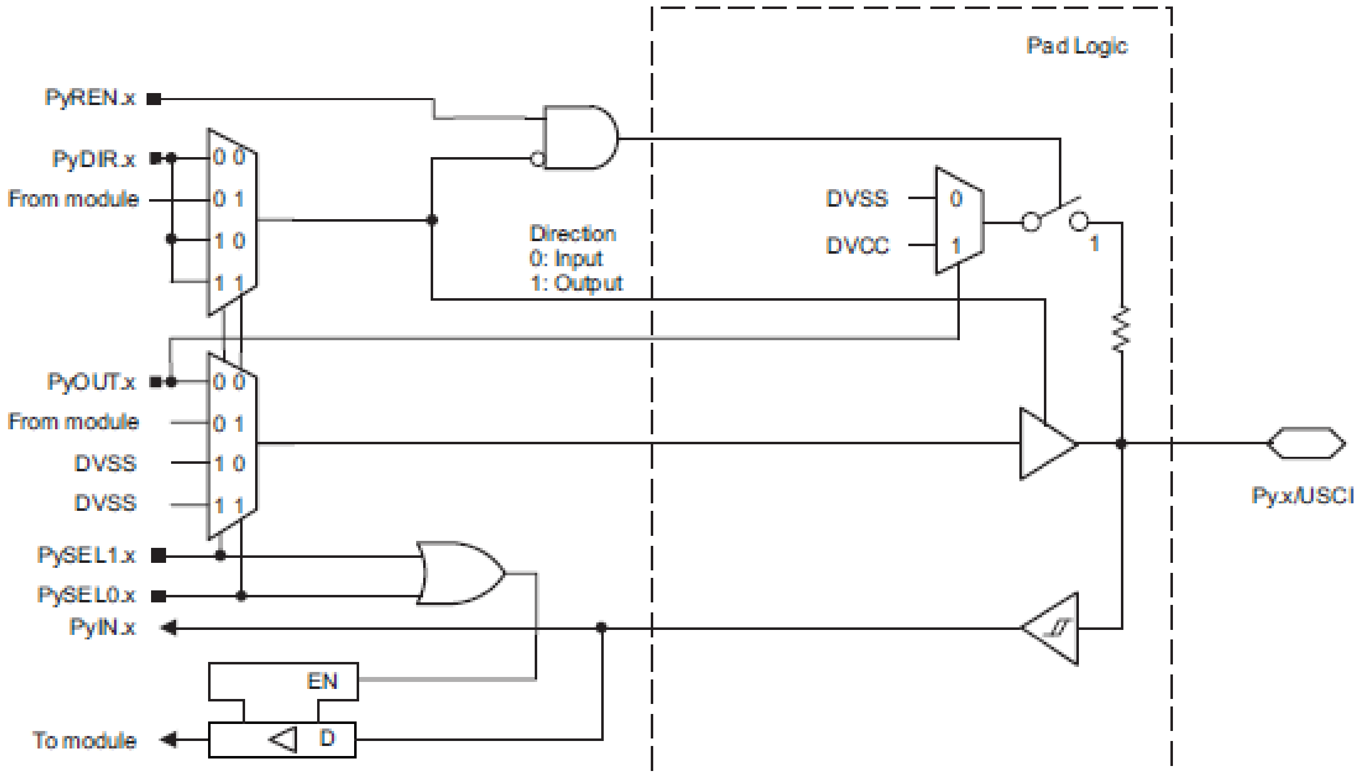
IO Ports

Last updated 5/16/19

IO Ports

- IO Structure (msp432)
  - Controlled by a series of registers

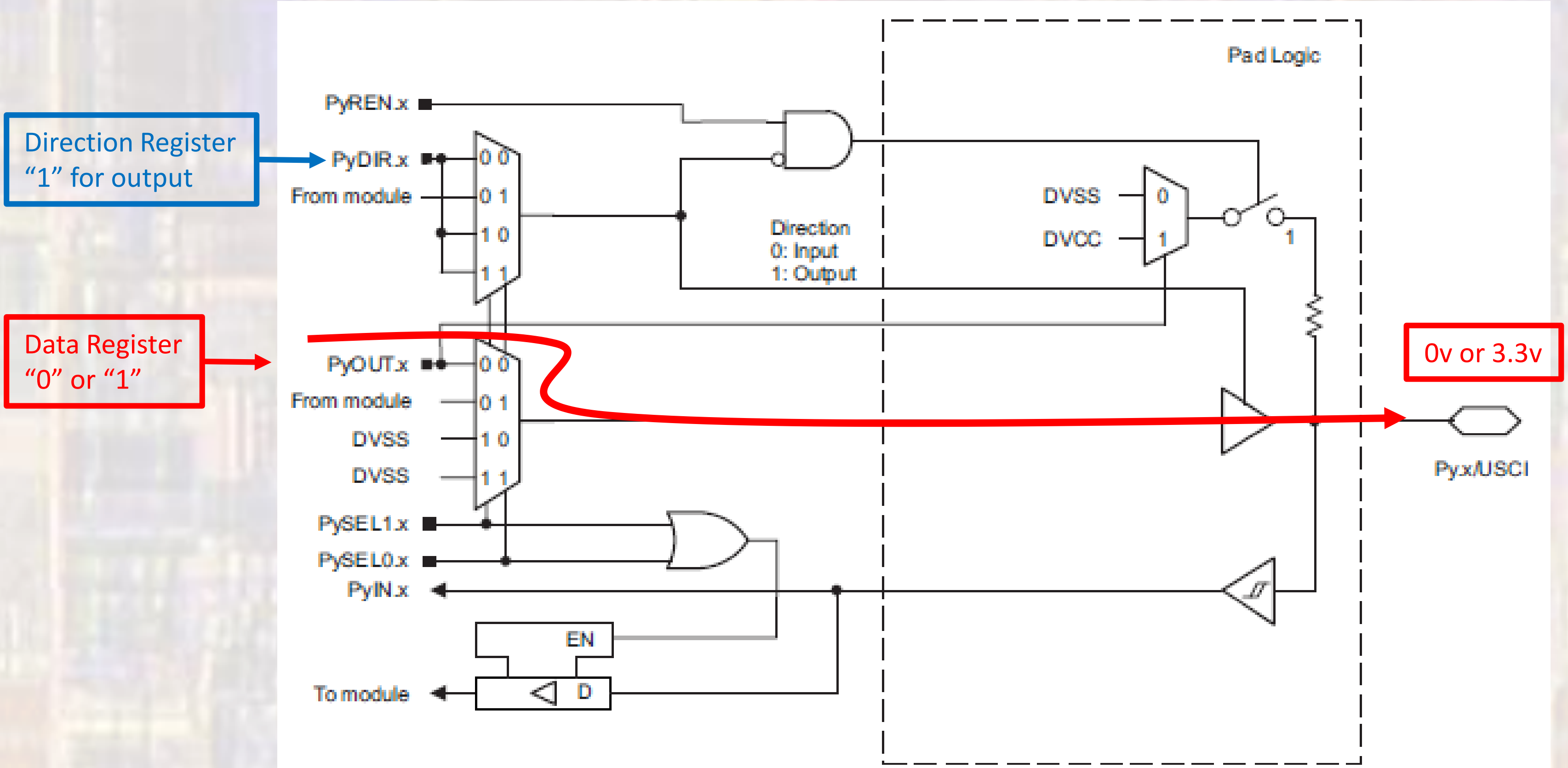


Functional representation only.

Figure 6-7. Py.x/USCI Pin Diagram

IO Ports

- IO Structure (msp432)
  - Output Path


Direction Register "1" for output

