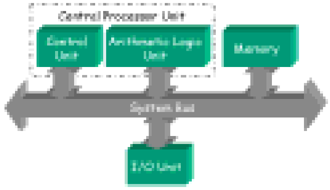
# Basic Computer Architecture



# What is a Computer?

**A Computer is a machine capable of performing [very] simple operations on a [very] large amount of data at [very] high speeds <u>according to a given set of instructions</u>.**

# Characteristics of a Computer?

- <u>Speed</u>: Millions of operations per second
  (MIPS) millions of instructions per second
- Only performs one operation at a time
- Very limited set of simple instructions
  (eg: Load, Store, Add, Compare etc. )
- Deterministic: RIRO [Rubbish In Rubbish OUT]
- Reliable
- Can operate on and store very large data quantities

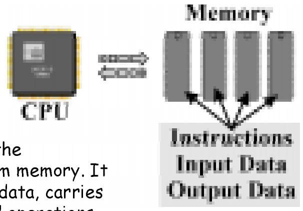
# Important Computer Components

- CPU: Central Processing Unit (MC68000/68332)
- Memory: Data Storage Area
- Peripherals: Disk, Monitor, Keyboard, Mouse, etc...
- Bus: Wires connecting everything together

# Central Processing Unit (CPU)

- The CPU must be supplied with:
  - Input Data to opearte on
  - A set of instructions detailing exactly what to do
- The CPU generates:
  - Output Data
- Input Data, Instructions and Output Data all reside in memory

# READ -> EXECUTE -> WRITE

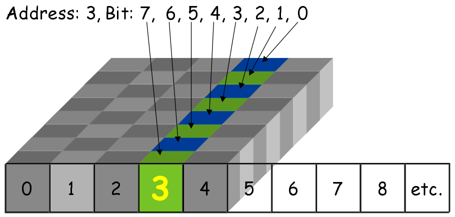


The CPU reads the instructions from memory. It reads the input data, carries out the required operations and stores the result back in memory.

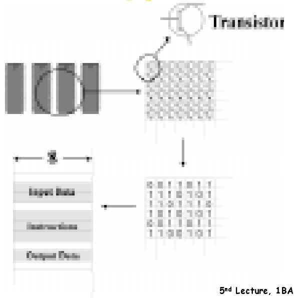
## What is memory?

- Memory is composed of millions of switches
  (implemented in current memory chips as transistors)
- Each switch represents a binary value:
  - ON or OFF
  - 1 or 0
  - True or False
  - < 0.4 Volts or > 2.4 Volts

## Memory - Address - Bit



## The Structure of Memory



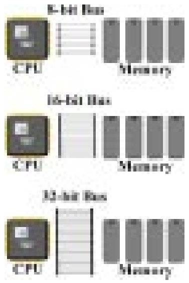
## The Unit of Memory

- The fundamental unit of memory is the BIT (Binary Digit)
- Each bit can take on the value Logic-1 or Logic-0
- Accessing these bits individually is not of great use
- Each value can only be a Logic-1 or Logic-0 (not very useful for storing large numbers!)

## Group BITs Together

- View memory as a list of larger elements:
  - 4 bits = 1 NYBBLE
  - 8 bits = 1 BYTE
  - 16 bits = 1 WORD
  - 32 bits = 1 LONGWORD
- When reading data/writing data to/from memory (using the MC68332), we can do so in units of bytes, words or longwords only

## Bus



The number of bits that can be read at any given time is determent by the **BUS SIZE**.

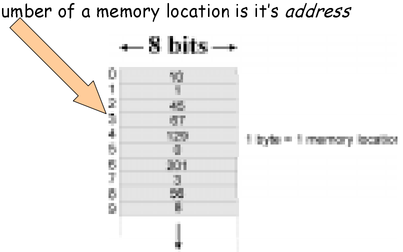
BUS SIZE = Number of wires (data) connecting the CPU to the Memory

## Memory Capacity

- Memory size is expressed in terms of bytes:
  - 1024 Bytes = 1 KILOBYTE (Kb.)
  - 1024 Kilobytes = 1 MEGABYTE (Mb.)
  - 1024 Megabyte = 1 GIGABYTE (Gb.)
- Individual bytes in memory (memory locations) are numbered sequentially.

## Memory Location

The number of a memory location is it's *address*



E.g.: The contents of memory location 3 is 67.

## Bit Group Encoding

Bit 3
Bit 2
Bit 1
Bit 0

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | = | 0 | 1 | 0 | 0 | 0 | = | 8 |
| 0 | 0 | 0 | 1 | = | 1 | 1 | 0 | 0 | 1 | = | 9 |
| 0 | 0 | 1 | 0 | = | 2 | 1 | 0 | 1 | 0 | = | 10 |
| 0 | 0 | 1 | 1 | = | 3 | 1 | 0 | 1 | 1 | = | 11 |
| 0 | 1 | 0 | 0 | = | 4 | 1 | 1 | 0 | 0 | = | 12 |
| 0 | 1 | 0 | 1 | = | 5 | 1 | 1 | 0 | 1 | = | 13 |
| 0 | 1 | 1 | 0 | = | 6 | 1 | 1 | 1 | 0 | = | 14 |
| 0 | 1 | 1 | 1 | = | 7 | 1 | 1 | 1 | 1 | = | 15 |

