MEMORY & I/O SYSTEMS

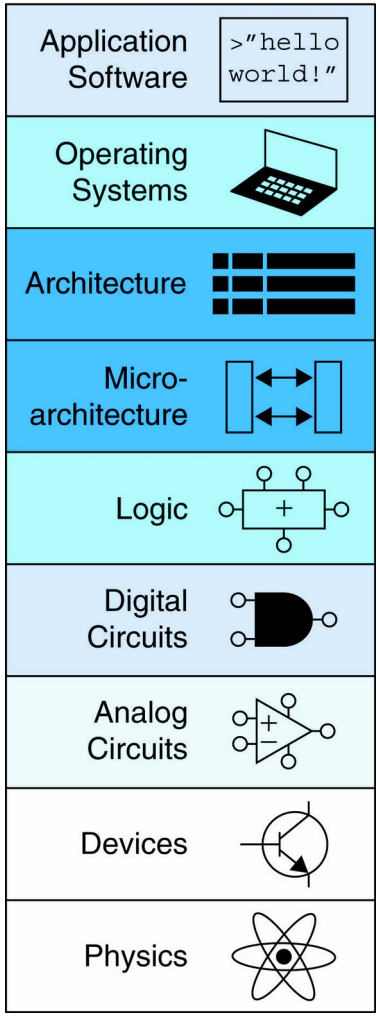
Chapter 8

***Digital Design and Computer Architecture*, 2nd Edition**

David Money Harris and Sarah L. Harris

Chapter 8 :: Topics

- **Introduction (done)**
- **Memory System Performance Analysis (done)**
- **Caches (done)**
- **Virtual Memory (done)**
- **Memory-Mapped I/O (now)**
- **Summary (now)**



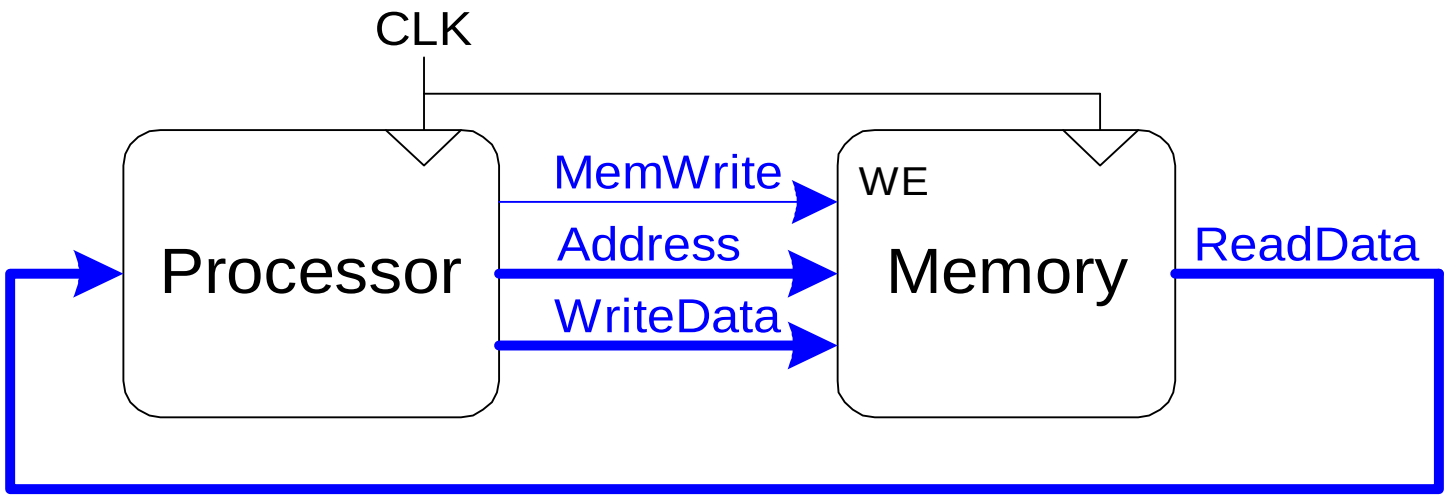
Memory-Mapped I/O

- Processor accesses I/O devices just like memory (like keyboards, monitors, printers)
- Each I/O device assigned one or more address
- When that address is detected, data read/written to I/O device instead of memory
- A portion of the address space dedicated to I/O devices

