



การจัดองค์การคอมพิวเตอร์

W5.3 ซีพียูแอสกซ์

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 3 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิติศรีวรพันธุ์

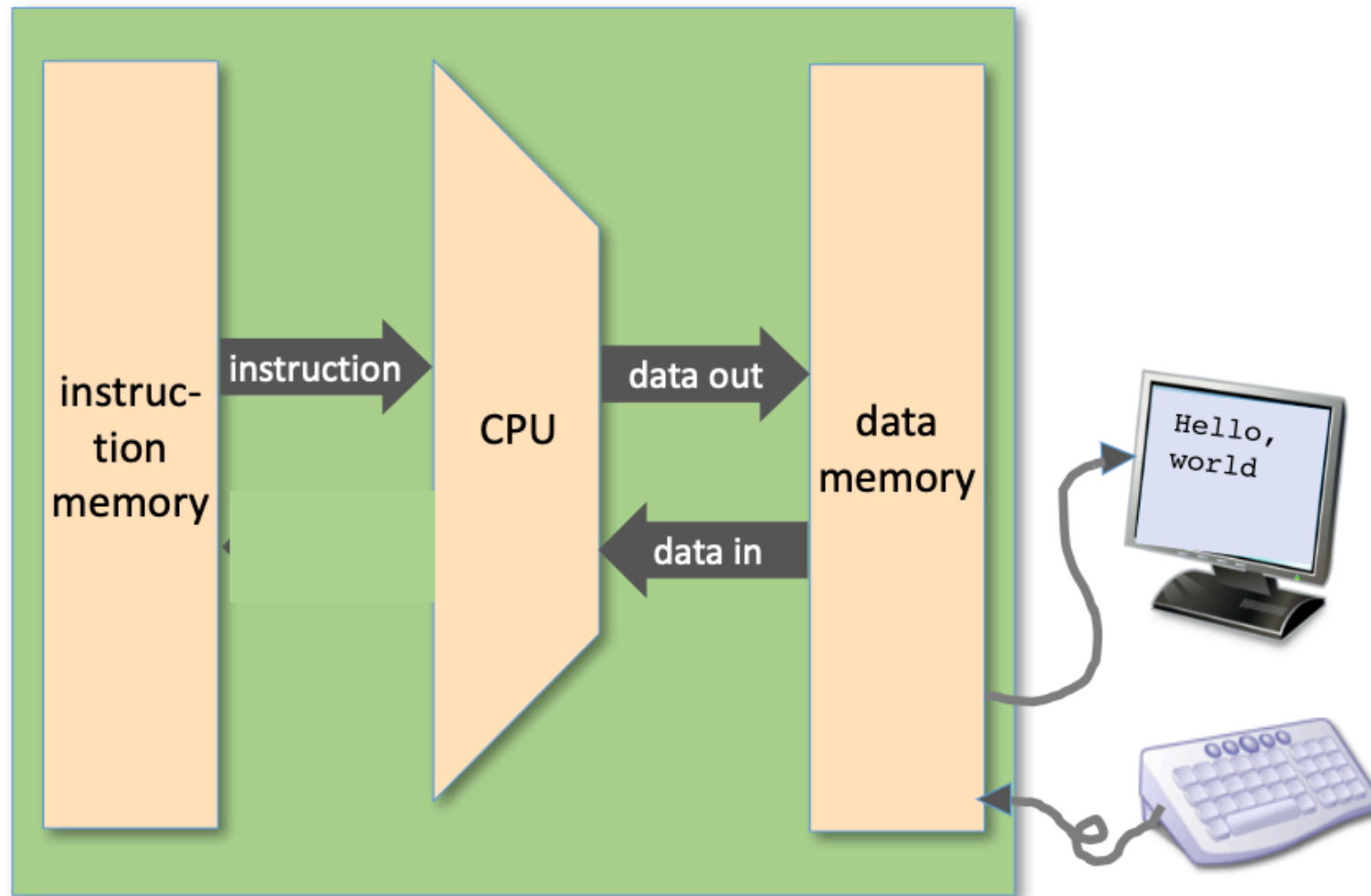
songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

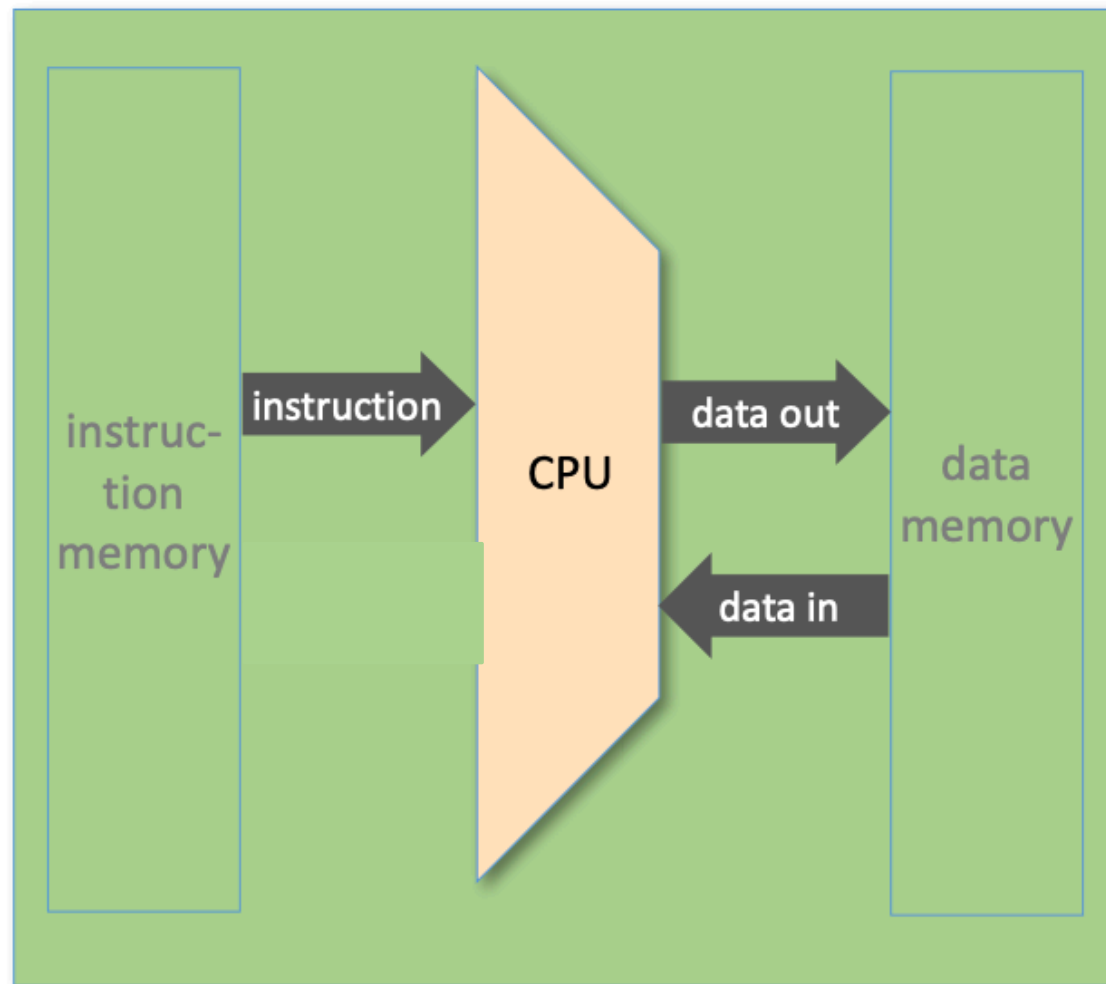
Lecture plan

- 5.1 สถาปัตยกรรมฟอนนอยมันน์
- 5.2 Fetch-Execute Cycle
- **5.3 ซีพียูแอสกซ์**
- 5.4 แอสกซ์คอมพิวเตอร์
- 5.5 ภาพรวมโปรเจกต์ 5

