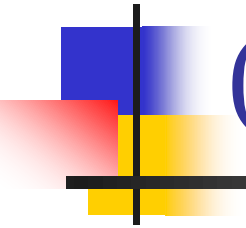
Lab 02

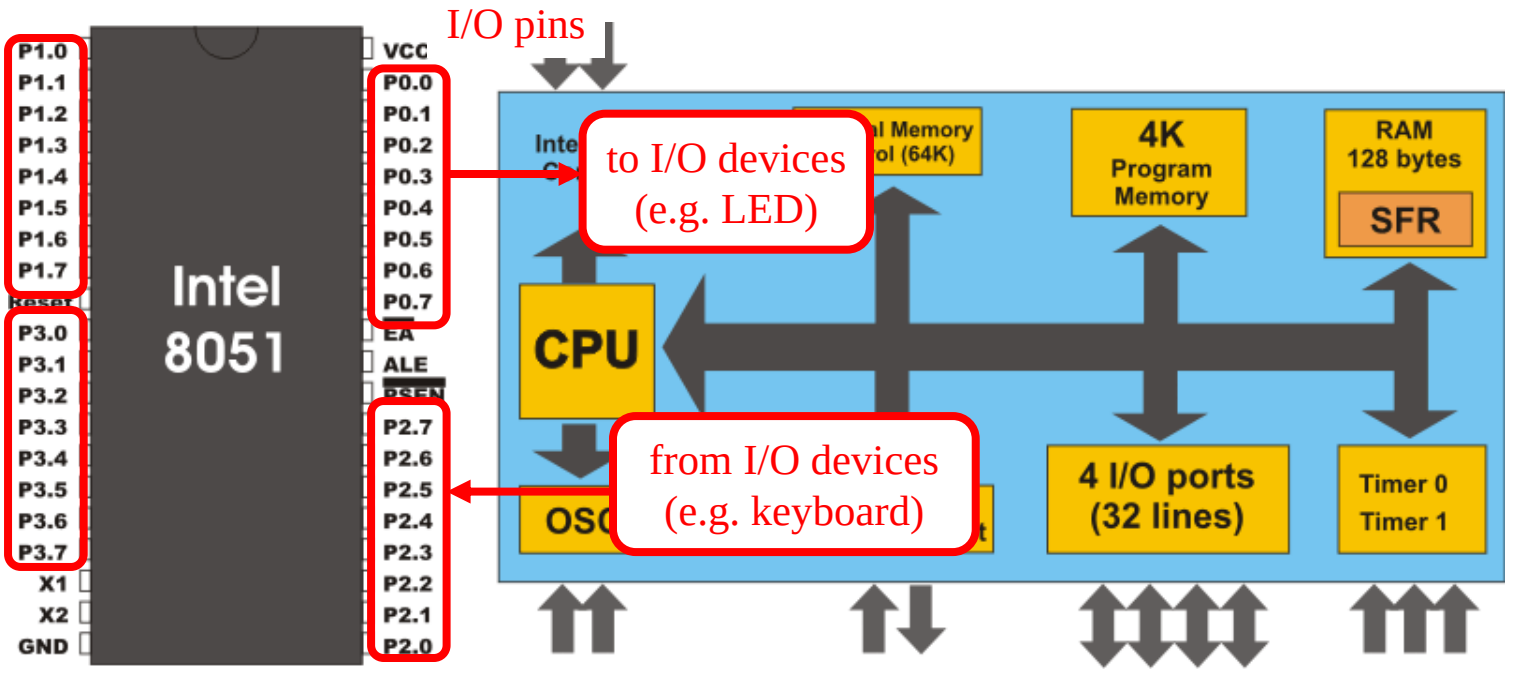
General Purpose Digital I/O (GPIO)



Objectives of this lab

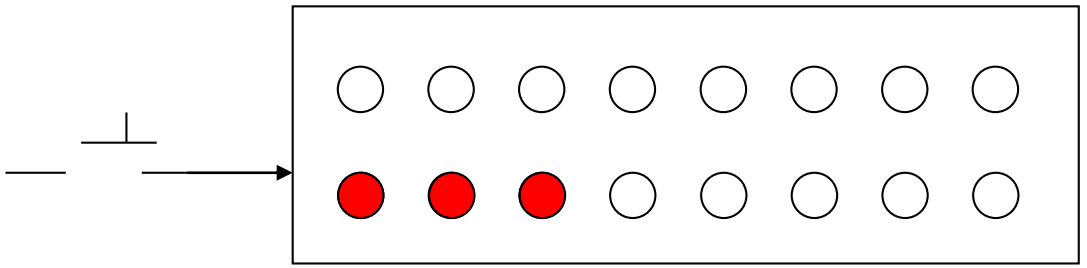
(1) To build up your imagination on how a program affects hardware signals

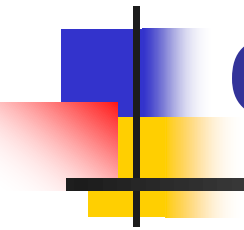
(2) To learn how to send/receive signals from an application processor to external devices through I/O pads



Your work

- design a LED box
  - initial: all LED off
  - the LED runs some pattern after some button pressed
    - you can design your own pattern