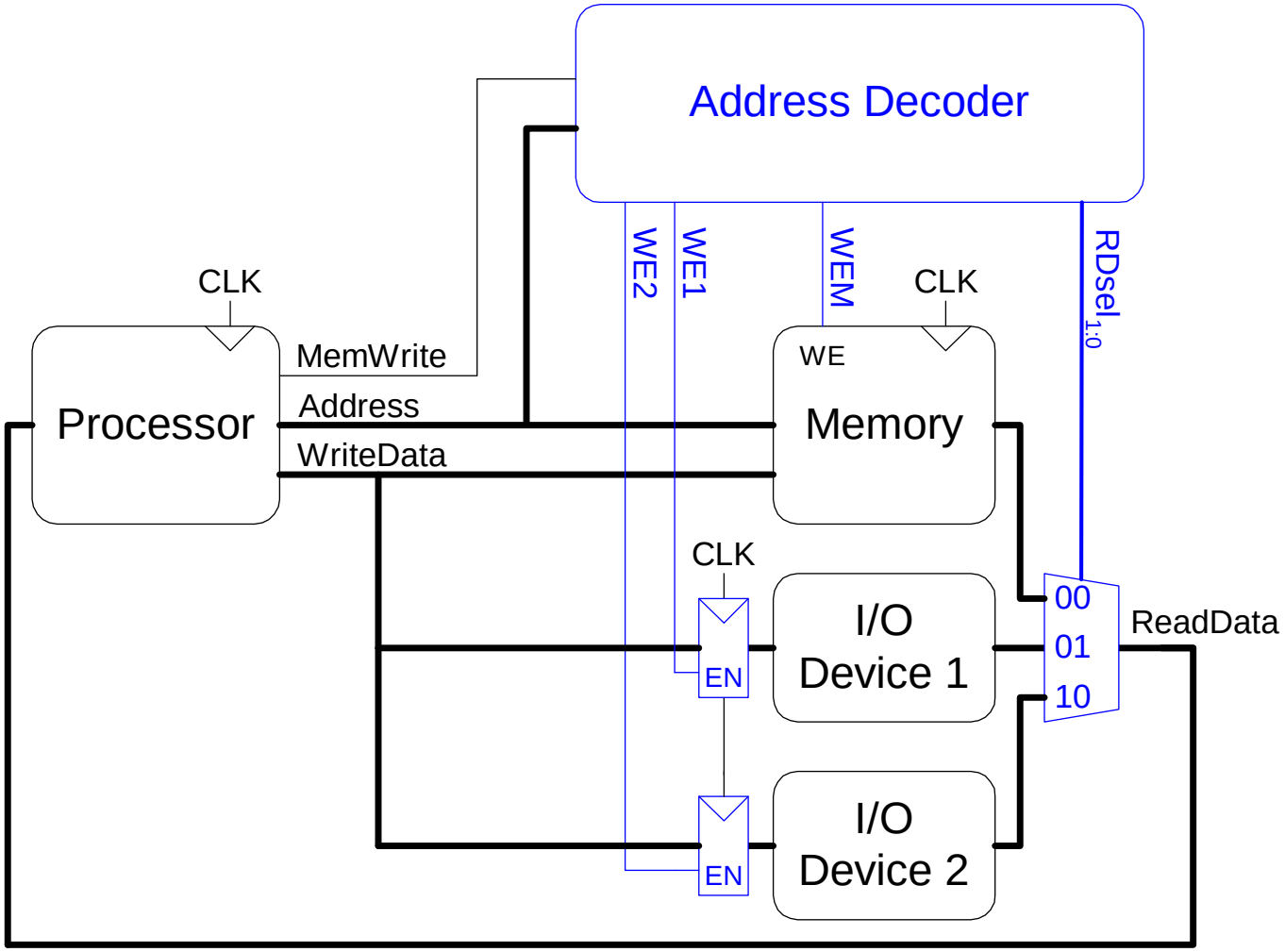
Memory-Mapped I/O Hardware

- **Address Decoder:**
  - Looks at address to determine which device/memory communicates with the processor
- **I/O Registers:**
  - Hold values written to the I/O devices
- **ReadData Multiplexer:**
  - Selects between memory and I/O devices as source of data sent to the processor

