FUNCTIONAL SAFETY IN EMBEDDED SYSTEMS

Part III

Embedded Systems Hardware

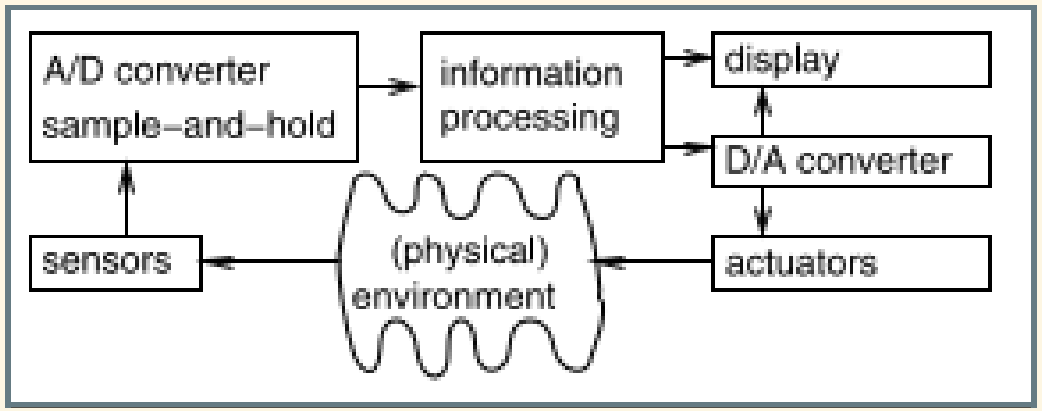
STEPHAN HEIDINGER

OUTLINE

- Introduction
- Input
- Processing Units
- Reconfigurable Logic
- Memory
- Communication
- Output
- Secure Hardware

INTRODUCTION

- System consists of Software **and** Hardware.
- Hardware for Embedded Systems is much less standardized than PC hardware is.



INPUT

- Sensors
- Discretization of time
- Discretization of values

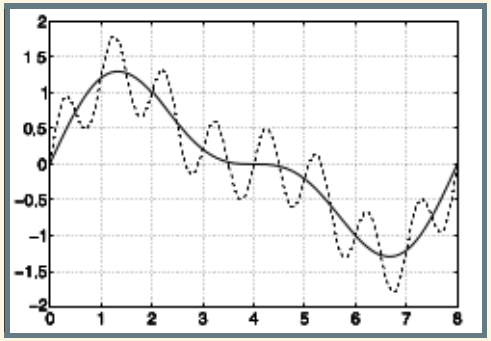
INPUT - SENSORS

Sensors for any given quantity.

- Acceleration
- Rain
- Image - CCD
- Image - CMOS
- Biometric Sensors
- RFID
- …

INPUT - DISCRETIZATION OF TIME

- Computers only work in discrete time domain.
  => Need to take samples at fixed points in time.
  => Only those samples can be used by the computer.
- Reconstruction of unsampled data generaly not possible.
  => Can be possible in some cases.
  => **Nyquist sampling criterion:**
  "Restrict the frequencies of the incoming signal to less than half of the sampling frequency."
- Sample-and-Hold circuits: value is sampled and held, until next value is sampled (sampling takes some time).



INPUT - DISCRETIZATION OF VALUES

- Computers only work in discrete data domain.
- large variety of converters
- *Flash A/D converter*
  + very fast
  - complex hardware (n-1 comparators)
- Successive Approximation:
  + hardware efficient
  - slow (log(n) steps)
- SNR => "how much good signal" (dB, dimensionless)
- Automatic techniques for selecting most approriate converter exist.

PROCESSING UNITS

- Overview
- ASIC
- Processors

PROCESSING UNITS - OVERVIEW

- "consumed" energy
- power and energy efficiency important
  => "*green IT*"
- energy efficient => not flexible
- flexible => not energy efficient
- high power consumption => may be more energy efficient, if also faster

PROCESSING UNITS - ASIC

APPLICATION SPECIFIC INTEGRATED CIRCUITS

- Cost of design and manufacture very high.
  => only feasible for large quantities
- Design is time consuming.
  => Correcting faults is expensive.
- Very small energy consumption.

PROCESSING UNITS - PROCESSORS

- Processors are programmable.
  => very flexible
- Embedded processors
  => no need for instruction set compatibility.

PROCESSING UNITS - PROCESSORS

ENERGY EFFICIENCY

- gate clocking
  => parts of processor are disconnected from clock if idle
- *Dynamic Power Management* (DPM)
  => more power saving states (e.g. run, idle, sleep)
- *Dynamic Voltage Scaling* (DVS)
  => power changes quadratically with voltage
  => runtime changes linearly

PROCESSING UNITS - PROCESSORS

CODE-SIZE EFFICIENCY

- Memory very limited => want to take as little space as possible for code
- RISC => designed for speed
- CISC ("old processors") => designed for code-efficiency
- Compression Techniques => hardware decoder
- second instruction set (ARM)
- dictionaries

