TDTS 10

Computer Architecture

[*Datorarkitektur*]

www.ida.liu.se/~TDTS10

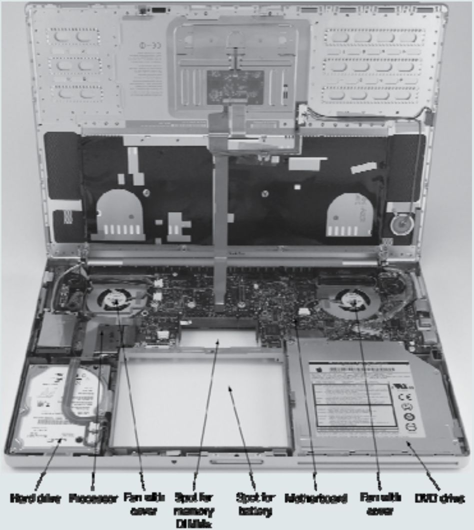
Zebo Peng

Embedded Systems Laboratory (ESLAB)
Dept. of Computer and Information Science (IDA)
Linköping University

Teachers and Contact Info

- Zebo Peng, course leader and examiner
  Email: zebo.peng@liu.se
- Arian Maghazeh, course and lab assistant
  Email: arian.maghazeh@liu.se
- Rouhollah Mahfouzi, lab assistant
  Email: rohollah.mahfoozi@liu.se
- Mina Niknafs, lab assistant
  Email: mina.niknafs@liu.se

Objectives



- How does a computer work?
  - Hardware components
  - Structure & interconnects
  - Program execution
- Terminology and definition:
  - Architecture concepts
  - Performance evaluation
  - Technical specification

Course Information

- **Website** — https://www.ida.liu.se/~TDTS10
- **Lectures**
  - 8 lectures.
  - Lecture notes will be available at the website, usually one day before each lecture.
  - The whole set of last year's lecture notes is at the website.
  - Lecture notes for the course book are also available at the website.
- **Examination**
  - Written exam, closed book.
  - The questions can be answered either in English or Swedish.
  - Previous exam examples are given at the website.

Course Information (Cont'd)

- **Literature**
  - William Stallings: Computer Organization and Architecture, 11th edition, Peason, 2018.
  - http://williamstallings.com/ComputerOrganization/
    - Student resources, including relevant information and documents.
  - You can also use:
    - Older editions of Stallings' book.
    - Books covering the same subjects.

      Ex. Hennessy and Patterson: "Computer Architecture: A Quantitative Approach."

Course Information (Cont'd)

- **Labs**
  - Hands-on exercises with concepts taught in the course.
  - Use a tool for architecture evaluation via simulation.
  - Give insights in various trade-offs involved in the design of computers.
  - Enhance the understanding of several advanced concepts.

- **Lab assignments** (6 lab sessions of 2 hours each)
  - Lab 1: Cache Memories.
  - Lab 2: Instruction Pipelining.
  - Lab 3: Superscalar Processors.

- Please build groups of two students and sign up for the labs in the website before November 13.

- Additional information to be given in the lab seminar (lesson)
  - Tuesday November 13, 15-17.
  - Two lesson groups, one by Arian, and the other by Rouhollah.

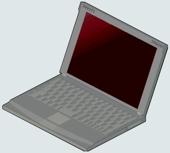
Lecture 1

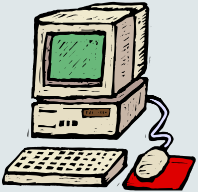
- Introduction and technology trend
- Central Processing Unit (CPU)
- Instruction execution

Many Definitions

- Computer architecture refers to those attributes of a computer that are visible to programmers, or have a direct impact on the logical execution of programs.
- The theory behind the design of a computer.
- The conceptual design and fundamental operational structure of a computer system.
- The arrangement of computer components and their relationships.
- …

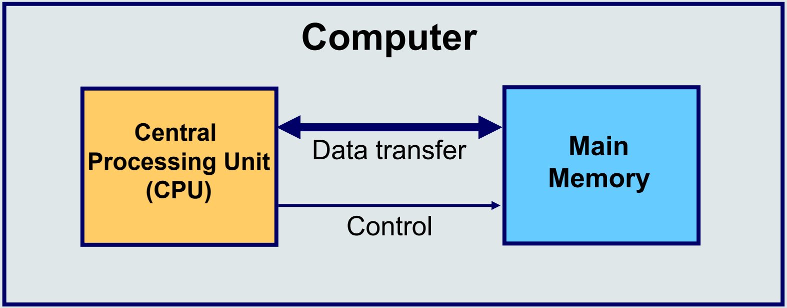
What is a Computer?