Data Register "0" or "1"

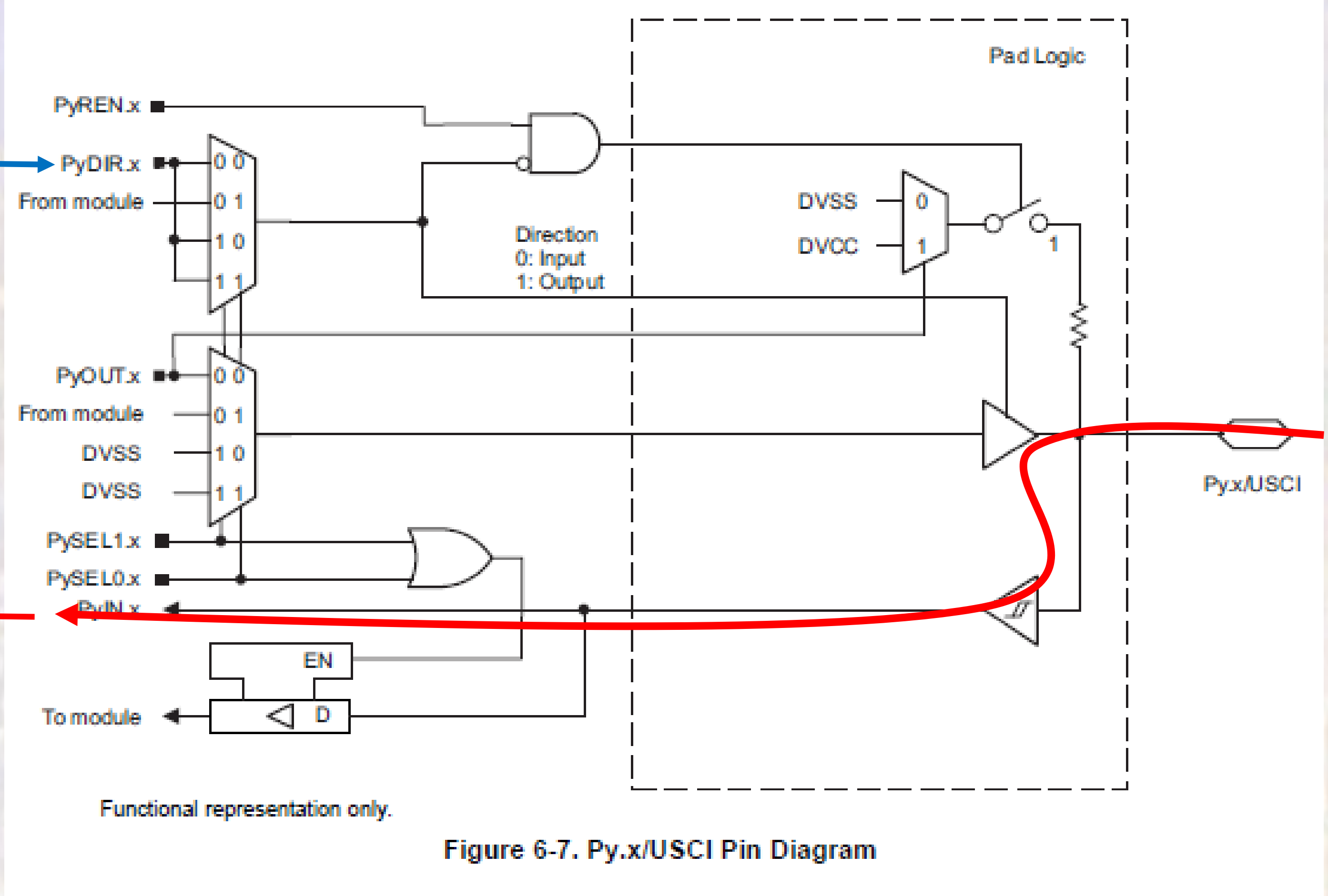
0v or 3.3v

Functional representation only.

Figure 6-7. Py.x/USCI Pin Diagram

IO Ports

- IO Structure (msp432)
  - Input Path



Direction Register
"0" for input

Input Register
"0" or "1"
depending on
pin voltage

Functional representation only.

