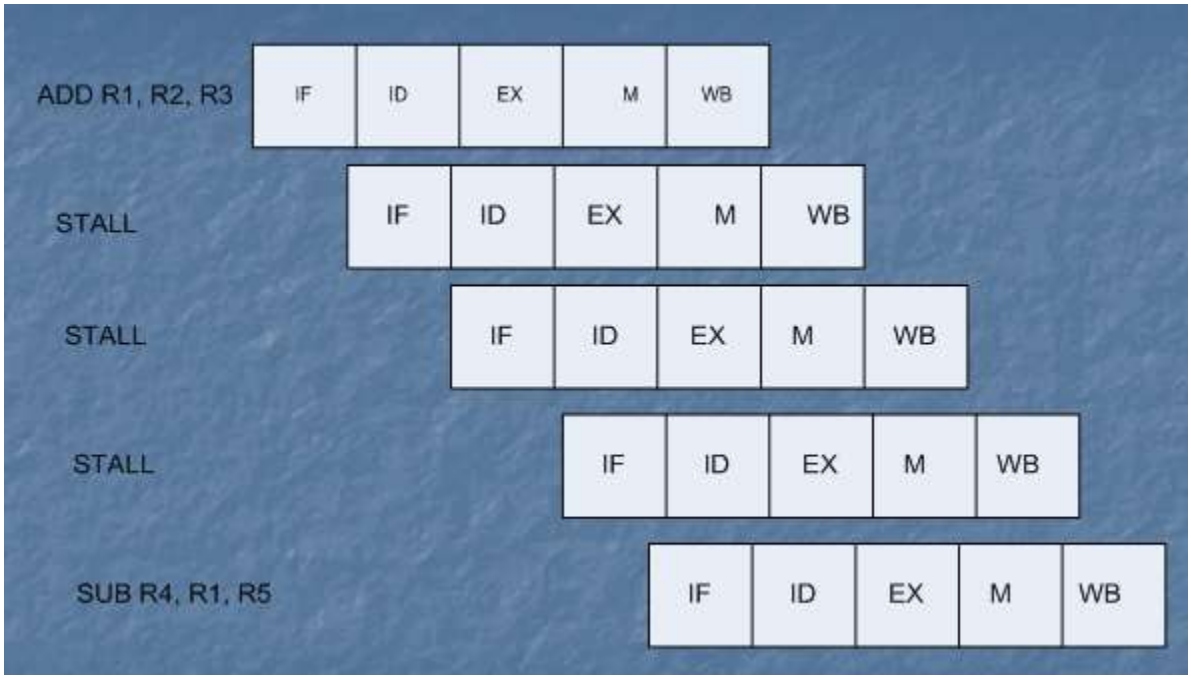
Pipeline Hazards

- Data Hazards – an instruction uses the result of the previous instruction. A hazard occurs exactly when an instruction tries to read a register in its ID stage that an earlier instruction intends to write in its WB stage.
- Control Hazards – the location of an instruction depends on previous instruction For example **Branch** instruction affects PC(program counter) contents .
- Structural Hazards – two instructions need to access the same resource



Stalling:

- Stalling involves halting the flow of instructions until the required result is ready to be used. However stalling wastes processor time by doing nothing while waiting for the result.



Type of Pipelining:

