Introduction to Data Structures

Course Code: CS102
Winter 2021-22

Introduction to Data Structures

- Books

1. Data Structures & Algorithms in JAVA, by Robert Lafore, Pearson
2. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, Universities Press

Course Objectives

- To learn various ways how our program data can be stored in memory so that it can be efficiently accessed.
- To familiarizes the basic data structures (DS) such as arrays, linked lists, stacks, queues, heaps, binary trees, and graphs.
- To learn various DS operations such as insertion, deletion, searching, and sorting.
- Also, we learn how to design and analyse an algorithm. We use Big-O notation for analysis purpose.

Tentative Evaluation Scheme - Theory

	Type of Examination	Weightage (%)
1	Pre-mid semester Quiz	10%
2	Mid-semester examination	20% (10+10)
3	Pre-End semester Quiz	15%
4	End-semester examination	30% (15+15)
5	Class room participation and Assignment (s)	25%

Computer system basics

- A digital hardware: a lot of switches integrated
- A digital switch: the electronic device react to presence or absence of voltage
- Symbolically we represent
  - Presence of voltage as “1”
  - Absence of voltage as “0”

Computer system basics cont.

- An electronic device can represent uniquely only one of two things
  - Each “0” or “1” is referred to as a Binary Digit or Bit
  - Bit: Fundament unit of information storage
- To represent more things we need more bits
  - E.g., 2 bits can represent four unique things: 00, 01, 10,11
  - k bits can distinguish 2^k distinct items
- Combination binary bits together can represent some info. or data. E.g., 01000001 can be
  1. Decimal value 65
  2. Alphabet (or character) ‘**A**’ in ASCII notation
  3. Command to be performed, e.g., performing **Add** opration

We can divide a computer into three broad parts or subsystems:

1. Central Processing Unit (CPU),
2. main memory and
3. input/output subsystem.

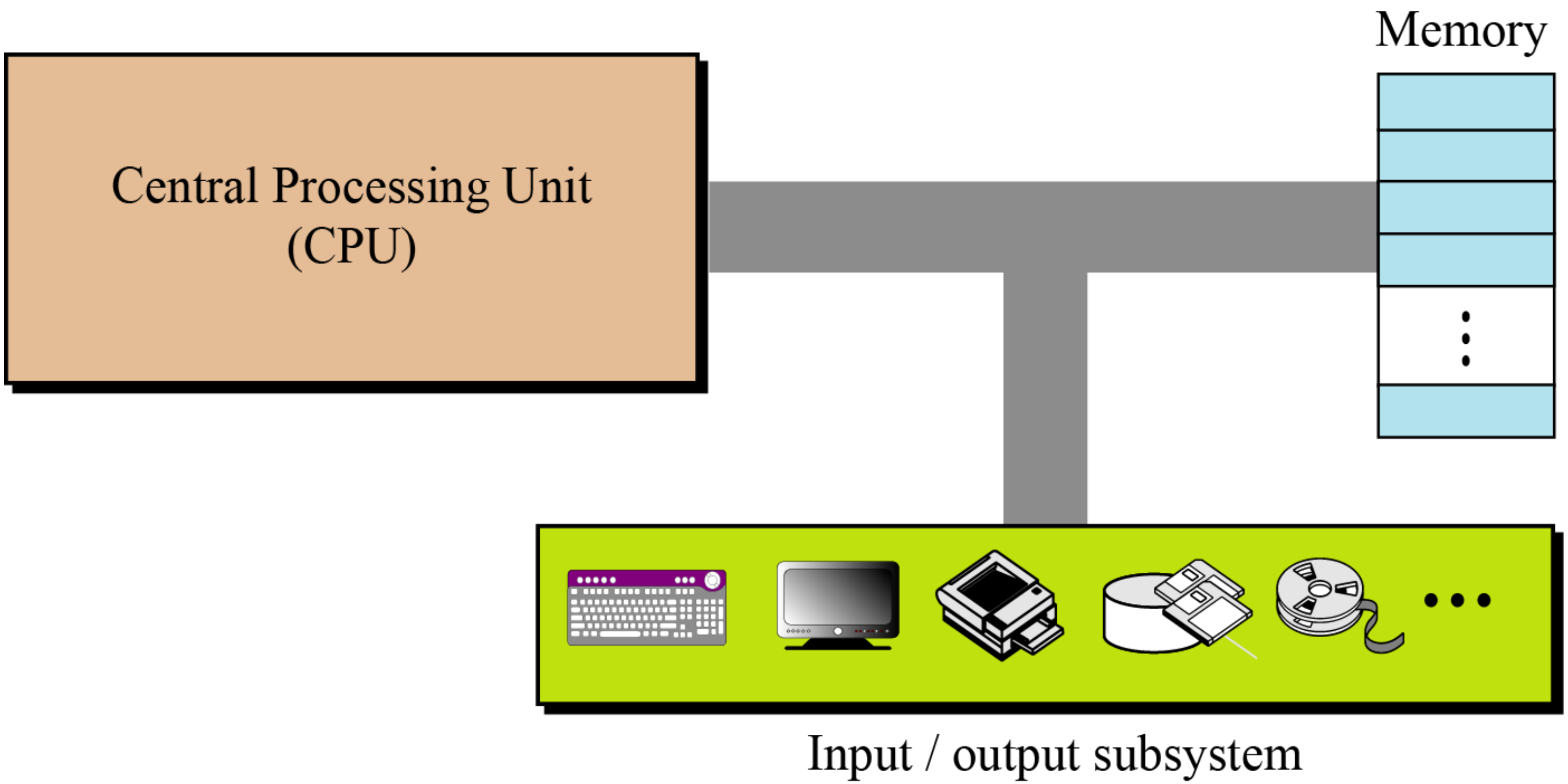


Figure: Computer hardware (subsystems)

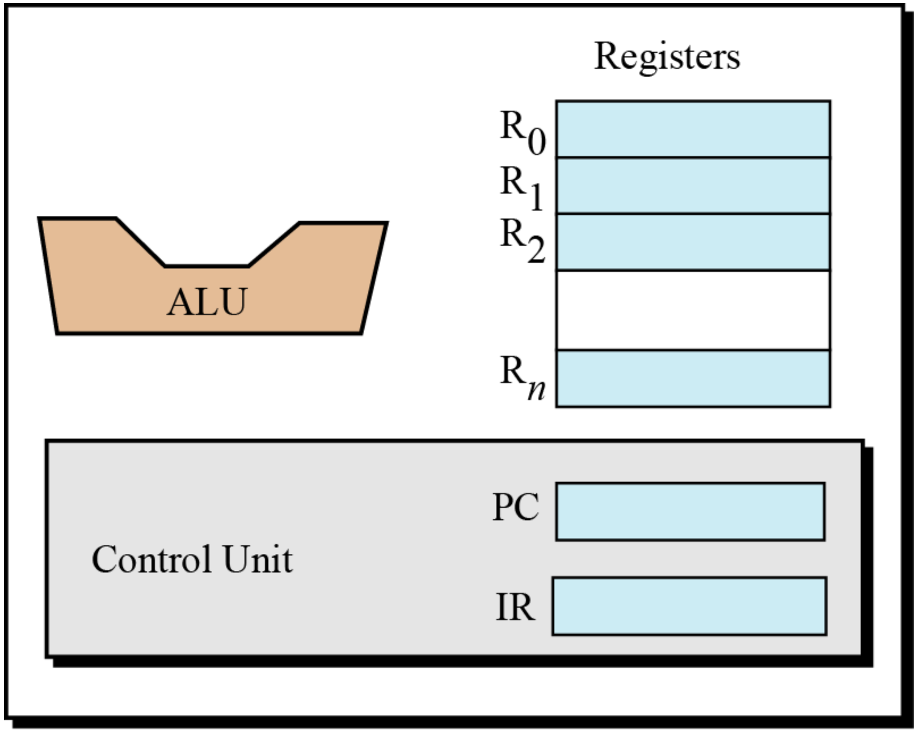
Central Processing Unit (CPU)

