

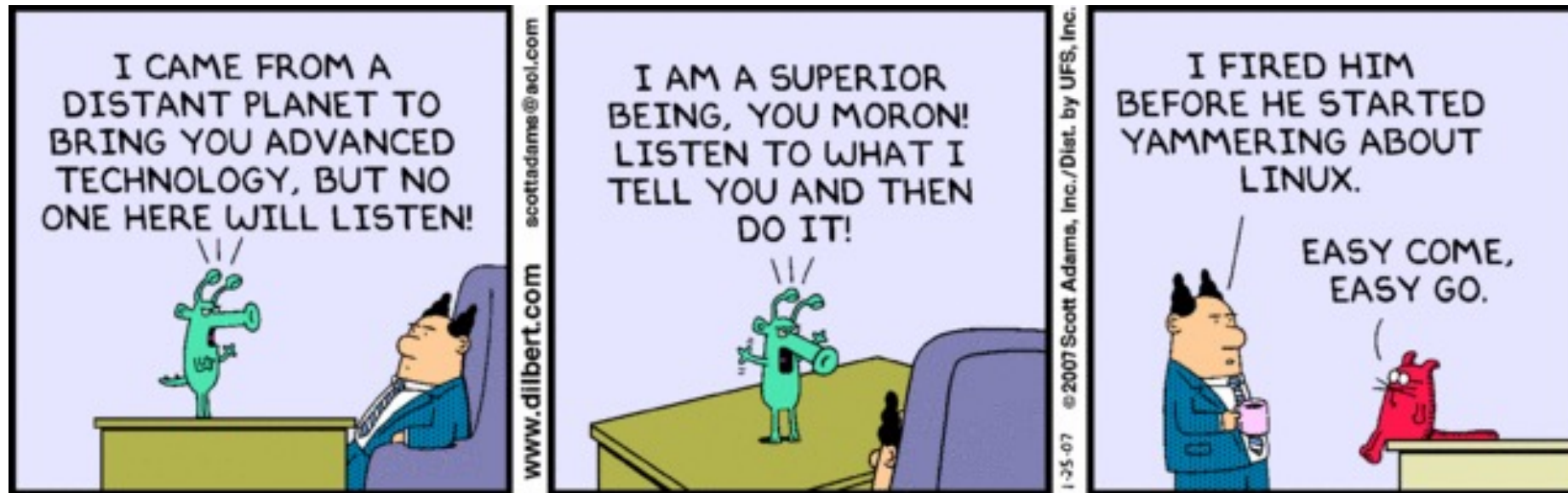
# Calculatoare Numerice (2)

– Cursul 6 –

Dispozitive periferice

Facultatea de Automatică și Calculatoare  
Universitatea Politehnica București

# Comic of the Day



<http://dilbert.com/strips/comic/2007-01-25/>

# I/O Devices

---

- Ce este un dispozitiv?
- Registre
  - Exemplu: NS16550 UART
- Întreruperi
- Direct Memory Access (DMA)
- PCI (Peripheral Component Interconnect)
- Rezumat



# Ce este un device?

---

Concret, pentru un programator de SO:

- Un hardware care este vizibil din software
- Ocupă un spațiu de adresă pe un **bus**
- La adrese sunt mapate **registre**
  - Spațiul I/O mapat ca o memorie
- Generează **întreruperi**
- Poate să inițializeze transferuri **Direct Memory Access (DMA)**



# Registre

---

- CPU poate să încarce dintr-un registru al unui device I/O:
  - Obține informații legate de status
  - Citește date de intrare
  
- CPU poate să încarce într-un registru al unui device I/O:
  - Setează starea dispozitivului și configurația
  - Scrie datele de ieșire
  - Resetează starea dispozitivului



# I/O Mapat în memorie

---

- Procesorul accesează dispozitivele I/O în aceeași manieră ca memoria (tastaturi, monitoare, imprimante)
- Fiecare dispozitiv I/O are atribuite una sau mai multe adrese
- Atunci când adresa este pusă de procesor pe magistrală, datele sunt citite/scrise din I/O în loc din memorie
- Un segment al spațiului de adresă este dedicat dispozitivelor I/O



