Embedded and Ambient Systems
2020. 09. 15.

Introduction

Méréstechnika és
Információs Rendszerek
Tanszék

Preliminaries

- **Embedded and Ambient Systems**
  - Subject Code: BMEVIMIAC06 (ENG Course)
  - Lectures and Practice:
    - Lecture: every Tuesday 16:15-17:50 (at 17:00, 5-minutes break)
    - Practice: every odd Wednesday 16:15-17:50 (at 17:00, 5-minutes break)
  - Lecturer: Krébesz, Tamás(BME-MIT)
    - Email: krebesz@mit.bme.hu
    - Szoba: IE.413
  - Requirements:
    - Midterm (Oct. 20. at lecture time)
    - Homework
      - Simple homework in 2-person group
      - To be presented at the end of the term

Embedded systems

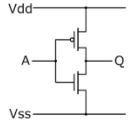
- **Possible definitions**
  - Those computer-based application systems, that are:
    - Autonomous in operation,
    - In strong information-based connection with their physical/technological environment.
  - Such a unit that control or supervise a machine, instrument or industrial process.
  - A computer without a keyboard ☺. I.e. every processor-based or digital unit that is not a PC.
- **The traditional microprocessor-based systems can be considered embedded systems.**

Embedded systems: examples

- **Examples:**
  - Consumer electronics: music player, TV, watch, wireless headphone, camera, display, wireless mouse/keyboard
  - Handheld devices: mobilephone, GPS, calculator
  - Household appliances: washing machine, microwave oven, fridge
  - Home automatization: elevators, alarm system, heating control, remote home surveillance
  - Vehicular electronics: ECU, ABS, assisted steering, mirror remote control, parking radar, on-board computer, gear control, etc.
  - Industrial robots, intelligent power supply, engine control
  - Ticket machine, ATM, electronic information center
  - Medical instrument: blood pressure meter, complex diagnostic devices,
  - Measurement instruments: software defined measurement
  - Info communication: modem, router, switch

Developers of embedded systems

- **Why is it good to learn embedded sysmtes?**
  - Development is done at the edge of the HW-based and SW-based worlds: the SW developed can acquire direct information from the rela physical world and can react into real-world processes.
  - Starting from the circuit design through SW development one can get in touch with PCs and higher level information systems.
  - Continuously developing industrial field, makes a living for lots of people, new professionals are always needed.

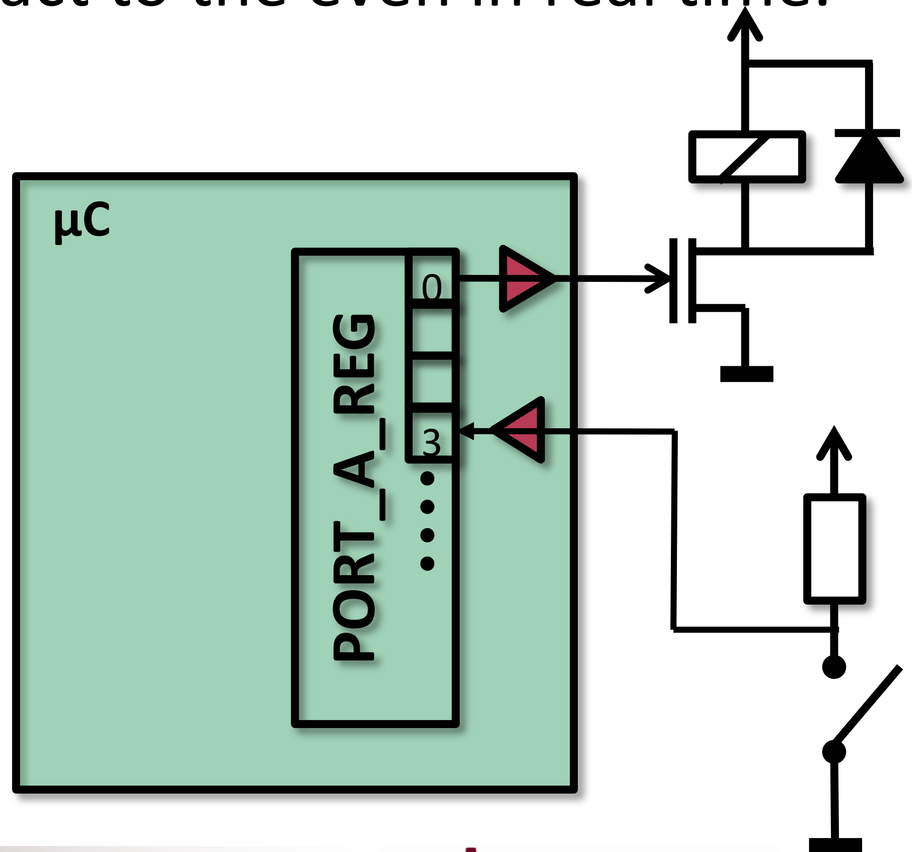


Example: direct connection with environ.

