ESET 349 – Microcontroller Architecture

Introduction

Dr. Muhammad Faeyz Karim

Instructor and TA Details

Instructor:	Dr. Muhammad Faeyz Karim
Office:	Fermier 111B
Phone:	979-458-6496
E-Mail:	faeyz@tamu.edu
Office Hours:	Monday, 3:00 PM – 4:00 PM, Wednesday, 3:00 PM – 4:00 PM, Friday, 3:00 PM – 4:00 PM, or by appointment

Lab Teaching Assistants: Mr. Sanjaya Mallikarachchi (sanjayadpf@tamu.edu)
Section - (501, 505, 507. 508)
Ms. Nazanin Golshanara (n.golshanara@tamu.edu)
Section- (502, 503, 504, 506)

Tentative Course Schedule

Week	Lecture	Lab
1/14	Course introduction and overview	No Lab
1/21	Number Systems and floating point	No Lab
1/28	Microcontroller Architecture	No Lab
2/4	Introduction to ARM Assembly language	Lab 1 - Introduction to MSP432 architecture - at CPU register level and memory level using debugging tools
2/11	Load, Store and Addressing	Lab 1 - Cont.
2/18	Port Programming	Lab 2 - Programming for data transfer between register and memory locations and to find particular value in the memory
2/25	Review and Mid Term Exam (02/28)	Lab 3 - Toggling LEDs using assembly language programming
3/3	MOV, Constants and Literal Pool	Lab 4 - Selecting an LED based on switch input
3/10	Spring Break	No Lab
3/17	Logical, Arithmetic and Shift Operations	Lab 5 - Obstacle detection
3/24	Decision making and loops	Lab 6 - Interfacing 16x2 LCD display to MSP432 launchpad Project - Introduction
3/31	Tables	Lab 7 - Binary search Project - Design and work
4/7	Stack Operations	Project - Design and work
4/14	Subroutines	Project - Design and work
4/21	C and Assembly	Project - Submission
4/28	Exception and Revision	No Lab
5/5	Revision	No Lab
	Final Exam 05/03 @ 3:30 PM (Section 505-508) and 05/06 @ 3:30 PM (Section 501-504)	

Course Information

Sections: 501, 502, 503, 504, 505, 506, 507, 508

Lecture Time: Mon, Wed, Fri - 1:50 PM – 2:40 PM, RICH 101, Section 501,502,503, 504
Mon, Wed, Fri - 4:10 PM – 5:00 PM, FERM 303, Section 505,506,507, 508

Lab Time: 501 – Monday 5:00 PM – 7:30 PM, THOM 101A
502 – Tuesday 2:00 PM – 4:30 PM, THOM 101A
503 – Thursday 2:00 PM – 4:30 PM, THOM 101A
504 – Thursday 5:00 PM – 7:30 PM, THOM 101A
505 – Tuesday 2:00 PM – 4:30 PM, THOM 101
506 – Wednesday 11:00 AM – 1:30 PM, THOM 101
507 – Thursday 2:00 PM – 4:30 PM, THOM 101
508 – Thursday 5:00 PM – 7:30 PM, THOM 101

Grading Policy

Grading Scale	
Homework	8%
Midterm Exam	20%
Quiz	10%
Labs	20%
Project	10%
Final Exam	25%
Attendance & participation	7%

Course Management

- ❑ Each work/assignment must be submitted within deadline. Late submissions will be penalised with a deduction of 20% points every day.
- ❑ Make up may be given with excused letter. Work submitted by a student as makeup work for an excused absence is not considered late work and is exempted from the late work policy.

Course Objectives & Motivations

- Understand what is a Microprocessor.
- Know the internal workings of an ARM processor.
- Ability to program an ARM processor in assembly.
- Why Assembly program is important in understanding ARM Microprocessor Architecture?
  - High-level C program hides all the low-level data manipulation and data movement between registers and memory; but assembly code can illustrate these low-level functions. Assembly code is the closest on what the computer understand.
  - Assembly code helps student to build up logic thinking in the latest microprocessor architecture.
  - Numerical computation can be best manipulated and understood at low level. It builds up student deeper understanding in digital computing and have confidence in dealing with digital data.

Sample Lab/Project demonstrations

- ❑ Registers
- ❑ Memory access
- ❑ Toggling LEDs
- ❑ Switches
- ❑ Photo sensor
- ❑ Obstacle Detection
- ❑ LCD display
- ❑ Stepper Motor

Text/Reference Books

- Textbook
  - William Hohl, ARM Assembly Language Fundamentals and Techniques, CRC Press, 2009 (ISBN 978-1-4398-0610-4).