# Hardware pentru maparea I/O

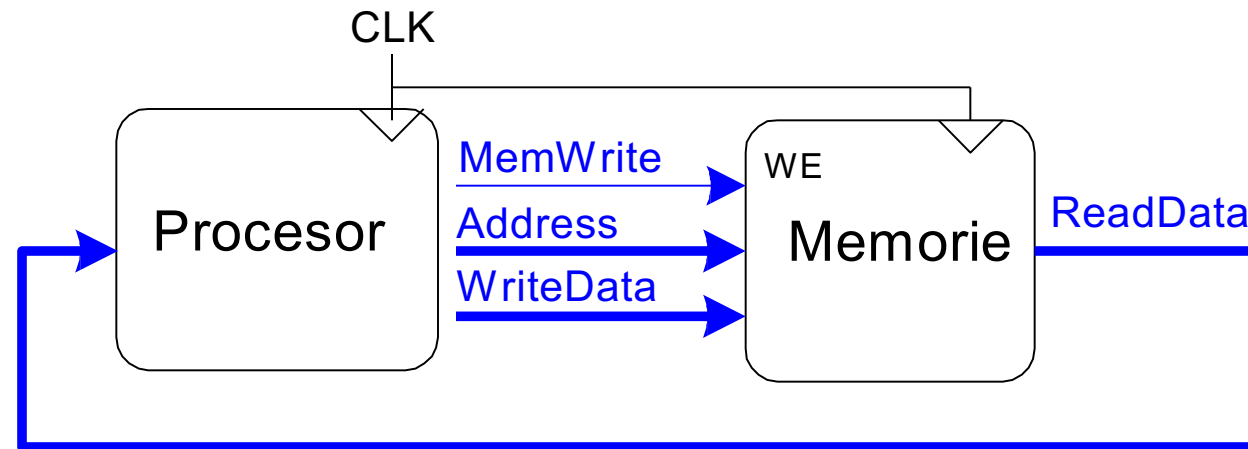
---

- **Decodificator de adrese:**
  - Inspectează adresele pentru a determina care dispozitiv/memorie comunică cu procesorul
- **Registre I/O:**
  - Memorează valorile scrise către dispozitivele I/O
- **Multiplexor pentru datele citite:**
  - Selectează care este sursa de date (memorie sau dispozitive I/O) care trebuie să ajungă la procesor



