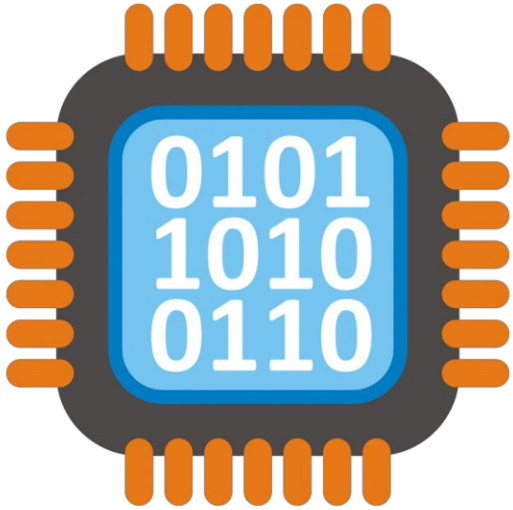
Secure Assembly Coding

Week # 1 Lectures



Dr. Qasem Abu Al-Haija,
Department of Cybersecurity,

Basic information about the course

❑ Course Name and Code:

- ✓Secure Assembly Coding – 15232

❑ Main Textbook:

- ✓Kip R. Irvine. Assembly Language for x86 Processors, 8th edition, 2019.
- ✓B. Dang, A. Gazet, E. Bachaalany. Practical Reverse Engineering: x86, x64, ARM, Windows® Kernel, Reversing Tools, and Obfuscation. John Wiley & Sons, June 2014. ISBN: 978-1-118-78731-1

❑ Instructor Information:

- ✓Name: Dr. Qasem Abu Al-Haija.
- ✓Email: q.abualhaija@psut.edu.jo
- ✓Department: Department of Cybersecurity.

Prerequisites and Grading

❑ Prerequisite Course:

- ✓ CS 11103 (Structured Programming)

❑ Prerequisite Skills:

- ✓ Number representation systems (decimal, binary, octal, hexadecimal).
- ✓ Arithmetic operations in binary, octal and hexadecimal.
- ✓ Some skills in Logic Circuits and design.
- ✓ Clocking, Chips, ICs data Sheets, and Pin Diagrams.
- ✓ Computer skills to prepare written reports and graphing, and data presentation.

❑ Grading Policy:

Assessment Tool	Expected Due Date	Weight
Midterm Exam	To be decided	30%
Class Activities	Quizzes and Assignments	30%
Final Exam	To be decided	40%

Student Responsibilities

❑ Attendance Policy

- ✓ In accordance with the University Regulations, it is the student's responsibility to be punctual and to attend all classes.

❑ Cheating and Plagiarism

- ✓ Plagiarism: Using the words, thoughts, ideas, results, etc., of another person in a written assignment, without acknowledging the source, as if it were the student's own work.

Course Regulations

- A Student is completely prohibited from doing any of the following:
  - ✓ Copying, attempting to copy, from another student's work (exams or others)
  - ✓ Permitting another student to copy from your work.
  - ✓ Using notes of whatever kind during closed book examinations.
  - ✓ Disrupting the conduct of examinations by any illegal action.
- A Student is recommended of doing the following:
  - ✓ Please use email whenever possible for your inquiries and appointments.
  - ✓ Please read the assigned materials and lecture notes before each class.
  - ✓ Class participation and interaction with instructor are very essential.
  - ✓ You are responsible for downloading and printing lecture notes or other materials

Overview of previous concepts (signal, logic design, …)

What is signals- Overview

- **Signal in electronics is:**
  - Electric current or electromagnetic field used to convey data between places.
  - The simplest form of signal is a direct current (DC) that is switched on and off.

Analog Signal

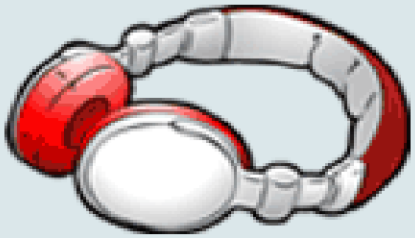
Digital Signal

- **Examples of signals**
  - Analog: Speech, Music, Photos, Videos, Sonar, Radar, Temperature calibration ...
  - Digital: digitized speech, digitized Music, digitized Images, digitized radar…

- **What is Signal Processing?**

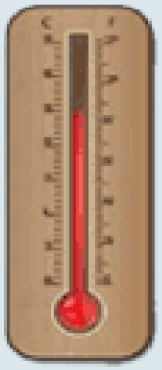
Signals - Overview

Analogue Devices



Headphones

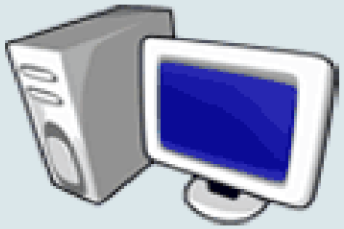
Speakers



Thermometer

Microphone

Digital Devices



Desktop PC

Digital Camera

Mobile Phone

IPad

MP3 Player

Digital Logic Design (DLP)- Quick Review

• What Is Digital Logic Design?