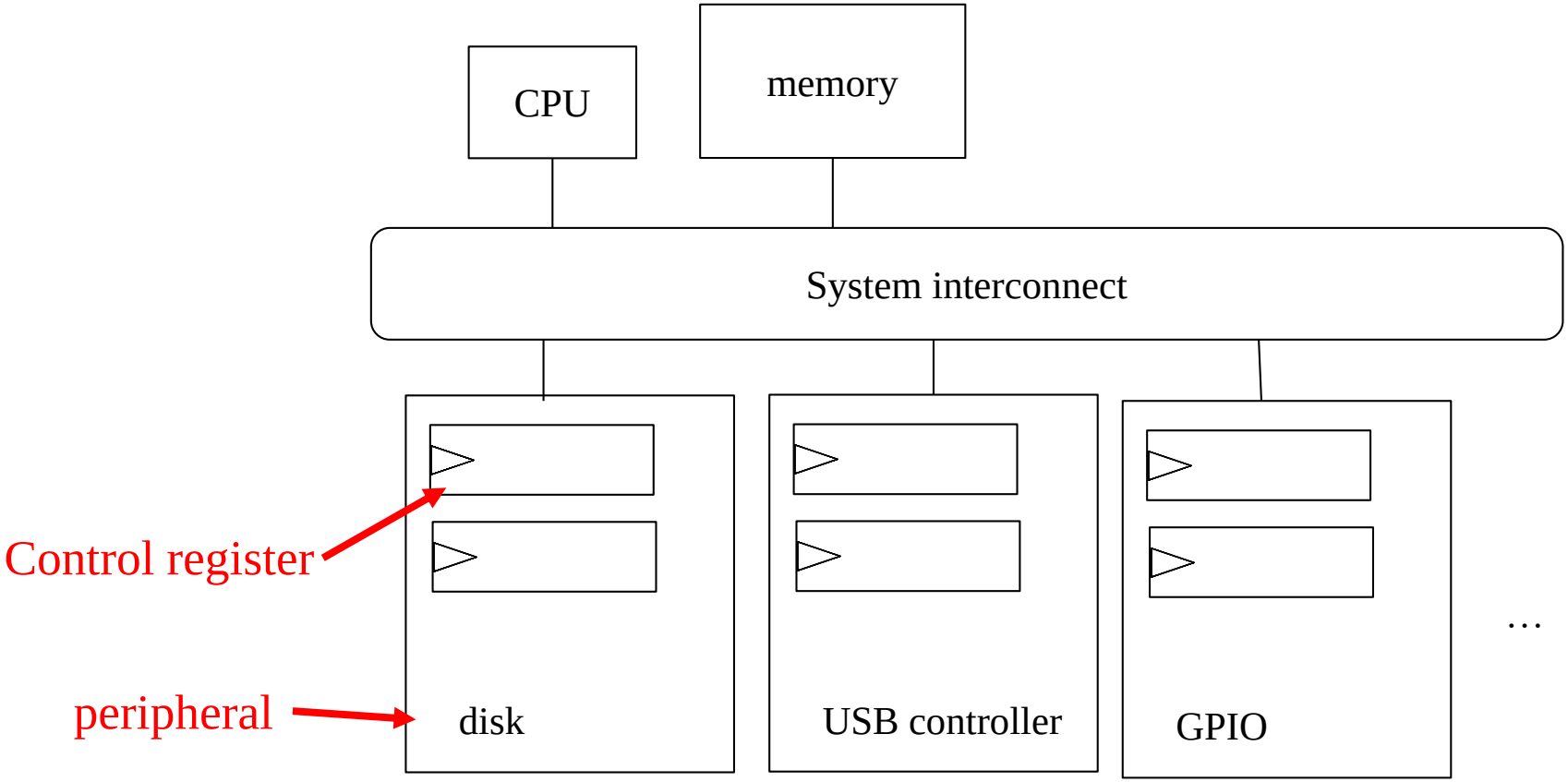




General I/O Control Model

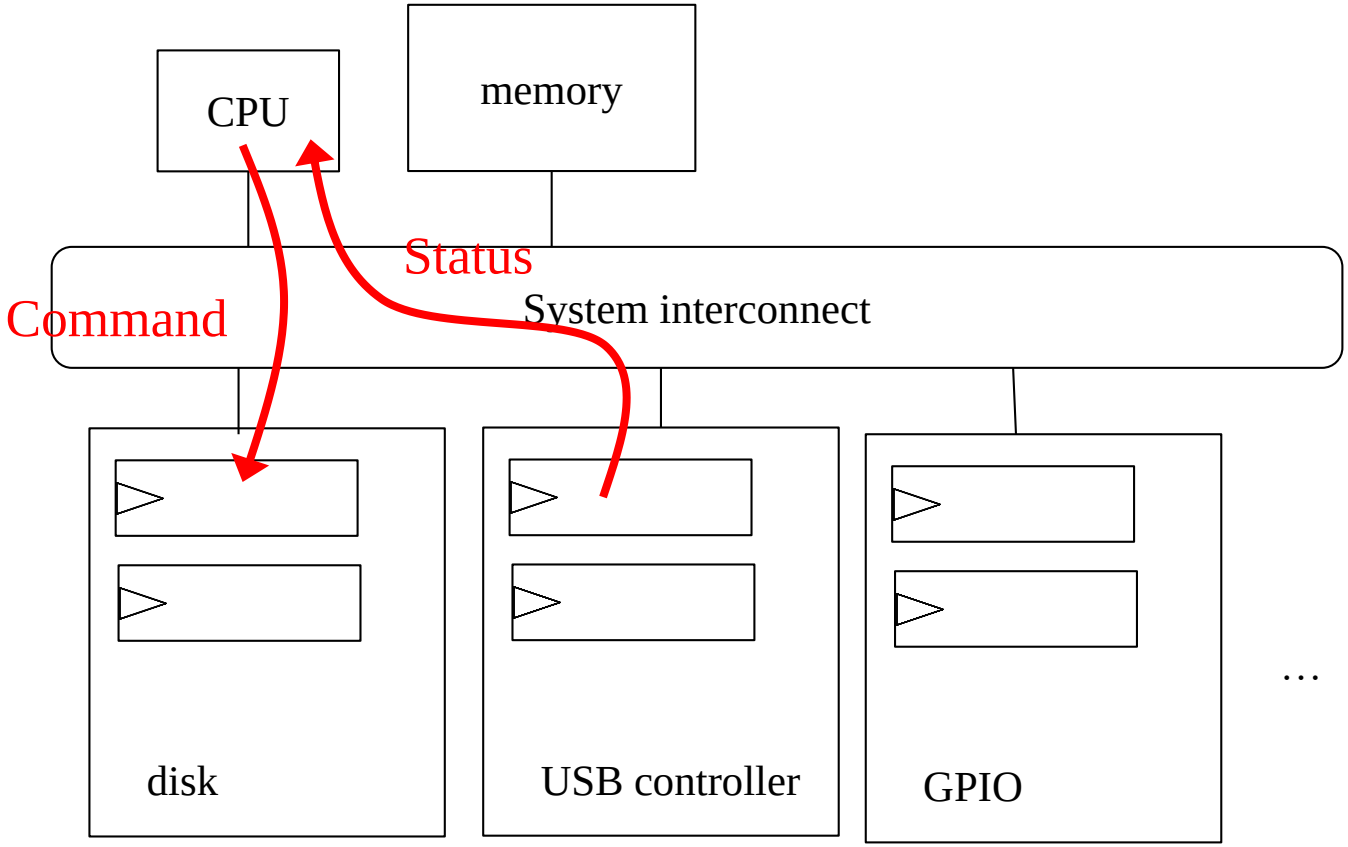
How a processor commands an I/O peripheral

- Through access control registers



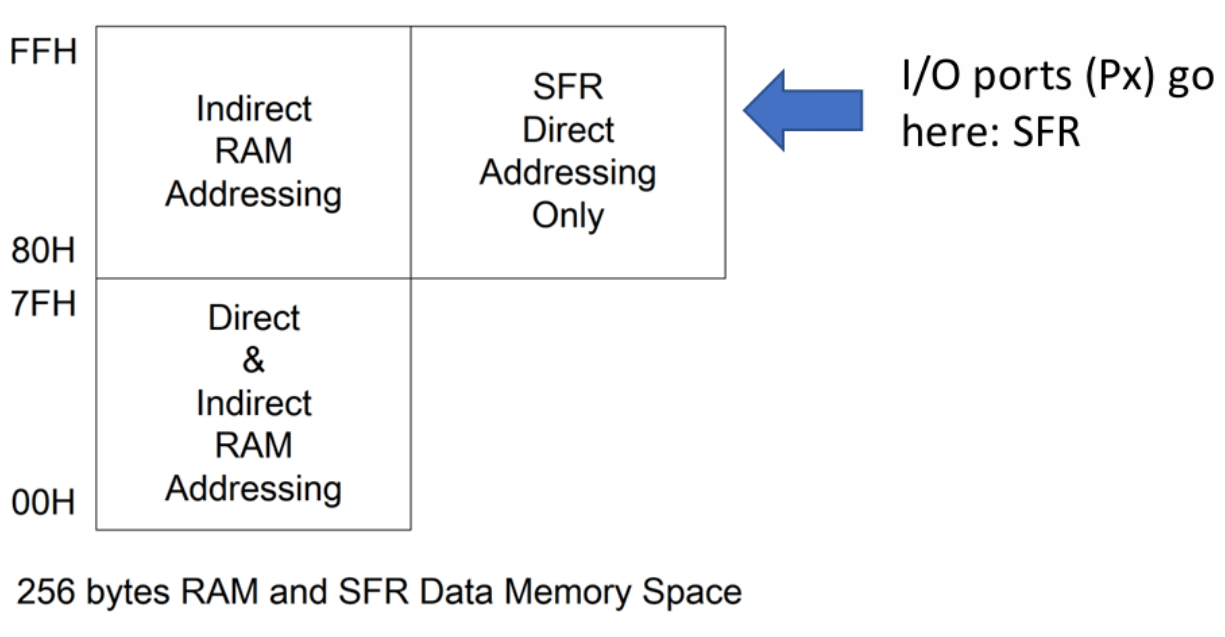
How a processor commands an I/O peripheral