The Memory Interface



Memory-Mapped I/O Hardware



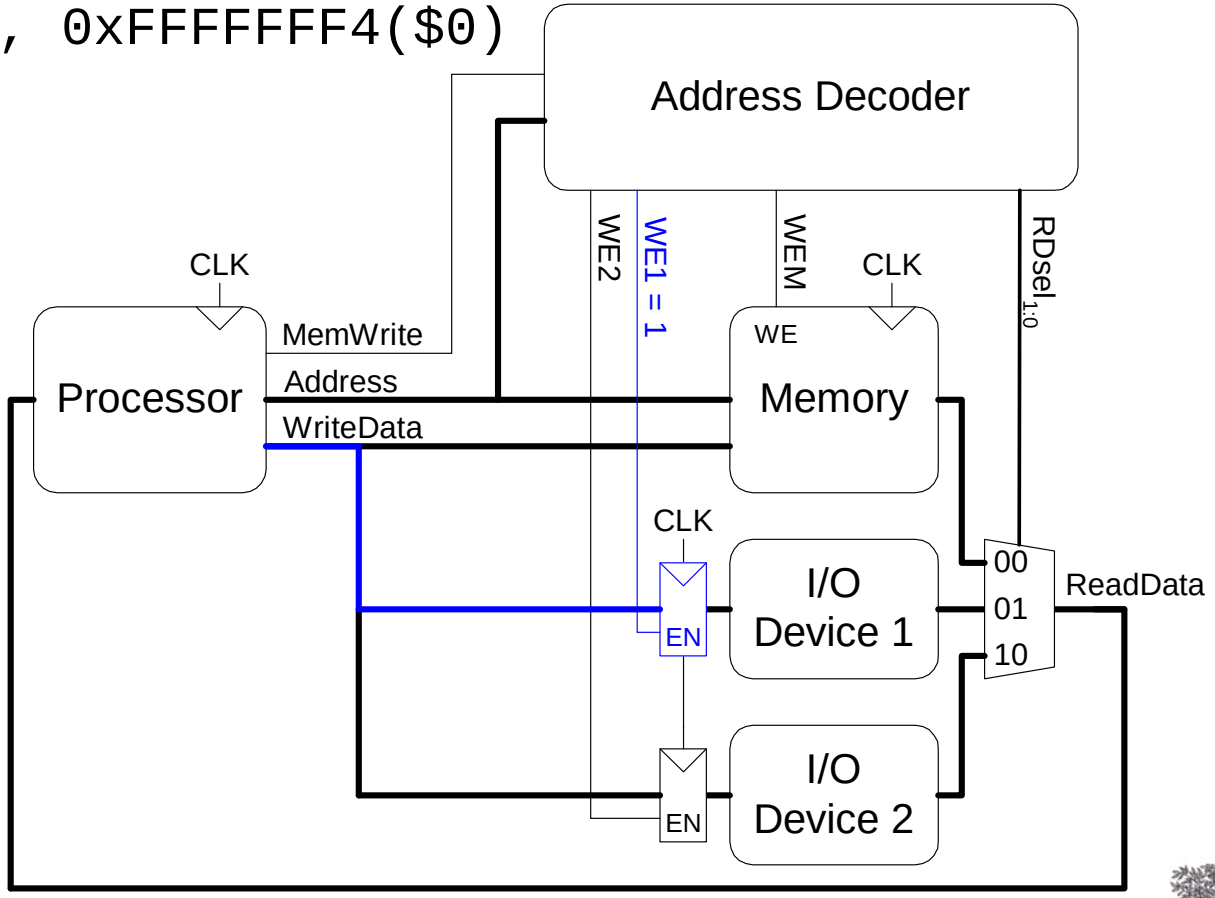
Memory-Mapped I/O Code

- Suppose I/O Device 1 is assigned the address 0xFFFFFFF4
  - **Write the value 42 to I/O Device 1**
  - **Read value from I/O Device 1 and place in `$t3`**

