

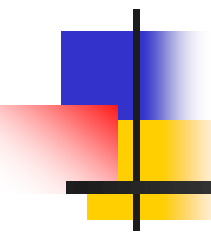


Microcontroller Experiments

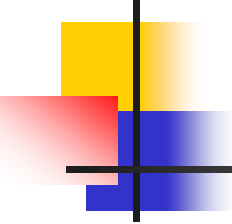
Che-Wei Chang

chewei@mail.cgu.edu.tw

Department of Computer Science and Information
Engineering, Chang Gung University

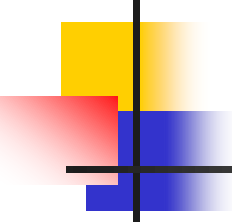


Lab 00: The Course Overview



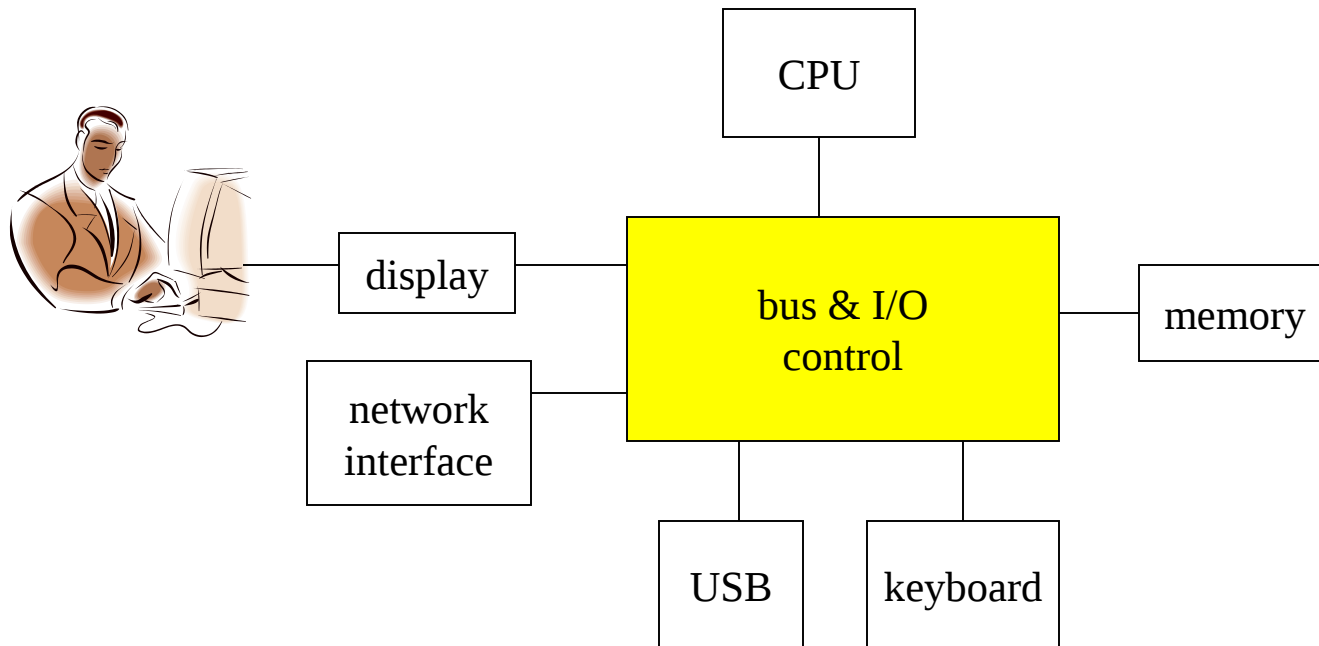
Syllabus

- Lecturer: Che-Wei Chang <chewei@mail.cgu.edu.tw>
- TAs: 劉宇恆 <m1329026@cgu.edu.tw>
蕭琦峰 <m1429025@cgu.edu.tw>
- Lecture Hours: Thursday 2:10 p.m. – 5:00 p.m.
- Office Hours: Send an email or FB message to me
- Classroom: 數位電路LAB
- Website: <https://icechewei.github.io/webpage/teaching.html>
- Grading:
 - Pre-lab reports: 20%
 - Lab demo results and reports: 40%
 - Mid-term project: 20%
 - Final term project: 20%
 - We might run a diff program to check your reports and codes: 0 for plagiarism