- Reference books
  - M. A. Mazidi, et. el. TI MSP432 ARM Programming for Embedded Systems. ISBN- 13:978-0997925913, ISBN-10:1512185671.
  - Jonathan Valvano. Introduction to the MSP432 Microcontroller Embedded Systems. 6th Edition. ISBN-13:978-1512185676.
  - D. A. Patterson, Computer Organization and Design, 5th Edition, 2013.
  - MSP432 materials from Texas Instruments web site
  - https://www.ti.com/lit/ug/slau356i/slau356i.pdf?&ts=1589807021901

Tips to perform well in the course

1. Taking good notes is the essence of performing well in the course.
2. For each topic, tutorial questions will be discussed and made available in the Canvas site, This will be a very valuable asset for you. Take advantage of it. Because, by solving these questions, you can assess your depth of understanding of the materials as well as you will get prepared step by step for your tests.
3. If needed, feel free to make an appointment for office help. Instructor will try to prioritize your times.

Academic Integrity Statement and Policy

"An Aggie does not lie, cheat, or steal or tolerate those who do."

- At the end of the day, Cheating comes back to self
- Nobody can cheat others

Academic Integrity Statement and Policy
"An Aggie does not lie, cheat or steal, or tolerate those who do."

"Texas A&M University students are responsible for authenticating all work submitted to an instructor. If asked, students must be able to produce proof that the item submitted is indeed the work of that student. Students must keep appropriate records at all times. The inability to authenticate one's work, should the instructor request it, may be sufficient grounds to initiate an academic misconduct case" (Section 20.1.2.3, Student Rule 20).

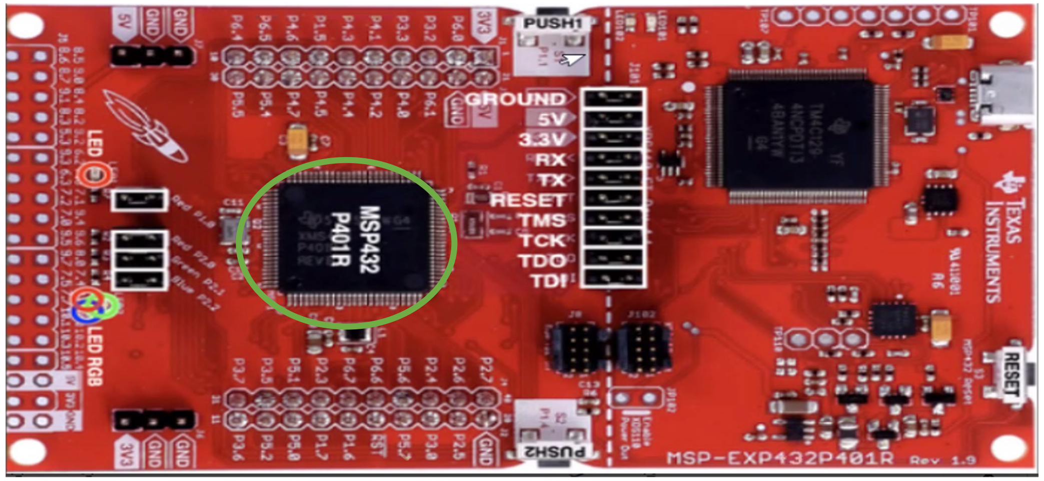
Americans with Disabilities ACT (ADA) Policy

- ❑ Texas A&M University is committed to providing equitable access to learning opportunities for all students. If you experience barriers to your education due to a disability or think you may have a disability, please contact Disability Resources in the Student Services Building or at (979) 845-1637 or visit http://disability.tamu.edu.

- ❑ Disabilities may include, but are not limited to attentional, learning, mental health, sensory, physical, or chronic health conditions. All students are encouraged to discuss their disability related needs with Disability Resources and their instructors as soon as possible.

What are we going to Learn?