Memory-Mapped I/O Code

- **Write the value 42 to I/O Device 1 (0xFFFFFFF4)**

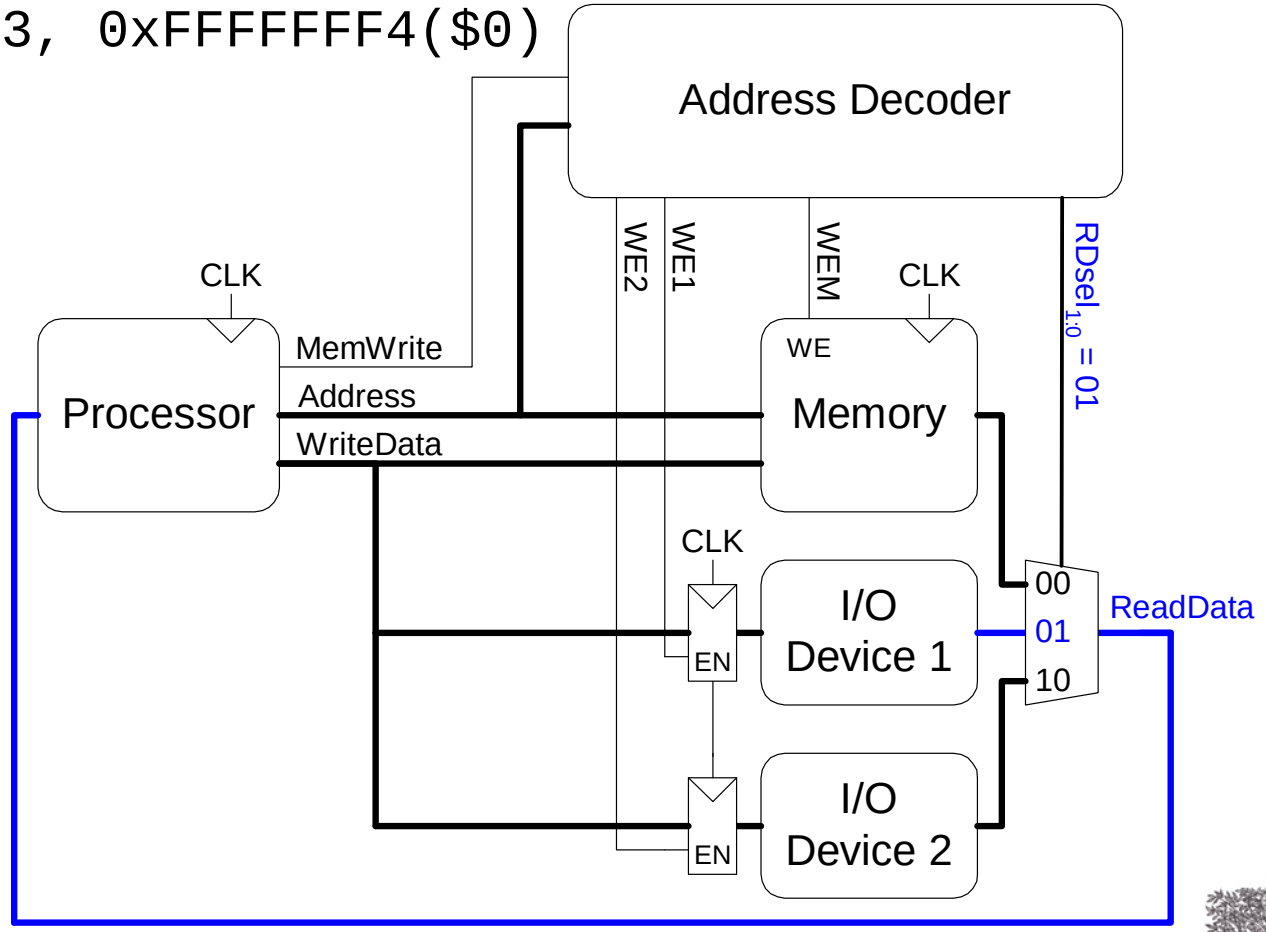
```
addi $t0, $0, 42
sw $t0, 0xFFFFFFF4($0)
```



