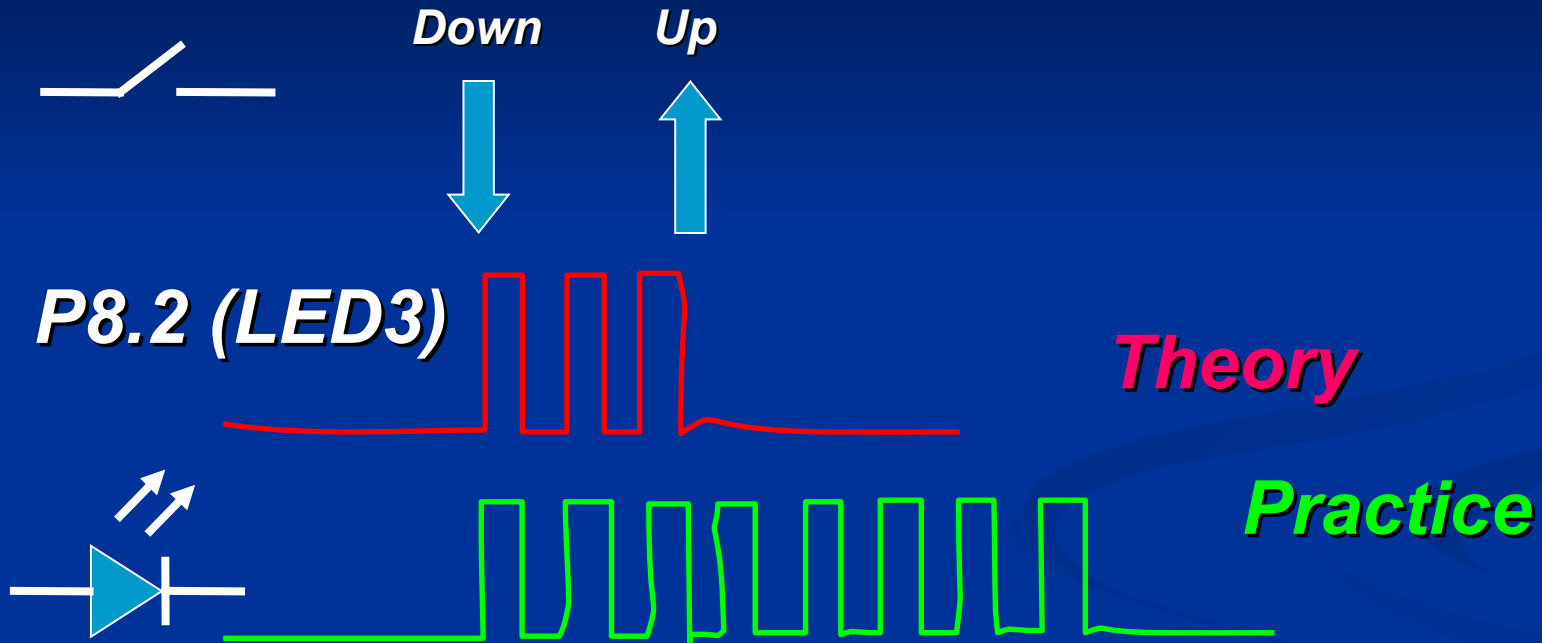


Управление выводами (портами) MSP430F5529

```
#include <msp430.h>
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    volatile unsigned int i;
    P1DIR &= !(BIT7);          // P1.7 (S1) set as input
    P8DIR |= BIT2;              // P8.2 (LED3) set as output
    while(1)
    {
        if (P1IN & BIT7)       // hi level - don't press button
        { P8OUT &= !(BIT2); } // off led
        else P8OUT |= BIT2; // low level - press button - on led
        for(i=25000;i>0;i--);
        P8OUT &= !(BIT2); // off led after small pause
        for(i=25000;i>0;i--); // small pause
    }
    return 0;
}
```

Управление выводами (портами) MSP430F5529



Управление выводами (портами) MSP430F5529

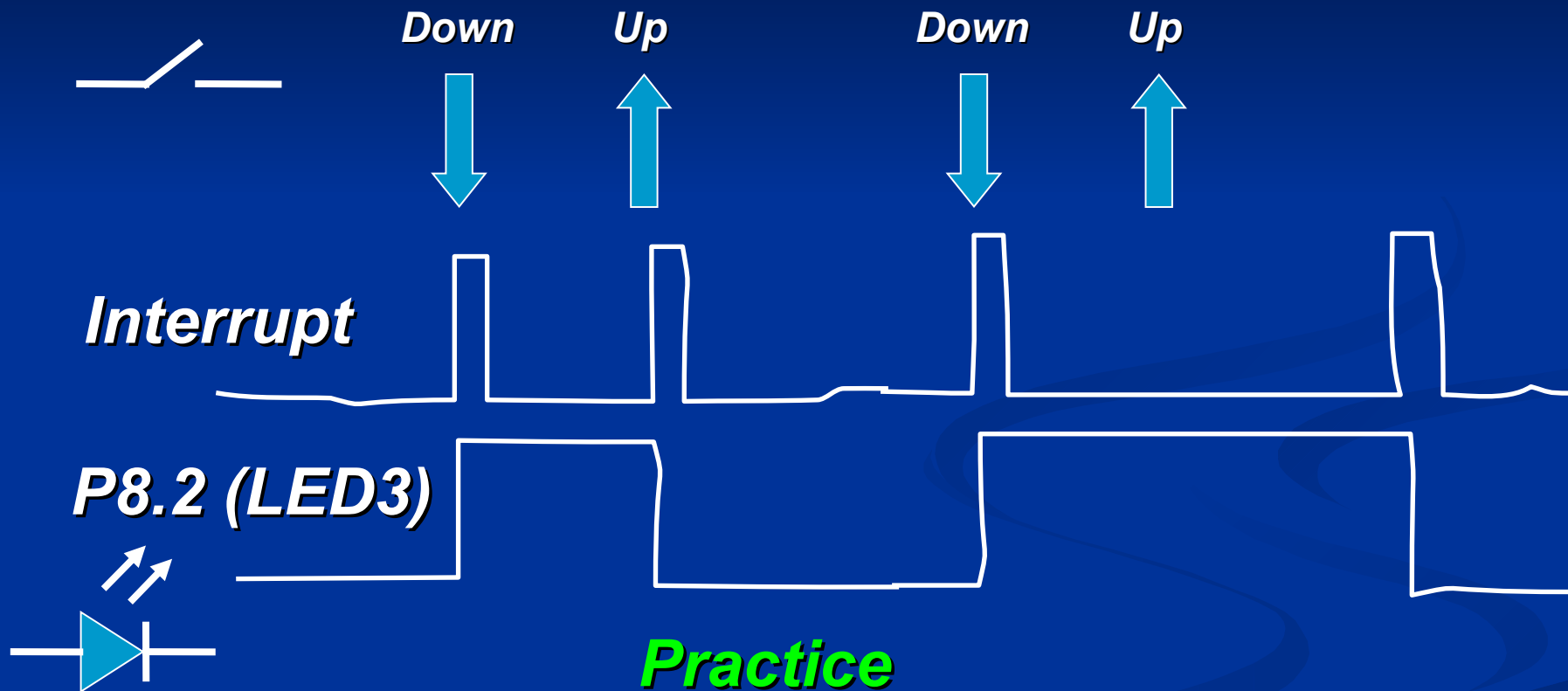
```
#include <msp430.h>
int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    P1DIR &= !(BIT7); // P1.7 (S1) set as input
    P8DIR |= BIT2; // P8.2 (LED3) set as output

    P1OUT |= BIT7; // ????????
    P1IE |= BIT7; // P1.7 (S1) interrupt enable
    P1IES |= BIT7; // P1.7 (S1) edge for interrupt:
                  // high-to-low
    // Enter LPM0, enable interrupts
    __bis_SR_register(LPM0_bits + GIE);
    __no_operation(); // For debugger
    return 0;
}
```

Управление выводами (портами) MSP430F5529

```
#pragma vector=PORT1_VECTOR
__interrupt void PORT1_ISR(void) {
    if (P1IN & BIT7)          // high level - don't press button
        {P8OUT &= !(BIT2);} // off led
    else P8OUT |= BIT2; // low level- press button- on led
    if (!(P1IES & BIT7)) // ?????
        P1REN ^= BIT7; // ?????
    P1IES ^= BIT7; // ?????
    P1IFG = 0; // reset interrupt flag
}
```

Управление выводами (портами) MSP430F5529



Логическая ошибка!

```
#include <msp430.h>
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    volatile unsigned int i;
    P1DIR &= !(BIT7);          // P1.7 (S1) set as input
    P8DIR |= BIT2;              // P8.2 (LED3) set as output
    while(1)
    {
        if (P1IN & BIT7)      // hi level - don't press button
        { P8OUT &= !(BIT2); } // off led
        else P8OUT |= BIT2;    // low level - press button - on led
        for(i=25000;i>0;i--);
        P8OUT &= !(BIT2);      // off led after small pause
        for(i=25000;i>0;i--);  // small pause
    }
    return 0;
}
```

Логическая ошибка!

```
#include <msp430.h>
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    volatile unsigned int i;
    P1DIR &= ~(BIT7);          // P1.7 (S1) set as input
    P8DIR |= BIT2;              // P8.2 (LED3) set as output
    while(1)
    {
        if (P1IN & BIT7)       // hi level - don't press button
        { P8OUT &= ~(BIT2); } // off led
        else P8OUT |= BIT2;    // low level - press button - on led
        for(i=25000;i>0;i--);
        P8OUT &= ~(BIT2);      // off led after small pause
        for(i=25000;i>0;i--);  // small pause
    }
    return 0;
}
```

/* Ответ на вопрос 5 */

Логическая ошибка!