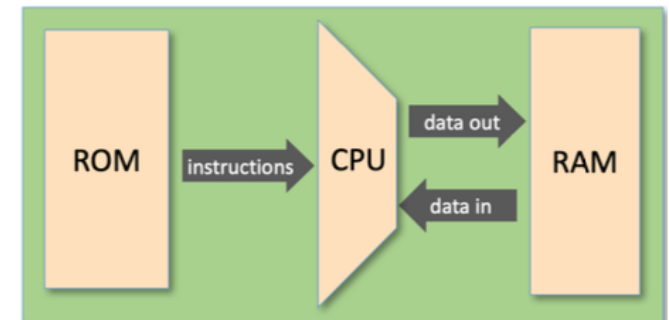
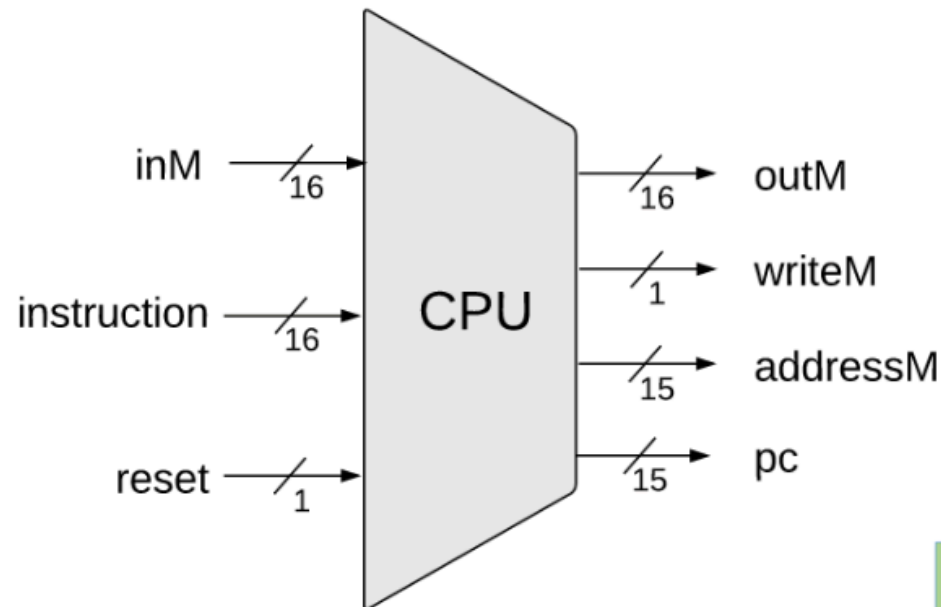
Hack Computer



Hack CPU



Hack CPU interface



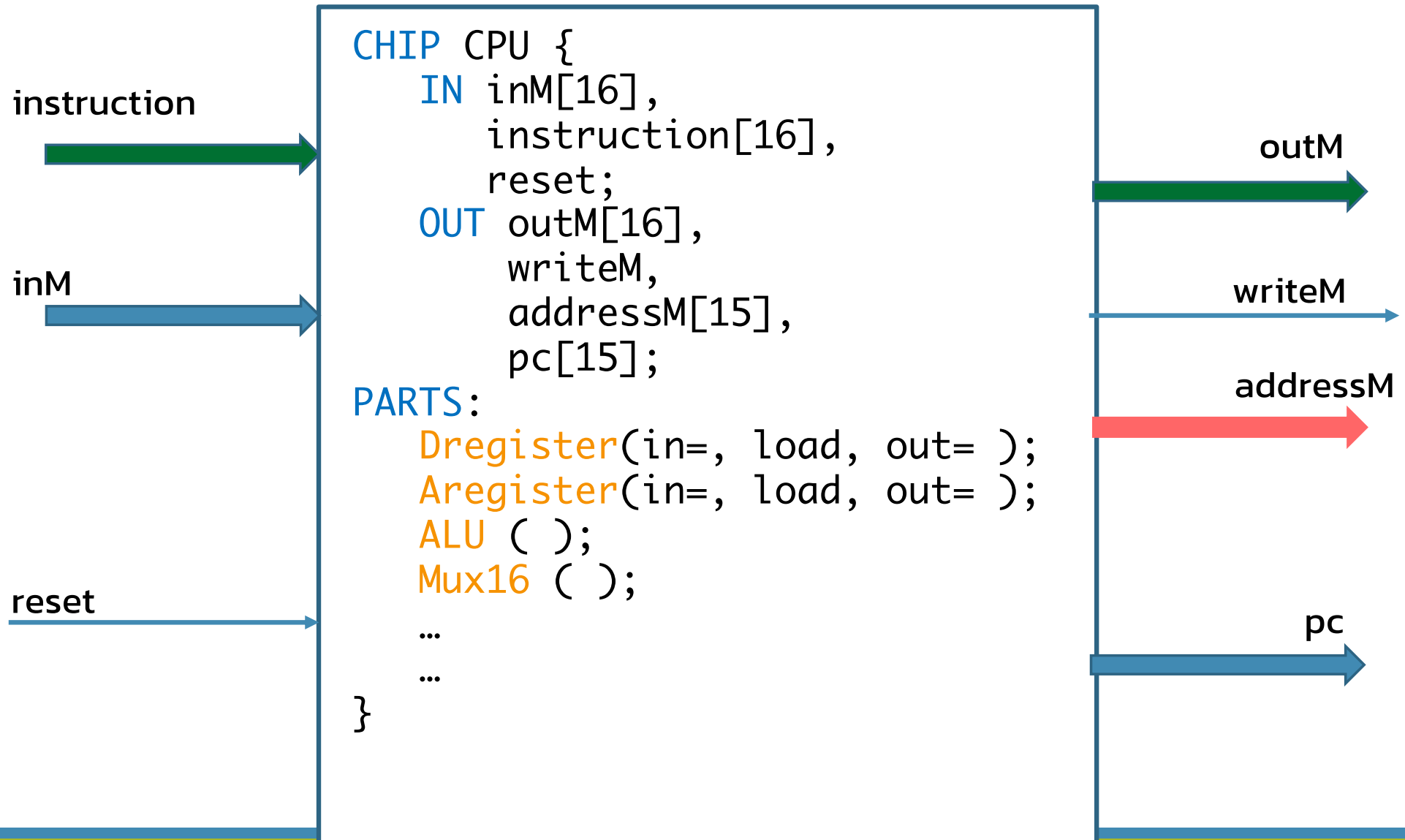
- Inputs:

- Data value
- Instruction
- Reset bit

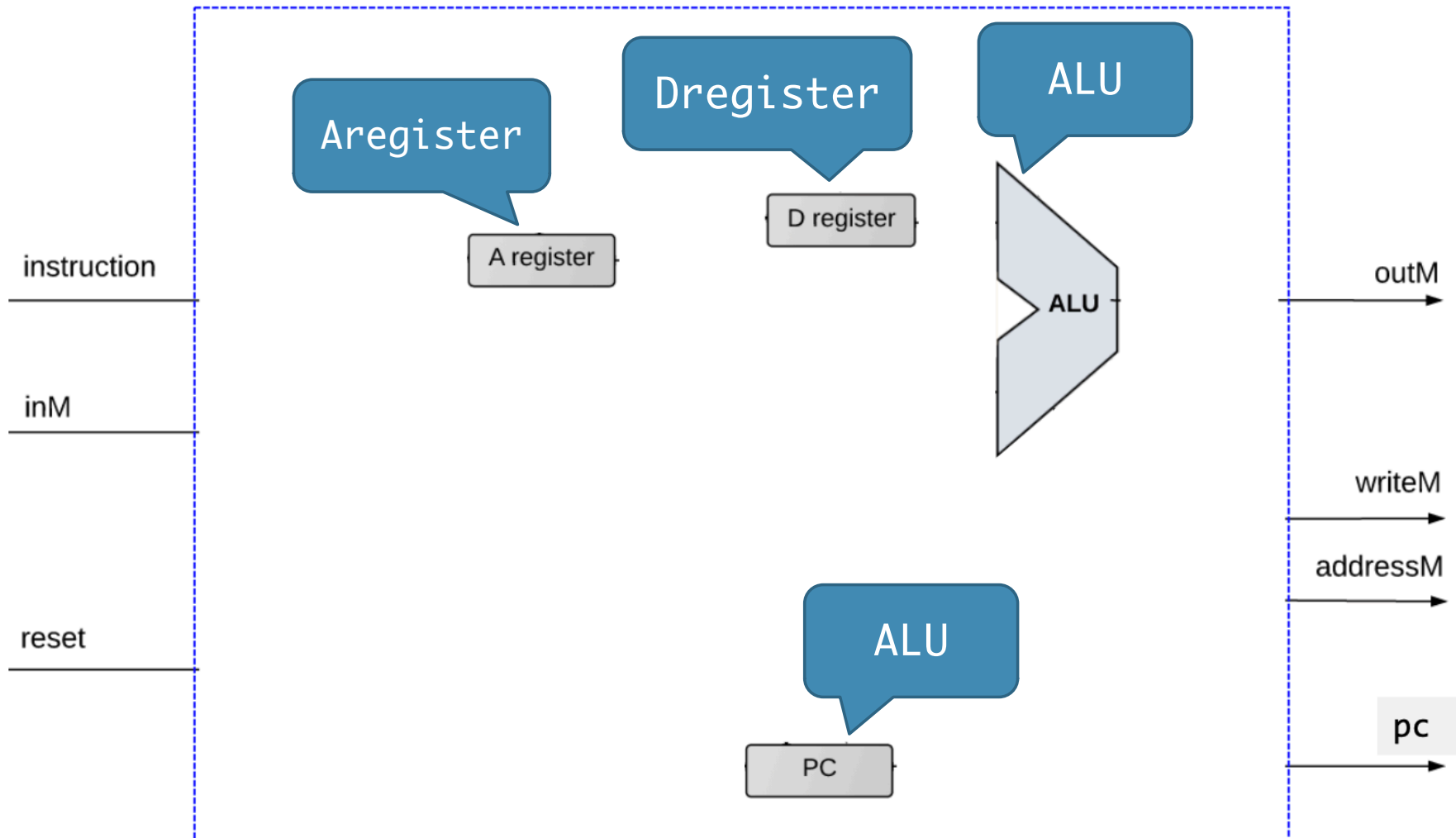
- Outputs:

- Data value
- Write memory
- Memory address
- Address of next instruction

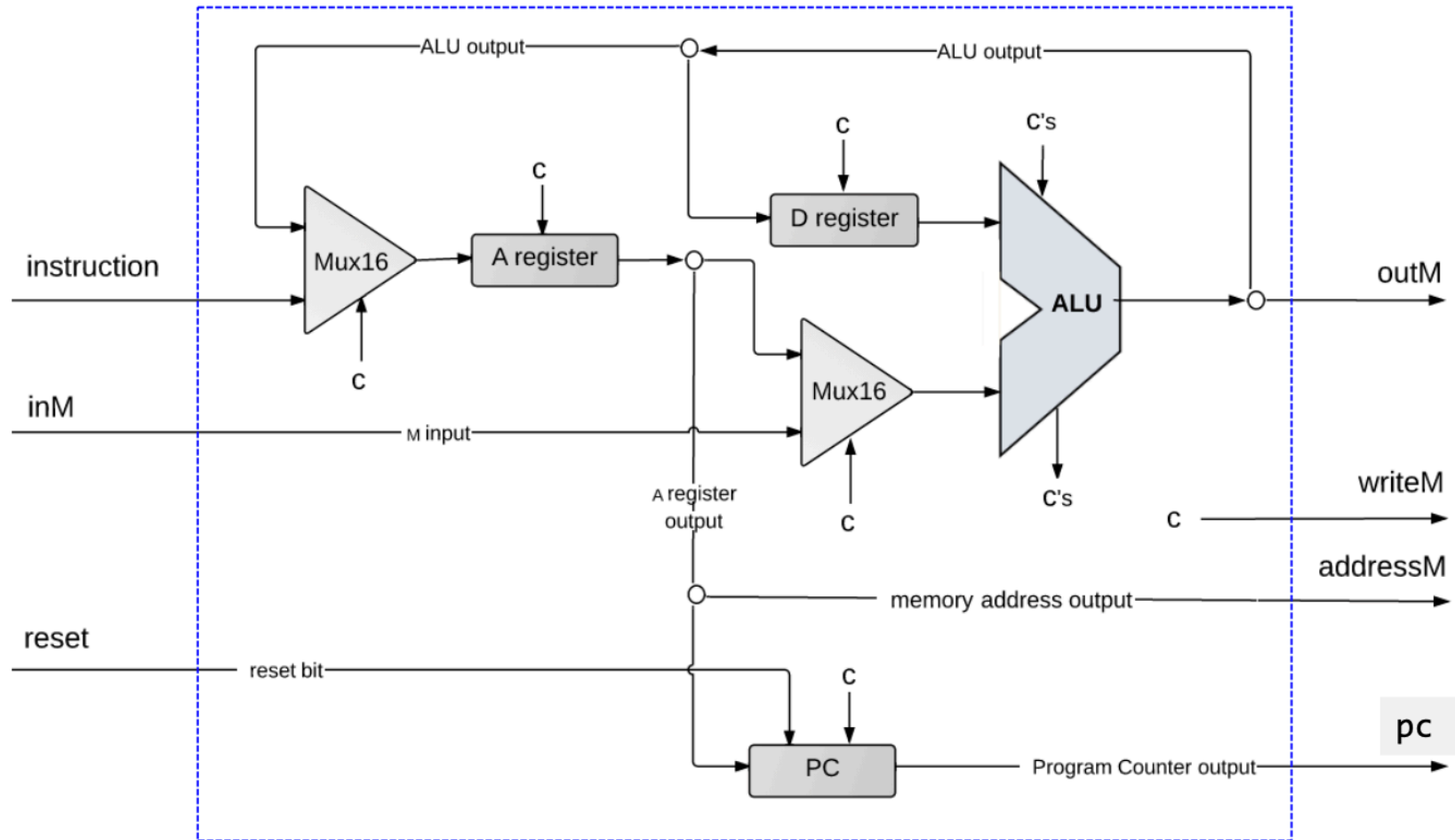
CPU interface



Hack CPU Implementation

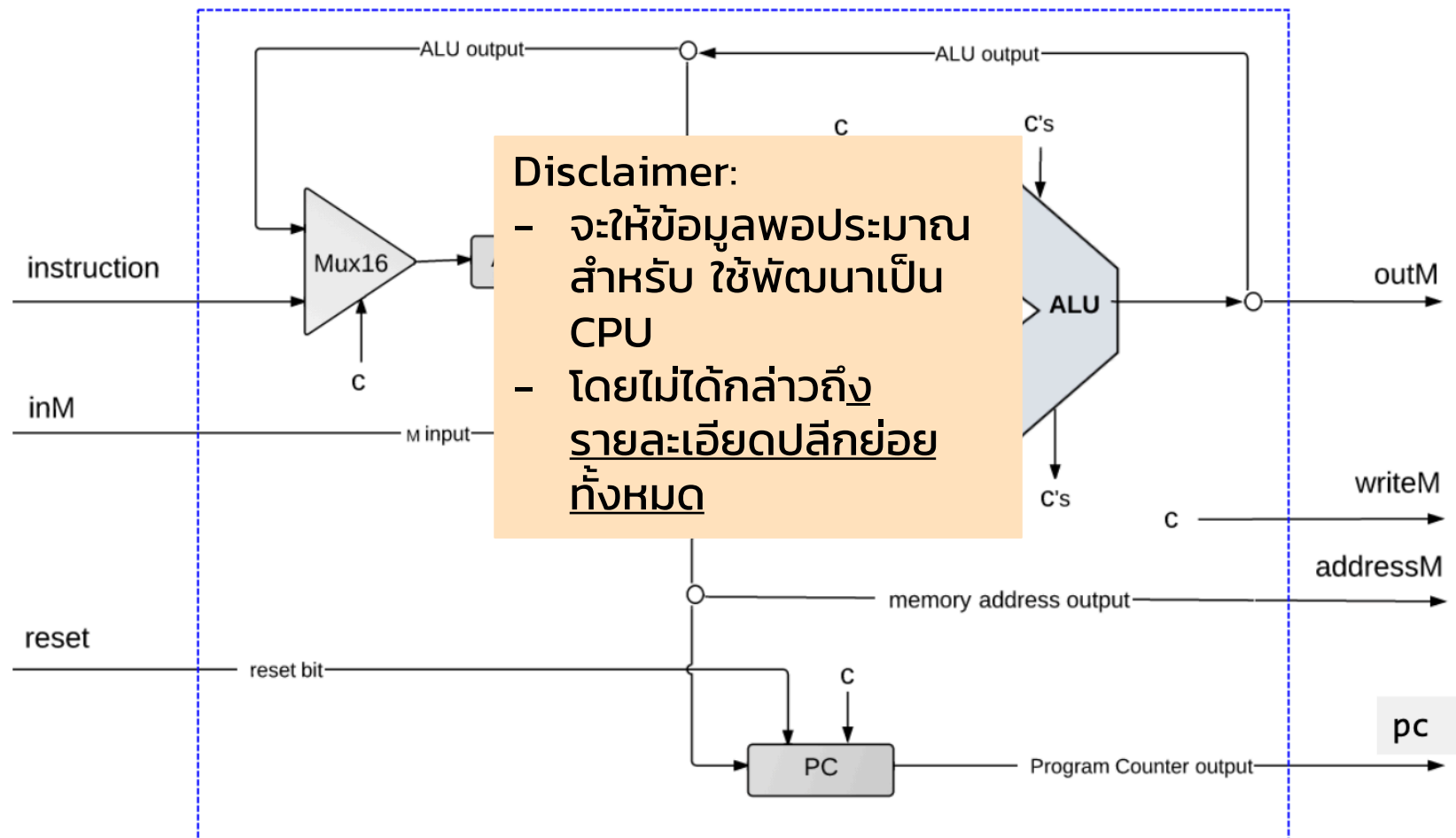


Hack CPU Implementation

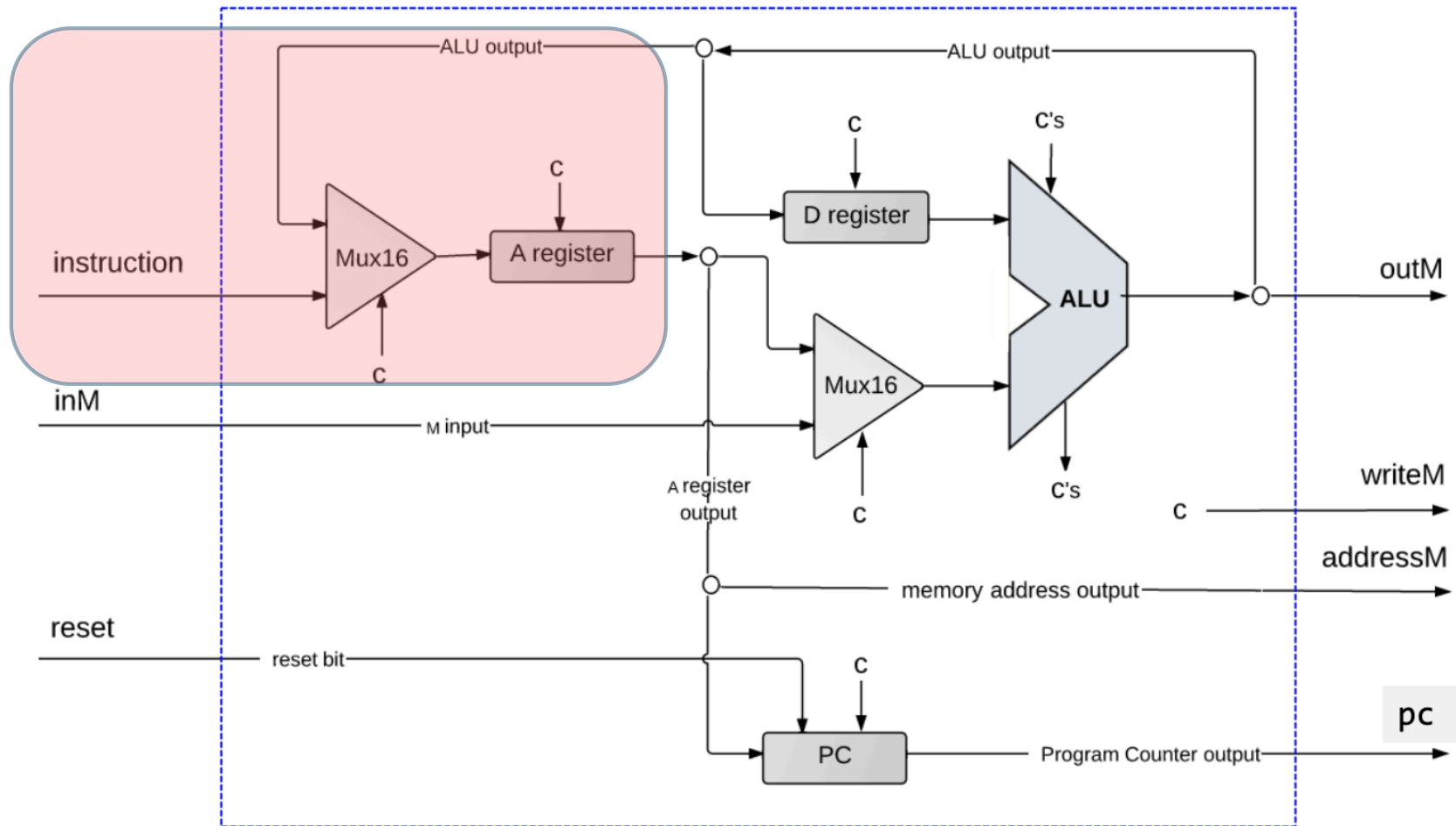


- C คือ control bit

Hack CPU Implementation



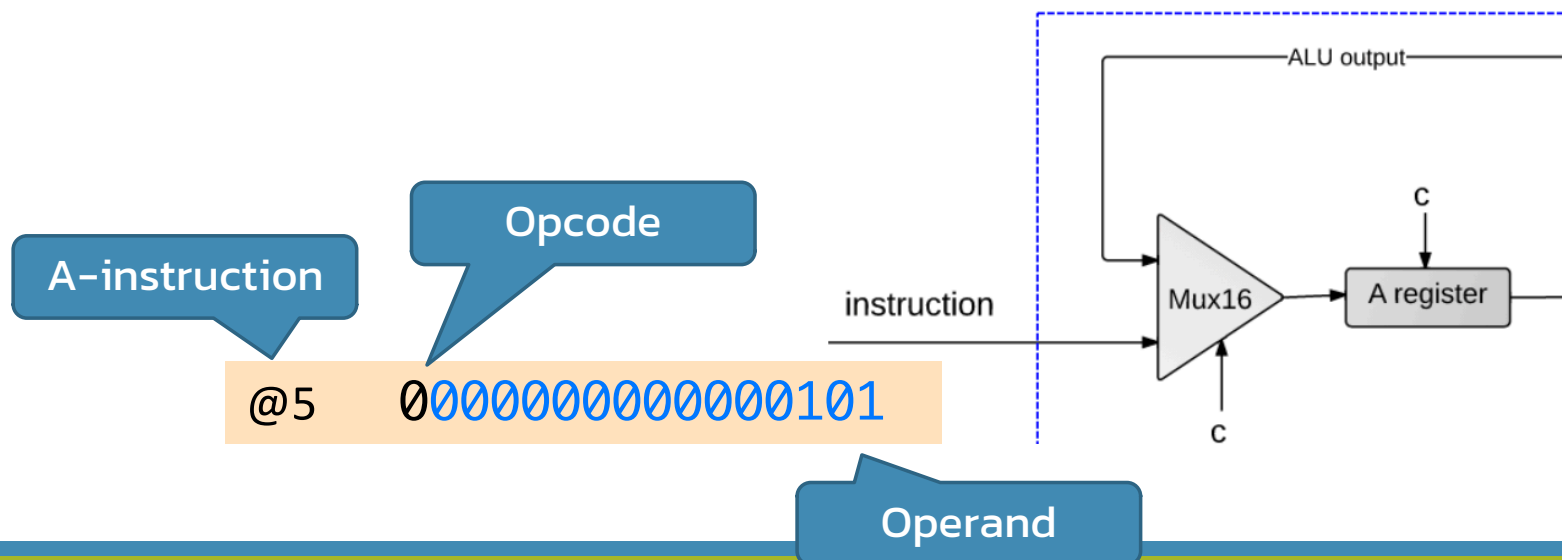
Hack CPU Implementation



- C คือ control bit

CPU : A-instruction

- หลังจากถอด mnemonic เป็น instruction
- รหัสไบนารีขึ้นต้นด้วยบิต 0 (b_{15})
- A-instruction มี 2 ส่วน
 - opcode (ขนาด 1 บิต)
 - operand (ขนาด 15 บิต)



HDL : A-instruction

```
CHIP CPU {  
  IN ...;  
  OUT ...;
```

```
  PARTS:
```

```
    Not(in=instruction[15], out=n);  
    Mux16(a=outtM,b=instruction,sel=ni,out=i);  
    Or(a=ni,b=instruction[5],out=intoA);  
    ARegister(in=i,load=intoA,out=A,out[0..14]=addressM);
```

```
    ...
```

