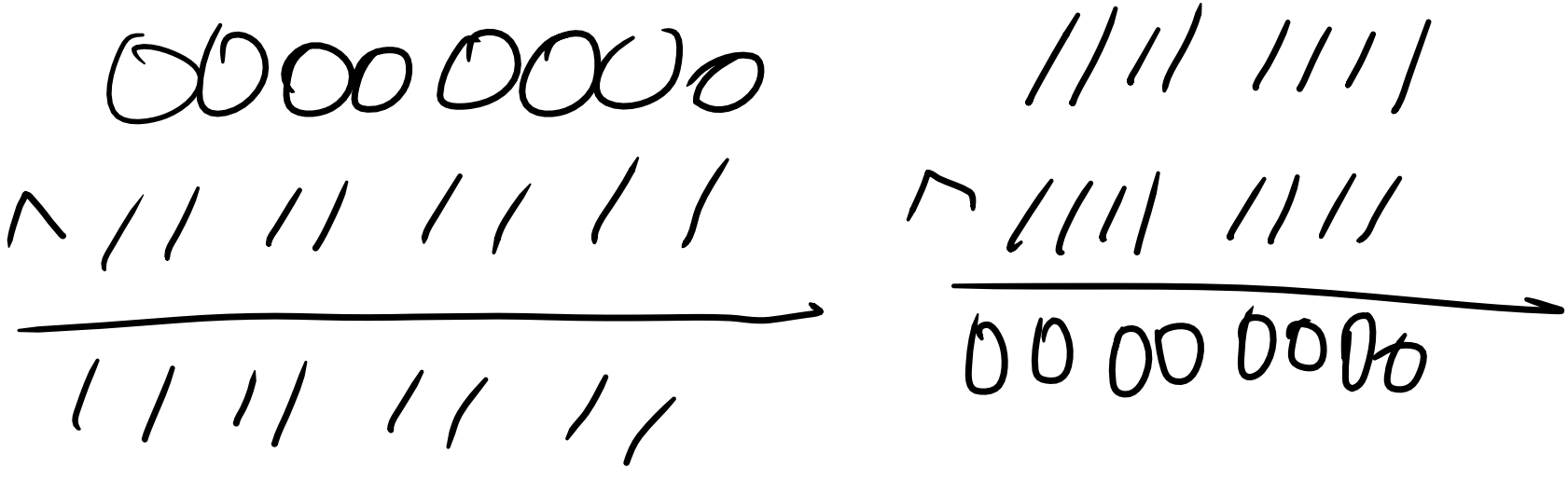
| Button | 동작 |
| --- | --- |
| 1 | Left shift |
| 2 | Right shift |
| 3 | All Led Blink |
| 4 | All Led off |

```
static uint8_t ledBlinkData = 0x00;
ledBlinkData ^= 0xff;
PORTD = ledBlinkData;
```

ledBlinkData = 0x00;

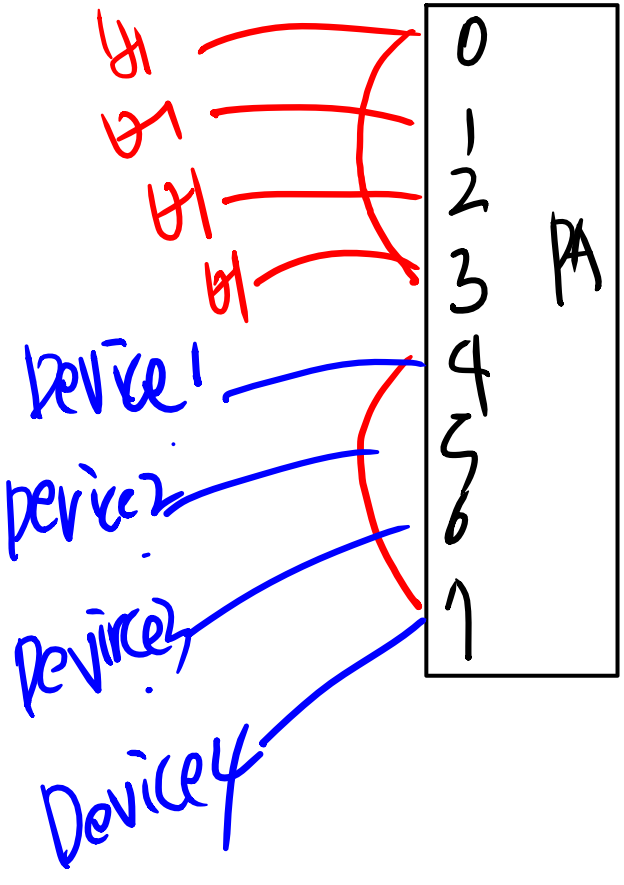
ledBlinkData = ledBlinkData ^ 0xff;



```
BUTTON_DDR &= ~((1<<BUTTON_LEFT) | (1<<BUTTON_RIGHT) | (1<<BUTTON_BLINK) | (1<<BUTTON_OFF));
```

