

# COMP S266F

## Computer Architecture

### Chapter 1. Programmable Computers

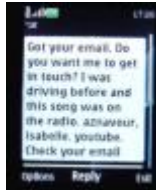


# **computer systems and computing processes**

# What is a Computer?

- Computers are machines that perform calculation:
  - Machines can be
    - mechanical
    - electronic
    - a combination of both mechanical and electronic
  - ***Programmable***, in general
    - Automatically carry out a sequence of operations under control of a stored program
  - High to low ***programmability*** (discuss in detailed later)
  - Not limited to desktop / personal computers!
- Computers are **ubiquitous**!
- Computing means calculating

# Ubiquitous (Examples)



*sms*



*auto-toll*



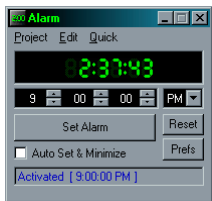
*ATM*



*elevator control*



*digital camera*

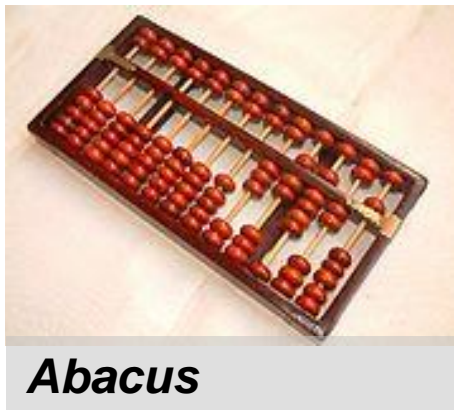


# Computers in the Old Days

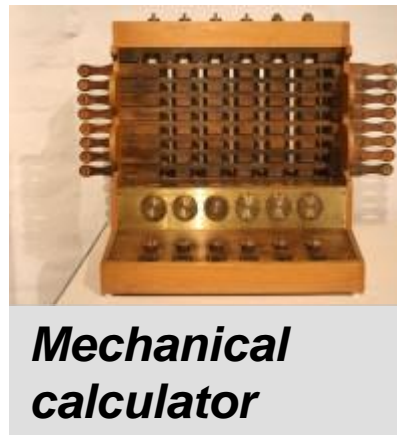
- Mechanical devices
  - Abacus
  - Mechanical calculator by Wilhelm Schickard in 1623
- Electronic devices
  - ENIAC
    - Ballistic calculation



