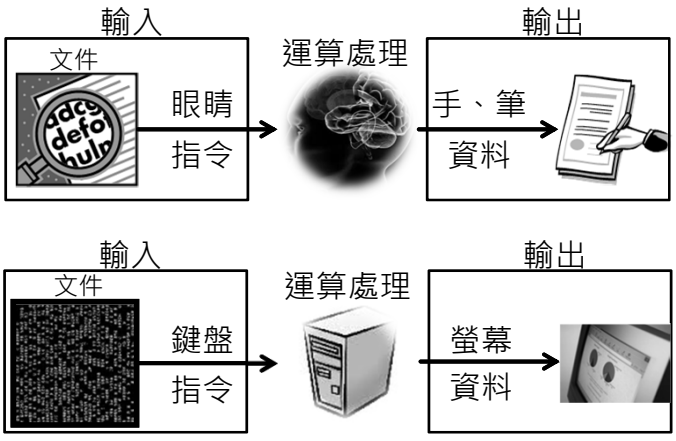
# 作業系統
# Operating Systems

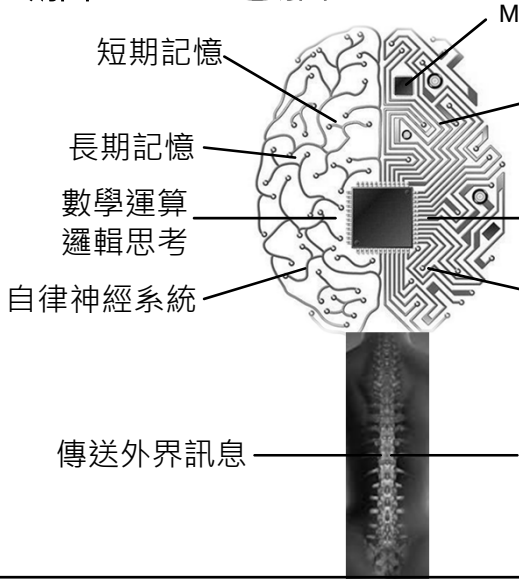
大綱-
資料處理模式
人腦 vs. 電腦

靜宜大學資訊傳播工程學系
劉國有 助理教授
kyliu@pu.edu.tw

## 資料處理模式



# 人腦 vs. 電腦




# 電腦之父-馮紐曼

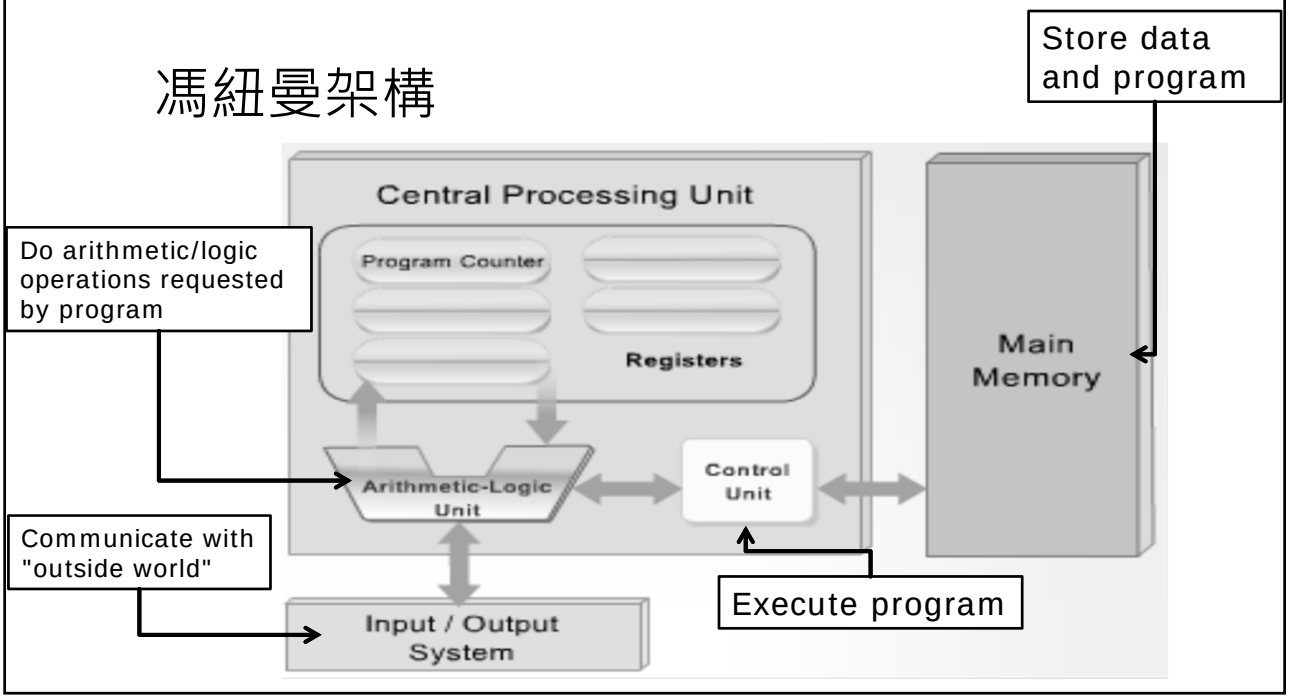
- 1945年-世界上第一部通用電子電腦：Electronic Numerical Integrator And Computer (ENIAC, 電子數值積分計算機)
  - 由賓州大學莫奇來（Mauchly）博士和他的學生愛克特（Eckert）設計以真空管為元件所設計的電腦，目的是用來計算砲彈彈道。
- 1945年-馮紐曼架構：儲存程式邏輯架構。為現代各式電腦所遵循的主要設計架構。

約翰·馮·諾伊曼
John von Neumann

20世紀40年代的馮·諾伊曼

| | |
|---|---|
| 出生 | 1903年12月28日<br>奧匈帝國布達佩斯 |
| 逝世 | 1957年2月8日（53歲）<br>美國華盛頓沃爾特·里德陸軍醫院 |
| 居住地 | 美國 |
| 國籍 | 匈牙利人、美國人 |
| 研究領域 | 數學、電腦科學、經濟學 |
| 任職於 | 柏林大學<br>普林斯頓大學<br>普林斯頓高等研究院<br>洛斯阿拉莫斯國家實驗室 |