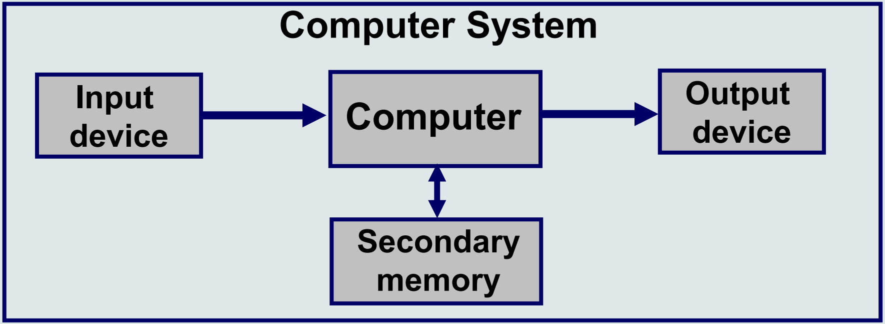
Definition of a Computer

- A computer is a data processing machine that is operated automatically under the control of a list of instructions stored in its main memory.

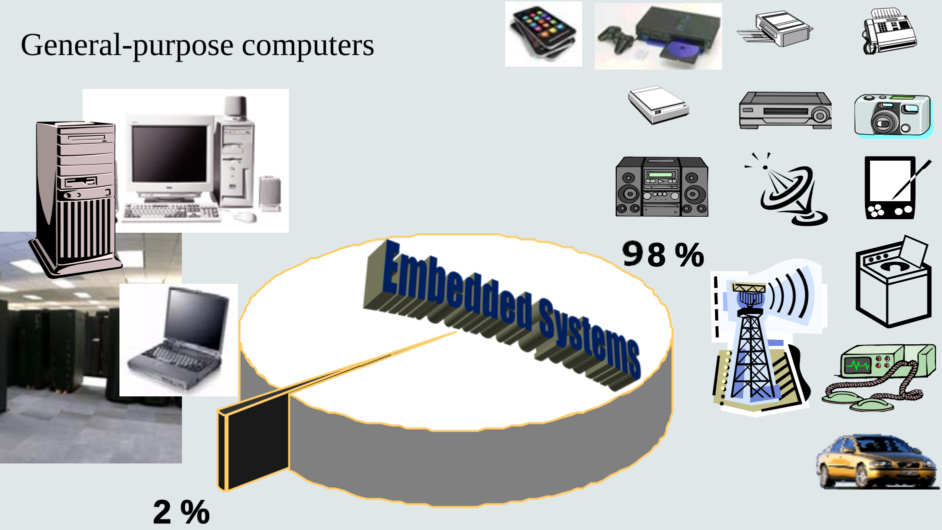


A Computer System

- A **computer system** consists of a computer and its peripherals.
- **Computer peripherals** include input devices, output devices, and secondary memories.



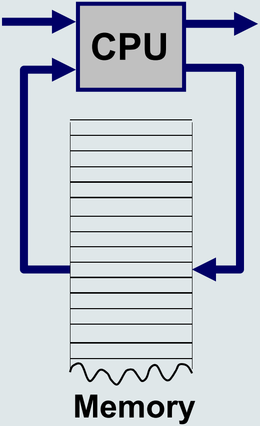
Microprocessor Market Share



Basic Principles of Computers

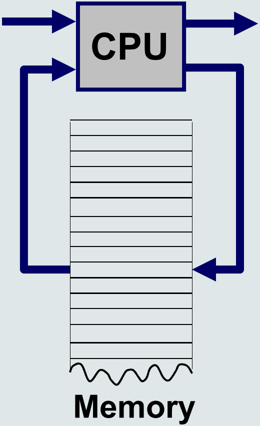
Virtually all modern computer designs are based on the <u>von Neumann architecture</u> principles:

- Data and instructions are stored in a single read/write memory.
- The contents of this memory are addressable by location, without regard to what are stored there.
- Instructions are executed sequentially (from one instruction to the next) unless the order is explicitly modified.

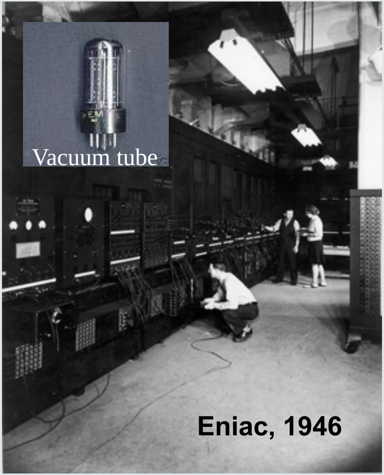


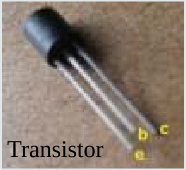
Why von Neumann Architecture?