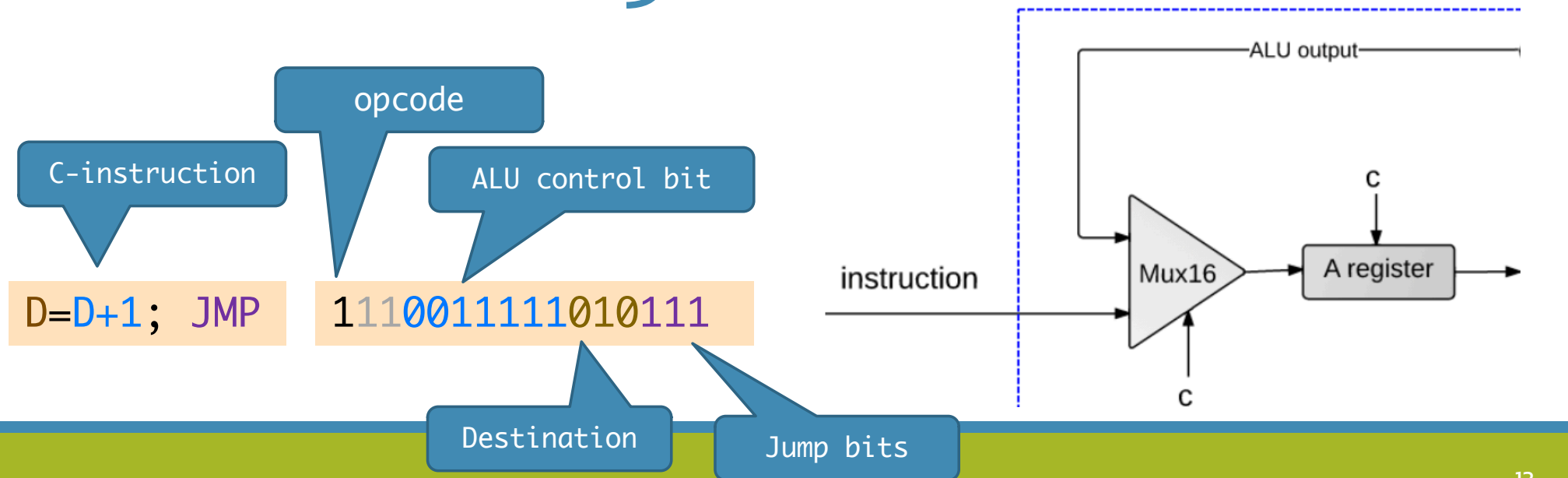
```
    ...
```

```
}
```

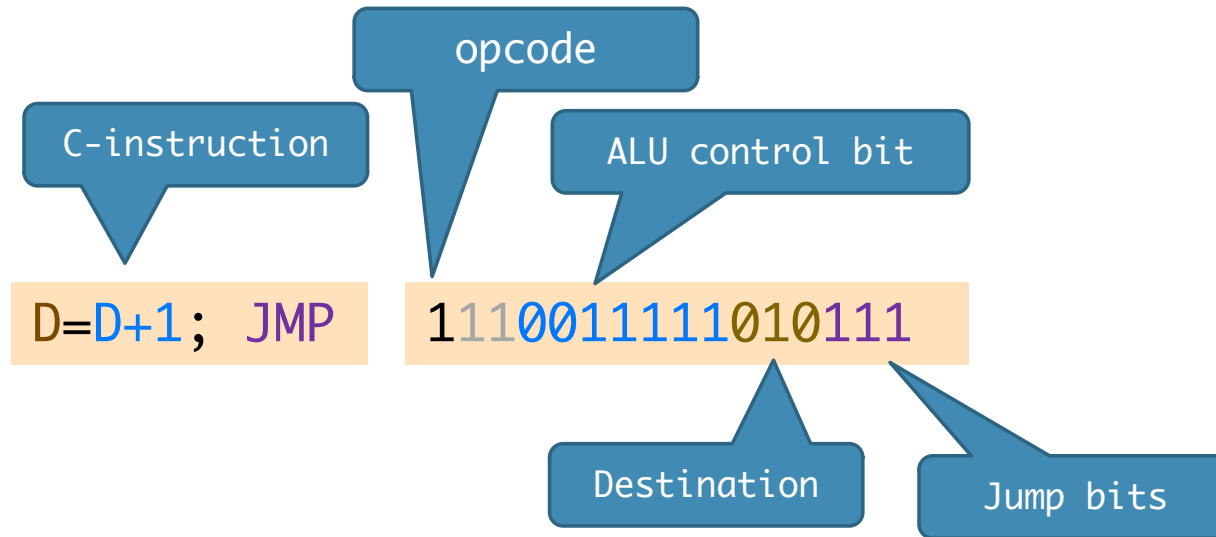
CPU : C-instruction

- รหัสไบนารีเริ่มต้นด้วยบิต 1 (b_{15})
- A-instruction มี 2 ส่วน
 - opcode (ขนาด 1 บิต)
 - ALU control bits
 - Destination load bits
 - Jump bits

Operand

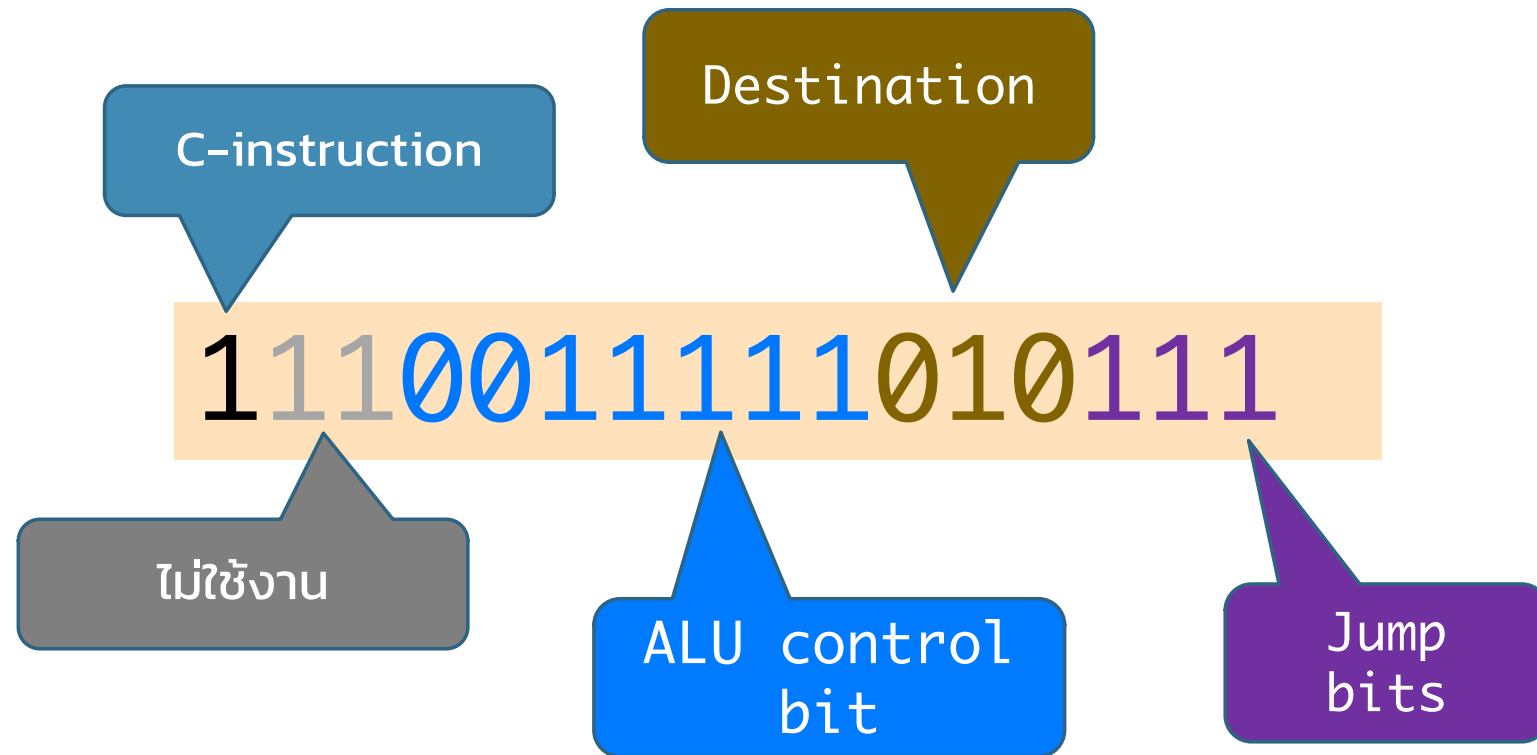


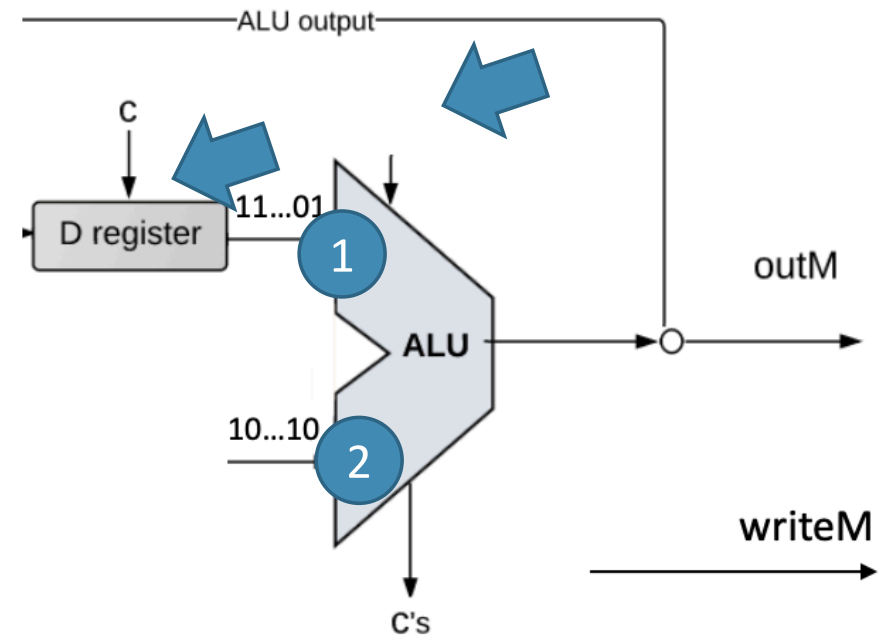
CPU operation : C-instruction



- CPU handling of C-instruction
 - ถอด instruction bit
 - อ่าน ALU control bits
 - อ่าน Destination bits
 - อ่าน Jump bits
- เชื่อมต่อแต่ละบิตกับ chip-part
- ALU, Aregister, ..

