

Assembly Language for Intel-Based Computers, 4th Edition

Kip R. Irvine

Chapter 2: IA-32 Processor Architecture

Slides prepared by Kip R. Irvine

Revision date: 09/25/2002

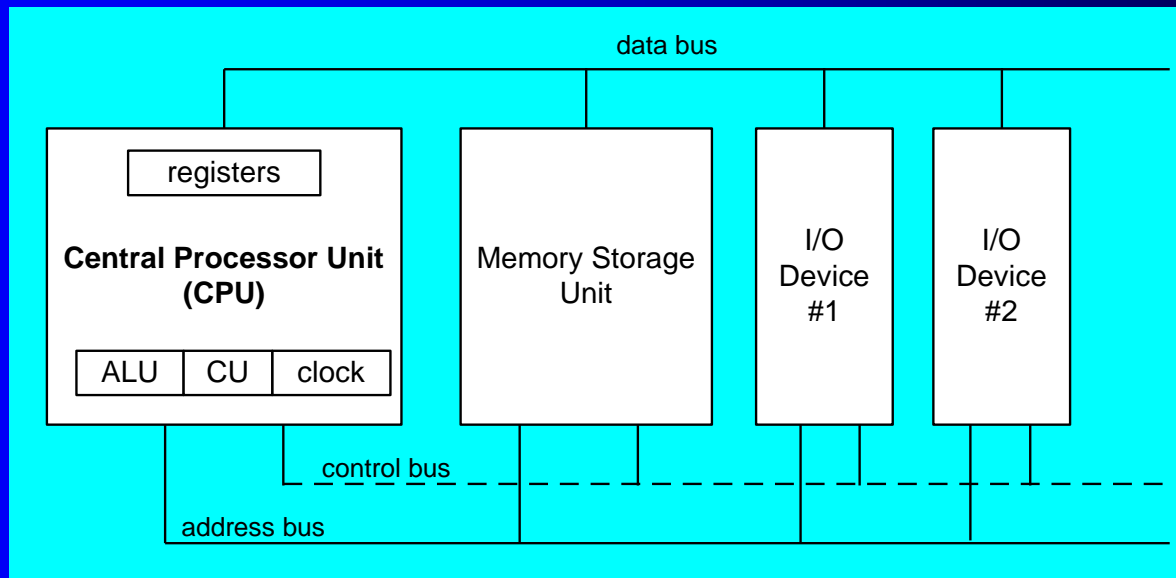
Modified by Dr. Nikolay Metodiev Sirakov, 2005, 2012, 2015

- [Chapter corrections](#) (Web) [Assembly language sources](#) (Web)
- [Printing a slide show](#)

(c) Pearson Education, 2002. All rights reserved. You may modify and copy this slide show for your personal use, or for use in the classroom, as long as this copyright statement, the author's name, and the title are not changed.

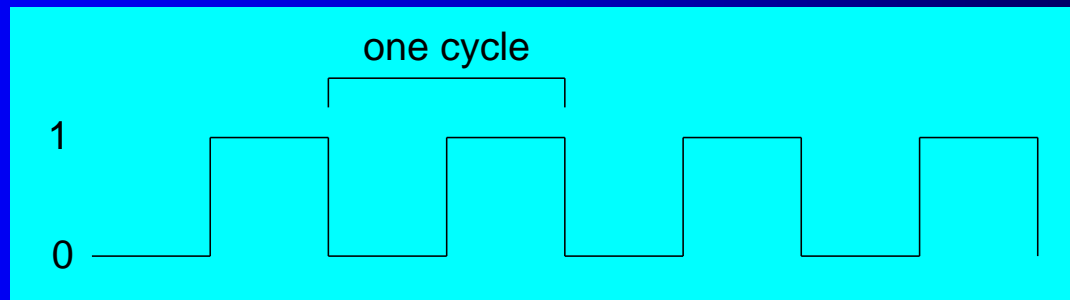
Basic Microcomputer Design

- clock synchronizes CPU operations
- control unit (CU) coordinates sequence of execution steps
- ALU performs arithmetic and bitwise processing
- The CPU size is not inconsideration.