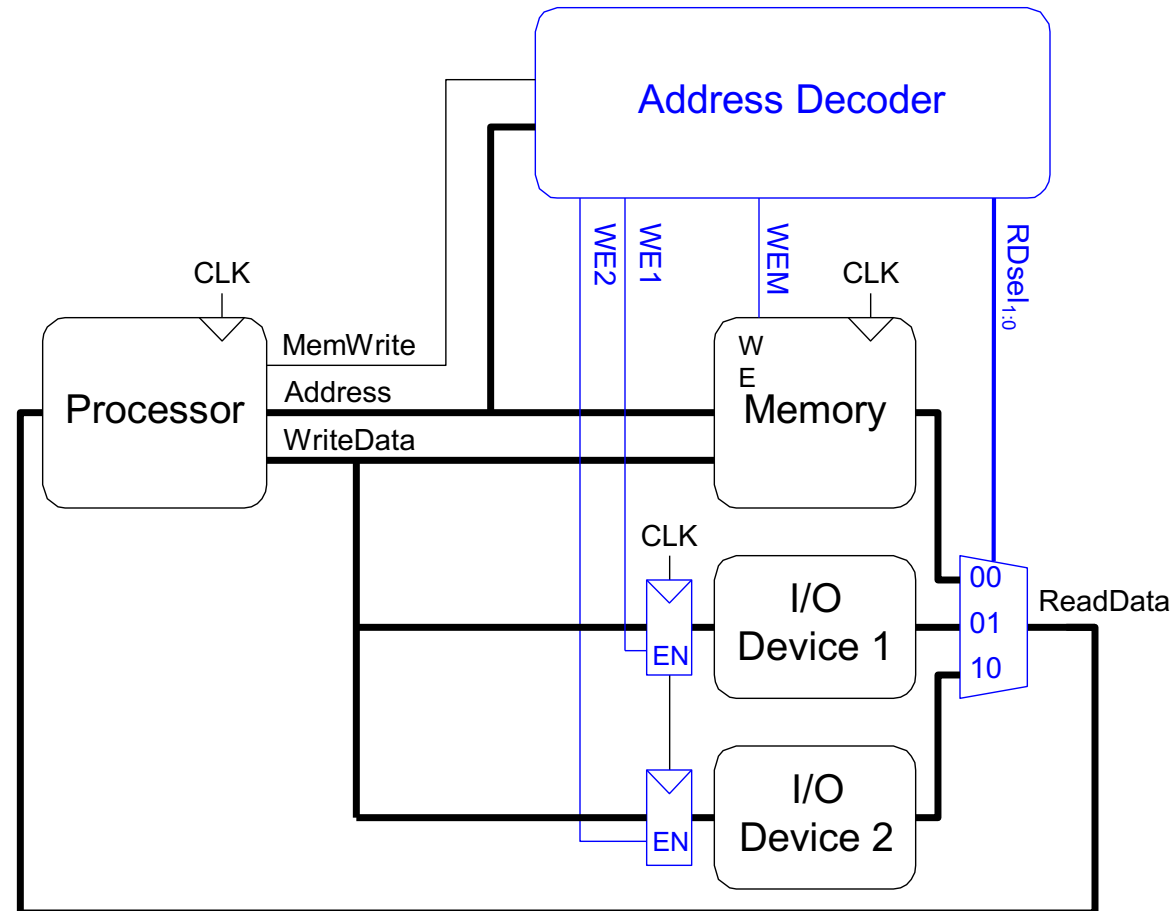
# Interfața cu memoria

---





# Hardware pentru maparea I/O



# Cod pentru maparea I/O

---

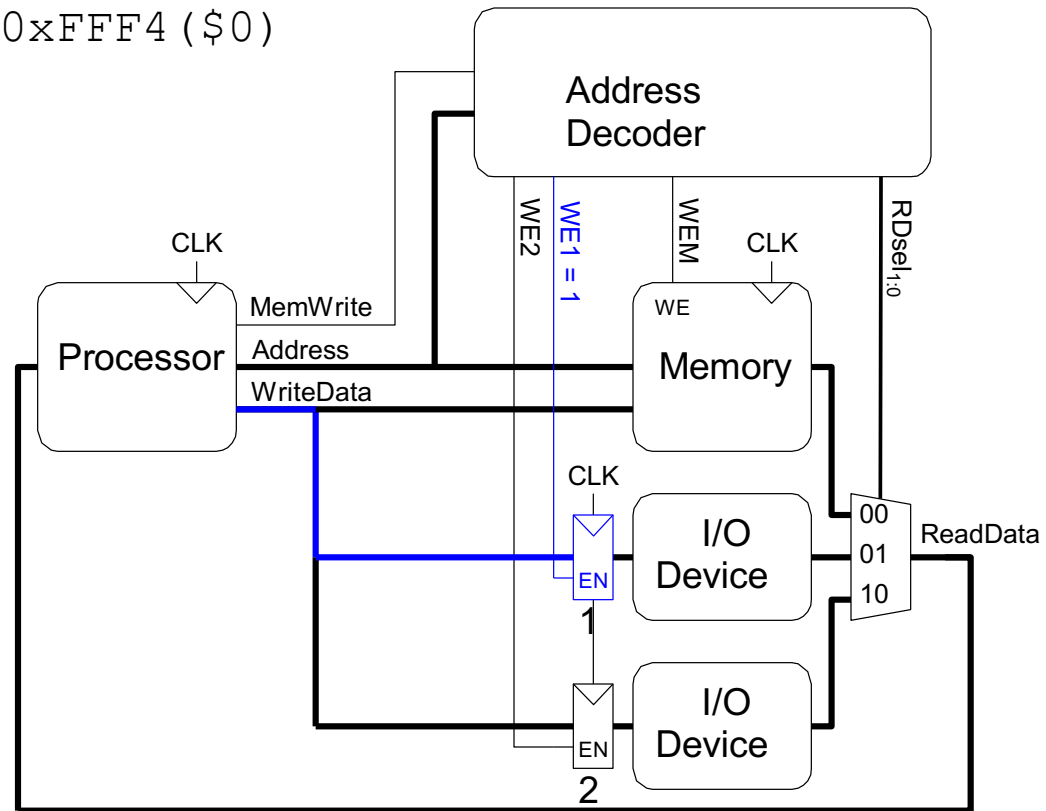
- Presupunem că Dispozitivul I/O 1 are adresa 0xFFFFFFFF4
  - Scrie valoarea 42 la I/O Device 1
  - Citește date din I/O Device 1 și plasează-le în \$t3