- Through access control registers



How to access control registers: the memory-mapped I/O

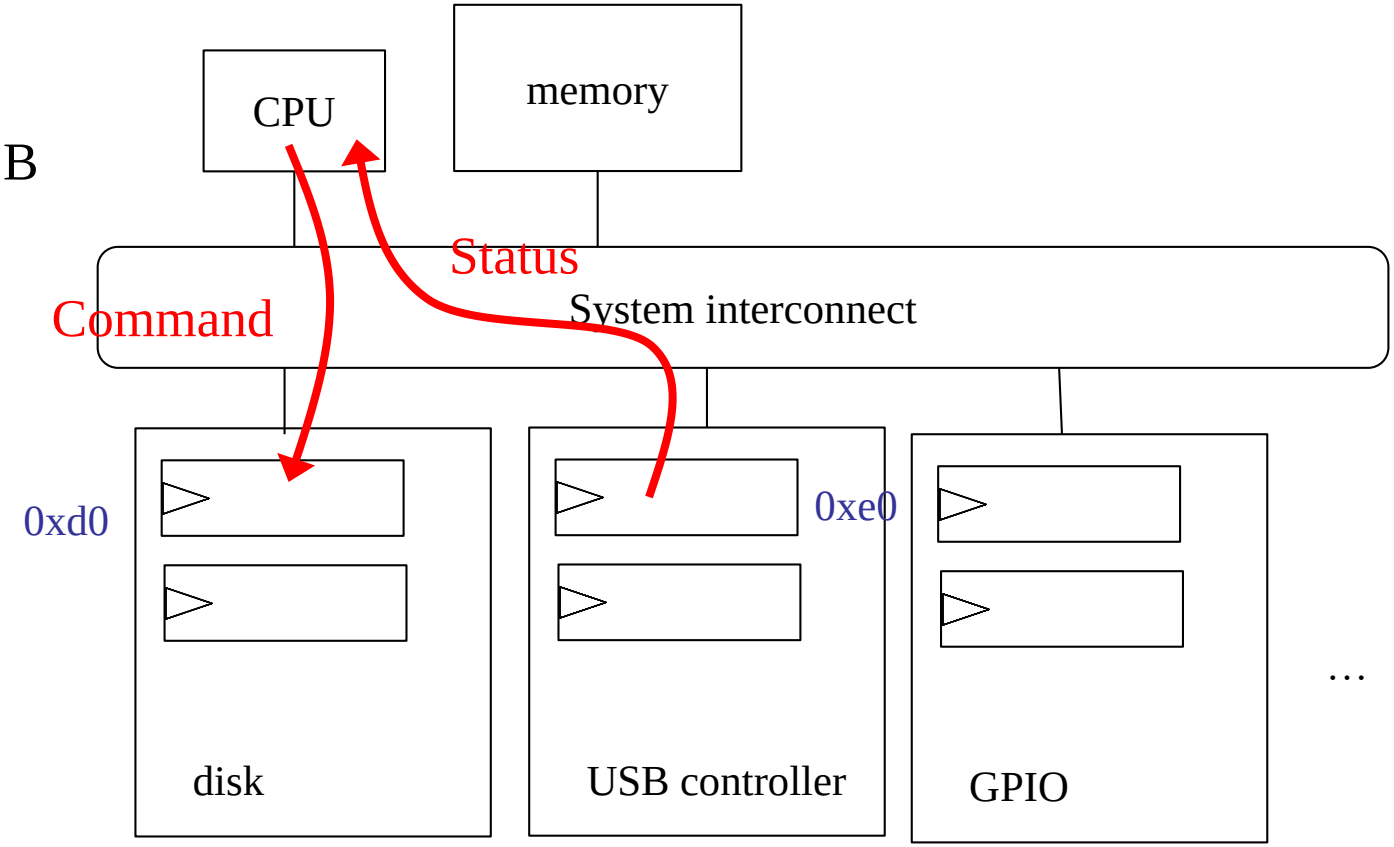
- Part of the addressing space is assigned to control registers
- Each control register is mapped to some memory address