| 母校 | 帕茨馬尼·彼得大學<br>蘇黎世聯邦理工學院 |
| 博士導師 | 利波特·費耶爾 |
| 博士學生 | 唐納德·B·吉利斯<br>以色列·霍爾珀林 |
| 其他著名學生 | 保羅·豪爾莫什<br>克利福德·休·賽克爾 |
| 獲獎 | 博修紀念獎 (1938)<br>恩里科·費米獎 (1956) |

## 馮紐曼架構



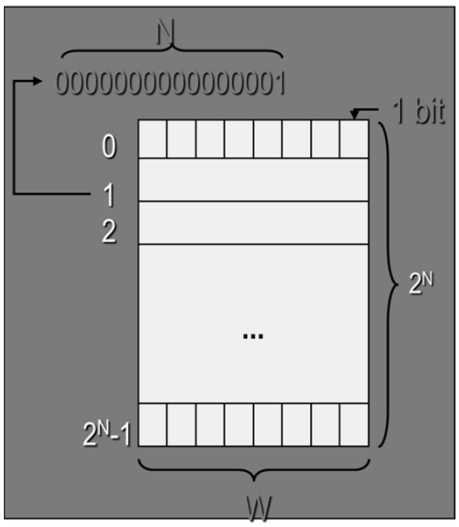

# Memory Subsystem

- Memory, also called RAM (Random Access Memory),
  - Consists of many memory cells (storage units) of a fixed size.
    Each cell has an address associated with it: 0, 1, …
  - All accesses to memory are to a specified address.
    A cell is the minimum unit of access (fetch/store a complete cell).
  - The time it takes to fetch/store a cell is the same for all cells.
- When the computer is running, both
  - Program
  - Data (variables)

  are stored in the memory.