# Cod pentru maparea I/O

- **Scrive valoarea 42 în I/O Device 1 (0xFFFFF4)**

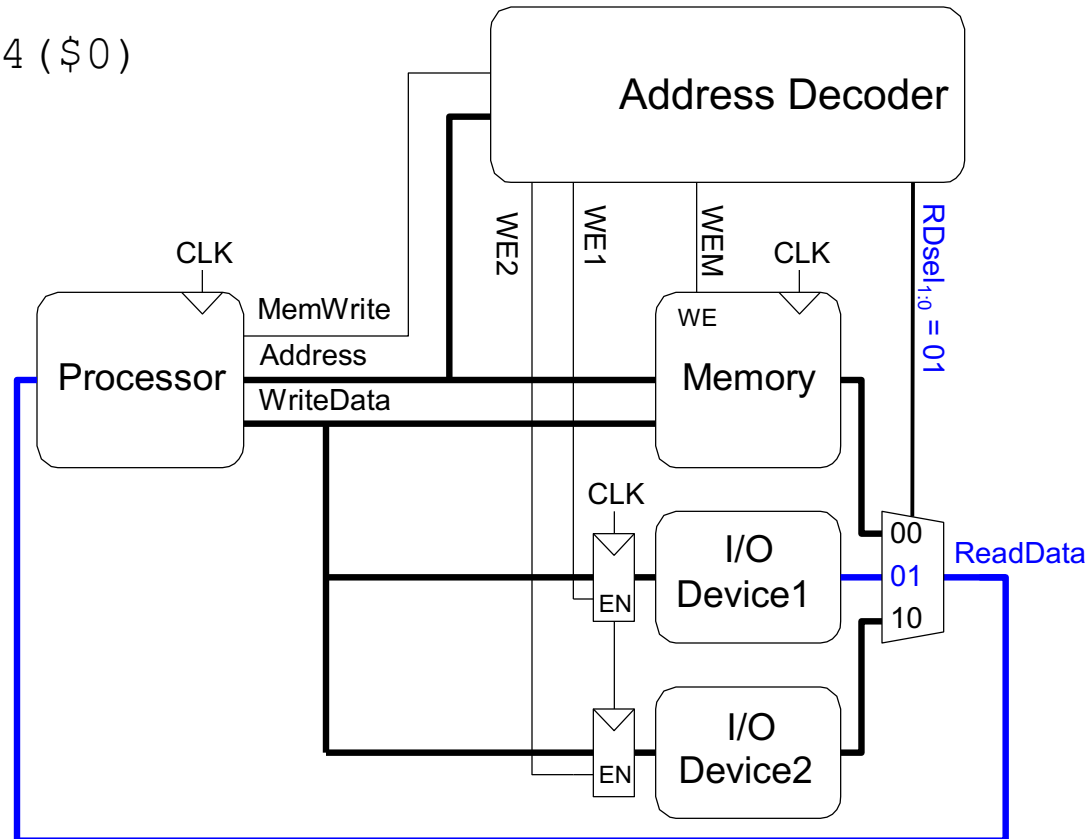
```
addi $t0, $0, 42  
sw $t0, 0xFFF4($0)
```



# Cod pentru maparea I/O

- Citește valoarea de la I/O Device 1 și pune-o în \$t3

```
lw $t3, 0xFFF4($0)
```



# Sisteme Input/Output (I/O)

---

- Sisteme I/O Embedded
  - Cuptoare cu microunde, mașini de spălat, routere, nave spațiale, etc.
- PC I/O Systems



# Sisteme I/O embedded

---

- Exemplu microcontroller: AVR
  - microcontroller
  - Procesor pe 8 biți, arhitectură RISC
  - Periferice low-level:
    - Porturi de I/O
    - Interfețe seriale (RS232, I2C, SPI)
    - Timere
    - Convertoare analog-digitale