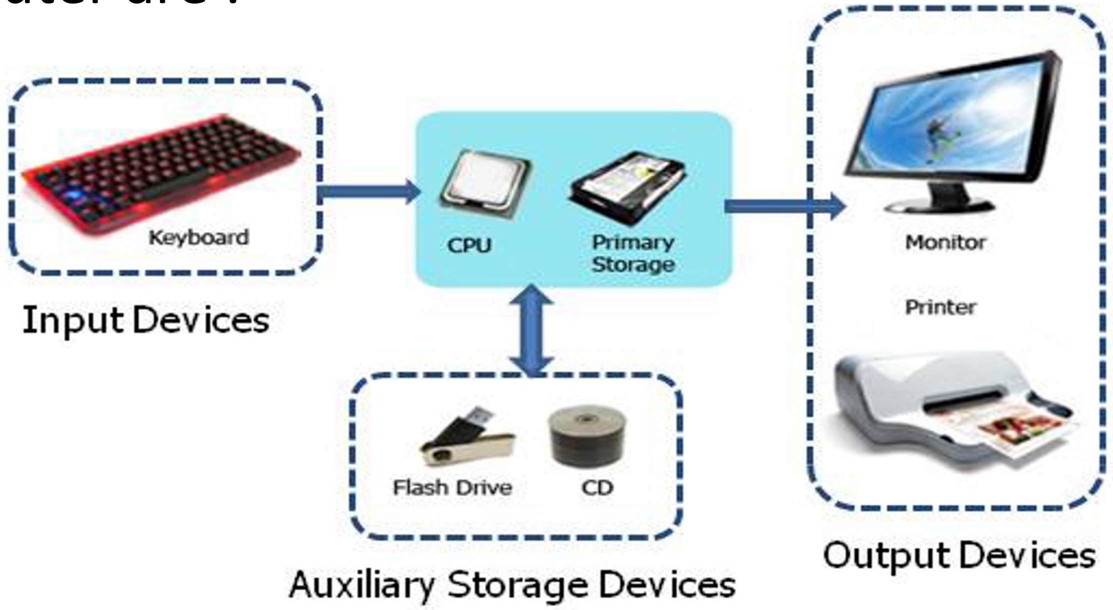
❑ **Microcontroller Architecture**



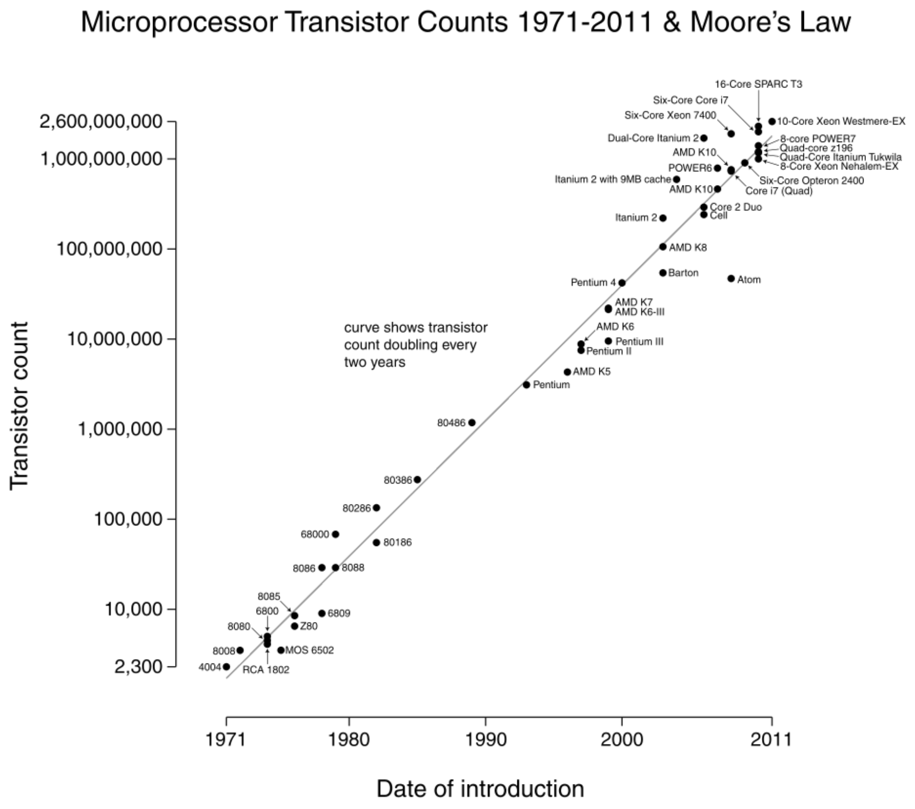
❑ Architecture of this Microcontroller (MSP432P401R)
- CPU Architecture
- Memory Components
- Port Architecture and Programming
- Peripheral Architecture and Programming
- Timer, UART, and ADC architecture and programming

