



การจัดองค์การคอมพิวเตอร์

พ4.1 : ภาพรวมภาษาเครื่อง

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 3 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิติศรีวรพันธุ์

songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

Lecture plan

- 4.1 ภาษาเครื่อง
- 4.2 ส่วนประกอบพื้นฐานของภาษาเครื่อง
- 4.3 ระบบแอสกคคอมพิวเตอร์และภาษาเครื่อง
- 4.4 ภาษาเครื่องแอสกค
- 4.5 อินพุต / เอาต์พุต
- 4.6 การเขียนโปรแกรมสำหรับเครื่องแอสกค
- 4.7 ภาพรวมโปรเจค #4

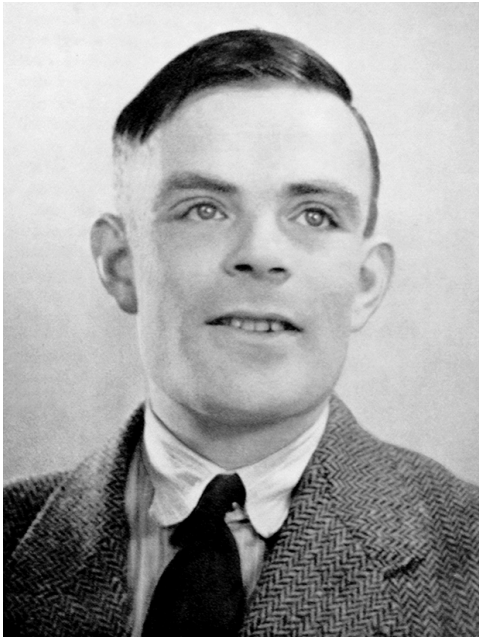
คอมพิวเตอร์

- General proposed



คอมพิวเตอร์ยุคเริ่มต้น

- ฮาร์ดแวร์เดียวกัน รับซอฟต์แวร์ได้หลากหลาย



แอลัน แมริสัน ทัวริง (19th – 20th)