- General-purpose, programmable.
  - They can solve very different problems by executing different programs.
- Instruction execution is done automatically.
- It can be built with very simple electronics components:
  - Data processing function is performed by electronic gates.
  - Data storage function is provided by memory cells.
  - Data communication is achieved by electrical wires.

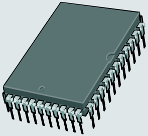


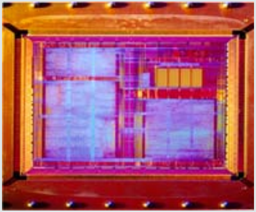
Technology Development


Vacuum tube
Eniac, 1946



Transistor

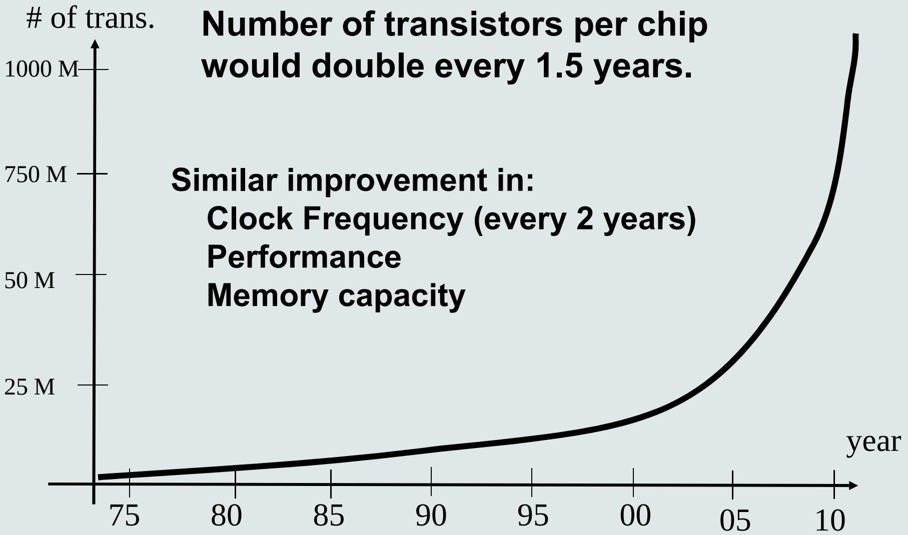

Integrated circuit


VLSI circuit

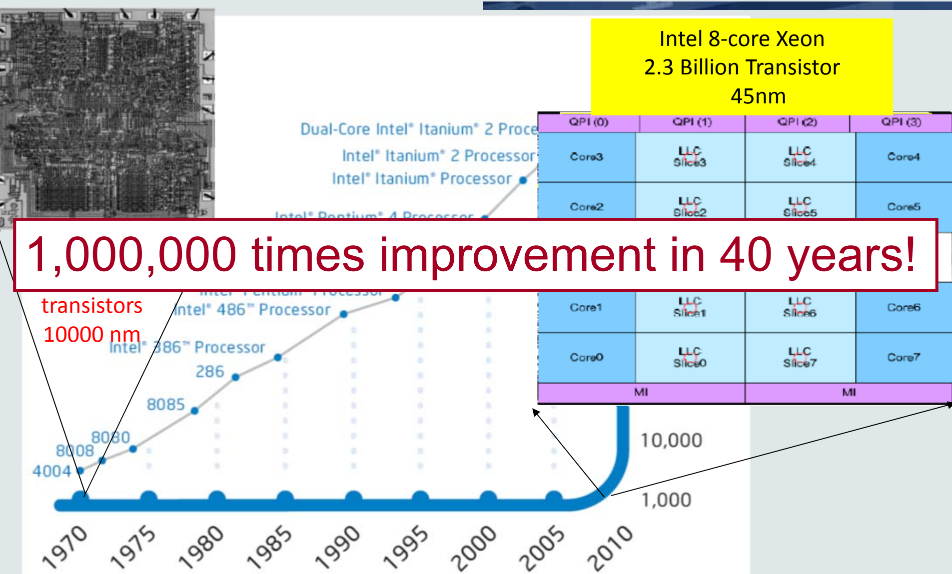
Moore's Law

Number of transistors per chip would double every 1.5 years.

Similar improvement in:
- **Clock Frequency (every 2 years)**
- **Performance**
- **Memory capacity**



Intel Microprocessor Evolution



1,000,000 times improvement in 40 years!