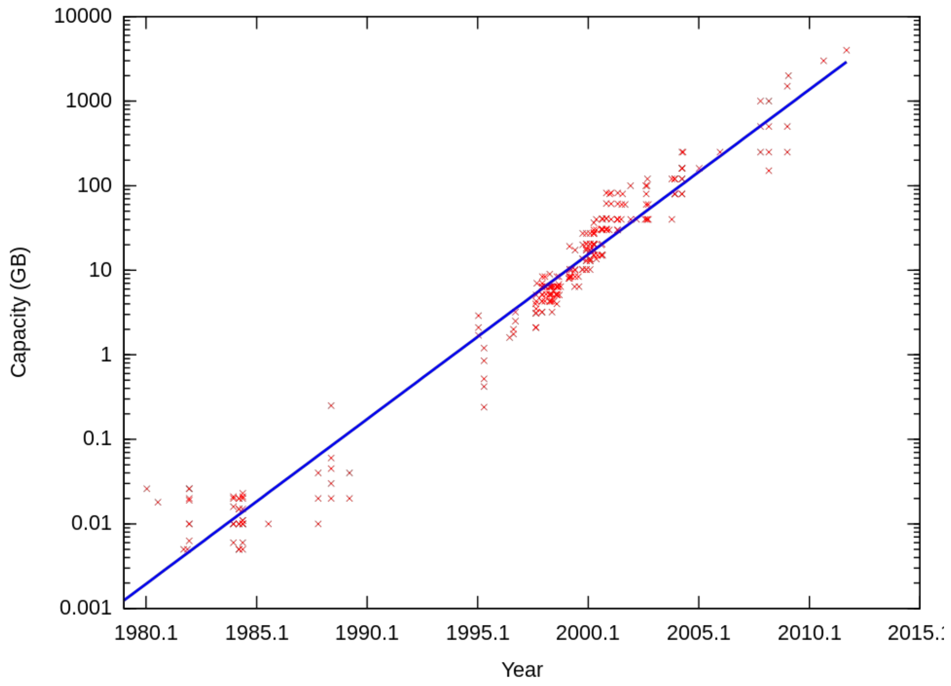
What is a Computer

- A ***computer*** is a **programmable** machine designed to sequentially and automatically carry out a sequence of arithmetic or logical operations. The particular sequence of operations can be changed readily, allowing the computer to solve more than one kind of problem.
- The basic functions of the computer are :
  - Inputting data
  - Outputting data
  - Processing data
  - Storing data



Computer Technology





Disk size increases 2x every year

Processor speed increases 2x every 18 months

Unit of Measurement in Computer System

- Normally, we use International System of Units (SI : Système international d'unités) as unit of measurement. But in computer system, we use binary prefix.