How a processor commands an I/O peripheral

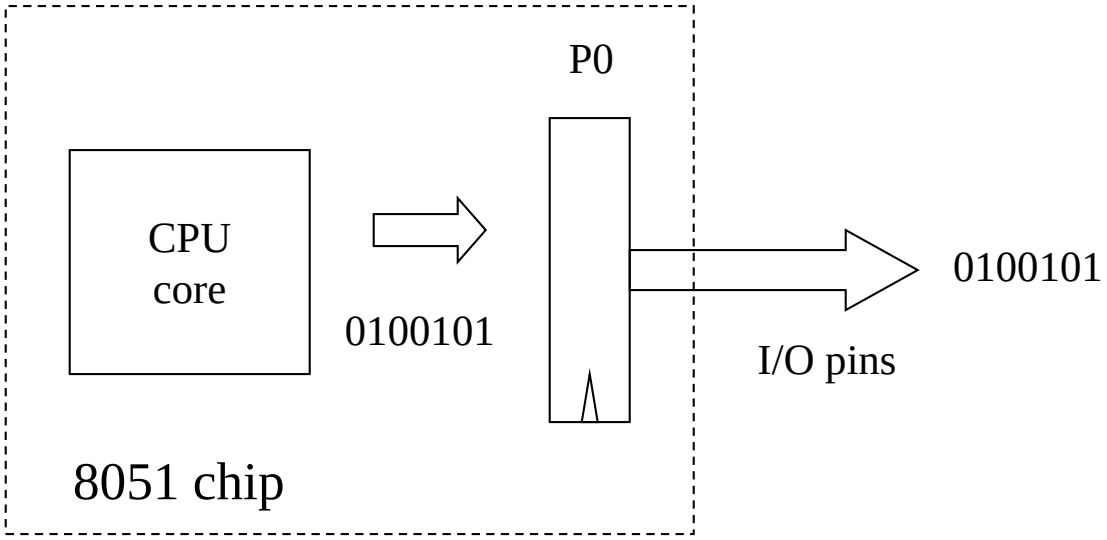
- Through access control registers

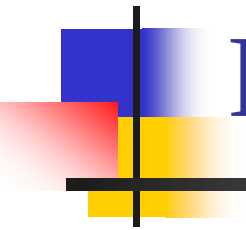
mov 0xd0, #0101B
mov R1, 0xe0



General Purpose Digital I/O

- The processor assigns/examines the logical status of some I/O pins directly





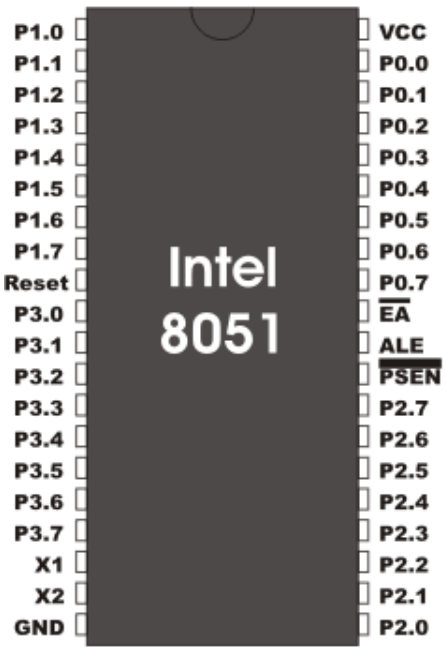
I/O Model of Legacy 8051 Processor

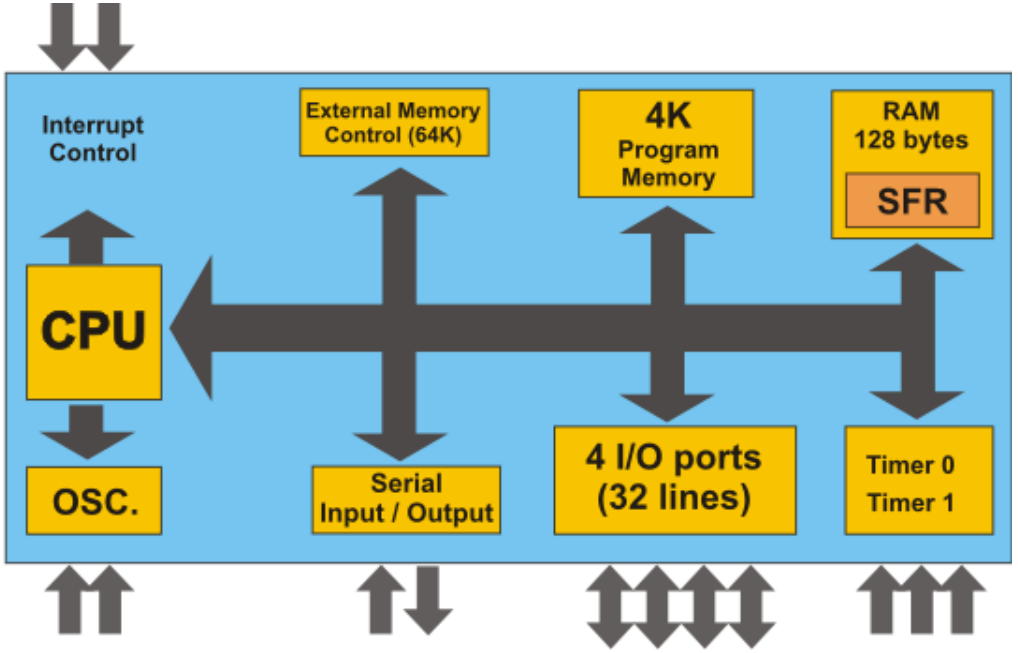
Features of 8051 I/O

(1) Four 8-bit I/O ports P0-P3

(2) Each pin is bidirectional

- sometimes input and sometimes output





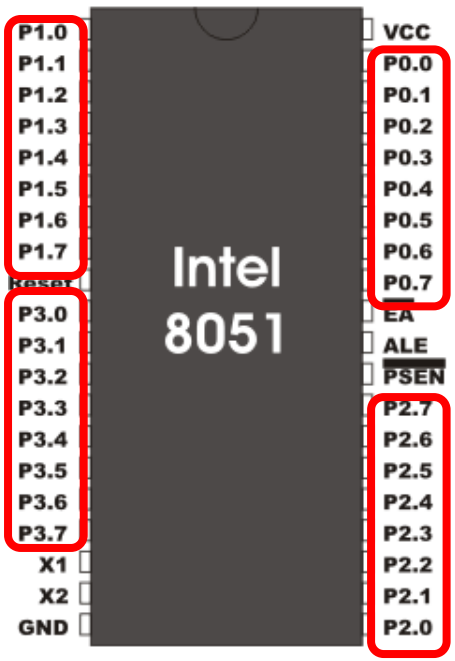
Features of 8051 I/O

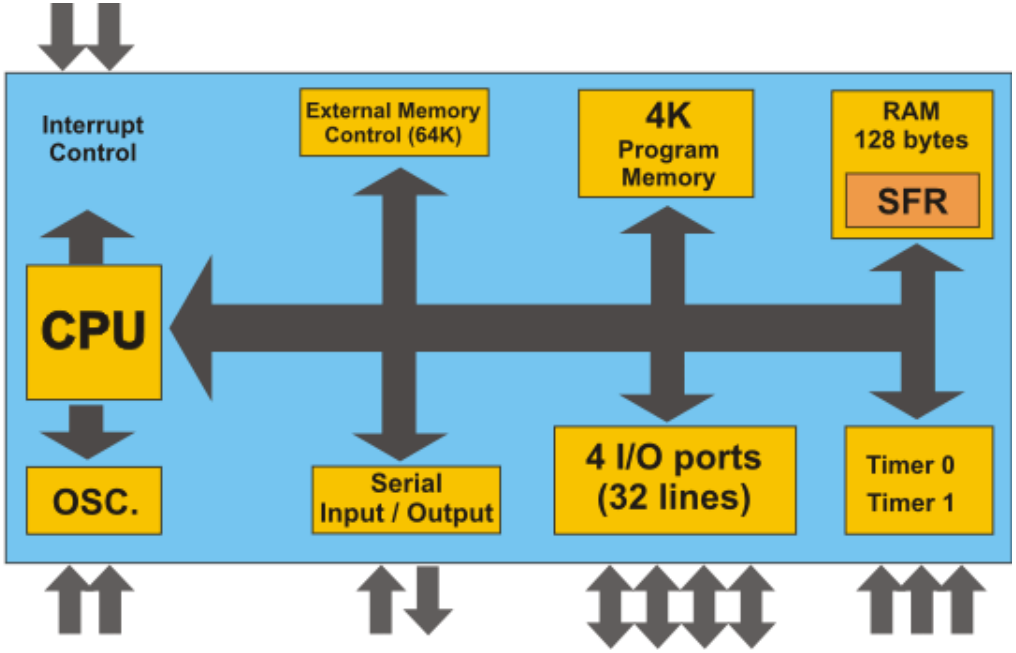
(1) Four 8-bit I/O ports P0-P3

(2) Each pin is bidirectional

- sometimes input and sometimes output

Note: in our 8052-like architecture, an additional P4 can be used. However, the basic principle is the same.



