



การจัดองค์การคอมพิวเตอร์

w4.3 Hack Computer

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 3 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิติศรีวรพันธุ์

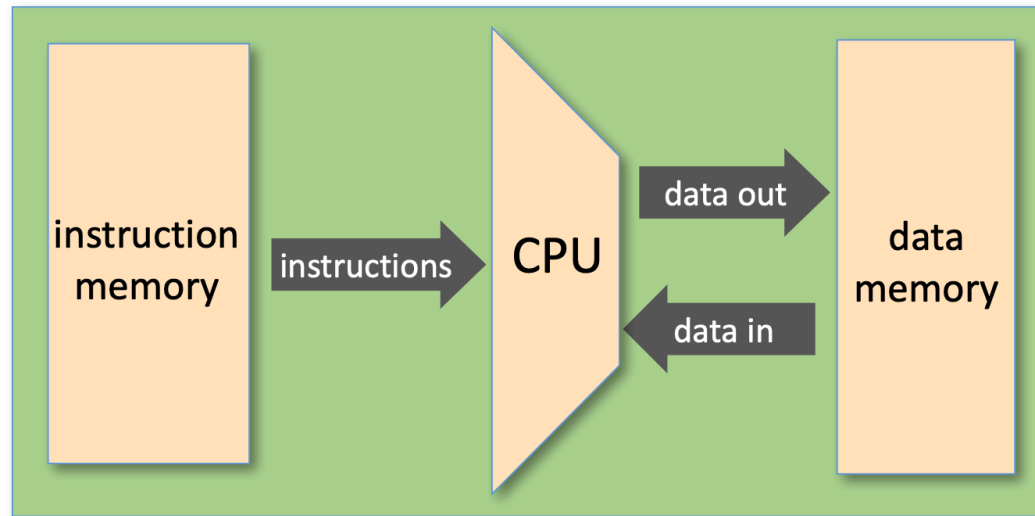
songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

Lecture plan

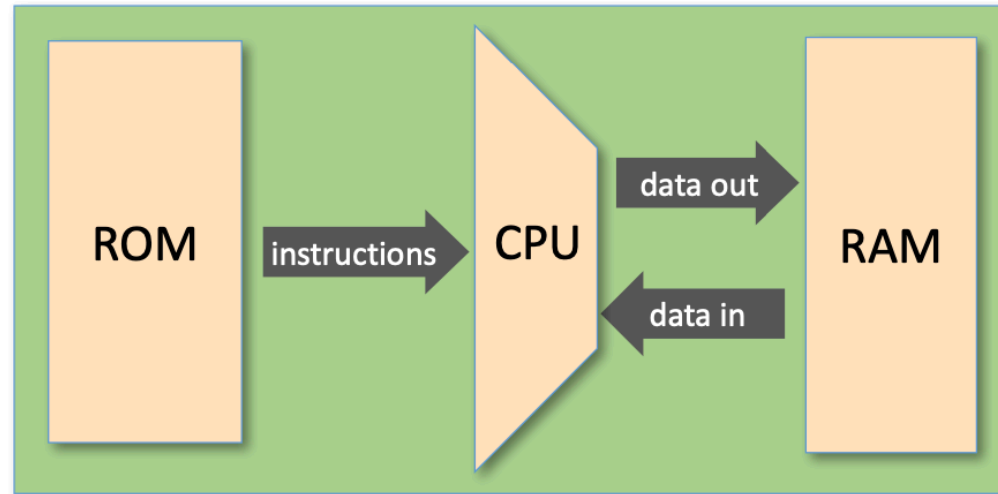
- 4.1 ภาษาเครื่อง
- 4.2 ส่วนประกอบพื้นฐาน
- **4.3 ระบบแอสกัคคอมพิวเตอร์และภาษาเครื่อง**
- 4.4 ภาษาเครื่องแอสกัค
- 4.5 อินพุต / เอาท์พุต
- 4.6 การเขียนโปรแกรมสำหรับเครื่องแอสกัค
- 4.7 ภาพรวมโปรเจกต์สัปดาห์ 4