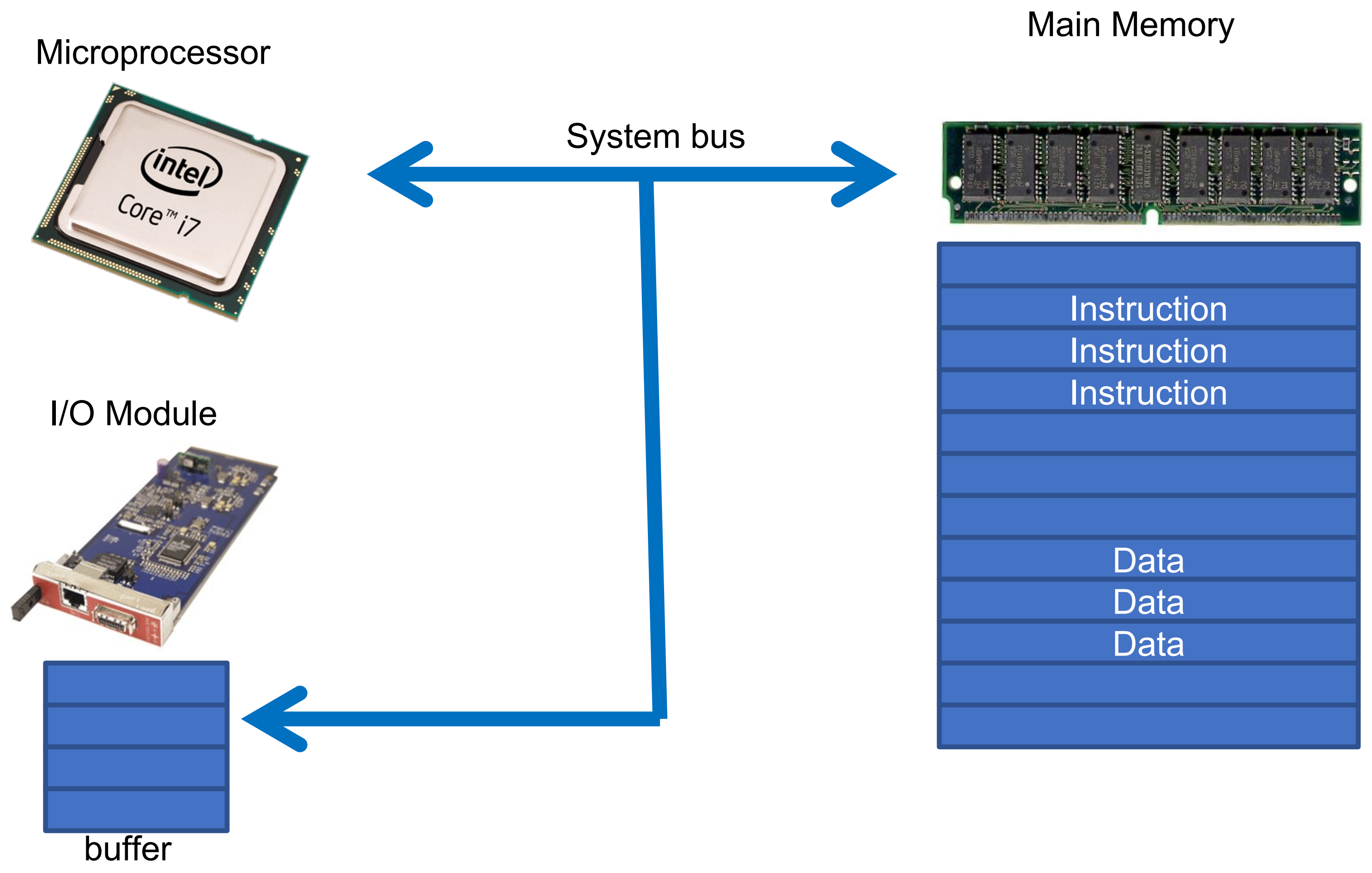
Name	Abbr	Factor	SI size
Kilo	K/k	2^{10} = 1,024 (K)	10^{3} = 1,000 (k)
Mega	M	2^{20} = 1,048,576	10^{6} = 1,000,000
Giga	G	2^{30} = 1,073,741,824	10^{9} = 1,000,000,000
Tera	T	2^{40} = 1,099,511,627,776	10^{12} = 1,000,000,000,000
Peta	P	2^{50} = 1,125,899,906,842,624	10^{15} = 1,000,000,000,000,000
Exa	E	2^{60} = 1,152,921,504,606,846,976	10^{18} = 1,000,000,000,000,000,000
Zetta	Z	2^{70} = 1,180,591,620,717,411,303,424	10^{21} = 1,000,000,000,000,000,000,000
Yotta	Y	2^{80} = 1,208,925,819,614,629,174,706,176	10^{24} = 1,000,000,000,000,000,000,000,000

Von Neumann Architecture

- The earliest computing machines had fixed programs. Changing the program of a fixed-program machine requires re-wiring, re-structuring, or re-designing the machine.
- The idea of the ***stored-program computer*** changed all that: a computer that by design includes an instruction set and can store in memory a set of instructions (a program) that details the computation.
- **Characteristics of von Neumann machine**
  - Both data and instructions are stored in a read/write memory.
  - Memory contents are addressable by location without regard for the type of data contained there
  - Execution occurs in a sequential fashion by reading consecutive instructions from memory

John Von Neumann
1948 : You insist that there is something a machine cannot do.
If you will tell me *precisely* what it is that a machine cannot do,
then I can always make a machine which will do just that!

Main Components of a Computer



What is the Microprocessor?

- Microprocessor is a very large-scale integrated circuit (VLSI) that uses the architecture of the general-purpose digital computer.
- Microprocessors are based on the von Neumann model of a stored program computer
- The stored program computer, a microprocessor’s program is stored in memory along with its data

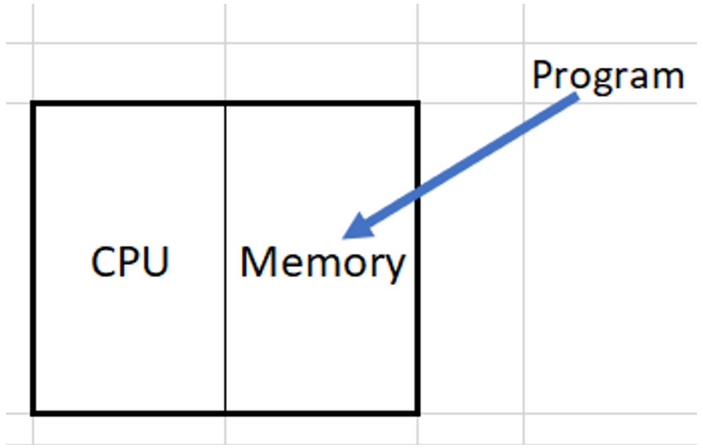
What is the difference - MPU, MCU

- **Microprocessor Units (MPU)** tend to be aimed at computer applications; they tend to have minimal "extras" on-chip.
  - Intel Pentium
  - AMD opteron

- **Microcontroller Units (MCU)** tend to be aimed at embedded control applications; they tend to consist of a processor plus a number of useful peripherals (internal I/O modules, memory, etc).
  - 8051
  - PIC
  - **ARM**

What is an Embedded System?

