CS4101 嵌入式系統概論

General Purpose IO

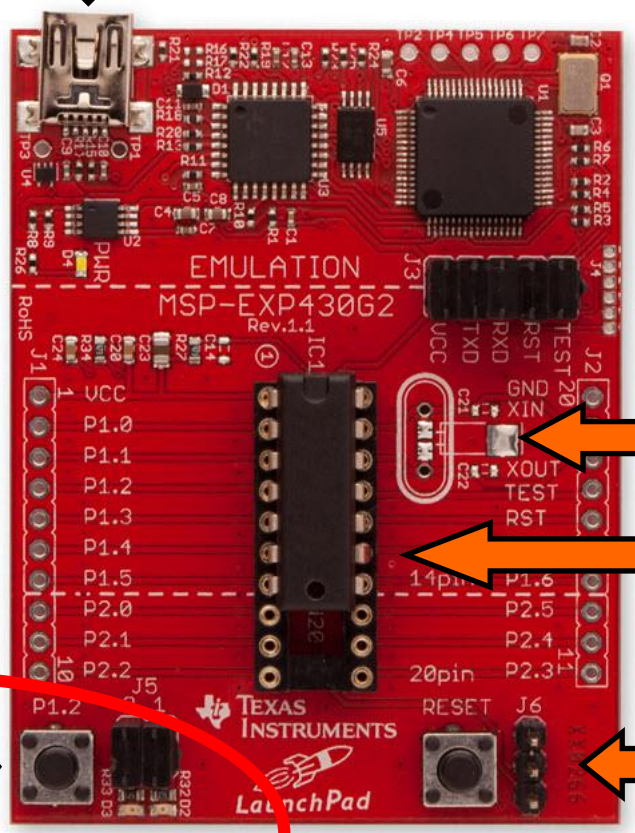
Prof. Chung-Ta King

Department of Computer Science

National Tsing Hua University, Taiwan

Materials from *MSP430 Microcontroller Basics*, John H. Davies, Newnes, 2008

LaunchPad Development Board



USB Emulator Connection

Embedded Emulation

6-pin eZ430 Connector

Crystal Pads

Chip Pinouts

Part and Socket

P1.3 Button