- What does it mean ‘being in direct connection with the physical environment’?
  - The signals of the environment can be sensed at a low (abstraction) level, or react to the even in real time.

Example #1: open door?

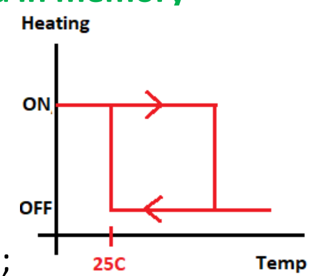
```
#define RELAY_ON  (0)
#define DOOR_BIT  (3)
.
.
.
// PORT_A_REG: I/O register nested in memory
door_is_open = PORT_A_REG & (1 << DOOR_BIT);
if (door_is_open){
    PORT_A_REG |= 1 << RELAY_ON;
}
.
.
.
```

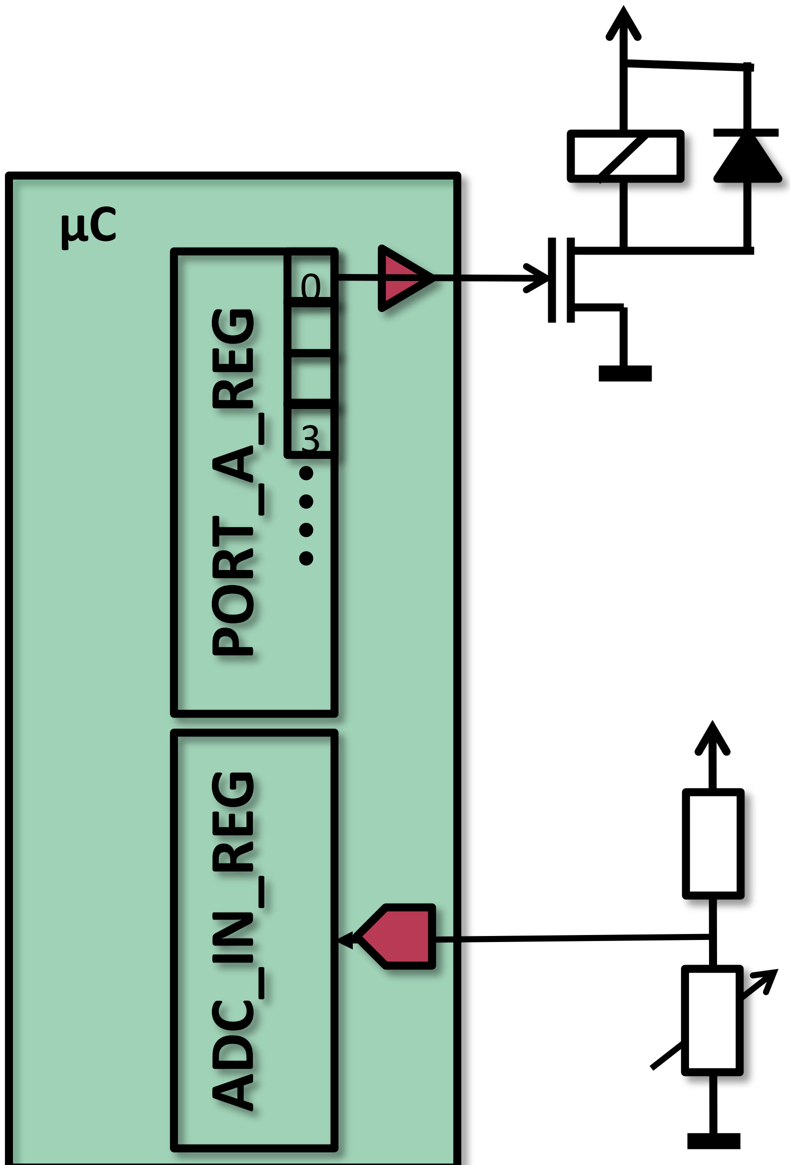


Example: direct connection with environ.

- Example #2: heating control:
  - PTC (Positive Temperature Coefficient) temperature-dependent resistance is used to measure the temperature
  - Analog-to-digital converter (ADC) is used to digitize the temperature-dependent voltage
  - Relay that controls heating is switched in accordance with the temperature

```
#define HEATING_ON_BIT        (0)
#define TEMP_LIMIT  (25) //below-> heating on
#define TEMP_HIST (2)   //HISTeresis
#define SCALE_FACT       (0.145)
.
.
// PORT_A_REG: I/O register nested in memory
// ADC_IN_REG: register for ADC resultnested in memory
TEMP = ADC_IN_REG*SCALE_FACT;
if (TEMP < TEMP_LIMIT){
     PORT_A_REG |= 1 << HEATING_ON_BIT;
} else
if  (TEMP > (TEMP_LIMIT+TEMP_HIST)){
    PORT_A_REG &= ~(1 << HEATING_ON_BIT);
}
.
.
```