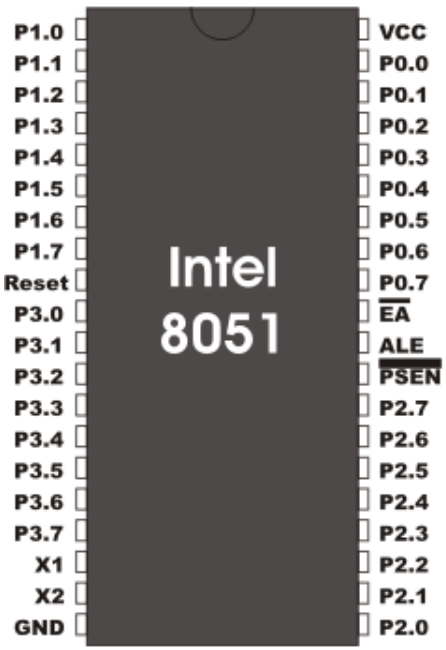


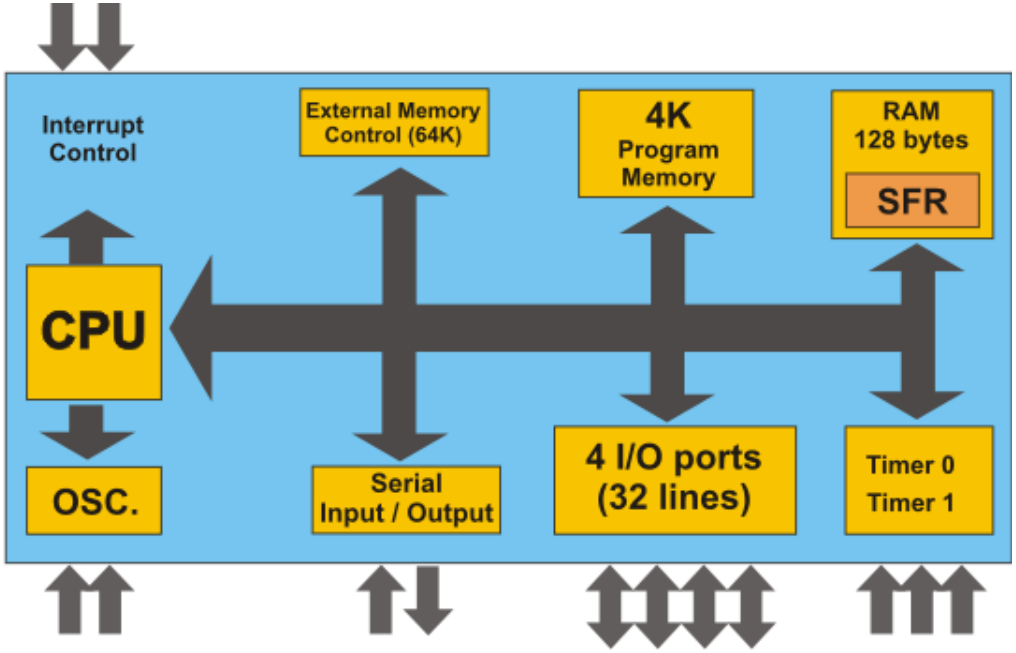
Features of 8051 I/O

(1) Four 8-bit I/O ports P0-P3

(2) Each pin is bidirectional

- sometimes input and sometimes output



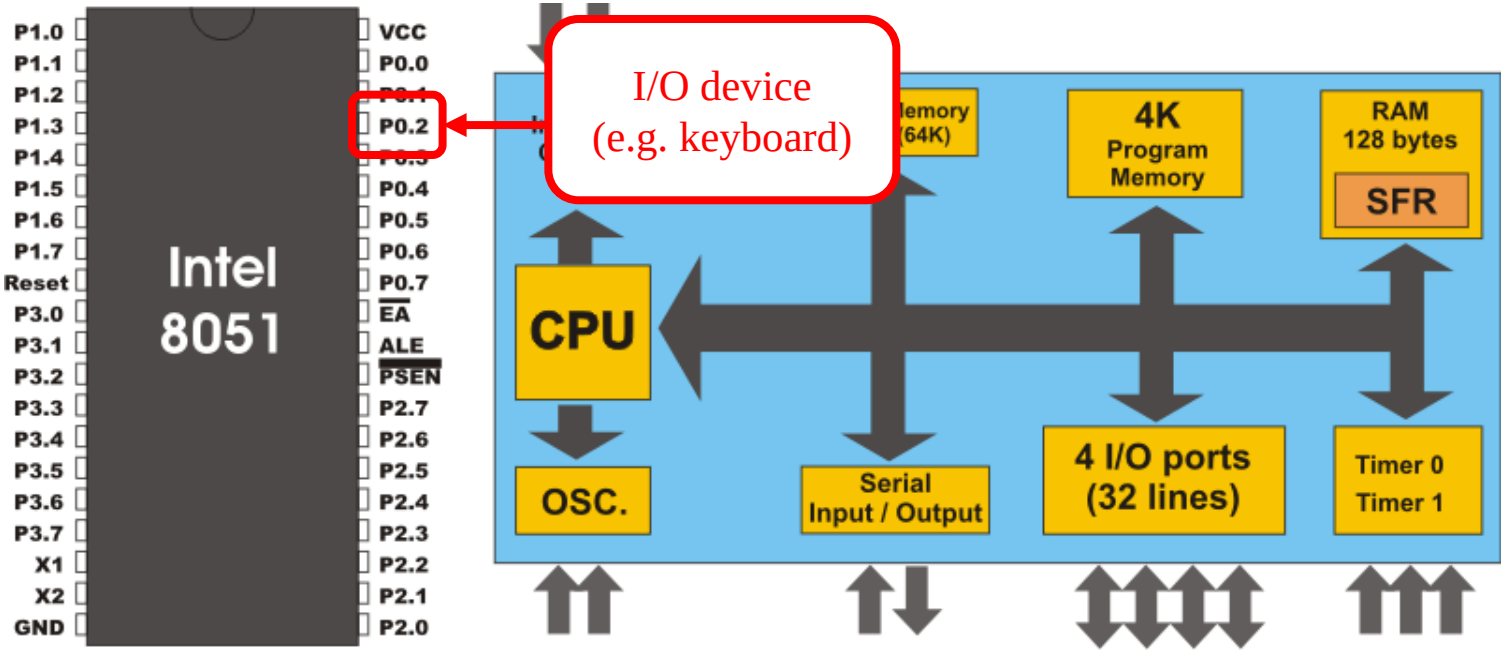


Features of 8051 I/O

(1) Four 8-bit I/O ports P0-P3

(2) Each pin is bidirectional

- Sometimes input and sometimes output