B_D = B_D & ~( ~~~~ )

↑ 2




~(1<<B_L) | (1<<B_R) |
(1<<B_B) | (1<<B_D)

(00000001) | (00000010) |
(00000100) | (00001000);

B_D = B_D & ~(0000 1111)
= B_D & (1111 0000)

PDRA &= ~(1<<2);

∴ PDRA = PDRA & ~(1<<2)

= PDRA & ~(00000100)

= PDRA & (11111011)

특정 bit만 0으로 변경

# 특정 비트만 1로 셋팅 (output)

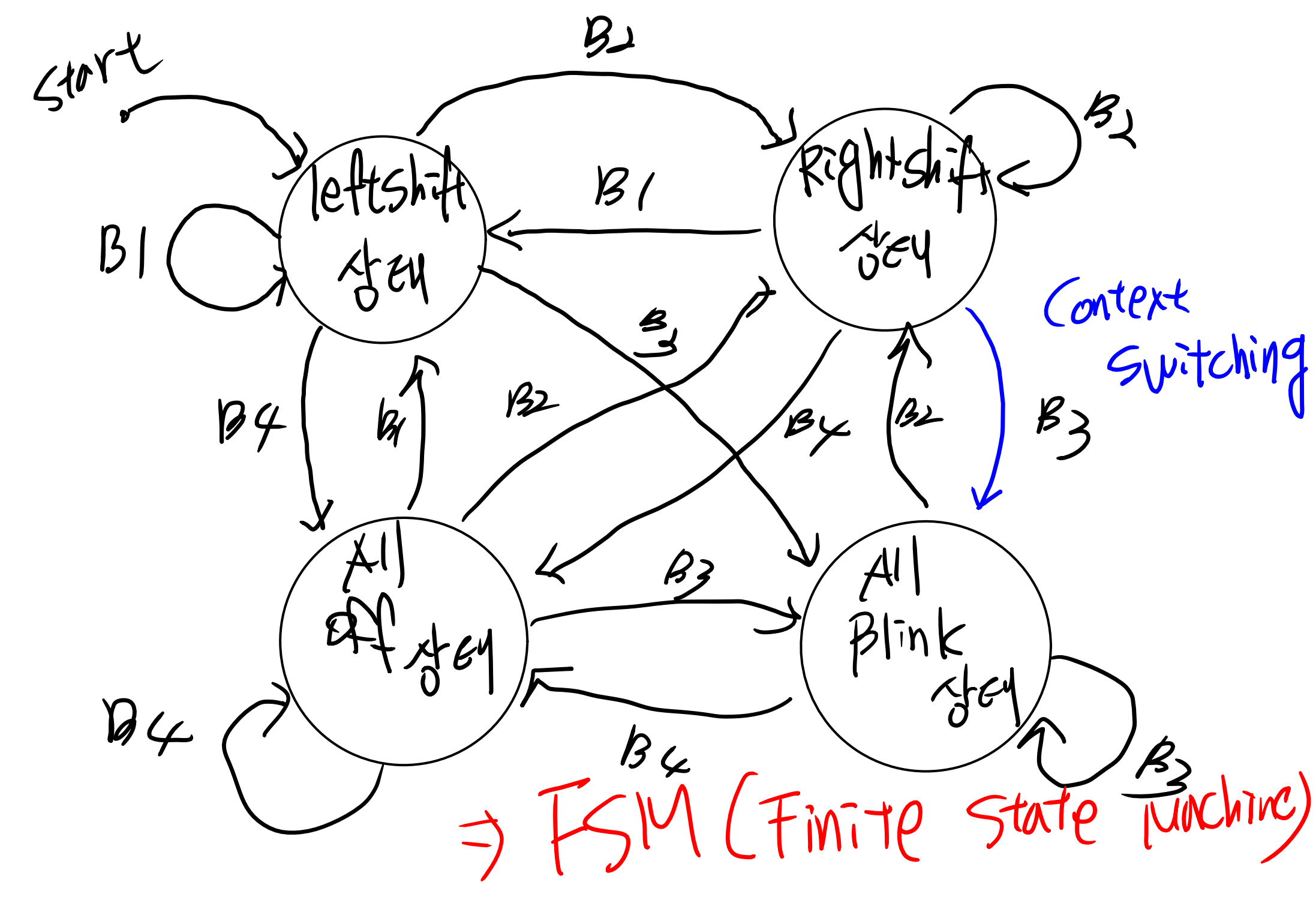
PDRA |= ( 1<<4);