Memory-Mapped I/O Code

- **Read the value from I/O Device 1 and place in `$t3`**

```
lw $t3, 0xFFFFFFF4($0)
```



Input/Output (I/O) Systems

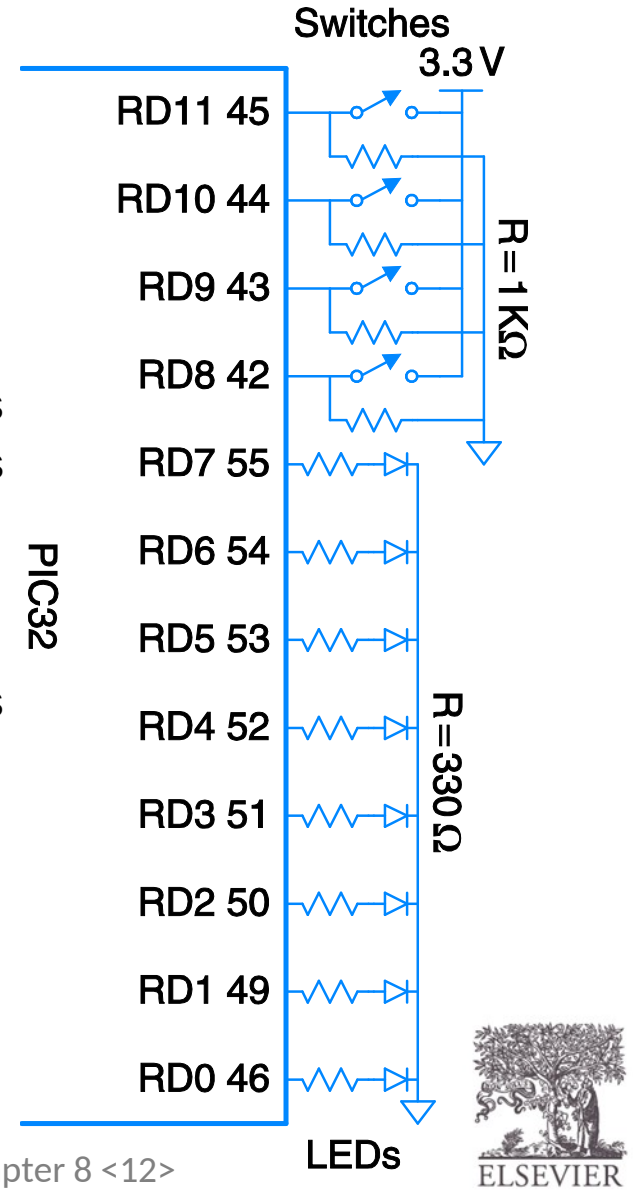
- Embedded I/O Systems
  - Toasters, LEDs, etc.
- PC I/O Systems

Embedded I/O Systems

- Example microcontroller: PIC32
  - microcontroller
  - 32-bit MIPS processor
  - low-level peripherals include:
    - serial ports
    - timers
    - A/D converters

Digital I/O

```
// C Code
#include <p3xxxx.h>

int main(void) {
  int switches;
  TRISD = 0xFF00;       // RD[7:0] outputs
                        // RD[11:8] inputs
  while (1) {
    // read & mask switches, RD[11:8]
    switches = (PORTD >> 8) & 0xF;
    PORTD = switches;  // display on LEDs
  }
}
```