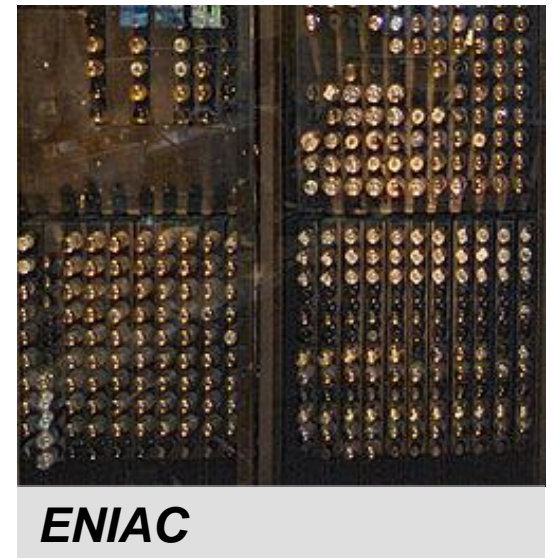
**TRANSISTORS**



**Abacus**



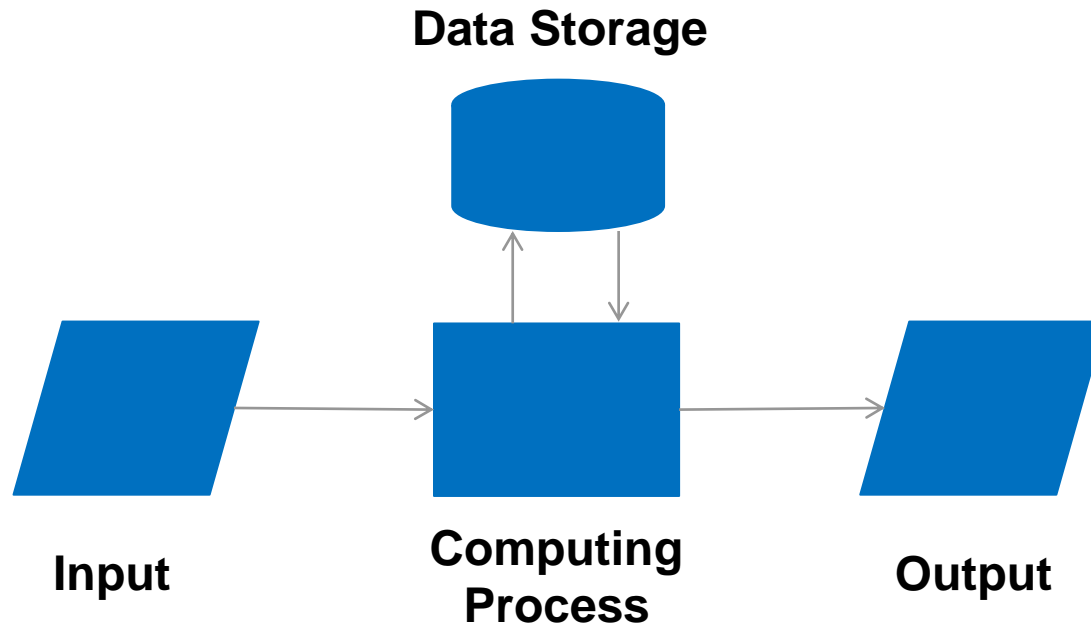
**Mechanical  
calculator**



**ENIAC**

# Computing Process – The User's Point of View

## The Input-Process-Output Model



# Computing Process – The User's Point of View

Basic process has three steps:

- Read input data
- Process data
- Write output data

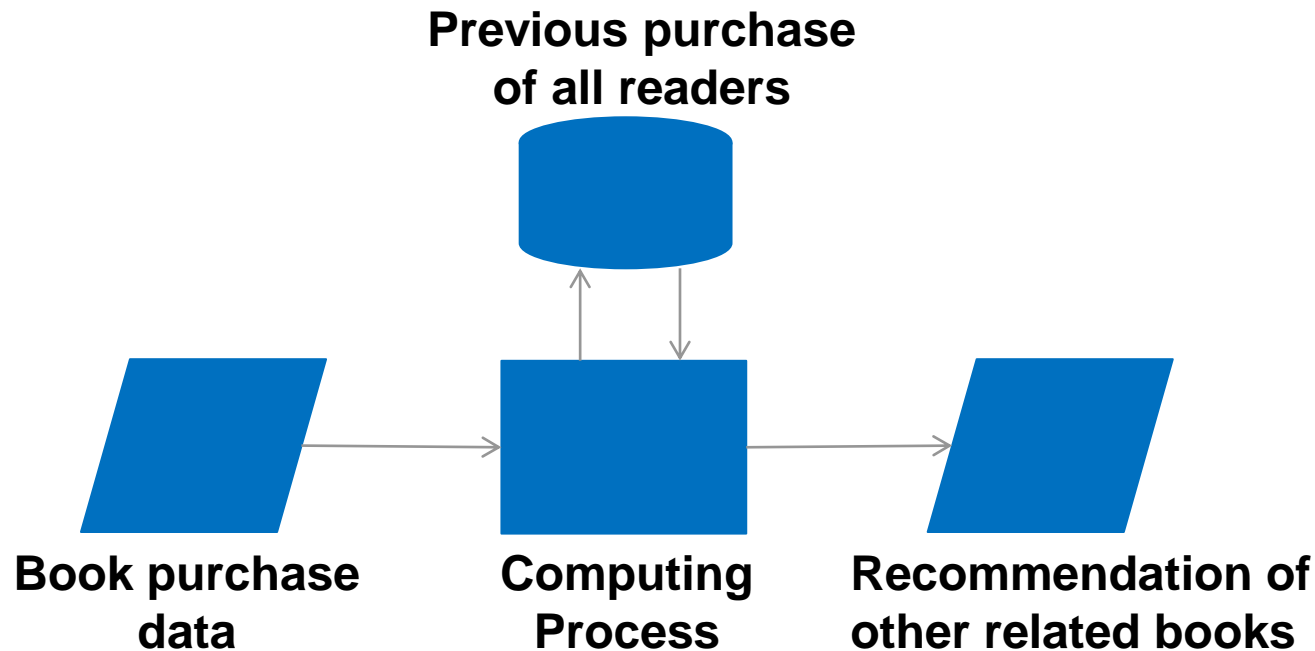
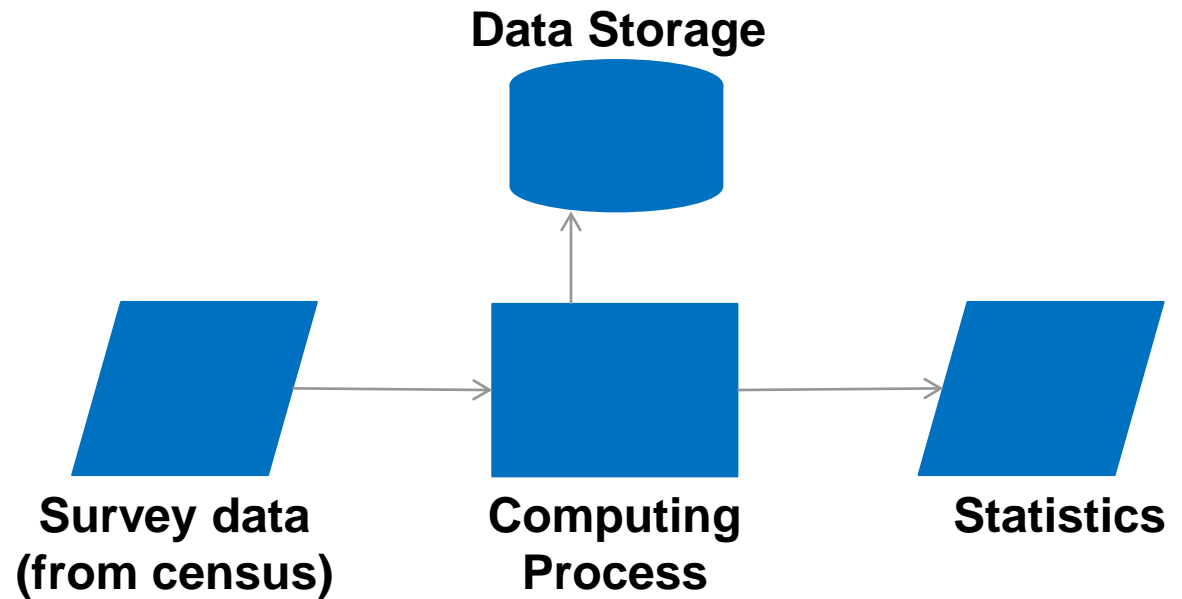
## Computing process

