

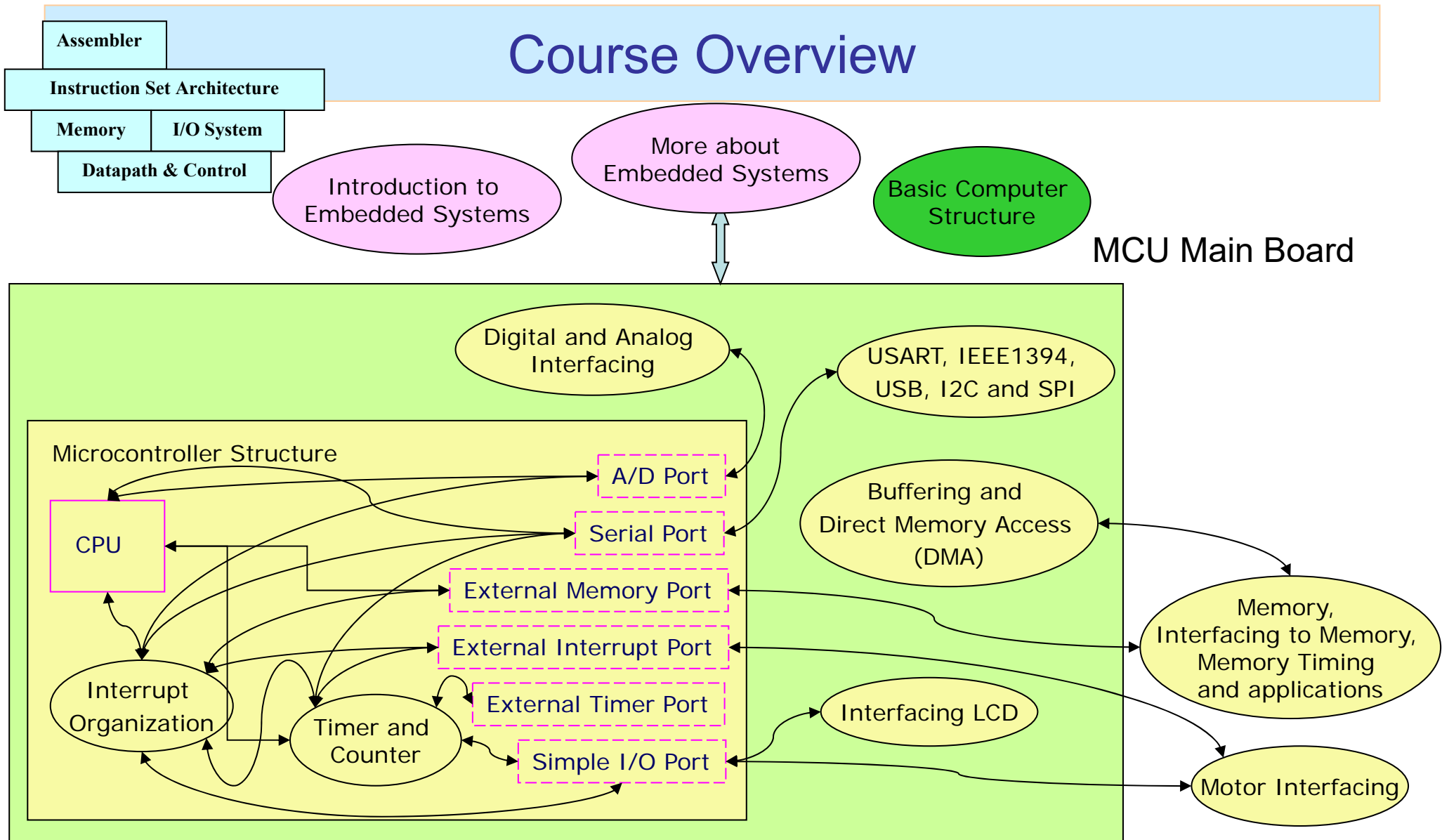
ELEC 3300

Introduction to Embedded Systems

Topic 3

Basic Computer Structure

Prof. Vinod Prasad



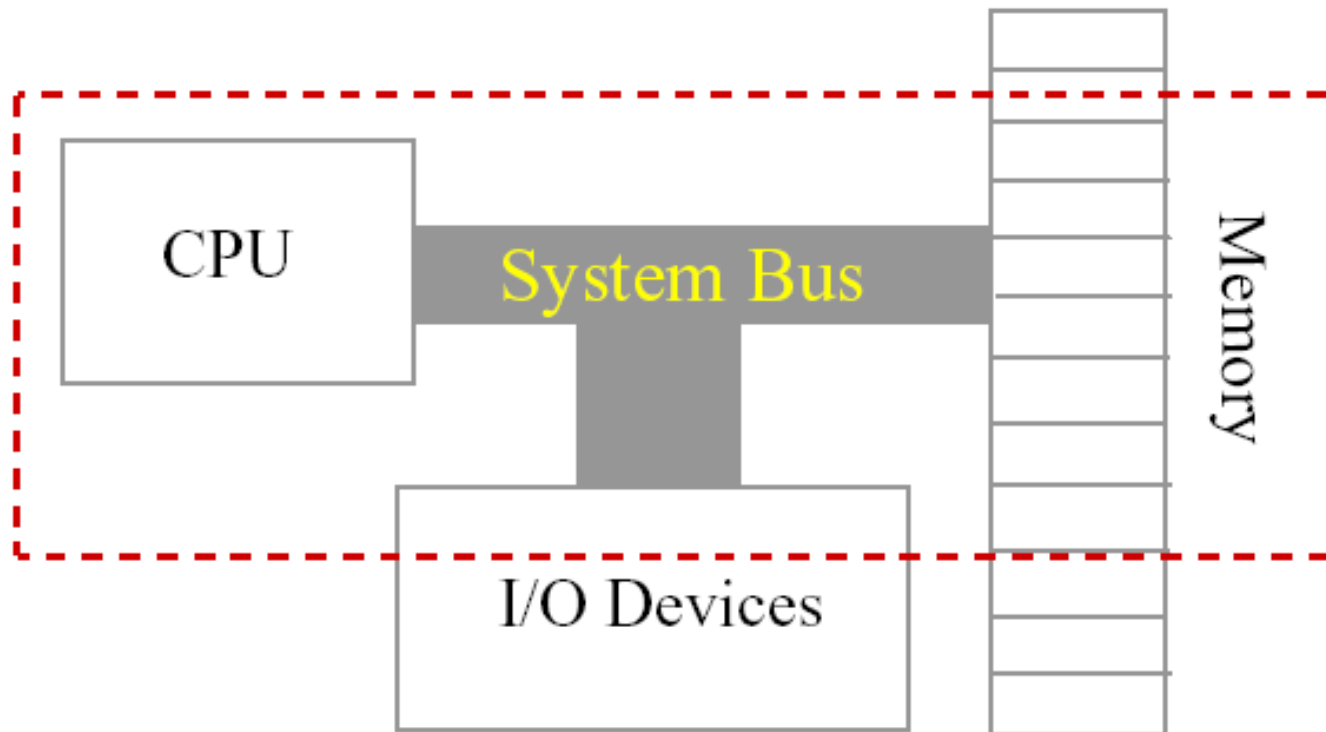
In this course, STM32 is used as a driving vehicle for delivering the concepts.

