MECH 423 Lecture #5

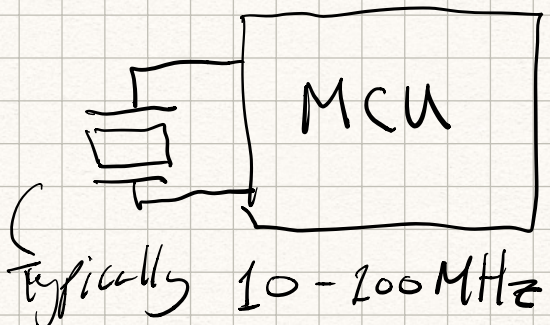
Microprocessor — CPU only

Microcontroller — CPU + peripherals (Memory, ADC, Comm...)

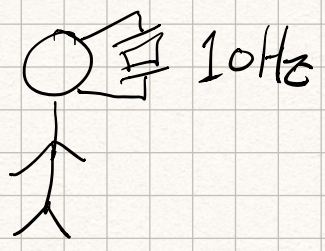
- interchangable.
- use "micro" or MCU

Reasons for using MCU

1. Autonomous decision making
   Sensor → MCU → decision
2. Fast & Precise timing



Typically 10 - 200 MHz



For Lab #2 our CPU 24 MHz max.

Timing resolution 41.7 ns

MCU

- CPU
  - Calculating machine
  - Switches that control switches
- Memory
  - Data Storage
  - switches that store info
- Peripherals
  - Ports, ADC, Comm.
  - Switches that control Hardware

MCU — Giant machine controlled by millions of switches

Firmware dev. → Complexity management.

Keep it super simple.

Bit-wise operations in C

1 bit = 1 switches
4 bits → 1 hexadecimal number 0-F

0011 == 0x3
1110 == 0xE
1100 == 0xC

Our MCU: 16 bit
- Memory is organized into 16 bit chunks
- Modify 16 bits at a time

P1OUT = 0x3E ⇒ 0011 1110

How to <u>set</u> (turn into a 1) one bit in P1OUT?

P1OUT |= 0x01 (|= ← Pipe, OR equals)

Long version: P1OUT = P1OUT | 0x01

```
  0011 1110
| 0000 0001
-----------
  0011 1111
```

Can also use "|" OR

to set multiple bits

P1OUT |= 0x01 + 0x40

Result: 0111 1111

How to Reset (make 0) one bit:

AND Equals

Complement

```
P1OUT &= ~0x01
```

```
  0011 1110
& 1111 1110
-----------
  0011 1110
```

To toggle one bit

```
P1OUT ^= 0x01
```

XOR Equals

Generic bit names (Defined in the header)

BIT0	0x01	0000 0001
BIT1	0x02	0000 0010
BIT2	0x04	0000 0100
BIT3	0x08	0000 1000
BIT4	0x10	0001 0000
⋮	⋮	⋮
BIT15	0x80	1000 0000

```
P1OUT |= BIT0 + BIT4;
```

Register Names | Bit Names

P1OUT

CSCTL1