- computer program that is being sequentially executed by a computer

## Data storage

- Store data
- Retrieve data

# Examples of Computing Process





# What are the inputs, outputs & processes of these systems?



# Programmable computers

# What is programmable computers?

- Programming is putting instructions together meaningfully
- Programmable computer can execute instructions
  - Instead of executing a fixed process
- Another composition of instructions, another process
- Programming can be done by
  - End-users (without *real* programming knowledge)
  - Computer programmers

# Programming versus Coding



## Programming

- process of creating a program that follows certain standards and performing a certain task

Include

- Planning
- Designing
- Testing
- Deployment
- Maintenance

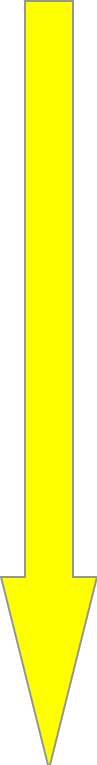
## Coding

- Subset of programming
- Translating human readable language (pseudo code) to machine readable language

# Programmability

- Degree of how a system can be programmed
- Low programmability
  - Some flexibility in adjusting the computing process
  - Vacuum robots, vehicle navigation systems
- High programmability
  - Re-purposed to serve a larger range of processes
  - Modern desktop computers, smart phones and tablets

# From Low to High Programmability

- 
- A toaster with a time knob
  - A washing machine with programs for various types of clothing.
  - A DVD recorder supporting various recording modes.
  - A programmable calculator supporting programmed sequences of calculation steps.
  - An Excel spreadsheet supporting functions and macros.
  - A modern general purpose computer system

**We aim high**

# How to achieve high programmability?

- A large amount of **instructions** available
- A high **flexibility** to compose and sequence the instructions
- Short time and **little effort** to re-program a system