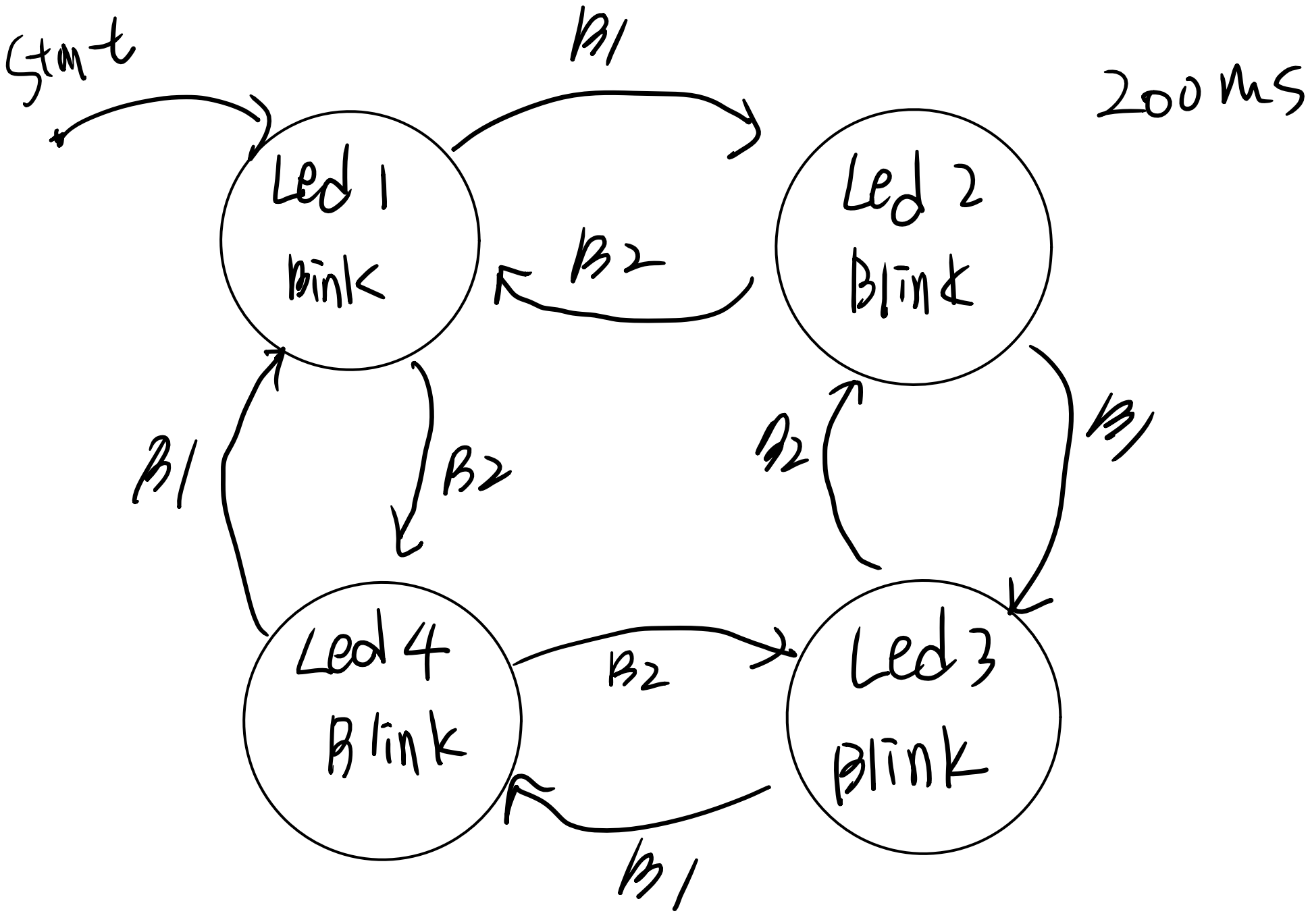
⇒ PDRA = PDRA | ( 1<<4)