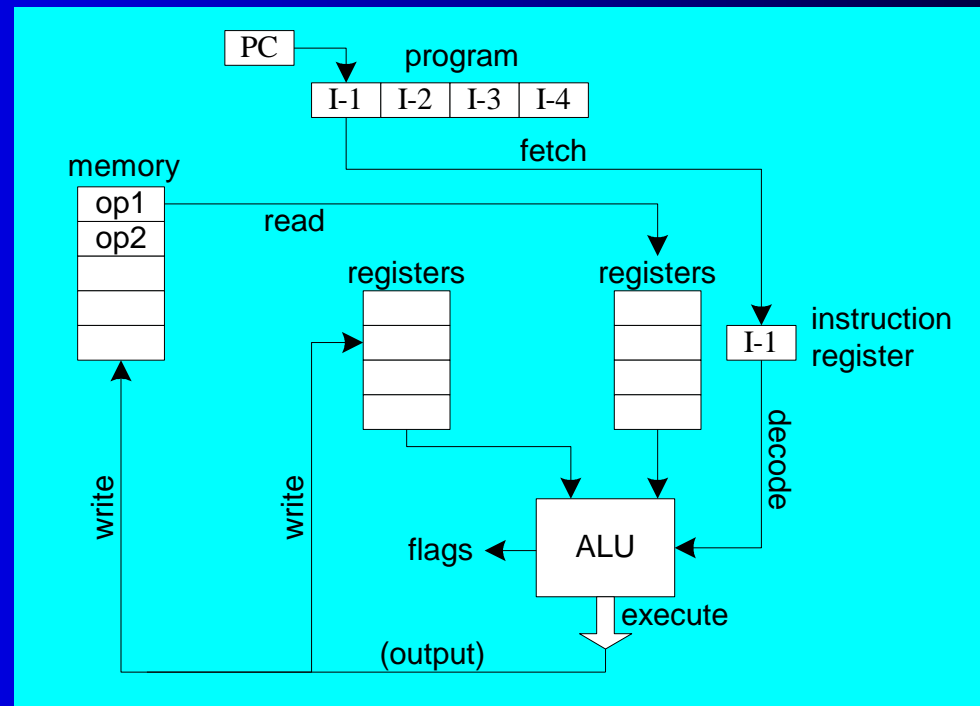
Clock

- synchronizes all CPU and BUS operations
- machine (clock) cycle measures time of a single operation
- clock is used to trigger events



Instruction Execution Cycle

- Fetch
- Decode
- Fetch operands
- Execute
- Store output



Multi-Stage Pipeline

- Pipelining makes it possible for processor to execute instructions in parallel
- Instruction execution divided into discrete stages

Example of a non-pipelined processor. Many wasted cycles.

| | | Stages | | | | | |
|--------|----|--------|-----|-----|-----|-----|-----|
| | | S1 | S2 | S3 | S4 | S5 | S6 |
| Cycles | 1 | I-1 | | | | | |
| | 2 | | I-1 | | | | |
| | 3 | | | I-1 | | | |
| | 4 | | | | I-1 | | |
| | 5 | | | | | I-1 | |
| | 6 | | | | | | I-1 |
| | 7 | I-2 | | | | | |
| | 8 | | I-2 | | | | |
| | 9 | | | I-2 | | | |
| | 10 | | | | I-2 | | |
| | 11 | | | | | I-2 | |
| | 12 | | | | | | I-2 |

Pipelined Execution

- More efficient use of cycles, greater throughput of instructions:

| | | Stages | | | | | |
|--------|---|--------|-----|-----|-----|-----|-----|
| | | S1 | S2 | S3 | S4 | S5 | S6 |
| Cycles | 1 | I-1 | | | | | |
| | 2 | I-2 | I-1 | | | | |
| | 3 | | I-2 | I-1 | | | |
| | 4 | | | I-2 | I-1 | | |
| | 5 | | | | I-2 | I-1 | |
| | 6 | | | | | I-2 | I-1 |
| | 7 | | | | | | I-2 |