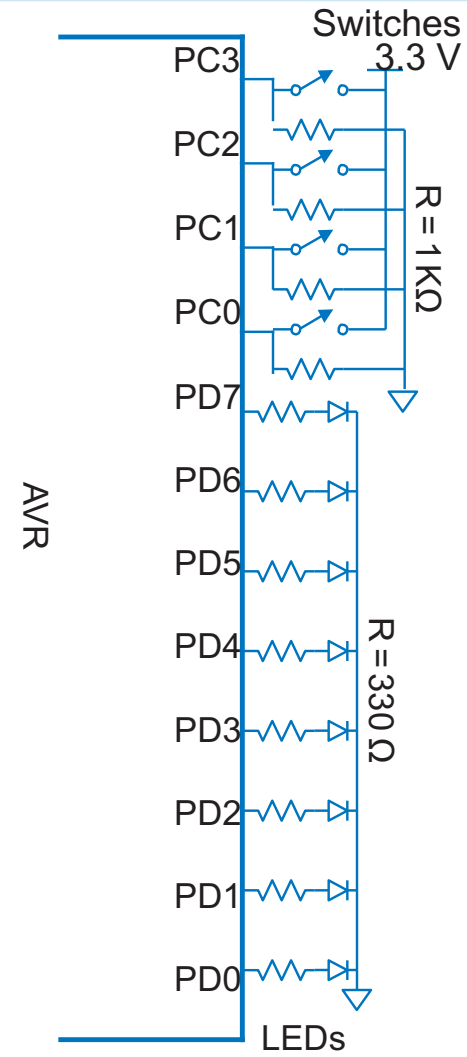


# I/O Digital

```
// C Code
#include <avr/io.h>

int main(void) {
    unsigned char switches;
    DDRD = 0xFF;          //PORTD output
    DDRC = 0x00;         //PORTC input

    while (1) {
        switches = PINC & 0x0F; //read & mask switches
        PORTD = switches;       //display on LEDs
    }
}
```



# I/O Serial

---

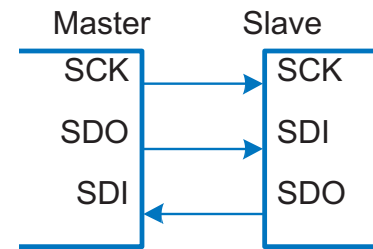
- Exemple protocoale seriale
  - **SPI:** Serial Peripheral Interface
  - **UART:** Universal Asynchronous Receiver/Transmitter
  - De asemenea: I<sup>2</sup>C, USB, Ethernet, etc.



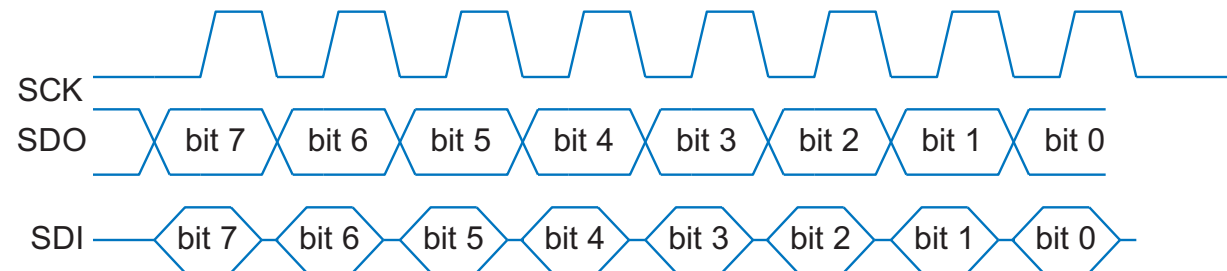


# SPI: Serial Peripheral Interface

- Master-ul inițializează comunicația cu slave-ul prin generarea de impulsuri de ceas pe linia SCK
- Master-ul trimite SDO (Serial Data Out) către slavee, msb first
- Slave-ul trimite datele (SDI) la master, msb first



(a)



(b)

# UART: Universal Asynchronous Rx/Tx

- Configurare:
  - Bit de start (0), 7-8 biți date, bit paritate (opțional), 1+ biți stop (1)
  - data rate: 300, 1200, 2400, 9600, ...115200 baud
- Linia rămâne pe idle în starea HIGH (1)
- Configurație tipică:
  - 8 biți date, fără paritate, 1 bit stop, 9600 baud