Power Connector

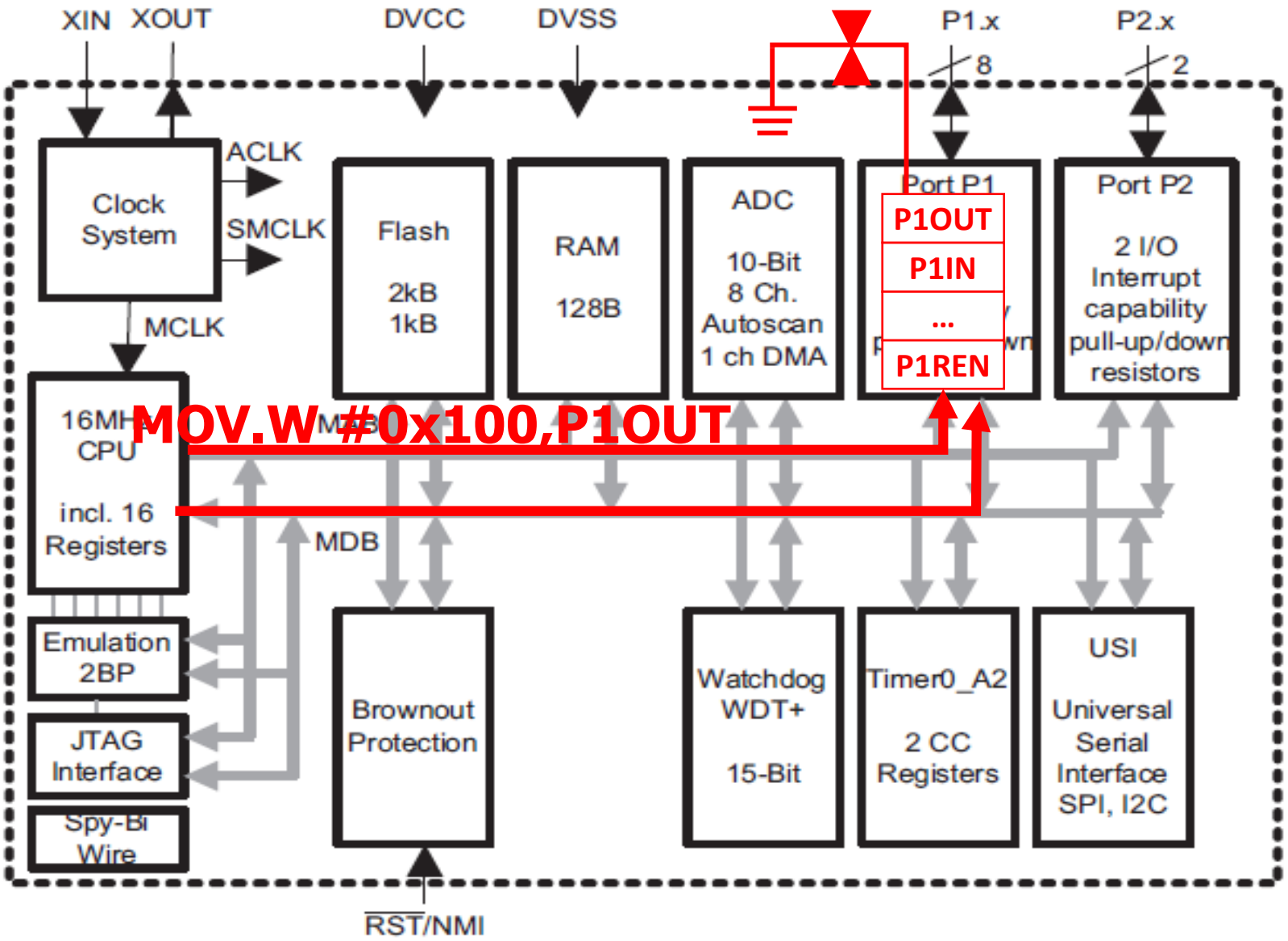
LEDs P1.0 & P1.6

Reset Button

Recall: Sample Code for Output

```
#include <msp430.h>
void main(void) {
  WDTCTL = WDTPW + WDTHOLD; // Stop watchdog
  P1DIR |= 0x41;  // set P1.0 & 6 to outputs
  for(;;) {
     volatile unsigned int i;
     P1OUT ^= 0x41; // Toggle P1.0 & 6
     i = 50000;      // Delay
     do (i--);
     while (i != 0);
  }
}
```

