Queen Mary
University of London
Science and Engineering

School of Electronic Engineering and Computer Science
QMUL-BUPT Joint Programme

EBU6475 Microprocessor System Design
EBU5476 Microprocessors for Embedded Computing

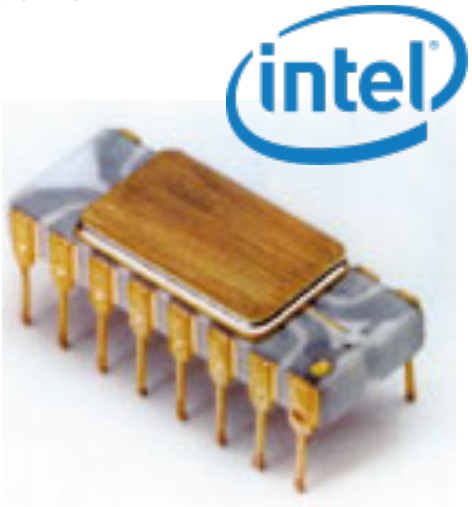
Computer Design and Organisation – the Basics

arm
University Program Education Kits

Last updated: 19 February, 2020

The First Microprocessor

- In 1971 Intel introduces the first commercially available microprocessor – the 4004
- 4-bit data bus, 45 instructions
- Required many support chips to build a functioning system
- 2,300 transistors on one IC



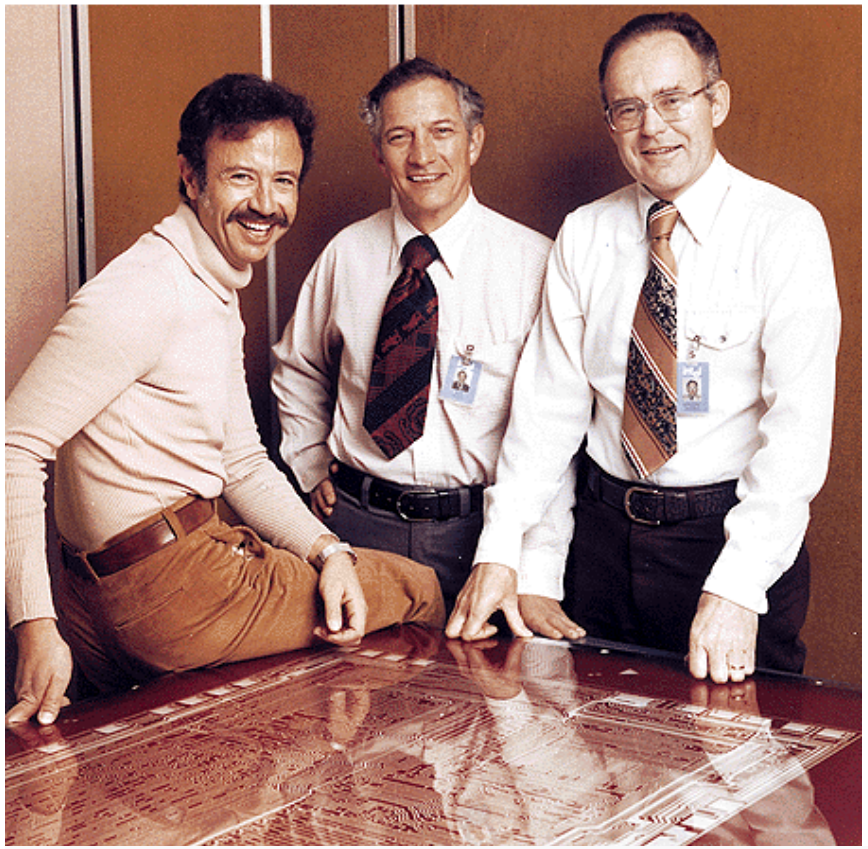
Since 1970s, we have been observing great developments in both computer architecture and integrated circuit fabrication.

Microprocessors become more powerful but at the same cheaper!

Moore's Law

The number of transistors on a chip will roughly **double** every two years.