The CPU performs operations on data. In most architectures it has three parts:

1. an arithmetic logic unit (ALU),

2. a control unit and

3. a set of registers, fast storage



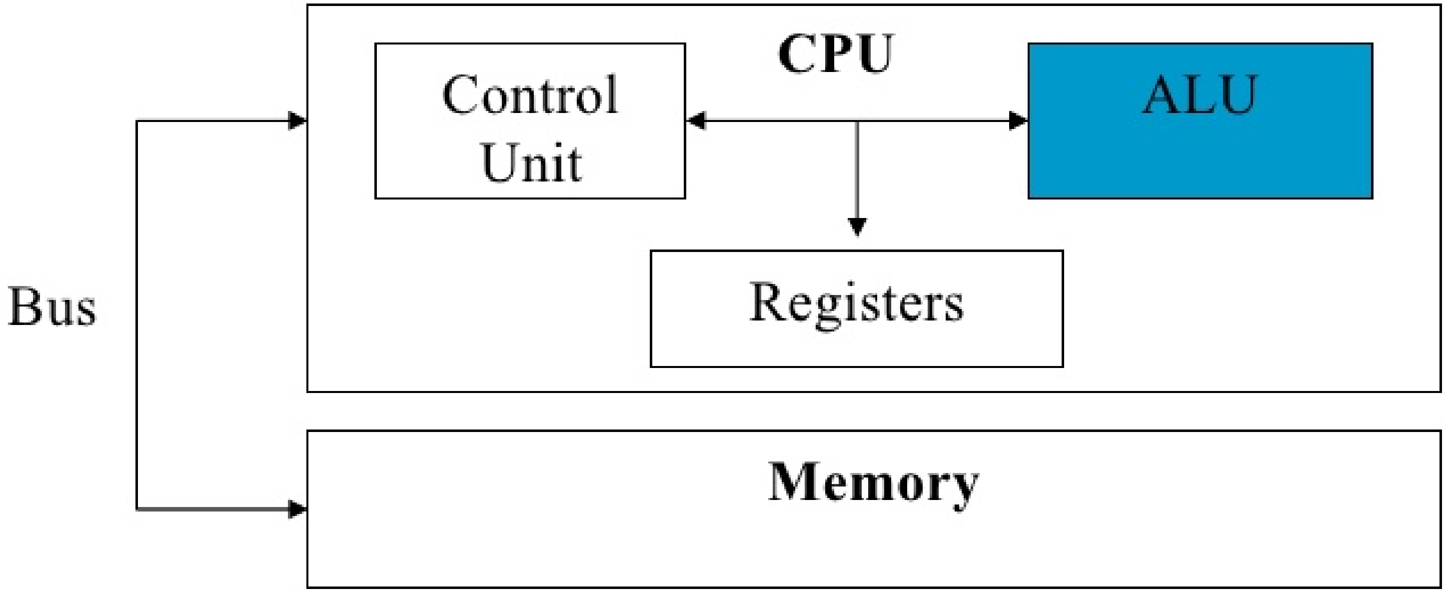
Central Processing Unit (CPU)