ด้านทฤษฎีคอมพิวเตอร์
เสนอเครื่องจักรทัวริง

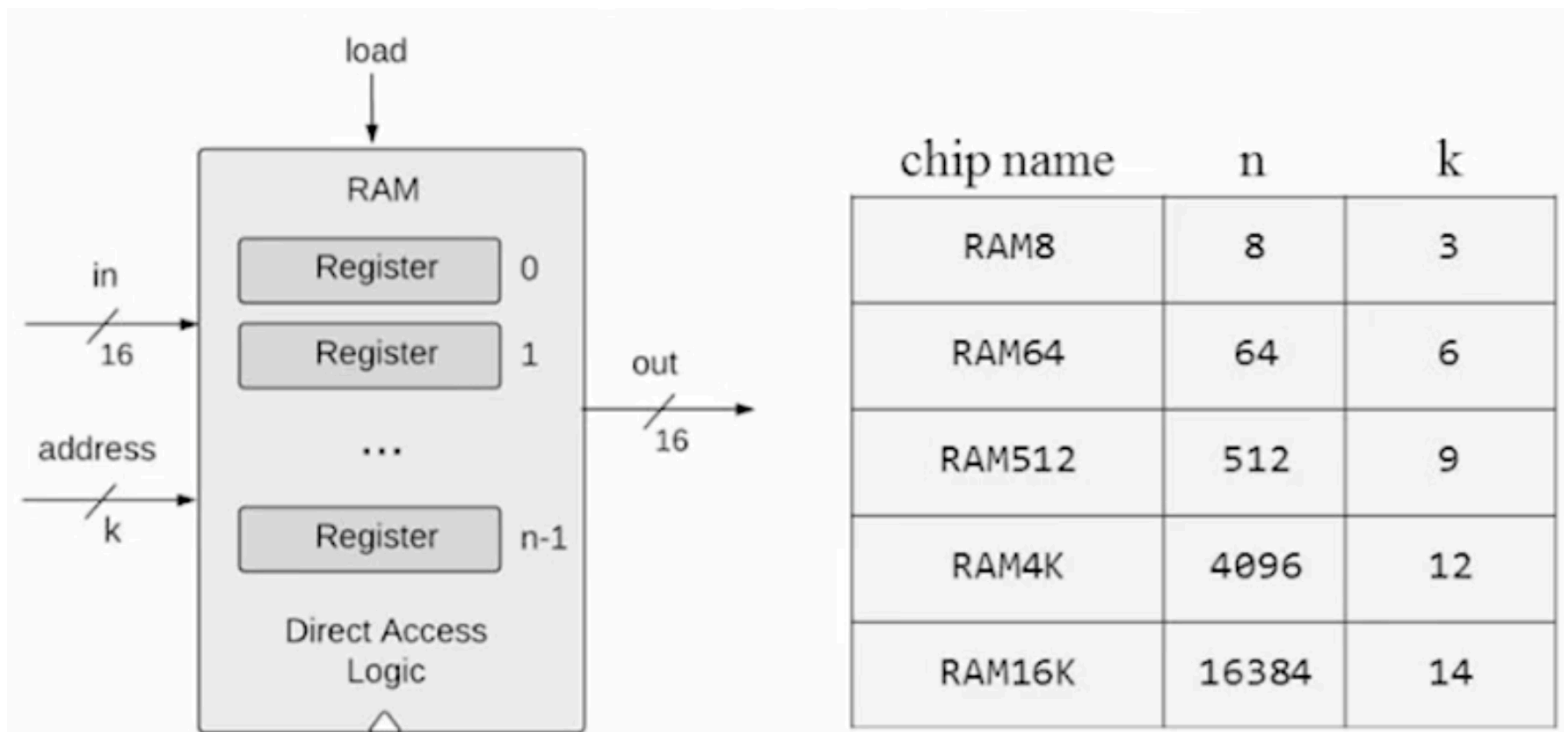


จอห์น ฟอน นอยมันน์

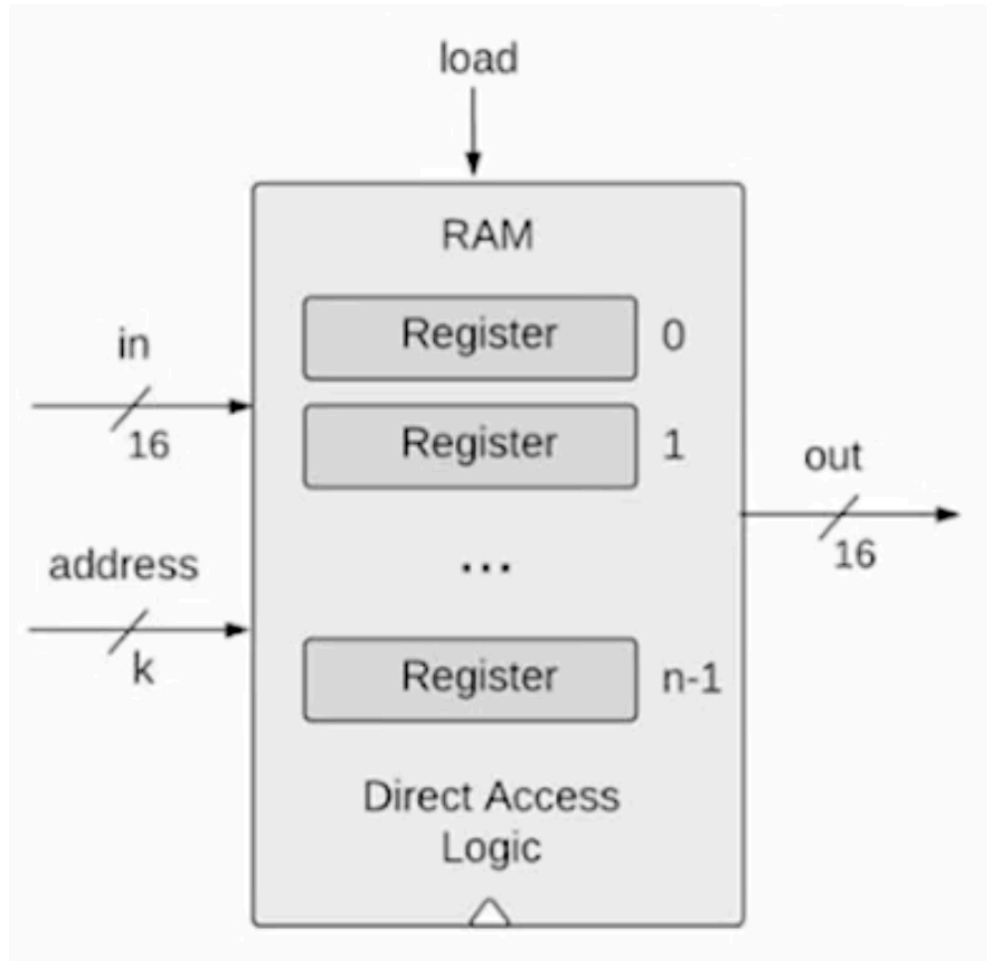
ด้านวิศวกรรมคอมพิวเตอร์
เสนอสถาปัตยกรรมฟอนนอยมันน์