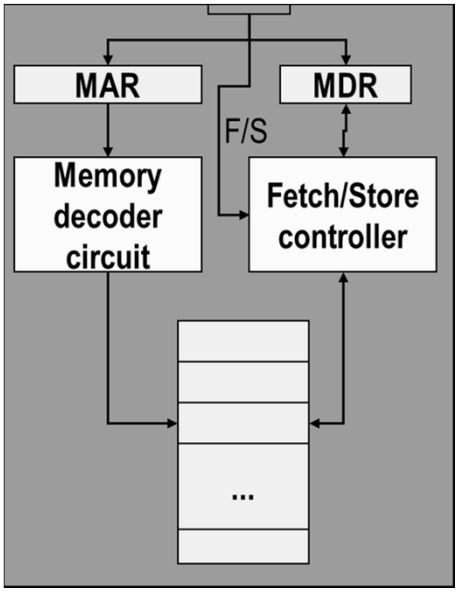
# RAM

- Need to distinguish between
  - the address of a memory cell and the content of a memory cell
- Memory width (W):
  - How many bits is each memory cell, typically one byte (=8 bits)
- Address width (N):
  - How many bits used to represent each address, determines the maximum memory size = address space
  - If address width is N-bits, then address space is $2^N$ (0,1,...,$2^N$-1)




# Structure of the Memory Subsystem

- Fetch(address)
  - Load address into MAR.
  - Decode the address in MAR.
  - Copy the content of memory cell with specified address into MDR.
- Store(address, value)
  - Load the address into MAR.
  - Load the value into MDR.
  - Decode the address in MAR
  - Copy the content of MDR into memory cell with the specified address.