To be covered	In progress	Done
---------------	-------------	------

Expected Outcomes

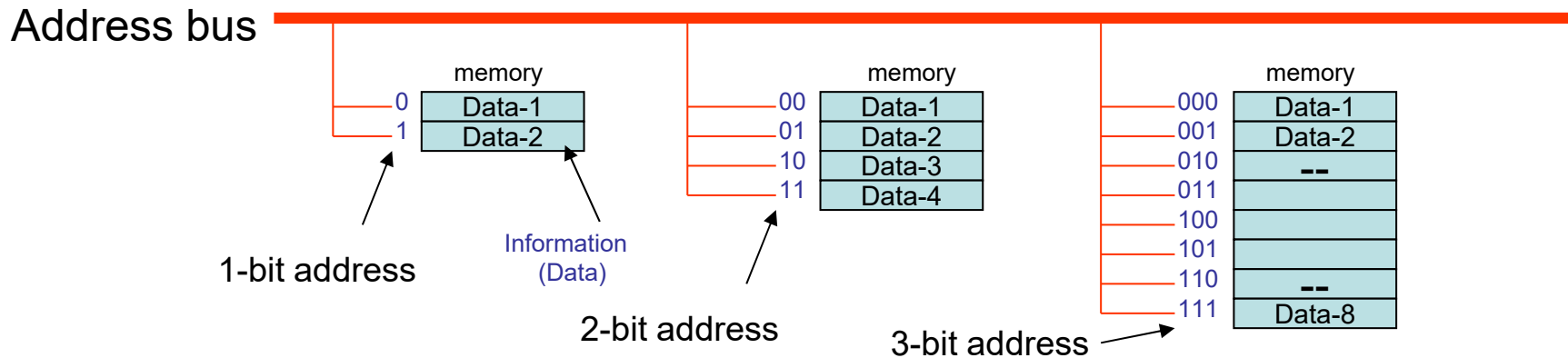
- On successful completion of this topic, you will be able to
 - Understand the basic system components and operations including
 - Data buses
 - Address buses
 - Control buses
 - Memory
 - Understand basic concepts of advanced computer architecture
 - Memory-mapped I/O
 - Bus Protocols
 - Analyze CPU Timing diagram of instruction codes

Basic System Components and Operations



The Address Bus

The address bus indicates which memory locations and I/O devices to be / being accessed.

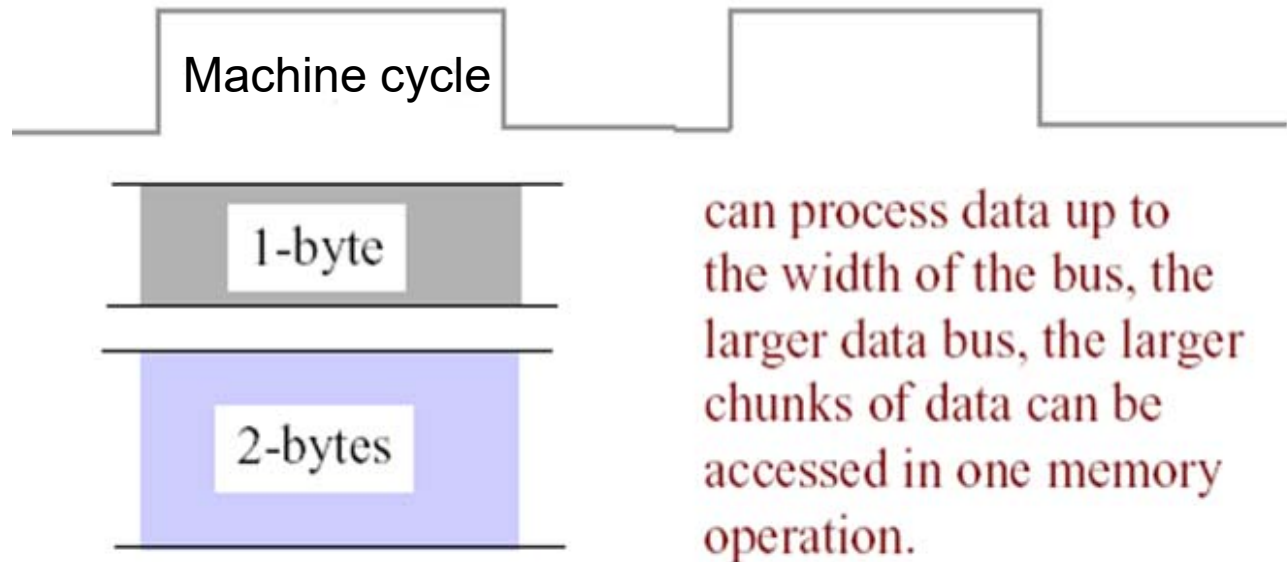


e.g. a 8088 or 8086 has 20-bit address buses, and their address spaces contain $2^{20} = 1\text{M}$ addressable memory locations