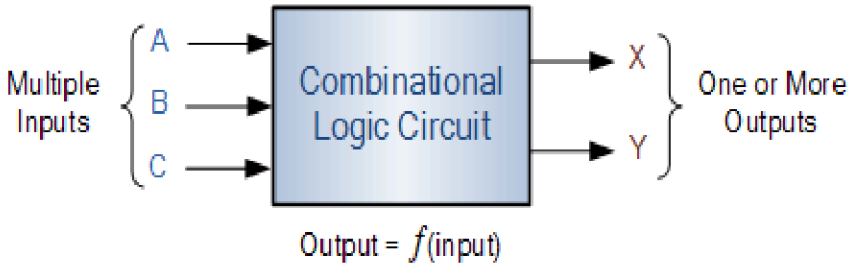
- The use of **numbers** and **codes** to produce input & output operations.
- Numbers such as: binary and Hex. Codes such as Gray, Excess-3 and BCD.
- Digital designers can develop various **simple/complex** applications.
- Hardware consists of a few simple building blocks **=> logic gates**.
- logic gates are either **two-state** (H, L) or **Tri-state** (H, L, Z).
- Gates are built using **transistors** such as **AND** gate requires 3 transistors.
- Many **Logical functions** can be composed of basic logic gates.
- **Functions** expressed as: Truth table, Boolean expression, Graphical.
- Functions can be optimized using **Boolean algebra** or **K- Maps**.
- Boolean equations can be expressed as: **SOM, POM, SOP, POS**

Digital Logic Design (DLP)- Quick Review

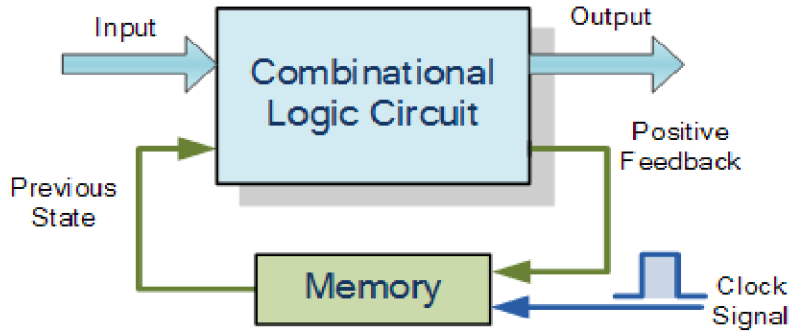
- Group of gates to accomplish a specific task => **integrated circuit-IC**
- **Integration levels** differ according to the complexity of logic circuit.
  - **SSI** (small scale integration): Introduced in late 1960s, 1-10 gates.
  - **MSI** (medium scale integration): introduced in late 1960s, 10-100 gates
  - **LSI** (large scale integration) introduced in early 1970s ,100-10,000 gates
  - **VLSI** (very LSI) Introduced in late 1970s, More than 10,000 gates.
- Digital logic design can be **combinational (CLD)** or **sequential SLD)**.
  - **CLD** composed from gates, no feedback, output depends on input only.
  - **SLD** composed from gates, flip-flops, with feedback, output depends on input and previous state.

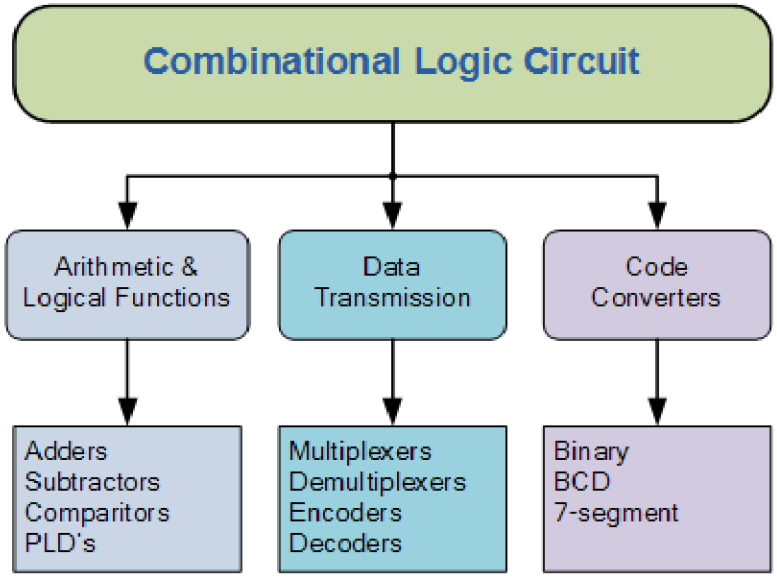
Digital Logic Design (DLP)- Quick Review