P1DIR &= ! (BIT7) ;

BIT7 = 1000 0000

BIT7 = TRUE

! (BIT7) = FALSE

! (BIT7) = 0000 0000 ~ (BIT7) = 0111 1111

P1DIR = XXXX XXXX

& 0000 0000

0000 0000

= XXXX XXXX

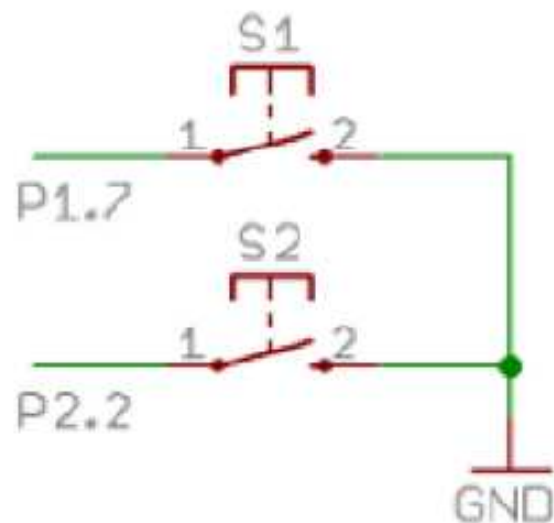
& 0111 1111

0XXX XXXX

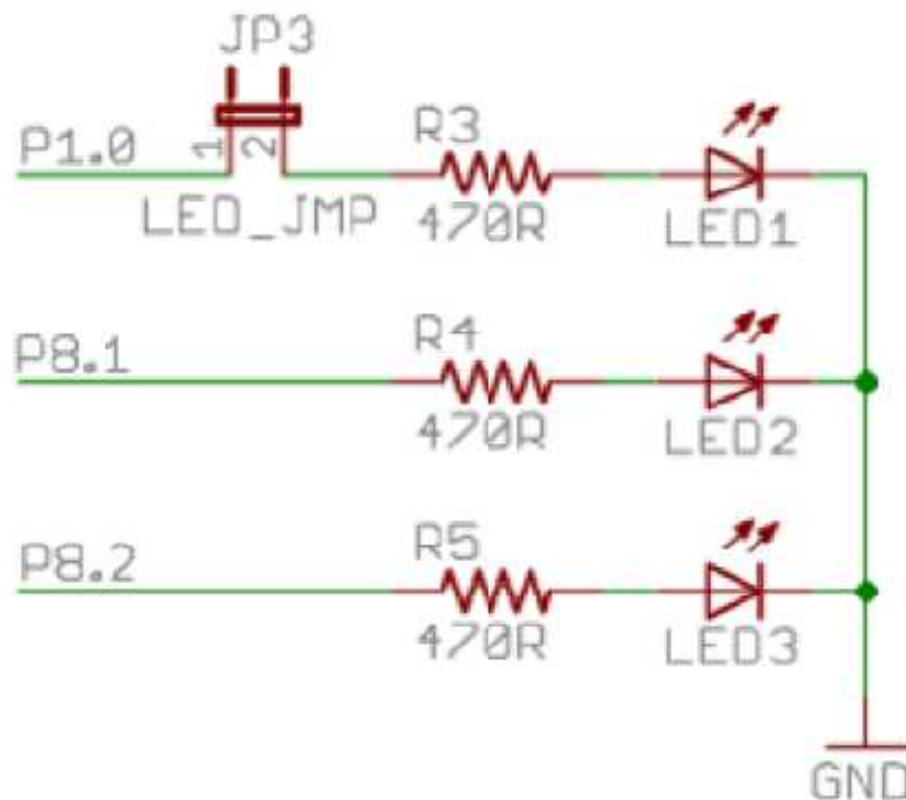
■ Нет, не влияет!

Управление выводами (портами) MSP430F5529

User Buttons

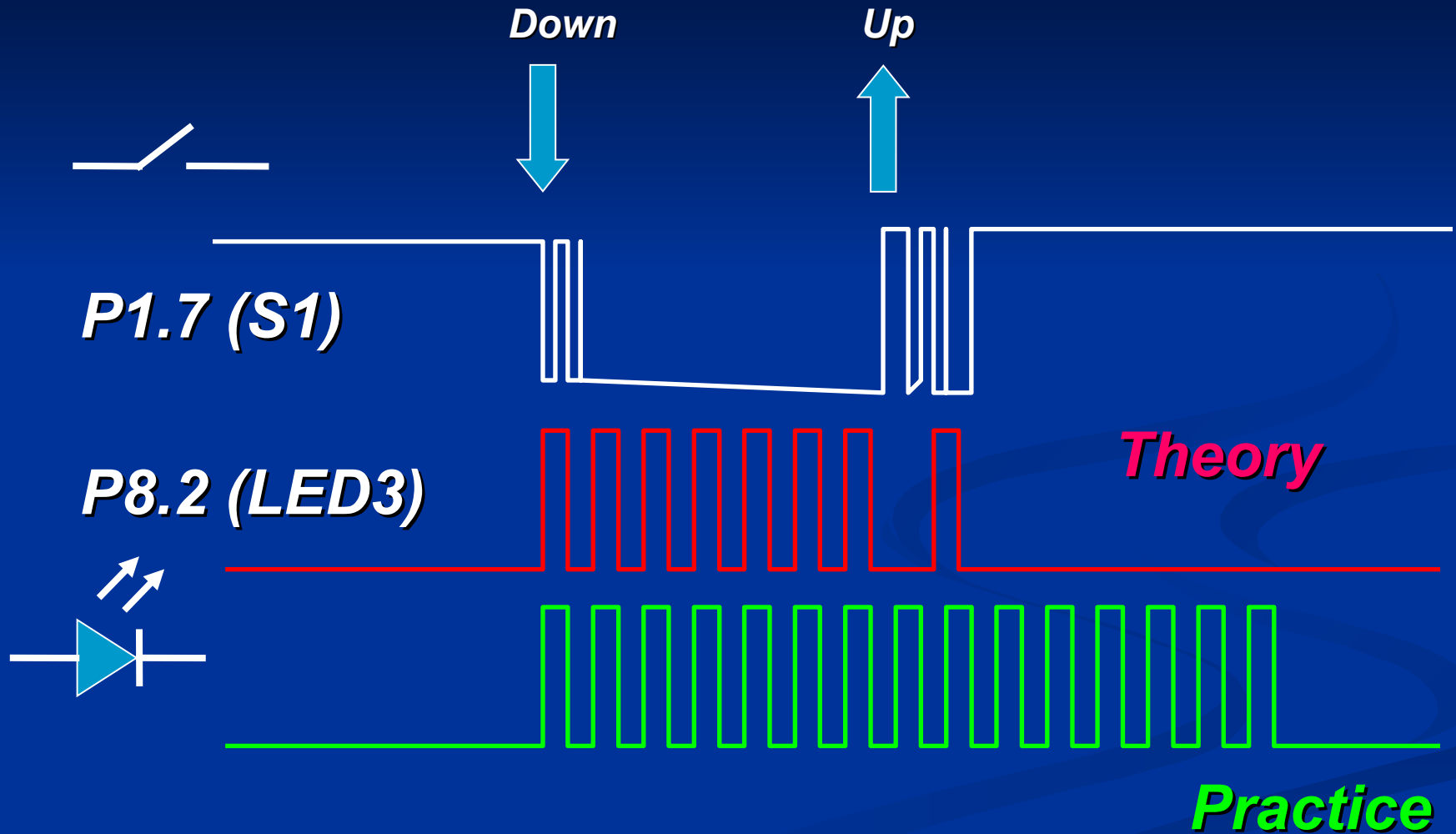


User LEDs



Что происходит?

■ Дребезг контактов?