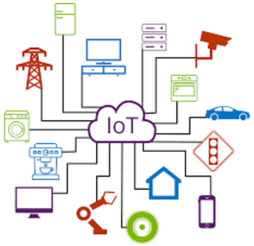
Evolution of embedded systems

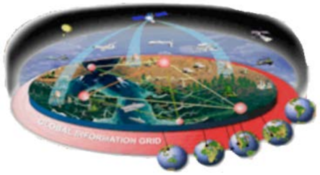
- **Milestones in short**
  - In the ‘60s: first embedded systems were the controllers used in Apollo program.
  - ‘70s: popular microprocessor manufactured in high volume (e.g. 8086), first PCs.
  - ‘80s: microcontrollers with *integrated pripherals.*
  - ‘90s: handheld devices, embedded systems in household appliances. System on chip ICs.
  - From year 2000: embedded systems become part of everyday life
    - Ambient systems (around us, in our environment)
  - 2010s: connecting embedded systems into complex systems:
    - Internet of Things (‘network of embedded systems’)
    - Cyber Physical Systems (‘embedded systems exploiting high level of artificial intelligence and integration of databases’)







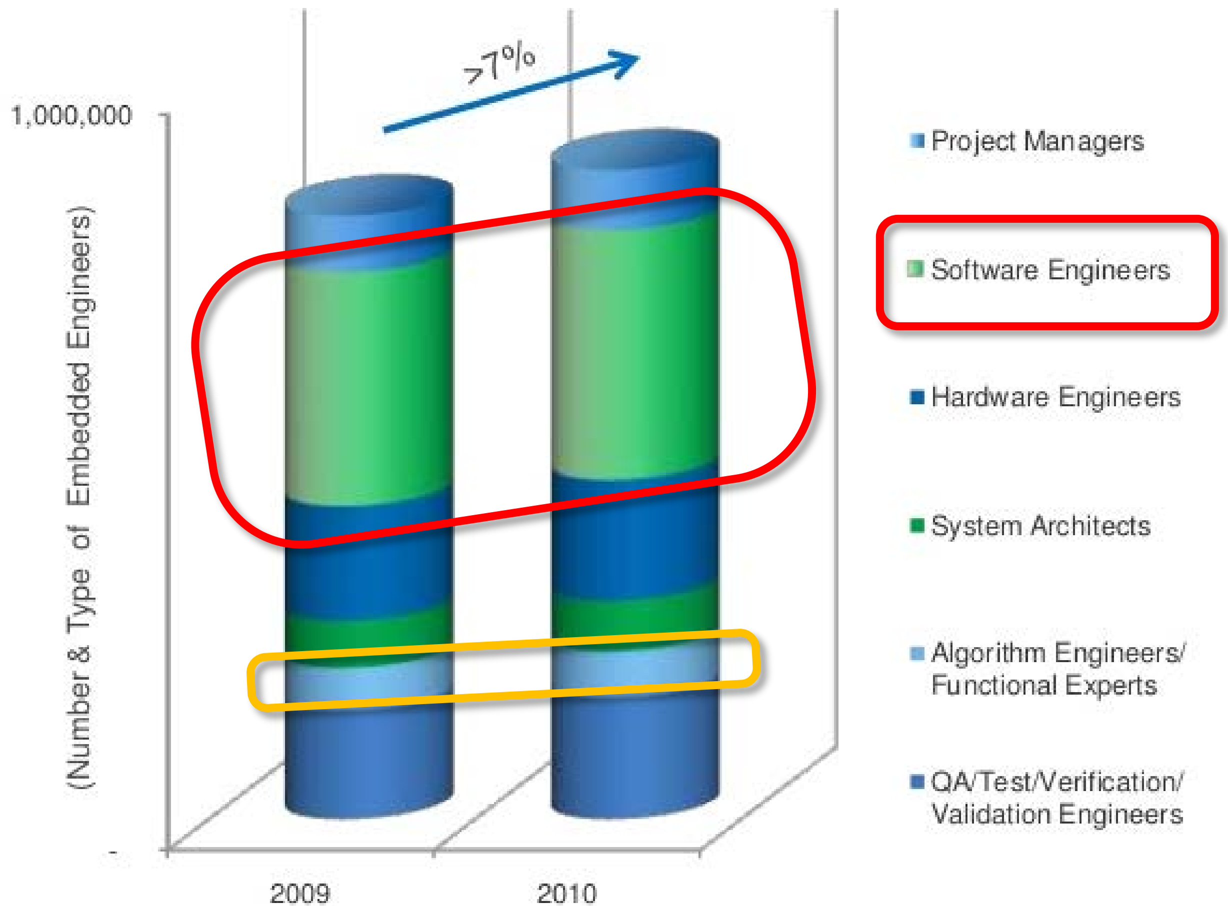


MŰEGYETEM 1782

Development of embedded systems

- **Development tasks**
  - HW development
  - SW development
  - Testing
  - Tight cooperation among different phases of development
- **HW development**
  - Circuit design, implementation, initial testing of operation
  - ‘Fine-tuning’ of circuit based on development experience
  - HW components change the least frequently among system components
  - More and more multifunction devices exist that reuire ‘only’ SW development
- **SW development**
  - SW development plan is needed
  - Development of both low- and high level components
  - Continuous development, modified dynamically, much more frequently changed compared to HW
  - Most of developers are SW oriented in the embedded field (including testers)
- **The course focuses on embedded SW development and data processing techniques and systems**