(a) DTE DCE



# Timere

---

```
void initTimer1(void)
{
    TCCR1A = 0x00;    //prescale the timer to be clock source/1024
    TCCR1B = _BV(WGM12)|_BV(CS12)|_BV(CS10);
    OCR1A = 10000;    // match 1Hz
    //set 8-bit Timer/Counter1 Output Compare Interrupt Enable
    TIMSK |= _BV(OCIE1A);
}

ISR(TIMER1_COMPA_vect)
{
    PORTB ^=0x01; //blink an LED once a second
}

int main(void)
{
    DDRB = 0xFF;
    initTimer1();
    sei();

    while(1)
    {}
    return 0;
}
```

---



# I/O Analogic

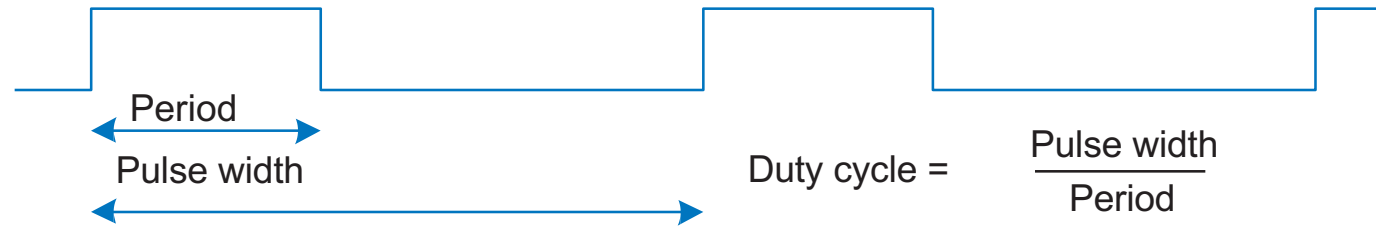
---

- Necesară pentru a interfața procesorul cu mediul în care funcționează
- **Analog input:** Conversie Analog-to-Digital (A/D)
  - De cele mai multe ori este inclus în microcontroller
  - $N$ -biți: convertește o tensiune analogică din gama  $V_{ref-}$ - $V_{ref+}$  într-un întreg de la  $0$ - $2^{N-1}$
- **Analog output:**
  - Conversie Digital-to-Analog (D/A)
    - De obicei este nevoie de un circuit exterior (e.g., AD558 or LTC1257)
    - $N$ -biți: convertește semnalul digital de la  $0$ - $2^{N-1}$  înapoi la  $V_{ref-}$ - $V_{ref+}$
  - Pulse-width modulation

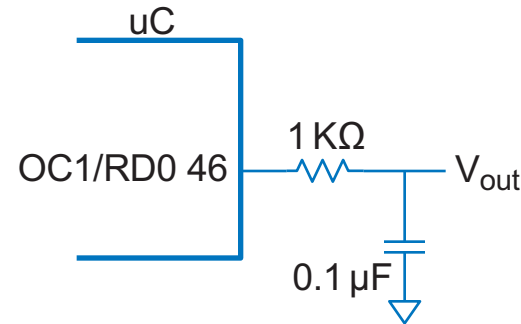


# Pulse-Width Modulation (PWM)

- Valoarea medie este proporțională cu factorul de umplere



- Adăugăm un filtru trece-sus pentru a transforma trenul de impulsuri într-o tensiune analogică de valoare = valoarea medie a semnalului PWM



# Alte periferice întâlnite pe un uController

---

- Exemple
  - LCD cu caractere
  - MonitorVGA
  - Transceiver Bluetooth, WiFi
  - Motoare



# Sisteme I/O pentru Personal Computers (PC)

---

- USB: Universal Serial Bus
  - USB 1.0 apărut în 1996
  - Cabluri și conectori standard/software pentru periferice
- PCI/PCIe: Peripheral Component Interconnect/PCI Express
  - Dezvoltat de Intel, apare în 1994
  - Magistrală paralelă pe 32-biți
  - Folosit pentru plăci de expansiune (de ex. Placă de sunet, video, ethernet etc.)
- DDR: double-data rate memory



# Sisteme I/O pentru Personal Computers (PC)

---

- TCP/IP: Transmission Control Protocol and Internet Protocol
  - Conexiune fizică: cablu Ethernet sau Wi-Fi
- SATA: interfață pentru hard-drive
- Input/Output (senzori, actuatore, microcontrollere etc.)
  - Data Acquisition Systems (DAQs)
  - USB Links





# Acknowledgements

---

- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
- MIT material derived from course 6.823
- UCB material derived from course CS252