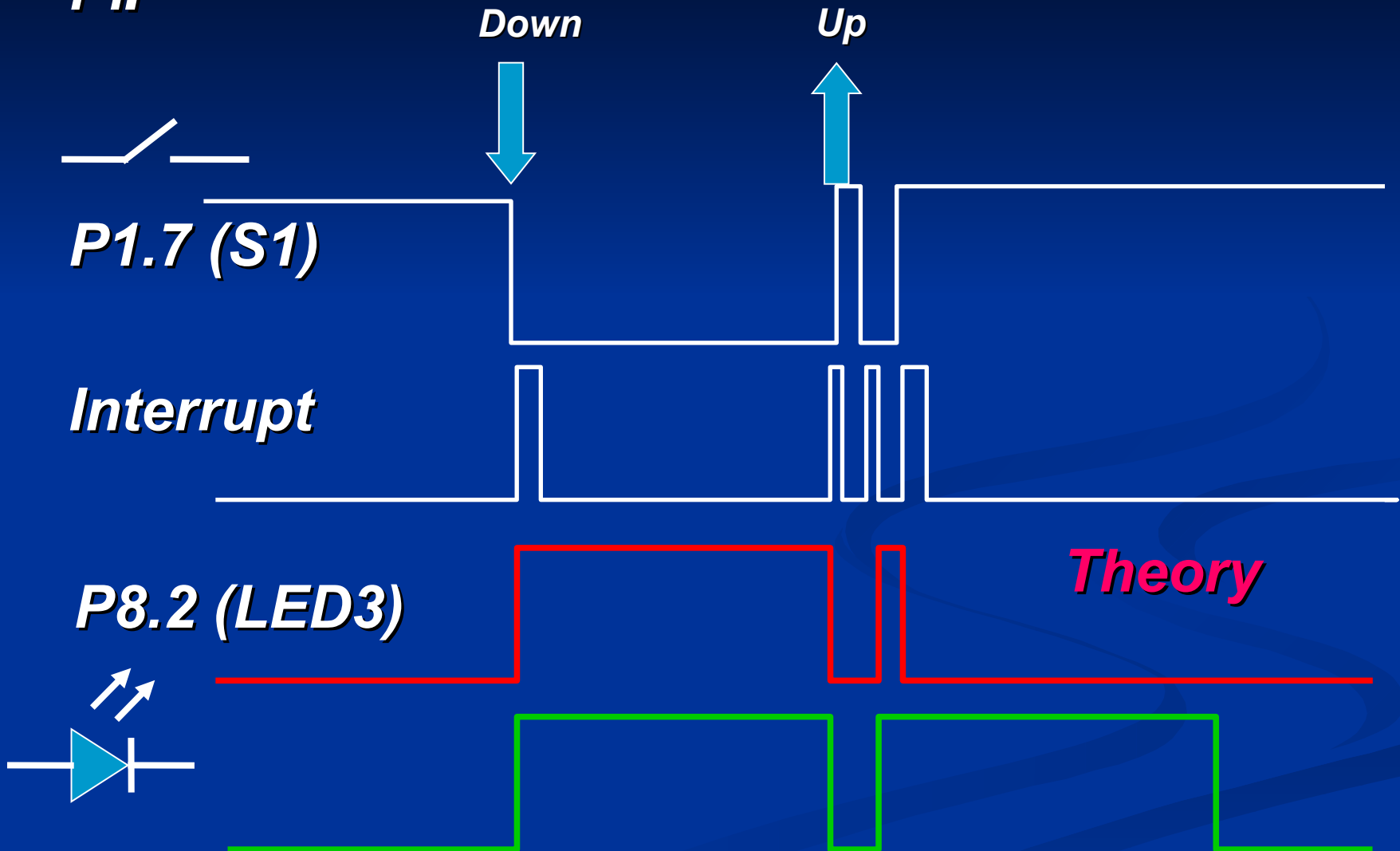


Что происходит?

Budeo
2. Digital IO 2.
Bugs

Что происходит?

■ Дребезг контактов?

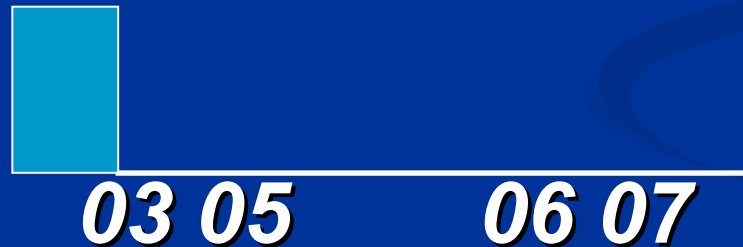


Проблема на стороне диода?

```
01 while(1)
02     {if (P1IN & BIT7)      // hi level - don't press button
03         { P8OUT &= ~(BIT2); } // off led
04     else P8OUT |= BIT2; // low level - press button - on led
05     for(i=25000;i>0;i--);
06     P8OUT &= ~(BIT2); // off led after small pause
07     for(i=25000;i>0;i--); // small pause
08 }
```

P8.2 OUT

02. TRUE



02. FALSE

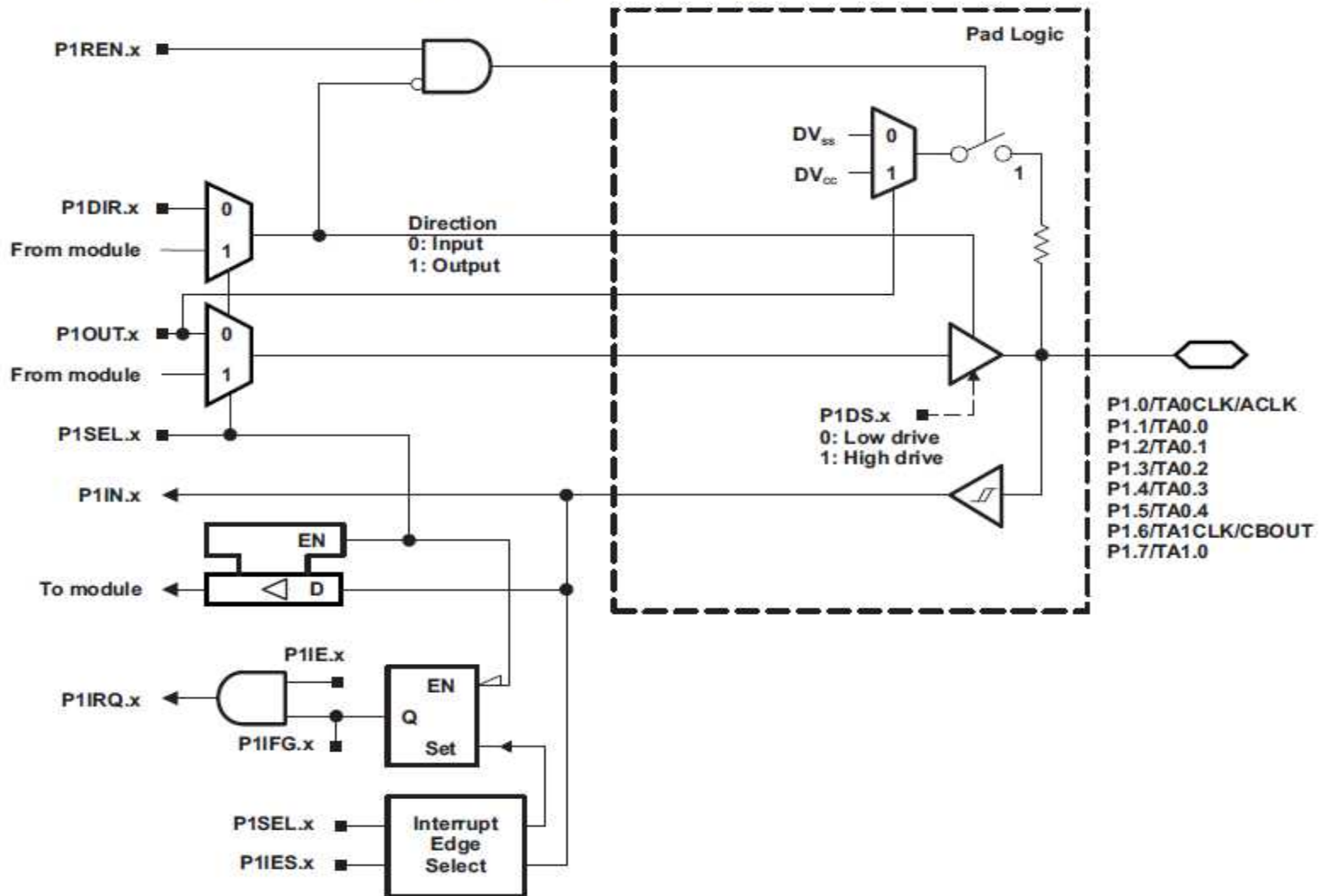


Нет! Проблема на стороне кнопки

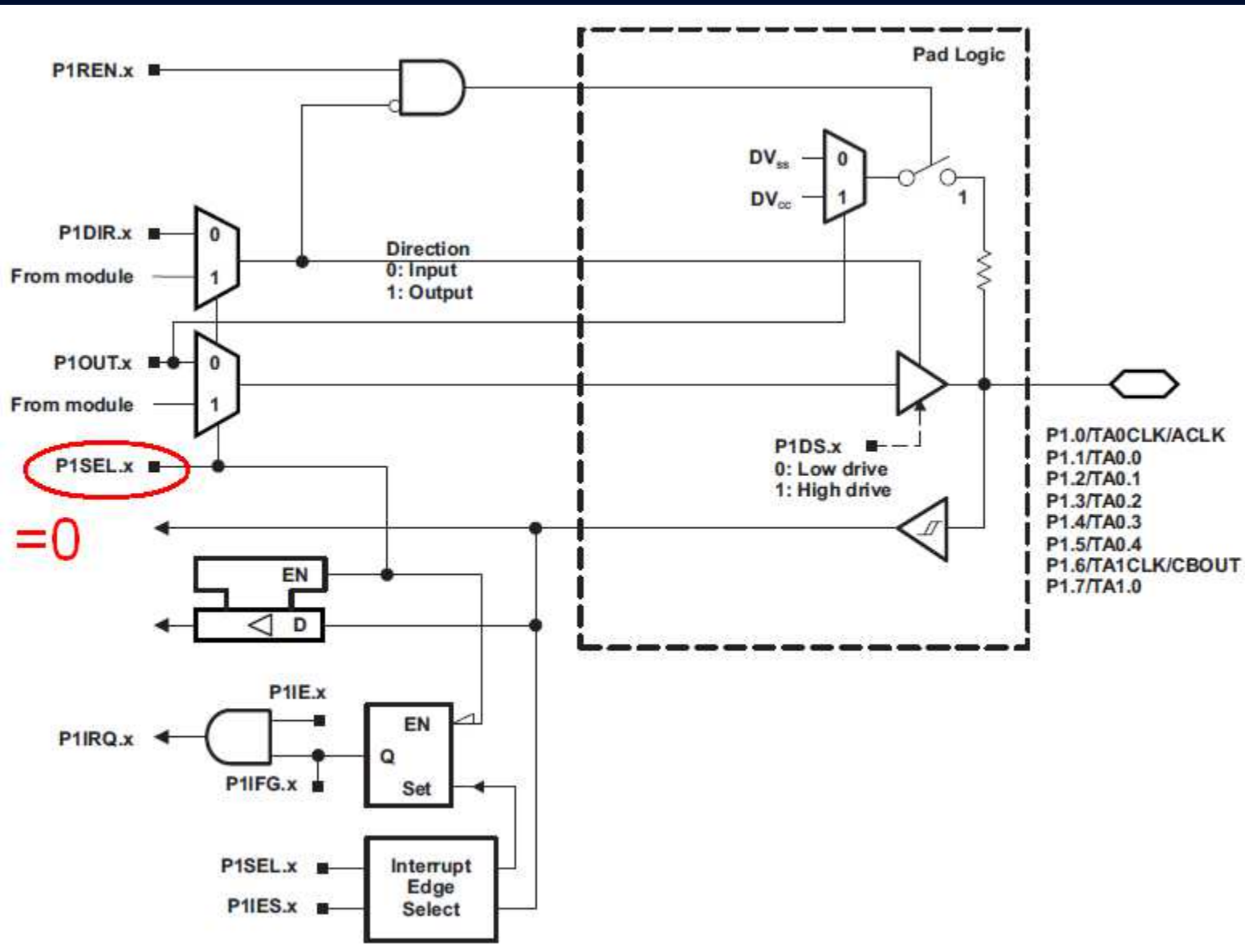
■ MSP430F552x. Mixed Signal Microcontroller