Engineering tasks

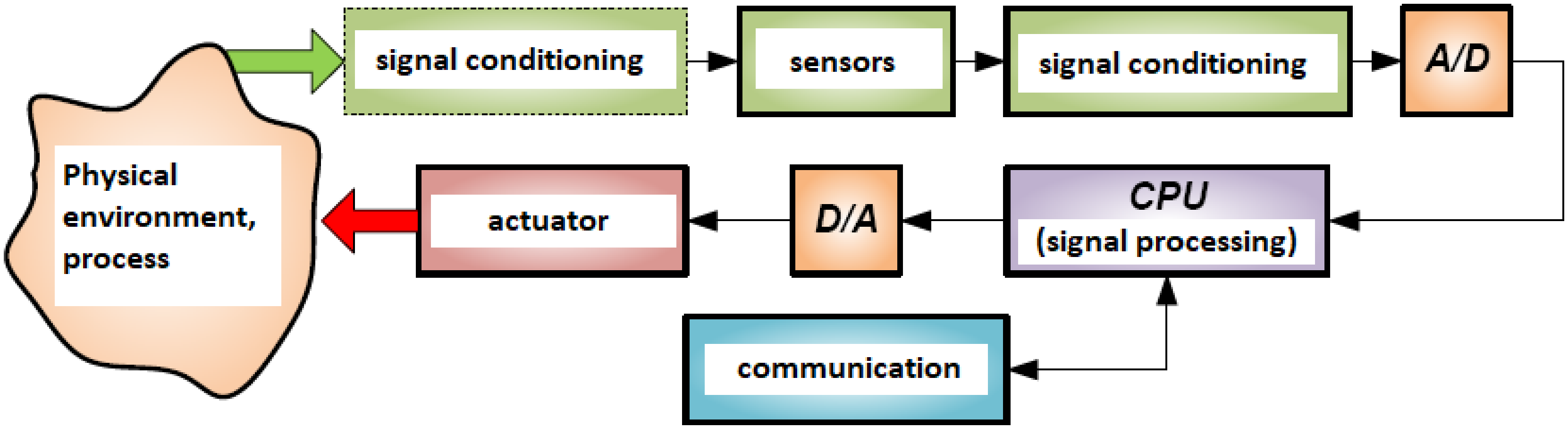


Specialties of embedded SW

- HW-aware programming
- Implementation of functionality in SW (either at system- or course-code level) is not enough, awareness is required
- The specialties of the HW must be considered
  - The SWs are for general use but they cannot be totally independent of the platform
- Save the resources:
  - Memory/Data
  - Processor time
    - Complexity of algorithms
  - Current consumption
- Runtimeutási may be critical (real-time systems)
- Understanding of the code operation is required: what resources are used, how much a code is resource consuming, etc

Architecture of embedded systems

- **Main components of embedded systems:**
  - Connection to physical world (input):
    - Sensor/transducer
    - Signal conditionerJelkondicionáló
    - Input devices
  - Computing unit
  - Communications
  - Actuator



Input devices

- **Input devices**
  - Signals from environment, e.g. temperature, luminance,…
  - Human Interface (HMI)/User Interface (UI), e.g. push button, encoder
- **Definition of ‘sensors’:**
  - Transducer: transforms a physical quantity into an other type of physical quantity
  - Sensor: transforms a physical quantity into an electrical quantity (voltage tipically)
    - Either in a direct or indirect way, e.g. strain-gauge: strech → change in resistance → change in voltage
- **Categories of sensors:**
  - Active: external excitation is needed (e.g. strain-gauge, thermistor)
  - Passive: electrical signal is generated by the device at its output (e.g. photo diode, thermocouple)

MŰEGYETEM 1782

Type of sensors

- **Signals from the environment**
  - Temperature, luminance, air pressure, humidity, gas presence, airflow, radiation, CCD (charge-coupled device)
- **Vibroacoustic signals**
  - Microphone, vibration sensor, geophone
- **Distance, proximity and presence sensors**
  - Ultrsound-based or IR-based distance sensing, PIR (passive infrared sensor) in motion detectors, reed relay, contact switch, inductive/capacitive proximity sensors
- **Sensing of position**
  - Accelerometer, magnetic compass, gyroscope, encoder, linear variable differential transformer
- **Mechanical signals**
  - Torque sensor, strain-gauge, force-sensing resistor (FSR)

Signal conditioning

- **Goals of signal conditioning**
  - Amplification (e.g. generate 1V from 5mV)
  - Level matching (e.g. from +/-1V range to 0V…2V range)
  - Galvanic coupling (e.g. high voltage disturbance)
  - Impedance matching (e.g. buffer amplifier)
  - Linearization (non-linear amplifier made linear usually digitally)
  - Filtering (removing noise)
- **Nowadays high complexity sensors provides compact form and integration of signal conditioning not only the sensor itself**
- **Further advancement when the sensor provides digital output, i.e., signal conditioning is obviously integrated as well**