ระบบแอสกคอมพิวเตอร์ : ฮาร์ดแวร์



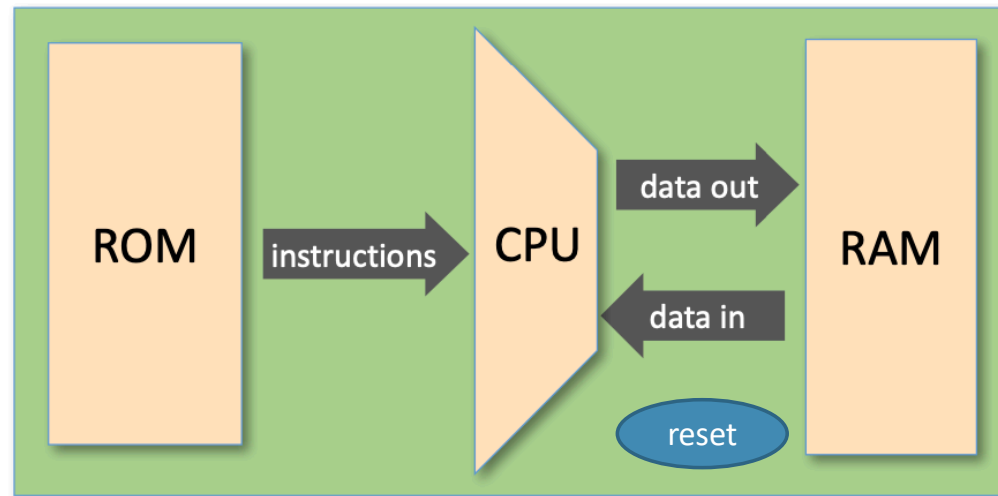
- คอมพิวเตอร์ 16-bit
 - Data memory (RAM) : มีแอดเดรส 16-bit
RAM[0], RAM[1], RAM[2],
 - Instruct memory (ROM) : มีแอดเดรส 16-bit
ROM[0], ROM[1], ROM[2]
 - ซีพียู (CPU) ประมวลผลข้อมูลครั้งละ 16-bit

ระบบแอสคคอมพิวเตอร์ : ซอฟต์แวร์



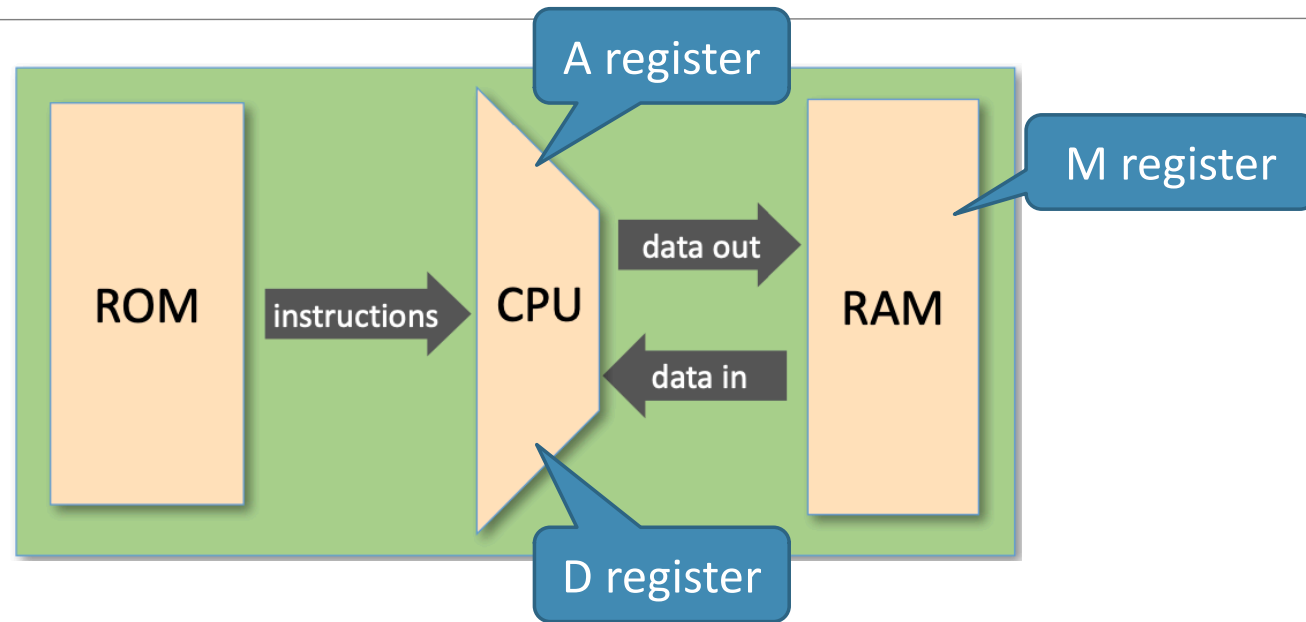
- ภาษาเครื่อง
 - A-instruction ขนาด 16-bit
 - C-instruction ขนาด 16-bit
- Hack program คือชุดคำสั่งภาษาเครื่องเขียนเป็นลำดับ