INPUT/OUTPUT SCHEMATICS

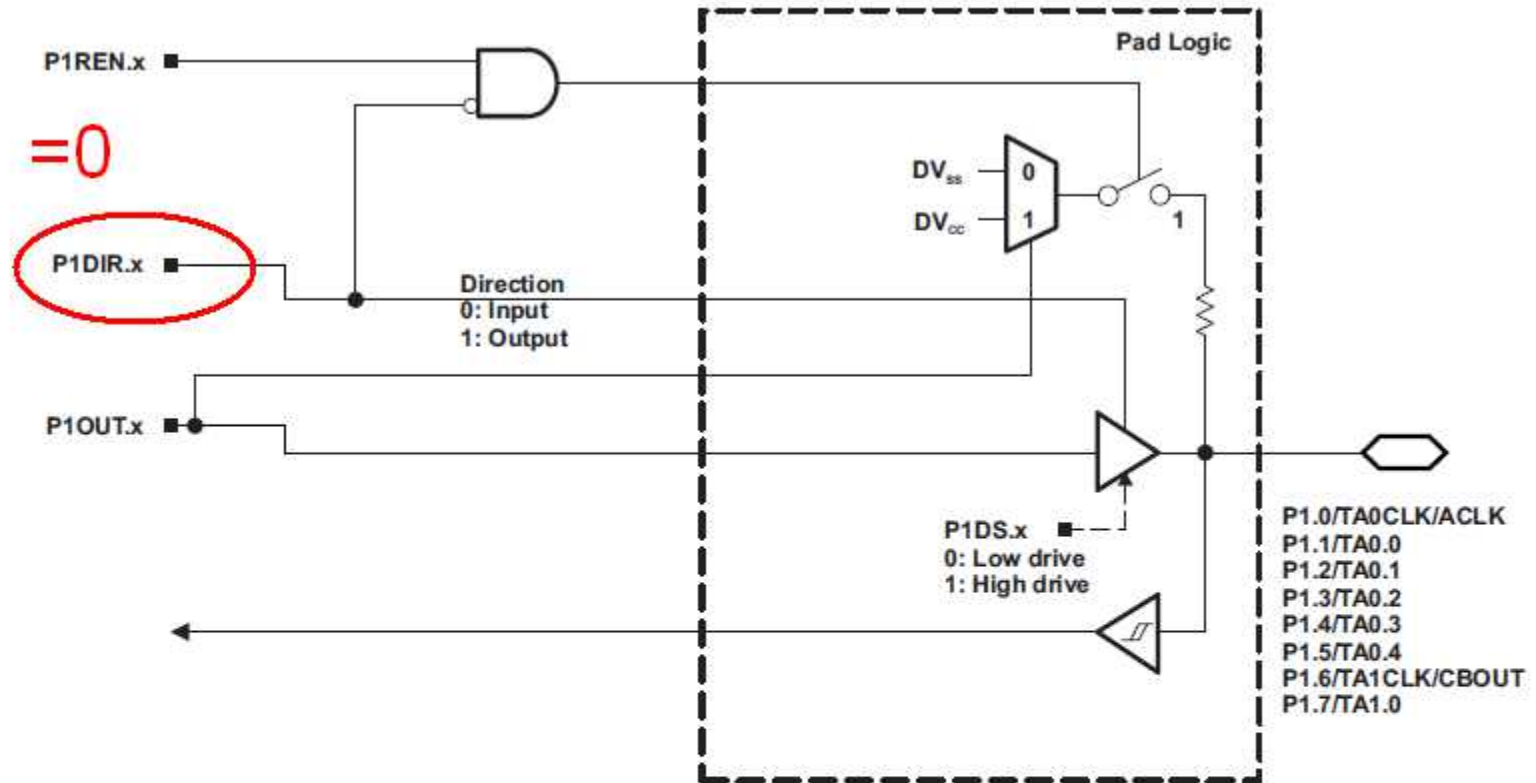
Port P1, P1.0 to P1.7, Input/Output With Schmitt Trigger