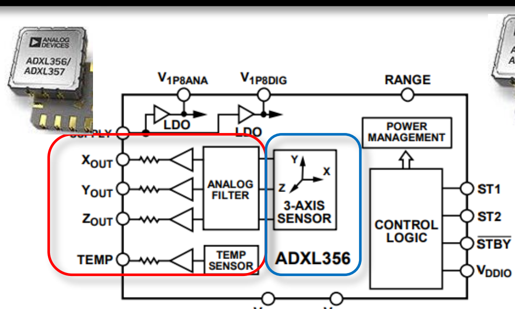
Complexity of sensors

- The complexity of sensors keeps increasing:
  - Only the sensor without any electrical components
  - Analogue signal conditioning, like amplification is integrated
  - Integrated ADC, digital interface, other high level functions:
    - e.g. internal calibration, identification, configuration
  - Fully integrates 'smart' sensor, high level data acquisition subsystem

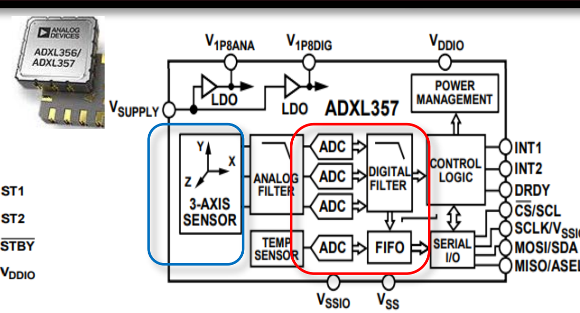
Increasing complexity

Example: accelerometer – increasing complexity

Accelerometer
Only the sensor
Output is charge



Sensor with
integrated signal conditioning
Output is charge



Sensor+signal conditioning+
internal ADC
Digital output

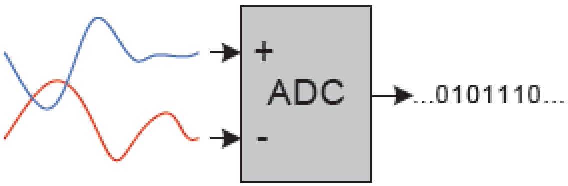
Fully integrated
Wireless comm.
Digital output

Sensor choice

- Advantages of high complexity sensors
  - Less external components
  - Less development time
  - Less errors
  - Many services are integrated
- Advantages of low complexity sensors
  - Cost efficient for high volume manufacturing
  - No unnecessary functions
  - Can be tailored for the specific development goal with special function and features

Analog-to-digital converter (ADC)

- **Frequently used ADC types**
  - Successive approximation
  - Flash
  - Sigma-delta
  - Dual slope



- **Main features**
  - Sampling frequency/Conversion time
  - Resolution (number of bits)
  - Zero order hold (ZoH) is needed or not
  - Linearity
  - Delay

ADC typical parameters

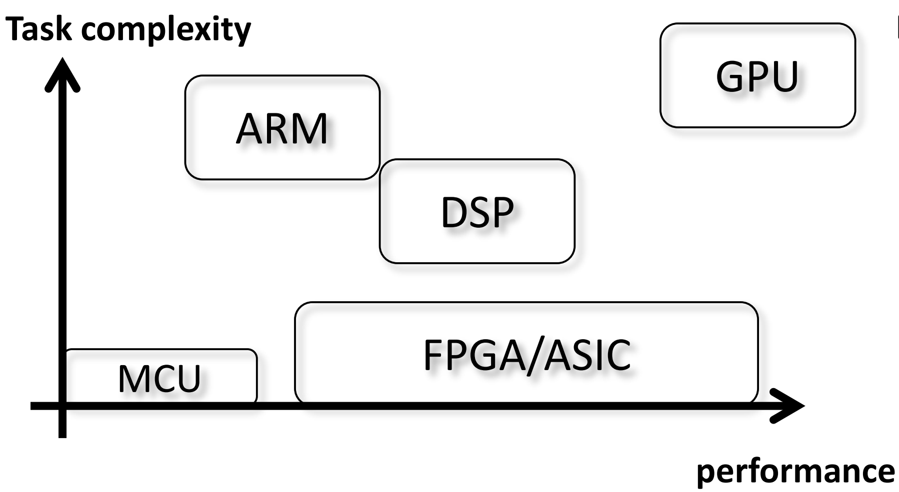
ADC Type	Flash	Delta-Sigma ΔΣ	Integrating (slope)	Successive approximation (SAR)
Application:	scope, RF	general use	audio	accurate
Operation principle	Parallel comparator array	Oversampling with digital filtering	Integration vs known reference charges	Binary search comparison
Speed	Very fast (up to few GHz)	Slow (Hz) to Fast (few MHz)	Very slow (mHz) to Medium (kHz)	Medium (kHz) – Fast (few MHz)
Resolution	Low, <14bit	Medium to very high, 12-32 bit	Can be very high, 32 bits	8 – 20 bit
Power	Very high	Low	Low-High	Low-Medium
Noise immunity	Low	Medium-High	High	Medium
Design complexity	High	Low	High	Low
Implementation cost	High	Low	Medium to very high for precision	Low