Figure 6-7. Py.x/USCI Pin Diagram

IO Ports

- IO Structure (msp432)
  - Input Path w/ Pull-up

Pull Up/Down Register
“1” for Enable

Direction Register
“0” for input

Data Register
“1” for pull-up

Input Register
“0” or “1”
depending on
pin voltage

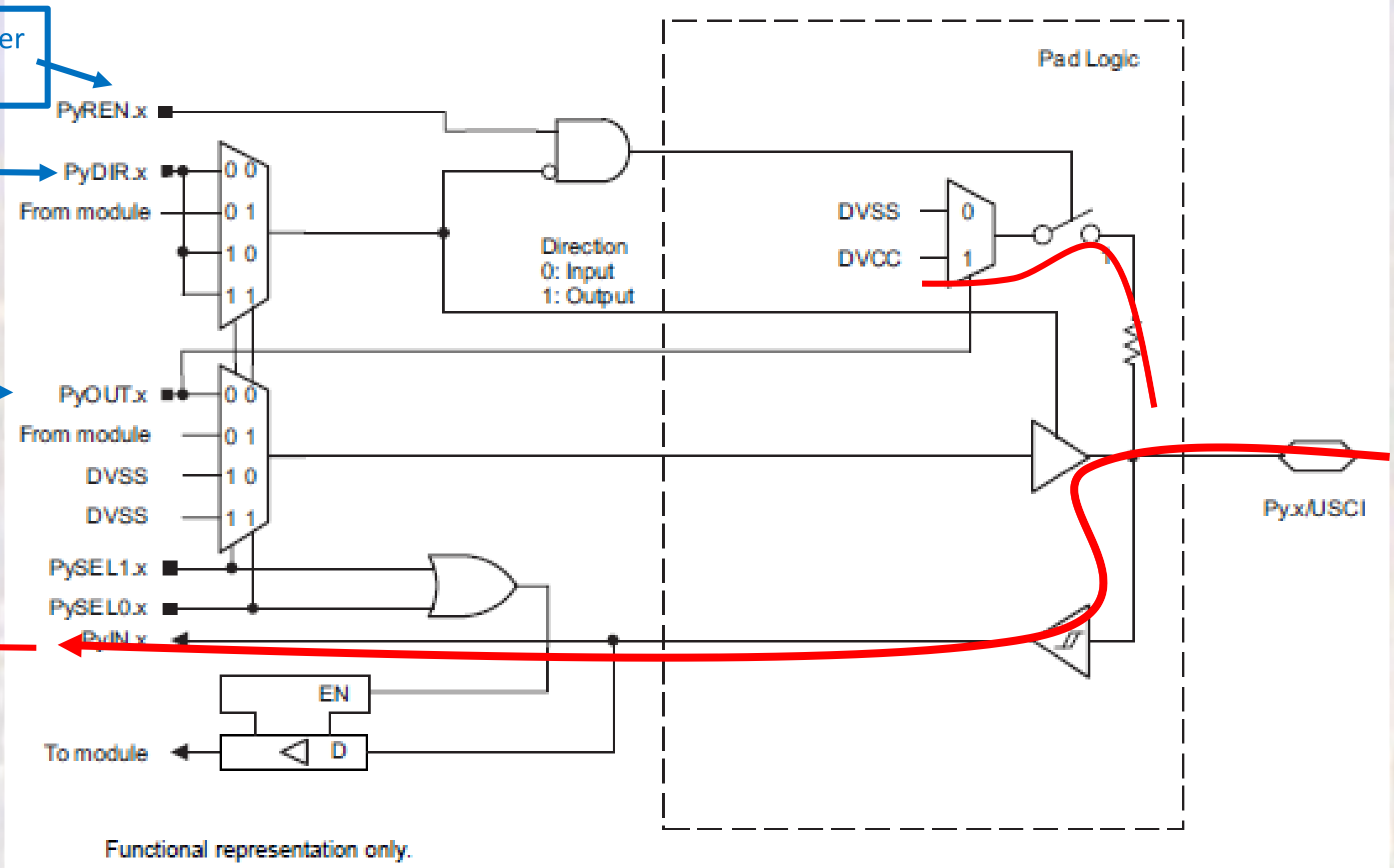


Figure 6-7. Py.x/USCI Pin Diagram

IO Ports

- IO Structure (msp432)
  - Input Path w/ Pull-down

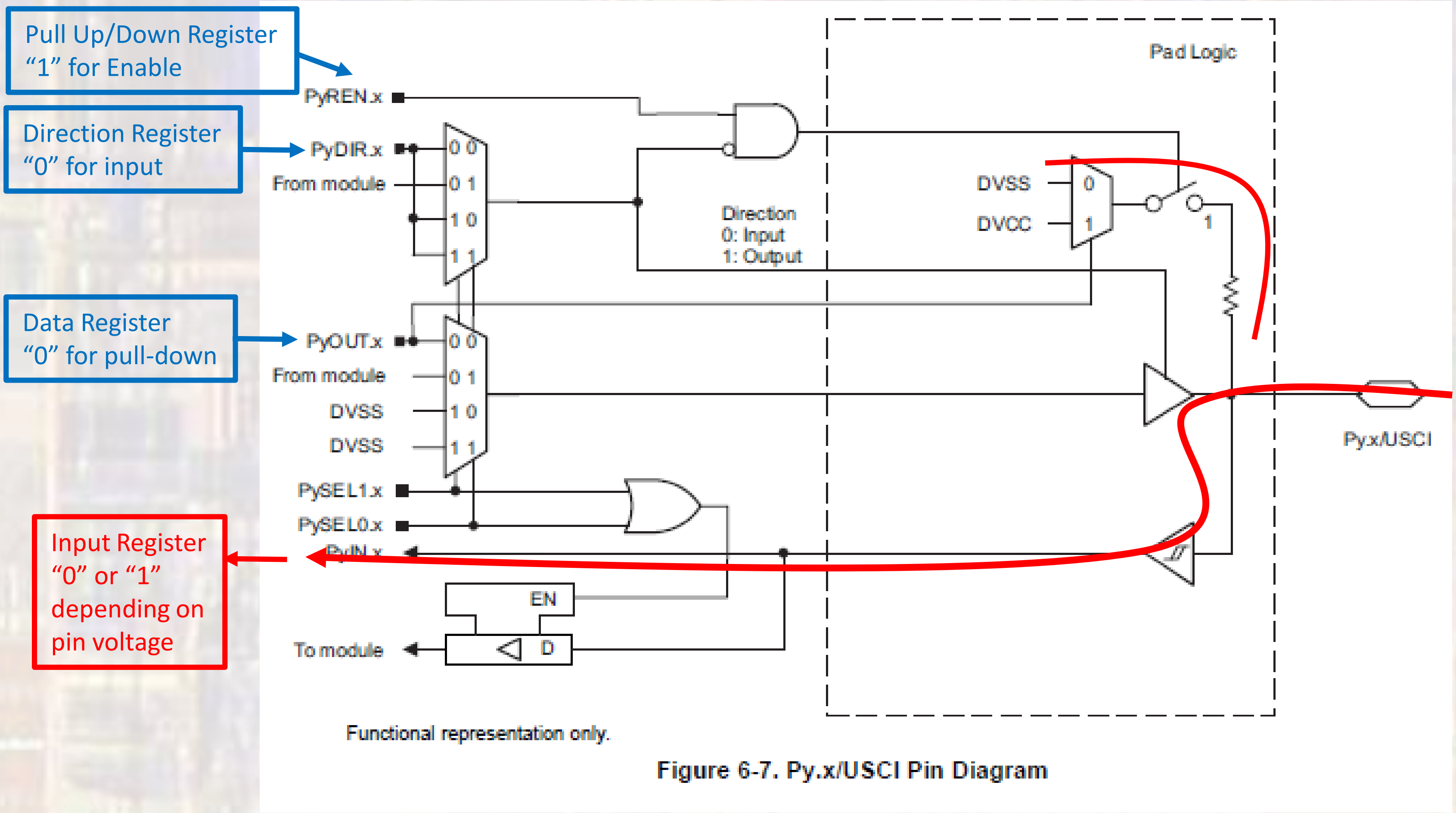


Figure 6-7. Py.x/USCI Pin Diagram

IO Ports

- Register Set (simplified)

  - 8 bit registers
    - 1 bit for each of 8 I/O pins
    - 8 I/O pins assigned to a port
    - 10 ports – 1 through 10

  - Register Names
    - P – port
    - x – port number