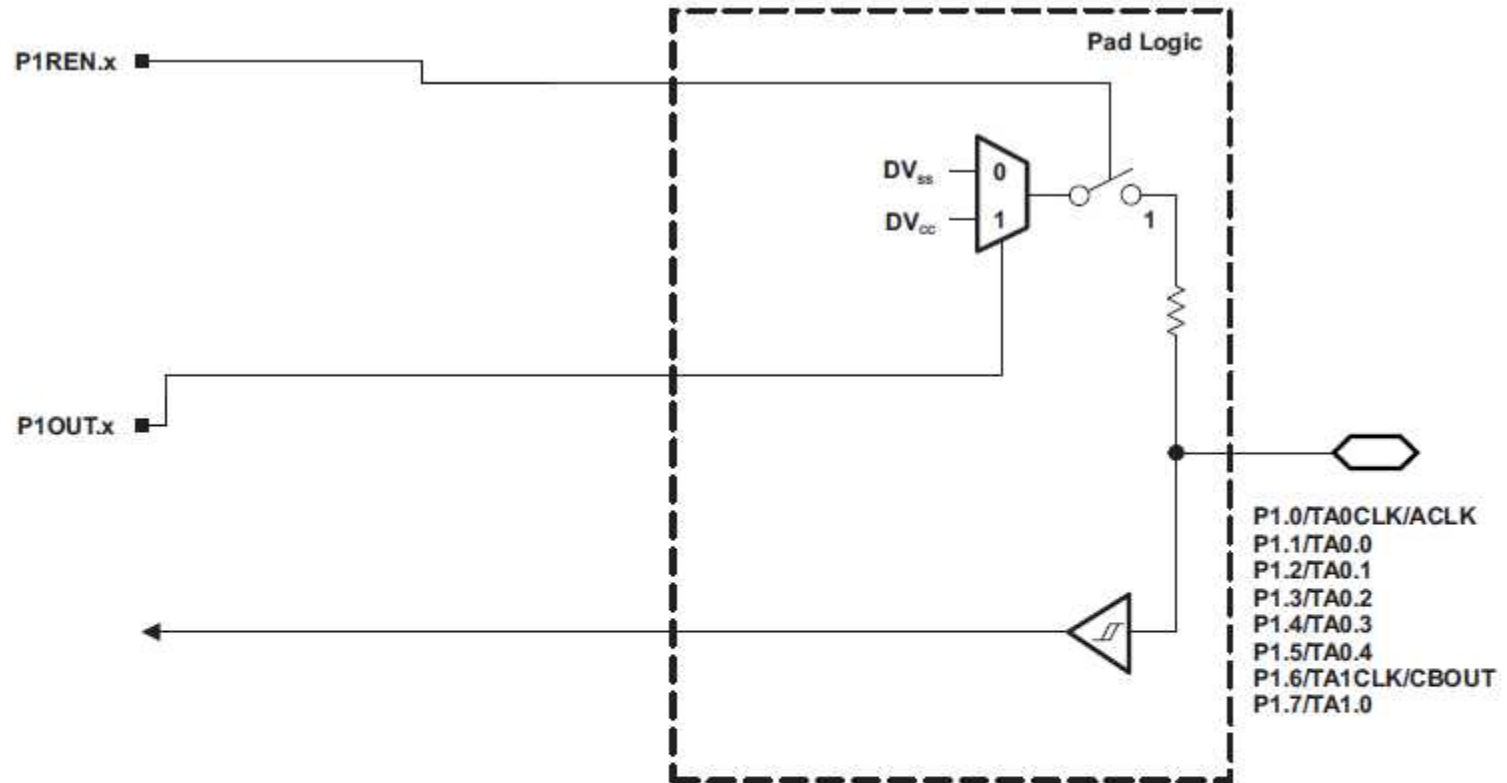
Порт ввода-вывода



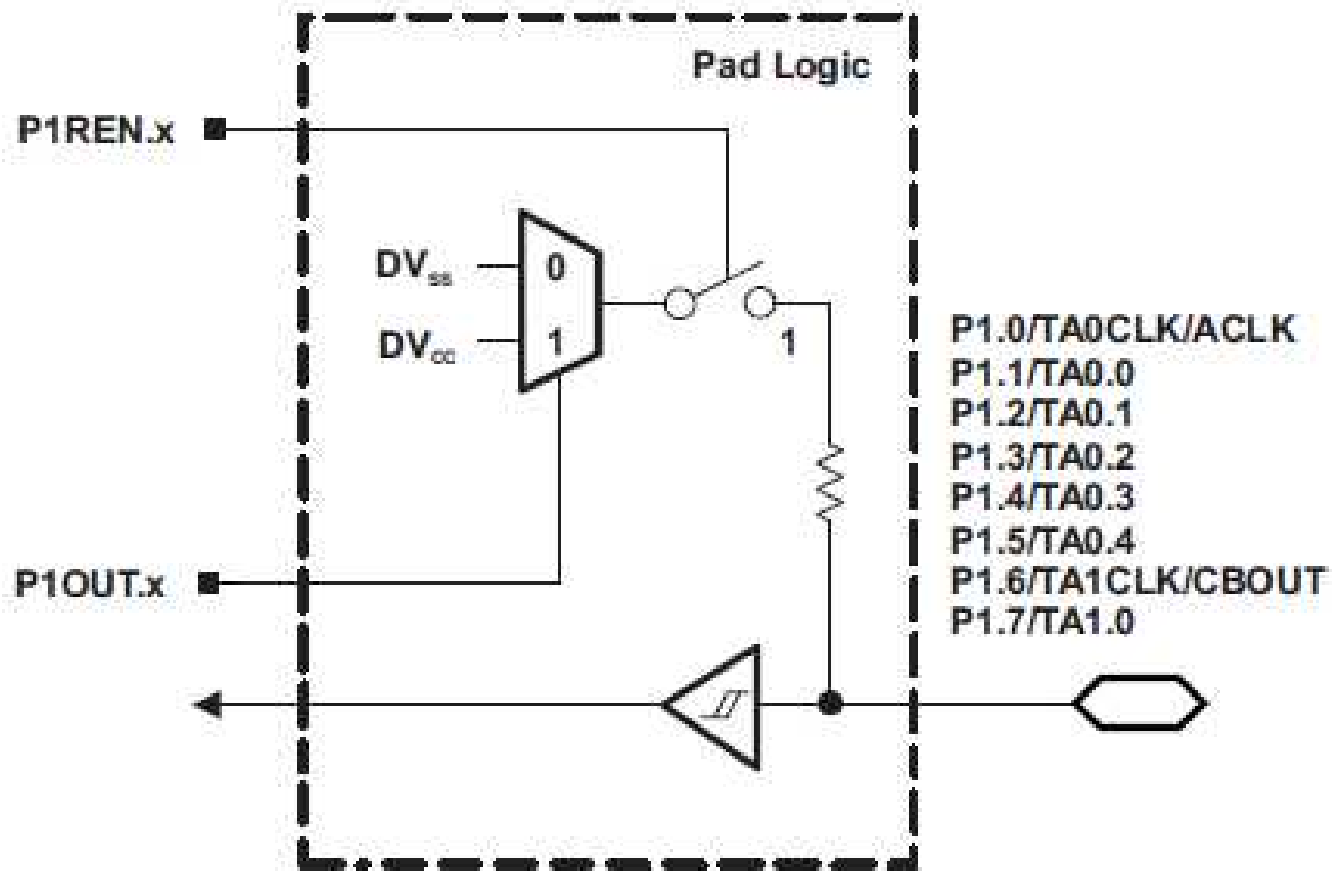
Порт ввода-вывода



Порт ввода-вывода

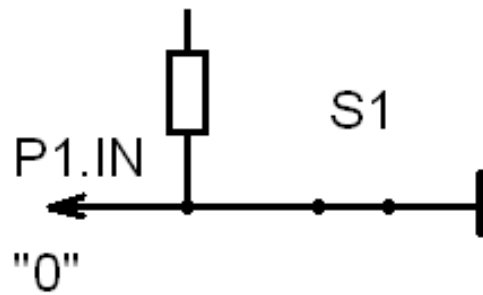
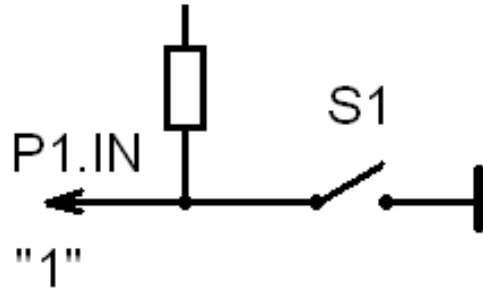


Порт ввода-вывода



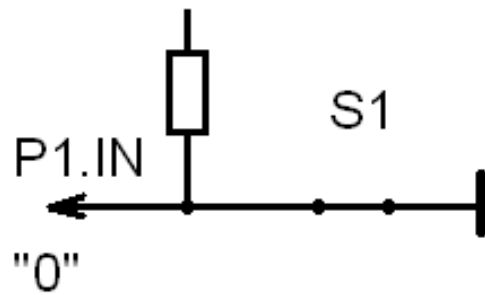
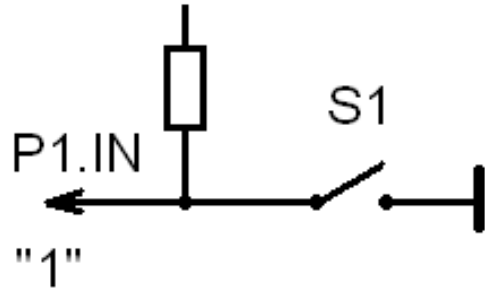
Порт ввода-вывода

P1REN = "0"



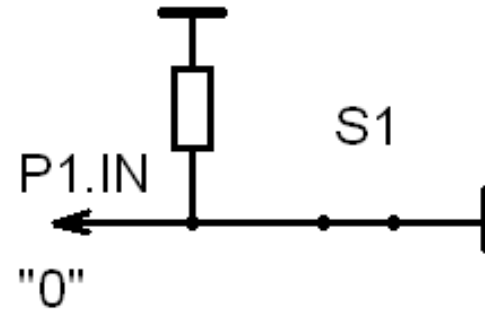
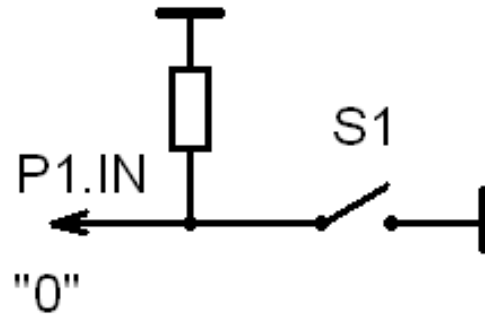
Порт ввода-вывода

P1REN = "0"



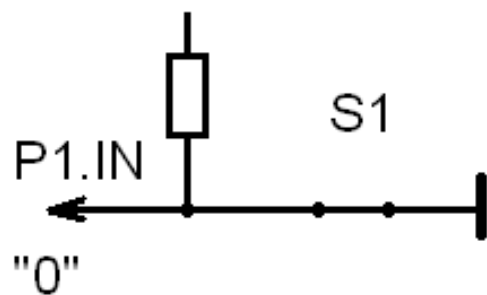
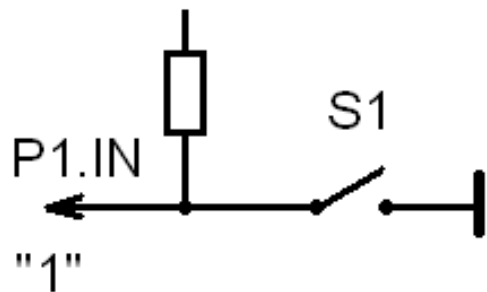
P1REN = "1"

P1OUT = "0"



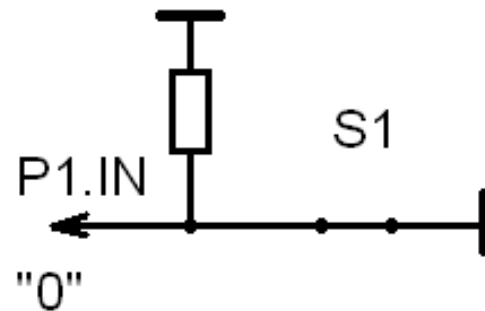
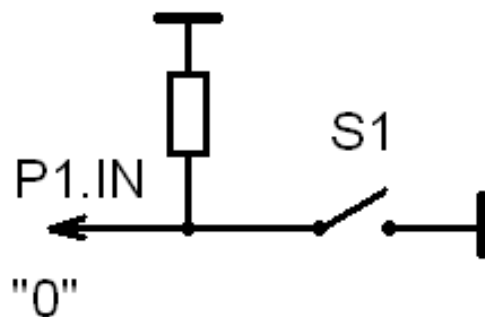
Порт ввода-вывода

P1REN = "0"



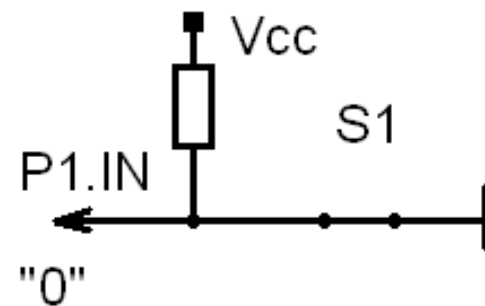
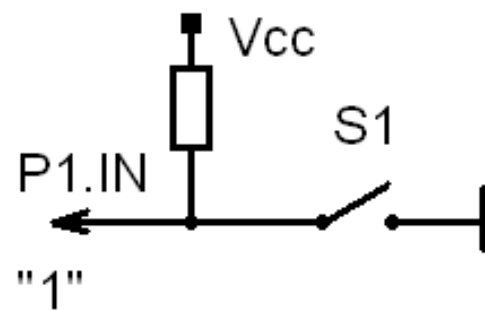
P1REN = "1"