For k states and n instructions, the number of required cycles is:

$$k + (n - 1)$$

Wasted Cycles (pipelined)

- When one of the stages requires two or more clock cycles, clock cycles are again wasted. **This may happen if an instruction which requires memory operation is performed.**

| | | Stages | | | | | |
|--------|----|--------|-----|-----|-----|-----|-----|
| | | S1 | S2 | S3 | S4 | S5 | S6 |
| Cycles | 1 | I-1 | | | | | |
| | 2 | I-2 | I-1 | | | | |
| | 3 | I-3 | I-2 | I-1 | | | |
| | 4 | | I-3 | I-2 | I-1 | | |
| | 5 | | | I-3 | I-1 | | |
| | 6 | | | | I-2 | I-1 | |
| | 7 | | | | I-2 | | I-1 |
| | 8 | | | | I-3 | I-2 | |
| | 9 | | | | I-3 | | I-2 |
| | 10 | | | | | I-3 | |
| | 11 | | | | | | I-3 |

For k states and n instructions, the number of required cycles is:

$$k + (2n - 1)$$

Superscalar

A superscalar processor has multiple execution pipelines. In the following, note that Stage S4 has left and right pipelines (u and v).

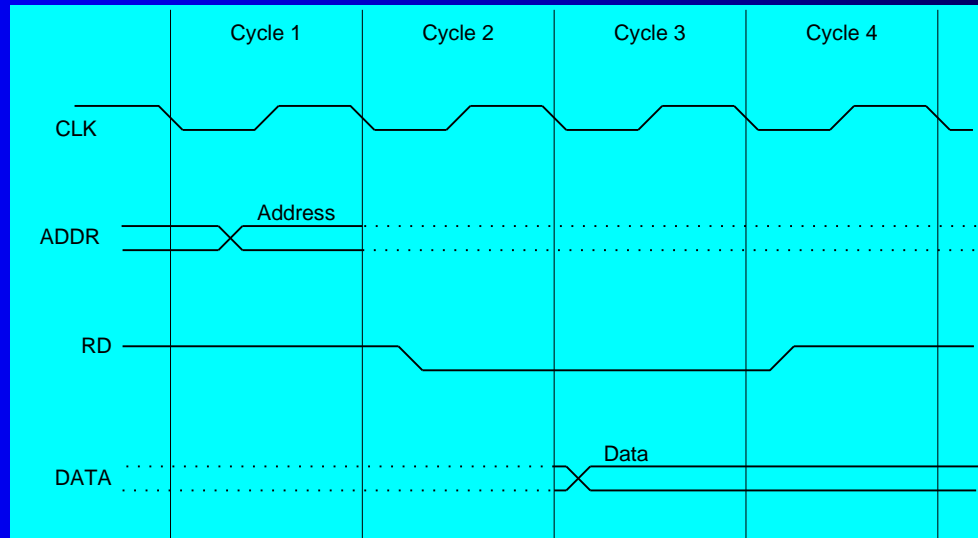
| | | Stages | | | | | | |
|--------|----|--------|-----|-----|-----|-----|-----|-----|
| | | S1 | S2 | S3 | S4 | | S5 | S6 |
| Cycles | 1 | I-1 | | | | | | |
| | 2 | I-2 | I-1 | | | | | |
| | 3 | I-3 | I-2 | I-1 | | | | |
| | 4 | I-4 | I-3 | I-2 | I-1 | | | |
| | 5 | | I-4 | I-3 | I-1 | I-2 | | |
| | 6 | | | I-4 | I-3 | I-2 | I-1 | |
| | 7 | | | | I-3 | I-4 | I-2 | I-1 |
| | 8 | | | | | I-4 | I-3 | I-2 |
| | 9 | | | | | | I-4 | I-3 |
| | 10 | | | | | | | I-4 |

