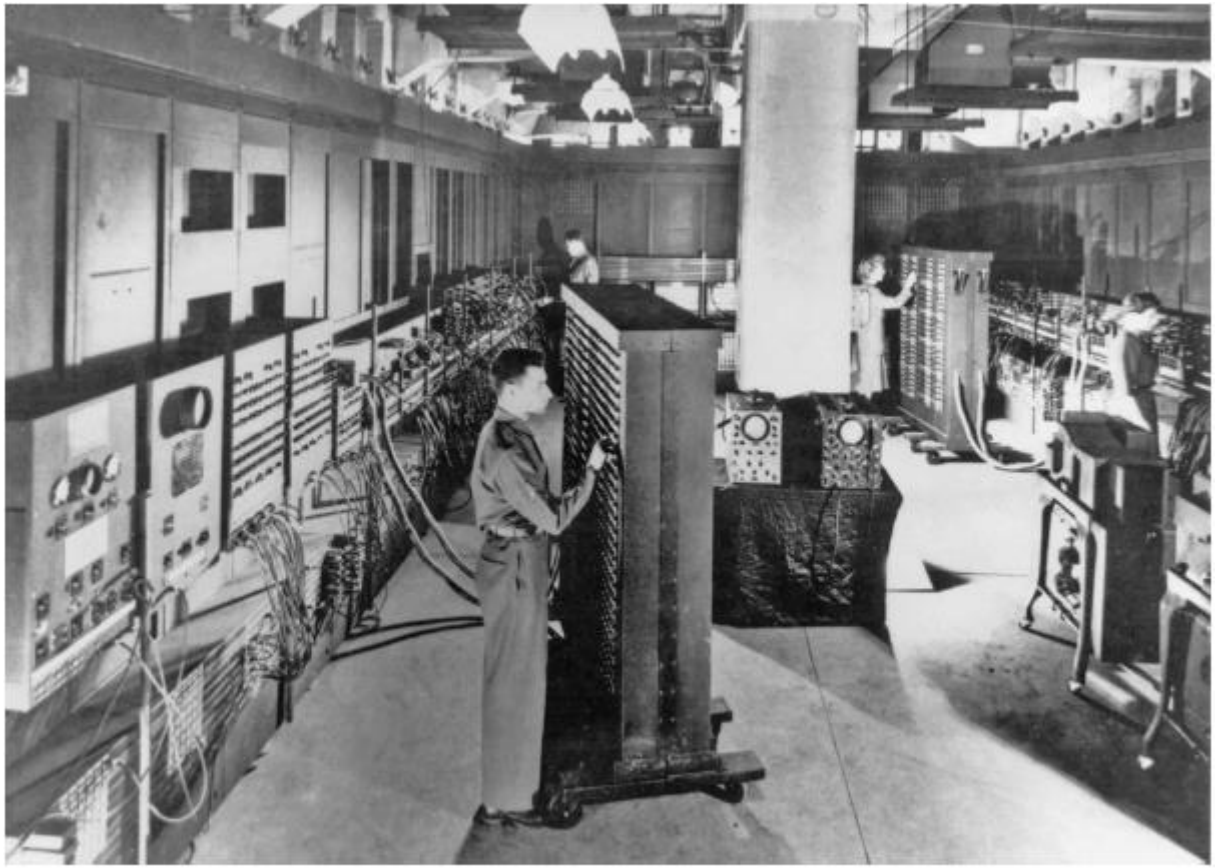
Programming environments

The first computers



The ENIAC

Binary numbers

- Computers stores information as number
  - Not as decimal numbers
    - 10 digits, (0 – 9)
    - Uses base 10

```
10^2|10^1|10^0
  1 |  9 |  3
```

the number 193 is really {(1*10^2)+(9*10^1)+(3*10^0)}.