P1OUT = "0"



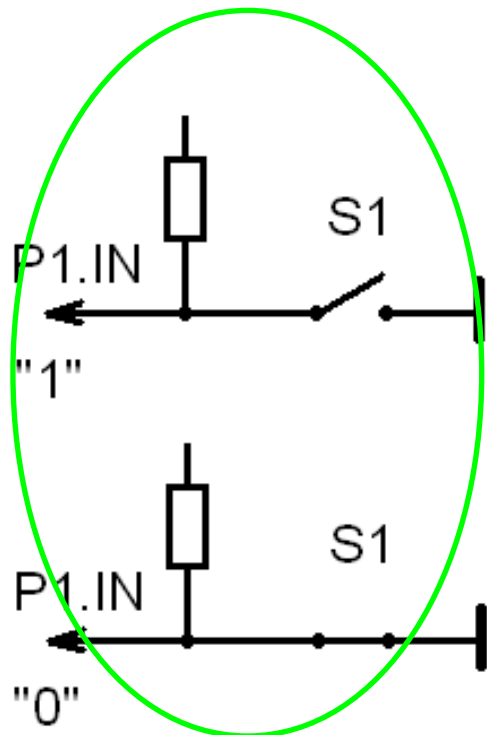
P1REN = "1"

P1OUT = "1"



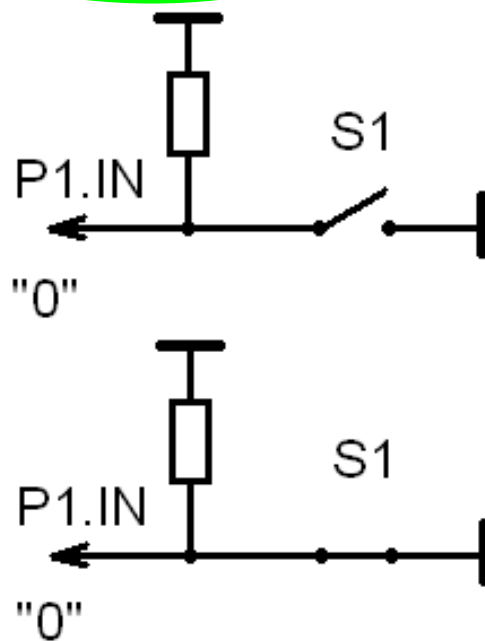
Порт ввода-вывода

P1REN = "0"



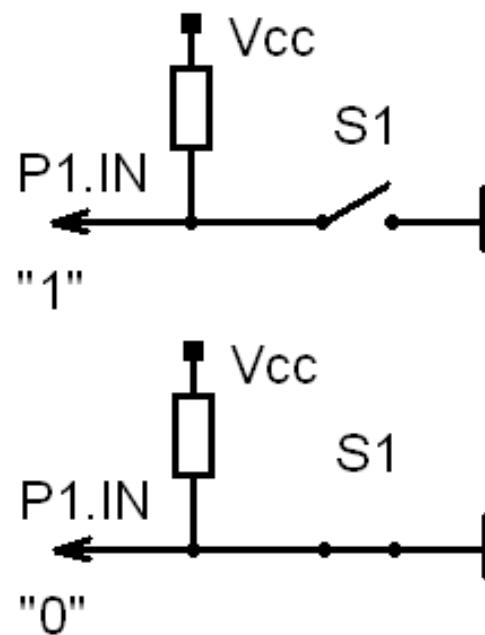
P1REN = "1"

P1OUT = "0"



P1REN = "1"

P1OUT = "1"



■ **Состояние после сброса**

Управление выводами (портами) MSP430F5529

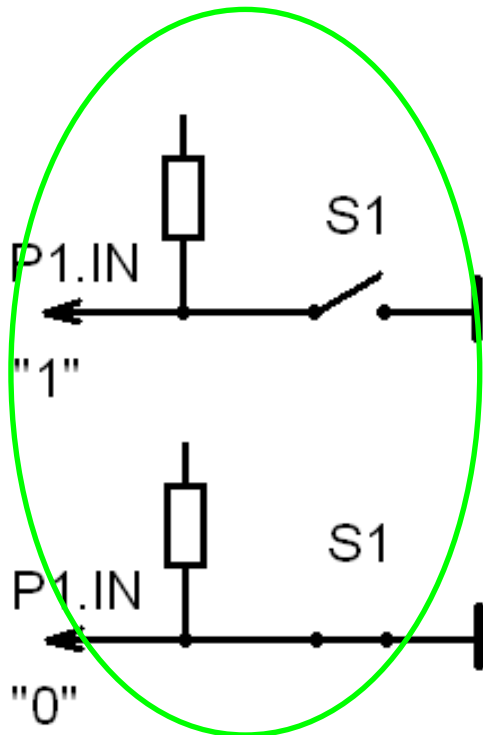
```
#include <msp430.h>
int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    P1DIR &= !(BIT7); // P1.7 (S1) set as input
    P8DIR |= BIT2; // P8.2 (LED3) set as output

    P1OUT |= BIT7; // P1.7 Pull-Up resistor mode
    P1IE |= BIT7; // P1.7 (S1) interrupt enable
    P1IES |= BIT7; // P1.7 (S1) edge for interrupt:
                  // high-to-low
    // Enter LPM0, enable interrupts
    __bis_SR_register(LPM0_bits + GIE);
    __no_operation(); // For debugger
    return 0;
}
```

Ответ на 2 вопрос: использование подтягивающего резистора

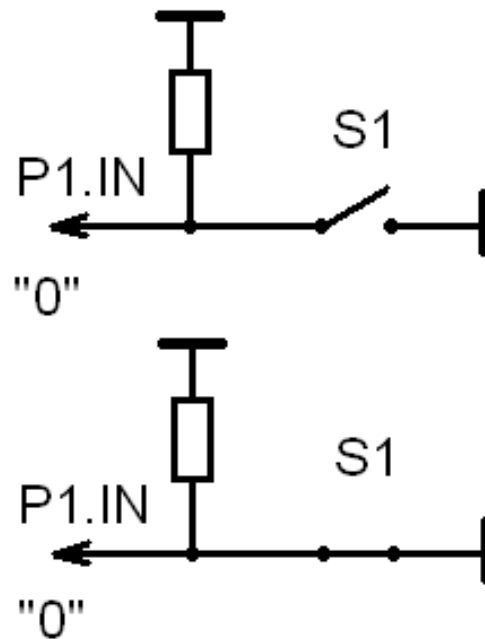
Порт ввода-вывода

P1REN = "0"



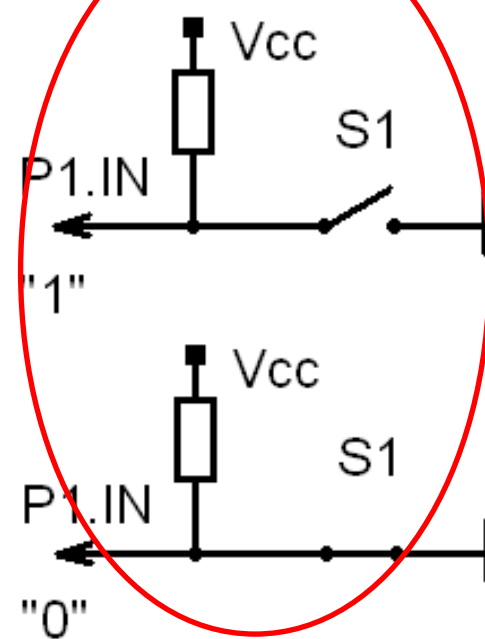
P1REN = "1"

P1OUT = "0"



P1REN = "1"

P1OUT = "1"