    - function – port function

    - P2->DIR → Port 2 Direction Register

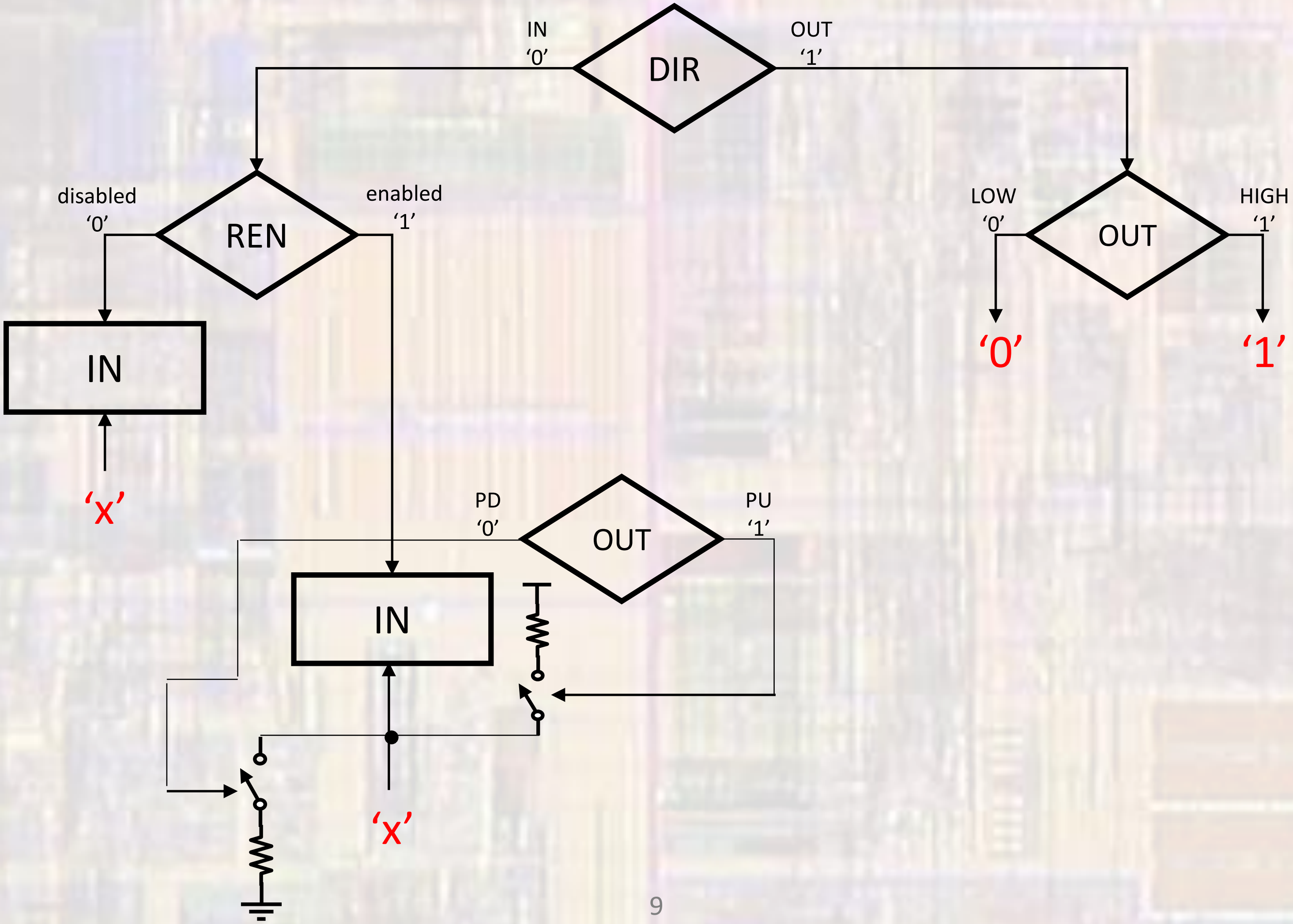
IO Ports

- Register Set (simplified)

  - Px->DIR – Direction register
    - “0” for input, “1” for output
  - Px->IN – Input Data Register
    - Holds the value of the input pin “0” or “1”
  - Px->OUT – Output Data Register
    - Holds the value for the output pin “0” or “1”
    - Holds the value of the pull-up/down when configured as an input and pull ups/down enabled
      - “0” for pull down, “1” for pull up
  - Px->REN – Enable Pull-Up/Down Register
    - “1” for enabled – only effects the pin in Input mode

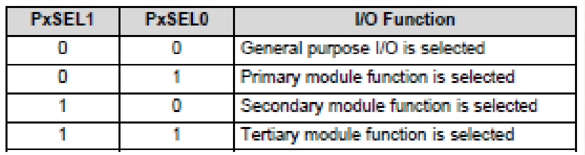
IO Ports

- Register Set (simplified)