Processor	Address Bus width	Address Space	In bytes
8088, 8086	20-bit	1,048,576	1MByte
286, 386SX	24-bit	16,777,216	16MBytes
386DX, 80486 Pentium	32-bit	4,294,976,296	4Gbytes

The Data Bus

The **data bus** transfers information between a particular memory location or I/O device

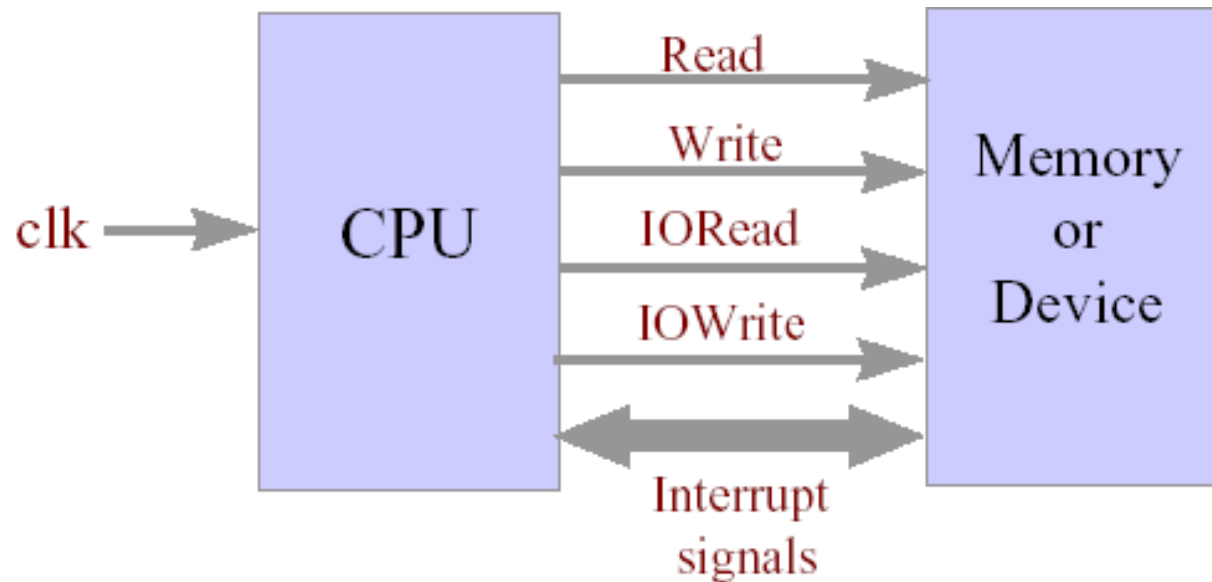


Example, a 100-MHz 32-bit bus can send 4 bytes of data to the CPU 100 million times per second.

A 66-MHz 16-bit bus can send 2 bytes of data 66 million times per second.

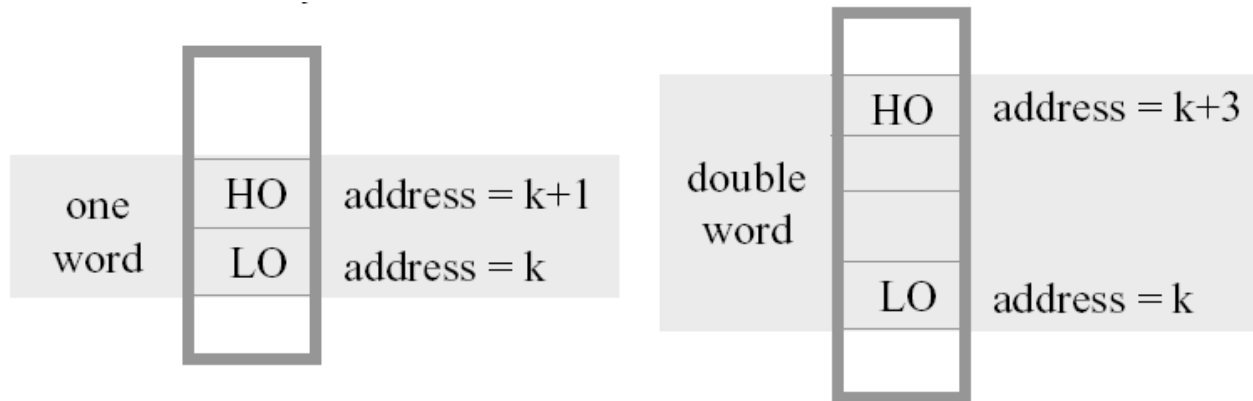
The Control Bus

- Control bus includes a set of signals controlling how the processor communicates with the rest of a system

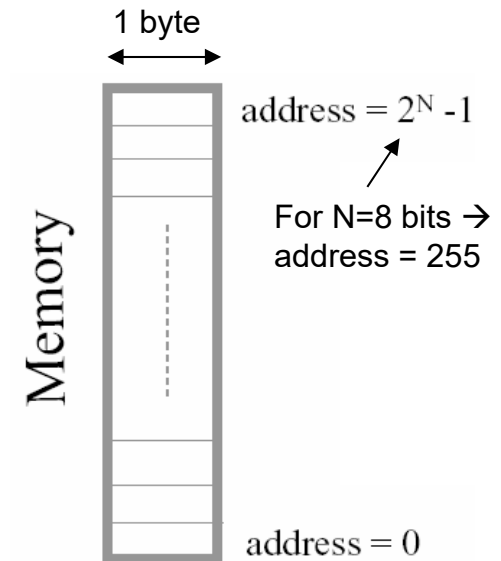


Memory Organization

- Memory can be thought of as a linear array of bytes
- Most of processors support byte-addressable memory (i.e. The basic memory unit is byte)
- Therefore, with 20, 24 and 32-bit address lines, the processors can address $2^{20} = 1\text{Mbyte}$, $2^{24} = 16\text{Mbytes}$ and $2^{32} = 4\text{Gbytes}$ of memory