  - But as binary numbers
    - 2 digits, (0-1)
    - Uses base 2
    - A single binary digit is called a bit

```
2^2|2^1|2^0
```

Binary Numbers -Permutations

1 bit 2 items	2 bits 4 items	3 bits 8 items	4 bits 16 items	5 bits 32 items	
0	00	000	0000	00000	10000
1	01	001	0001	00001	10001
	10	010	0010	00010	10010
	11	011	0011	00011	10011
		100	0100	00100	10100
		101	0101	00101	10101
		110	0110	00110	10110
		111	0111	00111	10111
			1000	01000	11000
			1001	01001	11001
			1010	01010	11010
			1011	01011	11011
			1100	01100	11100
			1101	01101	11101
			1110	01110	11110
			1111	01111	11111

Number of bit <> different values

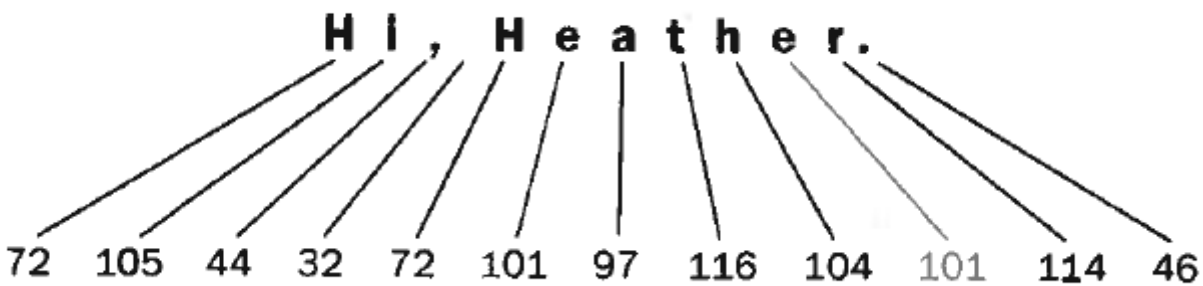
- How many bits are needed to represent 256 characters.

Key Concept

There are exactly 2^N permutations of N bits. Therefore, N bits can represent up to 2^N unique items.

Storing text in digital format

- Storing the sentence: “ Hi, Heather.”
- 12 numbers
- **Different number for ‘H’ and ‘h’**
- Space is also represented as a number (32)



Character tables

- The Original Character table (8-bit) ASCII table

  FX:

  Æ = 146 (Obs! rækkefølge ø-å-æ)

- 16-bit Table in C# UniCode

  FX:

  Æ = 198 (obs! rækkefølge å-æ-ø)