หน่วยความจำ

- RAM16K
- มี Register ทั้งหมด $16 \times 1024 = 16384$ ชิป
- มีขา Address เท่ากับ $\log_2(16384) = 14$ ขา

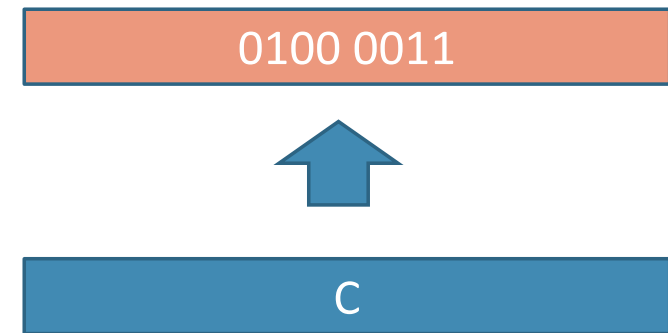
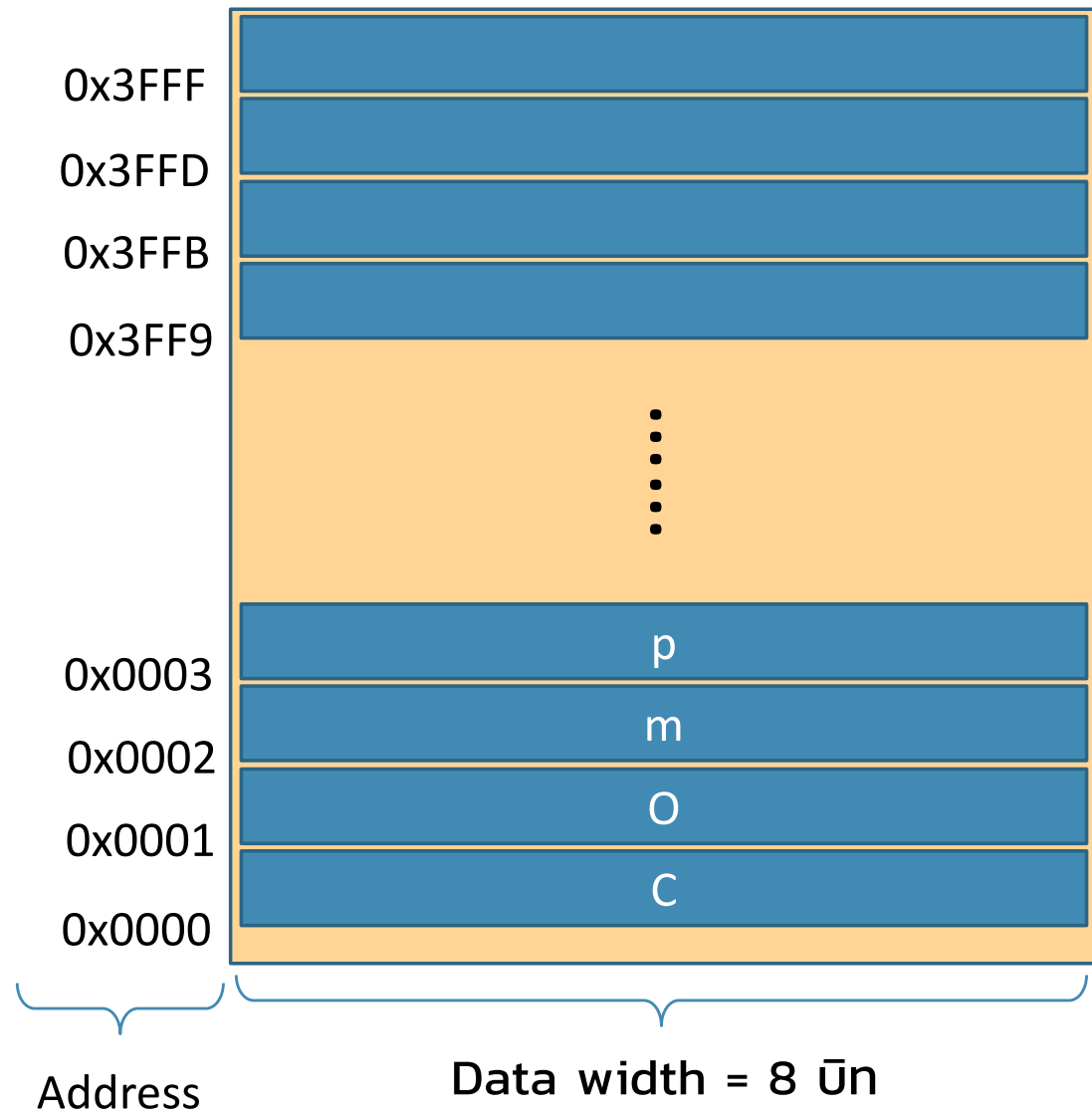