- A system that has CPU and Memory in a single chip and has a program loaded to it
  - An embedded system consists of CPU and Memory in a single chip as hardware component.
  - When a program is loaded to this hardware platform, the unit becomes an Embedded System, a Smart System or a Smart Device



Where are the CPU ?

Supercomputer

Personal computer

Mobile phone

High computing speed

High power consumption

Size : big

Embedded Devices

Low computing speed

Low power consumption

Size : small

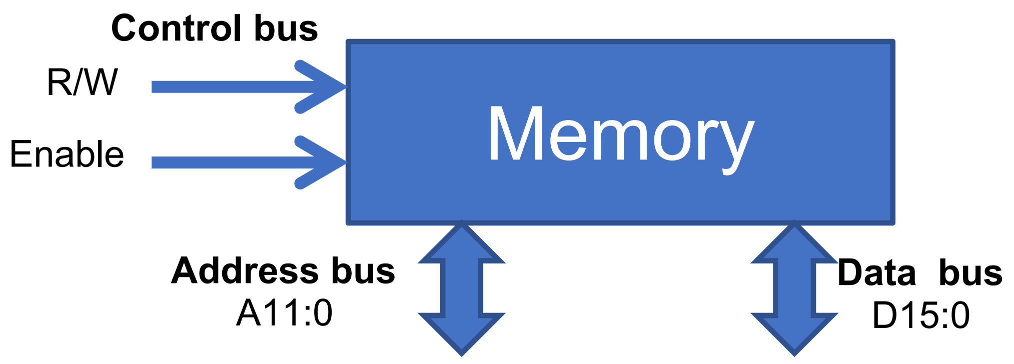
Embedded Devices

Central Processing Unit (CPU)

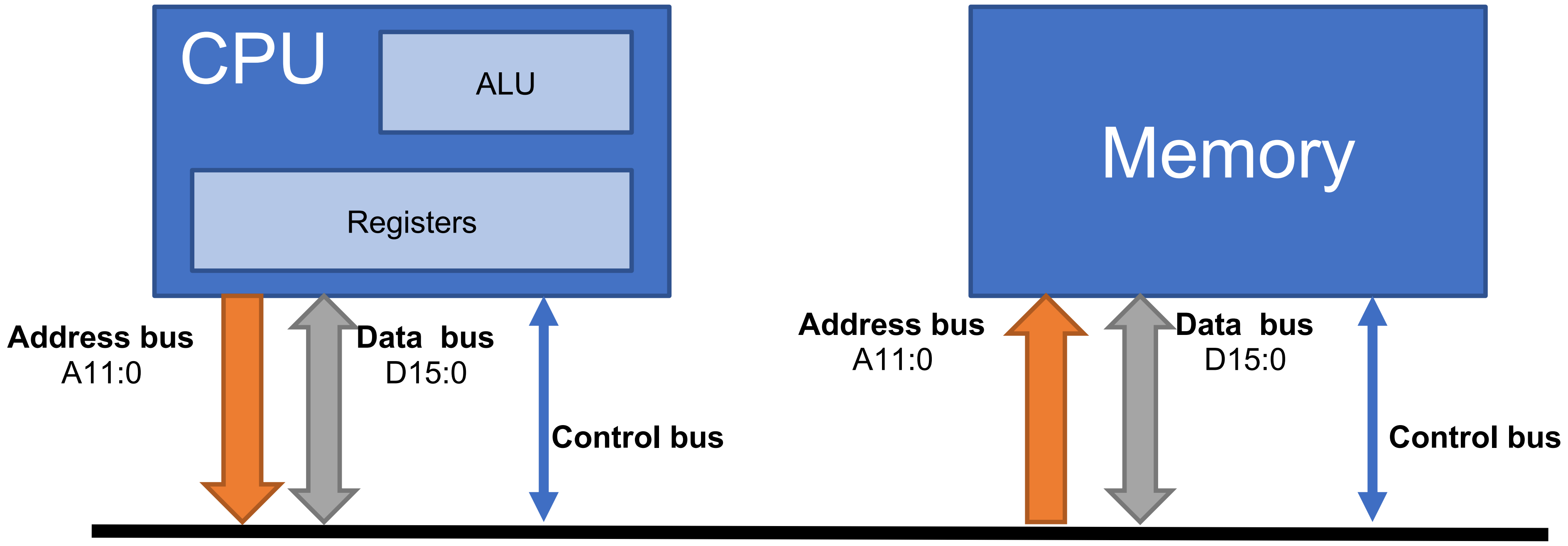
- Central Processing Unit (CPU) contains :
  - ALU (Arithmetic Logic Unit)
    - The ALU performs computational functions such as Add, Subtract, AND, OR, Compare, Increment, and Decrement.
  - Control Logic
    - The control logic decodes and executes the program. It also controls the memory, input, and output operation of the microprocessor
  - Registers :
    - The registers (fast memory, local to CPU) use to store temporary data and instructions