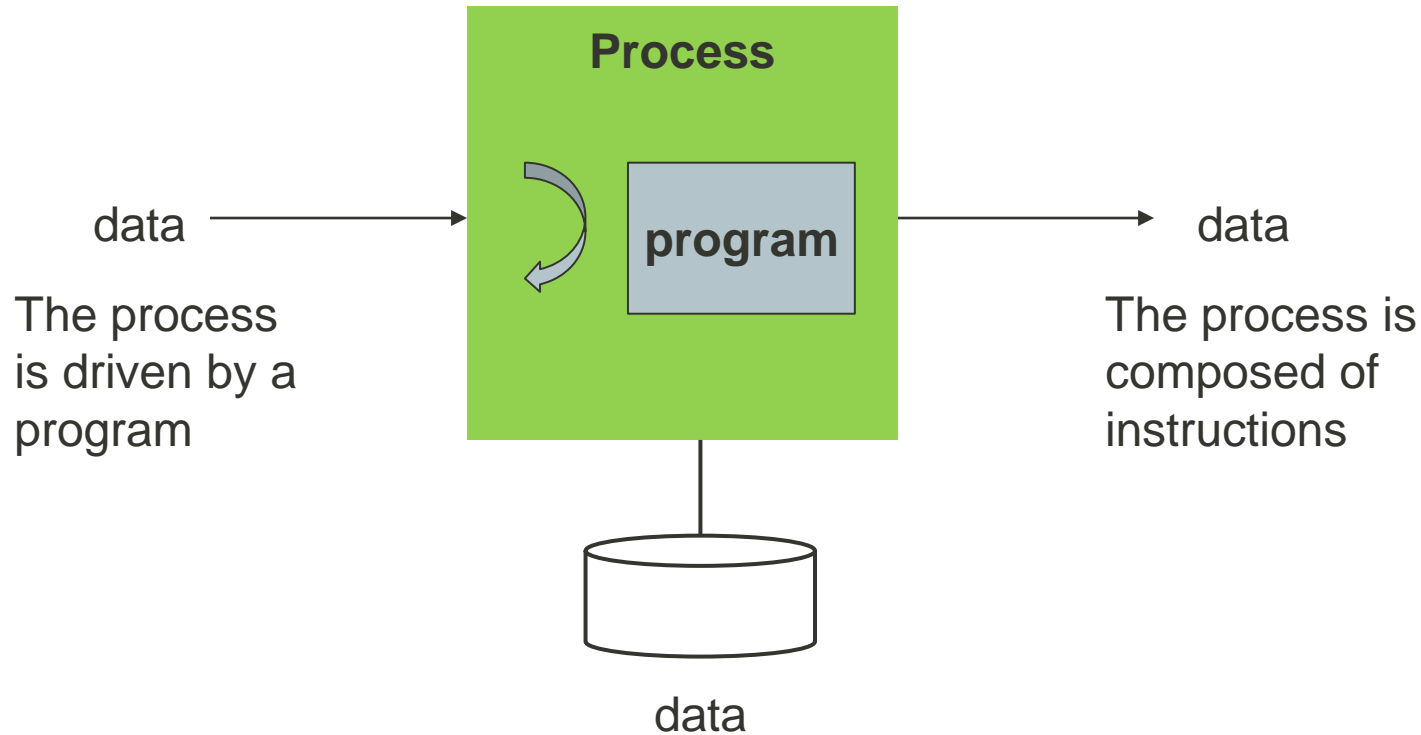
# **Components of programmable computers**



# Items relevant to programmability

- Programs: contain instructions
- Instructions: which are commands executable by the computer
- Data: to be processed by the computer

# Programmable enabled processes



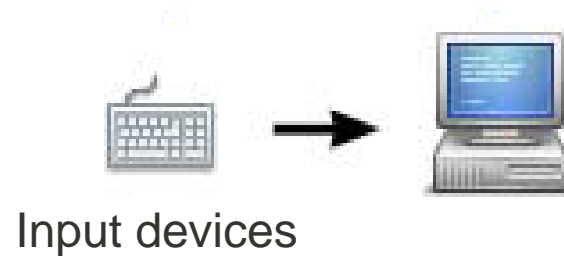
# Programmable enabled processes

- Instruction execution
  - an essential function of the programmable computer
- Data storage
  - a function for storing the data before and after the instruction execution
- Program storage
  - a function for storing the program in the programmable computer for instruction execution
- Inputting data
  - Data (including program) that a computer receives
- Outputting data
  - Data that a computer sends

# Be a useful computing process

- **MUST** include input and output for interacting with the outside world

Example of input:



Example of output:



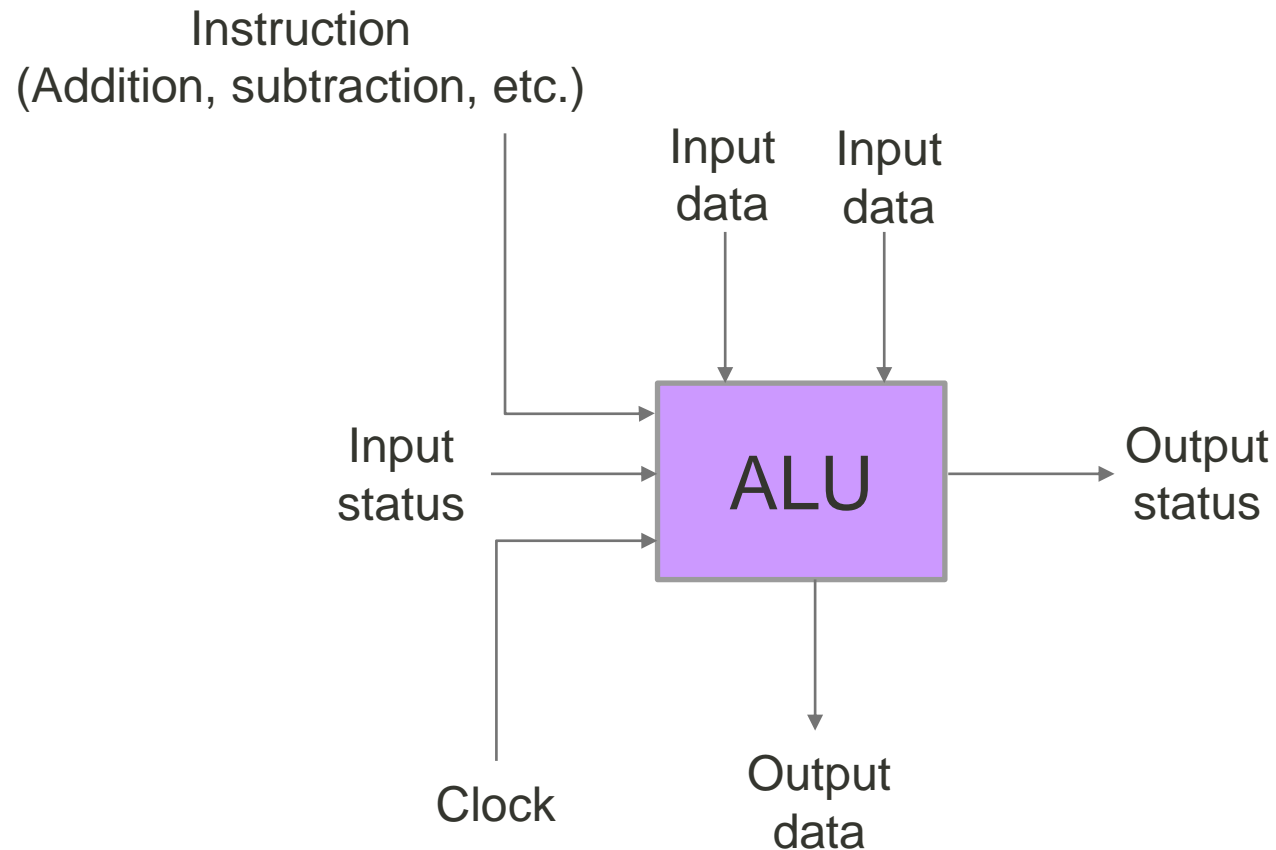
# Major components of a programmable computer

- Arithmetic and Logic Unit (ALU)
  - for instruction execution
- Memory system
  - for data and program storage
- Input device
  - for data input (including program) **into** the programmable computer
- Output device
  - for data output **from** the programmable computer

# Arithmetic and Logic Unit (ALU)

- Functional unit for instruction execution
  - ALU accepts data and instructions
  - ALU produces data output which is a result of instruction execution
- Fundamental building block of computing circuits (e.g. CPU, GPU)