RAM16K

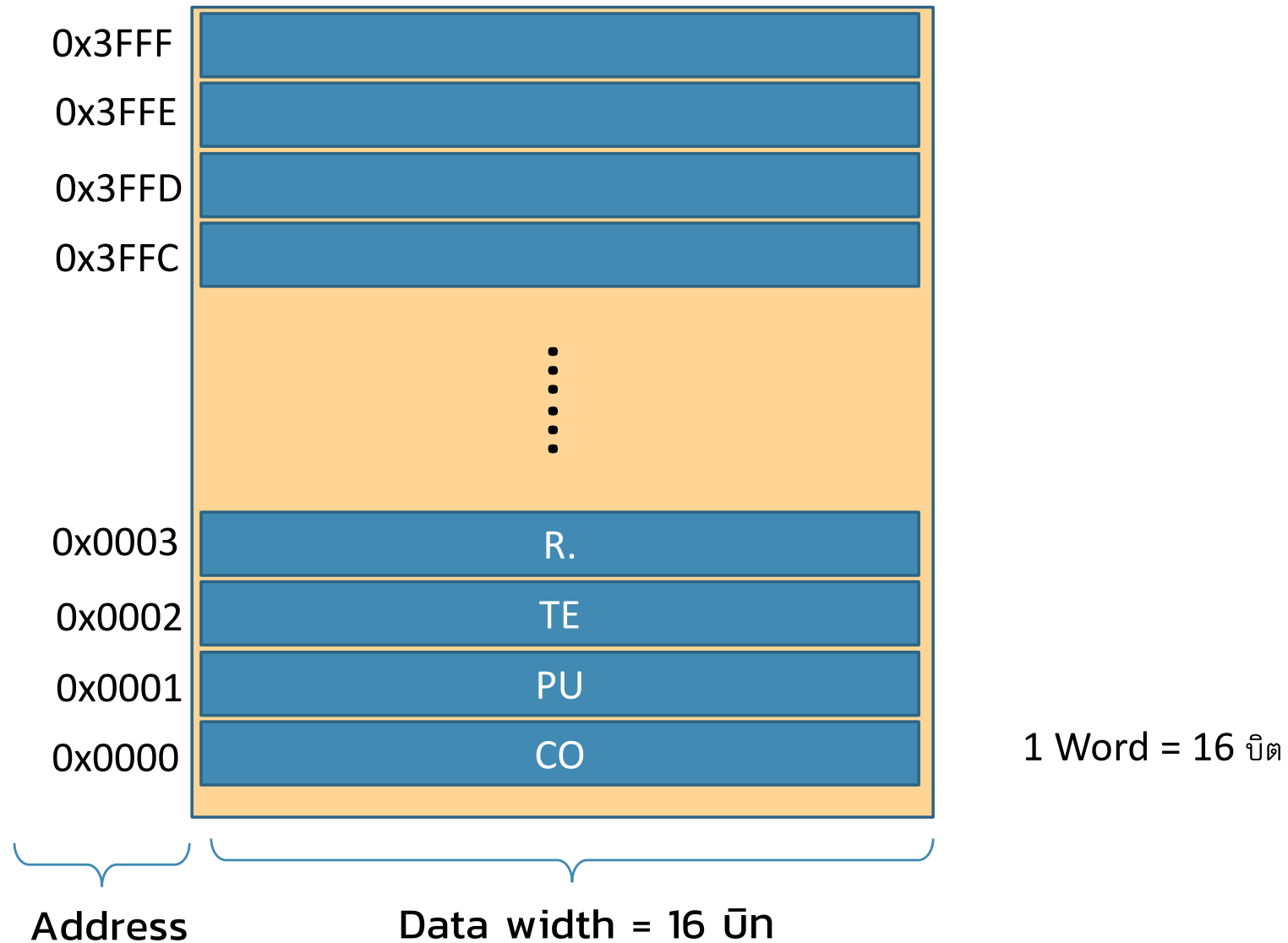


- เขียนข้อมูลครั้งละ 16 บิต
- หรือ 2 ไบต์

1 Word = 8 $\bar{u}n$

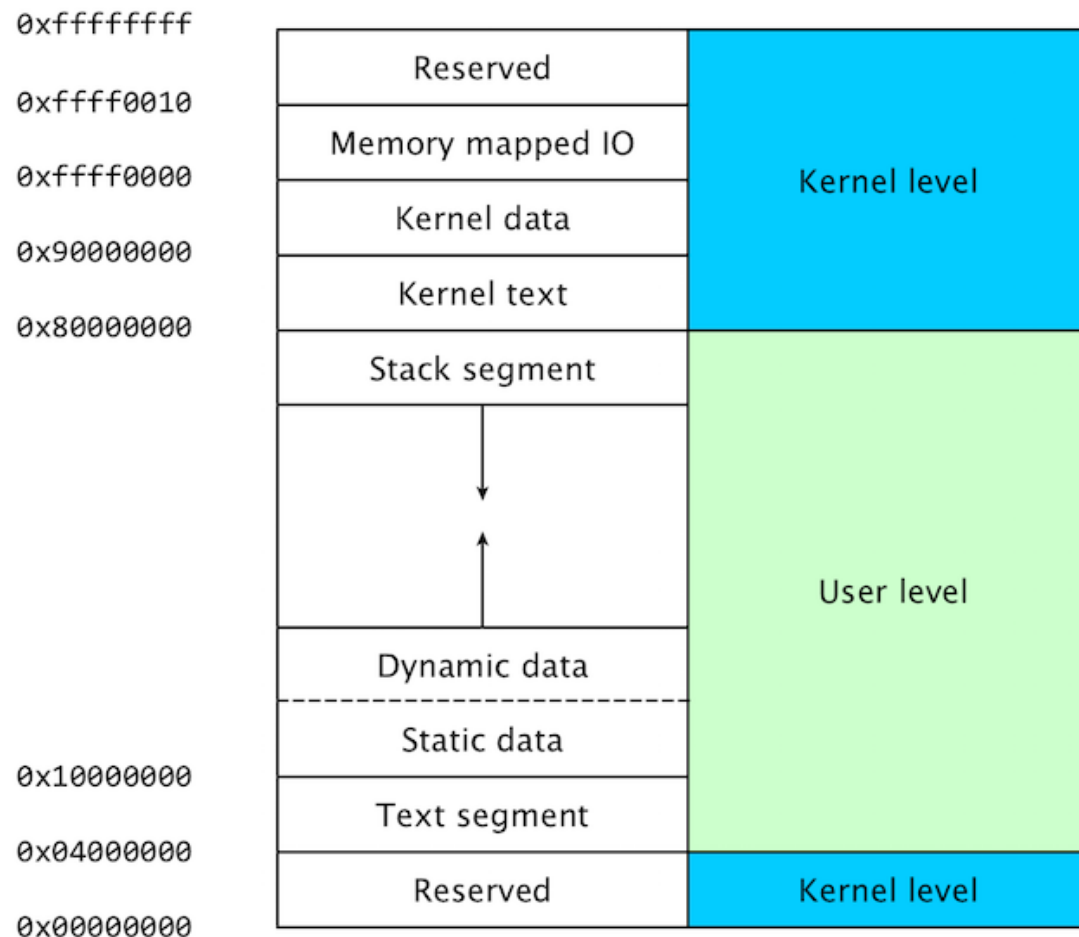


Hack computer (data 16 บิต)