= PDRA | ( 00010000 )

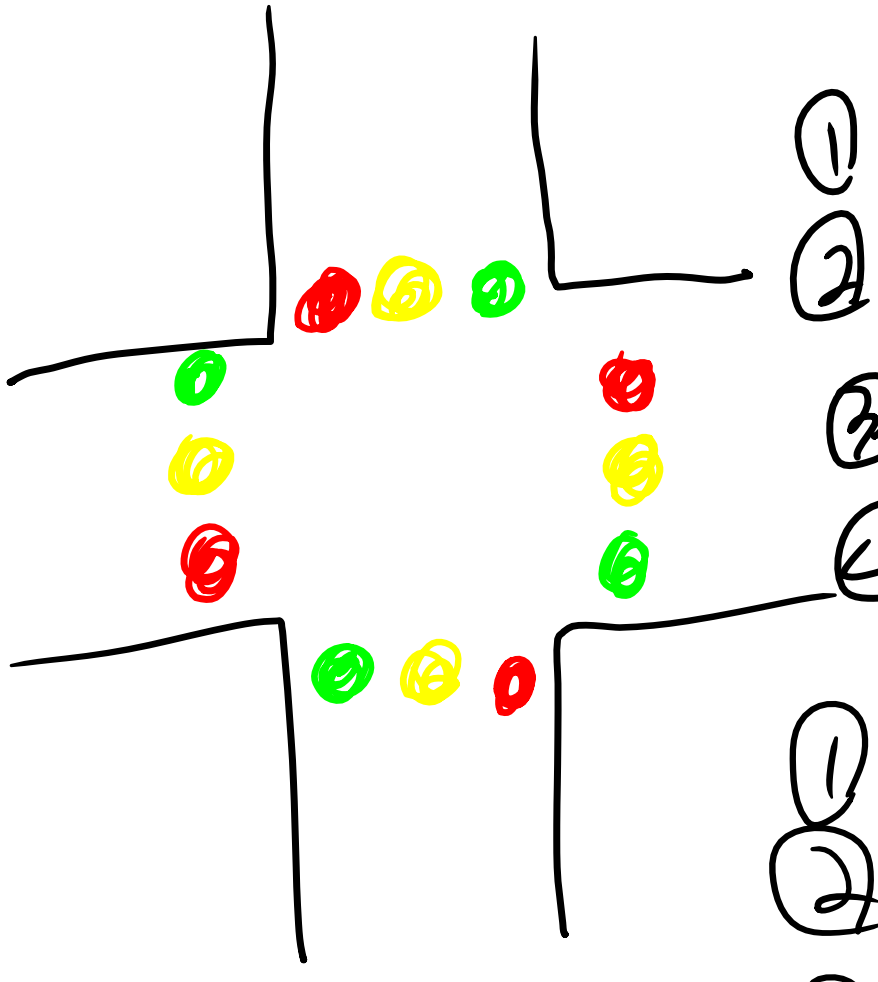
= ( xxxxxxxx ) | ( 00010000 )

