

2 Computer organization

Computer Architecture

2.1.1 The central processing unit (CPU) and its elements

Fig. 2.1 The input, process, and output model

- Input: data and instructions
- Output:
- Process:
- Storage: permanent memory, non-volatile

CPU

- Hardware component
- Can process operations:
- See fig. 2.2
- Contains:
 - Control unit (CU):
 - controls the retrieval of instructions from primary memory
 - sequence of their execution
 - provides the ALU with the data and instructions
 - has various registers (size of register is known as word)
 - Arithmetic logic unit (ALU)
 - Performs arithmetic, logical, I/O operations
 - Memory address register (MAR)
 - Holds addresses to be sent to RAM
 - Memory data register (MDR)
 - Holds data to be sent to RAM or coming from RAM.

Terms

- Bit, byte
- Kilo, mega, giga, tera
- Kb, KB

64-bit computer:

- CPU processes 64-bits
- Memory location has 64 bits
- Bus width is 64 bit

Primary memory consists of

- ROM
 - Not volatile
 - Used
 - to boot the computer
 - to operate the computer
- RAM
 - Volatile
 - Can be overwritten
 - Stores the executing programs
 - Stores the data needed by executing programs
- Cache
 - Volatile
 - Can be overwritten

Each memory location has an address and content.

Buses

	put 75 in address 648302	send the value found in address 827297
Data bus	75	any value
Memory (address) bus	648302	827297
Control bus	write	read

2.1.2 RAM and ROM

Primary memory:

- directly accessible by the CPU
- hold programs (and their data) that are currently being executed.

2.1.3 The cache memory

Two types of RAM:

- Dynamic RAM (DRAM)
 - Slower
 - Cheaper
- Static RAM (SRAM)
 - Faster
 - More expensive
 - Used as cache (holds data and instructions that are frequently used by the CPU, see fig. 2.3)

Cache is **graded** as Level 1 (L1), Level 2 (L2) and Level 3 (L3):

- **L1** is usually part of the CPU chip itself and is both the smallest and the fastest to access. Its size is often restricted to between 8 KB and 64 KB.
- **L2** and **L3** caches are bigger than **L1**. They are extra caches built between the CPU and the RAM. Sometimes L2 is built into the CPU with L1. L2 and L3 caches take slightly longer to access than L1. The more L2 and L3 memory available, the faster a computer can run.

2.1.4 The machine instruction cycle

Fetch-decode-execute cycle (machine instruction cycle)

1. Fetch instruction from primary memory to control unit
 - CPU instructs memory to send it the next instruction.
 - It does this by sending the address of the next instruction.
 - The information from the CPU to RAM and back again is carried by buses.
2. Decode instruction in control unit
 - The CU can discover that it needs data found in RAM to work out the instruction.
 - The CU informs the ALU what calculations to perform.
3. Execute instruction
 - The ALU executes the instruction
 - It may do it in more than one cycle, and it may still require to fetch more data from memory.
4. Store result of execution and check for next instruction
 - The result of the execution is stored in RAM

- Repeats the 4 steps

Secondary Memory

2.1.5 Persistent storage and secondary memory

Secondary memory:

- Compared with primary memory it is slow and large (high capacity).
- Also known as:
 - Secondary storage
 - Storage
 - Auxiliary storage
 - Persistent storage
- Non-volatile
- Data can be deleted; new data can be saved.
- See fig 2.4: data and programs required by the CPU must be copied from secondary storage to primary memory.
- Can be used as virtual memory i.e. if the primary memory is full and the system needs to use a program that is on secondary storage, then a part of the primary memory is cleared (its contents is copied on secondary storage) to make space of the required program (or part of it).
- List of storage devices:
 - HDD – 4TB, €73
 - CD-RW – 25 disks, 700MB, €7.50
 - DVD-RW – 50 disks, 4.7GB, €22
 - USB flash drive (pen drive, USB stick) – 128 GB, €29
 - SSD – 1 TB, €130
 - SD (Secure Digital) card – 128 GB, €14
 - CD (Compact Flash) card – 64 GB, €32
 - Zip disk (discontinued)
 - Magnetic tape

Operating and application systems

2.1.6 Functions of an operating system

Operating system (OS):

- Manager of computer
- A set of programs

See fig. 2.5: OS serves as an intermediary between application programs and hardware.

The main services of an OS:

- Peripheral communication
 - Peripheral devices reside outside the CPU.
 - OS communicates directly with the hardware.
 - It serves as interface between applications and hardware.
- Memory management
 - OS directs where programs are placed in memory.
 - It keeps a table of the occupied and free spaces in memory.
- Resource monitoring and multitasking
 - OS is responsible of assigning resources e.g., memory, CPU time, etc.
 - Sharing the CPU is called multitasking or timesharing.
- Networking
 - A PC OS has the means to connect to and be part of a network.
 - A network OS manages the interaction of PC on the network and how the sharing of resources can be done correctly.
- Disk access and data management
 - The OS also manages the files on secondary storage. It has a record of where all files are together with the parts that are empty.
- Security
 - Methods of authentication
 - Username and password.
 - Magnetic cards
 - Biometric data
 - Log files to keep trace of the activity of all users.