Images courtesy of Intel Corporation

Lecture 1

- Introduction and technology trend

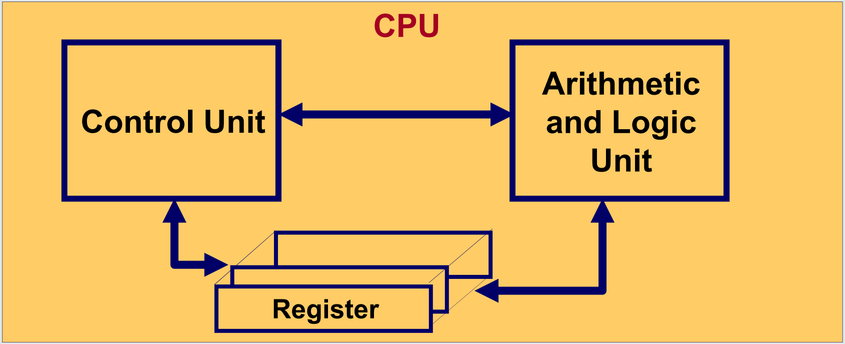
- Central Processing Unit (CPU)

- Instruction execution

Central Processing Unit (CPU)

The Central Processing Unit (CPU), also called processor, includes two main units:

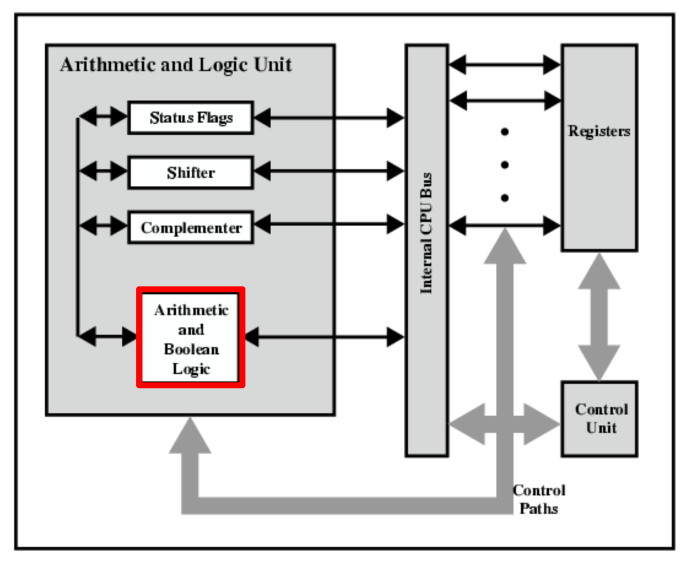
- A program control unit (CU), and
- An Arithmetic and Logic Unit (ALU).



CPU (Cont'd)

- The primary function of a CPU is to execute the instructions stored in the main memory.
- An instruction tells the CPU to perform one of its basic operations.
- The CPU includes a set of registers, as temporary storage devices, used to hold control information, key data, and intermediate results.
- It includes also an internal bus, which provides data movement paths among the control unit, ALU, and registers.
- The control unit interprets (decodes) the instruction to be executed and "tells" the other components what to do.

A CPU Example



Representation of Data

- Inside a computer, data and control information are all represented in binary format, with only two basic symbols, 0 and 1.
- The two basic symbols are usually represented by electronics signals.
- Data are represented as a sequence of bits, such as 10100001 (a byte consists of 8 bits).
- Different coding systems have been used. One commonly used system is ASCII (American Standard Code for Information Interchange).
  - Ex. The letter "A" is coded as 01000001.

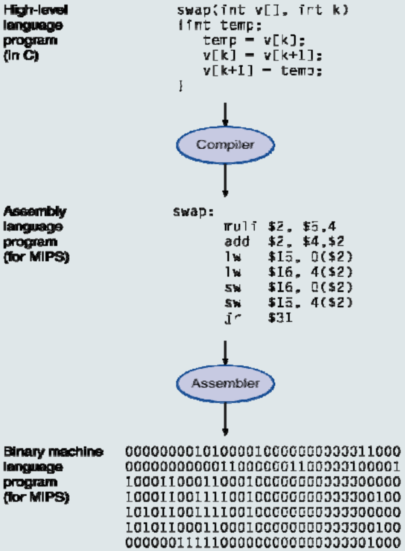
Representation of Numeric Data

- For numeric data, the binary system uses the same positional scheme as the decimal system.
- The positional values are factors of 2, i.e., 1, 2, 4, 8, 16, ..., instead of 10.
- Binary numbers are added, subtracted, multiplied, and divided directly, inside a computer.