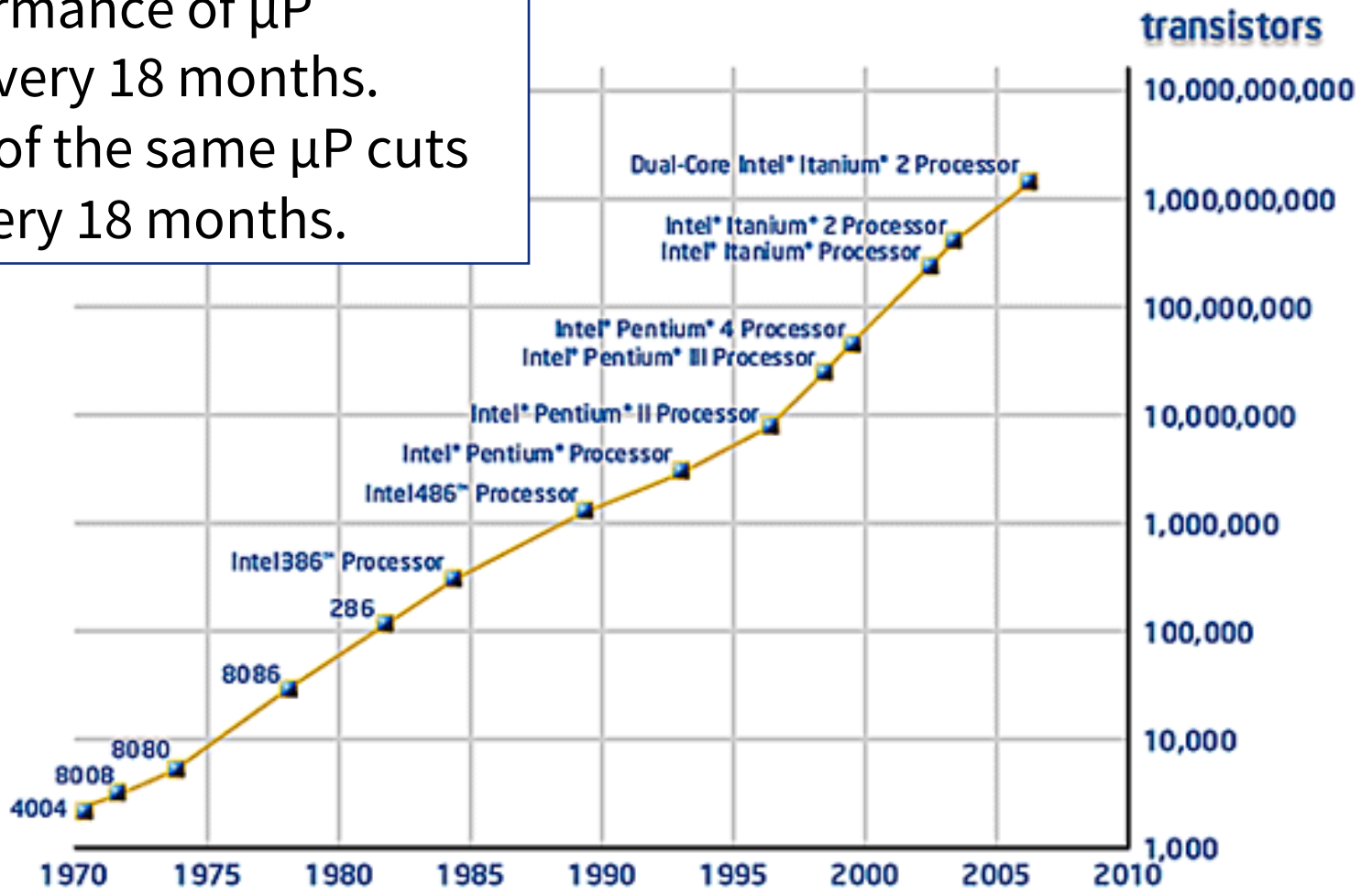
- Gordon Moore's prediction from 1965!
- He has been proved right so far...



Founders of Intel: Andy Grove, Robert Noyce and Gordon Moore

Moore's Law (Cont')

The performance of µP doubles every 18 months.
The price of the same µP cuts by half every 18 months.



Microcontrollers

Following Moore's law, the use of microprocessors has extended from personal computers to a lot of daily-life applications like VCR, clock radios, TVs, automobiles, etc.

A by-product of µP development was microcontroller, which is an integrated computer system, for a single or several designated task.

Micro-"Computer" is everywhere now!
BUT WHAT IS REALLY A COMPUTER?

What is a Computer?

- A computer is a machine that can perform simple calculations
- But a computer can also process algorithms where it ...
  - performs a sequence of calculations;
  - makes decisions based on the results of calculations; and
  - repeats the sequence if wanted.