IO Ports

- Register Set (advanced)
  - Px->DS – Drive Strength Register
    - “0” for regular strength, “1” for high drive strength
  - Px->SEL0, Px->SEL1 – Mode Select Register



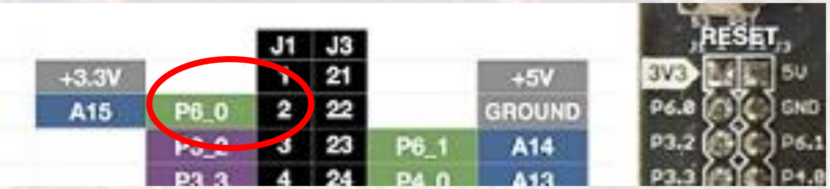
PxSEL1	PxSEL0	I/O Function
0	0	General purpose I/O is selected
0	1	Primary module function is selected
1	0	Secondary module function is selected
1	1	Tertiary module function is selected

  - Px->IE – Interrupt Enable Register
    - “1” for enable
  - Px->IES – Interrupt Edge Select Register
    - “0” for low → high, “1” for high → low
  - Px->IFG – Interrupt Flag Register
    - Set to “1” on selected edge transition
  - Px->IV – Interrupt Vector Register
    - Indicates the cause of an interrupt

IO Ports

- IO Port Configuration

Set pin 2 to an output



Pin 2 is actually Port 6, bit 0

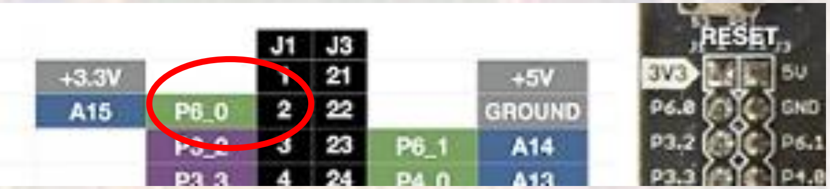
Need to set this pin to an output

```
   P6->DIR = 0x01;             // set port 6 bit 0 to an output
or
   P6->DIR = b00000001;        // set port 6 bit 0 to an output
or
   P6->DIR = 1;                // set port 6 bit 0 to an output
```

IO Ports

- IO Port Configuration

Set pin 2 to an output



Pin 2 is actually Port 6, bit 0

Need to set this pin to an output

```
   P6->DIR = 0x01;          // set port 6 bit 0 to an output
or
   P6->DIR = b00000001;     // set port 6 bit 0 to an output
or
   P6->DIR = 1;             // set port 6 bit 0 to an output
```

IO Ports

- IO Port Configuration

```
   P6DIR = 0x01;          // set port 6 bit 0 to an output
or
   P6DIR = b00000001;     // set port 6 bit 0 to an output
or
   P6DIR = 1;             // set port 6 bit 0 to an output
```

These set the other 7 pins to inputs – this may not be OK

```
P6->DIR |= 0x01;          // set port 6 bit 0 to an output
```

P6->DIR = bABCDEFGH | b00000001

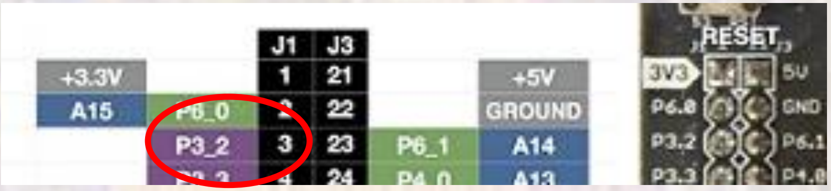
P6->DIR = bABCDEFG1

Only the bit we want to change is changed

IO Ports

- IO Port Configuration

Set pin 3 to an input



Pin 3 is actually Port 3, pin 2

```
P3->DIR &= 0xFB;            // set port 3 bit 2 to an input
```

or

```
P3->DIR &= b11111011;       // set port 3 bit 2 to an input
    P3->DIR = bABCDEFGH & b11111011
    P3->DIR = bABCDE0GH
```

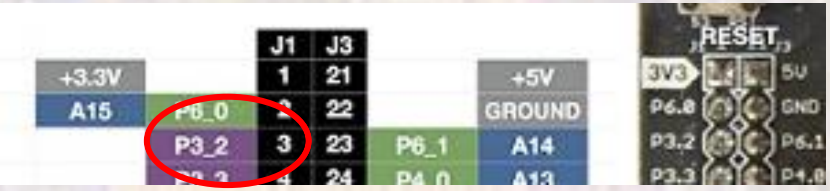
Only the bit we want to change is changed

Not intuitive

IO Ports

- IO Port Configuration

Set pin 3 to an input



Pin 3 is actually Port 3, pin 2

```
P3->DIR &= ~0x04;               // set port 3 bit 2 to an input
```

or

```
P3->DIR &= ~b0000100;           // set port 3 bit 2 to an input
```

```
P3->DIR = bABCDEFGH & ~b00000100
P3->DIR = bABCDEFGH & b11111011
P3->DIR = bABCDE0GH
```

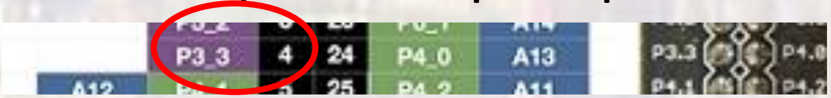
Only the bit we want to change is changed

More intuitive

IO Ports

- IO Port Configuration

Set pin 4 to input with pullup



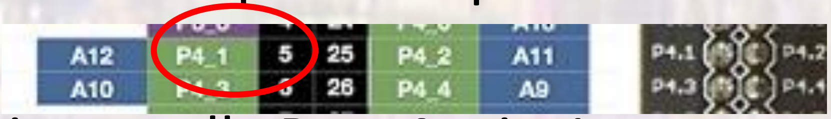
Pin 4 is actually Port 3, pin 3

```
P3->DIR &= ~0x08;          // set port 3 bit 3 to an input
P3->OUT |= 0x08;           // prepare to make pullup
P3->REN |= 0x08            // enable pull_x for port 3 pin 3
```