Position weights:		32	16	8	4	2	1
		(0)	(0)	(1)	(1)	(0)	(Carries)
7	...	0	0	0	1	1	1
+ 6	...	0	0	0	1	1	0
= 13	...	(0) 0	(0) 0	(0) 1	(1) 1	(1) 0	(0) 1

Program Code Hierarchy

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Machine language
  - Encoded instructions and data in binary format



Lecture 1

- Introduction and technology trend

- Central Processing Unit (CPU)

- Instruction execution

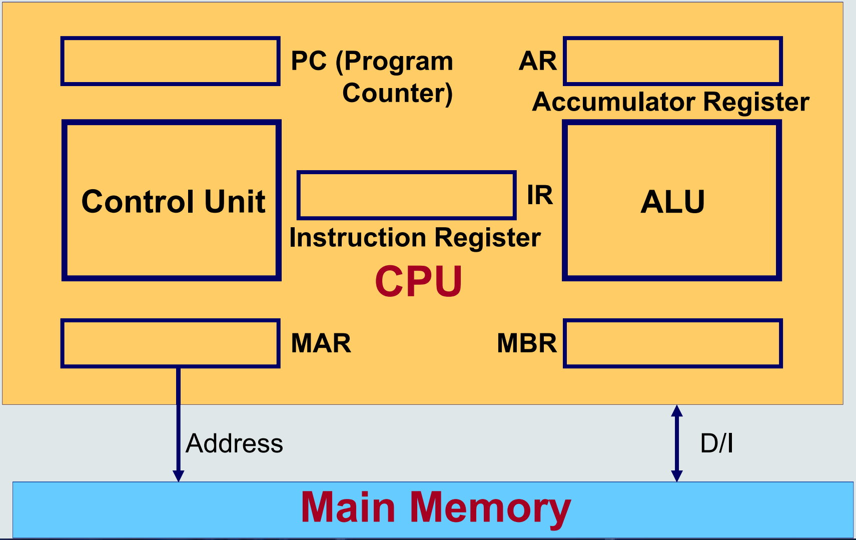
Machine Instructions

- The CPU can only execute machine code in binary format, called machine instructions.
- A machine instruction specifies the following information:
  - What has to be done (operation code)
  - To whom the operation applies (source operands)
  - Where does the result go (destination operand)
  - How to continue after the operation is finished (next instruction address).
- Machine instructions are of four types:
  - Arithmetic and logic operations.
  - Data transfer between memory and CPU registers.
  - Program control (e.g., conditional branches).
  - I/O transfer.

Some Instruction Examples

OP	Name	Assembly Code	Operation	Time (ns)
000	Read	LOAD A	Mem[A] -> AR	2
001	Write	STORE A	AR -> Mem[A]	2
010	Addition	ADD A	AR + Mem[A] -> AR	2
011	Subtraction	SUB A	AR - Mem[A] -> AR	2
100	Branch	BRA A	A -> PC	1
101	Branch if not zero	BNZ	A -> PC if AR ≠ 0, ELSE PC + 1 -> PC	1
110	Input	IN	IN_port -> AR	1
111	Output	OUT	AR -> OUT_port	1