Examples of operating systems:

- OS X
- Linux
- UNIX
- Microsoft Windows

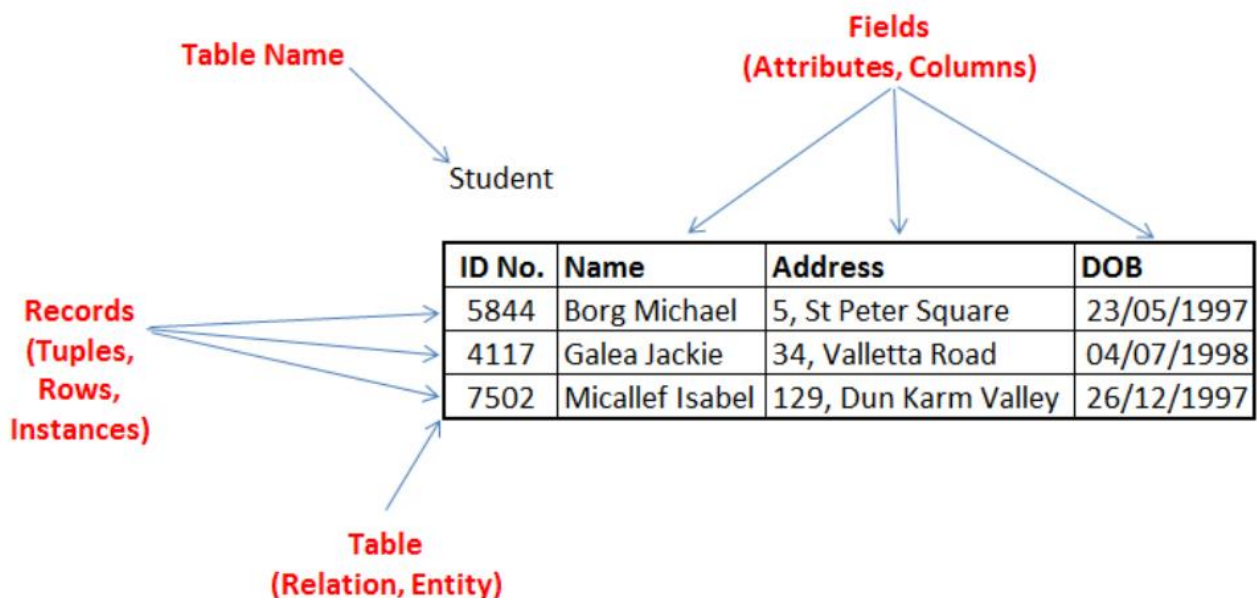
2.1.7 Software application

Word processors

Spreadsheet

Database management system (DBMS)

- Database: organized collection of related data.
- Record (tuple, row): a sequence of data about an item e.g. a person might be defined with an ID number, a name, an address, a mobile number etc.
- Fields (columns): one value of a record
- Primary field: in the table Student (below) it is ID No. It is unique.



Web browser

- URI (Uniform Resource Identifier) – the address of a web page e.g., <http://en.wikipedia.org/>

Email

- Short for electronic mail.
- An email has three components:
 - Envelope: communication required for the email to be sent.
 - Header: contains info about the sender, receiver date/time stamp etc.
 - Body: contains the message.
- The protocol SMTP (Simple Mail Transfer Protocol) is used to transfer the email from sender to receiver.

- Emails can be:
 - On your computer e.g., Microsoft Outlook
 - On the internet (webmail) e.g., Google Gmail

Computer Aided Design (CAD)

Graphic presentation software

- Also called graphics software or image editing software.
- Operations: cropping, erasing etc.

2.1.8 Common features of applications

- GUI:
 - Icons, toolbars, menus, dialogue boxes etc.
 - Also called WIMP (Windows, Icons, Menus and Pointers)
 - Compare with command line interface (CLI).
 - See table 2.1 Pros and Cons of CLIs and GUIs
 - Some GUI elements are provided by the operating system

Computer organization – Binary representation

2.1.9 Bit, byte, binary, decimal and hexadecimal

Bit and byte

Byte = 8 bits

Decimal number system

	10^2	10^1	10^0		10^{-1}	10^{-2}
multipliers	100	10	1		0.1	0.01
digits	4	3	9	.	6	7

Binary number system

	2^2	2^1	2^0		2^{-1}	2^{-2}
multipliers	4	2	1		0.5	0.25
digits	1	1	0	.	0	1

Conversions

- from binary to decimal
- from decimal to binary (2 ways, repeated division and putting ones according to the weights)

Binary number system and negative numbers

Numbers can be signed and unsigned. *Signed* numbers can be negative or positive. *Unsigned* are always positive.

Two's complement is the way most modern computers use to represent signed binary numbers.