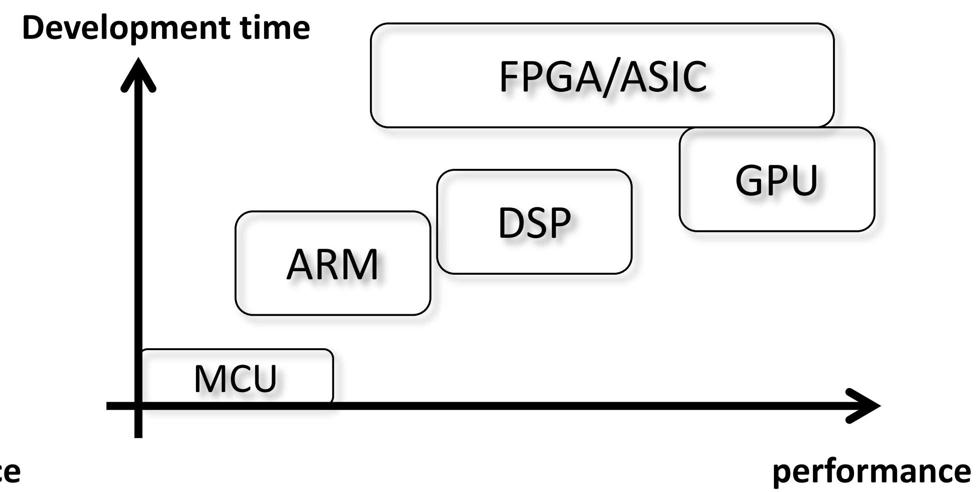
ADCs in embedded systems

- External ADC
  - In special cases (high resolution, accuracy, speed, low noise)
  - Difficulty: HW and SW matching to the uC is a must
    - Development time and possible errors
- Internal ADC
  - Lots of uC have internal ADC
    - In a general purpose uC: 10-16 bit successive approximation
    - Audio processors: rare, sigma-delta, ~16bit
  - Advantages: integrated, matching done, function library offered, template/example codes available

Control unit

- **Most important types of control units**
  - uP (Microprocessor)
  - uC (Microcontroller)
  - FPGA (Field Programmable Gate Array)
  - DSP (Digital Signal Processor)
  - GPU (Graphics Processing Unit)
  - ASIC (Application-specific Integrated Circuit)





Microcontroller

- Microcontroller = microprocessor + integrated peripherals
- Peripherals:
  - Memory (data and program in separated memory) – SRAM, Flash, ERAM
  - Timers – measurement of time, event generation
  - Communications (UART, SPI, I2C, CAN, USB, Ethernet)
  - ADC and DAC
  - GPIO (General Purpose Input Output)
  - Energy management
  - Debug interface
- Typical clock frequency: 1 MHz … 100 MHz+
- Choice preferences:
  - Availability
  - Adequate complexity for the task, peripherals (e.g. automotive, video, security, cryptography)
  - Price (not only chip but development SW and debugger must be considered)
  - Previous experiences
  - Support (technical support, forums, function library, development environment, examples, debug features)
  - Tokozás

Microcontroller examples

ATTiny25
1USD (compare:
1pc of inverter:
0.2USD)
8 bit architecture
10 MHz
2 kB prog MEM
128 Byte RAM
2 Timer, 2 PWM
10 bit ADC 4 ch
6 GPIO (spec func)
SPI

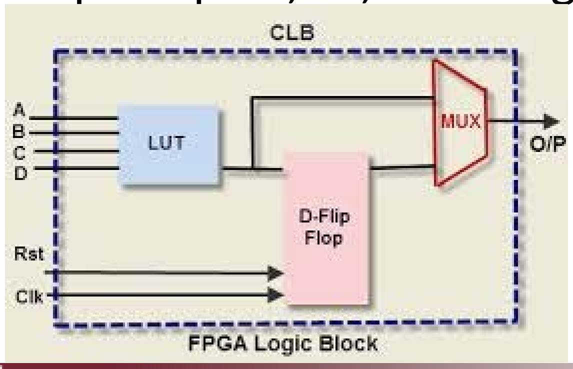
ATmega128
4USD
8 bit architecture
16 MHz
128 kB prog MEM
4 kByte RAM
4kB EEPROM
8 bit HW multiplier
5 Timer, 8 PWM
10 bit ADC 8 ch
53 GPIO (spec funkc)
SPI, UART, I2C