ระบบแอสคคอมพิวเตอร์ : ควบคุม



- Control
 - Hack โปรแกรมบันทึกใน ROM
 - มีปุ่ม reset เมื่อกดแล้ว ระบบจะกลับไปตำแหน่ง 0 ของ ROM
 - Boot เครื่อง

ระบบแอสกคอมพิวเตอร์ : รีจิสเตอร์



- ภาษาเครื่องแอสก มีรีจิสเตอร์ 3 ประเภท
 - A : ใช้เก็บข้อมูล / แอดเดรสของหน่วยความจำ
 - D : ใช้เก็บข้อมูล
 - M : ใช้อ้างอิงตำแหน่งรีจิสเตอร์ที่ระบบกำลังอ่าน

ชุดคำสั่ง A

- Syntax:

@value

- หน้าที

- กำหนดค่าให้ รีจิสเตอร์ A
- ผลของคำสั่ง
 - ทำให้เปลี่ยนตำแหน่ง RAM[A]
 - ทำให้เปลี่ยนตำแหน่ง ROM[A]

```
// Sets A to 17  
@17
```

เป็นคำสั่งเลือก RAM[17]

เป็นคำสั่งเลือก ROM[17]

- ค่า value

- ไม่เป็นลบ
- อาจเป็นตัวแปรใด ๆ

- เซตค่า -1 ที่ RAM[100]

```
// Set RAM[100] เท่ากับ -1  
@100 // A=100  
M=-1 // RAM[100] = -1
```

ชุดคำสั่ง C (1/3)

- Syntax

dest = comp ; jump

comp = 0, 1, -1, D, A, !D, !A, -D, -A, D+1, A+1, D-1, A-1, D+A, D-A, A-D, D&A, D|A
M, !M, -M, M+1, M-1, D+M, D-M, M-D, D&M, D|M

dest = null, M, D, MD, A, AM, AD, AMD

jump = null, JGT, JEQ, JGE, JLT, JNE, JLE, JMP

if (comp jump 0) jump to execute
The instruction in ROM[A]

- เซตรีจิสเตอร์ D = -1

```
// Set D register to -1  
D=-1
```


ชุดคำสั่ง C (2/3)

- Syntax

dest = comp ; jump

comp = 0, 1, -1, D, A, !D, !A, -D, -A, D+1, A+1, **D-1**, A-1, D+A, D-A, A-D, D&A, D|A
M, !M, -M, M+1, M-1, D+M, D-M, M-D, D&M, D|M

dest = null, **M**, D, MD, A, AM, AD, AMD

jump = null, JGT, JEQ, JGE, JLT, JNE, JLE, JMP

if (comp jump 0) jump to execute
The instruction in ROM[A]

- เซต RAM[300] ให้เท่ากับรีจิสเตอร์ D ลบด้วย 1

```
// Set RAM[300] to value D minus 1
@300    //A=300
M=D-1   //RAM[300]=D-1
```

ชุดคำสั่ง C (3/3)

- Syntax

dest = comp ; jump

comp = 0, 1, -1, D, A, !D, !A, -D, -A, D+1, A+1, **D-1**, A-1, D+A, D-A, A-D, D&A, D|A
M, !M, -M, M+1, M-1, D+M, D-M, M-D, D&M, D|M

dest = null, M, D, MD, A, AM, AD, AMD

jump = null, JGT, **JEQ**, JGE, JLT, JNE, JLE, JMP

if (comp jump 0) jump to execute
The instruction in ROM[A]

- if (D-1==0) jump ไปที่
ตำแหน่ง ROM[56]

```
// if (D-1==0) jump to execute the  
instruction stored in ROM[56]
```

```
@56          //A=56
```

```
D-1; JEQ     //if (D-1 == 0) goto 56
```