# Arithmetic and Logic Unit (ALU)



# Arithmetic and Logic Unit (ALU)

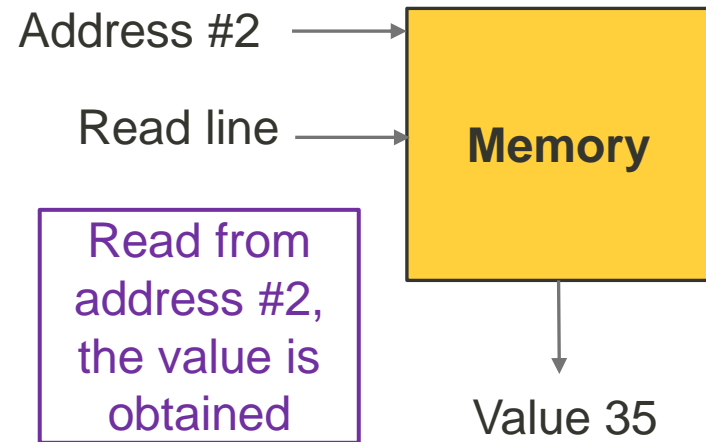
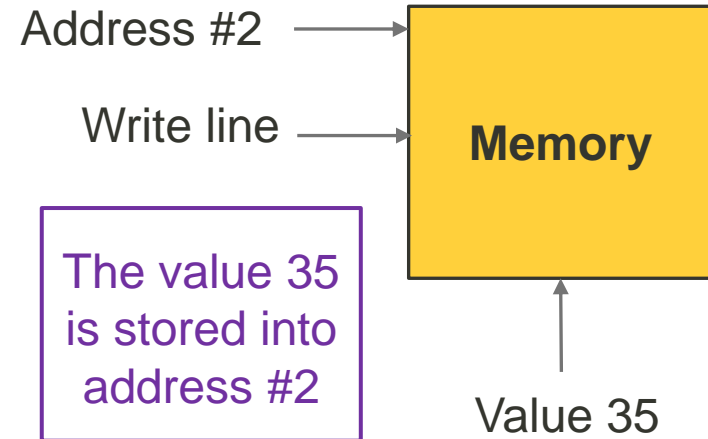
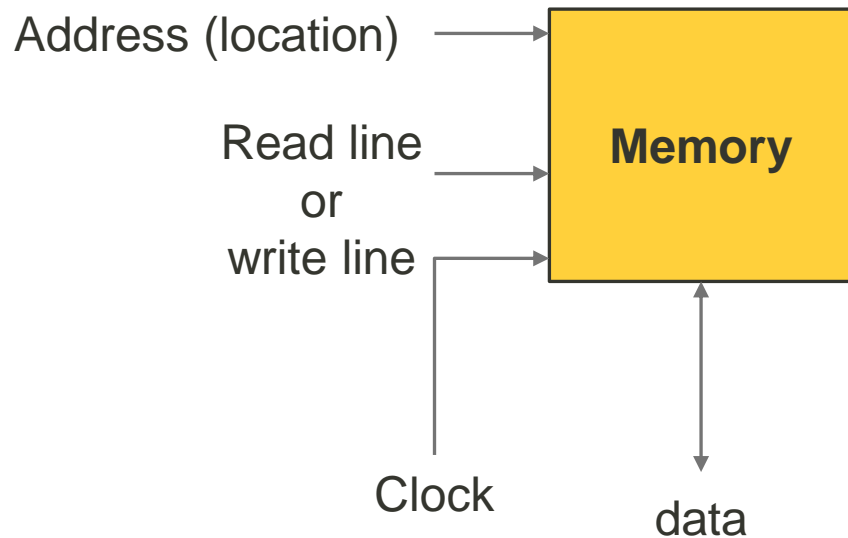
- Most ALU has two data input ports
  - Input data to be operated on, called operands
- ALU operation is controlled by the instruction port
  - The instruction operates on the data presented at the input ports
  - The code presented at instruction port mapped to a particular instruction
- ALU operation timing is controlled by the clock port
  - A clock signal is provided to control when operations take place.



# Memory

- Functions of memory
  - Retrieve data (Read)
  - Store data (Write)
  - Overwrite a previously stored data (Overwrite)

