Embedded systems

- An embedded system is a special-purpose system designed to perform one (or a few) dedicated functions.

- Some typical characteristics of embedded systems are:
  - Single purpose (with very specific requirements).
  - Not easily adapted.
  - Real-time computing constraints.
  - No operating system or small and simple operating systems.
  - High reliability.
  - Limited computer hardware resource, for instance fixed amount of memory and limited I/O expansion possibilities.
  - Small or non-existent keyboard/mouse or screen.
  - Low power (e.g. 50 mW vs. 50 W or more for a PC).
  - More difficult to program and to interface with compared to a general purpose computer.
What is a real-time (RT) system

• A real-time system gives you **determinism**
  – The correctness of the system depends not only on the logical result but also on the time it was delivered

• **Hard real-time**
  – systems where it is absolutely imperative that responses occur within the required deadline (Example: Flight control systems)

• **Soft real-time**
  – allows for some deadlines to be missed with only a slight degradation in performance but not a complete failure (example: DAQ-systems)

• In contrast, on an ordinary desktop PC (with Windows) the OS operates on a fairness basis
  – Each application gets time on the CPU regardless of its priority
  – Even our most time-critical application can be suspended for some routine maintenance
Hard vs. Soft Real-Time Applications

Hard Real-Time
- Average Time
- Worst-Case Time

Soft Real-Time
- Average Time
- Worst-Case Time
Embedded systems

Aerospace & Defence

1968

Apollo Guidance Computer

Today – consumer products

ABS Entertainment system
Autonomous Systems

• Embedded systems are everywhere today and will be even more important in the future.

• Most embedded systems perform “simple functions” that are pre-programmed.

• Increasingly, embedded systems are designed to carry out autonomous tasks (smart systems).

• Embedded systems/vehicles will be designed to be able to make decisions based on complex inputs and situation awareness.
Embedded processors

- Microprocessor
- Microcontroller
- DSP (Digital signal processor)
  - A specialized microprocessor with an optimized architecture for mathematical operations to be performed quickly (e.g. FFT)
- FPGA (Field Programmable Gate Array)
- GPU (Graphics Processing Unit)
Number representation

• **Fixed-point number system**
  – For example, the value 1.23 can be represented as 1230 (an integer), with scaling factor of 1/1000.
  – Fixed binary point.
  – To represent larger numbers or to achieve an accurate result a larger number of bits are needed.

• **Floating-point number system**
  – Floating-point representations are easier to use than fixed-point representations, because they can handle a wider dynamic range.
  – The logic needed to implement a given arithmetic operation is considerably more complex and area demanding compared to fixed-point numbers.
  – Limited use in FPGA logics, but can be used in DSP blocks and microprocessor core(s) inside the FPGA.
Embedded microprocessors

• Modern x86 CPUs are relatively uncommon in embedded systems and small low power applications, as well as low-cost microprocessor markets (e.g. home appliances and toys).

• Simple 8-bit and 16-bit based architectures are common, although the x86-compatible AMD's Athlon and Intel Atom are examples of 64-bit designs used in some relatively low power and low cost segments.
Microcontrollers

- The program instructions written for **microcontrollers** are referred to as **firmware**, and are stored in read-only memory (ROM) or Flash memory chips.
- In contrast, a general-purpose computer loads its programs into random access memory (RAM) each time.
FPGA Technology

FPGA = Field Programmable Gate Array
Xilinx 7 series FPGA example

SOC : System On a Chip
FPGA advantages

- High reliability
- High determinism
- High performance
- True parallelism
- Reconfigurable

The highest performance FPGAs (2012) have 600 MHz clock speed

FPGAs give low-latency processing, but they have limitations in terms of floating-point computations
Processor / FPGA Co-Processor Features

- Operating System
- Legacy code base
- Floating point math
- High-level abstraction (ex. C / C++)
- Complex decision making
- Non time-critical
- Interrupt-driven
- Fixed peripheral set

- Parallel execution
- High computation rates in fixed-point math
- Repetitive calculations
- Nested inner loops
- Fast access to deeply pipelined time-skew buffers
- Wide data words
- Custom peripherals
How to program an FPGA?