PC: Program counter

IR: Instruction register

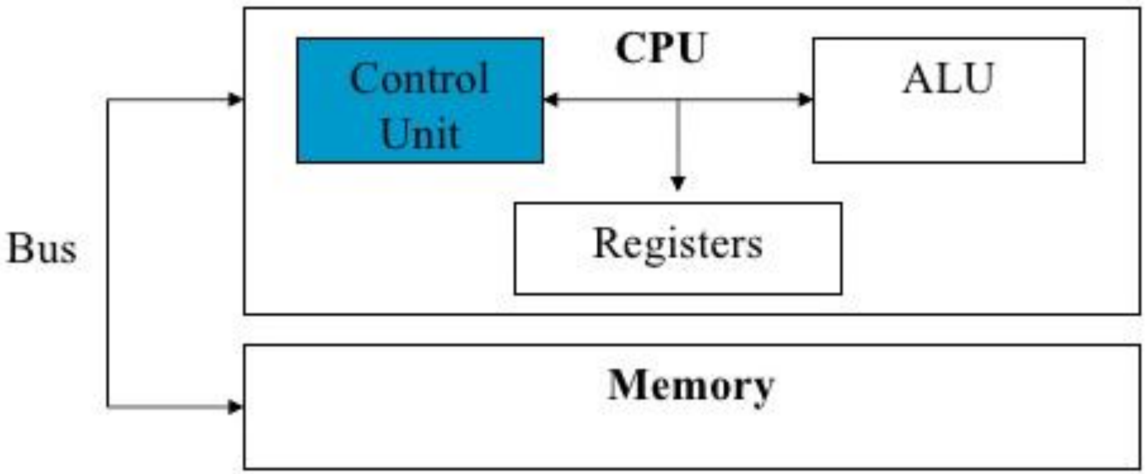
ALU

Computer System



- **Arithmetic and Logic Unit (ALU)**
  - It performs calculations and comparisons of data.

Computer System



Control Unit

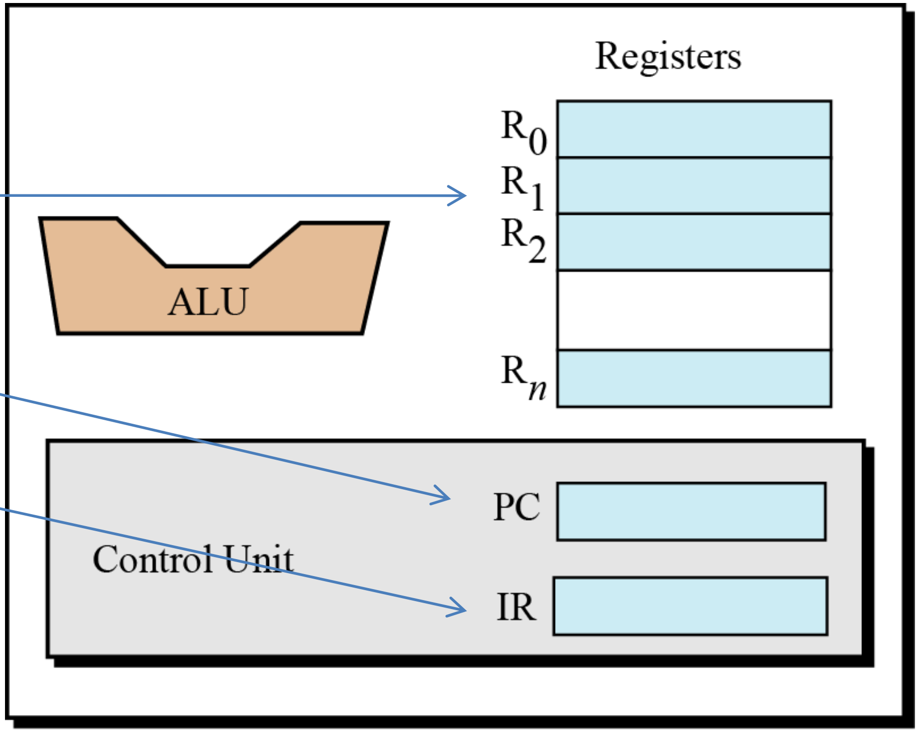
Note: The control unit controls the computer by repeating 4 operations, called the machine cycle. The 4 operations are: fetching program instructions from memory; decoding the instructions into commands that the computer can process; executing the commands; and storing the results in memory

Registers

Registers are fast stand-alone storage locations that hold data temporarily.