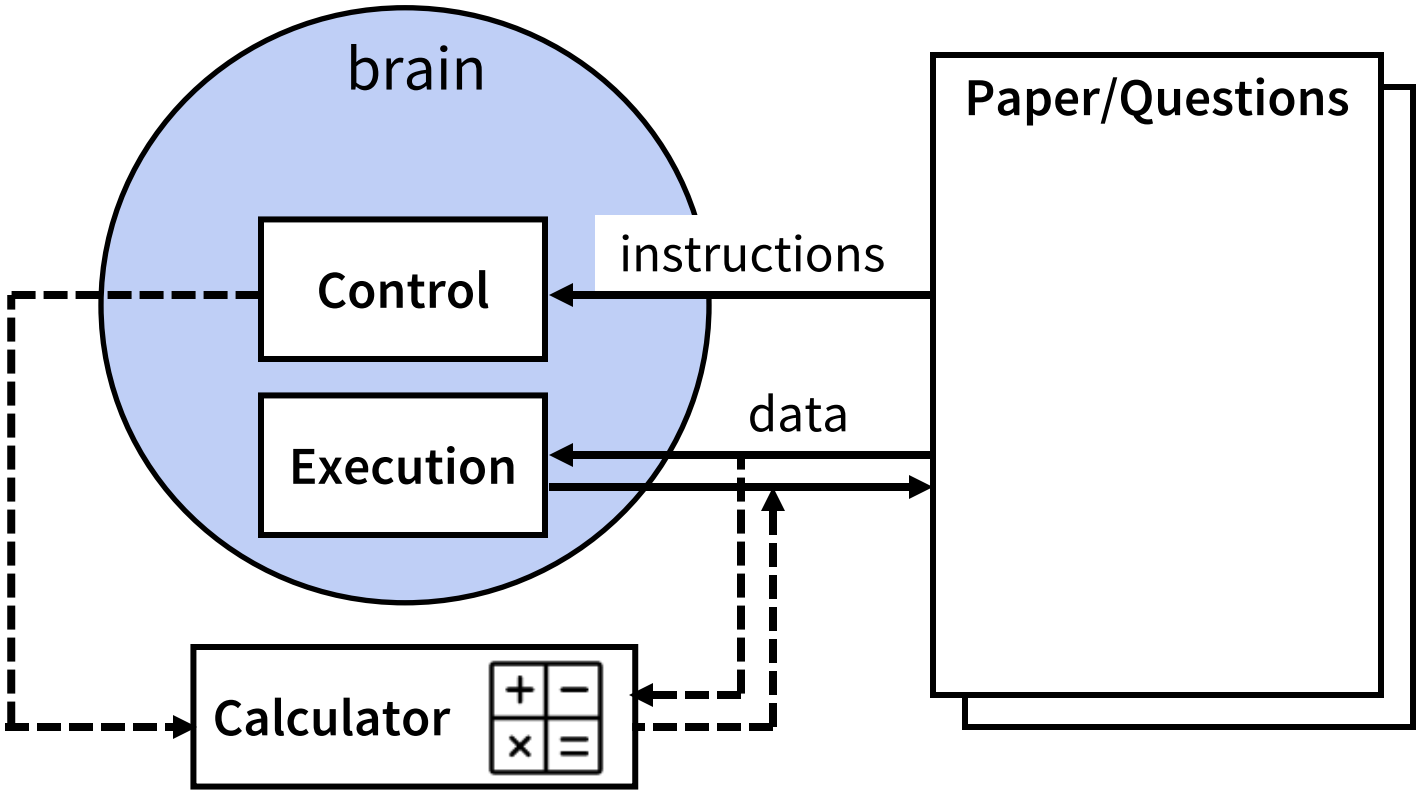
Math Quiz!

1. 1 + 2 + 3 + 4 + 5 = ?
2. 16 x 8 + 29 = ?
3. A circle has a radius of 5 cm, what is its area?

Let's think: how did you answer the above questions?