C-instruction





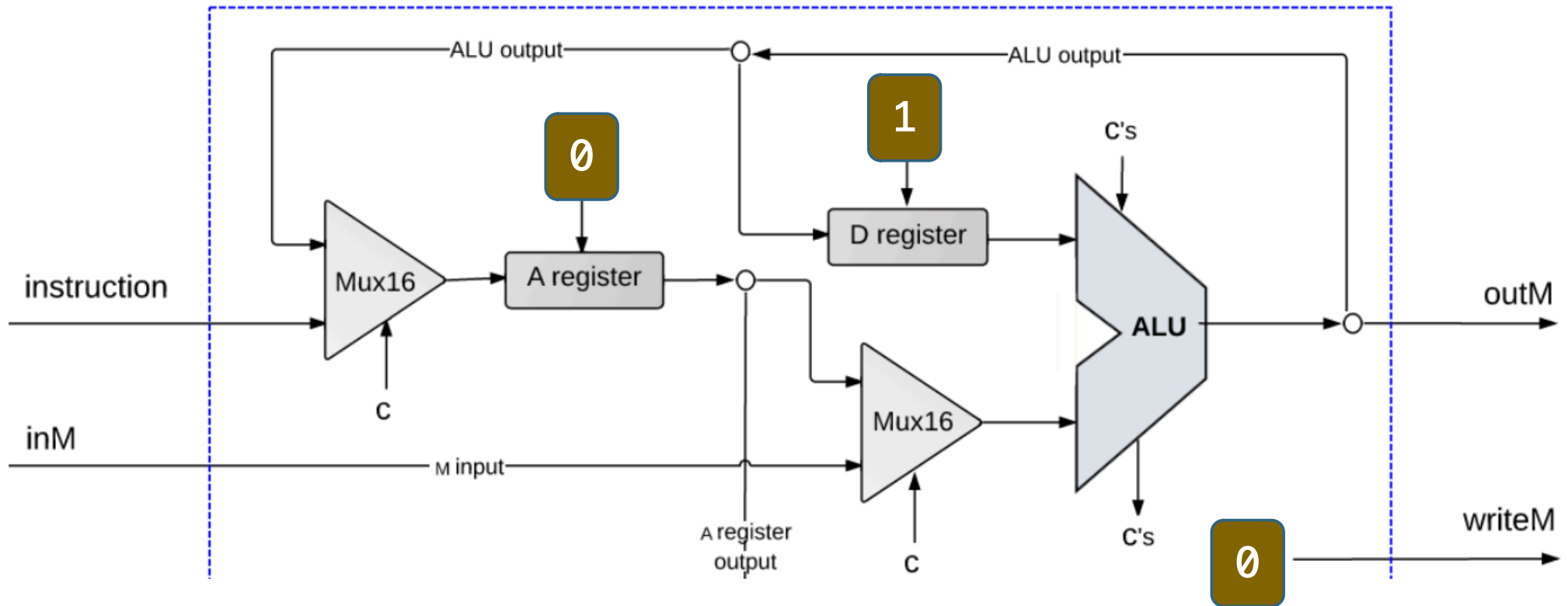
- ALU data input

- **Input 1** : הרח Dregister
- **Input 2** : הרח
 - Aregister
 - data memory

- ALU control bit

- Control bits
- הרח Instruction

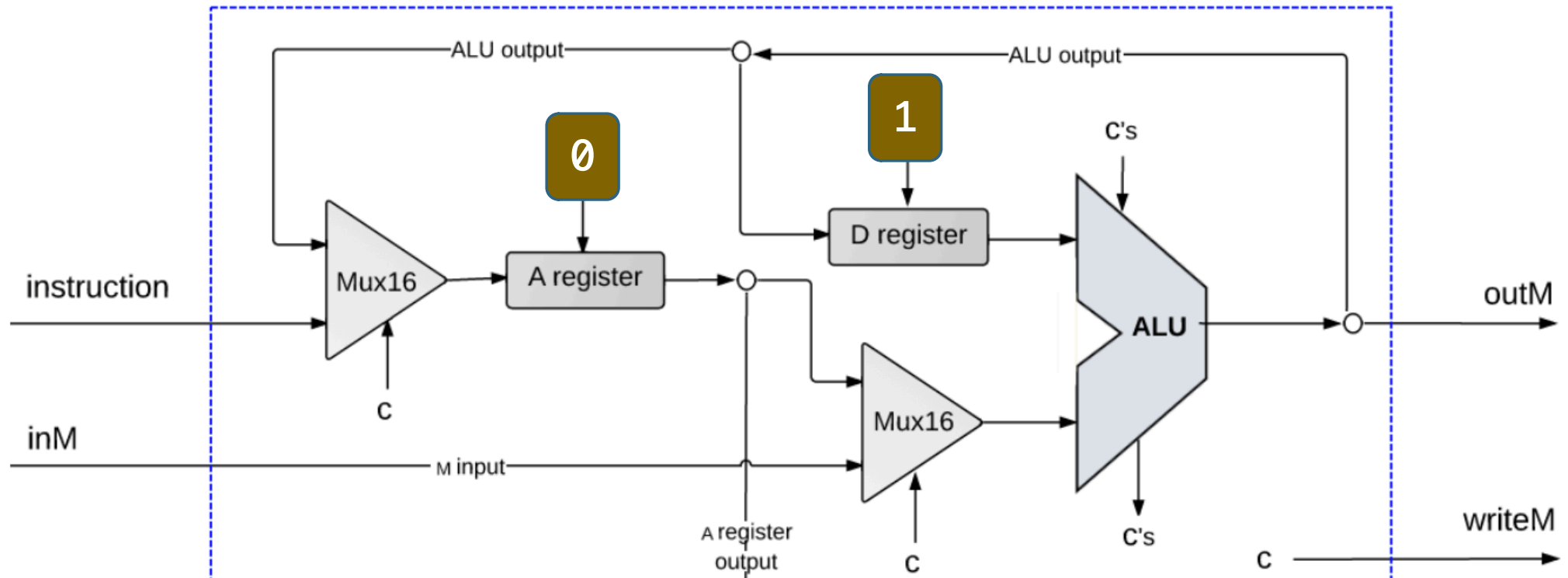
ALU Output



1110011111010111

Destination

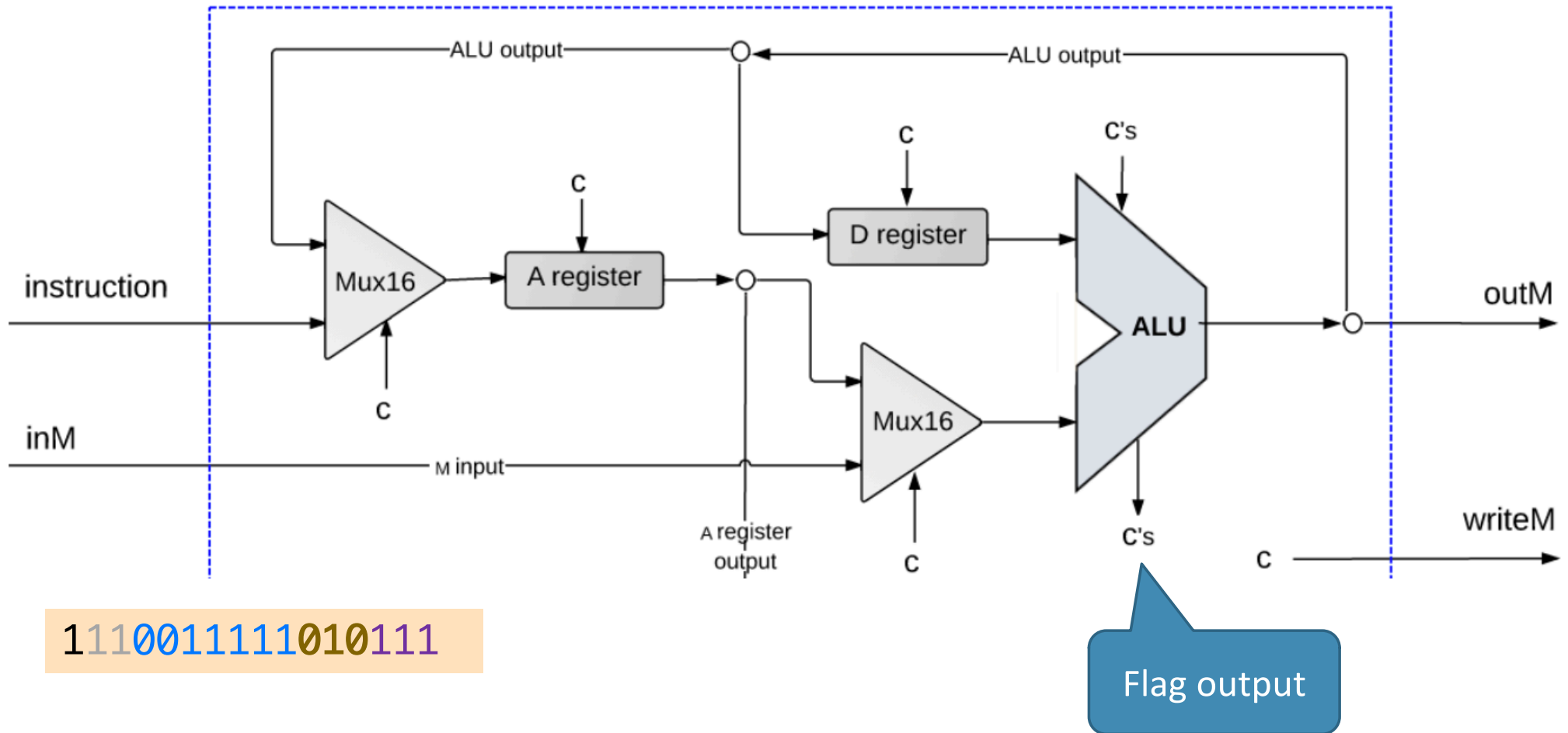
คำสั่ง ALU : Output