Recall: Memory-Mapped I/O

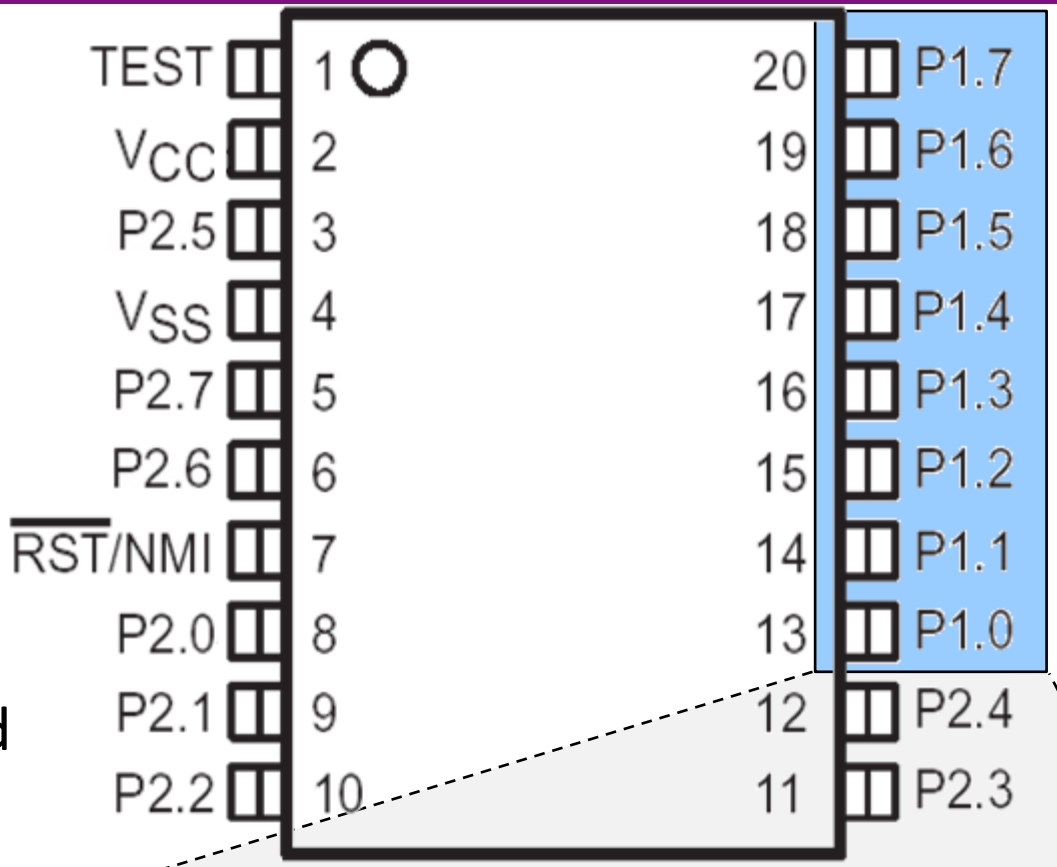


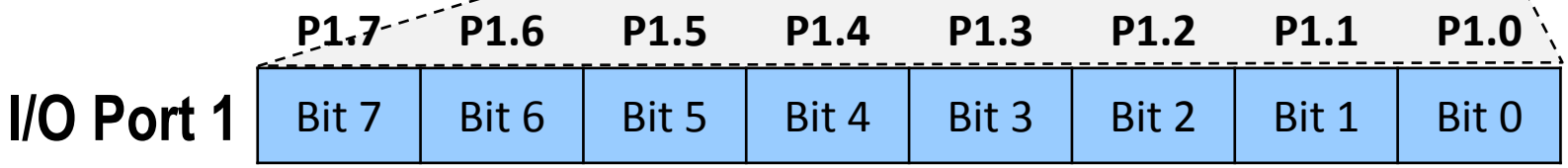
Configuring the I/O Ports

Registers (Mem Addr)	Functions	Descriptions
P1IN (0x0020)	Port *1* input	This is a read-only register that reflects the current state of the port's pins.
P*1*OUT (0x0021)	Port *1* output	The values written to this read/write register are driven out to corresponding pins when they are configured to output.
P*1*DIR (0x0022)	Port *1* data direction	Bits written as 1 (0) configure the corresponding pins for output (input).
P*1*SEL (0x0026)	Port *1* function select	Bits written as 1 (0) configure corresponding pins for use by the specialized peripheral (for general-purpose I/O).
P*1*REN (0x0027)	Port *1* resistor enable	Bits set in this register enable pull-up/down resistors on the corresponding I/O pins.

