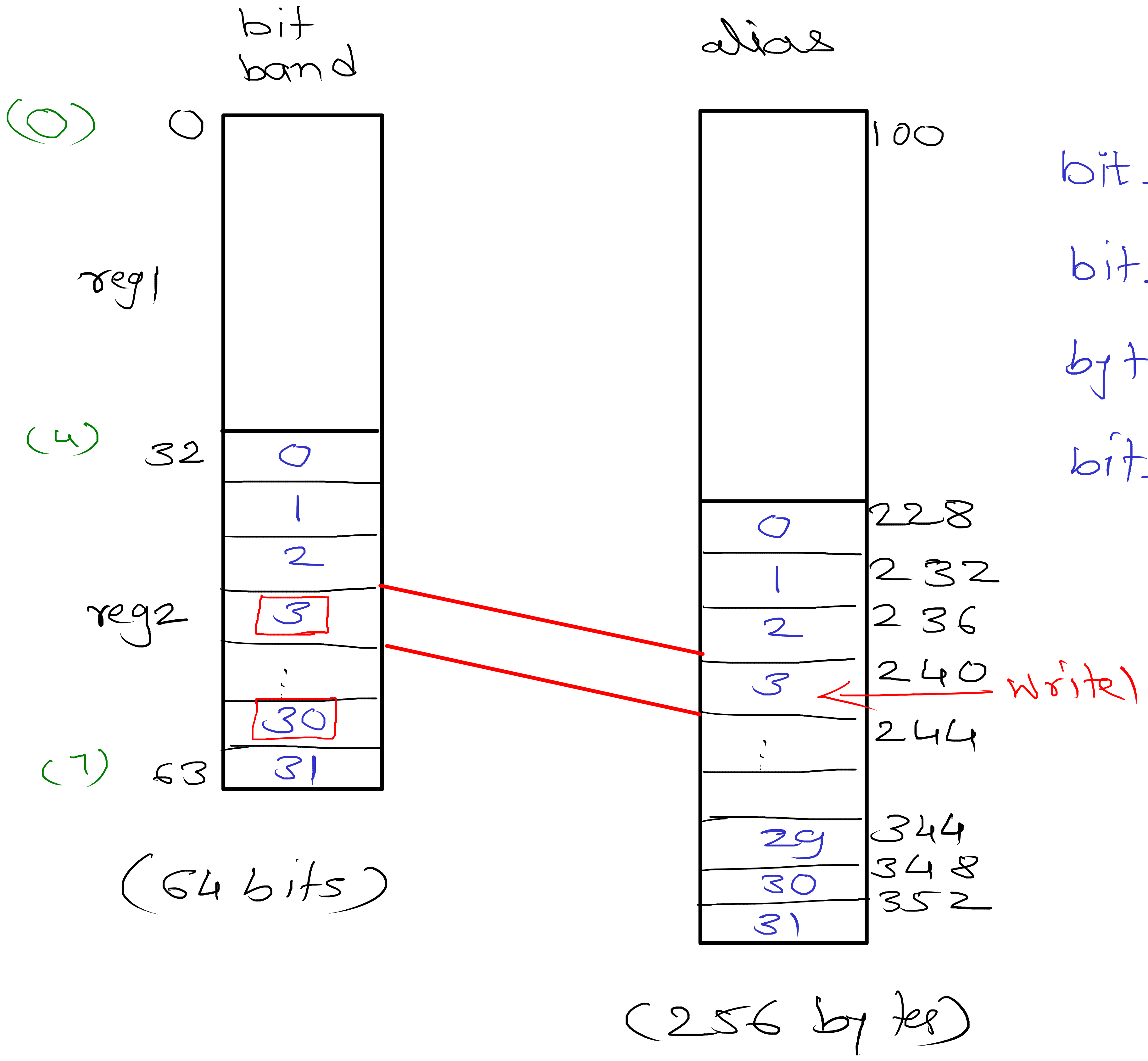
