# Course Digital Systems and computer Architecture

Course

Meets requirements of following modules(MID) Course Organization Assessment Course components <u>Lecture/Exercise</u> <u>Lab</u>

Responsible: Prof.Dr.G.Hartung

# Course

### Meets requirements of following modules(MID)

- in active programs
  - Ba MT2012 CA
  - Ba TIN2012 CA

### **Course Organization**

Version		Course identifiers			
created	2013_07_15	Long name	Digital Systems and computer Architecture		
VID	1	CID	F07_RA		
valid from	WS 2012/13	CEID (exam identifier)			
valid to					

Contact hours per week (SWS)		Total contact hours		Max. capacity	
Lecture	2	Lecture	30	Exercise (unsplit)	
Exercise (unsplit)	1	Exercise (unsplit)	15	Exercise (split)	40
Exercise (split)		Exercise (split)		Lab	18
Lab	1	Lab	15	Project	
Project		Project		Seminar	
Seminar		Seminar			
Tutorial(voluntary)	2	Tutorial (voluntary)	30		

### Total effort (hours): 150

### Instruction language

• Englisch

### **Study Level**

• Bachelor

### **Prerequisites**

- Basic skills in Digital Systems and Computer Engineering
- Basic Skills in Programming, especially C
- Basic Knowledge in Operating Systems

### **Textbooks, Recommended Reading**

• Wakerly: Digital Design Principles and Practices

• Tanenbaum: Computer Architecture

### Instructors

• Prof.Dr.G.Hartung

### **Supporting Scientific Staff**

• Dipl.-Ing. C. Ctistis

## **Transcipt Entry**

Digital Systems and Computer Architecture

### Assessment

	Туре
wE	normal case (few participants: oE)

Total effor	rt [hours]
oE	20

Frequency: 2/year

# **Course components**

### Lecture/Exercise

### **Objectives**

### Contents

- Digital systems
  - Description
    - Schematic Design
    - HDL
      - Gajski-Kuhn systematic for HDL
        - Structure
          - Hierarchical Digital Design
          - SOPC Design
        - Behavior
          - Switching networks
          - State machines
          - Algorithmic Behavior
        - Technology (see Implementation)
    - Automata
      - State machine
      - Programmable Processor
  - Implementation
    - CMOS Circuits
    - PLD
      - PLS
        - CPLD
        - FPGA
    - ASIC
- Computer Systems
  - Sequential Computing
    - Principal Modells
      - von Neumann
      - Harvard
    - Processor Examples

- CISC
  - Intel X86
  - RISC
    - e.g. Altera NIOS II
- Stack Machine
  - JVM
- Programming support
  - Runtime system
    - Variable handling for procedural languages
  - OS support
    - Memory Management
      - Cache
      - MMU
      - Virtual Memory
    - Interrupts
    - Timer
- Parallel Computing
  - Architectural Aspects
    - Taxonomies
    - NUMA architectures
    - COW architectures
  - Programming parallel Machines
    - Paradigms of parallel programming
    - Standards for high performance computing (HPC)

### **Acquired Skills**

- Design and Implementation of a hierarchical digital system
  - Designing Control with State machines
  - Interfacing to libraries
  - Algorithmic data processing
- Low-level programming of a processor
  - Assembler programming
  - Using Interrupt and Timer
  - Interfacing to hardware system description
- Parallel Programming
  - Implementation using a standard for HPC
  - Performance Evaluation

### **Additional Component Assessment**

### <u>Lab</u>

### **Objectives**

### Contents

- Digital Design
  - Development of a hierarchical digital design
  - Test using test vectors
  - error correction
- Assembler programming of SOPC system
  - Programming simple algorithms in Assembler
  - Translating state machines into Assembler programs
  - Using timer and interrupt
  - Testing and debugging
  - Comparison digital system to SOPC system
- Parallel programming
  - Parallelization of a program using a COW
  - Coding and debugging
  - Performance measurement

- Designing an IT system using various technologies
  - Digital technology based on HDL
  - SOPC technology combined with Assembler programming
- Exploring the potential of parallel processsing
  - Using a HPC programming standard
  - Performance evaluation of a parallel implementation

#### **Operational Competences**

- Extraction of relevant information from task description
- Implementation of
  - digital system
    - low level programming system
    - parallel system

### **Additional Component Assessment**

Туре			
fPS	supervised problem solving (4h)		
fSC	supervised scenario study (20 h)		

Contribution to course grade		
fSC	attestation	
fPS	attestation	

Frequency: 1/year

Das Urheberrecht © liegt bei den mitwirkenden Autoren. Alle Inhalte dieser Kollaborations-Plattform sind Eigentum der Autoren.

FW FOSWIKI

Ideen, Anfragen oder Probleme bezüglich Foswiki? Feedback senden