1110011111010111

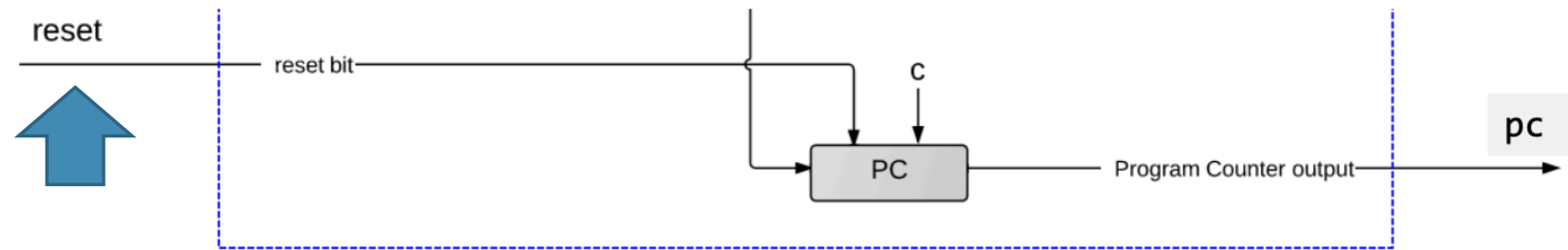
Destination

ALU Output



1110011111010111

CPU : control



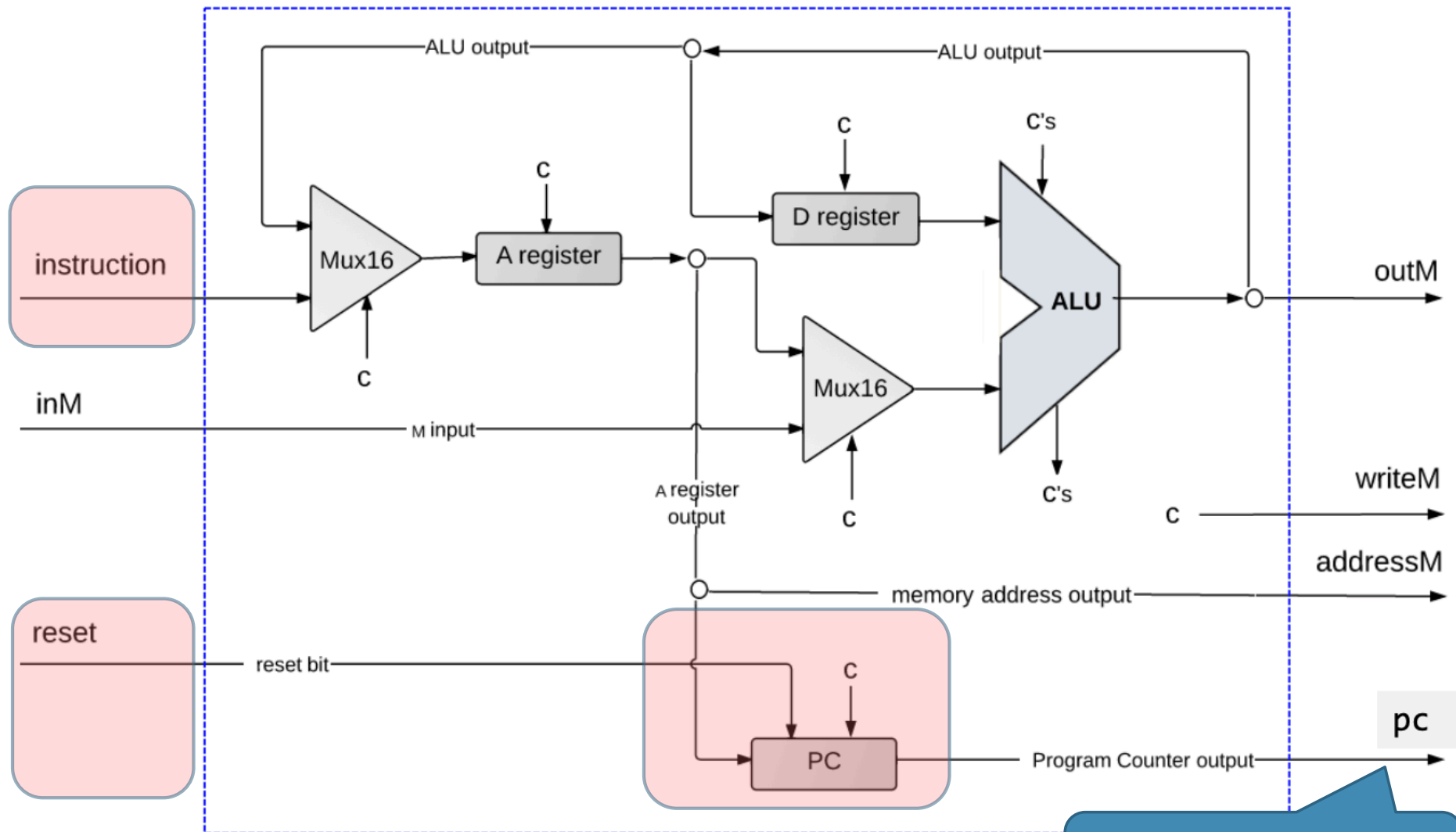
- CPU reset

CPU : reset

- ขณะคอมพิวเตอร์กำลังรันโปรแกรม
- เมื่อกด reset ทำให้โปรแกรมเริ่มต้นรันใหม่

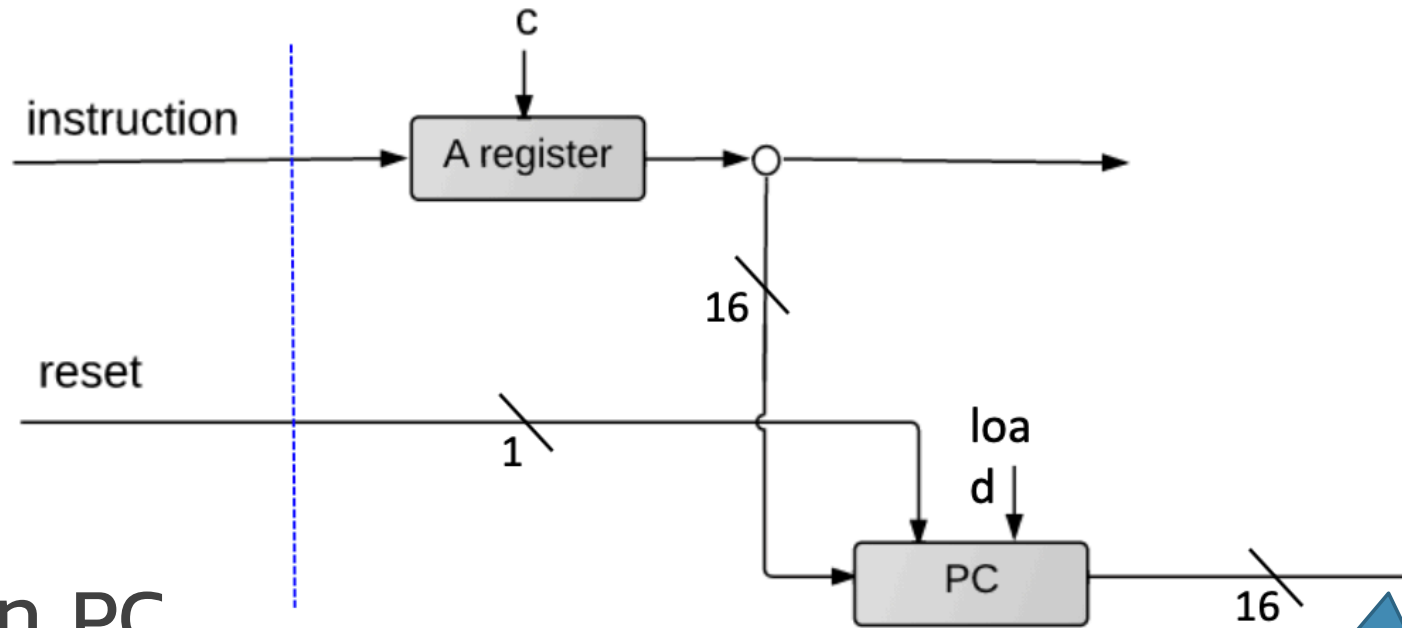


CPU : ส่วนเกี่ยวข้องกับการ reset