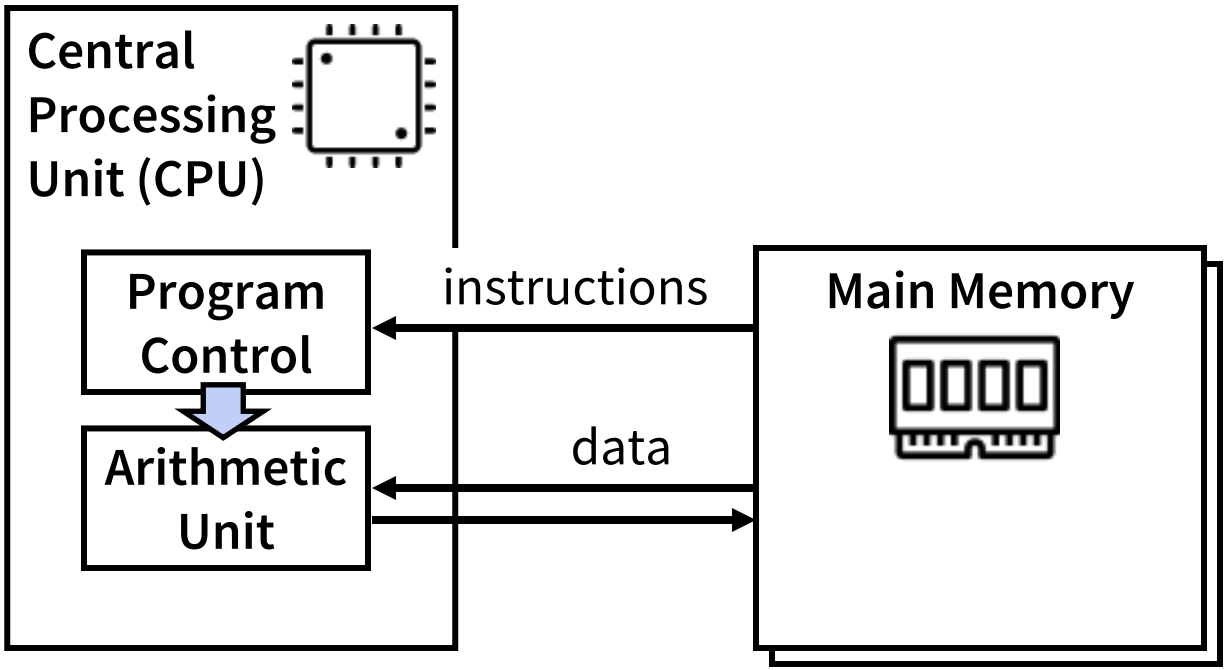
Concept of Computation

Let's start with the way we handle computations...



Concept of Computation (Cont')

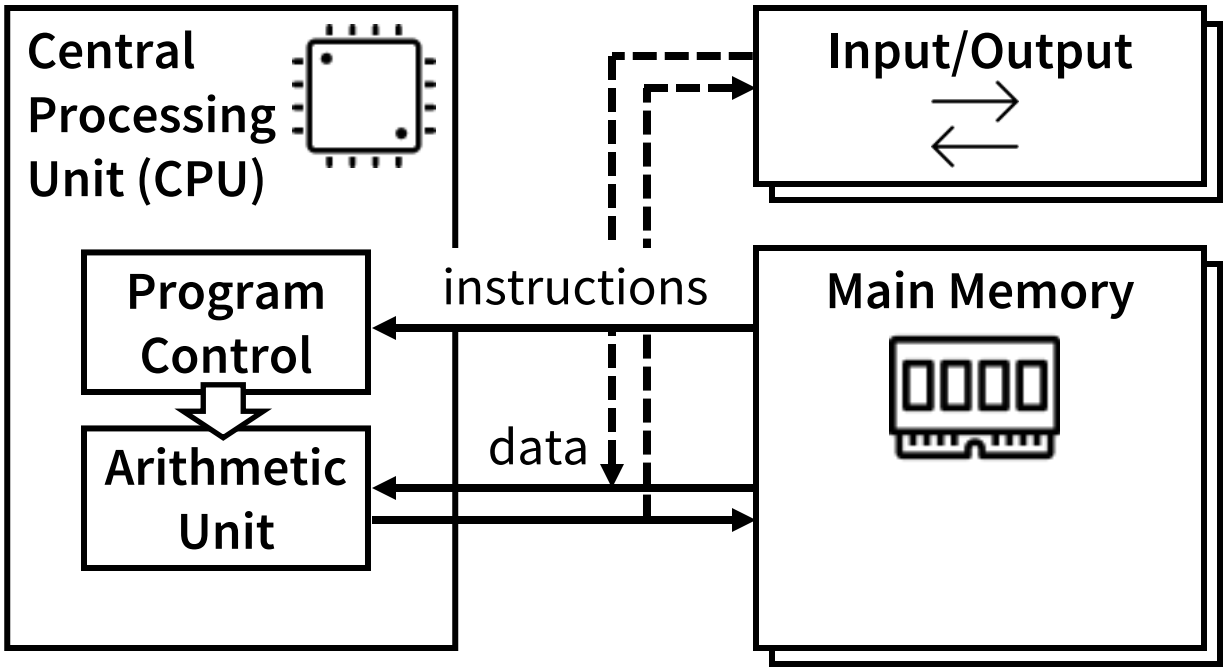
How about machine computations?