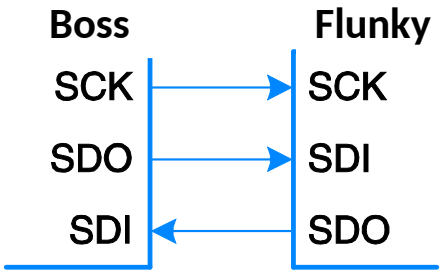


Serial I/O

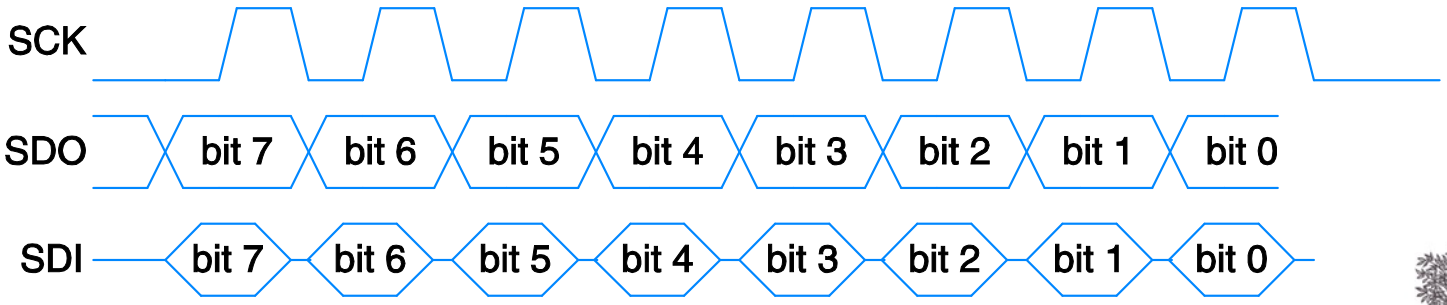
- Example serial protocols
  - **SPI:** Serial Peripheral Interface
  - **UART:** Universal Asynchronous Receiver/Transmitter
  - Also: I^2C, USB, Ethernet, etc.

SPI: Serial Peripheral Interface

- Boss initiates communication to flunky by sending pulses on SCK
- Boss sends SDO (Serial Data Out) to flunky, msb first
- Flunky may send data (SDI) to boss, msb first



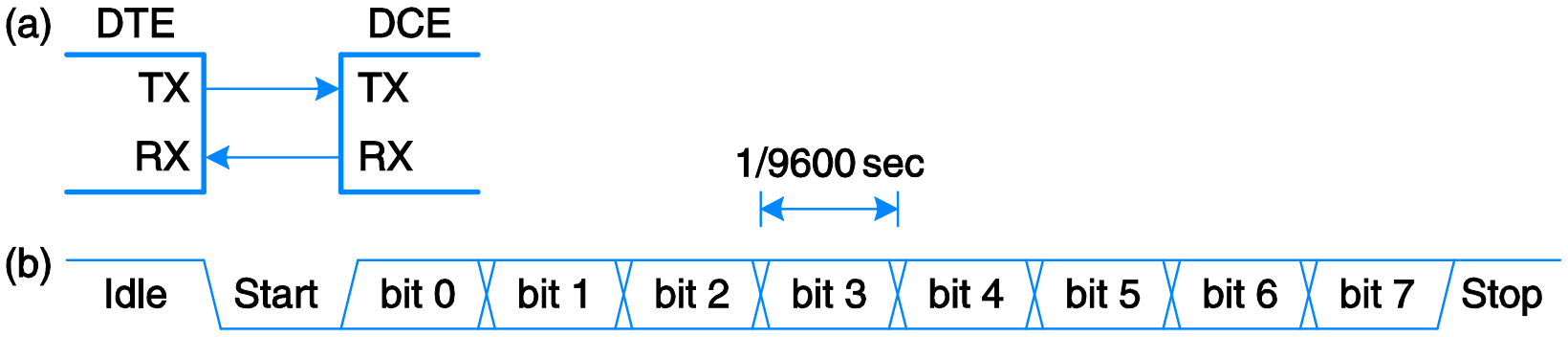
(a)



(b)

UART: Universal Asynchronous Rx/Tx

- Configuration:
  - start bit (0), 7-8 data bits, parity bit (optional), 1+ stop bits (1)
  - data rate: 300, 1200, 2400, 9600, …115200 baud
- Line idles HIGH (1)
- Common configuration:
  - 8 data bits, no parity, 1 stop bit, 9600 baud