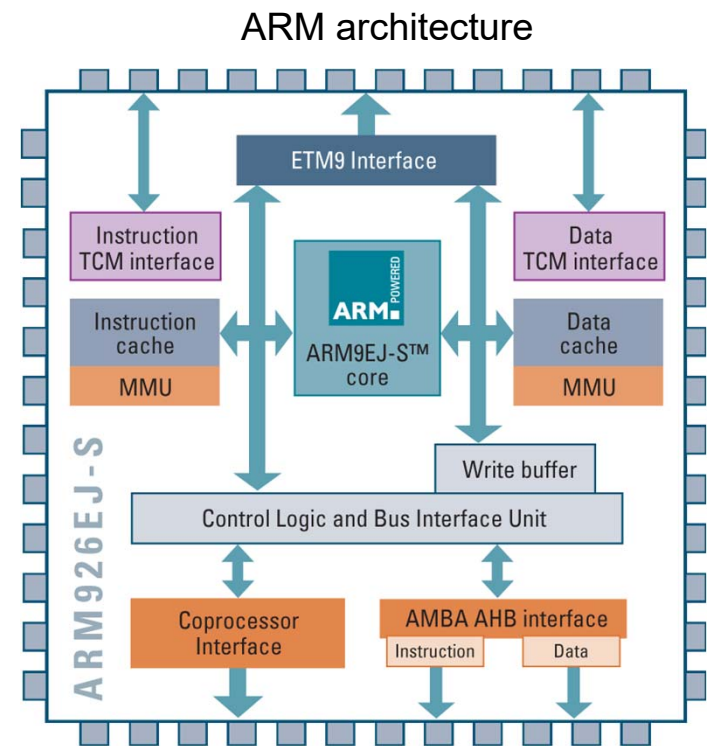
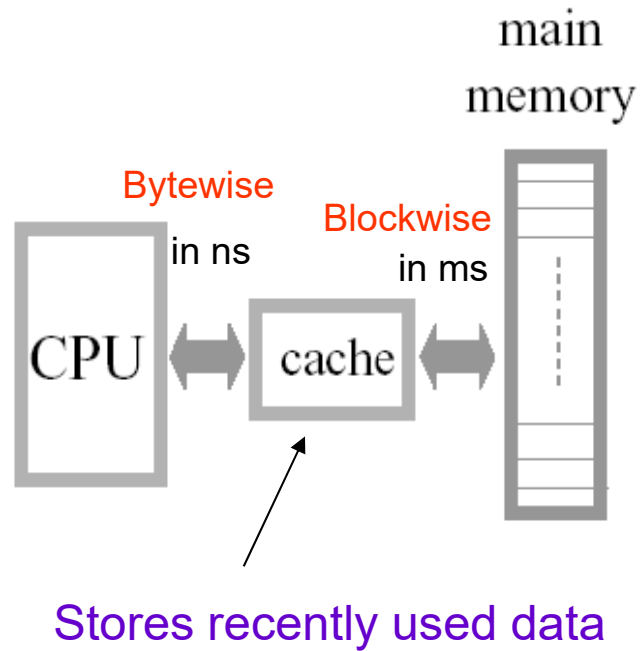
Alternative: Big-endian
Most significant value in the
sequence is stored first.



Little-endian Addressing
(e.g. Intel machine codes)

Least significant value in the
sequence is stored first.

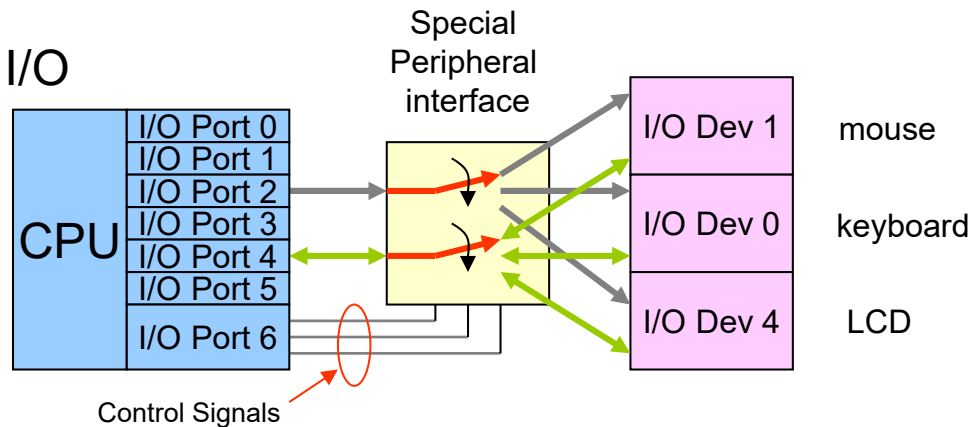
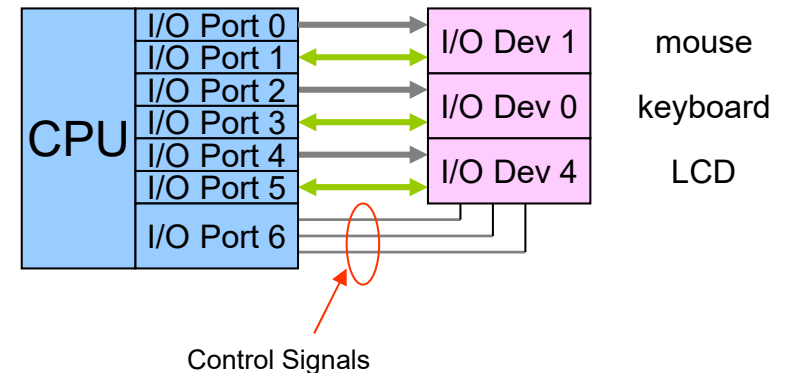
Cache Memory