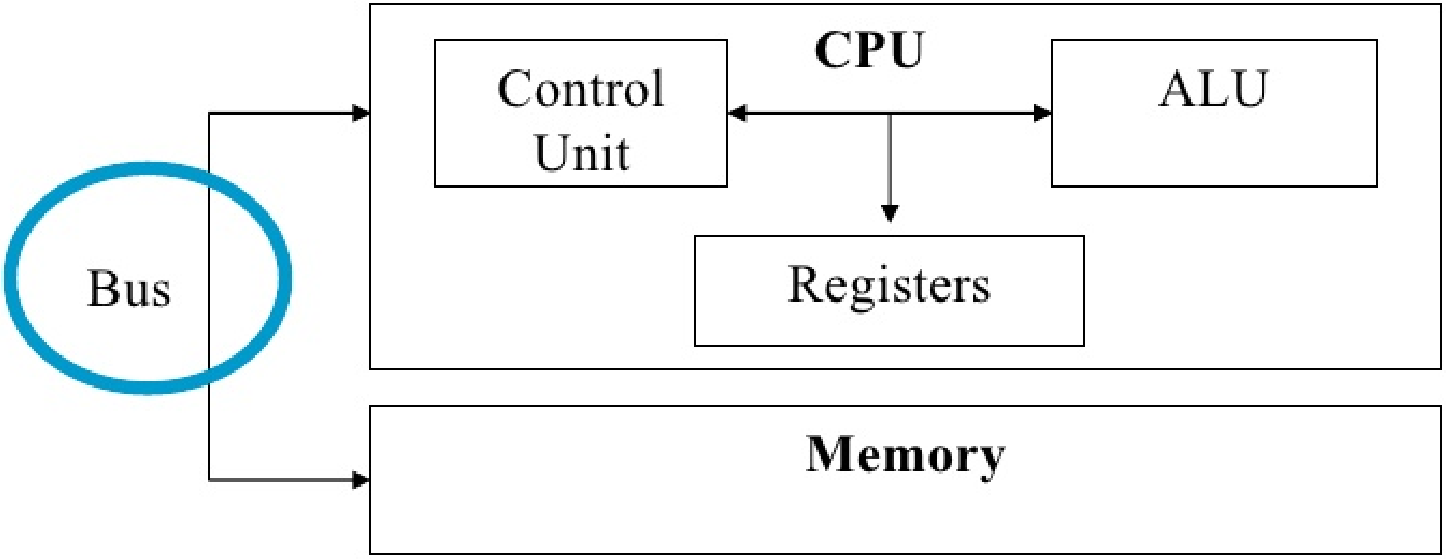
Multiple registers are needed to facilitate the operation of the CPU.

- ❑ Data registers
- ❑ Program counter
- ❑ Instruction register



Central Processing Unit (CPU)

Computer System



Buses

They are electrical pathways that carry signal (bits) between a CPU's components and outside devices.

