BMEN 428/ECEN 489
CSCE 489/BMEN 689

(G)eneral (P)urpose
(I)nput and (O)utput

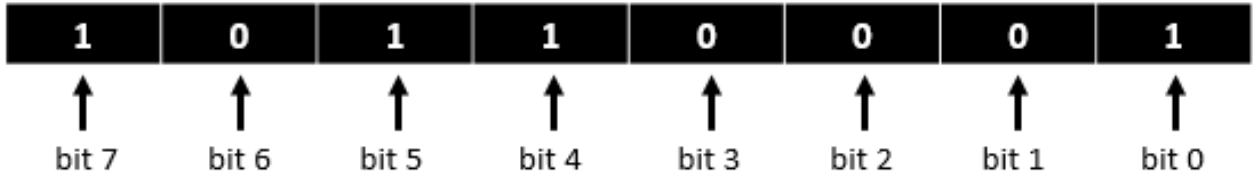
Please **SIGN IN** at the front!

Turn in all **POST LABS** at the front!

GPIO Basics

Register: memory unit containing a specified number of bits. An 8-bit register stores 8 bits (byte) or binary numbers.

- The MSP430 uses 8-bit registers to store functions, memory, and data.



In the MSP430, we'll primarily use **Port 1** for GPIO.

Each bit in the **Port 1** register(s) correspond(s) to a **PIN**, ex. P1.0 (pin 0), P1.1 (pin 1), etc…

- The family of the MSP430 we are using also has a Port 2 in case we need even more inputs and outputs but this is rarely used.

Example

General Port 1 Register

GPIO Registers

What does the "X" stand for?

Specifies the **NUMBER** of the port you are using…
e.g. P1DIR is for the direction register of Port 1.

1) **Direction Register (PXDIR)**
   a. Determines what PINS act as input and output.
   b. Bit 1 → port set as output.
   c. Bit 0 → port set as input.

input	input	input	output	input	input	output	output
0	0	0	1	0	0	1	1
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

2) **Input Register (PXIN)**
   a. When pins are configured as INPUTS (from PXDIR) each bit of these registers is a **READ-ONLY** bit and reflects the input signal at the pin.
   b. Bit 1 → input is HIGH.
   c. Bit 0 → input is LOW.

3) **Output Register (PXOUT)**
   a. Reflects the value that was written to the output pin; a **READ/WRITE** register.
   b. Bit 1 → input is HIGH.
   c. Bit 0 → input is LOW.

4) **Interrupt Enable (PXIE)**
   a. Enable interrupts on individual pins. Very important if your code uses interrupts and low power modes.

Hexadecimal Numbers

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Examples:

0xFF 1111 1111

0xF7 1111 0111

0xA3 1010 0011

0x00 0000 0000

0x03 0000 0011

0x14 0001 0100

Bitwise Operations

Work at the bit level, compare bit-to-bit, in contrast to expression-to-expression. Does not work with floating point numbers. **Why**?

Common operators in C: AND "**&**", "**|**" OR, "**^**" XOR, "**~**" one's complement, etc…

Common abbreviations for the MSP430:

BIT0 = 0 0 0 0 0 0 0 1
BIT1 = 0 0 0 0 0 0 1 0
BIT2 = 0 0 0 0 0 1 0 0

Examples

X = 1 0 0 0 1 1	A = 1 0 0 1 1 1 0 0
Y = 0 0 1 1 0 1	B = 1 0 0 0 0 0 0 0
X|Y = 1 0 1 1 1 1	**A|B** = 1 0 0 1 1 1 0 0
X&Y = 0 0 0 0 0 1	**A&B** = 1 0 0 0 0 0 0 0
~X = 0 1 1 1 0 0	**~A** = 0 1 1 0 0 0 1 1
~Y = 1 1 0 0 1 0	**~B** = 0 1 1 1 1 1 1 1
X^Y = 1 0 1 1 1 0	**A^B** = 0 0 0 1 1 1 0 0

Example Command Lines

P2DIR |= BIT3

- Technically means… P2DIR = P2DIR |BIT3
  - Setting direction of PINS in Port 2.
    - Setting P2.2 as OUTPUT, all other pins as INPUT.
      - Default is 0, which refers to an INPUT. Only way to change it to an OUTPUT is by making the bit a 1.

Example Command Lines

```
if (P1IN & BIT6) // Assume P1.6 as input pin…
        printf("Hello, World');
```

- If the input register (P1IN) notices a HIGH input at P1.6 (BIT6), then the MSP430 will print "Hello, World" into the console.

Example Command Lines

P1REN = 0x03;

- Enable a pull-up/pull-down resistor at P1.0 and P1.1.