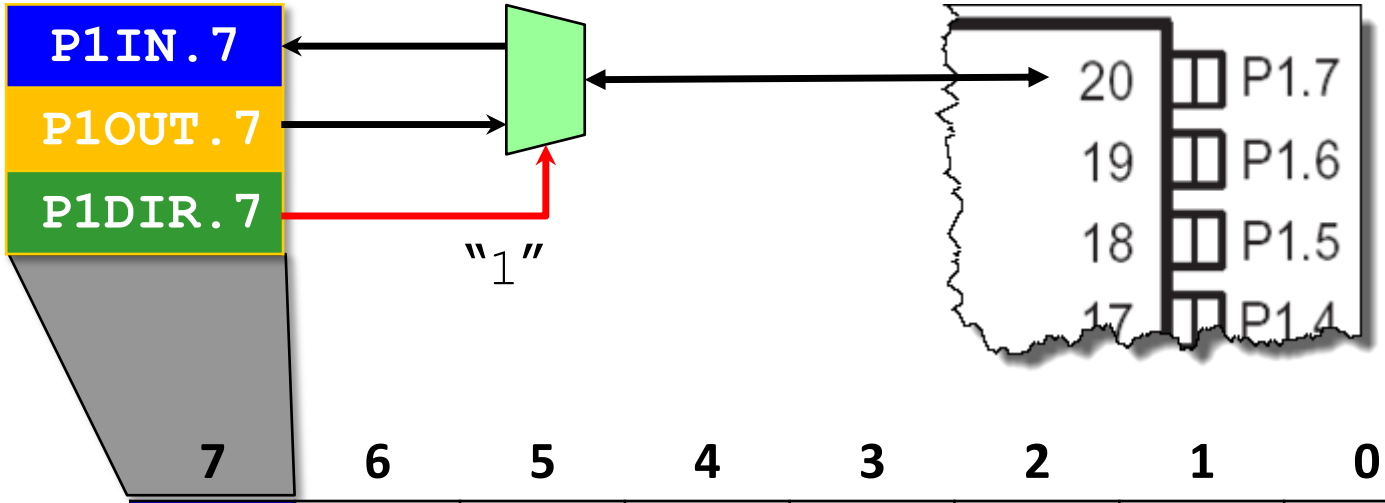
MSP430 GPIO

- GPIO = General Purpose Bit Input/Output
- 8-bit I/O ports
- Each pin is individually controllable
- Controlled by memory-mapped registers





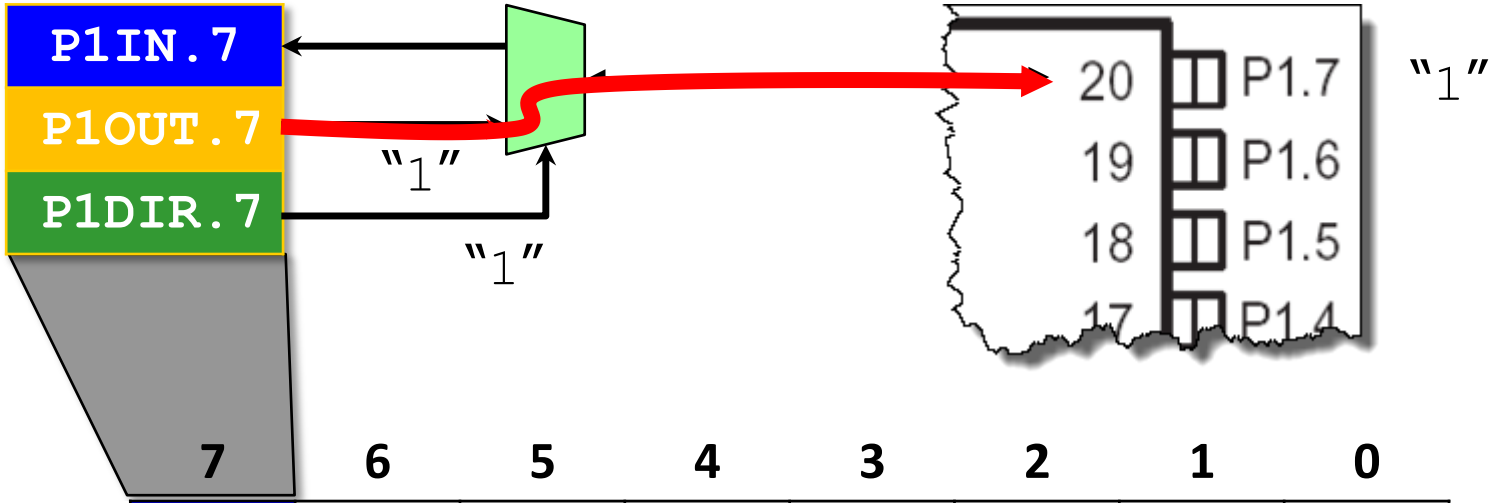
PxDIR (Pin Direction): Input or Output



	7	6	5	4	3	2	1	0
P1IN								
P1OUT								
P1DIR	1							1

- PxDIR.y: 0 = input 1 = output
- Register example: **`P1DIR &= 0x81;`**

GPIO Output



	7	6	5	4	3	2	1	0
P1IN	X							
P1OUT	1							
P1DIR	1							

- PxOUT.y: 0 = low 1 = high
- Register example: `P1OUT &= 0x80;`

Sample Code for Input

```
#include <msp430g2231.h>
// Pins for LED and button on port 1
#define LED1 BIT0    //P1.0 to red LED
#define B1 BIT3      //P1.3 to button

void main(void){
  WDTCTL = WDTPW + WDTHOLD; //Stop watchdog timer
  P1OUT |= LED1;  //Preload LED1 on
  P1DIR = LED1;   //Set pin with LED1 to output
  for(;;){   //Loop forever
    if((P1IN & B1) ==0){  //Is button down or not
      P1OUT &= ~LED1; // Turn LED1 off }
    else{
      P1OUT |= LED1; // T
  }
}
```