Memory

- The communication between microprocessor and memory has 3 types of signals (buses).
  - **Address bus** – determines the location of memory
  - **Data bus** – carries the contents of the location
  - **Control bus** – governs the information transfer
- The width of the address bus determines the size of memory. (how many location)
- The width of the data bus determines the size of content. (how many bits can each location store)



Memory, Registers, and ALU



What is a program?

A program is

- A set of statements (or instructions)
- Arranged in a specific order
- To solve a specific problem, in general

Does computer understand a program as written?

- Answer is **NO**.

- Then what to do?
- Answer is
  - All human written programs must be converted to voltage levels, a combination of high and low voltages
  - We call this as 1's and 0's.

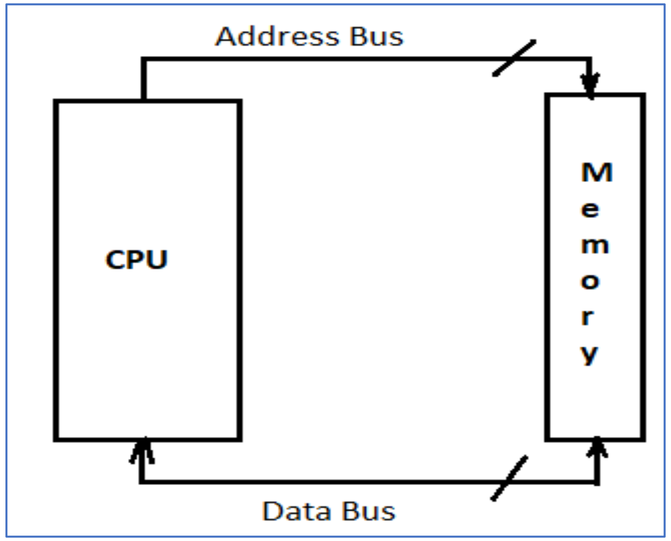
- The converted program is called an **object or machine language program**

The Instruction Cycle

- An instruction cycle is the basic operation cycle of a computer, sometimes called
  - fetch-and-execute cycle
  - fetch-decode-execute cycle

- Each computer's CPU can have different cycles based on different instruction sets, but will be similar to the following cycle:
  - **Fetch instruction** – Supply instruction address and read an instruction from memory on the data bus.
  - **Decode instruction** – Stored instruction is interpreted by CPU
  - **Fetch Operand** – Supply address of data and read data into CPU
  - **Execution instruction** - Perform the necessary action by CPU

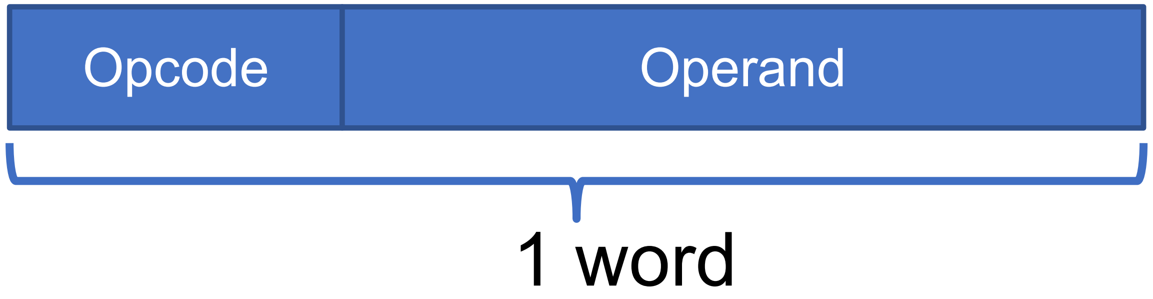
Fundamental Process (steps) of Execution of one Instruction



1. Fetch one instruction from memory. Program Counter addresses to the next instruction automatically.
2. Decode the instruction. Get the operands.
3. Execute the instruction.
4. Write result to register or to the memory
5. Start over to the next instruction, if the total of the program execution is not over.