= ( xxx | xxxx )


start
B2
leftshift 상태
B1
Rightshift 상태
B2
B1
B3
B4
B1
B2
B4
B2
B3
Context Switching
All Off 상태
All Blink 상태
B3
B4
B4
B3
⇒ FSM (Finite state machine)


Start
B1
200 ms
Led 1
Blink
B2
Led 2
Blink
B1
B2
B2
B1
Led 4
Blink
B2
Led 3
Blink
B1

# simple 신호등 만들기 (xW)



상하

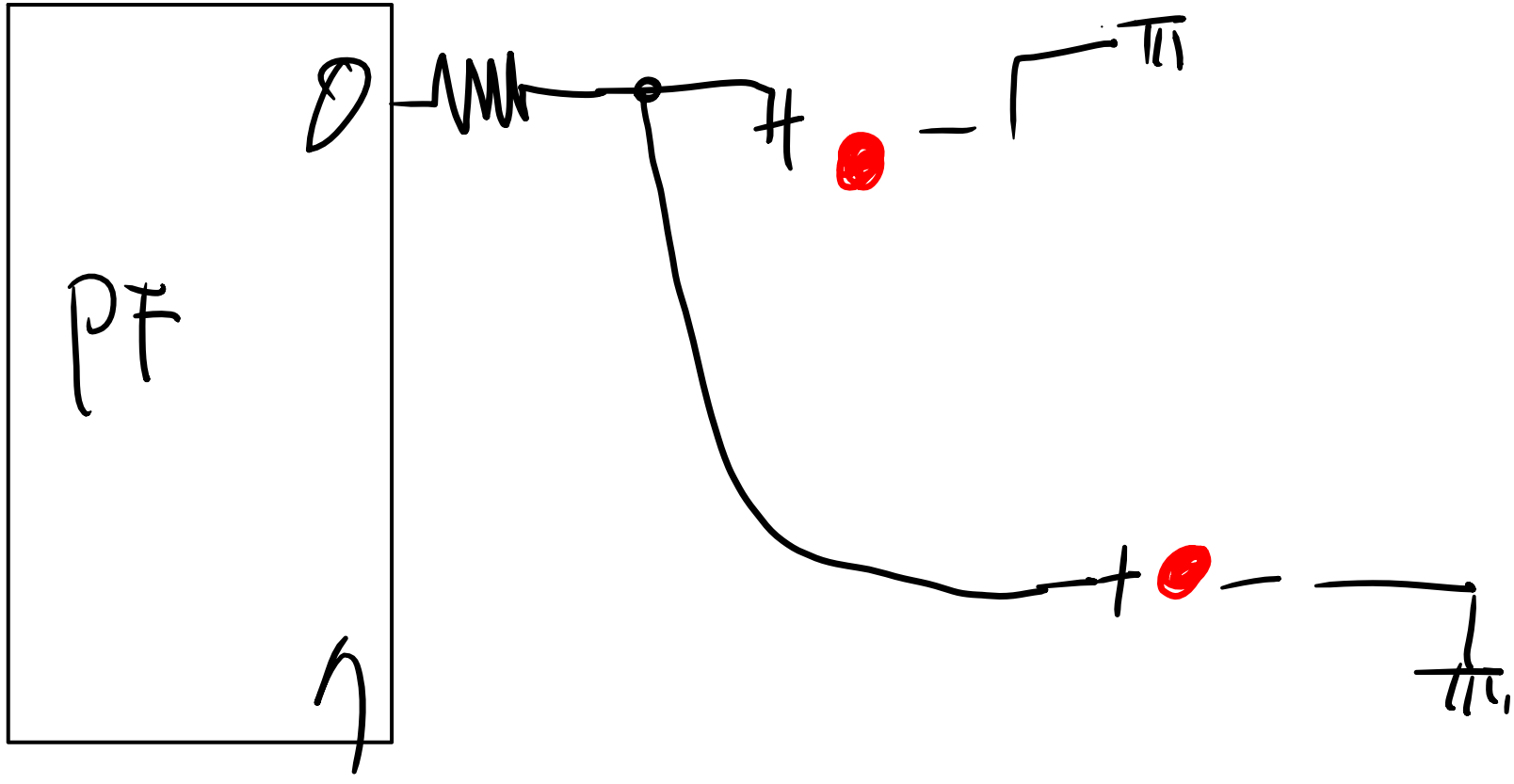
① Red
② Red
③ Green
④ Yellow

과목 (Auto)

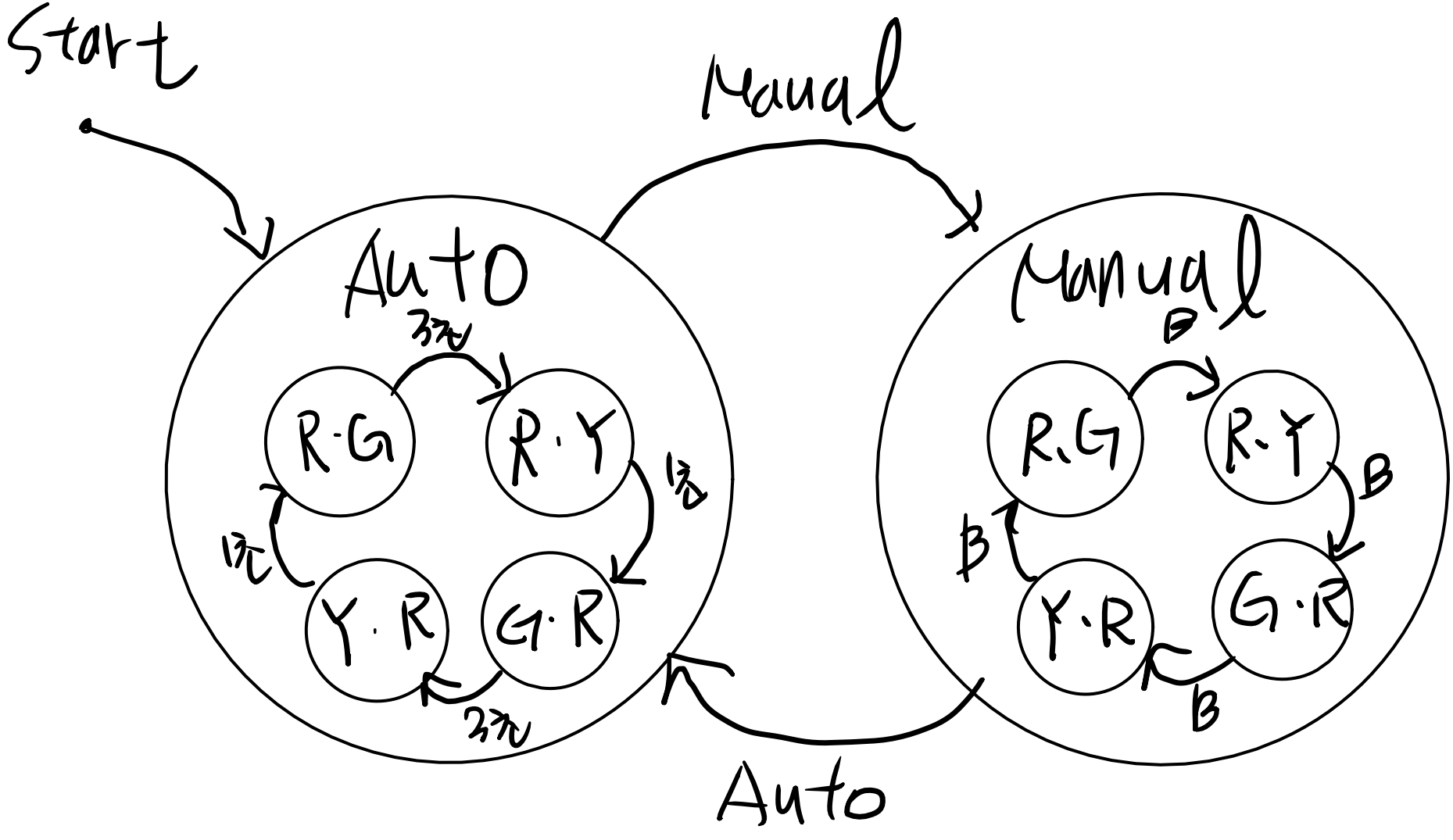
Green → 3초
Yellow → 1초
Red → 3초
Red → 3초