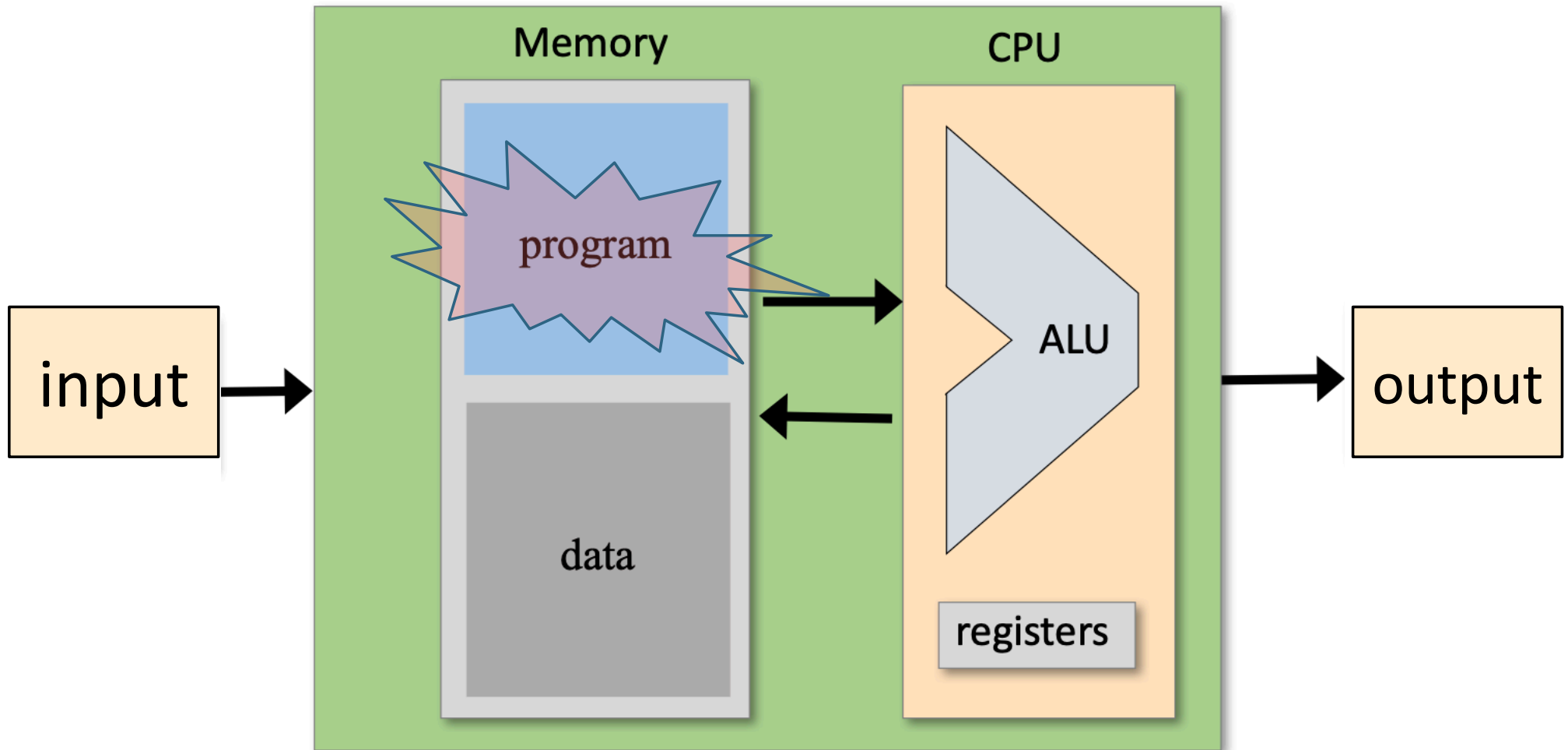
ภาพอธิบายหน่วยความจำ

วิชาระบบปฏิบัติการคอมพิวเตอร์ Operating System

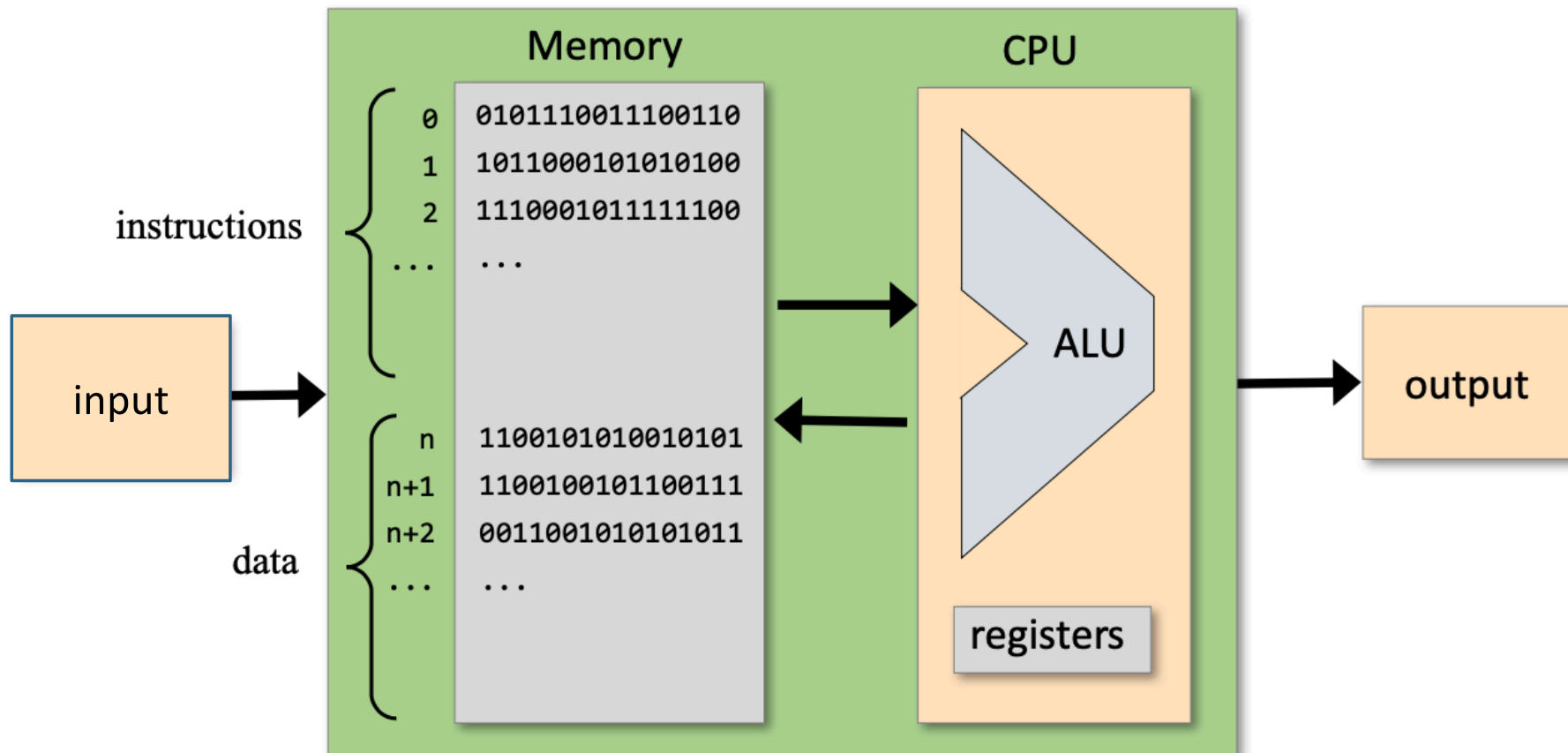


บันทึกโปรแกรมใน Memory

ระบบคอมพิวเตอร์

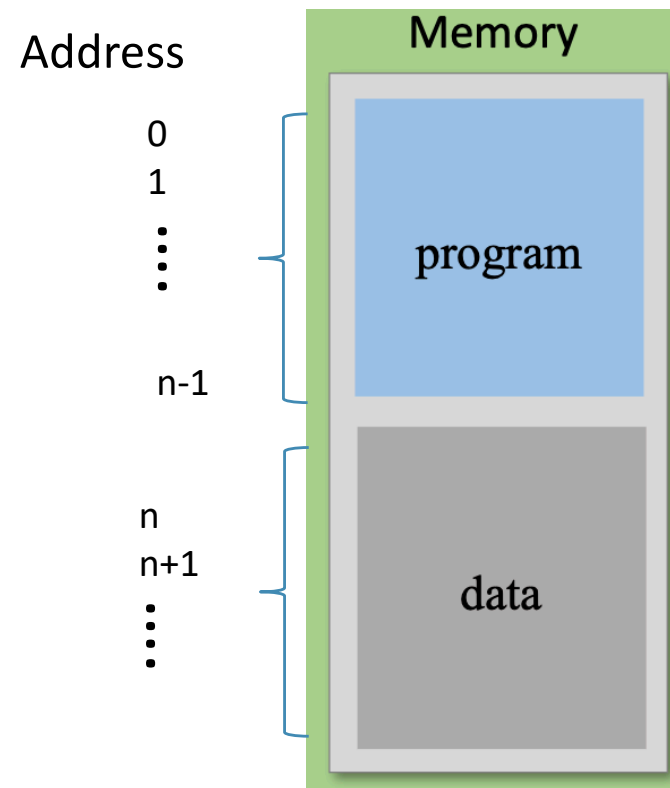