Address ของ instruction ต่อไป

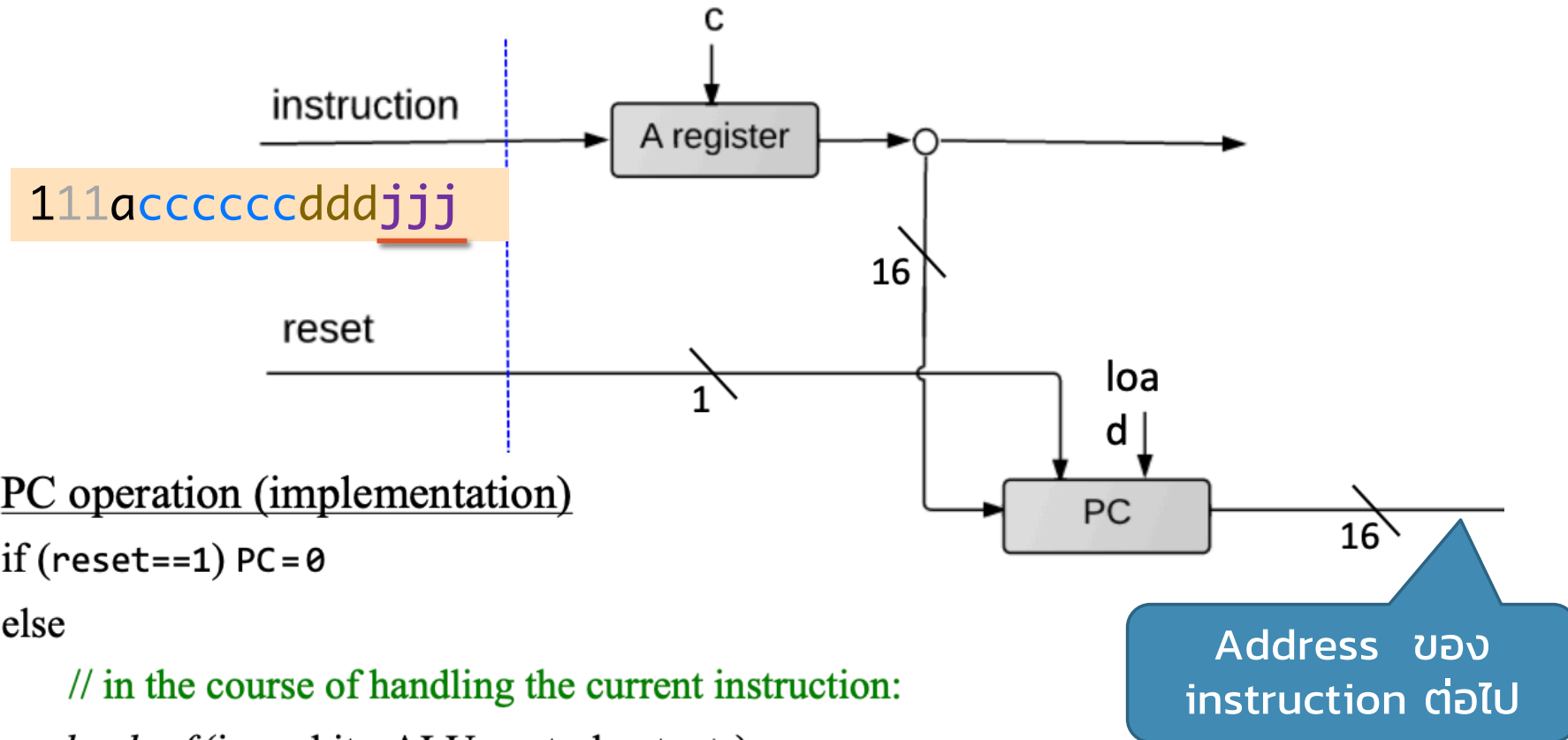
CPU operation: control



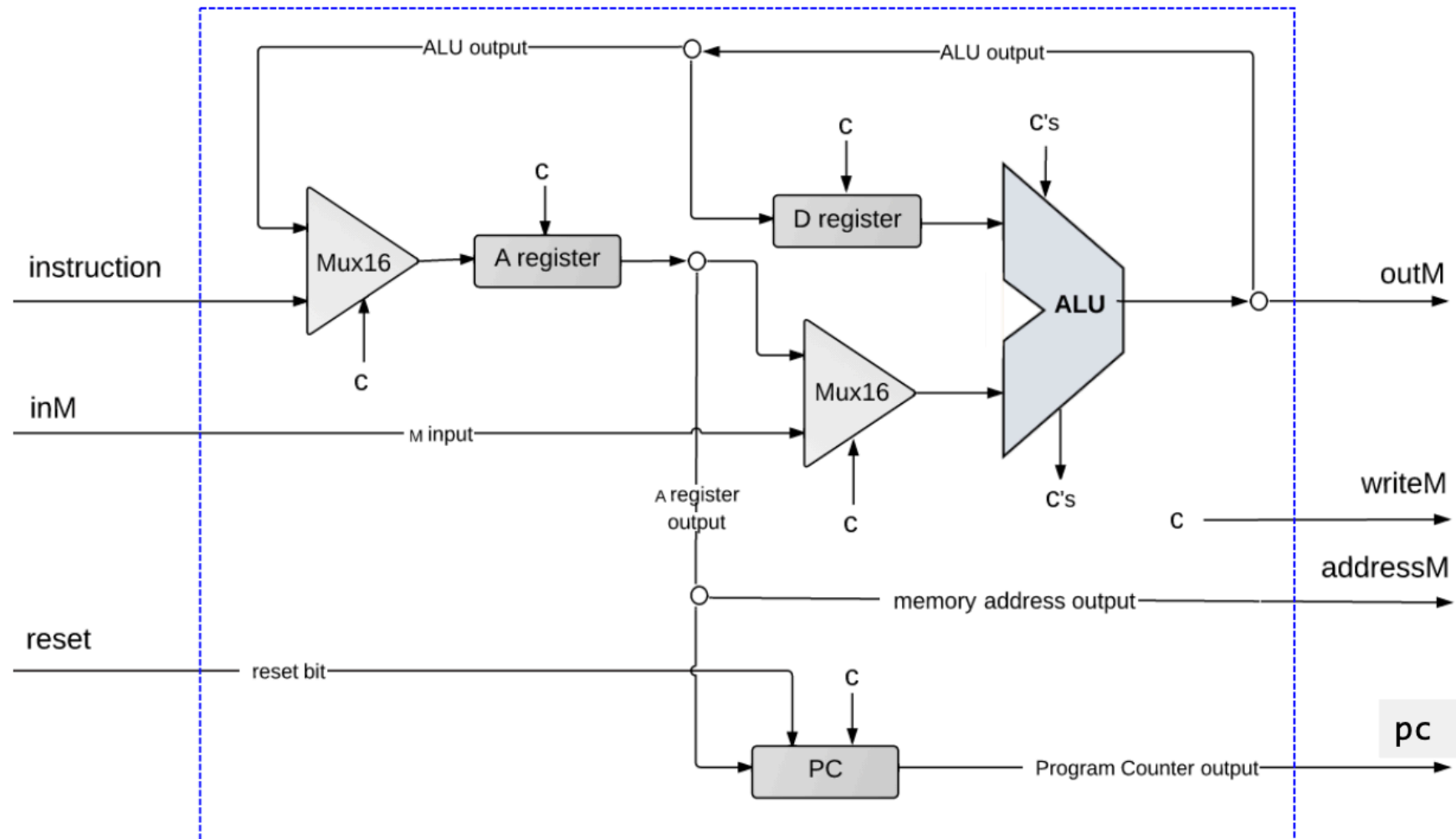
- เอาท์พุท PC

- reset: $PC = 0$
- no jump: $PC++$
- goto: $PC=A$
- Condition goto:
 - if(condition) $PC=A$ else $PC++$

CPU operation: control



Hack CPU implementation



Coming up: W5.4
แฮกคอมพิวเตอรื