■ Решение задачи 3

Управление выводами (портами) MSP430F5529

```
#include <msp430.h>
int main(void) {
    WDTCTL = WDTPW | WDTCTL; // Stop watchdog timer
    P1DIR &= ~(BIT7); // P1.7 (S1) set as input
    P8DIR |= BIT2; // P8.2 (LED3) set as output

    P1OUT |= BIT7; // P1.7 Pull-Up resistor mode
    P1IE |= BIT7; // P1.7 (S1) interrupt enable
    P1IES |= BIT7; // P1.7 (S1) edge for interrupt:
                    // high-to-low
                    // after reset undefined

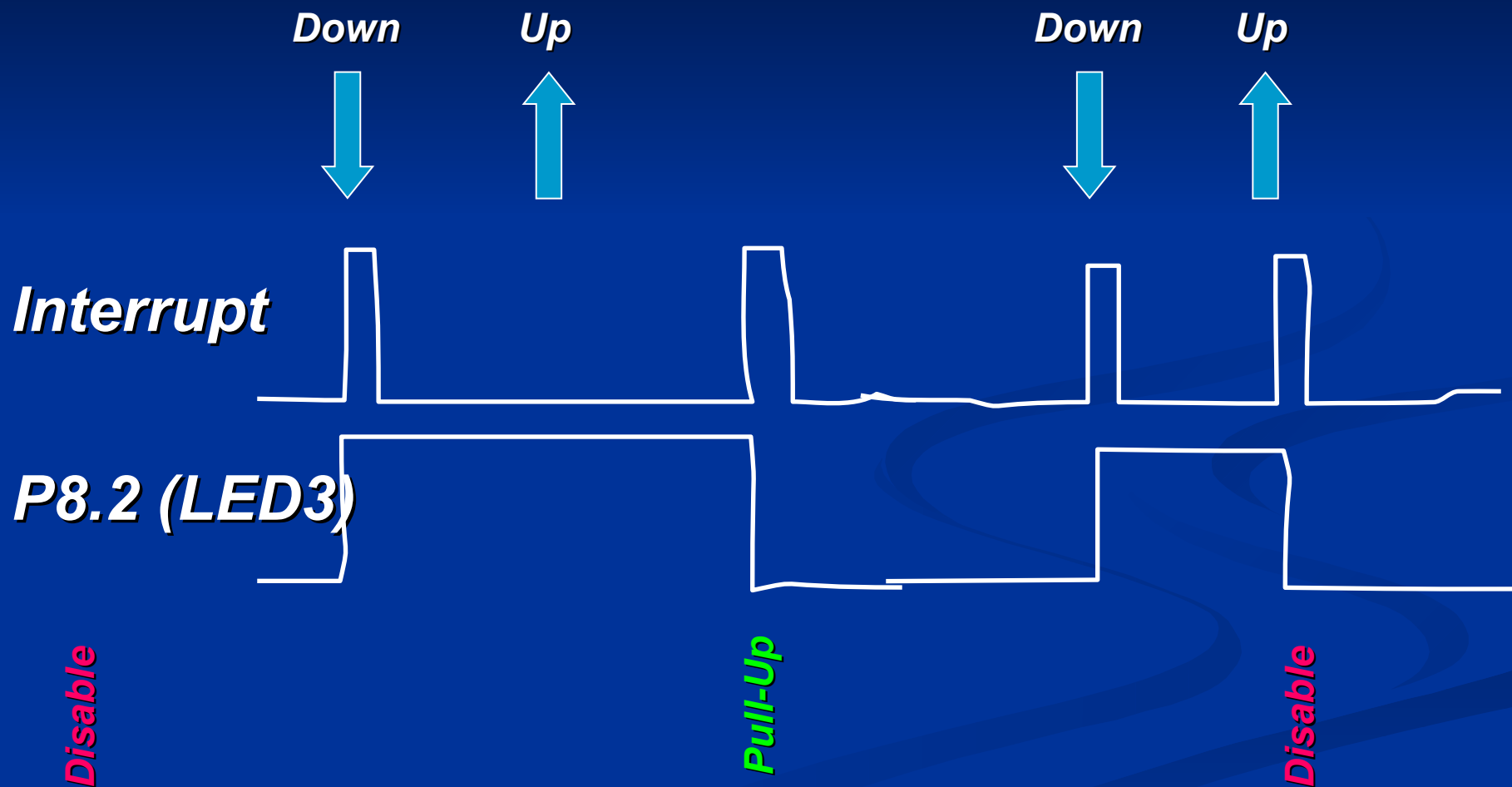
    // Enter LPM0, enable interrupts
    __bis_SR_register(LPM0_bits + GIE);
    __no_operation(); // For debugger
    return 0;
}
```

Управление выводами (портами) MSP430F5529

```
#pragma vector=PORT1_VECTOR
__interrupt void PORT1_ISR(void) {
if (P1IN & BIT7)      // high level - don't press button
    {P8OUT &= !(BIT2);} // off led
else P8OUT |= BIT2; // low level- press button- on led
if (!(P1IES & BIT7)) // if Low-to-High mode
    P1REN ^= BIT7;  // Enable/disable Pull-Up resistor
P1IES ^= BIT7;      // P1.7 Change Interrupt mode:
                    // High-to-Low, Low-to-High
P1IFG = 0;          // reset interrupt flag
}
```

- **Ответ на вопрос 3: по переднему фронту подтягивающий резистор ON / OFF**
- **Переключается режим возникновения прерывания: по спаду / по фронту**
- **Ответ на вопрос 4: чередование из-за подтягивающего резистора**

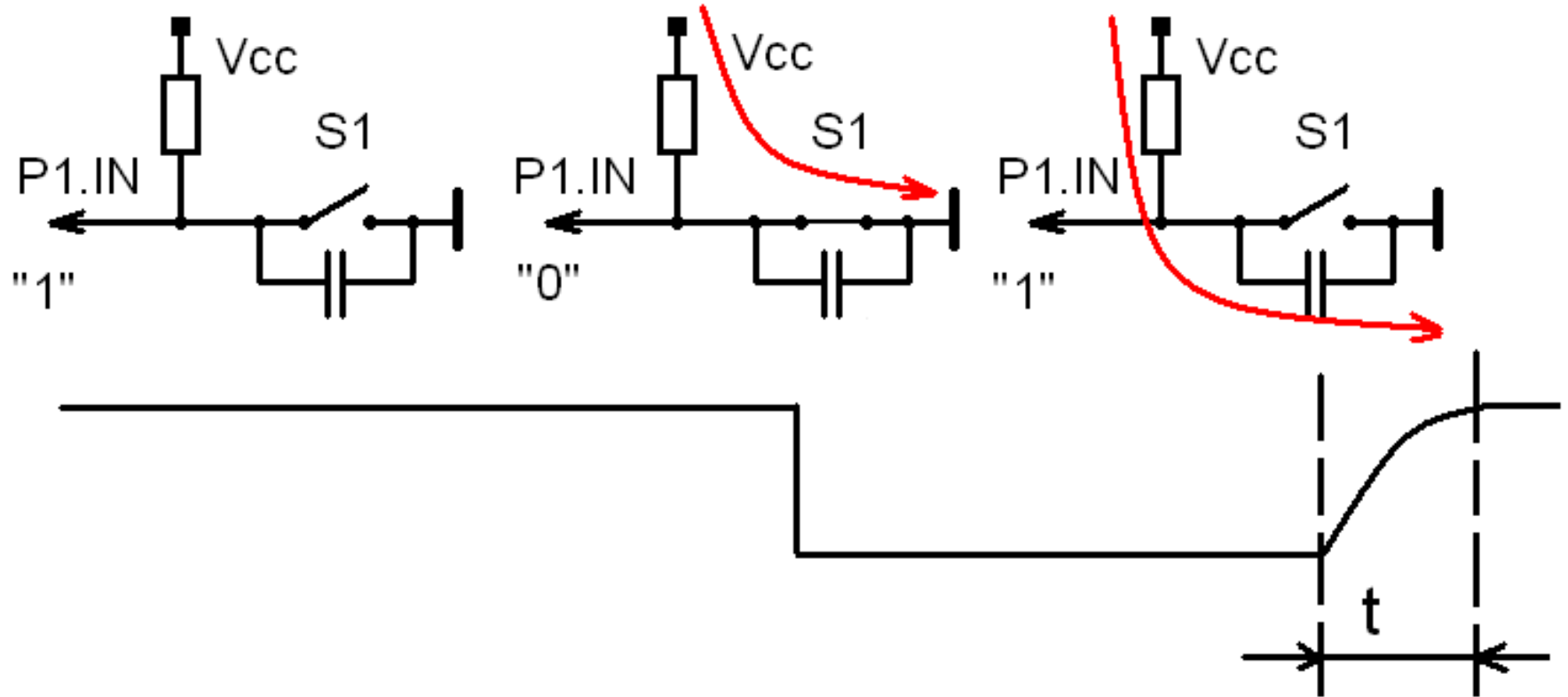
Управление выводами (портами) MSP430F5529



Эврика! Конденсатор!

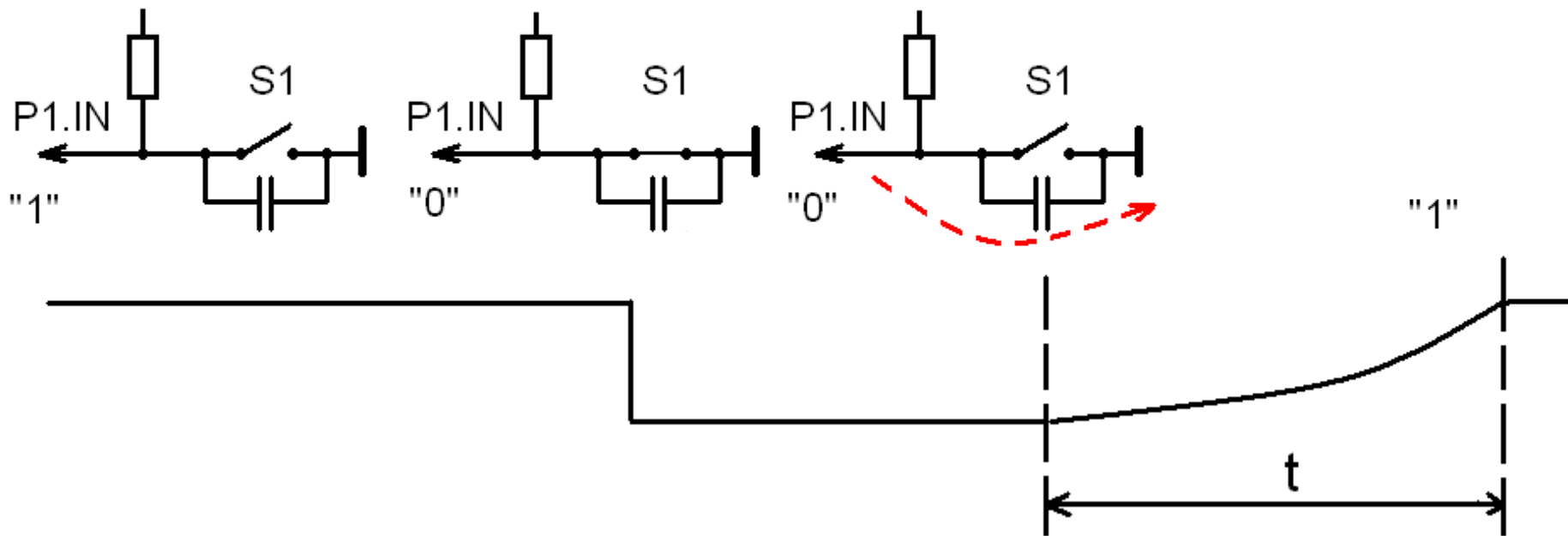
P1REN = "1"