บันทึกโปรแกรมใน Memory

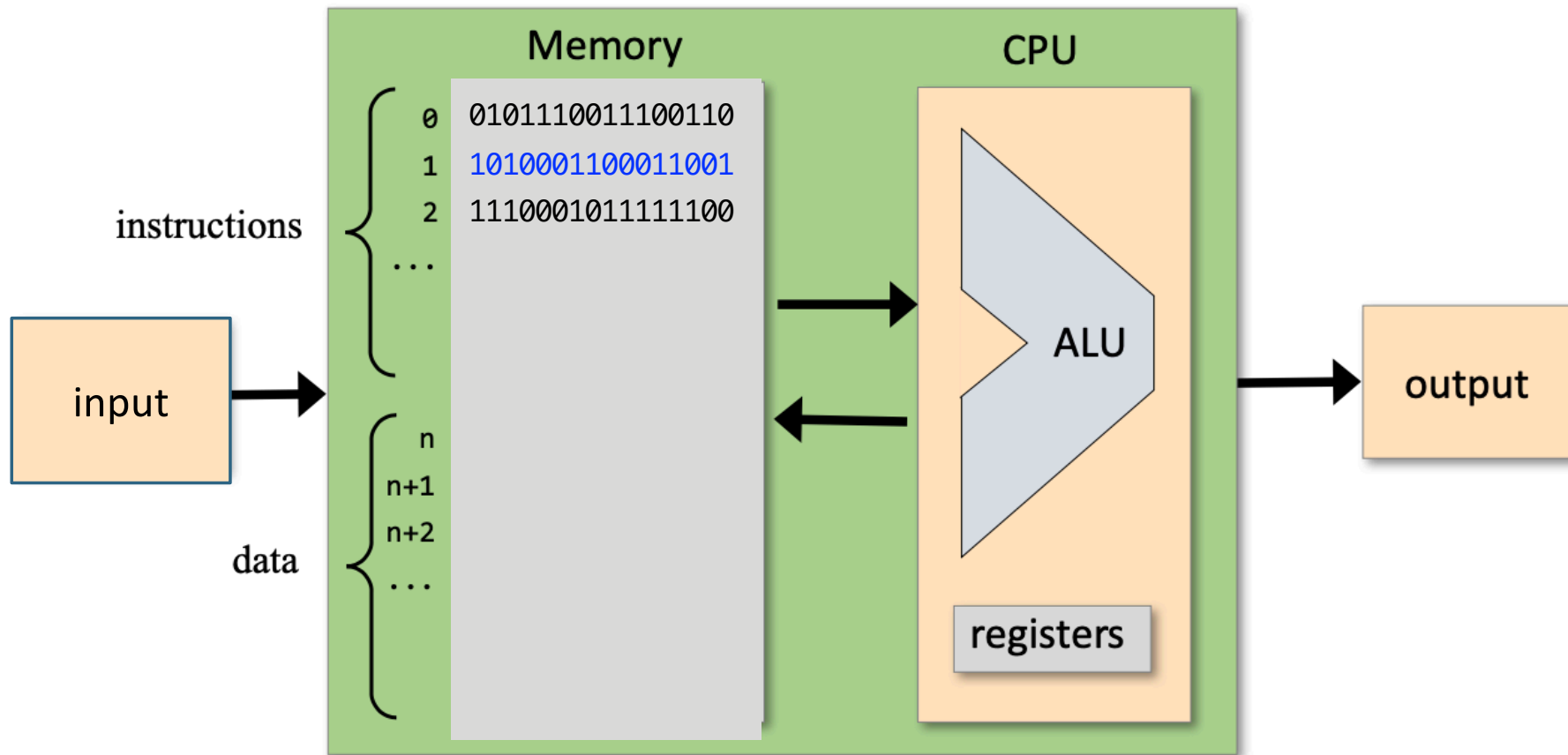


บันทึกโปรแกรมใน Memory

- ระบบคอมพิวเตอร์ ประมวลผลเฉพาะข้อมูลในหน่วยความจำ
- โหลดข้อมูลใส่ หน่วยความจำ ก่อนการประมวลผล
- แบ่งหน่วยความจำเป็น 2 ส่วน
 - ชุดคำสั่ง (Program)
 - ชุดข้อมูล (Data)