Basic computer architecture

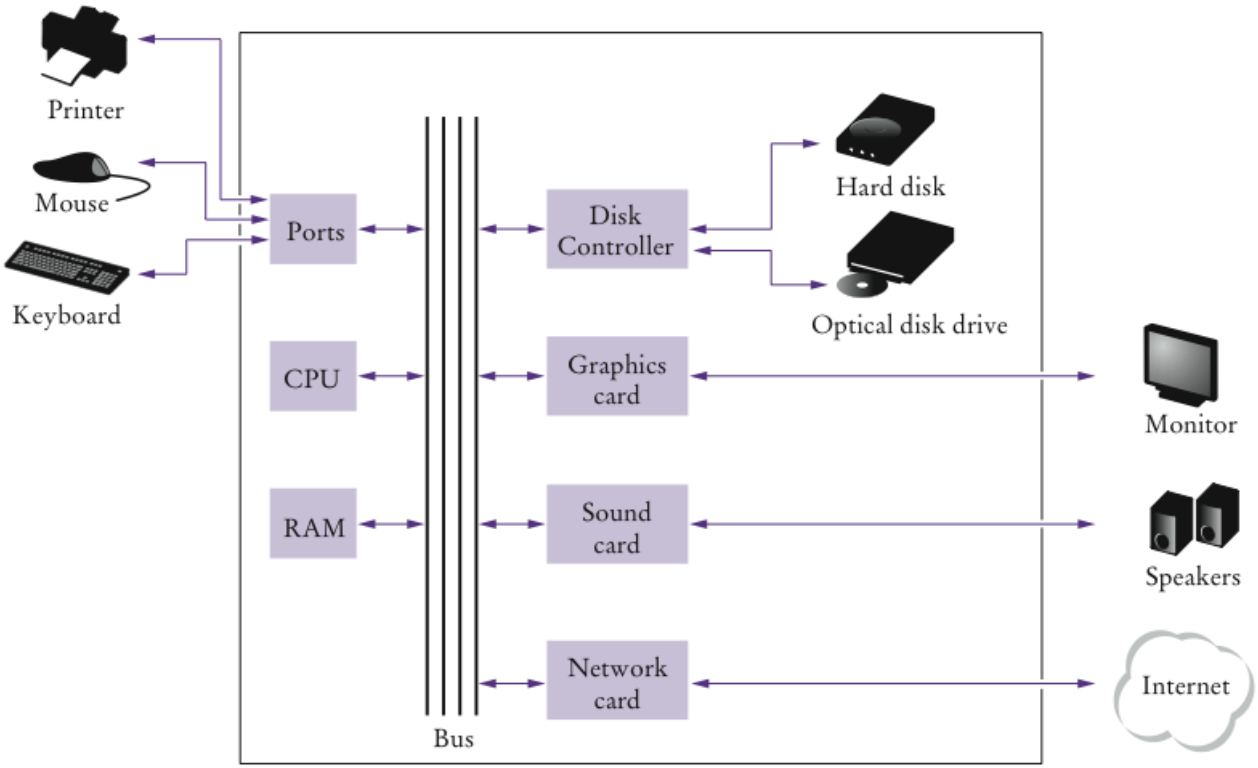


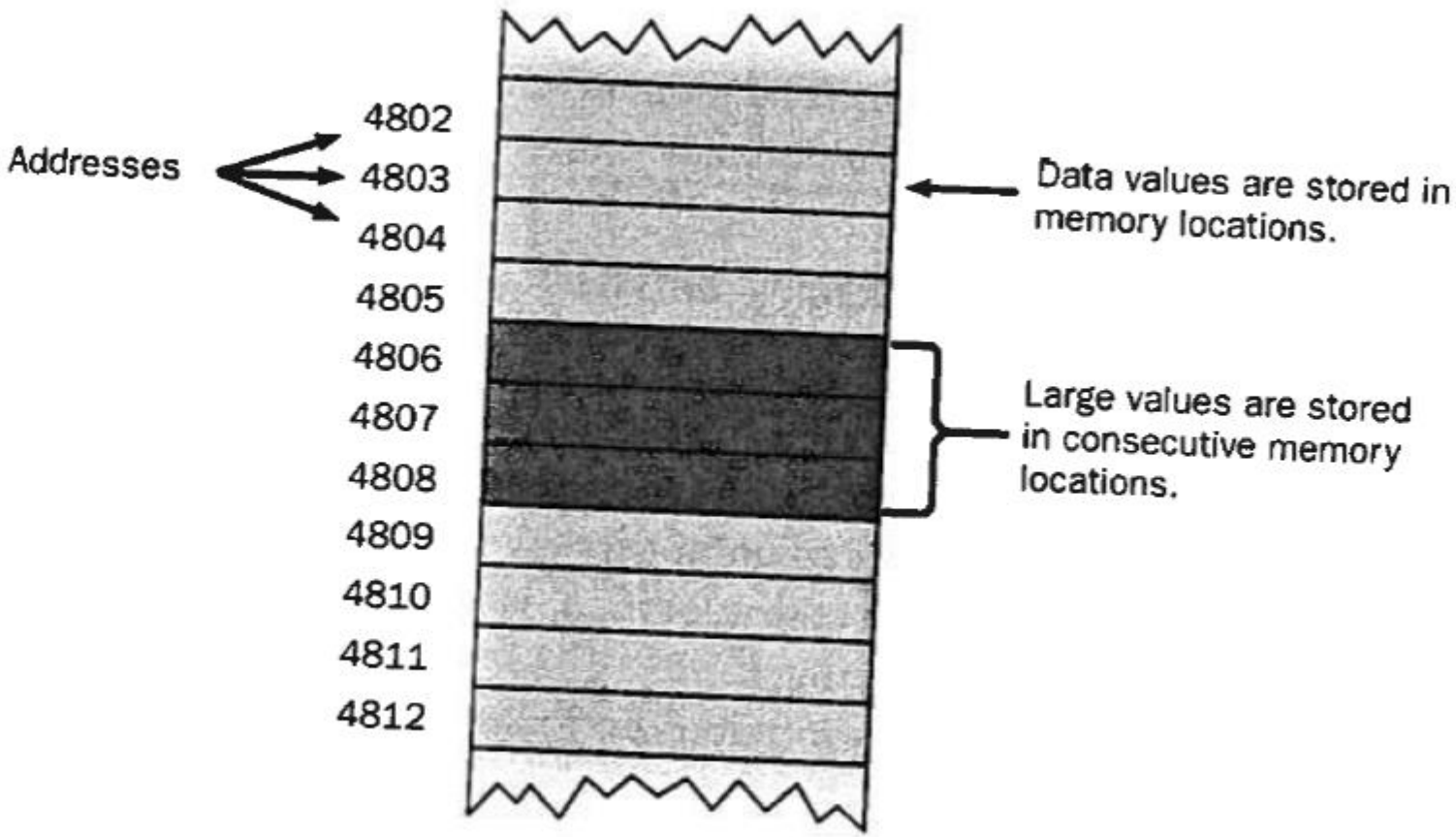
Figure 5 Schematic Diagram of a Computer