# Memory

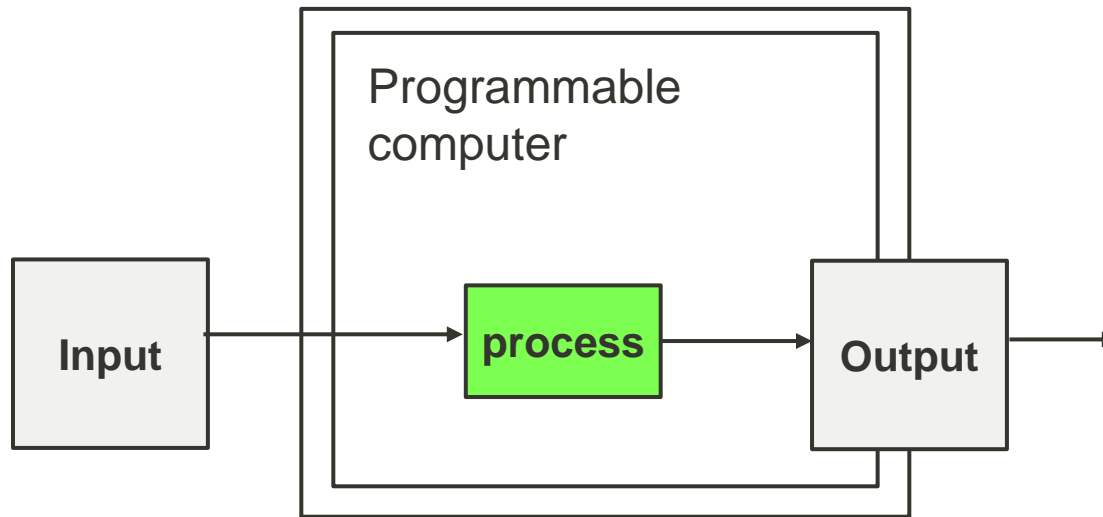


# Memory

- Duplex data channel
- Read/write line
- Clock line
- Data operations involve a data unit
  - The size of data unit varies: bit, byte
- Address to identify each data unit in memory
  - Numbered 0 onwards
  - Number of addresses determine the size of memory system
  - Address specifies which data unit for the current operation

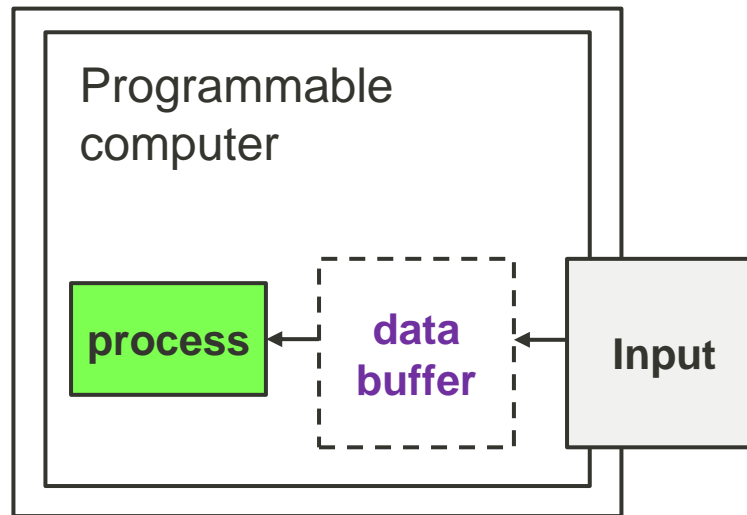
# Input and Output

- Connect the programmable computer to the outside world.