Source: ARM Corporation

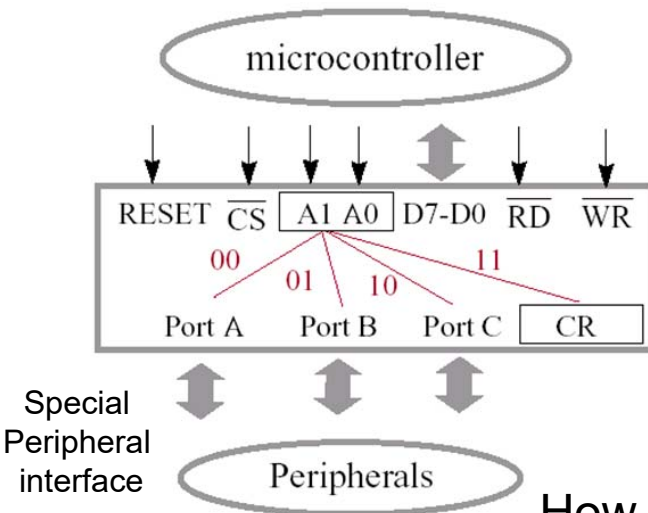
Advanced Computer Architecture: Memory-mapped I/O

- CPU needs to talk with I/O devices such as keyboard, mouse, video, disk driver, LEDs, etc.
- In design, not all the I/O devices are being accessed simultaneously. Do those devices share the I/O port of CPU and access them at different time? Can we increase the I/O capacity?
- Solution:
- Memory-mapped I/O



Tradeoff?

An Example: 8255 Programmable Peripheral Interface



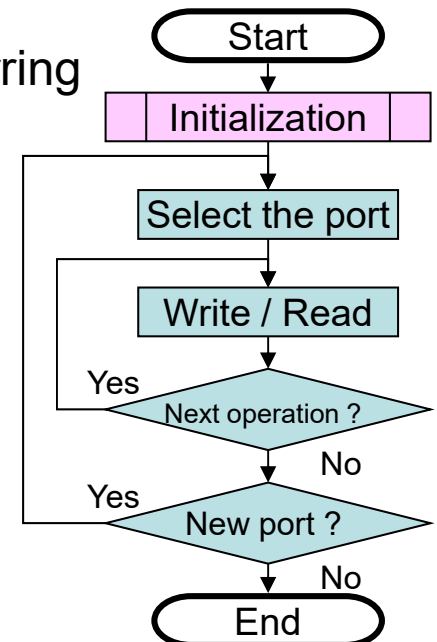
8255 occupies 4 memory locations: Port A, Port B, Port C and Control Register (CR).
Port A, Port B, Port C are configurable I/O ports
They are configured by the **Control Word** which is kept in the Control Register (CR).

A1, A0 are the selectors for Port A, Port B, Port C and CR.

How do you write the driver of 8255A for transferring data between microcontroller and peripherals?

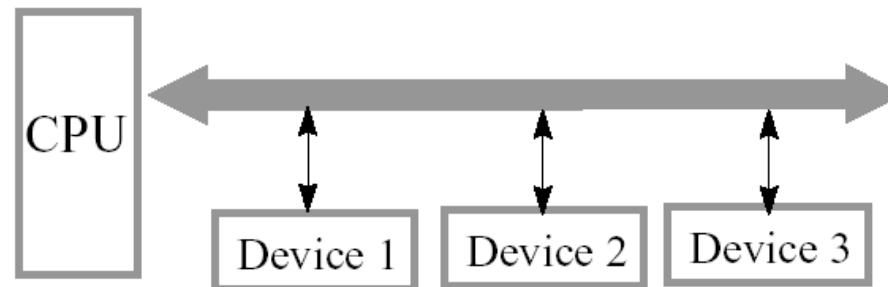
Step 1: Initialization
Configure the type of Ports
(input and/or output ports)

Step 2: Implementation
Select an appropriate port
Set a write/read operation command



Advanced Computer Architecture: Bus Protocols

- Protocol refers to the set rules agreed upon by both the bus master and bus slave



- Synchronous bus – transfers occur in relation to successive edges of a clock
- Asynchronous bus – transfer bear no particular timing relationship
- Semi-synchronous bus – operations/control initiate asynchronously but data transfer occurs synchronously