## Types of Memory:

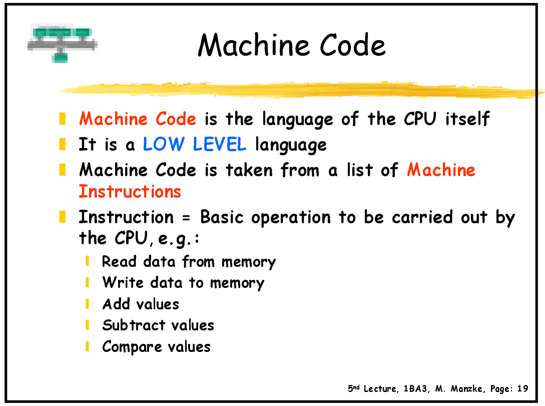
- Read Write Memory [RWM]: CPU allowed to read and write contents of memory. Commonly known as Random Access Memory [RAM], 2 flavours:
  - Volatile: contents lost when power is switched off.
  - Non-Volatile: memory has its own separate power supply (e.g.: a battery).
- Read Only Memory [ROM]: memory contents are physically etched onto the chip during manufacture. Contents can only be read, and are not lost on power-off.

## More Memory Types

- Programmable ROM [PROM]: can write contents into it ONCE only (using a device called a PROM Programmer). Henceforth, it acts like ROM.
- Erasable PROM [EPROM]: a PROM chip that may be written a number of times using an EPROM programmer
- Electrically Erasable PROM [EEPROM]
- The robot has:
  - 64Kb RAM
  - 64Kb EPROM

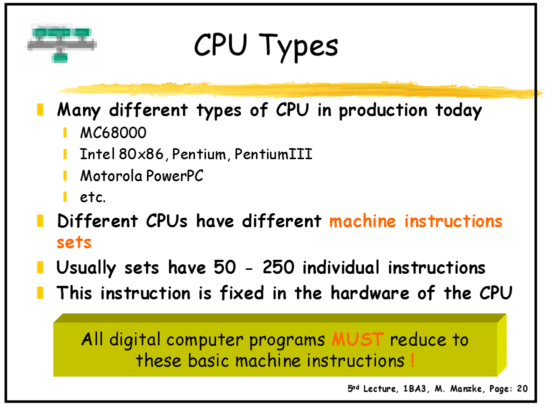
## Programming

- **A program is a list of instructions to the computer**
- **Programs are distinguished by the language in which they are written**
  - (e.g. C, PASCAL, MODULA-2, FORTRAN, LISP, C++, EIFFEL, PROLOG, BASIC, JAVA etc.)
- **In 1BA3, we are concerned with:**
  - Machine Code
  - Assembly Language



# Machine Code

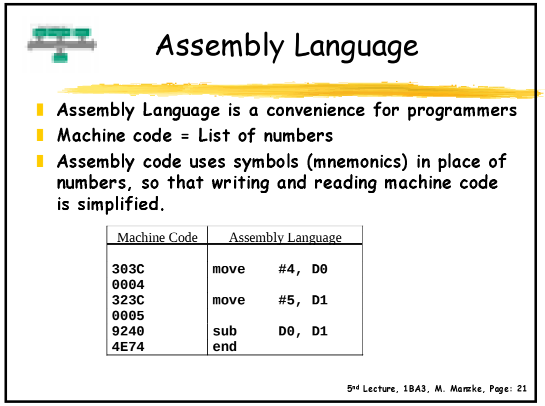
- Machine Code is the language of the CPU itself
- It is a LOW LEVEL language
- Machine Code is taken from a list of Machine Instructions
- Instruction = Basic operation to be carried out by the CPU, e.g.:
  - Read data from memory
  - Write data to memory
  - Add values
  - Subtract values
  - Compare values



# CPU Types

- Many different types of CPU in production today
  - MC68000
  - Intel 80x86, Pentium, PentiumIII
  - Motorola PowerPC
  - etc.
- Different CPUs have different machine instructions sets
- Usually sets have 50 - 250 individual instructions
- This instruction is fixed in the hardware of the CPU

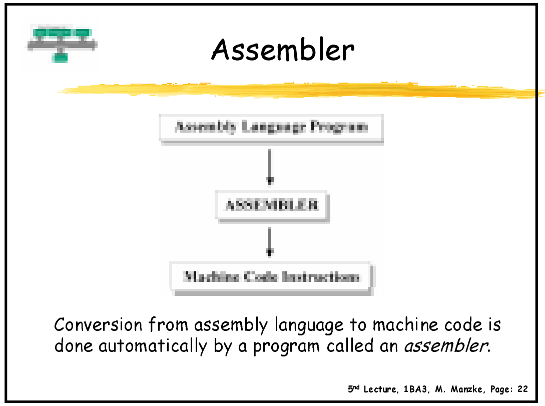
All digital computer programs MUST reduce to these basic machine instructions !



# Assembly Language

- Assembly Language is a convenience for programmers
- Machine code = List of numbers
- Assembly code uses symbols (mnemonics) in place of numbers, so that writing and reading machine code is simplified.

| Machine Code | Assembly Language |
|---|---|
| 303C | move #4, D0 |
| 0004 | |
| 323C | move #5, D1 |
| 0005 | |
| 9240 | sub D0, D1 |
| 4E74 | end |



# Assembler

Assembly Language Program

ASSEMBLER

Machine Code Instructions

Conversion from assembly language to machine code is done automatically by a program called an *assembler*.