Real Time Systems – SS2016

Prof. Dr. Karsten Weronek
Faculty 2
Computer Science and Engineering

embedded Systems

What is an embedded System

typical characteristics of an embedded system (just indicators):

- has at least one CPU
- has a physical context
- provides dedicated set of services to the end user
- does not provide general computing services to the end user
- often part of a larger environment
- have often real-time constraints

Ninety-eight percent for all microprocessors are manufactured as components of embedded systems.

Examples for embedded Systems

Embedded systems range from portable devices such as

digital watches and MP3 players,

to large stationary installations like

traffic lights, factory controllers,

and largely complex systems like

hybrid vehicles, magnetic resonance imaging, and avionics.

Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

....

embedded vs general purpose computers

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

embedded systems:

- low power consumption
- small size
- rugged operating ranges
- low per-unit cost

Constraints for embedded systems

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

- size of resources
  - clock frequency
  - bus width (8,16,.. Bit)
  - memory Size (no disk)

- lack of power
- minimizes I/O-devices
- often real-time requirements
- to be developed on a different systems

Requirements for embedded systems

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Integration

- integrate many stuff on chip to ovoid peripheral devices

Hardware requirements:

- think about temperature, humidity, dust, shock, vibration

Power consumption:

- when there is no power you need to save power by low current devices using CMOS, reduce uptime, clock speed

Real-Time requirements:

- chips are optimized regarding context change latency times

Security:

- optimized uptime, MTBF, think about EMC(EMV), watch dogs, self-test, restart procedures, emergency tasks

Price:

- optimize price per unit by optimizing development process

N-MOS-FET, P-MOS-FET

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

N-MOS-FET:

N-channel Metal Oxide Field-Effect Transistor

like an air gap switch (Relais), conducts on positive Voltage

blocks on 0V

P-MOS-FET:

P-channel Metal-Oxide Field-Effect Transistor

blocks on positive Voltage

conduct on 0V

Picture of NAND using MOS-FETS

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

will be added later

Remember Boolean Algebra

every logical gate (logisches Gatter) can be made out of NANDs:

Given:

NAND: $C = \neg(A \wedge B)$

you have a NOT by:

$C = \neg(A \wedge A) = \neg(A \wedge A) = \neg(A) = \neg A$

you than have an AND by combining NAND and NOT:

$C = \neg\neg(A \wedge B) = A \wedge B$

and using de Morgan's Law you get the OR:

$C = \neg(\neg A \wedge \neg B) = (\neg\neg A \vee \neg\neg B) = (A \vee B)$

The XOR will be your homework!

How to set a Bit

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

```
before:   00110101    01110101
OR        01000000    01000000
after:    01110101    01110101
```

How to drop a Bit

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

```
before:   00110101    01110101
AND       10111111    10111111
after:    00110101    00110101
```

How to toggle a Bit

before:	00110101	01110101
XOR	01000000	01000000
after:	01110101	00110101

More complex gates

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Using simple gates you can prepare:

- Flip-Flops (mono stable, bistable, J-K-Master-Slave-FF, etc.)
- Shift Registers
- Adders
- Latches
- Memory Cells
- ALUs (arithmetic, logical units)
- Control Units, Execution Units
- GPUs, FPUs, CPUs

So can develop the whole system based on gates.

....

What are the basic processor architectures?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

1. ISP

   Instruction Set Processor

2. ASIP

   Application-Specific Instruction-set Processor

3. ASP

   Application-Specific Processor

1. Instruction Set Processor (ISP)

These processors are all processors that come with a dedicated hardware with a pre-defined instruction set.

These processor are similar to general purpose CPUs like in PCs

However they may be adopted to the general needs of embedded systems (e.g. minimal size, reduced power consumption, reduced memory, limited I/O, etc.).

For the developer a customizing can only be done by Software

However such CPU can be used when developing an embedded system on a chip.

Since there are standardized, a large number of production leads to a low price per unit.

2. Application Specific Instruction-Set Processor (ASIP)

ASIPs are optimized in Hard- and Software for the specific use of the new system

Therefore the Hardware and the instruction set needs to be defined during the design of the embedded system

This architecture is then optimized in Hard-and Software for the intent purpose.

However the amount of produced Processors is limited an there is are lot of development activities to be done.

3. Application Specific Processor (ASP)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

This a completely new designed Hardware processor

This leads to an optimal performance.

However since a due to the complexity this should be done only for processors that are needed for a single purpose e.g. en- and de-cryption, DSPs(digital signal processors), IO-processors, network processor)

How to bring the architecture onto the Chip?

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

You can produce a processor on a chip in the following 3 ways (technologies):

A. Full-Custom Logic-Fabric

B. Configurable Gate-Fabric

C. Programmable Gate-Fabric

A. Full Custom Logic Fabric

This is a full custom design:

The System designer designs the complete hardware for the new system.

He may use standard block that are developed earlier for other systems (standard cell libraries).

When the design is complete, lithography masks will be produced for the processing of the Silicon wafers to produce the new processors

Since the production of the masks is very expensive (up to 1 Mio€) there should be no errors in the design!

B. Configurable Gate-Fabric

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

This is a semi-custom disign

This has a large number of gates (gate fabric by gate array).

Since the last process in chip production is the connection of the different gates you have only the masks for the wiring.

You may also configure the type of the gates.

You take may use an already existing design for the chip.

This is less expensive since you can use partly standard components.

These are called ASICs (Application-specific integrated circuit)

C. Programmable Gate-Fabric

The hardware of the Programmable Gate Fabric is very similar to the configurable Gate-Fabric.

However the customization can be done using Software.

This has the advantage, that the configuration may changer over time by applying Software patches.

Also called

- PLD (Programmable Logic Devices)
- FPGA (Field Programmable Gate Arrays)

System on a Chip (SoC)

At the end you have a system on y chip (SoC).

A typical SoC consists of:

- a microcontroller, microprocessor or digital signal processor (DSP) core – multiprocessor SoCs (MPSoC) having more than one processor core
- memory blocks including a selection of ROM, RAM, EEPROM and flash memory
- timing sources including oscillators and phase-locked loops
- peripherals including counter-timers, real-time timers and power-on reset generators
- external interfaces, including industry standards such as USB, FireWire, Ethernet, USART, SPI
- analog interfaces including ADCs and DACs
- voltage regulators and power management circuits

A bus – either proprietary or industry-standard such as the AMBA bus from ARM Holdings – connects these blocks. DMA controllers route data directly between external interfaces and memory, bypassing the processor core and thereby increasing the data throughput of the SoC.