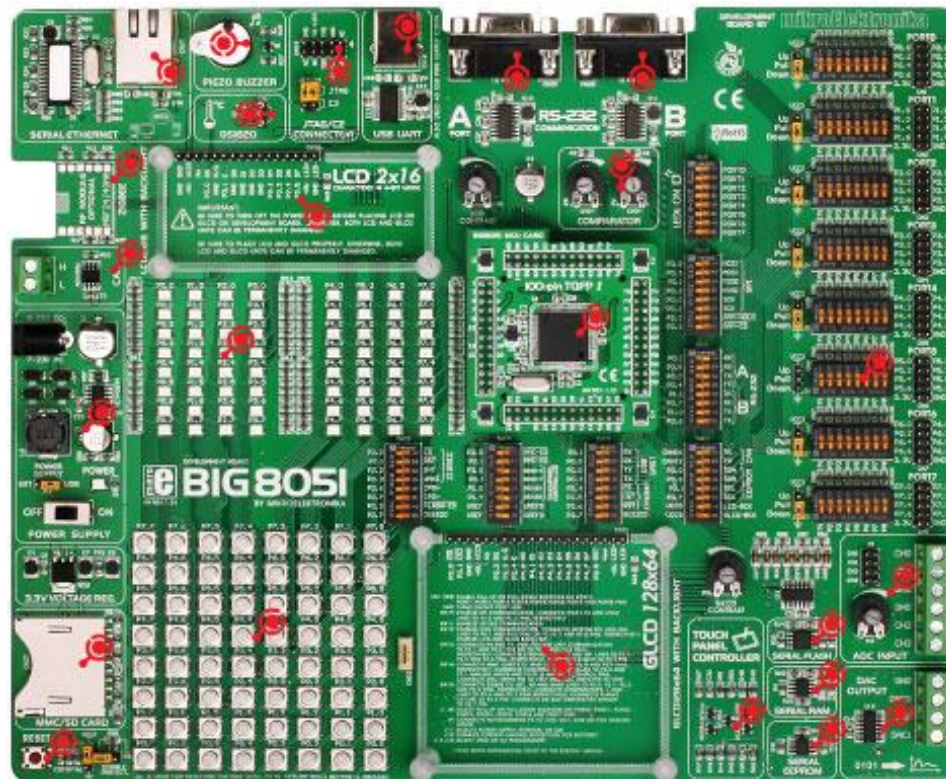
The Objective of this Course

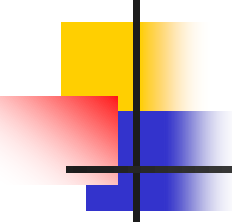
- to learn programming to control I/O devices