- **VHDL** (Hardware Description Language)
- **C-code** (need a development tool)
- Automatic Generation of VHDL code (or a bit stream) from a high level development tool, such as
  - **MATLAB** (HDL Coder)
  - **Simulink** (DSP Builder / System Generator for DSP)
  - **LabVIEW** (FPGA Module)
From LabVIEW to Hardware

Can also use the LabVIEW IP Integration Node to include VHDL code
Architecture for Advanced Embedded Applications

Avoid:
- File I/O
- Networking functions
- Memory re-allocation
GPUs

- GPU = Graphics Processing Unit
- Can be used as hardware accelerator
- Can be used in **Real-Time High-Performance Computing** systems
- GPUs have more transistors dedicated for processing than a CPU
  - The performance gain when using GPUs can be significant
- CUDA (Compute Unified Device Architecture) is developed by Nvidia and is a GPU interface for C
CPU VERSUS GPU

A simple way to understand the difference between a CPU and GPU is to compare how they process tasks. A CPU consists of a few cores optimized for sequential serial processing while a GPU has a massively parallel architecture consisting of thousands of smaller, more efficient cores designed for handling multiple tasks simultaneously.

GPUs have thousands of cores to process parallel workloads efficiently
# NVIDIA Tesla GPUs

## SELECT THE RIGHT TESLA GPU

<table>
<thead>
<tr>
<th>Features</th>
<th>Tesla K20X</th>
<th>Tesla K20</th>
<th>Tesla K10</th>
<th>Tesla M2090</th>
<th>Tesla M2075</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number and Type of GPU</strong></td>
<td>1 Kepler GK110</td>
<td>2 Kepler GK104s</td>
<td>1 Fermi GPU</td>
<td>1 Fermi GPU</td>
<td></td>
</tr>
<tr>
<td><strong>GPU Computing Applications</strong></td>
<td>Seismic processing, CFD, CAE, Financial computing, Computational chemistry and Physics, Data analytics, Satellite imaging, Weather modeling</td>
<td>Seismic processing, signal and image processing, video analytics</td>
<td>Seismic processing, CFD, CAE, Financial computing, Computational chemistry and Physics, Data analytics, Satellite imaging, Weather modeling</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Peak double precision floating point performance</strong></td>
<td>1.31 Tflops</td>
<td>1.17 Tflops</td>
<td>190 Gigaflops (95 Gflops per GPU)</td>
<td>665 Gigaflops</td>
<td>515 Gigaflops</td>
</tr>
<tr>
<td><strong>Peak single precision floating point performance</strong></td>
<td>3.95 Tflops</td>
<td>3.52 Tflops</td>
<td>4577 Gigaflops (2288 Gflops per GPU)</td>
<td>1331 Gigaflops</td>
<td>1030 Gigaflops</td>
</tr>
<tr>
<td><strong>Memory bandwidth (ECC off)</strong></td>
<td>250 GB/sec</td>
<td>208 GB/sec</td>
<td>320 GB/sec (160 GB/sec per GPU)</td>
<td>177 GB/sec</td>
<td>150 GB/sec</td>
</tr>
<tr>
<td><strong>Memory size (GDDR5)</strong></td>
<td>6 GB</td>
<td>5 GB</td>
<td>8 GB (4 GB per GPU)</td>
<td>6 GigaBytes</td>
<td>6 GigaBytes</td>
</tr>
<tr>
<td><strong>CUDA cores</strong></td>
<td>2688</td>
<td>2496</td>
<td><strong>3072</strong> (1536 per GPU)</td>
<td>512</td>
<td>448</td>
</tr>
</tbody>
</table>
Ethernet for real-time applications

- Remote I/O can demand reaction in the 5-10 ms region. Motion Control demands even higher determinism with cycle times into the microsecond region.
- Standard Ethernet communication utilizes TCP/IP, which is inherently non-deterministic and has a reaction time in the hundreds of milliseconds. In an effort to boost determinism some networks utilize custom technologies in the transport and network layers of the Ethernet stack. These networks merely use TCP/IP as a supplemental channel to provide non real-time data transfers. By bypassing the TCP/IP protocols, such proprietary networks limit the end user’s ability to use standard, off-the-shelf Ethernet products such as routers, switches, firewalls, etc. This limitation destroys one of the fundamental advantages of standard Ethernet - the availability of low-cost, ubiquitous COTS Ethernet hardware.
- By using UDP instead of TCP the reaction time comes down to about 10 ms at best. UDP is not suited for “hard” deterministic distributed systems.
Level shifting of bipolar signals

• Analog input signals can be unipolar or bipolar.
  – Unipolar signals swing between zero and positive full-scale.
  – Bipolar signal swings above and below some reference point, typically ground.

• In battery powered embedded systems **single-supply circuits** are often used to save power. Therefore a level shifting is required to convert a bipolar signal into a unipolar signal.
Level shifter example circuit

- +2.5 V single supply electronics
- +/- 10 V input signal
- NB: Sensor resistance Rs should be low!

Note: use the superposition principle to calculate the voltage at point Vin