P1OUT = "1"



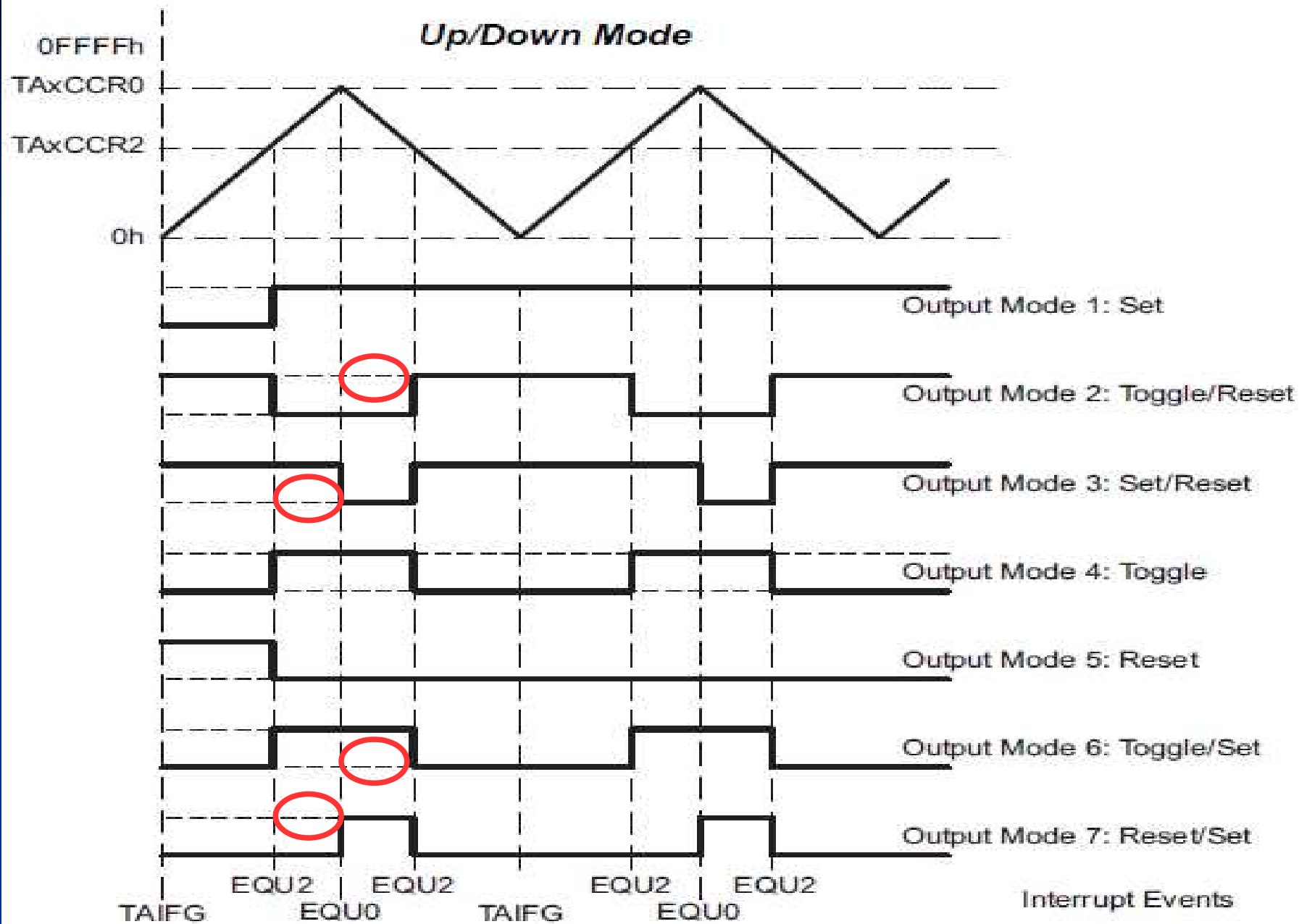
Эврика! Конденсатор!

P1REN = "0"

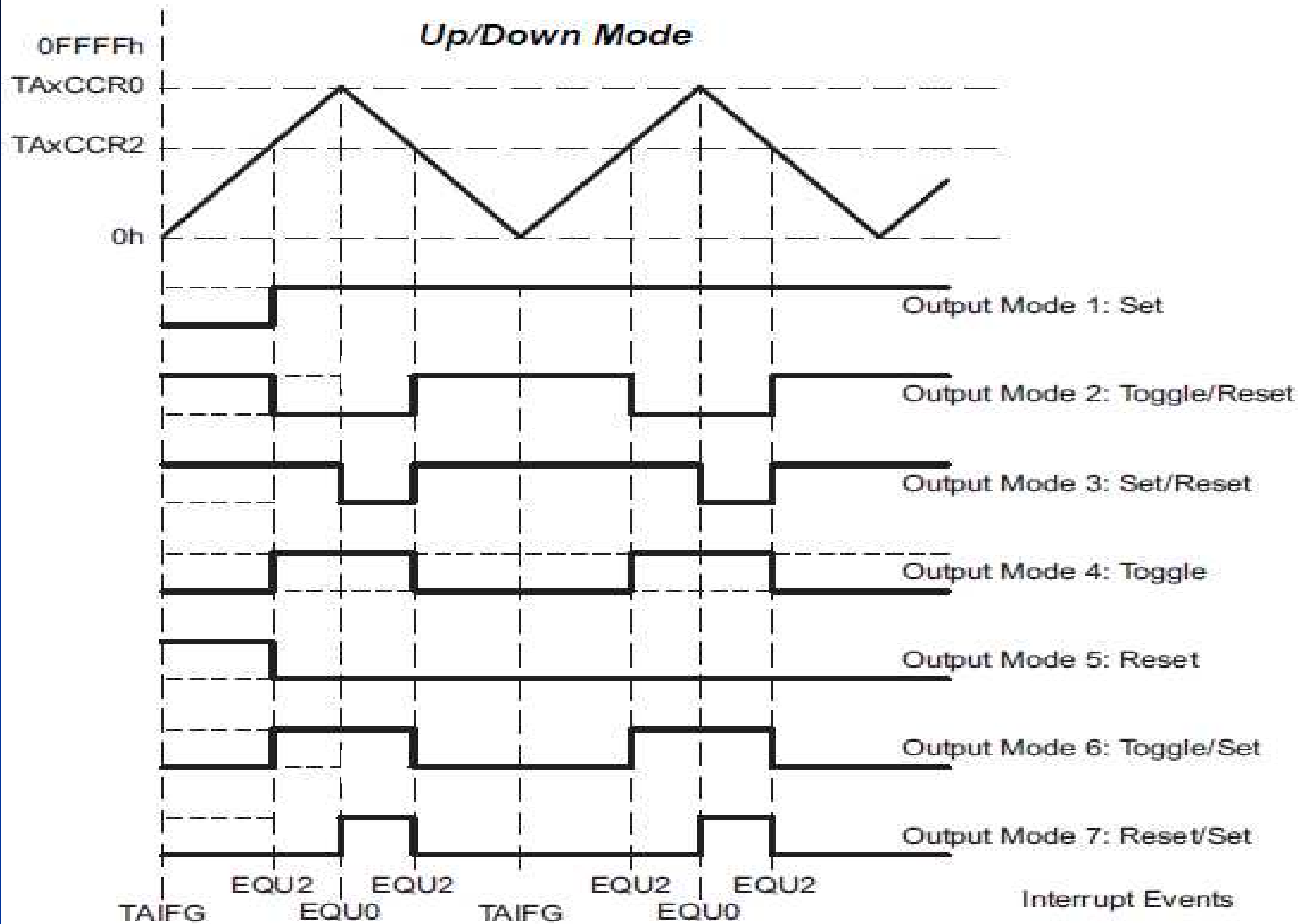


■ **Ответ на вопрос 1**

Таймер — задача 1



Таймер — задача 1



Таймер — задача 2, 3

■ Решение задачи

```
// SMCLK, divide by 4, contmode, clear TAR  
TA0CTL = TASSEL_2 | ID_2 | MC_2 | TACLR ;
```

DCOCLKDIV → SMCLK (после сброса)

DCORSEL = 2 после сброса (0,32 – 7,38 МГц)

SELREF = 0 после сброса, REFOCLK ← XT1

XT1 разделяет цифровые входы → после сброса резонатор недоступен // А что же будет с DCO?

После сброса DCOCLK = 2,097152 МГц

FFLD = 1 после сброса → DCOCLKDIV = DCOCLK/2 ~ 1 МГц

На входе: 1 МГц

Делитель на 4: 250 КГц // $262144 = 2^{18}$ Гц

Непрерывный режим — счет до FFFF = 2^{16} Гц

Частота мигания светодиодов = $2^{18} / 2^{16} = 4$

Нужно еще найти информацию, почему так