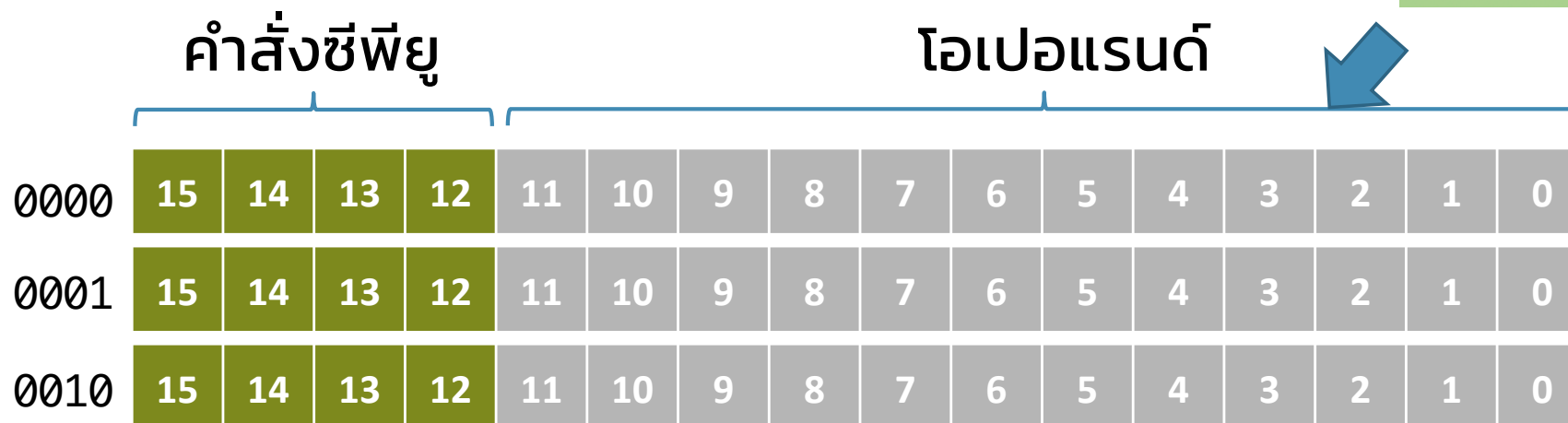
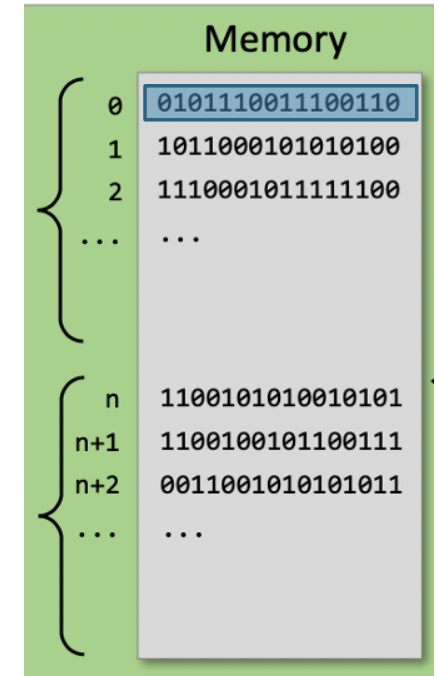


ชุดคำสั่ง และ ชุดข้อมูล



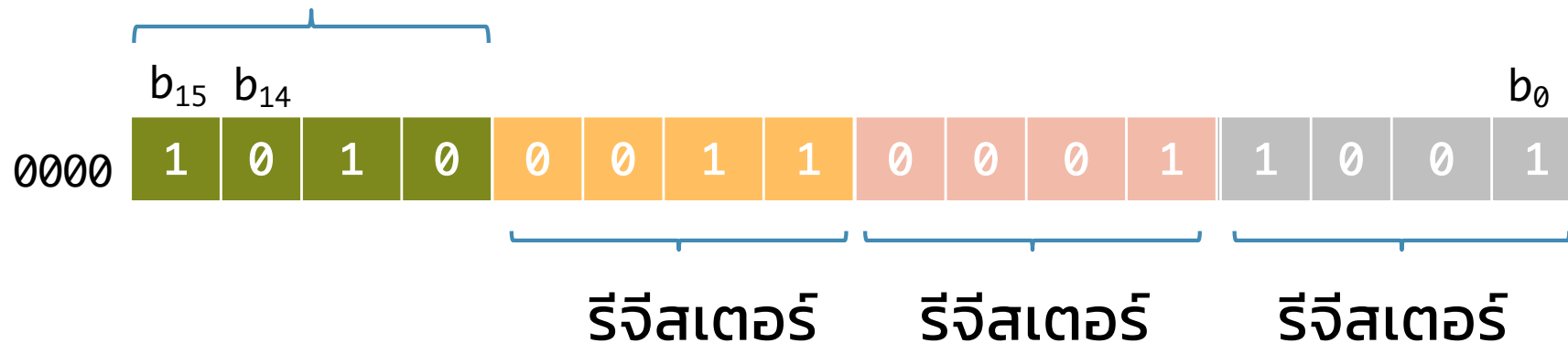
ตัวอย่าง instruction set

- มีการตกลงรูปแบบคำสั่งไว้ล่วงหน้า
- ตัวอย่างการออกแบบ instruction set
- หน่วยความจำมี 16-bit
 - แบ่งเป็นกลุ่มละ 4-bit มี 4 กลุ่ม
 - กลุ่ม 4-bit แรกเป็น**คำสั่งซีพียู** (CPU operation)
 - เหลืออีก 3 เรียกเรียกว่า **โอเปอเรนด์** (Operand)



นี่โมนิค (Mnemonic) 1/2

นี่โมนิค คือ อักษรภาษาอังกฤษใช้แทนรหัสไบนารี



1010 คำสั่งบวก "addition" = ADD

0011 รีจิสเตอร์ 3 (R3)