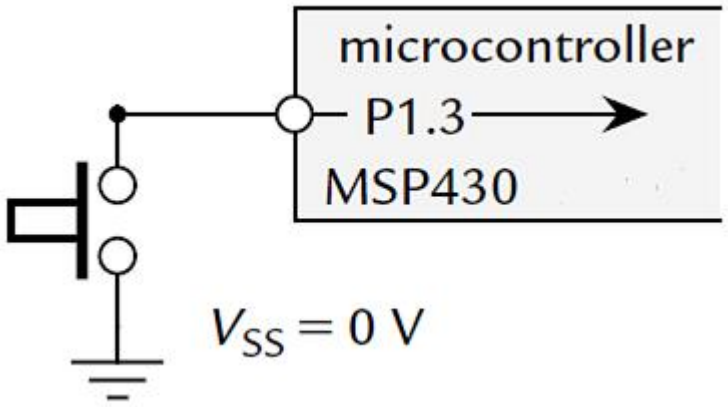
P1OUT is not initialized and must be written before configuring the pin for output.

Problem with Input Using a Button

- When the button is pressed down (closed), MSP430 will detect a 0 from the pin.
- When the button is up (open), what will MSP430 detect?

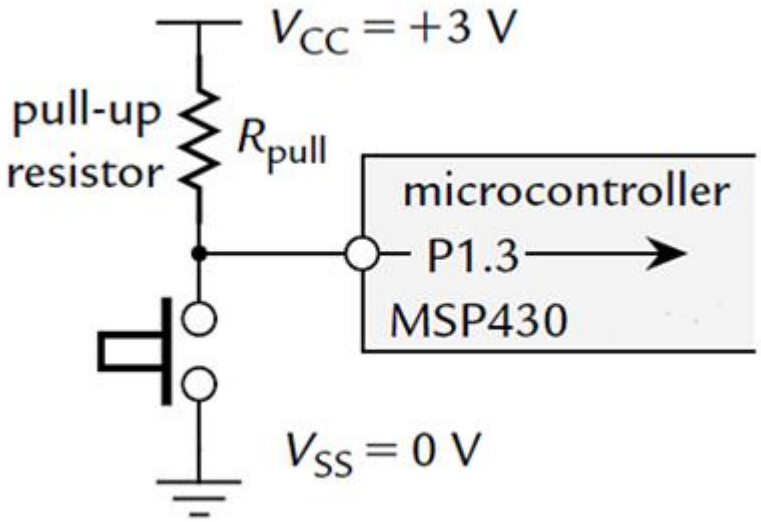
  Ans.: random value

  → Floating



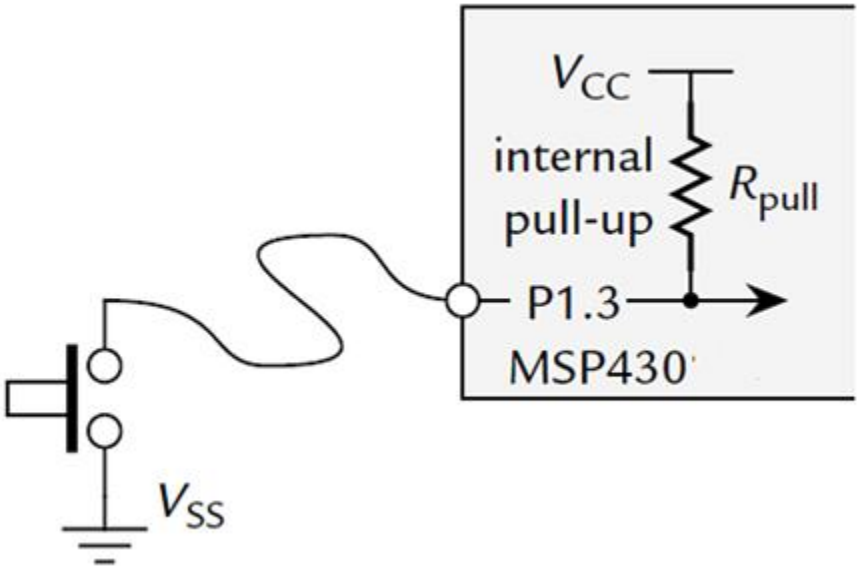
Typical Way of Connecting a Button

- The *pull-up resistor* R_{pull} holds input at logic 1 (voltage V_{CC}) while button is up and logic 0 or V_{SS} when button is down → *active low*
  - A wasted current flows through R_{pull} to ground when button is down
  - This is reduced by making R_{pull} large enough, typically *33 kΩ*



Typical Way of Connecting a Button

- MSP430 offers internal pull-up/down
- PxREN register selects whether this resistor is used (1 to enable, 0 to disable)
  - When enabled, the corresponding bit of PxOUT register selects whether the resistor pulls the input up to V_{CC} (1) or down to V_{SS} (0)