IO Ports

- IO Port Configuration

Set pin 5 to input with pull down



Pin 5 is actually Port 4, pin 1

```
P4->DIR &= ~0x02;               // set port 4 bit 1 to an input
P4->OUT &= ~0x02;               // prepare to make pulldown
P4->REN |= 0x02                 // enable pull_x for port 4 pin 1
```

IO Ports

- IO usage

Write a square wave to pin 5 to drive an LED

```
#include “msp.h”

…
P4->DIR |= 0x02;                 // set port 4 bit 1 (pin 5) to an output
P4->OUT &= ~0x02;                // initialize output to 0
…

while(1){
  _delay_cycles(1500000);
  P4->OUT |= 0x02;               // high
 _delay_cycles(1500000);
  P4->OUT &= ~0x02;              // low
}
```

IO Ports

- IO usage

Read from pin 5 as an input

```
#include “msp.h”

…
P4->DIR &= ~0x02;                // set port 4 bit 1 (pin 5) to an input
P4->REN &= ~0x02;                // Ensure pullx not enabled
…

while(1){
  int foo;

  foo = P4->IN & 0x02;           // check only bit 1
                                 // if high – foo has the value 0x02 NOT 1
  // or

  foo = (P4->IN & 0x02) && 0x01; // if high – foo has the value 0x01
}
```

IO Ports

- IO usage

  Make a decision based on pin 5 as an input

```
#include “msp.h”

…
P4->DIR &= ~0x02;                    // set port 4 bit 1 (pin 5) to an input
P4->REN &= ~0x02;                    // Ensure pullx not enabled
…


while(1){

  if ((P4->IN & 0x02) != 0)
    //do this if high
  else
    // do this if low

// or

  if (P4->IN & 0x02)
    //do this if high
  else
    // do this if low
}
```

THIS WILL NOT WORK

```
  if ((P4->IN & 0x02) == 1)
    //do this if high
  else
    // do this if low
```

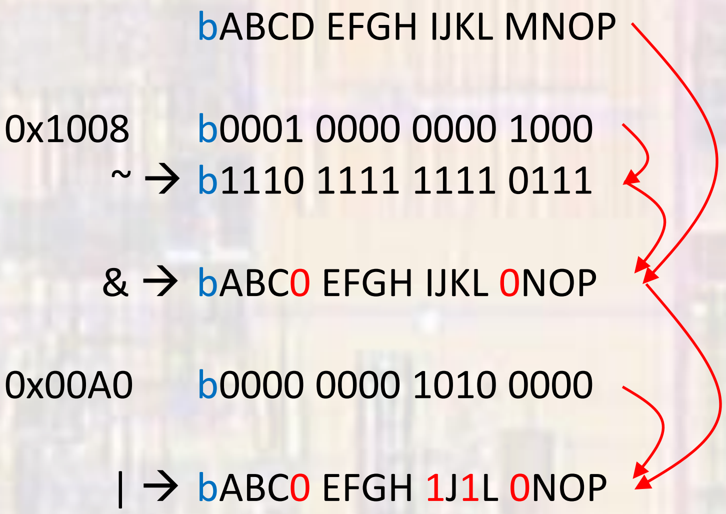
the test checks
0x00==1 or 0x02 ==1
never true

IO Ports

- Register modification
  - Change a few bits in a register to 1s and 0s

```
// Modify bits 5 and 7 to 1s, and 3 and 12 to zeros
// without altering any other bits

TIMER_A1->CTL = ((TIMER_A1->CTL & ~0x1008) | 0x00A0)
```



```
                bABCD EFGH IJKL MNOP

0x1008          b0001 0000 0000 1000
         ~ →    b1110 1111 1111 0111

         & →    bABC0 EFGH IJKL 0NOP

0x00A0          b0000 0000 1010 0000

         | →    bABC0 EFGH 1J1L 0NOP
```

IO Ports

- IO memory map

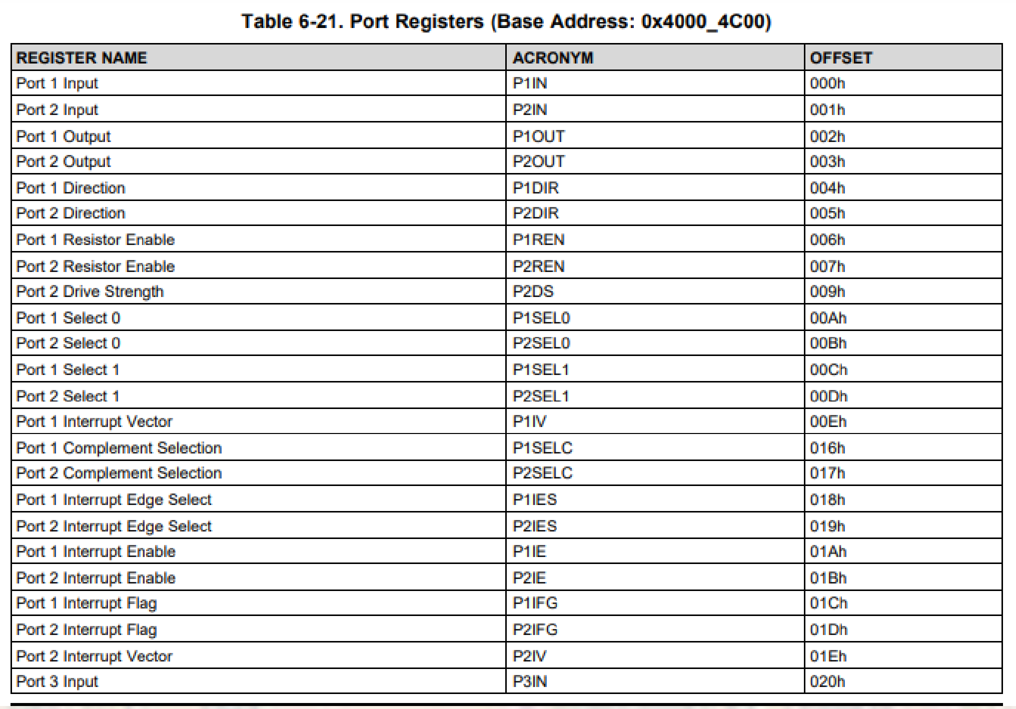


Table 6-21. Port Registers (Base Address: 0x4000_4C00)