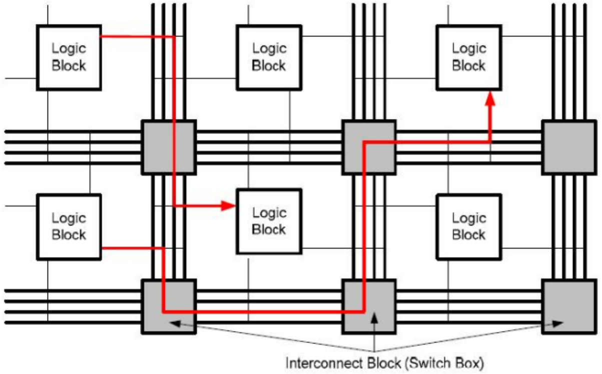
EFM32GG995 (Gecko)
8USD
32 bit architecture
48 MHz
1024 kB prog MEM
128 kByte RAM
6 Timer
12 bit ADC 8 ch
12 bit DAC 2 ch
93 GPIO (spec func)
3*SPI, 2*UART, 2*I2C,
USB
OPA
Sensor Interface
DMA
HW encryption
LCD driver

Balckfin BF537
30USD
16/32 bit architecture
600 MHz
1db 16 bit MAC
2 db 40 bit ALU
4db 8 bit ALU (video)
Parallel operations
132 kB prog/data RAM
Parallel data fetch
HW supported cycles
Circular buffer
8 Timer
48 GPIO (spec func)
SPI, 2*UART, 2*I2C, Ethernet,
CAN, I2S
DMA

FPGA (Field Programmable Gate Array)

- Circuit of general logic cells/gates (combinatorial and sequential logic)
- The logical relationship among logic gates is programmable
- Flexible: no new circuit is needed when functionality is modified – 'only' the SW has to be replaced
- Parallelism is inherently supported: one SW defined component can be duplicated 'endlessly' (the limit is the number of logic cells in the FPGA unit)
- Traditional FPGA development is difficult:
  - Development requires highly experienced professionals
  - The functionality has to be implemented at a low level (like shift register), therefore time consuming – nowadays higher level modules are readily available
- Used for high computation load, fast or parallel needs (high sampling rate, multiple inputs, RF, video signal processing)



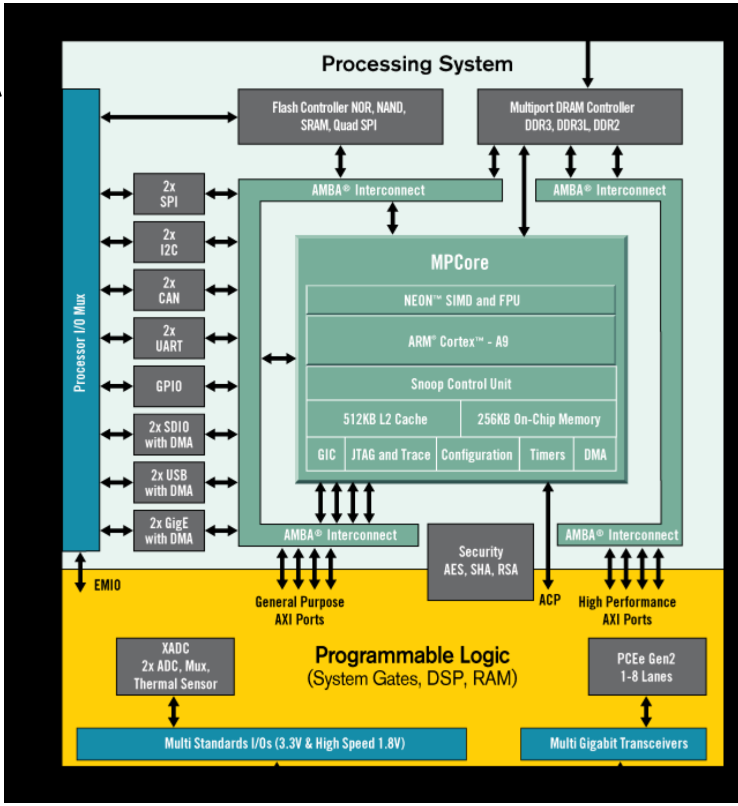


Digital signal processors (DSP)

- Special HW components and architectures to speed up computation, like:
  - MAC (Multiply and accumulate: a+=x*y)
  - Circular buffer
  - Parallel memory access to several memory blocks
  - HW supported cycles
  - Even floating point multiplication in one CLK cycle
  - HW supported division and extraction of roots,…
- Applications:
  - Multimedia: compressing, effects, coding (e.g. MP3, JPG, MP4), equalizer, noise filtering
  - Control systems: engine control, state observer, feedback systems
  - Math operations: mtx multiplication, trigonometrical functions
  - Measurements: noise filtering, parameter estimation
  - Info comm: modulation/demodulation, coding, compression
- More and more DSP functions appears into general purpose uC (DSC: digital signal controller)