Experiment Platform

- The BIG8051 experiment board

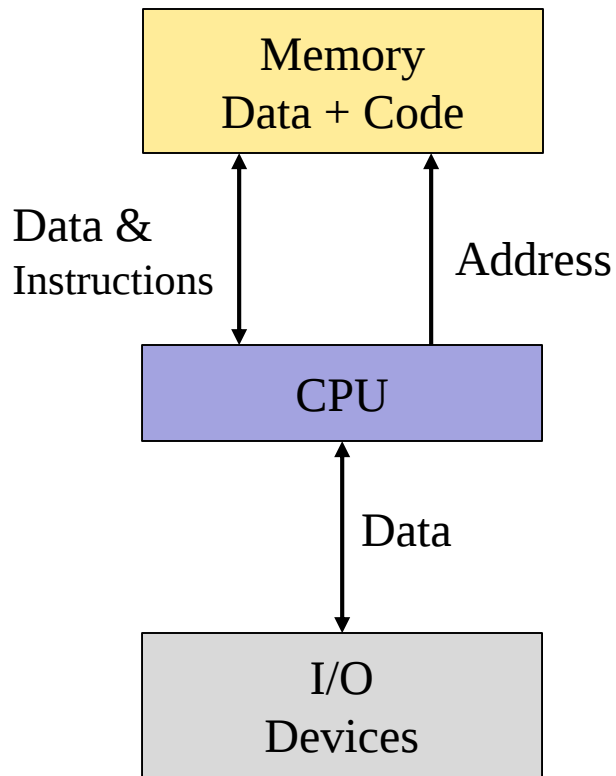




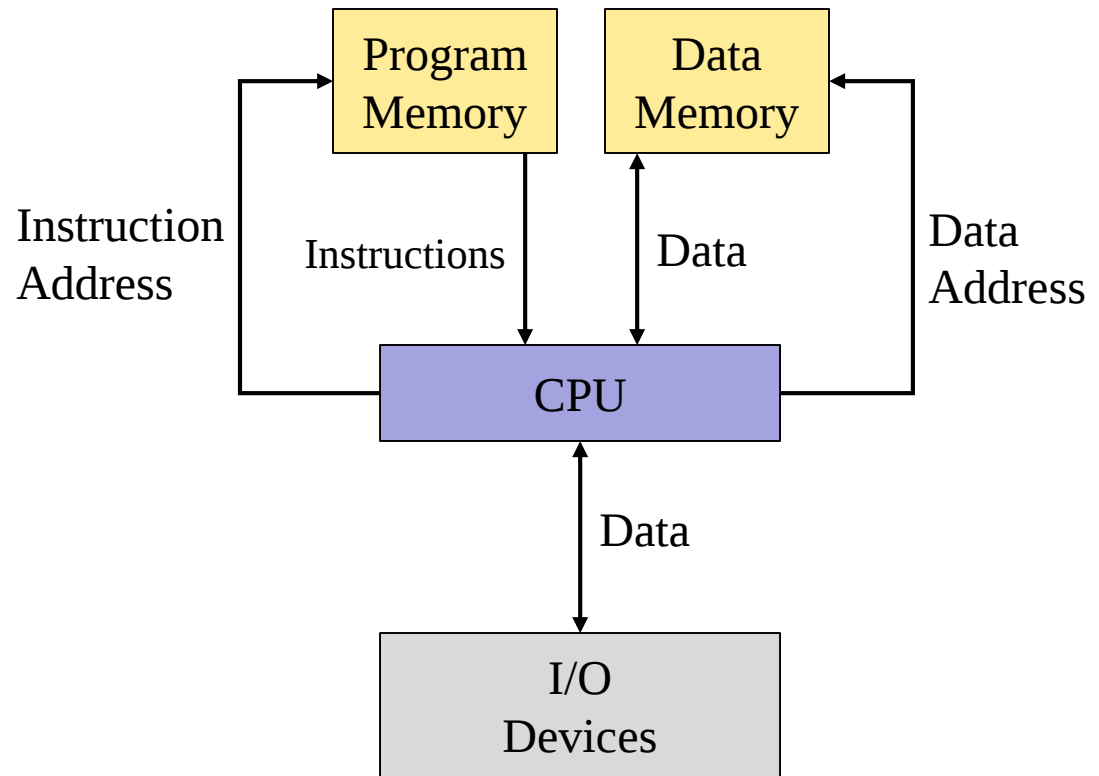
Reference

- Milan Verle, “Architecture and Programming of 8051 Microcontrollers”
 - free on-line book
 - <http://www.mikroe.com/en/books/8051book/>
- Silicon Lab C8051F04x data-sheet
 - <https://www.silabs.com/documents/public/data-sheets/C8051F04x.pdf>
- Big8051 schematic
 - <https://download.mikroe.com/documents/full-featured-boards/easy/big8051-v6/big8051-manual-v100.pdf>