REGISTER NAME	ACRONYM	OFFSET
Port 1 Input	P1IN	000h
Port 2 Input	P2IN	001h
Port 1 Output	P1OUT	002h
Port 2 Output	P2OUT	003h
Port 1 Direction	P1DIR	004h
Port 2 Direction	P2DIR	005h
Port 1 Resistor Enable	P1REN	006h
Port 2 Resistor Enable	P2REN	007h
Port 2 Drive Strength	P2DS	009h
Port 1 Select 0	P1SEL0	00Ah
Port 2 Select 0	P2SEL0	00Bh
Port 1 Select 1	P1SEL1	00Ch
Port 2 Select 1	P2SEL1	00Dh
Port 1 Interrupt Vector	P1IV	00Eh
Port 1 Complement Selection	P1SELC	016h
Port 2 Complement Selection	P2SELC	017h
Port 1 Interrupt Edge Select	P1IES	018h
Port 2 Interrupt Edge Select	P2IES	019h
Port 1 Interrupt Enable	P1IE	01Ah
Port 2 Interrupt Enable	P2IE	01Bh
Port 1 Interrupt Flag	P1IFG	01Ch
Port 2 Interrupt Flag	P2IFG	01Dh
Port 2 Interrupt Vector	P2IV	01Eh
Port 3 Input	P3IN	020h

IO Ports

- IO memory map

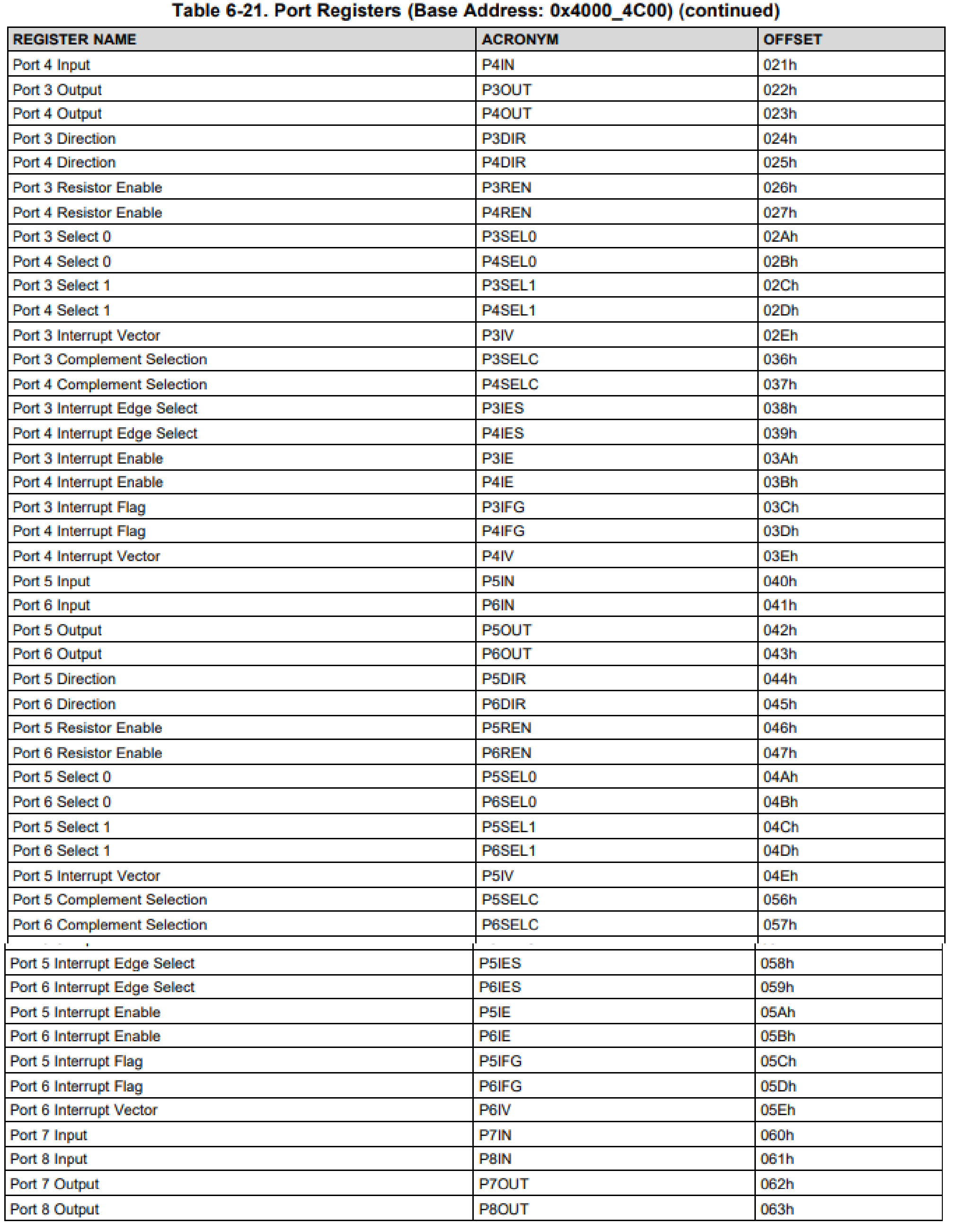


Table 6-21. Port Registers (Base Address: 0x4000_4C00) (continued)