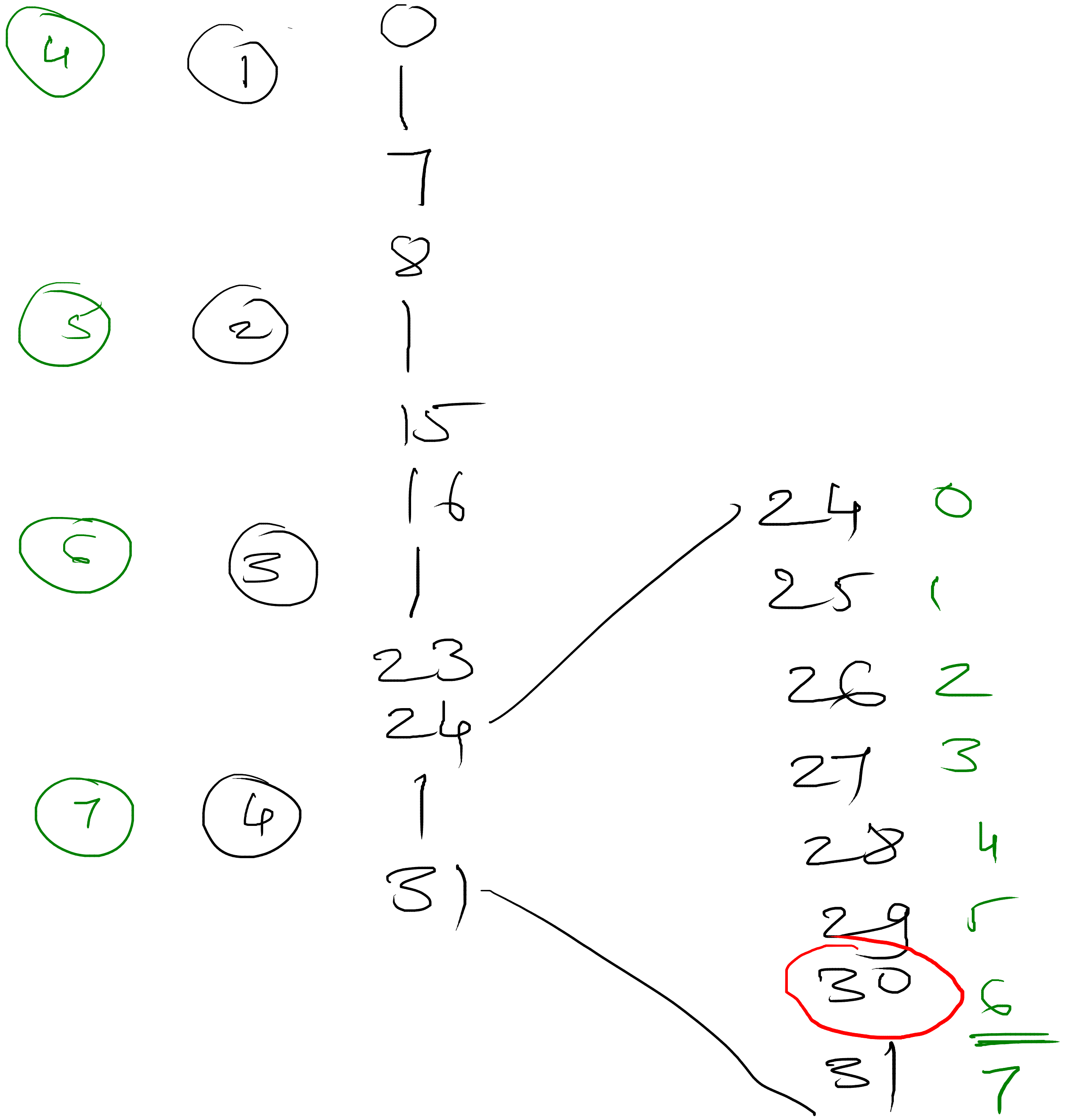


