Computer peripherals

- What’s a “peripheral”? 
- Parallel Input/Output 
- Serial communication 
- A/D and D/A conversion 
- Pulse width modulation 
- Interrupts

What’s a "peripheral"?

Any device connected to the bus system through a dedicated interface.

E.g.
- Parallel I/O
- Serial I/O
- Disc drives
- Sensors
- Actuators
- ...

Diagram showing a bus system with CPU, memory, and multiple IO interfaces connected to it.
Parallel Output

Typical application
"ON/OFF"

LED (Light Emitting Diode)
ON

Control bus Data buss
"1"

IO Interface

Control
CS
Data Register
OE

Control bus Data buss
"0"

IO Interface

Control
CS
Data Register
OE

LED (Light Emitting Diode)
OFF
Parallel Input

Typical application
Sense “ON/OFF” switches

Control bus  Data buss

"0"

IO Interface

Switch
OFF

5V

Parallel Input

Typical application
Sense “ON/OFF” switches

Control bus  Data buss

"1"

IO Interface

Switch
ON

5V
Serial communication

IO-interface: Serial Communication Interface (SCI)

Central Processing Unit

Serial to parallel conversion

Receive Data: RxD

Parallel to serial conversion

Transmit Data: TxD

Serial communication

Parallel/serial conversions, asynchronous

**Sender**

Data buss

Control

CS

OE

Data Register

Shift Register

Serial data out

TxD

**Receiver**

Data buss

Control

OE

Data Register

Shift Register

Serial data in

RxD

Requires *Sender* and *Receiver* Clocks to be the same...
Parallel/serial conversions, synchronous

**Master**

- Data buss
- Control
- OE
- Shift Register
- TxD
- Serial data out
- Clock Out

**Slave**

- Data buss
- Control
- OE
- Shift Register
- RxD
- Serial data in
- Clock In

Requires Extra Wire and a Clock Master...

---

**Example: RS232**

Common asynchronous serial communication protocol, for wire lengths up to 25 meters (approx).
RS232 – transmission of ’z’

Character “z” is represented by the bit pattern “0111 1010” (ASCII-character).

+ 10 volt
0 volt
- 10 volt

 idle 0 0 1 0 1 1 1 1 0 0 1 idle

’z’ – least significant bit first

Overhead, 3 bits plus the required “idle time”

D/A conversion

Load, e.g. loudspeaker, light bulb, heater, etc...
Digital/Analog Converter (DAC)

\[ U = -E \frac{r}{R} \left( \frac{1}{S_3/8 + S_2/4 + S_1/2 + S_0} \right) \]

<table>
<thead>
<tr>
<th>Switches</th>
<th>Currents</th>
<th>( I_{\text{tot}} ) (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>( I_0 )</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>( I_1 )</td>
<td>0.2</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>( I_3 + I_0 )</td>
<td>0.4</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>( I_3 + I_0 )</td>
<td>0.6</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>( I_3 )</td>
<td>0.8</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>( I_3 + I_0 )</td>
<td>1.0</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>( I_3 + I_0 )</td>
<td>1.2</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>( I_3 + I_0 )</td>
<td>1.4</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>( I_3 )</td>
<td>1.6</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>( I_3 + I_0 )</td>
<td>1.8</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>( I_3 + I_0 )</td>
<td>2.0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>( I_3 + I_0 )</td>
<td>2.2</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>( I_3 + I_0 )</td>
<td>2.4</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>( I_3 + I_0 )</td>
<td>2.6</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>( I_3 + I_0 )</td>
<td>3.0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>( I_3 + I_0 )</td>
<td>3.0</td>
</tr>
</tbody>
</table>
A/D conversion

Analog/Digital Converter (ADC): Ramp

\[ U = \frac{t_2}{t_1} E \]
Analog/Digital Converter (ADC): Flash

Pulse Width Modulation (PWM)

\[ U_{out} = \frac{\text{duty cycle}}{\text{period}} U \]

Period and duty cycle are programmable...
**PWM as replacement for DAC**

![PWM diagram]

Load, e.g. loudspeaker, light bulb, heater, etc...

---

**Interrupts**

A peripheral can interrupt the CPU, executing a program by means of a hardware signal, called *Interrupt Request*.

The CPU then acknowledge this request, freeze the current program and executes a dedicated *Interrupt Handler*.

When the Interrupt Handler is finished the CPU resumes execution of the interrupted program.
Interrupts, hardware

Interrupts, software

Signal: **IRQ**: (interrupt request)

hardware signal **IRQ**

special instruction: ReTurn from Interrupt

"service interrupt"
Interrupts, multiple interrupt sources

Conclusion

we have got a brief introduction to

- What's a “peripheral”?
- Parallel Input/Output
- Serial communication
- A/D and D/A conversion
- Pulse width modulation
- Interrupts

which finishes today's lecture ...