MAIN MEMORY

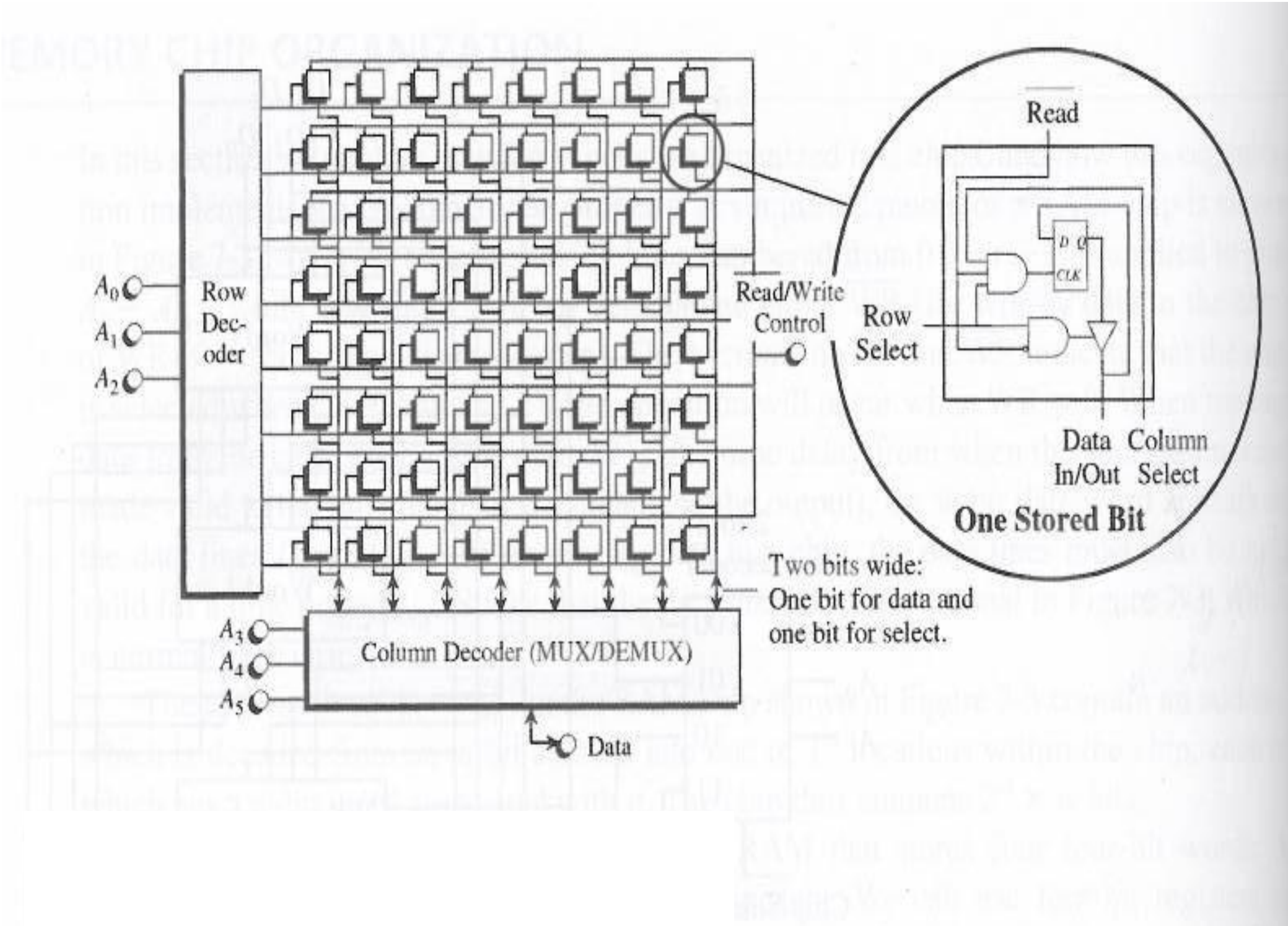
- accepts and holds program instruction and data
- acts as the CPU’s source for data and instructions and as a destination for operation results
- holds the final processed information until it can be sent to the desired output or storage devices, such as printer or disk drive

MAIN MEMORY

Main memory consists of a collection of storage locations, each with a unique id, called an address.

Data is transferred to and from memory in groups of bits called words. A word can be a group of 8 bits, 16 bits, 32 bits or 64 bits (and growing). If the word is 8 bits, it is referred to as a byte.