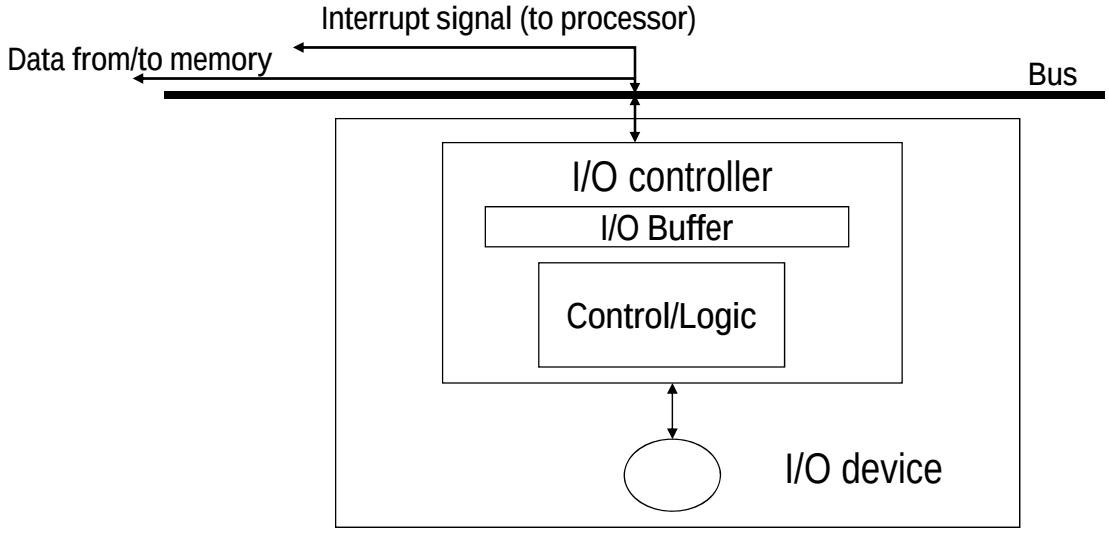
# Input/Output Subsystem

- Handles devices that allow the computer system to:
  - Communicate and interact with the outside world
    - Screen, keyboard, printer, ...
  - Store information (mass-storage)
    - Hard-drives, floppies, CD, tapes, ...
- Mass-Storage Device Access Methods:
  - Direct Access Storage Devices (DASDs)
    - Hard-drives, floppy-disks, CD-ROMs, ...
  - Sequential Access Storage Devices (SASDs)
    - Tapes (for example, used as backup devices)

# I/O Controllers

- Speed of I/O devices is slow compared to RAM
  - RAM ~ 50 nsec.
  - Hard-Drive ~ 10msec. = (10,000,000 nsec)
- Solution:
  - I/O Controller, a special purpose processor:
    - Has a small memory buffer, and a control logic to control I/O device (e.g. move disk arm).
    - Sends an interrupt signal to CPU when done read/write.
  - Data transferred between RAM and memory buffer.
  - Processor free to do something else while I/O controller reads/writes data from/to device into I/O buffer.

# Structure of the I/O Subsystem



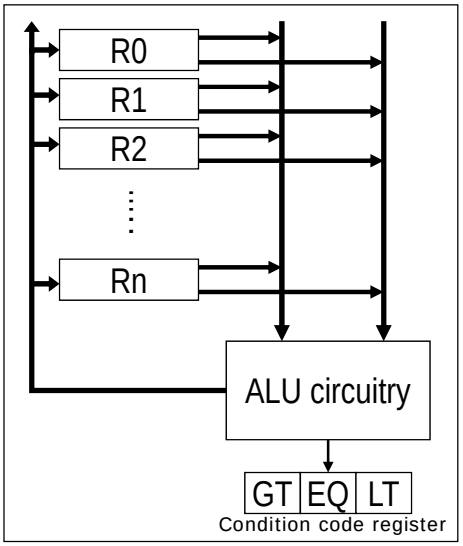

# The ALU Subsystem

- The ALU (Arithmetic/Logic Unit) performs
  - mathematical operations (+, -, x, /, …)
  - logic operations (=, <, >, and, or, not, ...)

  in today's computers integrated into the CPU
- Consists of:
  - Circuits to do the arithmetic/logic operations.
  - Registers (fast storage units) to store intermediate computational results.
  - Bus that connects the two.

# Structure of the ALU

- Registers:
  - Very fast local memory cells, that store operands of operations and intermediate results.
  - CCR (condition code register), a special purpose register that stores the result of <, = , > operations
- ALU circuitry:
  - Contains an array of circuits to do mathematical/logic operations.
- Bus:
  - Data path interconnecting the registers to the ALU circuitry.




# The Control Unit

- Program is stored in memory
  - as machine language instructions, in binary
- The task of the control unit is to execute programs by repeatedly:
  - Fetch from memory the next instruction to be executed.
  - Decode it, that is, determine what is to be done.
  - Execute it by issuing the appropriate signals to the ALU, memory, and I/O subsystems.
  - Continues until the HALT instruction