bit_word_addr = 240
=
bit_band_base = 100
+ +
byte_offset * 32 = 4−0 * 32 = 128
+ +
bit_number * 4 = 3*4 = 12

bit_word_addr = 348
=
bit_band_base = 100
+ +
byte_offset * 32 = 7−0*32 = 224
+
bit_number * 4 = 6*4 = 24



bit_word_addr

=

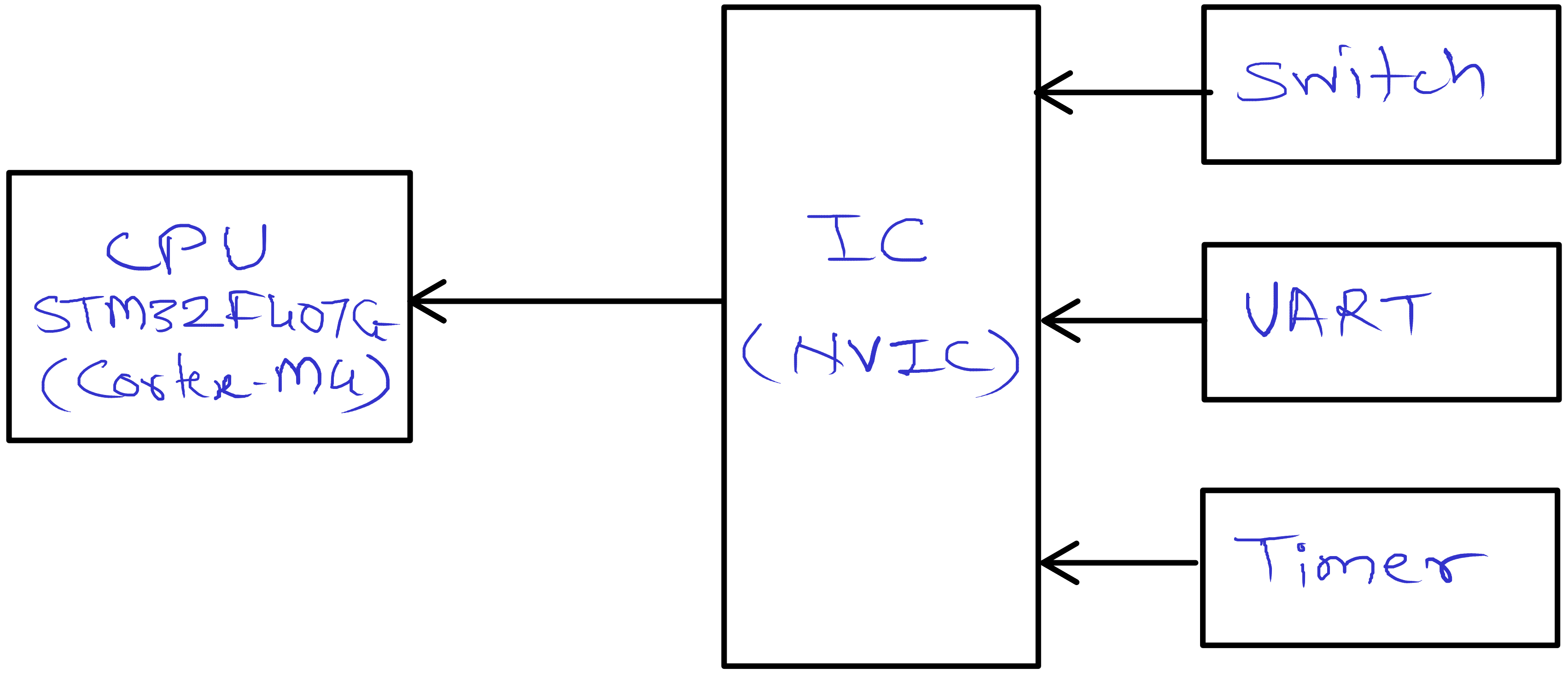
bit_band_base

+

byte_offset * 32

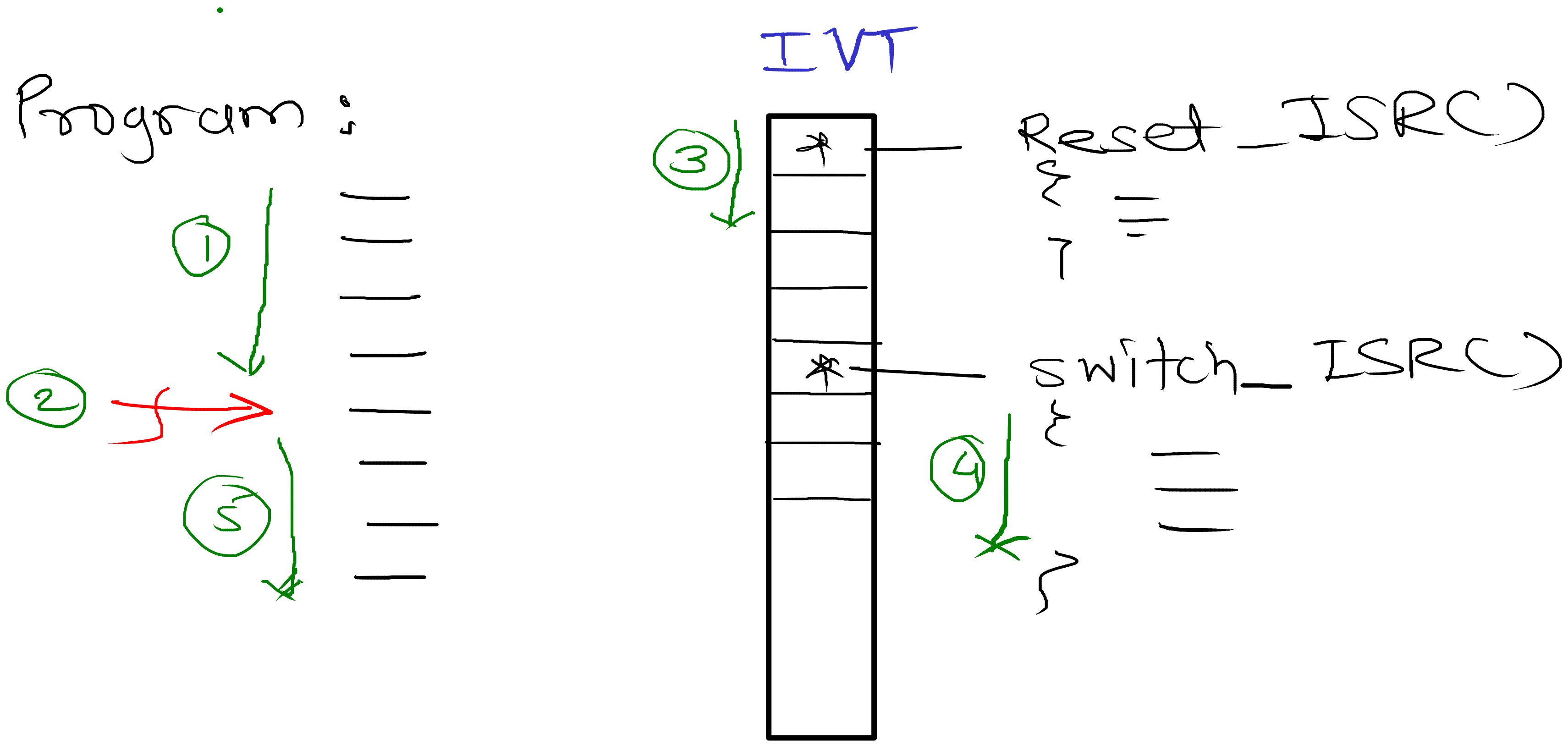
+

bit_number * 4

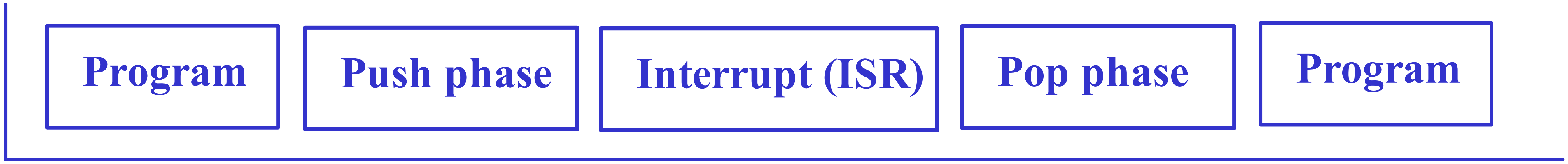


Interrupt Controller

- Convey interrupts to CPU
- decide the priority of interrupt



IRQn - Interrupt Number
Priority - Interrup priority
ISR - Interrupt Service Routione
IVT - Interrupt Vector Table



Program

Push phase

Interrupt (ISR)

Pop phase

Program

Optimization : -O0(None)

```
int flag = 0;
void EXTI0_IRQHandler(void)
{
    EXTI->PR |= BV(0);
    flag = 1;
    /*  MOV r7, = flag
        LDA r6, #1
        STA r6, [r7]
    */
}

int main(void)
{
    while(1)
    {
        while(flag == 0)
            ;
        /*  MOV r7, = flag
            start:
                LDA r6, [r7]
                CMP r6, #0
                beq start
        */}
}
```

Optimization : -O3(Most)

```
int flag = 0;
void EXTI0_IRQHandler(void)
{
    EXTI->PR |= BV(0);
    flag = 1;
    /*  MOV r7, = flag
        LDA r6, #1
        STA r6, [r7]
    */
}

int main(void)
{
    while(1)
    {
        while(flag == 0)
            ;
        /*  MOV r7, = flag
            LDA r6, [r7]
            start:
                CMP r6, #0
                beq start
        */}
}
```

Optimization : -O3(Most)

```
volatile int flag = 0;
void EXTI0_IRQHandler(void)
{
    EXTI->PR |= BV(0);
    flag = 1;
    /*  MOV r7, = flag
        LDA r6, #1
        STA r6, [r7]
    */
}

int main(void)
{
    while(1)
    {
        while(flag == 0)
            ;
        /*  MOV r7, = flag
            start:
                LDA r6, [r7]
                CMP r6, #0
                beq start
        */}
}
```