Is there anything missing?

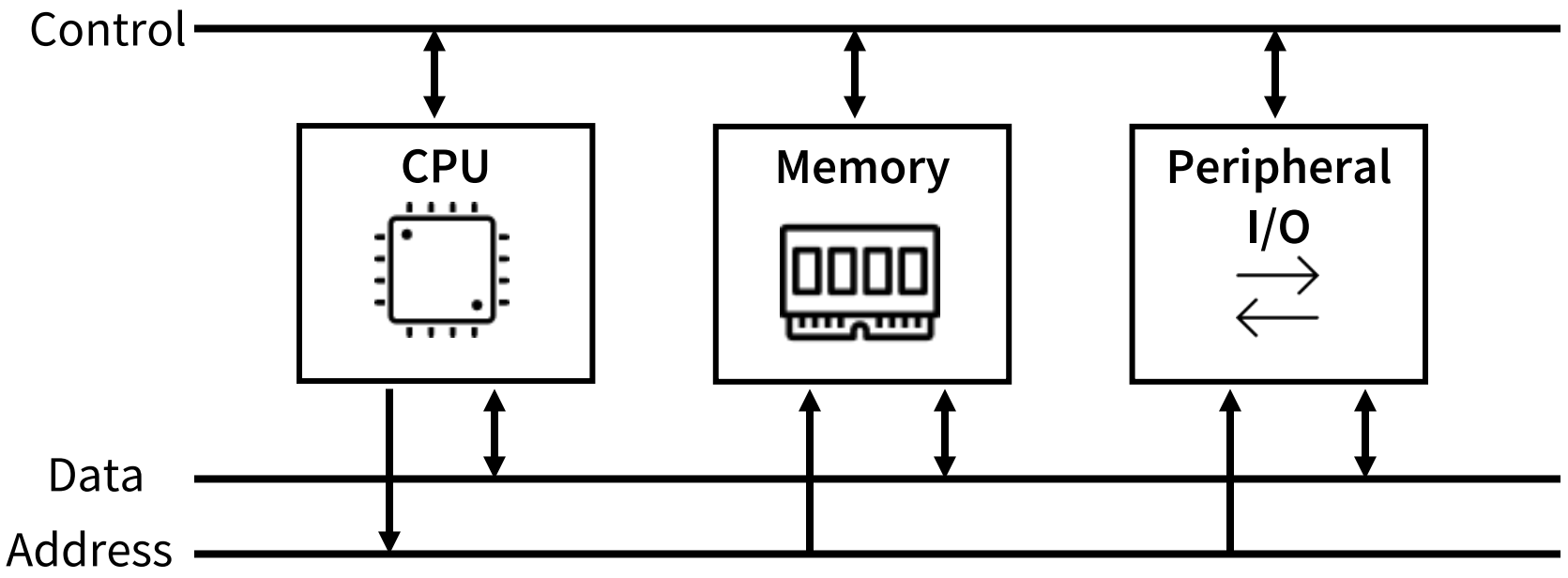
Concept of Computation (Cont')

I/O provides extra data and allow interactions.



Do we have a simple model to describe a computer?