Main memory(RAM) and secondary memory

Is made up of a series of consecutive memory locations

- Each memory location has a number called an address
- When data is stored in a memory location it overwrites any information that was previously stored at that location
- In many computers memory location consists of 8 bits called a **byte**. You can save large pieces of information in many consecutive bytes.
- Main memory is volatile
  Meaning - Data is lost if the power is turned of

Memory locations



Units of binary storage

Unit	Symbol	Number of Bytes
byte		$2^0 = 1$
kilobyte	KB	$2^{10} = 1024$
megabyte	MB	$2^{20} = 1{,}048{,}576$
gigabyte	GB	$2^{30} = 1{,}073{,}741{,}824$
terabyte	TB	$2^{40} = 1{,}099{,}511{,}627{,}776$

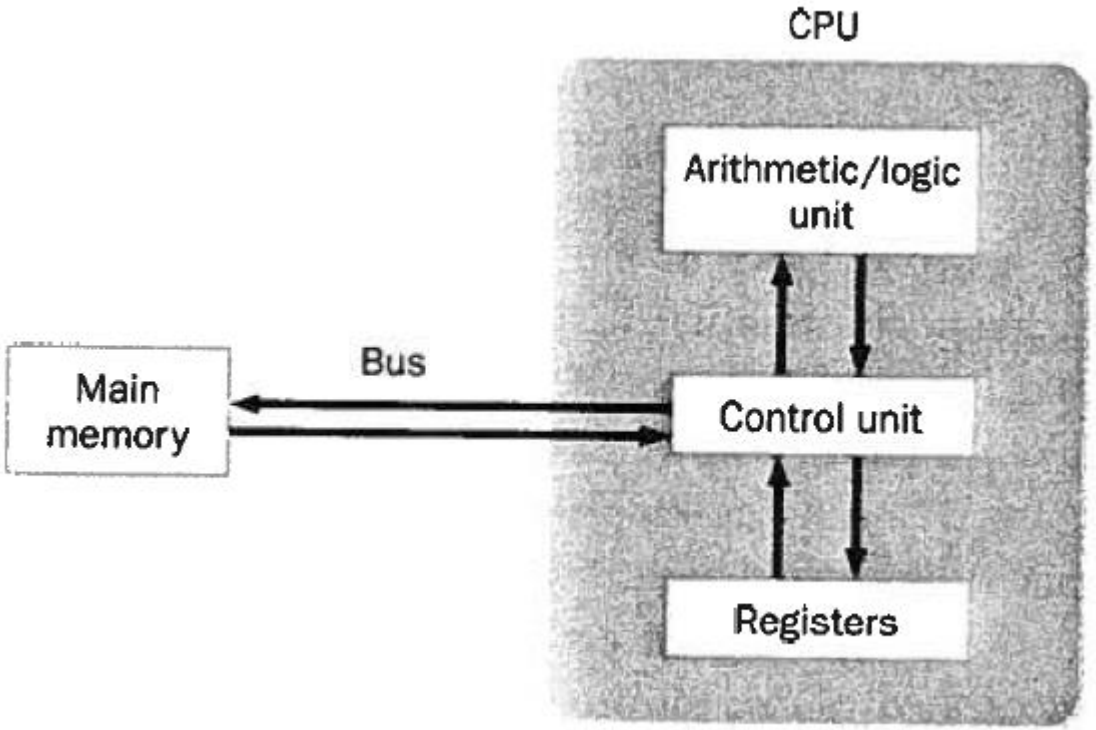
RAM & ROM

- RAM
  - Random Access Memory (Read/write memory)
  - Also called **main memory**.
    Data are lost when the power is turned of
- ROM
  - Read Only Memory
  - ROM chips are built into main circuit board of a computer.