For k states and n instructions, the number of required cycles is:

$$k + n$$

Reading from Memory

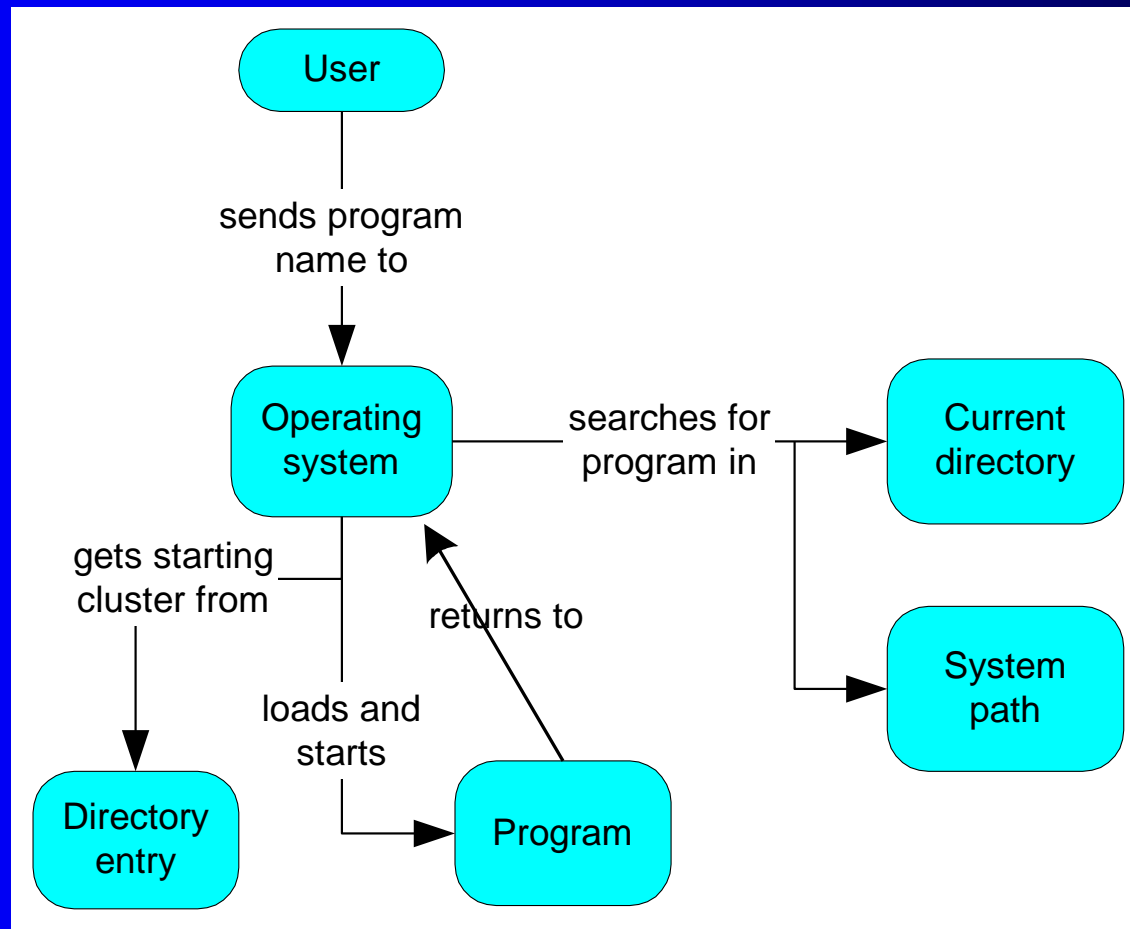
- Multiple machine cycles are required when reading from memory, because it responds much more slowly than the CPU. The steps are:
 - address placed on address bus
 - Read Line (RD) set low
 - CPU waits one cycle for memory to respond
 - Read Line (RD) goes to 1, indicating that the data is on the data bus



Cache Memory

- High-speed expensive static RAM both inside and outside the CPU.
 - Level-1 cache: inside the CPU
 - Level-2 cache: outside the CPU
- Cache hit: when data to be read is already in cache memory
- Cache miss: when data to be read is not in cache memory.

How a Program Runs



Multitasking

- OS can run multiple programs at the same time.
- Multiple threads of execution within the same program.
- Scheduler utility assigns a given amount of CPU time to each running program.
- Rapid switching of tasks
 - gives illusion that all programs are running at once
 - the processor must support task switching.