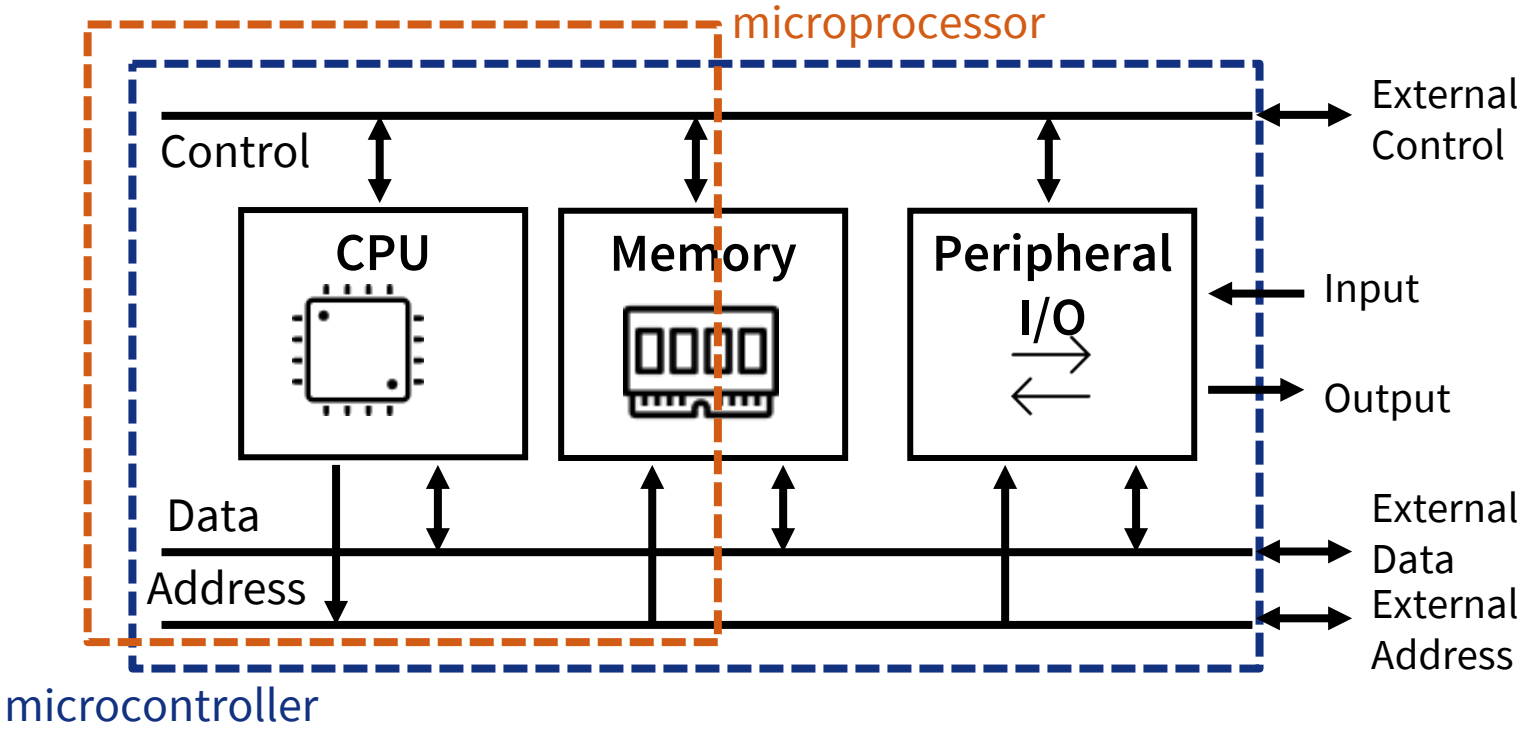
Von Neumann Architecture



Component	Functions
Central Processing Unit (CPU)	controls the system and performs calculations
Main Memory	stores both programs and data
Peripheral Input/Output (I/O)	allows data to be input to system allows results to be output from system

Microprocessor vs Microcontroller

A microprocessor mainly refers to the CPU with some memories.



A microcontroller unit (MCU) is a microprocessor integrated with both memory and I/O. It is a general-purpose device that is designed to fetch data, perform limited calculations and control the environment.