About Timing Diagram

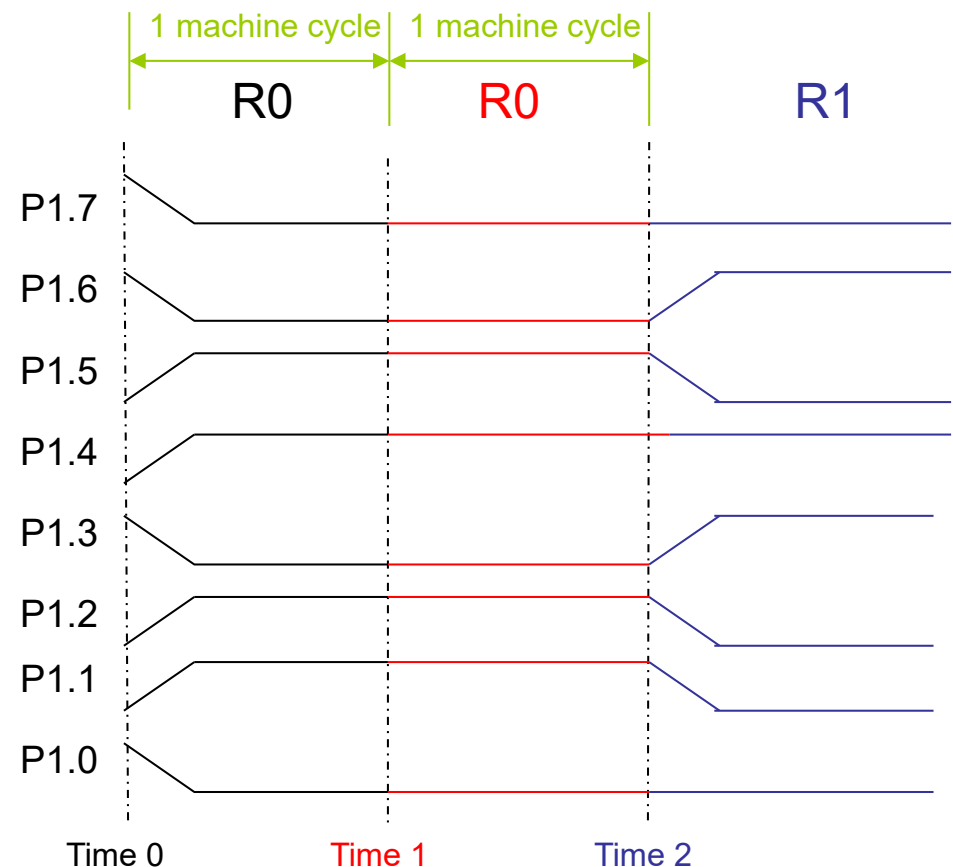
- The timing diagram represents a set of signals in the time domain. It describes how the device is being operated.
 - Example: the timing diagram of the port P1 when the following instructions are executed one-by-one.

MOV P1, R0 ; at time 0; R0 = 36H

MOV P2, R4 ; at time 1; R4 = 4AH

MOV P1, R1 ; at time 2; R1 = 58H

Assume all are 8-bit ports



About Timing Diagram

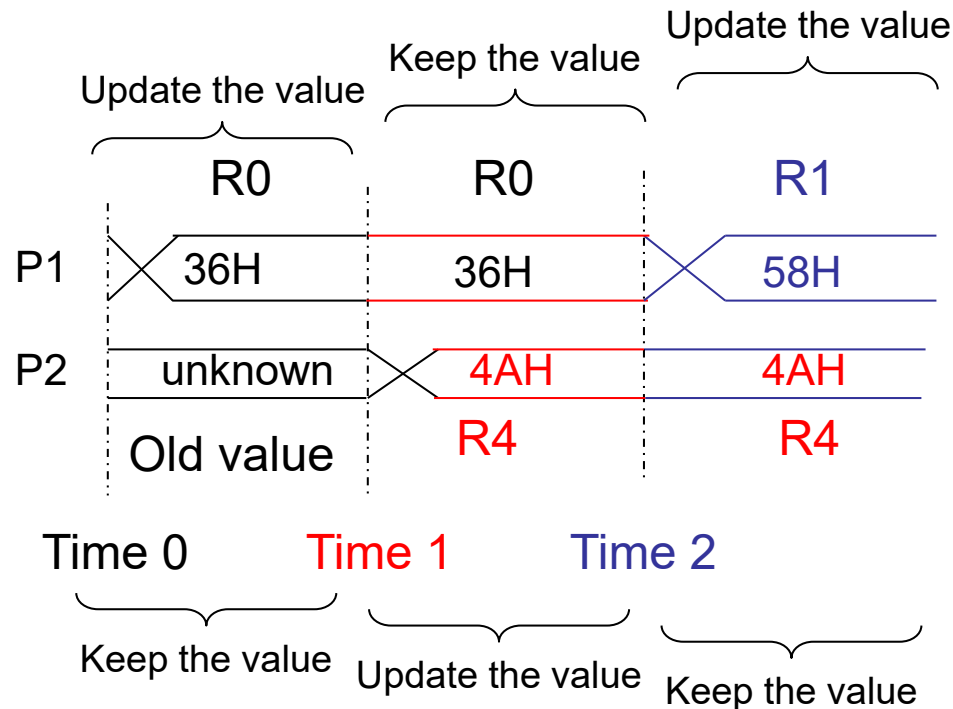
- For simplicity, it usually represents as