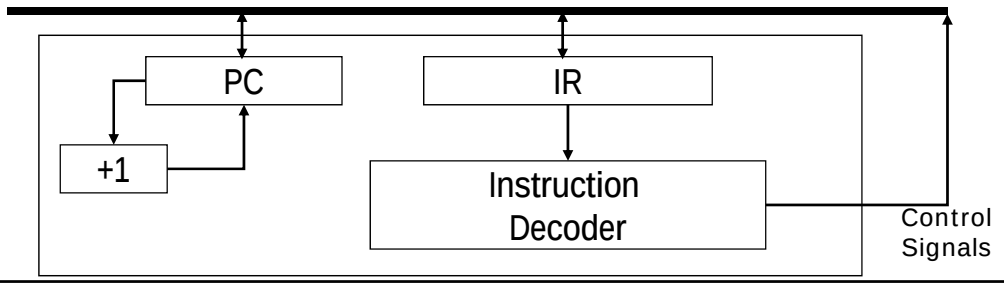
# Machine Language Instructions

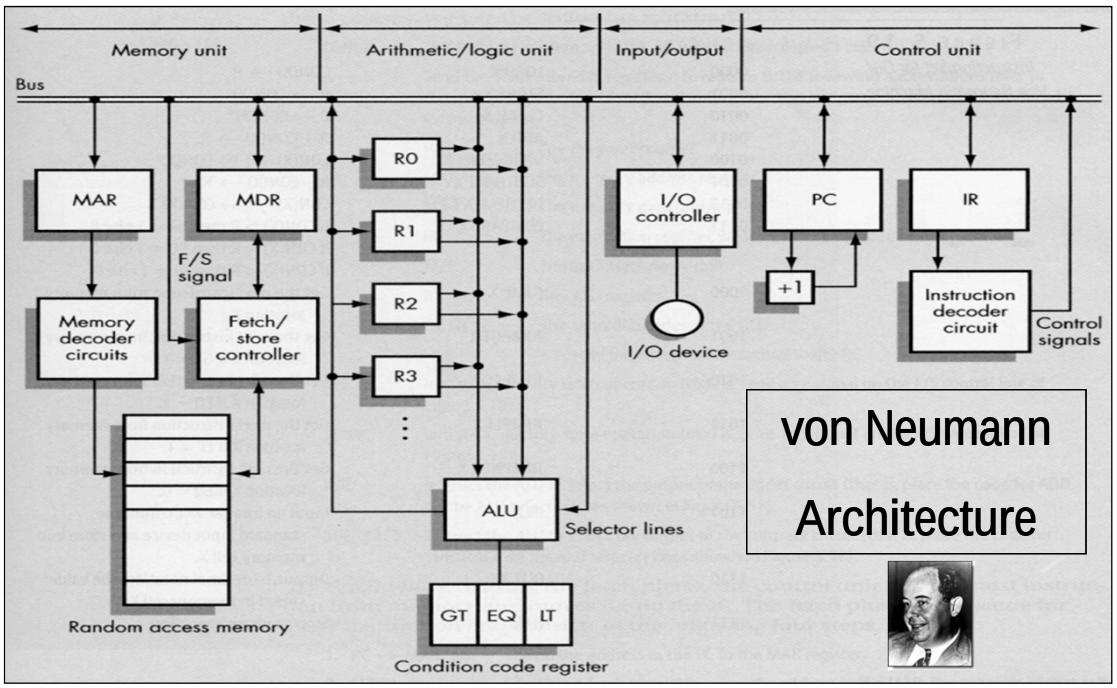
- A machine language instruction consists of:
  - Operation code, telling which operation to perform
  - Address field(s), telling the memory addresses of the values on which the operation works.
- Example: ADD X, Y (Add content of memory locations X and Y, and store back in memory location Y).
- Assume: opcode for ADD is 9, and addresses X=99, Y=100

| Opcode (8 bits) | Address 1 (16 bits) | Address 2 (16 bits) |
|---|---|---|
| 00001001 | 0000000001100011 | 0000000001100100 |

# Structure of the Control Unit

- PC (Program Counter):
  - stores the address of next instruction to fetch
- IR (Instruction Register):
  - stores the instruction fetched from memory
- Instruction Decoder:
  - Decodes instruction and activates necessary circuitry




Memory unit
Arithmetic/logic unit
Input/output
Control unit
Bus
MAR
MDR
F/S signal
Memory decoder circuits
Fetch/ store controller
Random access memory
R0
R1
R2
R3
ALU
Selector lines
GT
EQ
LT
Condition code register
I/O controller
I/O device
PC
+1
IR
Instruction decoder circuit
Control signals
von Neumann Architecture