Stored Program Concept

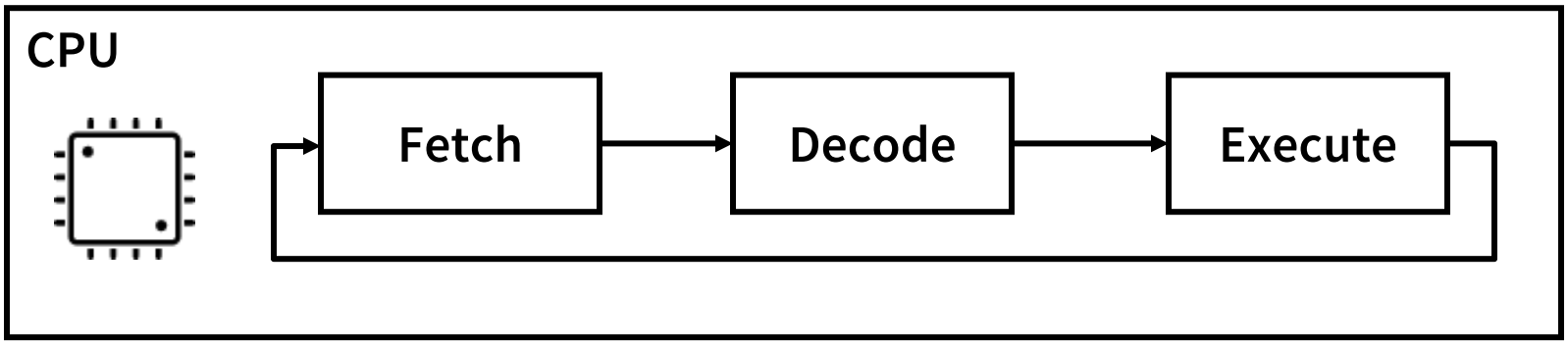
What is inside my computer program?

Stored Program Concept

- The CPU executes instructions stored in memory.
  - So called the "stored program" concept
- Recall there are two kinds of memory to store programs and data that are in use.
  - RAM – Random Access Memory
    Can be used for both programs & data
  - ROM - Read-Only Memory
    Can be used for fixed programs & constant data
- Different computers have different styles in arranging the memories – computer architectures

Fetch-Decode-Execute Cycle

- The CPU is a finite state machine (FSM) which runs the programs stored in the memory by the user.
- It repeatedly performs three operations:
  - Fetch – retrieve an instruction from memory
  - Decode – interpret the instruction
  - Execute – control appropriate hardware to carry out the instruction



What is inside a Program?

- We had written "computer programs" in C or Java.
  - These were compiled and executed.
- If we do not forget a µP/µC is a digital electric circuit (hardware - HW), then the program must be stored as strings of '0' or '1' bits in the memory.
- Assembly programming is considered the closest and the lowest level of programming to the HW.
  - We write instructions that the machine readily understands and executes.

Instructions

To command computer's hardware, you must speak its language.

- The words of a computer's language are called instructions, and its vocabulary is called an instruction set.
- It is interesting that there are different dialects of computer languages. It is easy to pick up others if you once learn it.
- The functionalities of a computer are revealed with its instruction set.

Instructions: Operations

Every computer must be able to perform simple arithmetic:

```
ADD r3, r1, r2     ;r3 = r1 + r2
```

instruct a computer to add two variables r1 and r2 and to put their sum back in r3.

The words to the right of the semi-colon (;) are comments for the human reader.

Q: Why is an instruction usually simple?
A: Simplicity favours regularity.

Instructions: Operands

The operands of instructions are restricted. They must be from a limited number of special locations in HW called registers.

```
ADD r3, r1, r2      ;r3 = r1 + r2
```

In this example, all r1, r2 and r3 are registers.

In writing instructions, we often need to assign variables to registers. You should note that some registers are dedicated for a special purpose, e.g. r15 is the program counter.

Q: Why is the number of registers small and limited?
A: Smaller is faster. A very large number of registers may increase the clock cycle time.

Data Transfer between Registers

If arithmetic operations occur only on registers, then it should be allowed to transfer data between registers.

```
MOV r2, r1      ;r2 = r1
```

instruct a computer to copy/move the value stored in register r1 to the register r2.

We'll see how to transfer data between register and memory (load & store).

Compiling Two C Statements

Translate the following C-style statement into instructions:

```
w = x – (y + z);
```

We can translate into two instructions

```
ADD r3, r1, r2    ; y: r1, z: r2, w: r3
SUB r3, r0, r3    ; x: r0
```

This is what a C compiler will do.
It analyses our statements and work out the sequence of machine instructions.

We will study the principles of compilation into assembly in details later in the module.