Timers

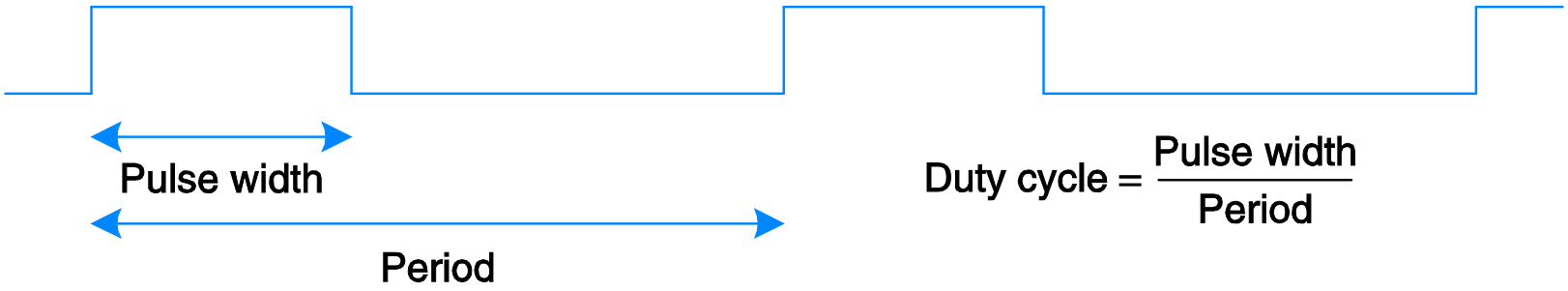
```
// Create specified ms/us of delay using built-in timer
#include <P32xxxx.h>

void delaymicros(int micros) {
  if (micros > 1000) {         // avoid timer overflow
    delaymicros(1000);
    delaymicros(micros-1000);
  }
  else if (micros > 6){
    TMR1 = 0;                  // reset timer to 0
    T1CONbits.ON = 1;          // turn timer on
    PR1 = (micros-6)*20;       // 20 clocks per microsecond
                               // Function has overhead of ~6 us
    IFS0bits.T1IF = 0;         // clear overflow flag
    while (!IFS0bits.T1IF);    // wait until overflow flag set
  }
}

void delaymillis(int millis) {
  while (millis--) delaymicros(1000); // repeatedly delay 1 ms
}                                     // until done
```

Analog I/O

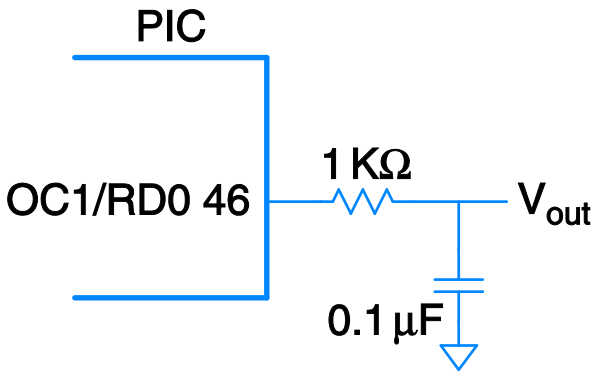
- Needed to interface with outside world
- **Analog input:** Analog-to-digital (A/D) conversion
  - Often included in microcontroller
  - *N*-bit: converts analog input from V_{ref-}-V_{ref+} to 0-2^{N-1}
- **Analog output:**
  - Digital-to-analog (D/A) conversion
    - Typically need external chip (e.g., AD558 or LTC1257)
    - *N*-bit: converts digital signal from 0-2^{N-1} to V_{ref-}-V_{ref+}
  - Pulse-width modulation

Pulse-Width Modulation (PWM)

- Average value proportional to duty cycle



- Add low-pass filter on output to deliver average value



Other Microcontroller Peripherals

- Examples
  - Character LCD
  - VGA monitor
  - Bluetooth wireless
  - Motors

Personal Computer (PC) I/O Systems

- USB: Universal Serial Bus
  - USB 1.0 released in 1996
  - standardized cables/software for peripherals
- PCI/PCIe: Peripheral Component Interconnect/PCI Express
  - developed by Intel, widespread around 1994
  - 32-bit parallel bus
  - used for expansion cards (i.e., sound cards, video cards, etc.)
- DDR: double-data rate memory

Personal Computer (PC) I/O Systems

- TCP/IP: Transmission Control Protocol and Internet Protocol
  - physical connection: Ethernet cable or Wi-Fi
- SATA: hard drive interface
- Input/Output (sensors, actuators, microcontrollers, etc.)
  - Data Acquisition Systems (DAQs)
  - USB Links