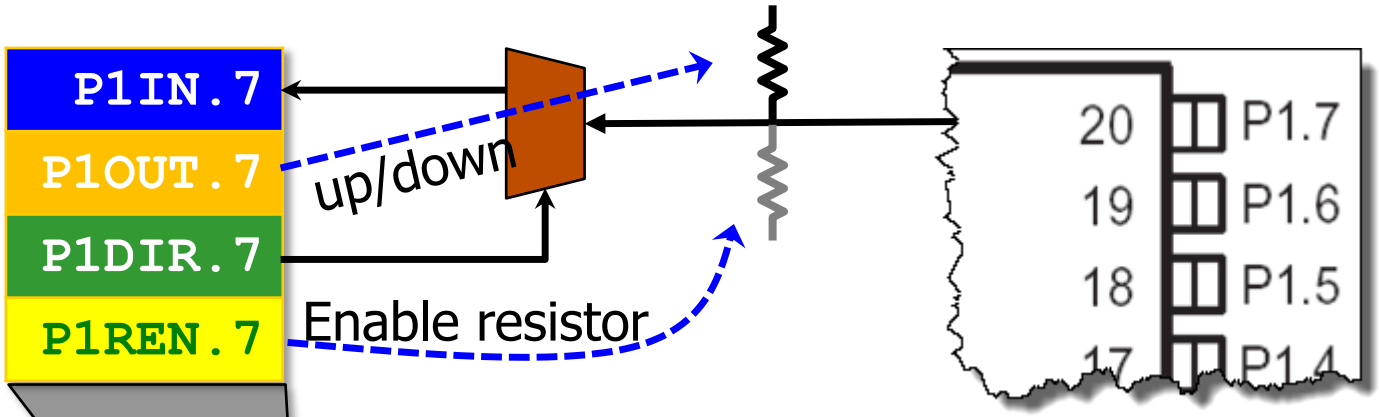


Sample Code for Input (MSP430g2553)

```
#include <msp430g2553.h>
#define LED1 BIT0    //P1.0 to red LED
#define B1 BIT3      //P1.3 to button

void main(void){
  WDTCTL = WDTPW + WDTHOLD; //Stop watchdog timer
  P1OUT |= LED1 + B1;
  P1DIR = LED1; //Set pin with LED1 to output
  P1REN = B1;    //Set pin to use pull-up resistor
  for(;;){    //Loop forever
    if((P1IN & B1) ==0){  //Is button down or not
      P1OUT &= ~LED1; // Turn LED1 off  }
    else{
      P1OUT |= LED1; // Turn LED1 on }
  }
}
```

GPIO Input



	7
P1IN	x
P1OUT	1
P1DIR	0
P1REN	1

- Input pins are held in high-impedance (Hi-Z) state, so they can react to 0 or 1
- When not driven, Hi-Z inputs may float up/down … prevent this with pullup/pulldown resistors
- PxREN enables resistors
  PxOUT selects pull-up (1) or -down (0)
- Lower cost devices may not provide up/down resistors for all ports