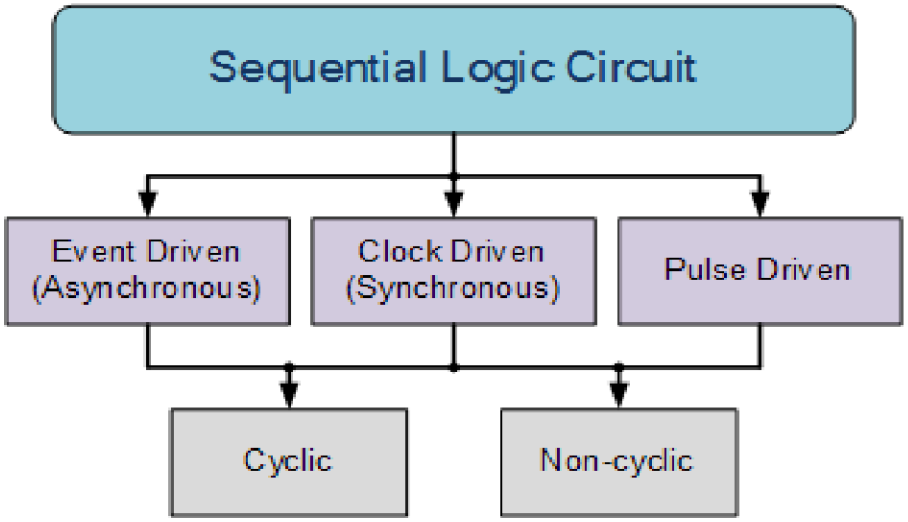
Combinational design



Sequential Design

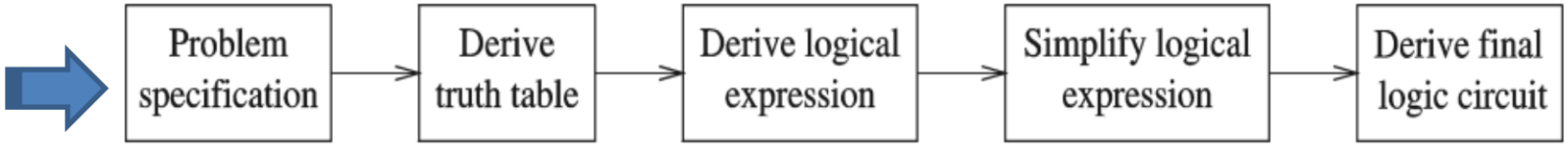






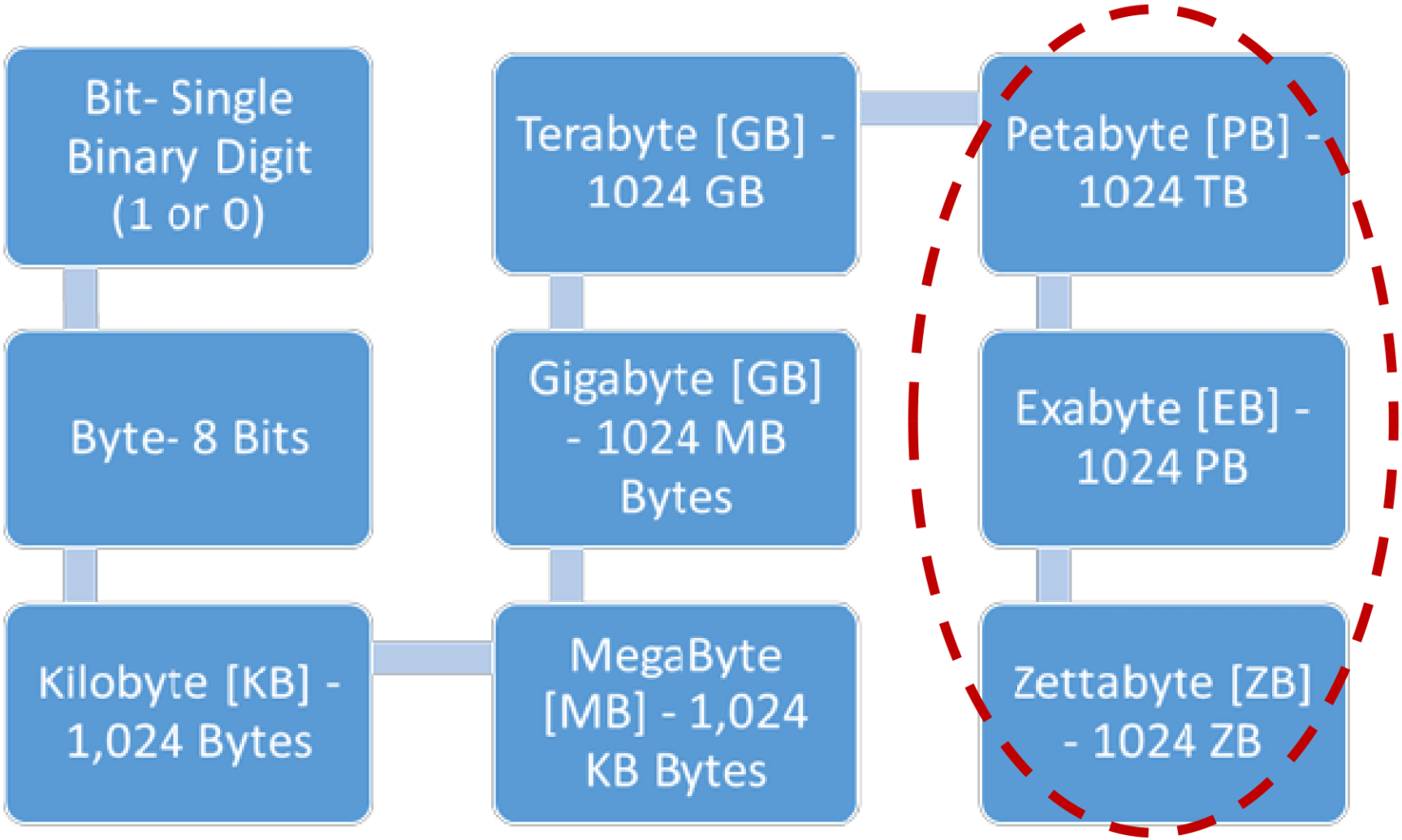
E.G. COUNTERS, REGISTERS, MEMORIES

Digital Circuit Design Steps



Digital Logic Design (DLP)- Quick Review

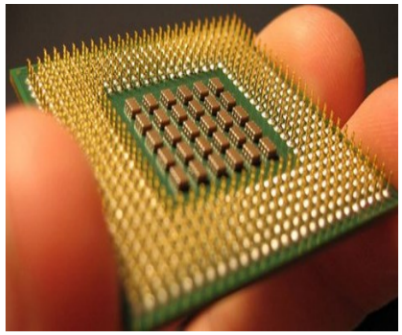
Data Storage Measurement Units



Big Data (BD)

Microprocessors (µP)- Overview

- **µP: E**lectronic chip that functions as CPU of a computer.
- **µP: H**eart of any computer system (or brain).
- **µP** based systems with limited resources => **µcomputers**.
- **µPs** are now exist in almost all electronic machines in different forms
- Any middle-class-house holds many **µPs** inside various appliances.
  - *Computer printers, automobiles, washing machines, microwave ovens, mobile phones, fax machines, satellites, etc.,*



Microprocessors (µP)- Overview

- Almost all **µPs** use the concept of **"Stored Program Execution".**
  - Instructions stored sequentially in memory locations.
  - **µP** fetches instructions one by one & executes them it in its ALU.

- A **µP** can be programmed to do any task.
  - Programmer has to know about internal resources & features of **µP**.
  - Programmer has to understand the instructions supported by the **µP**.

- Every **µP** supports its own Instructions-Set (Manufacturer-Based).
  - Mnemonic form (**Assembly code)** or binary form (**Machine code)**.
  - Assembly is converted to Machine codes vi "**Assembler".**

Assembly	Machine Code
SUB AX,BX	001010111000011
MOV CX,AX	100010111001000
MOV DX,0	10111010000000000000

Microprocessors (µP)- Overview

- Programs are written for **µP** to work with real world data.
- Also, output of **µP** must be displayed at end effect instrument.
  - We need an input interfacing & processing circuits (converters & ports)

- **µP** based system to be efficiently used, it will need:
  - A set of memory units + Interfacing circuits for inputs and outputs.

- Put all together with **µP** called as microcomputer system (**µCS**).
  - The physical components of MCS are called **Hardware.**
  - The connections between hardware component called **System Bus.**
  - The program part that makes this hardware useful is called **Software.**

Thank you