RAM Grid


Read
Row Select
Data In/Out
Column Select
One Stored Bit
Read/Write Control
Row Dec-oder
A_0
A_1
A_2
Two bits wide: One bit for data and one bit for select.
A_3
A_4
A_5
Column Decoder (MUX/DEMUX)
Data

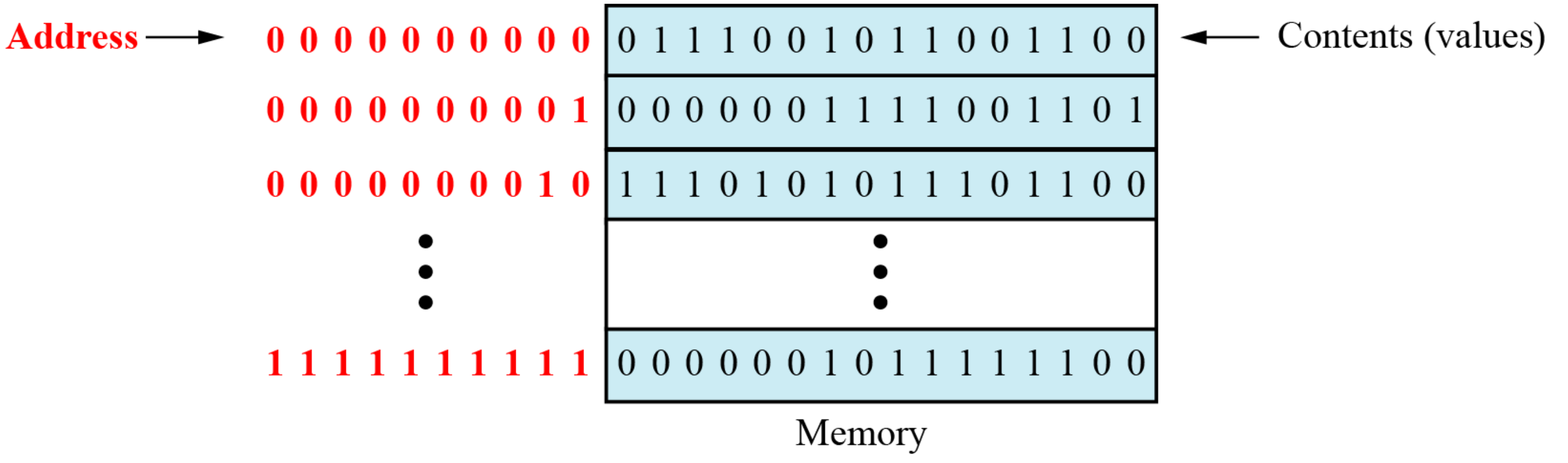


Figure: Main memory

Address space

To access a word in memory requires an identifier. Although programmers use a name to identify a word (or a collection of words), at the hardware level each word is identified by an address.

The total number of uniquely identifiable locations in memory is called the address space.
For example, a memory with 64 kilobytes and a word size of 1 byte has an address space that ranges from 0 to 65,535.

a byte-addressable 32-bit computer can address
2^{32} = 4,294,967,296 bytes of memory, or 4 gibibytes (GiB)

Table Memory units

Unit	*Exact Number of Bytes*	*Approximation*
kilobyte	2^{10} (1024) bytes	10^3 bytes
megabyte	2^{20} (1,048,576) bytes	10^6 bytes
gigabyte	2^{30} (1,073,741,824) bytes	10^9 bytes
terabyte	2^{40} bytes	10^{12} bytes

Memory addresses are defined using unsigned binary integers

Example 1

A computer has 32 MB (megabytes) of memory. How many bits are needed to address any single byte in memory?

Solution

The memory address space is 32 MB= 2^{25} ($2^5 \times 2^{20}$). This means that we need $\log_2 2^{25}$=25 bits, to address each byte.

Example 2

A computer has 128 MB of memory. Each word in this computer is eight bytes. How many bits are needed to address any single word in memory?

Solution

The memory address space is 128 MB, which means 2^{27}. However, each word is eight (2^3) bytes, which means that we have 2^{24} words. This means that we need $\log_2 2^{24}$, or 24 bits, to address each word.