① Button 1
② Button 2
③ Button 3

Auto mode
manual Mode
switching
① → ② → ③ → ④


PF

# Traffic Signal FSM



문제점 1

Auto → Manual Mode 변화시

B2를 꼭 누르거나 연타 처리함

문제점 2

Manual 모드에서 switching

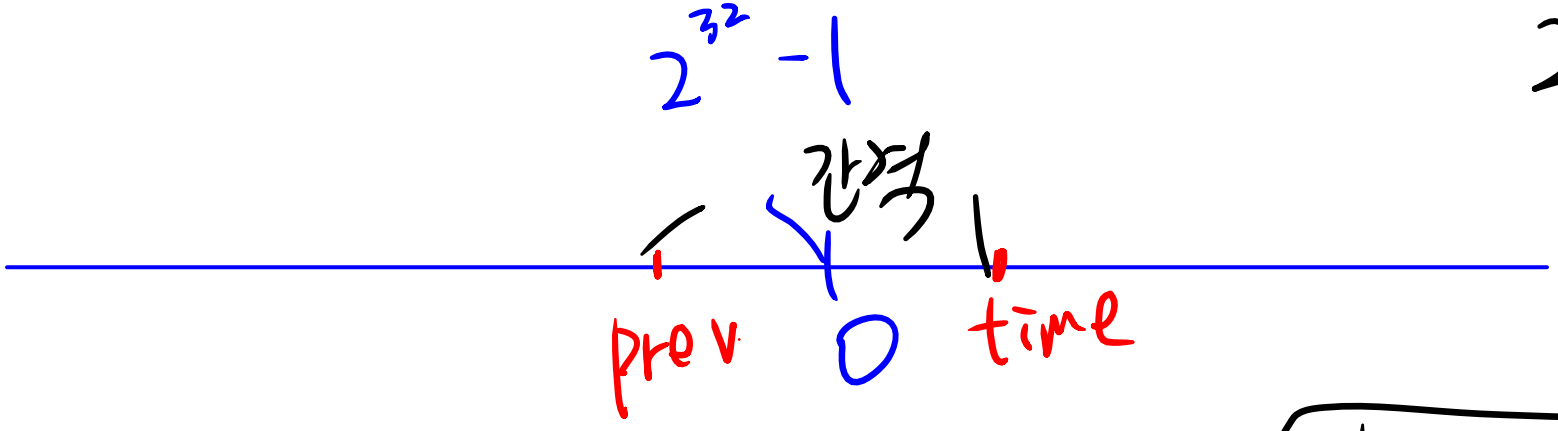
버튼을 눌렀을 때 상태 변화가

다수 일어난다.

- delay() 함수 ) 권장
  sleep() 함수 지양해야함.

- delay 함수가 CPU 점유한 상황
  ⇒ Blocking 현상
  Blocked I/O code

non-Blocking
non-Blocked I/O code




5 - 2 ?

0101

+ 1110

6011

5 + (1-4)

2의 보수

0010

과제 LED Machine 다시 짜보기