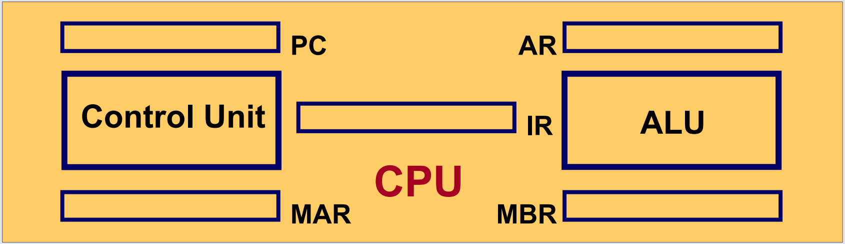
Instruction Execution Mechanism



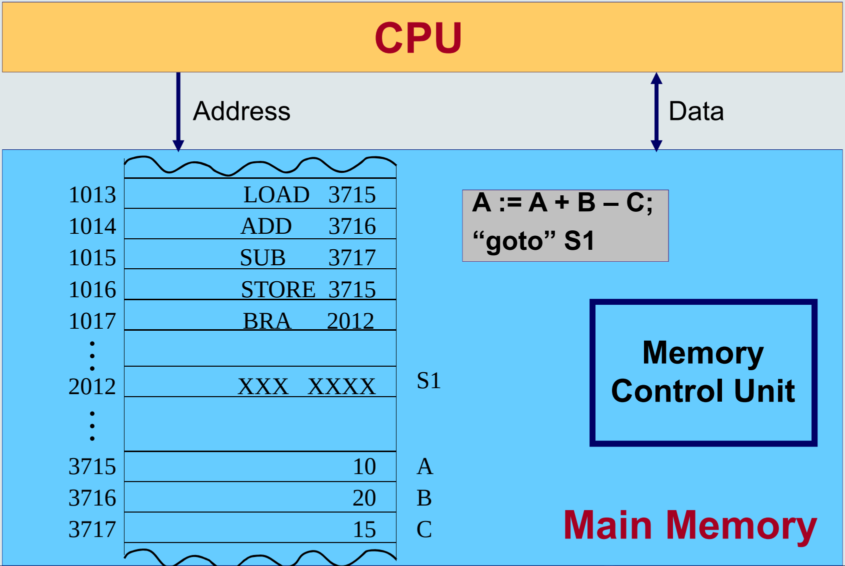
Instruction Execution

fetch cycle		execute cycle	
fetch	decode	execute	

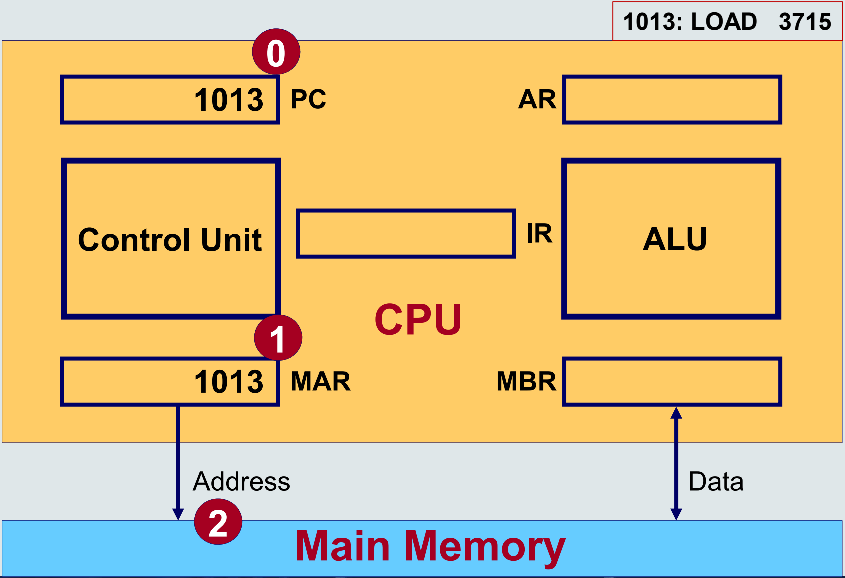
fetch	decode	execute
PC -> MAR	Decode(IR)	Perform the specified operation
M[MAR] -> MBR		
MBR -> IR		(memory access may be needed)
PC + 1 -> PC		(PC may be changed)