Von Neumann Architecture and Harvard Architecture



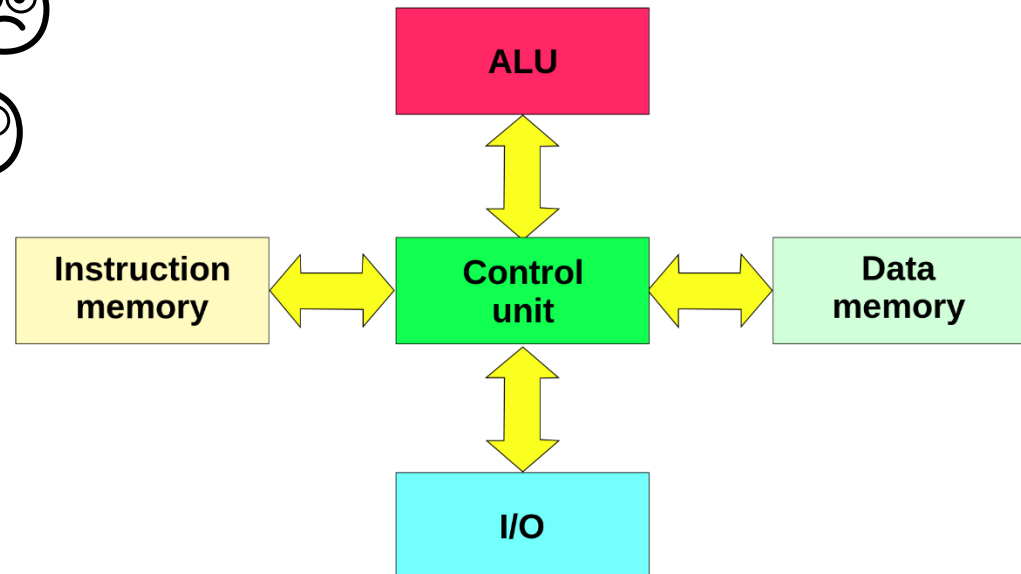
Von Neumann Architecture



Harvard Architecture

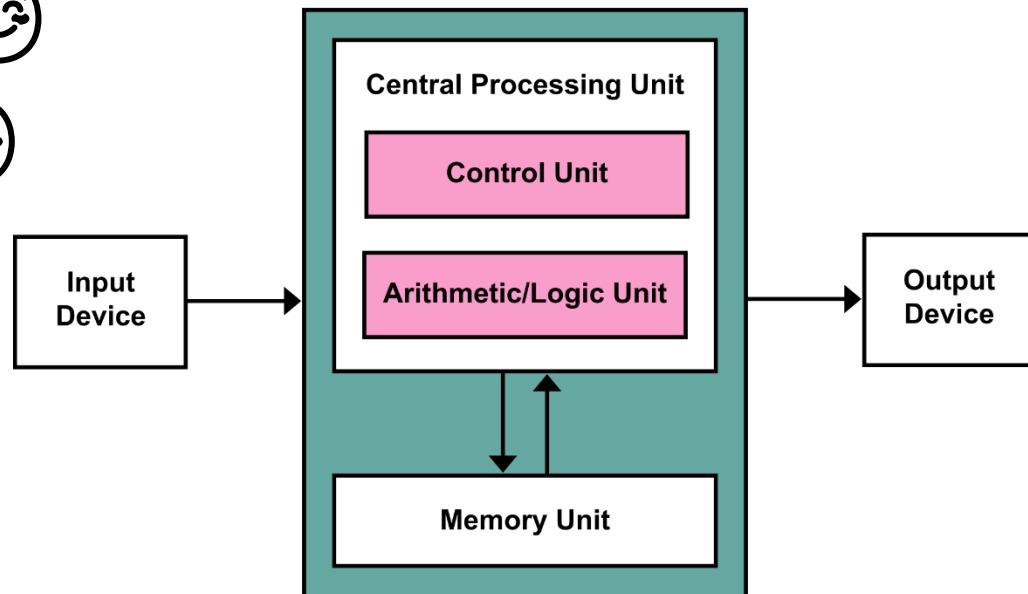
Harvard Architecture

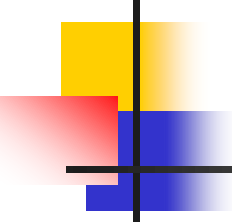
- Speed of execution: 😊
- Hardware requirements: 😞
- Space usage: 😞
- Controlling: 😞