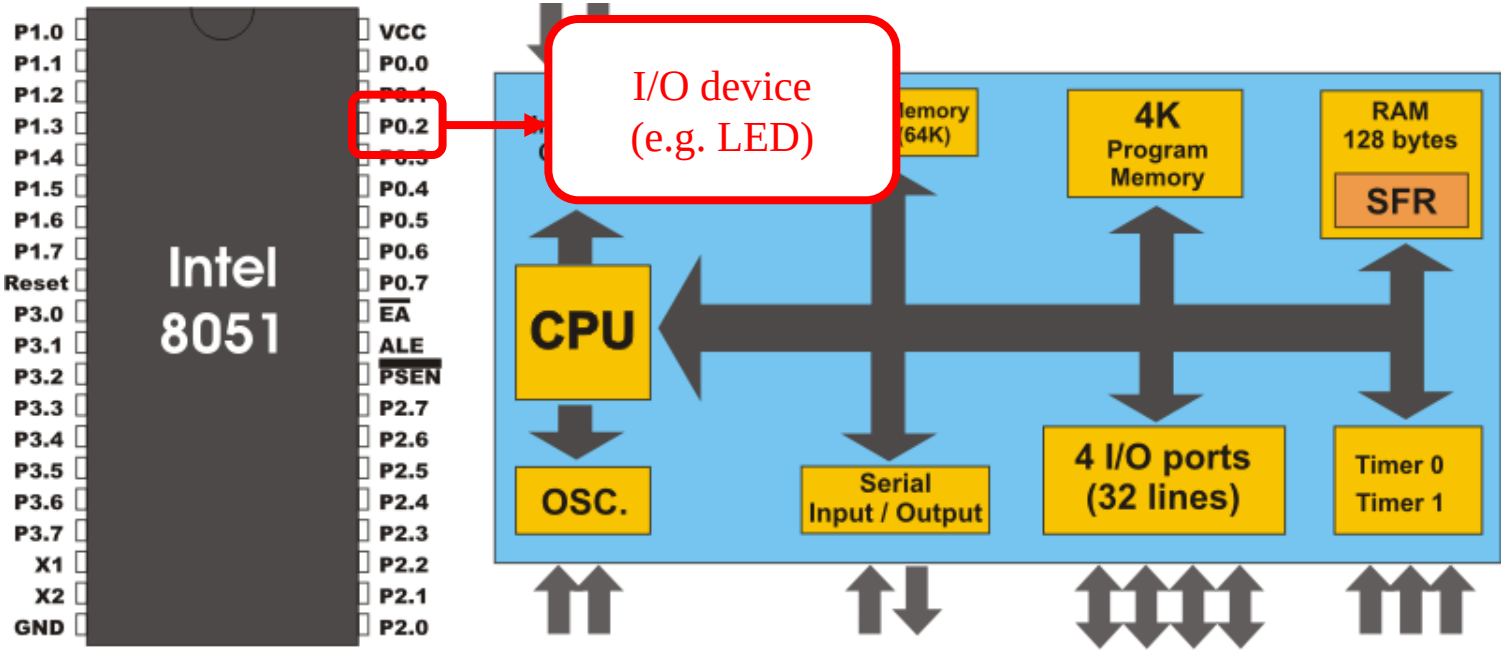


Features of 8051 I/O

(1) Four 8-bit I/O ports P0-P3

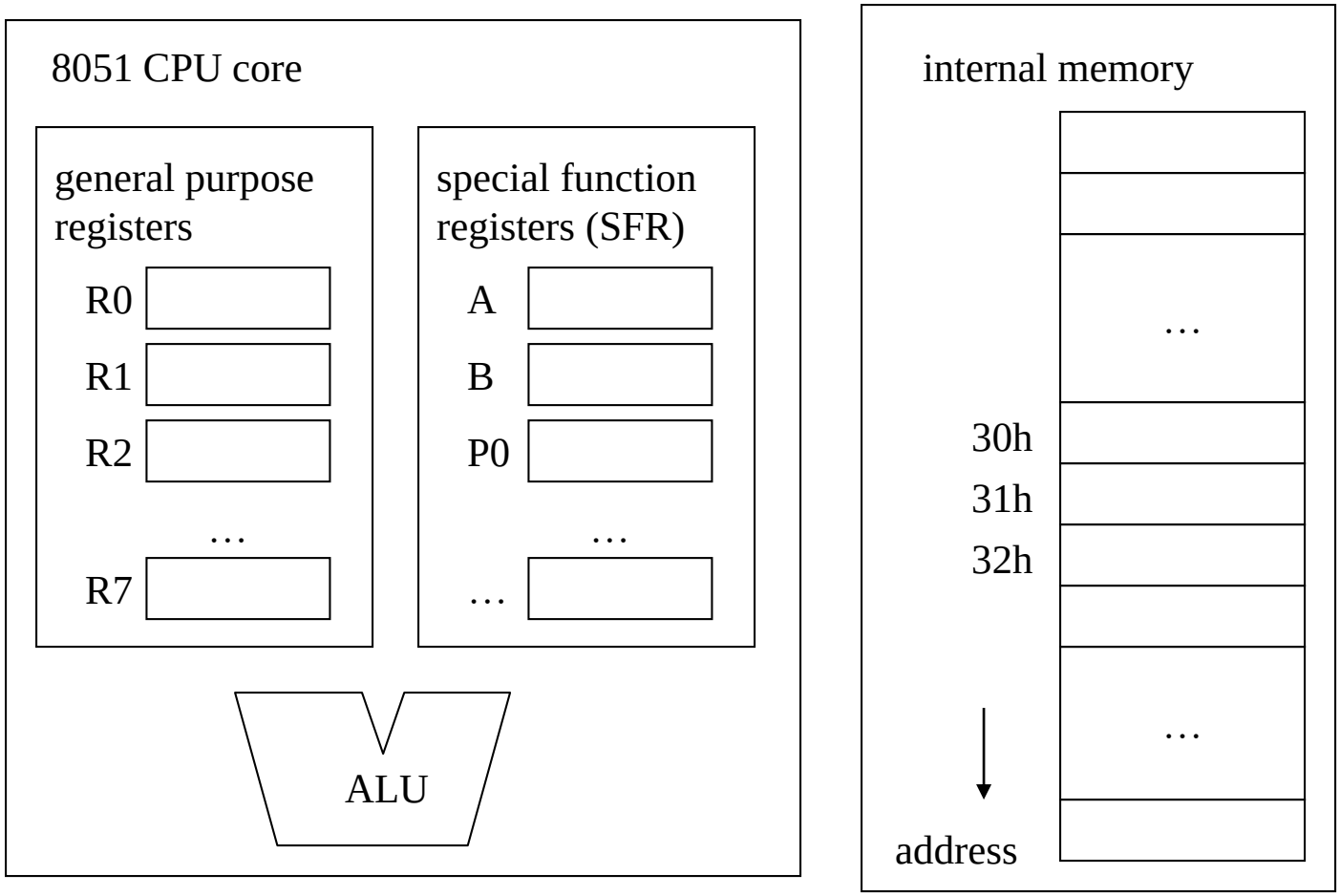
(2) Each pin is bidirectional

- sometimes input and sometimes output



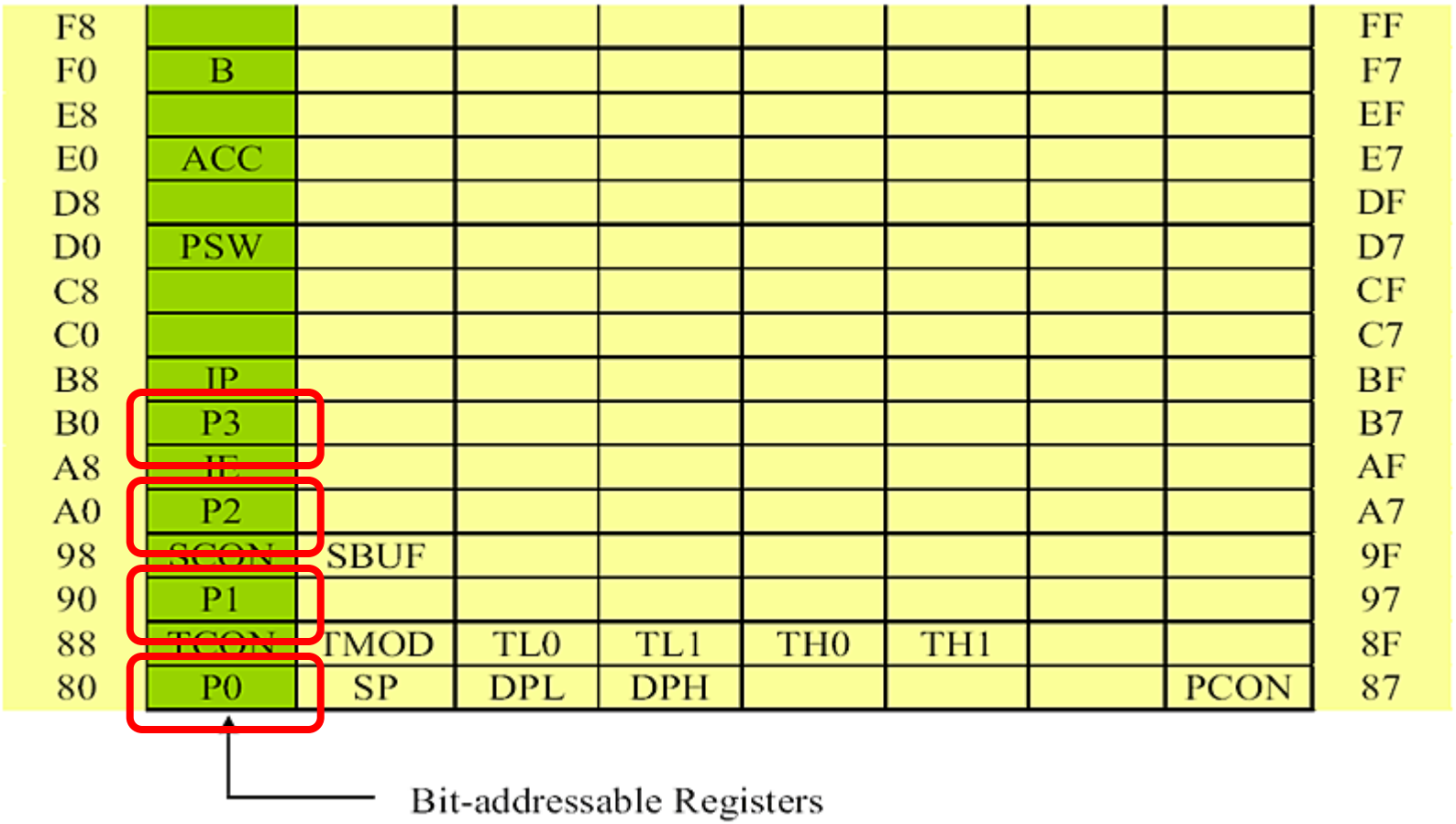
Imagination on 8051 architecture

- Imagine how data flow in the architecture!



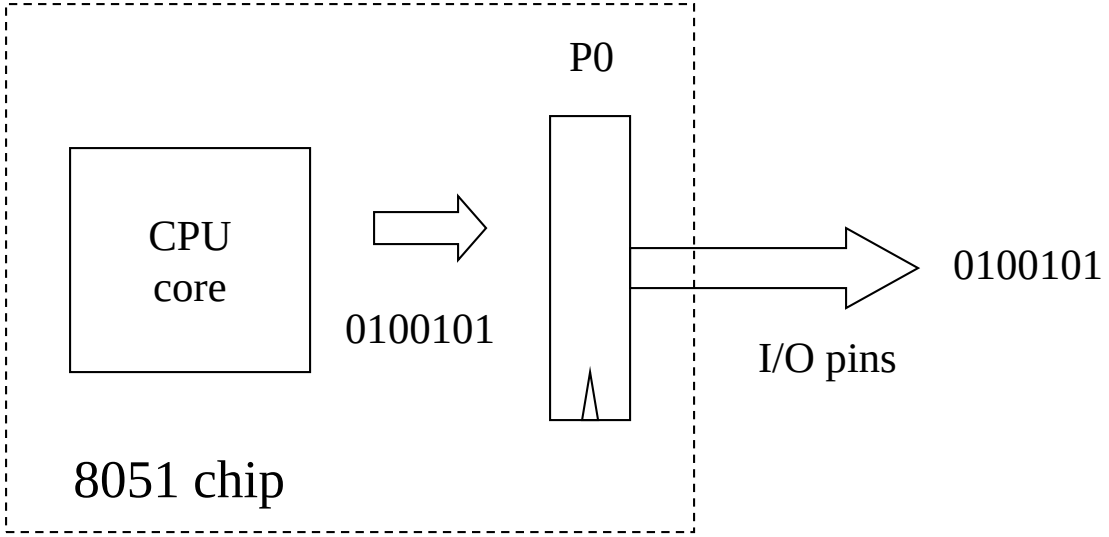
How to program I/O ports?