MSB = most significant bit; LSB = least significant bit

To find the two's complement of a number:

Method 1

1. find ones complement (reverse the bits)
2. add 1

Method 2

1. start from the right and (moving left) copy all zeros
2. copy the first one you meet
3. reverse all the other bits

So if 52 = 00110100 (in one byte), then -52 = 11001100

To evaluate a number from two's complement to decimal, add the weights as shown

1	1	0	0	1	1	0	0
-128	64	32	16	8	4	2	1

$$=-128 + 64 + 8 + 4 = -52$$

Binary fractions

Unsigned numbers – fixed point representation

1	0	0	1	•	1	1	0	1
8	4	2	1		1/2	1/4	1/8	1/16
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}

$$1010.1101_2 = 8 + 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = 9 \text{ and } \frac{13}{16}_{10}$$

Signed numbers

1	0	1	0	•	1	1	0	1
-8	4	2	1		1/2	1/4	1/8	1/16
-2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}

$$1010.1101_2 = -8 + 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = -6 \text{ and } \frac{3}{16}_{10}$$

Hexadecimal number system

Denary base 10

Binary base 2

Hexadecimal base 16 (see table 2.5 on pg 93)

Symbols: 1, 2, 3, ..., 9, A, B, ... F

Adding binary and hexadecimal numbers.

2.1.10 Data representation

If one has n bits, then the number of different permutations is 2^n e.g.:

- With one bit you can have 2 permutations i.e. 0 or 1
- With two bits you can have 4 permutations i.e. 00, 01, 10, and 11. Etc.

Take for example 8 bits

- For unsigned integers 00000000 (0) is the least number and 11111111 (255) is the largest number. Range is 0 to 255 (in all there are 256 permutations).
- For signed integers (2's complement) the smallest number is 10000000 (-128) and the biggest number is 01111111 (127). Range is from -128 to 127.
- For sign and magnitude numbers (signed) the smallest number is 11111111 (-127) and the biggest number is 01111111 (127)

Characters

ASCII

- 7 bits
- To represent English alphabet
- E.g. K is 75, k is 107
- See table 2.7

Unicode

- Can represent all the characters of all the languages of the world.

There are other character representations (e.g. EBCDIC)

Strings

Another code for characters is UTF-8 (Unicode Transformation Format). It uses 8 bits per character. Others are UTF-16 and UTF-32. Unicode itself stands for Universal Code.

Colours

Bit depth: The number of bits used to represent each pixel.

One representation is the RGB representation e.g. 70EF5A₁₆ (70₁₆ is the amount of red; EF amount of green and 5A amount of blue) – see fig. 2.19.

Simple logic gates

2.1.11 Boolean operators

Boolean operators: AND, OR, NOT, NAND, NOR, XOR.

Truth tables.

Venn diagrams: to represent Boolean operators. Figures 2.21, 2.23, 2.25, 2.26, 2.27, 2.28, 2.29

$$A \text{ AND } B = A \cdot B = AB$$

$$A \text{ OR } B = A + B$$

$$\text{NOT } A = \bar{A} = A'$$

$$A \text{ NAND } B = \overline{A \cdot B}$$

$$A \text{ NOR } B = \overline{A + B}$$

$$A \text{ XOR } B = A \oplus B = A\bar{B} + \bar{A}B$$

See fig. 2.15 – logic gates and more symbols for the operators

2.1.12 Truth tables using Boolean operators

Truth tables with three inputs (in this course we will not have truth tables with more than three inputs).

Truth tables that are devised from a real situation e.g. “Tania won’t go to play if it is cold and raining, or if she has not done her homework.”

Let the variables be:

- cold:

- cold = 0 implies that it is not cold
- cold = 1 implies that it is cold
- we can take the above to mean cold=false and cold=true
- rain
 - rain = 0 implies it is not raining
 - rain = 1 implies that it is raining
- homework
 - homework = 0 implies no homework (or homework is done)
 - homework = 1 implies homework is not done
- stay_home
 - stay_home = 0 implies that she can go out and play
 - stay_home = 1 implies that she cannot go out and play

The equation representing the problem is the following:

$$\text{stay_home} = \text{cold} \cdot \text{rain} + \overline{\text{homework}}$$

INPUT			Intermediary results		OUTPUT
cold	rain	homework	cold.rain	$\overline{\text{homework}}$	stay_home
0	0	0	0	1	1
0	0	1	0	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	0	1

See other examples on pgs. 110 to 116.

Note that there are other symbols for NOT, AND, OR etc. They are shown in the table below:

NOT	\neg
AND	\wedge
OR	\vee
XOR	$\underline{\vee}$
NAND	$\overline{\wedge}$
NOR	$\overline{\vee}$

(Additional Section) Simplification and Boolean expressions

See table 2.21

See examples on pgs. 117-118

2.1.13 Logic and Venn diagrams using logic gates

Logic gates

Logic diagram (logic circuit)

See exercise on logic diagrams and Venn diagrams.