REGISTER NAME	ACRONYM	OFFSET
Port 4 Input	P4IN	021h
Port 3 Output	P3OUT	022h
Port 4 Output	P4OUT	023h
Port 3 Direction	P3DIR	024h
Port 4 Direction	P4DIR	025h
Port 3 Resistor Enable	P3REN	026h
Port 4 Resistor Enable	P4REN	027h
Port 3 Select 0	P3SEL0	02Ah
Port 4 Select 0	P4SEL0	02Bh
Port 3 Select 1	P3SEL1	02Ch
Port 4 Select 1	P4SEL1	02Dh
Port 3 Interrupt Vector	P3IV	02Eh
Port 3 Complement Selection	P3SELC	036h
Port 4 Complement Selection	P4SELC	037h
Port 3 Interrupt Edge Select	P3IES	038h
Port 4 Interrupt Edge Select	P4IES	039h
Port 3 Interrupt Enable	P3IE	03Ah
Port 4 Interrupt Enable	P4IE	03Bh
Port 3 Interrupt Flag	P3IFG	03Ch
Port 4 Interrupt Flag	P4IFG	03Dh
Port 4 Interrupt Vector	P4IV	03Eh
Port 5 Input	P5IN	040h
Port 6 Input	P6IN	041h
Port 5 Output	P5OUT	042h
Port 6 Output	P6OUT	043h
Port 5 Direction	P5DIR	044h
Port 6 Direction	P6DIR	045h
Port 5 Resistor Enable	P5REN	046h
Port 6 Resistor Enable	P6REN	047h
Port 5 Select 0	P5SEL0	04Ah
Port 6 Select 0	P6SEL0	04Bh
Port 5 Select 1	P5SEL1	04Ch
Port 6 Select 1	P6SEL1	04Dh
Port 5 Interrupt Vector	P5IV	04Eh
Port 5 Complement Selection	P5SELC	056h
Port 6 Complement Selection	P6SELC	057h
Port 5 Interrupt Edge Select	P5IES	058h
Port 6 Interrupt Edge Select	P6IES	059h
Port 5 Interrupt Enable	P5IE	05Ah
Port 6 Interrupt Enable	P6IE	05Bh
Port 5 Interrupt Flag	P5IFG	05Ch
Port 6 Interrupt Flag	P6IFG	05Dh
Port 6 Interrupt Vector	P6IV	05Eh
Port 7 Input	P7IN	060h
Port 8 Input	P8IN	061h
Port 7 Output	P7OUT	062h
Port 8 Output	P8OUT	063h

IO Ports

- IO memory map

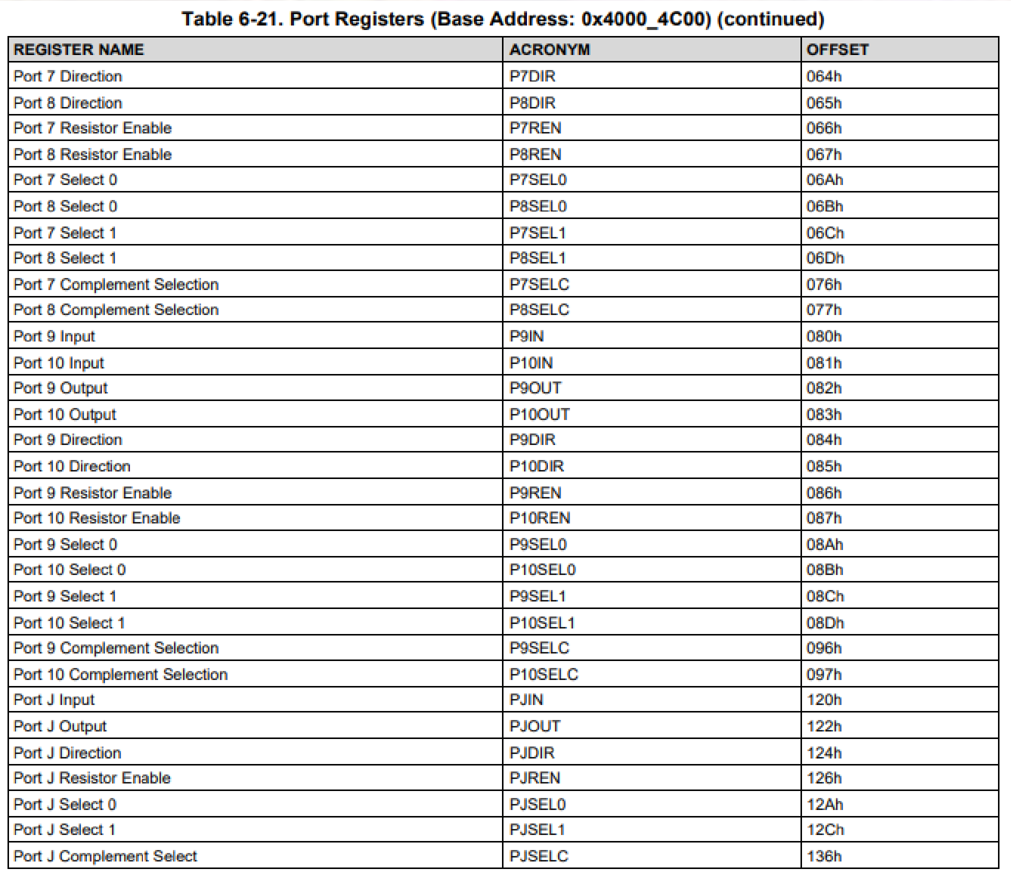


Table 6-21. Port Registers (Base Address: 0x4000_4C00) (continued)

REGISTER NAME	ACRONYM	OFFSET
Port 7 Direction	P7DIR	064h
Port 8 Direction	P8DIR	065h
Port 7 Resistor Enable	P7REN	066h
Port 8 Resistor Enable	P8REN	067h
Port 7 Select 0	P7SEL0	06Ah
Port 8 Select 0	P8SEL0	06Bh
Port 7 Select 1	P7SEL1	06Ch
Port 8 Select 1	P8SEL1	06Dh
Port 7 Complement Selection	P7SELC	076h
Port 8 Complement Selection	P8SELC	077h
Port 9 Input	P9IN	080h
Port 10 Input	P10IN	081h
Port 9 Output	P9OUT	082h
Port 10 Output	P10OUT	083h
Port 9 Direction	P9DIR	084h
Port 10 Direction	P10DIR	085h
Port 9 Resistor Enable	P9REN	086h
Port 10 Resistor Enable	P10REN	087h
Port 9 Select 0	P9SEL0	08Ah
Port 10 Select 0	P10SEL0	08Bh
Port 9 Select 1	P9SEL1	08Ch
Port 10 Select 1	P10SEL1	08Dh
Port 9 Complement Selection	P9SELC	096h
Port 10 Complement Selection	P10SELC	097h
Port J Input	PJIN	120h
Port J Output	PJOUT	122h
Port J Direction	PJDIR	124h
Port J Resistor Enable	PJREN	126h
Port J Select 0	PJSEL0	12Ah
Port J Select 1	PJSEL1	12Ch
Port J Complement Select	PJSELC	136h