Von Neumann Architecture

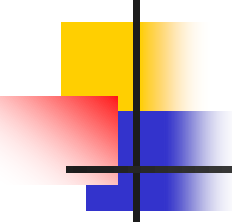
- Speed of execution: 😞
- Hardware requirements: 😊
- Space usage: 😊
- Controlling: 😊





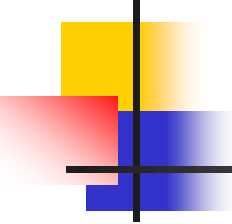
Lab00 Study Report

- File name: Bxxxxxxx-MCE-Lab0-Study
- File type: PDF only
- Reading the on-line ebook:
 - <https://www.mikroe.com/ebooks/architecture-and-programming-of-8051-mcus>
- The scope of reading
 - Form: 1.1 What are microcontrollers and what are they used for?
 - To: 2.6 Counters and Timers (including 2.6)
- The requirements of report
 - List 3 advantages of using 8051
 - List at least 3 techniques you have learned from the ebook
 - Summarize your reading
 - No more than one A4 page (one more page if you have done the bonus)
 - Grading: 80 ± 15
- Deadline: 2026/09/16 23:00 (不收遲交)
- Upload to e-learning system
- Bonus: reading the remaining part of the ebook: + 1~15



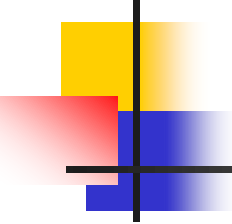
What are in 8051 (1/5)

- Read Only Memory (ROM)
 - For permanently saving the program and data
- Random Access Memory (RAM)
 - For temporary storing data and intermediate results created and used during the operations
- Electrically Erasable Programmable ROM (EEPROM)
 - For saving data permanently, but the process of programming is relatively slow



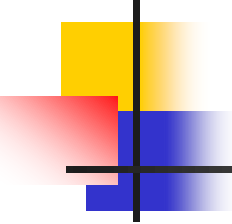
What are in 8051 (2/5)

- Special Function Register (SFR)
 - Special function registers are part of RAM memory
- Program Counter (PC)
 - PC points to the memory address containing the next instruction to execute
- Central Processor Unit (CPU)
 - Instruction Decoder
 - Arithmetical Logical Unit (ALU)
 - Accumulator



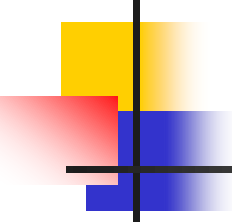
What are in 8051 (3/5)

- Input/Output Port
 - Each microcontroller has one or more registers (called a port) connected to the microcontroller pins
- Oscillator
 - An oscillator generates pulses for the cycles
 - Program instructions are not executed at the rate imposed by the oscillator itself, but several times slower



What are in 8051 (4/5)

- Timers/Counters
 - For measuring the time
 - Work with interrupts
- Watchdog Timer
 - When a watchdog timer complete its countdown, it will reset the system
 - Programs should reset the watchdog timer when the programs run normally



What are in 8051 (5/5)

- Power Supply Circuit
 - It converts electric current from a source to the target voltage and current (and frequency) to power the device
- Your Programs!
 - `mov 30h,#48`
 - Write 48 (decimal) to the register with address 30h (hexadecimal)
 - `mov R0,30h`
 - Write the number in the register with address 30h to register R0
 - `mov A,@R0`
 - Find the register with the address stored in R0 and write the number in the register to register A