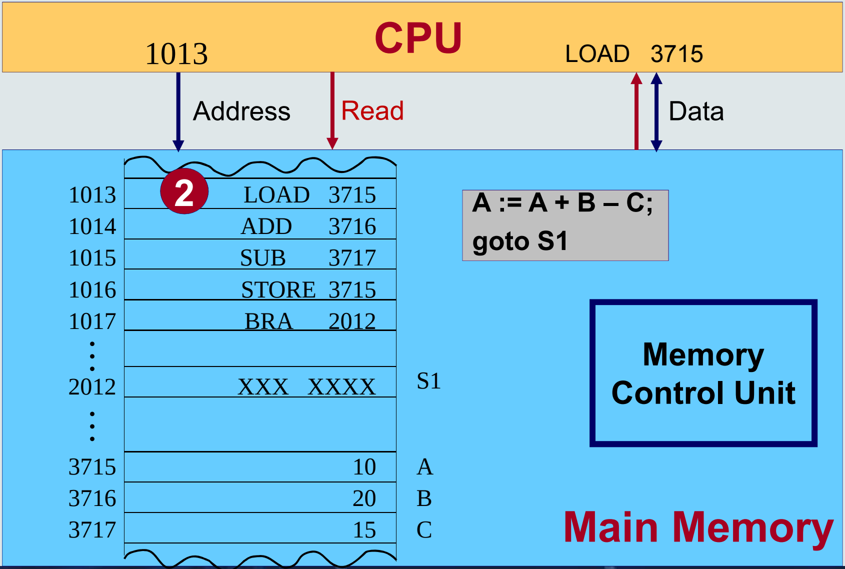
Instruction Execution Example



Instruction Execution Example

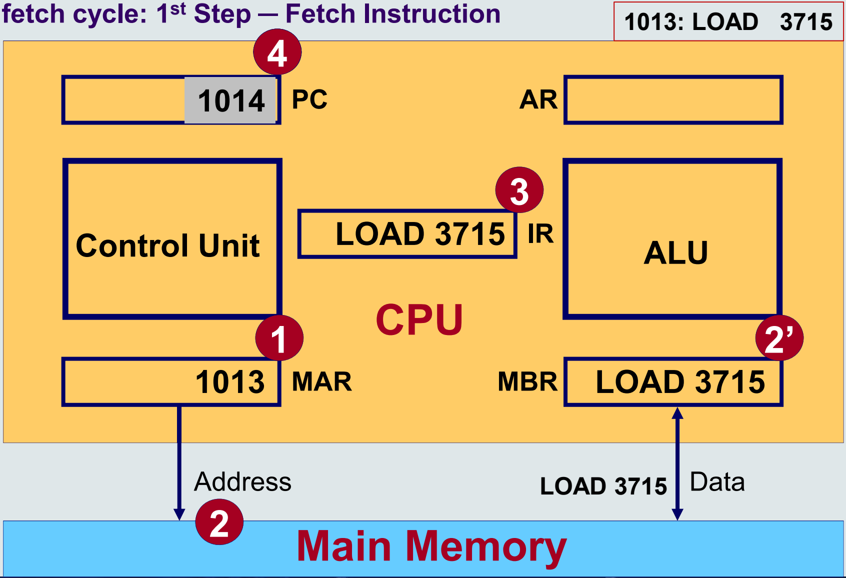


Instruction Execution Example



Instruction Execution Example

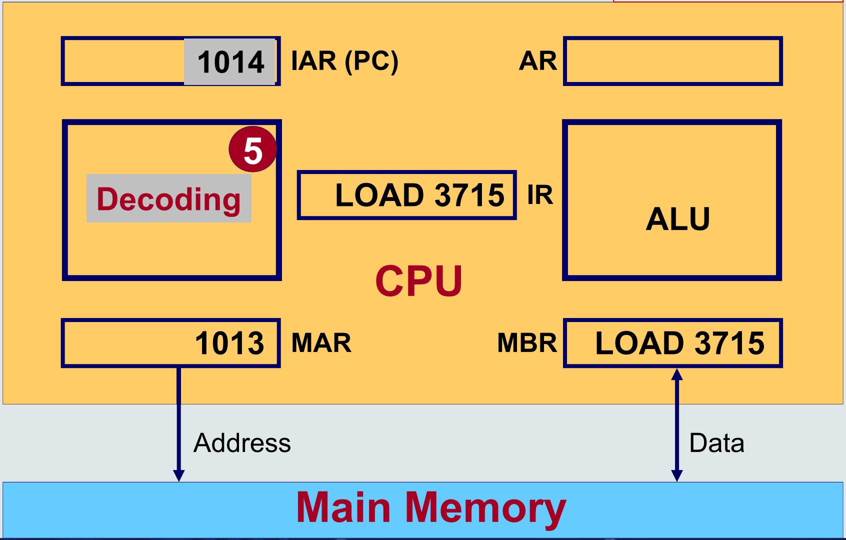
The fetch cycle: 1st Step — Fetch Instruction



Instruction Execution Example

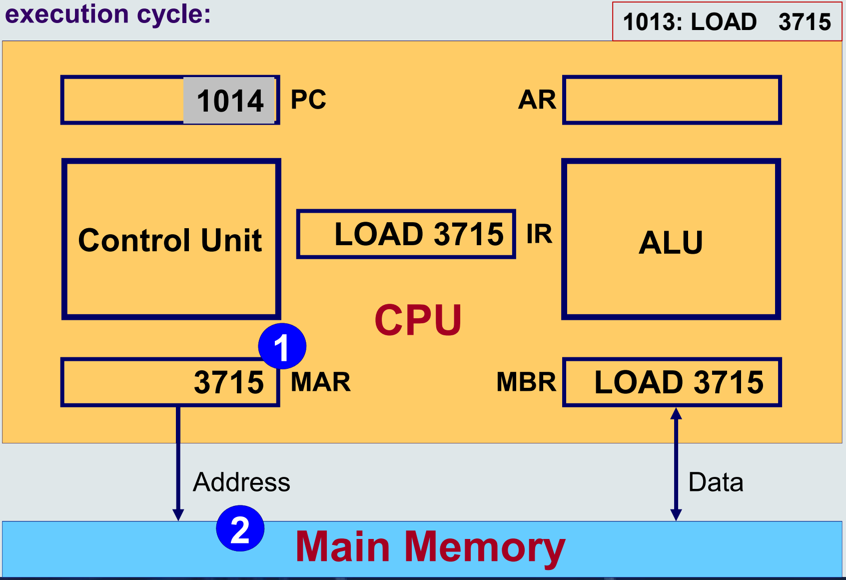
The fetch cycle: 2nd Step — Decode Instruction