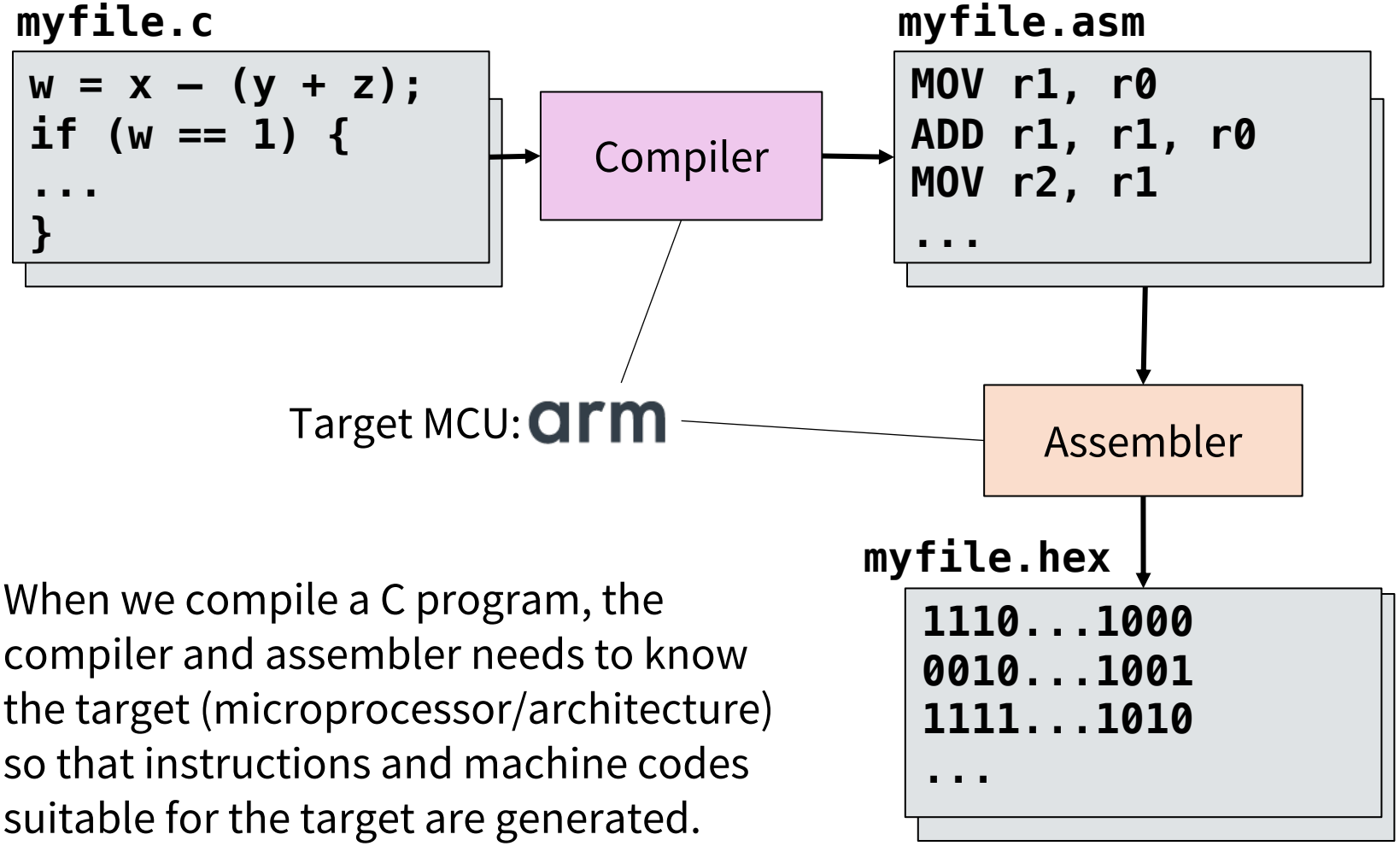
Assembly Language

To make the machine executes our instruction, an assembler will translate our symbolic notation into machine code:

`MOV r2, r1 → 1110 0001 1010 0000 0010 0000 0000 0001`

- Writing in binary is far too difficult for us.
- We prefer to use the symbolic language that called the assembly language (ADD, MOV, etc.).
- Assembly language requires the programmer to write one line for every instruction that the machine will follow, forcing the programmer to think like the machine.

Compilation and Assembling



When we compile a C program, the compiler and assembler needs to know the target (microprocessor/architecture) so that instructions and machine codes suitable for the target are generated.

Summary (1)

- Development of microprocessors has been following the famous Moore's law.
  - As a result, microcontrollers are widely used in many embedded applications.
- Classic design of a computer contains three main components: central processing unit (CPU), memory and I/O ports.
  - A program is stored in memory and then executed by the CPU using the fetch-decode-execute cycle.
- Program is a sequence of computer instructions like arithmetic and data transfer operations.

Summary (2)

- Assembly language is the symbolic language used by programmer to instruct the computer in a primitive way.
  - Assembler, which translate assembly language into machine format, is one of the earliest software development tools.