MOV P1, R0 ; at time 0; R0 = 36H

MOV P2, R4 ; at time 1; R4 = 4AH

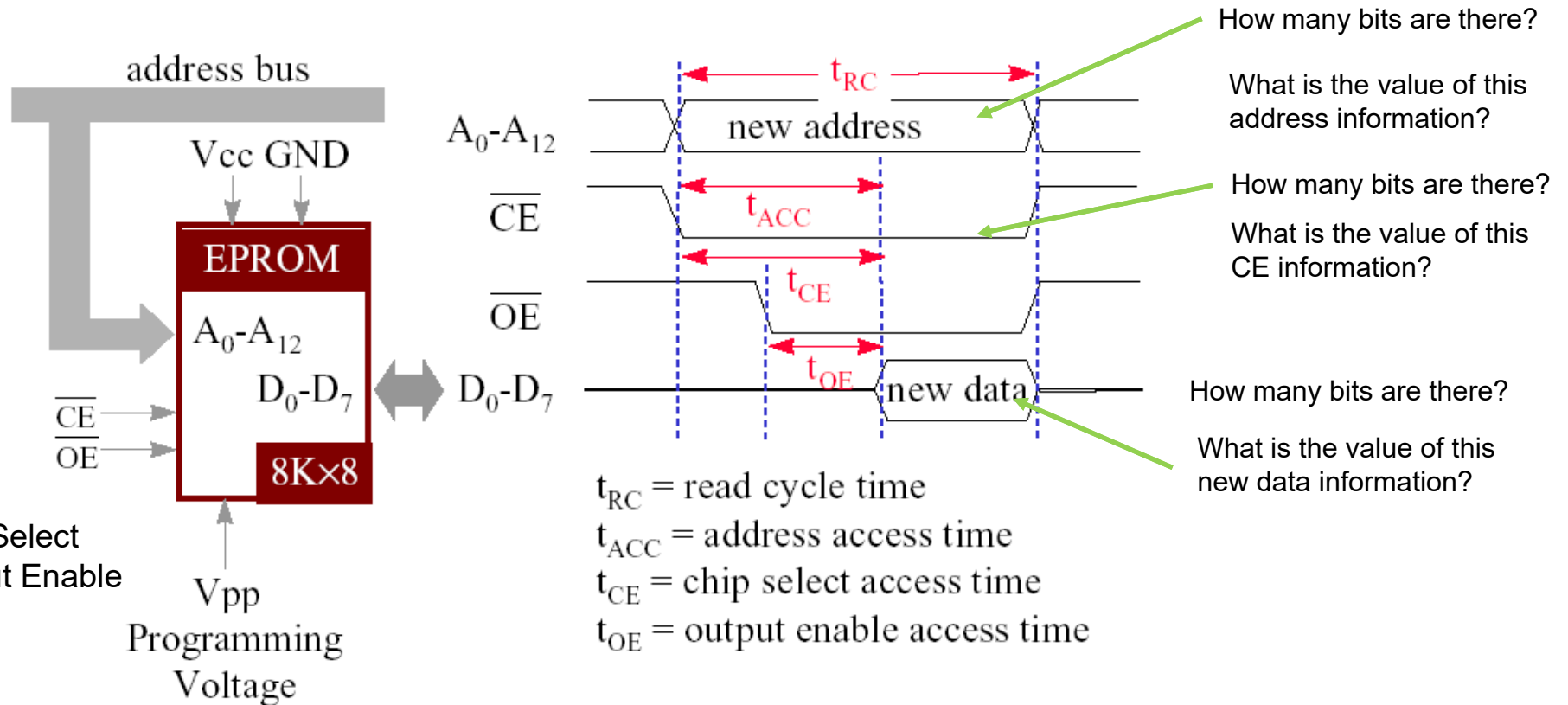
MOV P1, R1 ; at time 2; R1 = 58H

Assume all are 8-bit ports



About Timing Diagram – Data Sheet

- An example of EEPROM Timing diagram for Read Access



About Timing Diagram

Here is a timing diagram of a character type LCD device

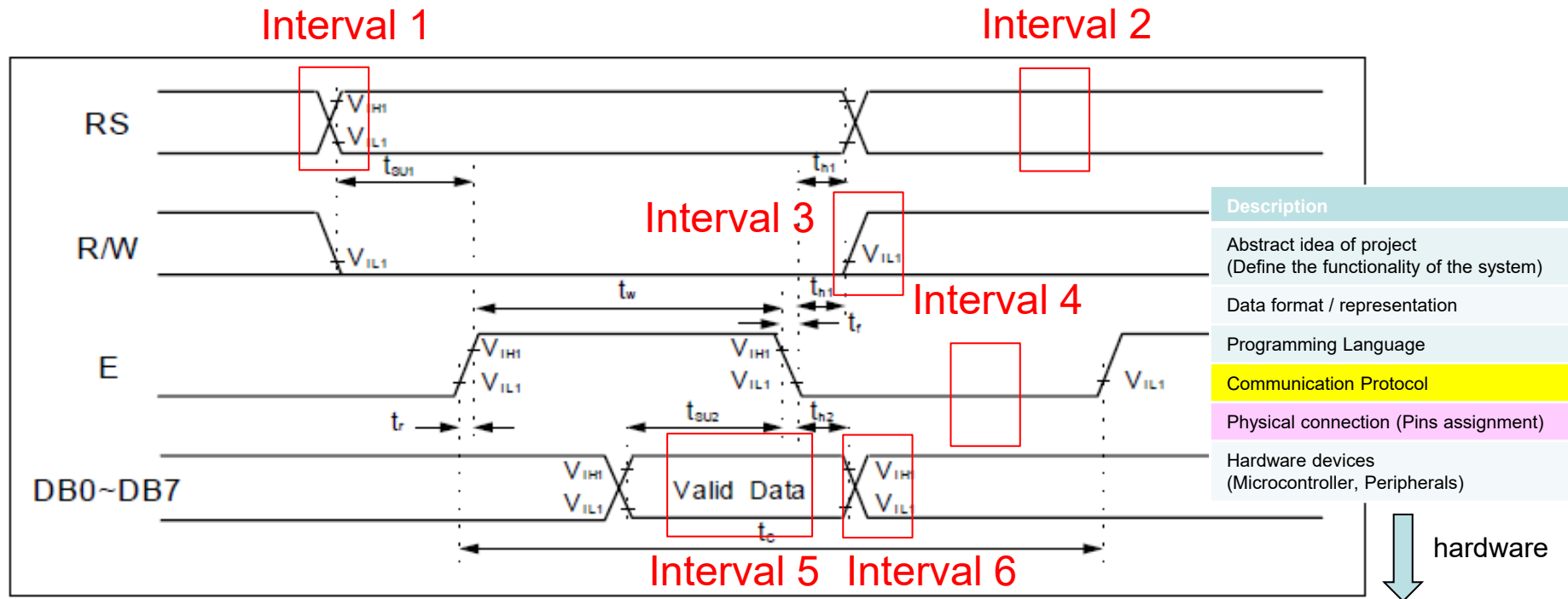
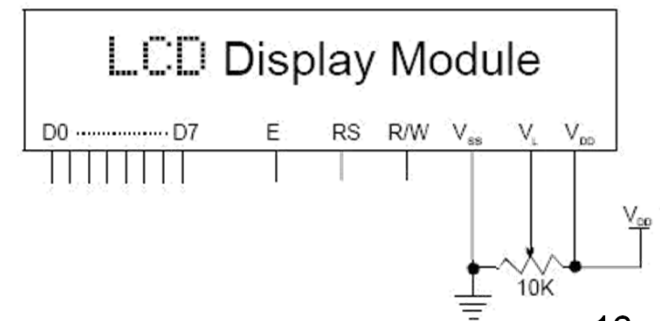
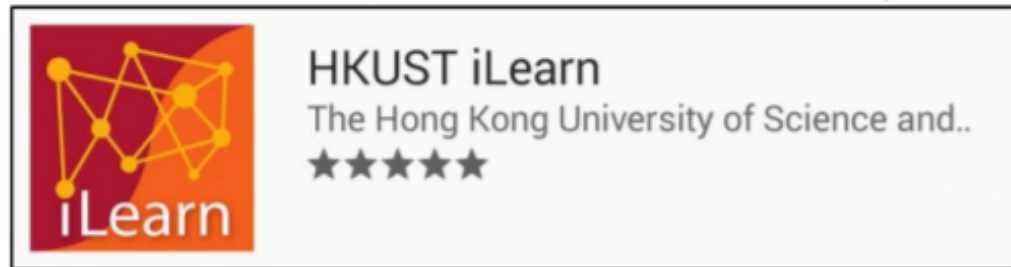