- through SFRs P0-P3



How 8051 send out dedicated control signals

```
MOV  R0, #01001101B
MOV  P0, R0
```



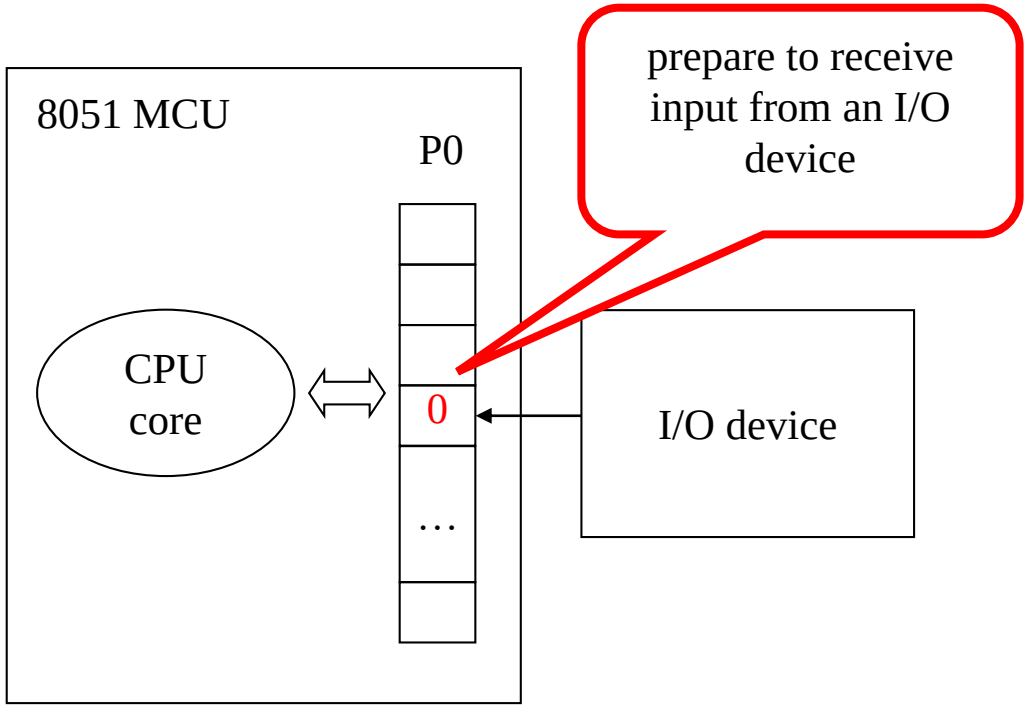
The case of input (receive)

- initial: set a bit (pin) with value 0
- receive (input): wait for the bit to be toggled to be 1

```
P0.3 = 0

//wait unit P0.3 been set to 1
while (P0.3==0);

//action for the I/O event
```



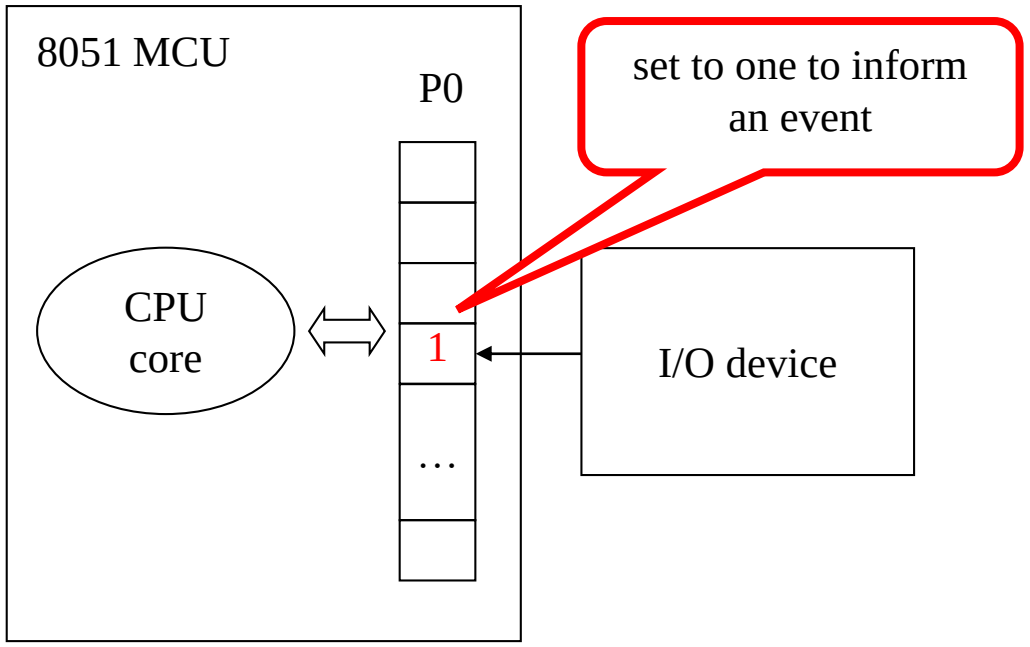
The case of input (receive)

- initial: set a bit (pin) with value 0
- receive (input): wait for the bit to be toggled to be 1

```
P0.3 = 0

//wait unit P0.3 been set to 1
while (P0.3==0);

//action for the I/O event
```



Example 1: wait for a button pressed

Show how to input signal

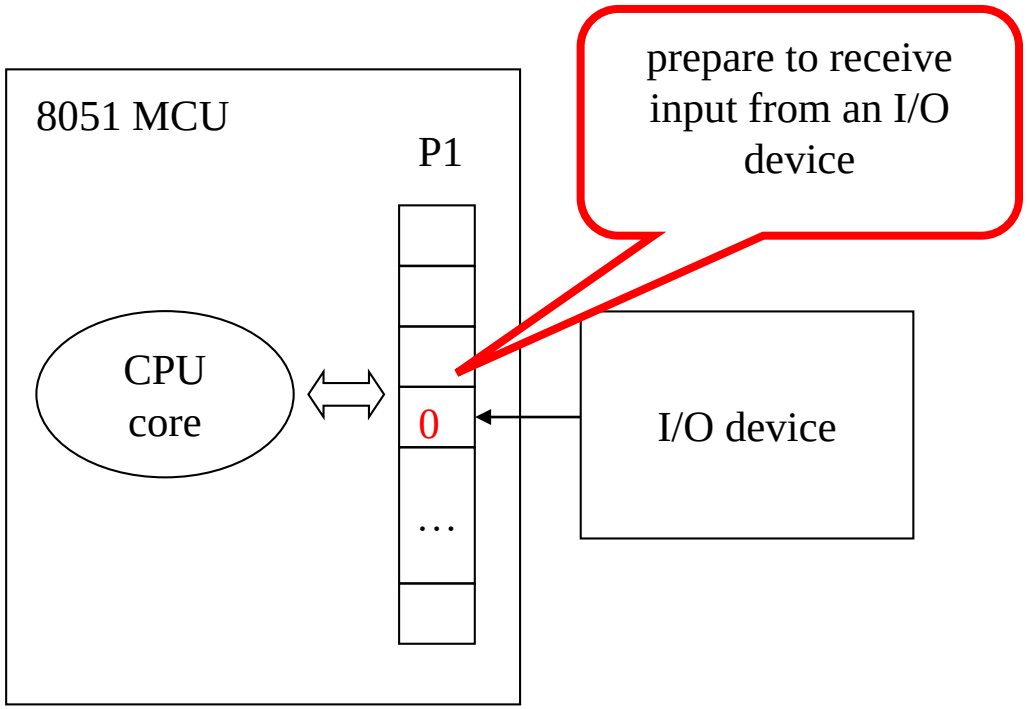
Demo: wait for a button pressed

```
wait:
        A  = P1;
        if (A==0) goto wait;

exit:
        //something after button pressed
```

```
wait:
        mov A, P1
        JZ wait

exit:
        //something after button pressed
```