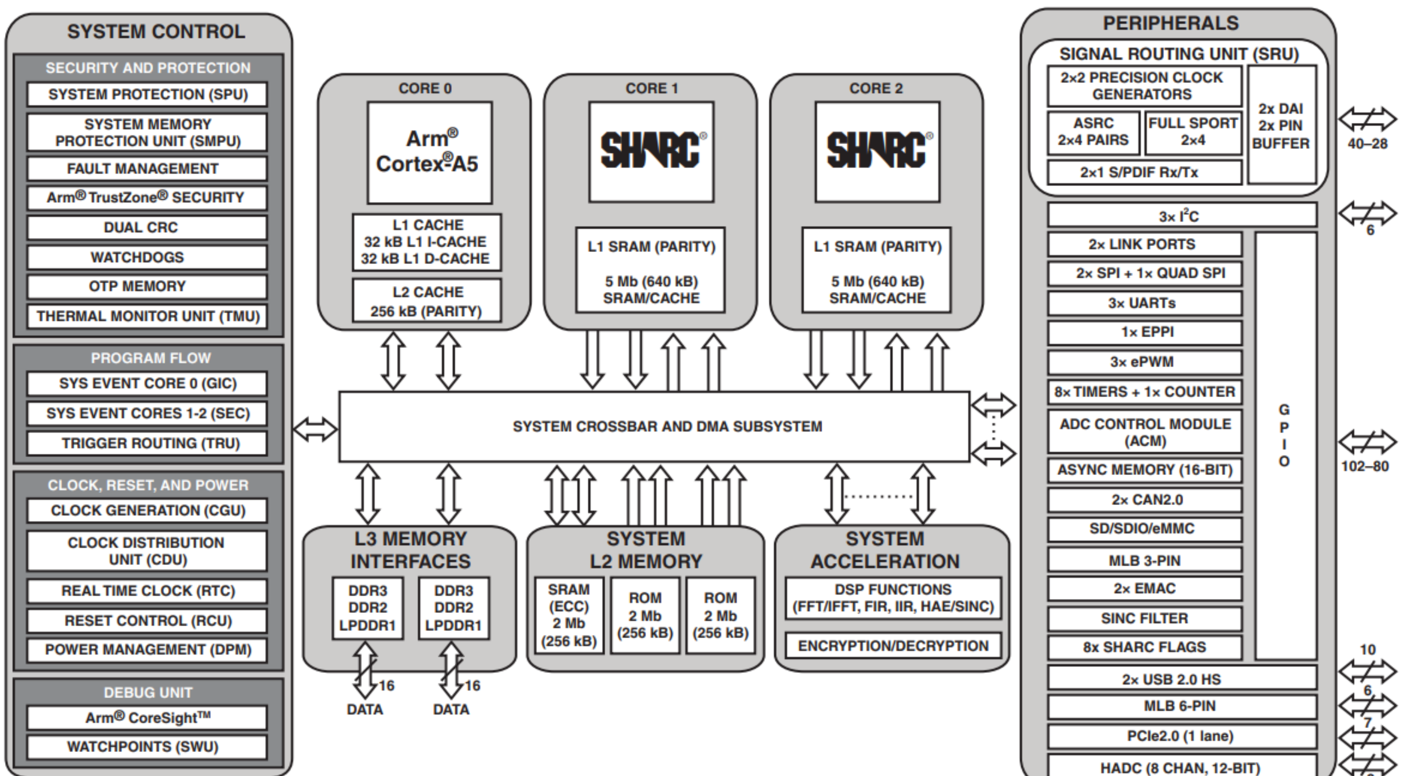
Hybrid solutions

- Several types of processing unit integrated into one application
- Tasks can be decomposed into for most appropriate computing type:
  - Decoding digital graphic information by FPGA DSP in a TV
  - uC-based handling of remote controller and menu system
- System containing several types of processing units can be integrared into a single IC (system-on-chip: SoC) :
  - Soft-core processors to be downloaded to an FPGA are available
  - A uC can be integrated into an FPGA



Hybrid solutions

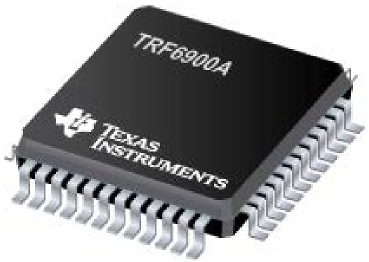
- Analog Devices SC589:
  - 2 pcs DSP core: computational tasks
  - 1 pc ARM cortex-A5: general tasks, e.g. communications, peripheral handling, etc.



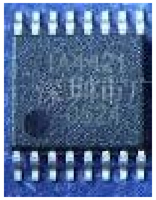
Communications

- Fully integrated
  - E.g. SPI, USB
- Integrated comm protocol + external level matching
  - pl.: UART + RS232 level matching: logical levels transformed into +/-3V…25V range
- External peripherals (Ethernet, CAN, WiFi, Bluetooth, ZigBee)
- SW implementation is possible
  - Change of GPIO pins in SW
  - Timing is critical since background processes may violate timings

Communications units, examples



- TRF6900 (TI)
  - Bit level communications (‘wireless wire’)
- IA4420 (Silabs)
  - Byte level communications



- CC2420 (TI)
  - Packet level comm
  - Automatic receiver detection

- ESP8266
  - WiFi modul
  - Integrated protocol stack

- Some uCs available with integrated RF module
- Complex unit offers shorter development time, but special functionality may not be implemented or can be a hard task