Instructions

- Instruction usually has 2 parts
  - **Opcode** determines what is to be done
  - **Operand** specifies where/what is the data

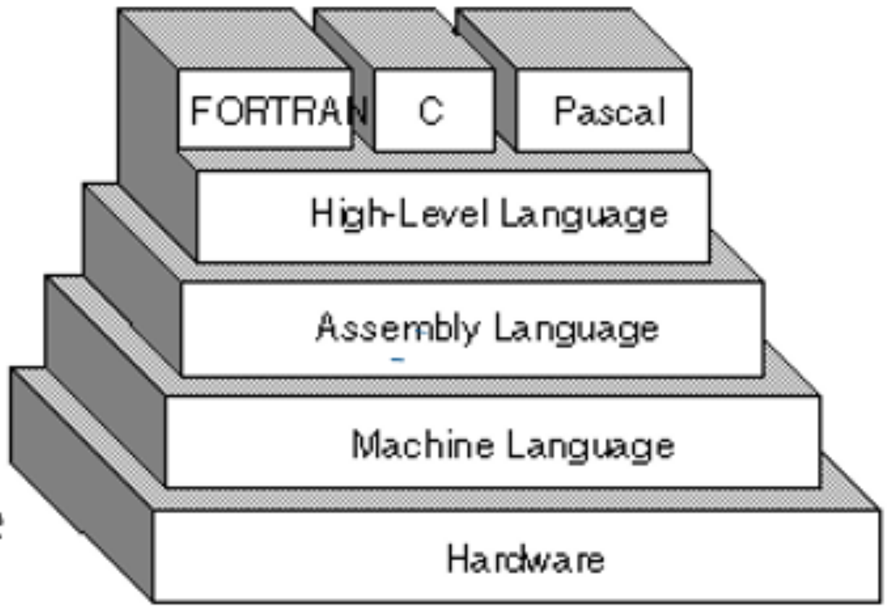


Types of Programming Languages

- ❑ High-Level Languages
- ❑ Low-Level Languages

Quick Questions:

1. What is the difference?
   - Assembly language is machine Dependent
   - High-level language is machine Independent
2. Which one is better?
   - Depends on speed of execution needed.



Distinction between and High- and Low-level languages

- **High-level language**
  - Examples: C/C++, Java, Python, Visual Basic, C#, Delphi, Perl, PHP, ECMAScript, Ruby, and many others.
  - Machine independent: program works across, processors like, intel, AMD, and operating systems
  - Always bound to run slower than Assembly language programs
  - Variables are located in the memory, with names, like,
    - x, y, myValue, rain, city, option, i, j, k, etc.
  - Each access to memory requires thousands of times compared to accessing a Register attached to the CPU

Distinction between and High- and Low-level languages

- **Low-level language:**
  - Machine depended: program written for intel architecture will not work in any other architecture.
    - Even program written for one version of intel will not work for a newer version
    - Not only that, one program written for one version of TI microcontroller may not work for another version
  - Program always runs faster than any program written in high level languages
  - Values are fed to registers as much as possible, like, r1, r3, r7 of MSP432P401R
  - Quick access to registers used in an instruction makes assembly language program run faster

Data Sizes and Instruction Sets

- The ARM is a 32-bit architecture.
- When used in relation to the ARM:
  - **Byte** means 8 bits
  - **Halfword** means 16 bits (two bytes)
  - **Word** means 32 bits (four bytes)
- Most ARM's implement two instruction sets
  - 32-bit ARM Instruction Set
  - 16-bit Thumb Instruction Set

Development Tools

- There are many development tools for ARM based processors.
- **For this course, we will use Keil µvision.**
- Keil is an ARM company.
- Keil tools are widely used.
- Keil supports different microprocessors.
- http://www.keil.com

CISC and RISC Machine

- The term "CISC" (complex instruction set computer or computing) refers to computers designed with a full set of computer instructions that were intended to provide needed capabilities in the most efficient way.

- Later, it was discovered that, by reducing the full set to only the most frequently used instructions, the computer would get more work done in a shorter amount of time for most applications. Since this was called reduced instruction set computing, RISC.