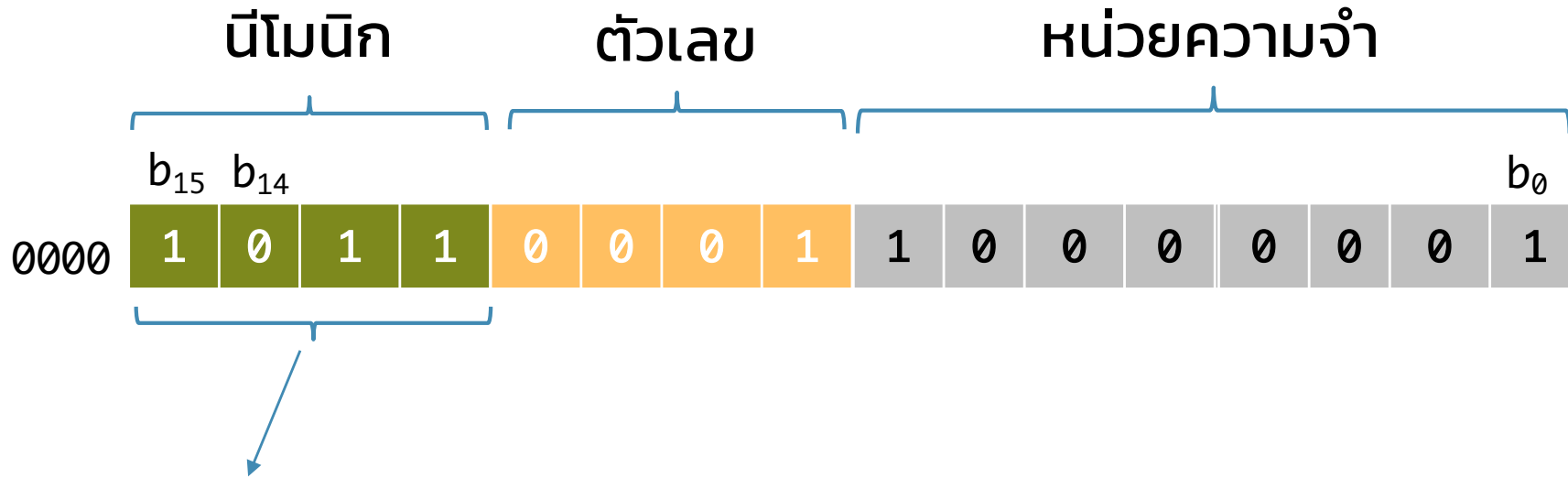
0001 รีจิสเตอร์ 1 (R1)

1001 รีจิสเตอร์ 9 (R9)

นี่โมนิค ใช้ตัวอักษรทดแทน

- ADD R3,R1,R9 → 1010001100011001

ប៉ិនប៉ាគ (Mnemonic) 2/2



1011 บวก เลขใดๆ กับ ข้อมูลในหน่วยความจำ

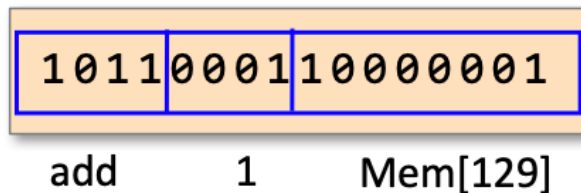
แทนด้วยอักษร

- `ADD 1, MEM[129]`

บิต	แทนด้วยอักษร
1010	ADD R_i, R_j, R_k
1011	ADD value, Mem[addr]
...	
...	

ตัวอักษร (Symbols) → แอสเซมบลี

- คำสั่งไบนารี



1010001100011001

ADD R3,R1,R9

- แอสเซมบลี :

add 1, Mem[129]

เพื่อให้อ่านง่าย เราเขียน
index = Mem[addr]

add 1, index

บิตโมนิก	ภาษาแอสเซมบลี
1010	ADD R _i , R _j , R _k
1011	ADD value, Mem[addr]
...	
...	

คอมพิวเตอร์

high-level program

```
while (n < 100) {  
    sum += arr[i];  
    n++  
}
```

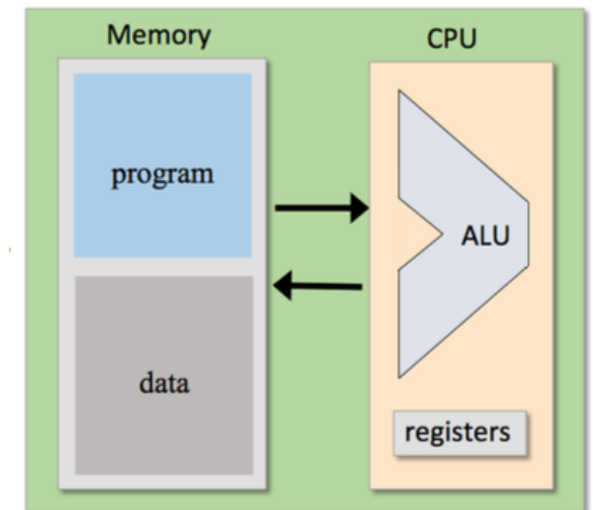


compile

machine language

```
0101111100111100  
1010101010101010  
1101011010101010  
1001101010010101  
1101010010101010  
1110010100100100  
0011001010010101  
1100100111000100  
1100011001100101  
0010111001010101  
...
```

load and
execute



Lecture plan

- 4.1 ภาษาเครื่อง
- **4.2 ส่วนประกอบพื้นฐานของภาษาเครื่อง**
- 4.3 ระบบแอสกคคอมพิวเตอร์และภาษาเครื่อง
- 4.4 ภาษาเครื่องแอสกค
- 4.5 อินพุต / เอาท์พุต
- 4.6 การเขียนโปรแกรมสำหรับเครื่องแอสกค
- 4.7 ภาพรวมโปรเจค #4