Central processing unit

- The CPU
  - interacts with the main memory to perform the processing in a computer
  - Interprets and executes instructions

Von Neuman Architecture



Parts of the CPU

- Control unit
  - Coordination the processing

- Registers
  - Temporary memory
  - Registers for special purposes
    - Program counter (holds address of the next instruction
    - Instruction register holds the current instruction

- Arithmetic/logic unit (ALU)
  - Performs ***all*** calculations/decisions

- System clock
  - Synchronizes events of the CPU

Programming language levels

- Historical development of computer languages
  - **Machine language**
    - Each instruction accomplishes only a simple operation
    - Expressed in binary digits
  - **Assembly language**
    - Each instruction accomplishes only a simple operation
    - Use mnemonics
  - **High-level language**
    - Expressed in simple English-like phrases
    - E.g. C#, Java, C++……
  - Fourth-generation languages
- To run a program on a computer it must be expressed in the specific computer's Machine language.
- Each CPU has its own Machine language

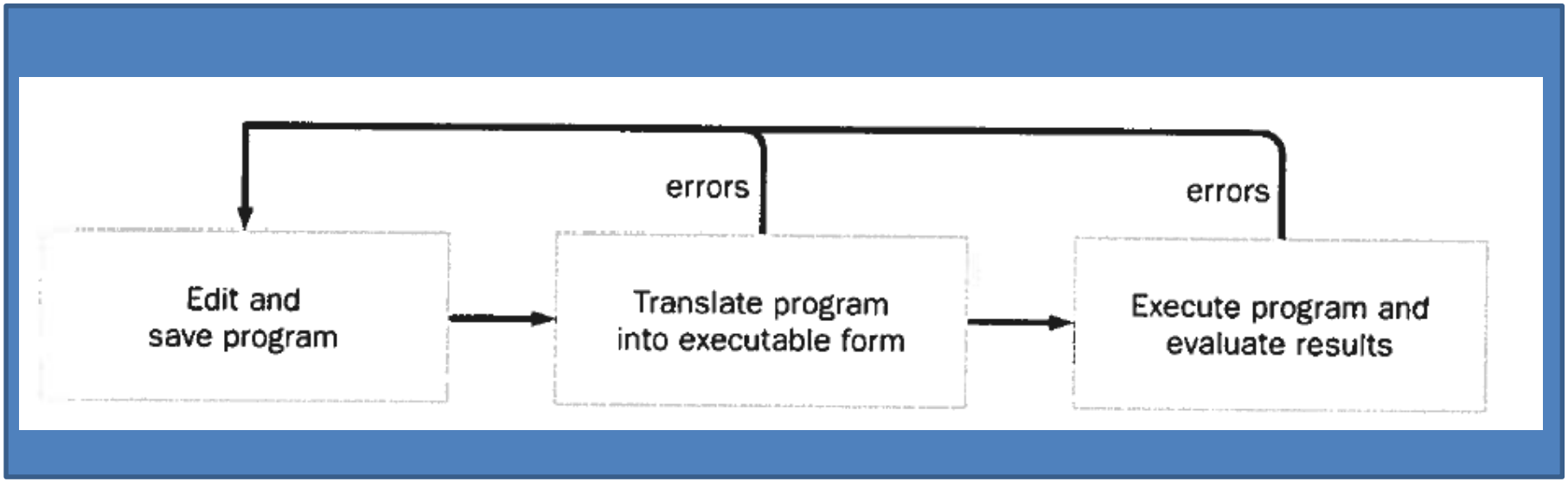
Languages

High-Level Language	Assembly Language	Machine Language
a + b	ld [%fp-20], %o0	...
	ld [%fp-24], %o1	1101 0000 0000 0111
	add %o0, %o1, %o0	1011 1111 1110 1000
		1101 0010 0000 0111
		1011 1111 1110 1000
		1001 0000 0000 0000
		...

Editor, compiler, interpreter

- Editor
  - Can be used to type in a program and store in a file
- Compiler
  - Translates code in one language into another language
  - Original code is called: Source code
  - High level languages can be translated directly to machine language
- Interpreter
  - Translates a statement and executes it one by one.

Editing and running a program



.NET execution

- C# combines the compiler and an interpreter
- C# is compiled to Intermediate language (IL)
  similar to a machine language code
- C# interpreter reads C# IL and executes it on a specific machine.
- C# IL is not tied to any particular processor type.
  - This makes C# architecture neutral and platform independent.
  - Easy portable from one machine to another
  - There must be an IL interpreter for each processor

.NET Program Execution