PROCESSING UNITS - PROCESSORS

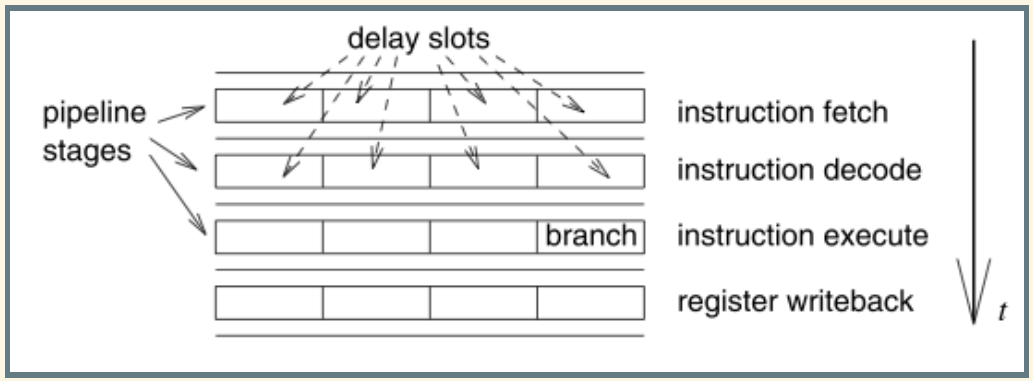
RUN-TIME EFFICIENCY

- Optimize architecture for specific domain.
  => more efficient without higher clock frequency
- Example: Digital Signal Processing
    - Specialized Addressing Modes
    - Seperate Address Generation Units
    - Saturation Arithmetic
    - Fixed-Point Arithmetic
    - Multiple Memory Banks
    - Multiply/Accumulate Instructions

PROCESSING UNITS - PROCESSORS

VERY LONG INSTRUCTION WORD PROCESSORS

- no instruction set compatibility
  => explicit identification of parallelism
  => EPIC (explicitly parallel instruction computing)
- several operations in one instructions
  => execution in parallel
- possible *delay penalty* (due to branch instructions):



PROCESSING UNITS - PROCESSORS

MICRO-CONTROLLERS

- not complex
- easy usage

MULTIPROCESSOR SYSTEM-ON-A-CHIP (MPSOC)

- increased clock speed => higher energy consumption
- in PC
  => homogenous *multi-core* systems
- in embedded systems
  => highly specialised processors
  => energy efficiency close to ASICs

RECONFIGURABLE LOGIC

- ASIC often too expensive, processor too slow
  => field programmable gate Arrays (FPGAs)
- timing of FPGA precisely predictable
- reconfigurable hardware
  => ROM is non-volatile
  => RAM is volatile
- processors may be included in FPGA:
  - *hard core*: physical processor (efficient)
  - *soft core*: synthesize processor from CLB

MEMORY

- efficient storage
  => run-time - memory hierarchy
  => code-size - compiler, compression
  => energy - memory hierarchy
- Moore's Law
  => speed of processors doubles every 18-36 months
  => speed of memory only increases by 1.07
- Use small, fast memory between processor and main memory.
- Scratch Pad Memory (SPM)

COMMUNICATION

- Common Communication Requirements
- Electrical Robustness
- Real-Time Behavior
- Examples

COMMUNICATION - REQUIREMENTS

- real-time behavior
- efficiency
- bandwidth, communication delay
- event-driven communication
- robustness
- fault tolerance
- maintainability
- privacy

COMMUNICATION - EL. ROBUSTNESS

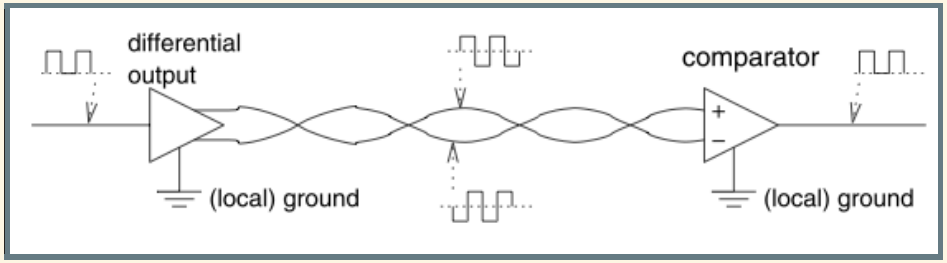
SINGLE-ENDED SIGNALING

- voltage over commong ground
- succeptible to noise
- common ground may be hard to achieve

COMMUNICATION - EL. ROBUSTNESS

DIFFERENTIAL SIGNALING

- first -> second = positive voltage => '1'
- noise added to both wires => no effect
- magnitude of voltage => irrelevant
- no common ground
- large throughput



COMMUNICATION - REAL TIME

- point-to-point
  => many wires needed
  => shared buses very common
- priority based arbitration
  => poor timing, can lead to starvation
- TDMA (timed division multiple access)
  => real time behavior may be to slow
- CSMA/CD: upon collision: stop, wait, restart
- CSMA/CA: priority based arbitration phase
  => starvation possible

COMMUNICATION - EXAMPLES

- Sensor/Actuator Bus
- Field Bus
  => controller area network
  => time-triggered Protocol
- Wired Multimedia Communication
  => MOST (media oriented system transport)
  => IEEE 1394 "FireWire"
- Wireless Communication
  => HSPA, LTE
  => Bluetooth
  => IEEE 802.11

OUTPUT

- Displays
- Electro-Mechanical Devices
  (motors, gauges, …)
- D/A Converter
  => generate current proportional to signal x
  => convert current into voltage y
- If Nyquist Theorem holds => interpolation is possible
  => Shannon-Whittaker interpolation
  => possible via low-pass filter
- Actuators: very wide range
  => relatively new: *Microsystem Technology*

SECURE HARDWARE

Embedded Systems may need special security elements:

- cryptography, en-/decryption
- side-channel attacks

Different levels of security and enemy need to be taken into account.