Figure 6 . Write Mode Timing Diagram

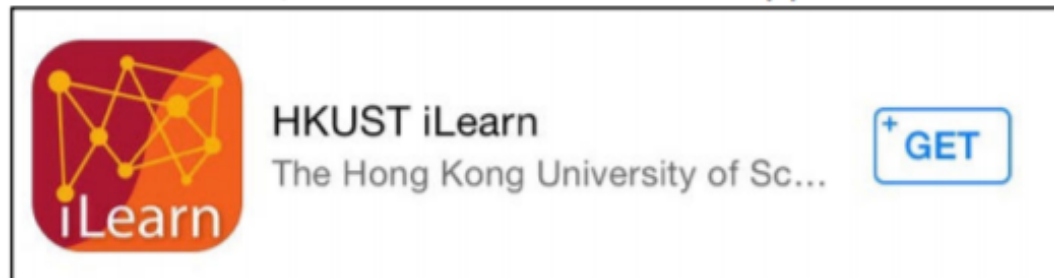


In-class activities

For Android devices, search **HKUST iLearn** at Play Store.



For iOS devices, search **HKUST iLearn** at App Store.



Lecture 3

Reflection (Self-evaluation)

- Could you
 - List out the basic system components and operations ?
 - Understand the following items and their operating principles in advanced computer architecture ?
 - Memory-mapped I/O
 - Bus Protocols
 - State the two basic steps in writing a device driver ?
 - List 64-bit computing in computer architecture?
 - Read CPU Timing diagram of instruction codes?

Course Overview

Assembler

Instruction Set Architecture

Memory

I/O System

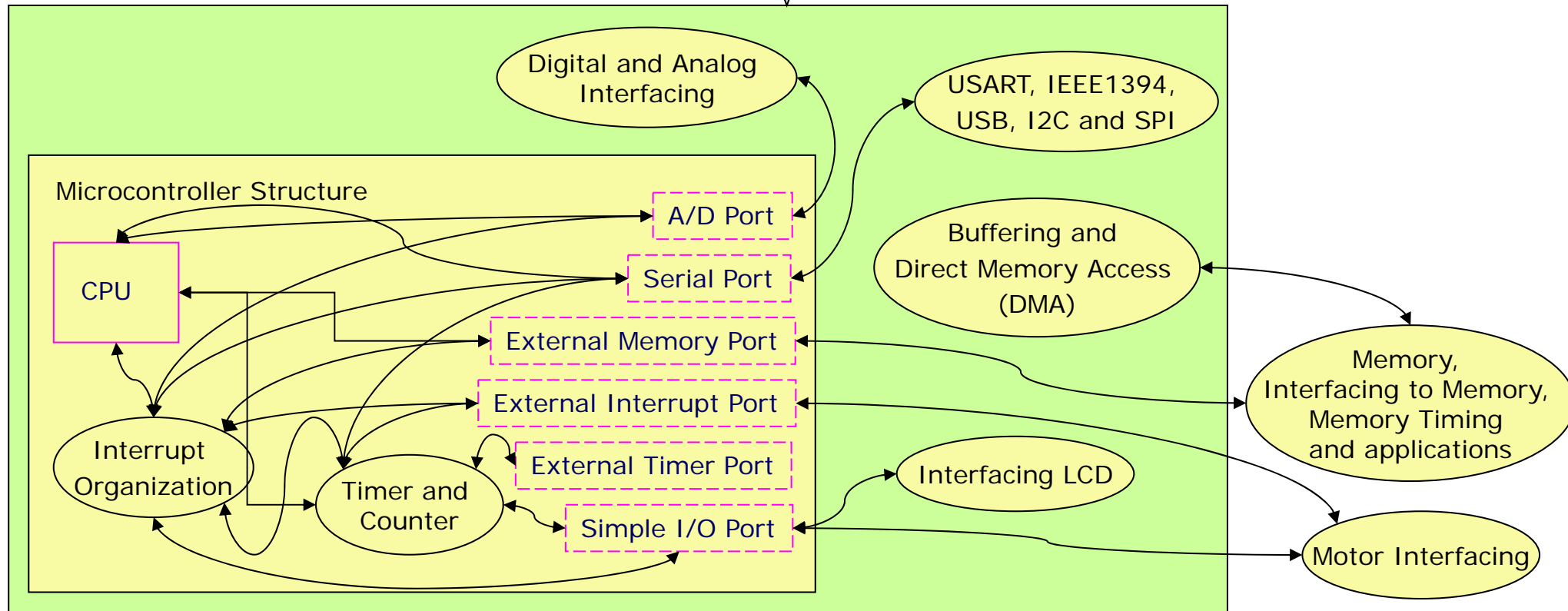
Datapath & Control

Introduction to
Embedded Systems

More about
Embedded Systems

Basic Computer
Structure

MCU Main Board



In this course, STM32 is used as a driving vehicle for delivering the concepts.

To be covered

In progress

Done