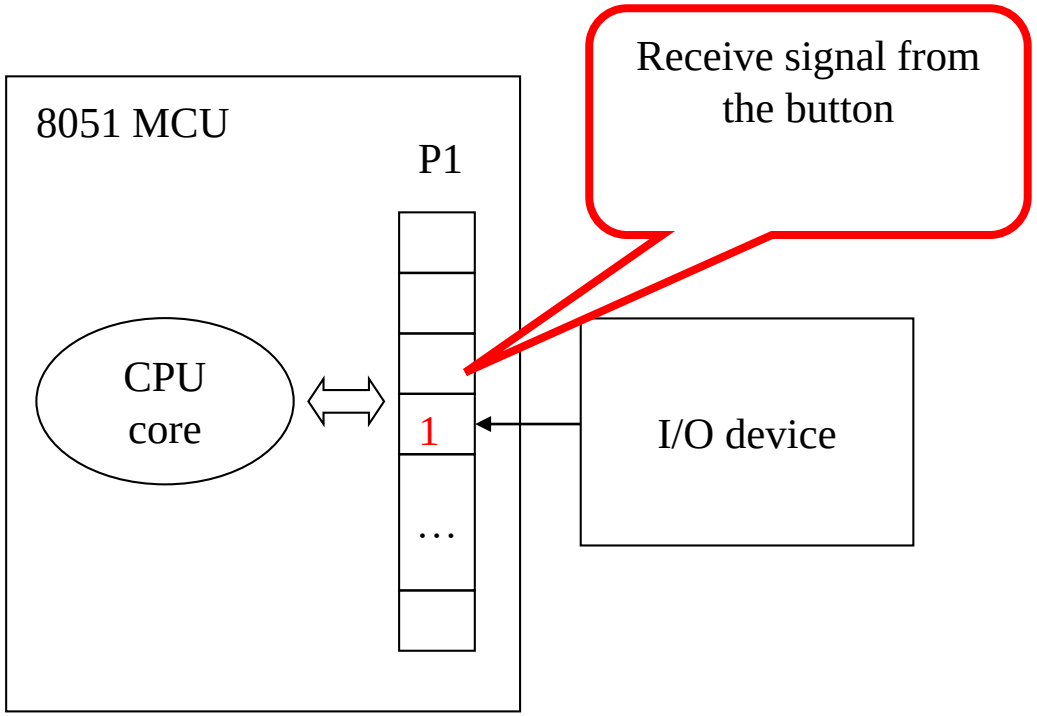
Demo: wait for a button pressed

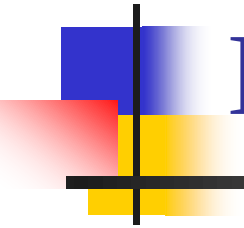
```
wait:
        A  = P1;
        if (A==0) goto wait;

exit:
        //something after button pressed
```

```
wait:
        mov A, P1
        JZ wait

exit:
        //something after button pressed
```





Example 2: make LED run

Show how to output signal

Demo: rotate the LED light

```
MAIN:
        MOV A, #00000001B
        MOV PSW, #00H

Loop:
        MOV     P0, A
        LCALL   Delay
        RR      A
        LJMP    Loop

                MOV     R0, #50
Delay:  MOV     R1, #40
Delay1: MOV     R2, #249
Delay2: DJNZ    R2, Delay2
                DJNZ    R1, Delay1
                DJNZ    R0, Delay
                RET
```

Demo: rotate the LED light

```
MAIN:
            MOV A, #00000001B
            MOV PSW, #00H

Loop:
            MOV     P0, A
            LCALL   Delay
            RR      A
            LJMP    Loop

                    MOV     R0, #50
Delay:      MOV     R1, #40
Delay1:     MOV     R2, #249
Delay2:     DJNZ    R2, Delay2
                    DJNZ    R1, Delay1
                    DJNZ    R0, Delay
                    RET
```

control the LED through content of A

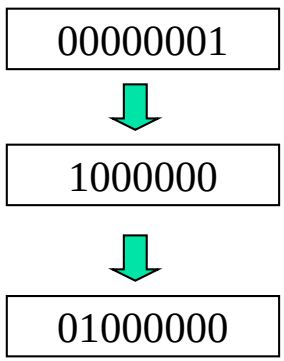
Demo: rotate the LED light

```
MAIN:
        MOV A, #00000001B
        MOV PSW, #00H

Loop:
        MOV     P0, A
        LCALL   Delay
        RR      A
        LJMP    Loop

        MOV     R0, #50
Delay:  MOV     R1, #40
Delay1: MOV     R2, #249
Delay2: DJNZ    R2, Delay2
        DJNZ    R1, Delay1
        DJNZ    R0, Delay
        RET
```

- rotate right (RR) A



Demo: rotate the LED light

```
MAIN:
        MOV A, #00000001B
        MOV PSW, #00H

Loop:
        MOV     P0, A
        LCALL   Delay
        RR      A
        LJMP    Loop

        MOV     R0, #50
Delay:  MOV     R1, #40
Delay1: MOV     R2, #249
Delay2: DJNZ    R2, Delay2
        DJNZ    R1, Delay1
        DJNZ    R0, Delay
        RET
```

call a function at label "Delay"

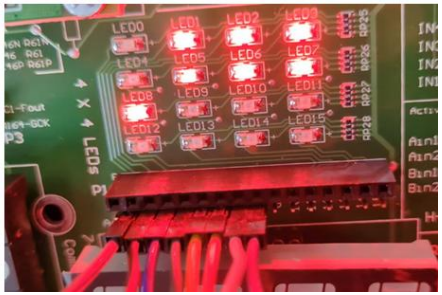
a nested loop to delay some time

Demo Requirements

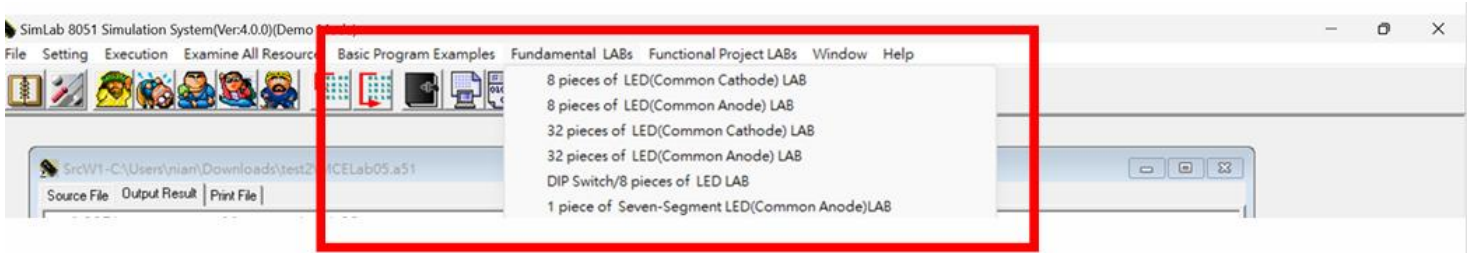
- Assembly only!
- Adjust the nested loop (in the previous slide) to let the LED light shift one step for EACH SECOND
- Bonus 1 (於結報10%加分)
  - 請將學號的數字部分加總後對100取餘數得到XY
  - 改成每 (X+1)/2 秒移動一格
  - Y對5取餘數得到Z
  - 改成每次亮連續的Z+1顆LED燈
- Bonus 2 (於結報 5% or 10%加分)
  - 設計其他LED燈變換的樣態，助教會從兩種程度的加分擇一
  - 給分依據：這個樣態是不是很多人做一樣的?是否能明確解釋source code?

Hints

- 參考助教前次實驗給大家練習的軟體和硬體操作
- 請用杜邦線連接對應的pin腳
  - 板子上對應的腳位 (P0.0 → pin 11; P0.1 → pin 12; P0.2 → 13; …)



- SimLab裡有些參考範例可以協助大家快速上手此次實驗



Lab02 Study Report

- File name: Bxxxxxxx-MCE-Lab2-Study
- File type: PDF only
- The requirements of report
  - Summarize the content of this slide set
  - Provide your plan for this lab exercise
  - No more than one A4 page
  - Grading: 80 ± 15
- Deadline: 2025/10/01 23:00 (不收遲交)
- Upload to e-learning system

Lab02 Lab Exercise Report

- File name: Bxxxxxxx-MCE-Lab2-Result
- File type: PDF only
- The requirements of report
  - Summarize the problems and results you have in this exercise
  - Some screen shots or some code explanation can be provided
  - No more than two A4 pages
  - Grading: 80 ± 15
- Deadline: 2025/10/8 23:00 (不收遲交)
- Upload to e-learning system
- Bonus:
  - Read the Demo Requirements