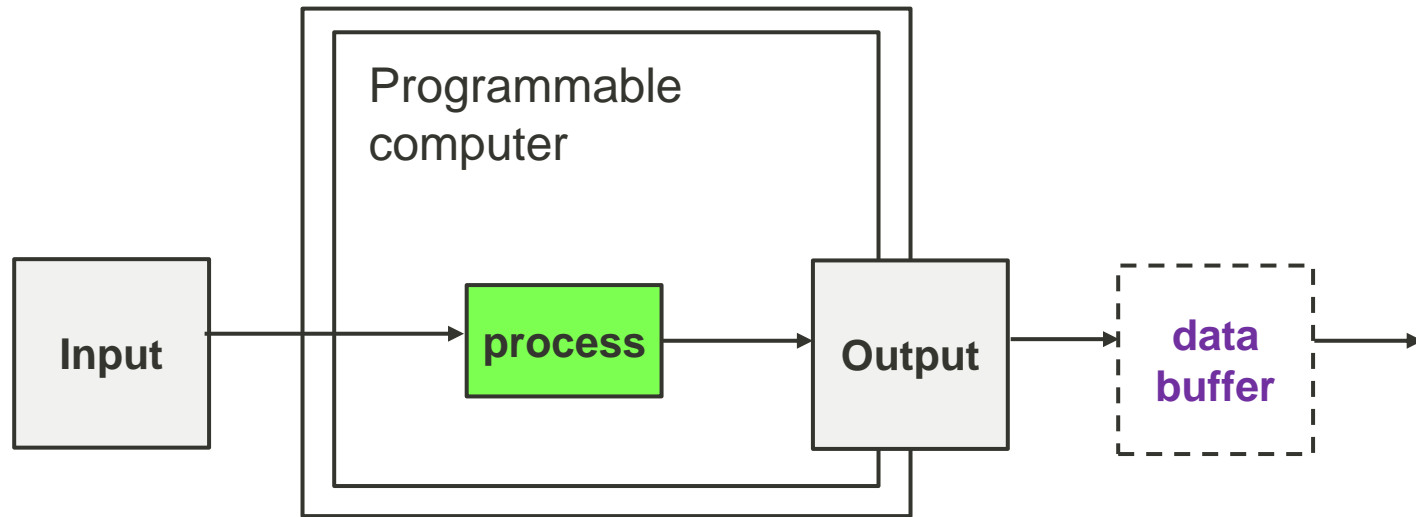
# Input and Output with Data Buffer

- **Data buffer**
  - is a region of a physical memory storage used to temporarily store data while it is being moved from one place to another
  - is typically used to compensate for a difference in rate of flow of data transferring from one device to another



- **Input data buffer** is used to store the data temporarily
  - Input data comes but the computer may not be ready to receive it

# Input and Output with Data Buffer

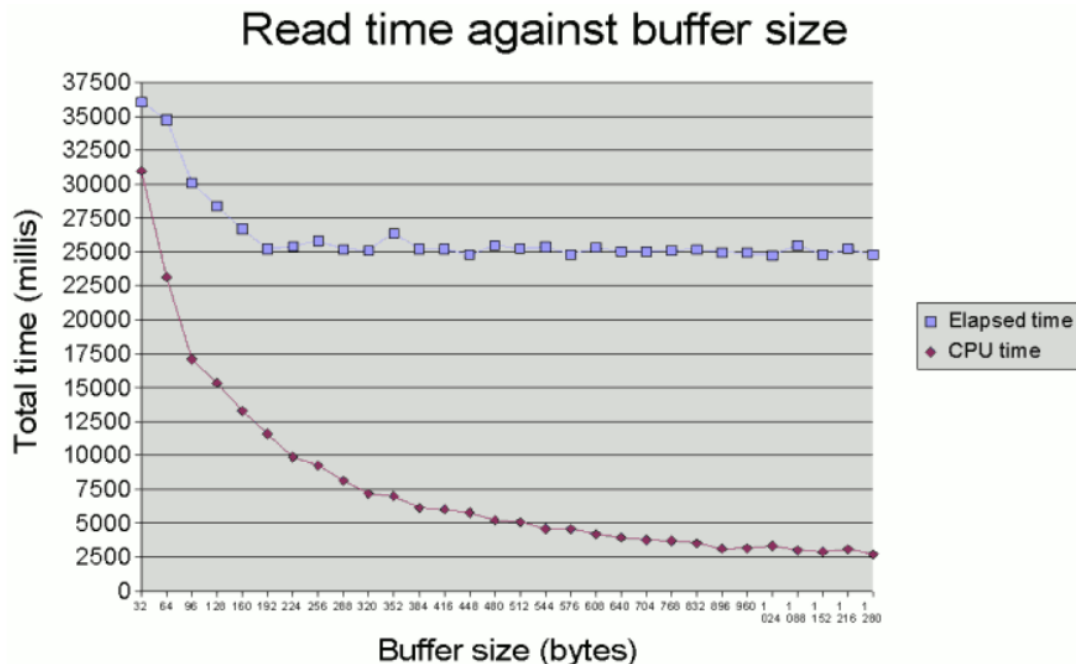


- **Output data buffer** is used to improve efficiency
  - Data is sent in large batches
- The outside world is beyond the realm of computers

# What is the ideal input buffer size (bytes)?

Let's refer to the following example

- CPU time: Quantity of processor time taken by the process. This does not indicate duration.
- Elapsed time: Total duration of the task.



[https://www.javamex.com/tutorials/io/input\\_stream\\_buffer\\_size.shtml](https://www.javamex.com/tutorials/io/input_stream_buffer_size.shtml)

# What is the ideal input buffer size (bytes)?

- Shorter elapsed time indicates faster response time
- Shorter CPU time, longer elapsed time
- Longer CPU time, shorter elapsed time