1013: LOAD 3715

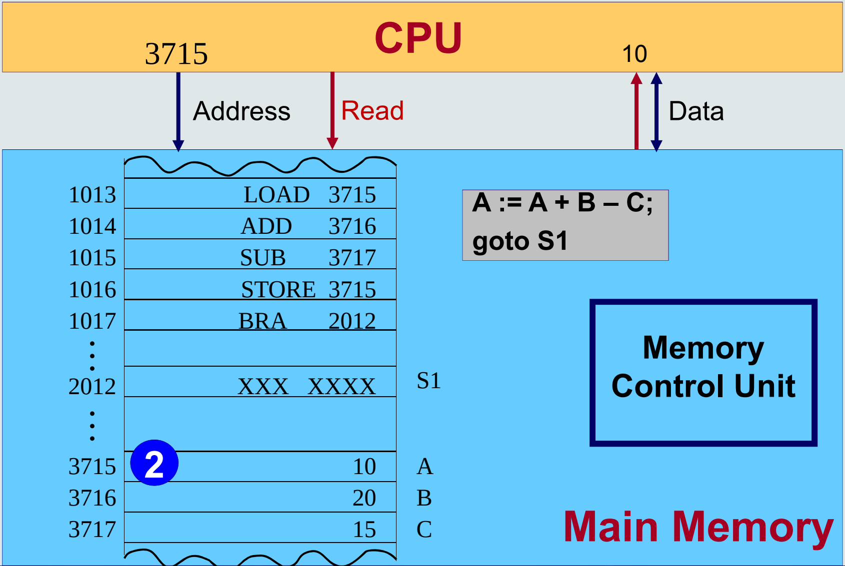


Instruction Execution Example

The execution cycle:

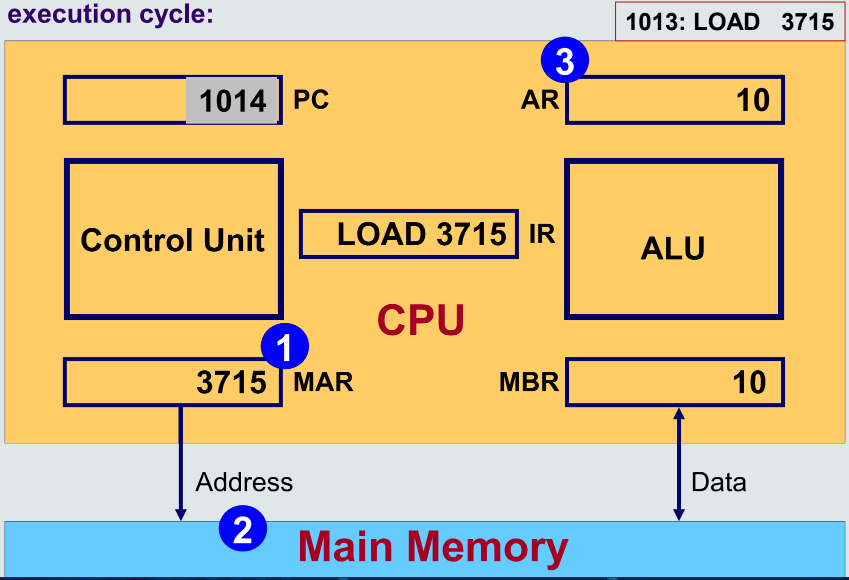


Instruction Execution Example



Instruction Execution Example

The execution cycle:



Machine Cycles and Performance

- The execution of an instruction is carried out in a machine cycle (instruction cycle).
- The CPU executes one instruction after the other, cycle by cycle, repeatedly.
- The machine cycle time (or instruction execution time) of a computer gives an indication of its performance (speed).
  - Ex. A computer can have a performance of 733 MIPS (Millions of Instructions Per Second).
- Since different instructions need different time to execute, the average instruction execution time is often used.
- Very common, FLOPS (FLoating-point Operations Per Second) is used nowadays.
  - Ex. A PC can have a performance of 10 GigaFLOPS.

Summary

A computer executes repeatedly a series of instructions (called programs) stored in its main memory:

- It performs data processing operations specified by the programs.
- It runs the programs automatically, with no need for human intervention.
- It can perform the operations in extremely high speed.
- It can store and manipulate a large amount of data.
- It can communicate with each other and with users in an efficient way.

The huge computation power is due to rapid technology development and architecture innovation.