- Software Pipelining

1) Can Handle Complex Instructions

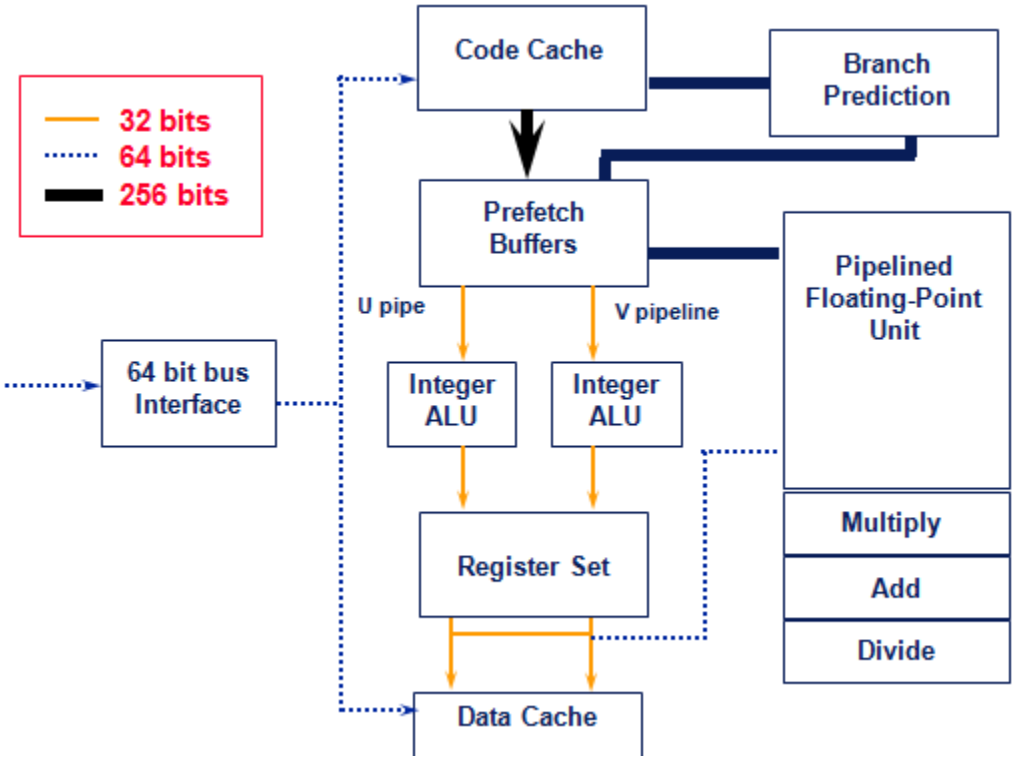
2) Allows programs to be reused

- Hardware Pipelining

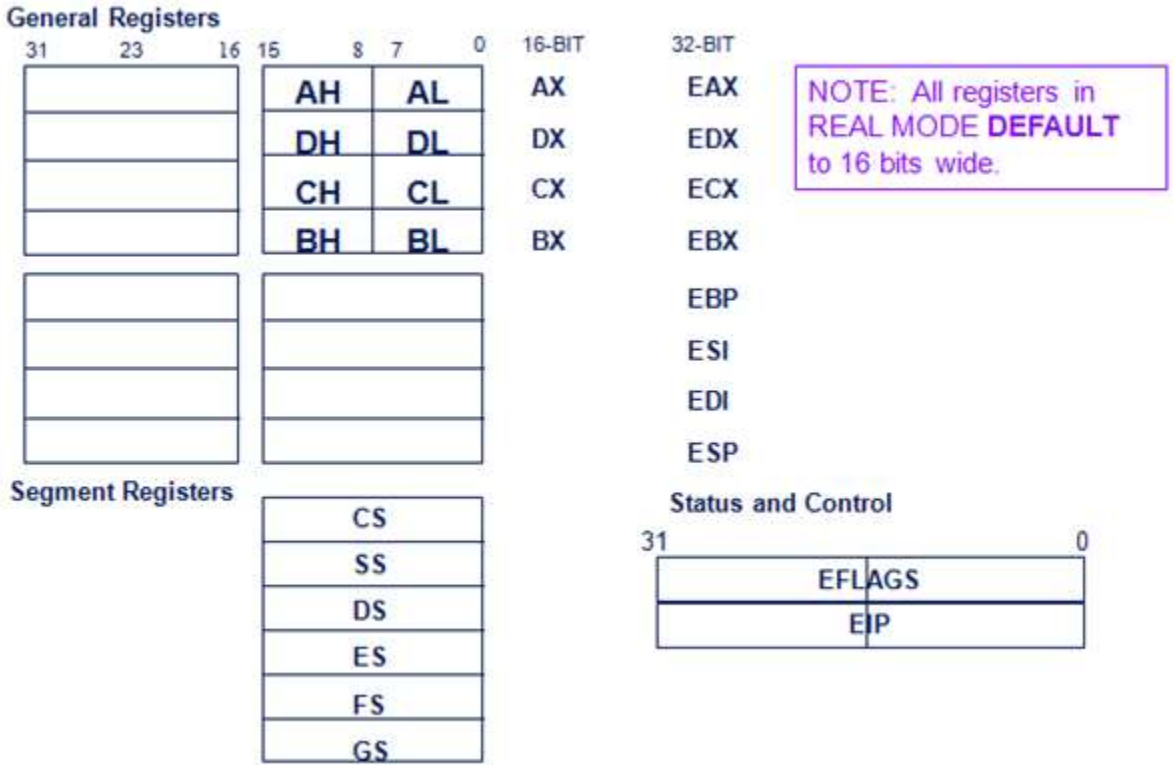
1) Help designer manage complexity – a complex task can be divided into smaller, more manageable pieces.

2) Hardware pipelining offers higher performance

Pentium processor architecture:



Pentium registers:



Pipelining on the 486/Pentium

- 486 has a 5-stage pipeline
  - Fetch
    - Instructions can have variable length and can make this stage out of sync with other stages. This stage actually fetches about 5 instructions with a 16 byte load
  - Decode1
    - Decode opcode, addressing modes – can be determined from the first 3 bytes
  - Decode2
    - Expand opcode into control signals and more complex addressing modes
  - Execute
  - Write Back
    - Store value back to memory or to register file