Practical Embedded Systems Engineering Syllabus for Graduate Students with Multidisciplinary Backgrounds

Bastian Haetzer
Gert Schley
Rauf Salimi Khaligh
**Martin Radetzki**
Embedded Systems Engineering / CSEE / U Stuttgart
Overview

1. Context
2. Embedded systems courses
3. Practical (lab) course
4. Experiences
5. Conclusion
International Students

Faculty 5 (CS, EE & IT) International Study Programs

Degree

Non-Degree

Credit-Transfer (ECTS) (typ. Erasmus)

Double / Joint Degree Programs
INFOTECH Master

- INFOTECH is a multidisciplinary (EE & CS) international master program
  - On top of a qualifying Bachelor degree
  - Leading to a Master of Science Degree (M.Sc.) in Information Technology
  - With a regular study time of 4 terms (3 teaching + 1 Master Thesis)
- Is truly international
- Is multidisciplinary
- 4 specializations (majors) can be selected
  - Communication Engineering and Media Technology
  - Embedded Systems
  - Micro- and Optoelectronics
  - Computer Hardware/Software Engineering
History and Facts

- has been set up in 1999 (is in its 12th year)
- Attracted some 9500 applications
  - Had applications from graduates of more than 1000 different universities
  - From 116 different countries
- Admitted some 600 students
- Conferred 310 Master Degrees
- Offers 4 majors
- Is supported by some 20 professors of the faculty (faculty total of 30), 10 professors from other faculties and some 5 industry lecturers
- Some 55 Lecture Courses
- Some 10 Lab Courses
- Some 10 Seminars
- Currently admits 80 students / yr from > 1200 applicants
Admission Selection

- Several “Competition Groups” (e.g. World, Partner, etc.)
- Selection mainly on GPA basis
  - Within group / University/ Country/ World
  - Based on comprehensive experience and reference data from previous applications
  - With weighted GPA (Itec Performance Factor)
  - And based on Ranking of applicant
  - And based on “Reputation of University” (Media, Own experience)
INFOTECH

the international Master’s program in Information Technology

“Old” Program - OSER

Introductory Semester

Qualifying Exams

Term 1 and 2

Internship

Master Thesis Project

“New” Program - NSER

Term 1

Term 2 and 3

Master Thesis Project

Time

- ½ yr

0 yr

½ yr

1 yr

1½ yr

2 yr
The Curriculum INFOTECH Master

- **Total 120 Credit Points**

**1st Term**
- 30 Credit Points: Core Modules (CM) 6 CP each
- 27 Credit Points: Basic Modules (BM)

**2nd and 3rd Term**
- 30 Credit Points: Supplementary Modules (SM) Group 3 (3 CP)
- 12 Credit Points: Supplementary Modules (SM)
- 9 Credit Points: 1 Lab, 1 Seminar

**4th (5th) Term**
- 30 Credit Points: Master Thesis Project
- 9 Credit Points: Internship

Conditions: Internship, German Proficiency Certificate

2 (2.5) Years
Basic Modules (1st Term)

- Advanced Higher Mathematics (AHM)
- “Container” Electronics and Communication (EC) with
  - Communications (Com)
  - System and Signal Theory (SST)
  - Radio Frequency Technology (RF)
  - Electronic Circuits (EIC)
- “Container” Computer Science (CS) with
  - Programming Languages (PL)
  - Computer Architecture and Organization (CAO)
  - Data Structures and Algorithms (DSA)
  - Operating Systems (OS)
Embedded Systems Core Modules

Core Modules (CM): 5 out of 12

- Advanced Processor Architecture
- Communication Networks I
- Communications III
- Design and Test of Systems-on-a-Chip
- Distributed Systems
- Embedded Systems Engineering
- Industrial Automation Systems
- Intelligent Sensors and Actors
- Modelling, Simulation and Specification
- Real-Time Programming
- Software Engineering for Real-Time Systems
- Statistical and Adaptive Signal Processing
Syllabus design

University Environment
(studies program, infrastructure, staff, ECTS)

Teaching Techniques
- exercises vs. laboratory
- group work vs. individual work
- big project vs. small exercises
- predetermined vs. self-selected assignment

Students
(multidisciplinary background, working style)

high-level goals
(hw/sw co-design etc.)

low-level goals
(hw design, cross-compiling)

ESE Lecture
4.5 CP

ESE Exercises
1.5 CP

ES Lab. 6 CP
ES syllabus focus areas

**Systems-oriented**
- System specification
- Modelling
- Simulation
- Performance estimation
- Platform architecture

**Lecture with exercises**

**Hardware-oriented**
- System specification
- Modelling
- Simulation
- Performance estimation
- Platform architecture

Lab

**Software-oriented**
- Real-time analysis
- RT programming, RTOS
- Cross compilation
- Remote debugging

**Application-oriented**
- Multimedia
- Automotive
- Robotics
- …
ESE lecture with exercises

Theory (exercises on paper, with ILP):
High-level synthesis: from algorithm to RTL,
Pipelined data paths and controllers,
Software scheduling: periodic and sporadic,
Software on bare CPU vs. RTOS, schedulability tests,
System architecture, storage and communication,
HW/SW integration

Practice (exercises in lab):

<table>
<thead>
<tr>
<th>Task</th>
<th>Session</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>Session 1</td>
<td>Processor-peripheral communication with memory-mapped I/O</td>
</tr>
<tr>
<td>Task 2</td>
<td>Session 2</td>
<td>Implementation of the interrupt handler</td>
</tr>
<tr>
<td>Task 3</td>
<td>Homework</td>
<td>Design of a frame-based scheduler</td>
</tr>
<tr>
<td>Task 4</td>
<td>Session 3</td>
<td>Implementation of a framed-based scheduler</td>
</tr>
</tbody>
</table>

Session 4
Embedded Systems Lab
## Lab design trade-offs

<table>
<thead>
<tr>
<th>Lab experiments</th>
<th>Structured lab assignment</th>
<th>Large project</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ guidance</td>
<td></td>
<td>(+) motivation</td>
</tr>
<tr>
<td>+ focus</td>
<td></td>
<td>+ overview</td>
</tr>
<tr>
<td>– no big picture</td>
<td></td>
<td>– getting lost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working individually</th>
<th>Working in groups</th>
<th>Project team</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ grading</td>
<td>+ learn from each other</td>
<td>+ real-world</td>
</tr>
<tr>
<td>+ focus</td>
<td>– one does all the work</td>
<td>+ soft skills</td>
</tr>
<tr>
<td>+ individual advice</td>
<td>+ higher admission possible</td>
<td>– teaching costs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fixed assignment</th>
<th>Individual assignments</th>
<th>Self-assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ costs</td>
<td>+ adaptation to interests</td>
<td>(+) motivation</td>
</tr>
<tr>
<td>+ focus</td>
<td>– arbitration necessary</td>
<td>– risk</td>
</tr>
<tr>
<td>– invites plagiarism</td>
<td>– teaching costs</td>
<td>– teaching costs</td>
</tr>
</tbody>
</table>
Design flow

Canonical embedded systems design flow

specification

partitioning

hardware design

interface design

software design

integration

verification

Design flow in lab

spec. from paper: intersect. test algorithm

manual partitioning

hw impl. (VHDL)

coproc unsched.

coproc sched.

simulation

FSL

sw impl. (C)

1st iteration:
algorithm

2nd iteration:
coproc. driver

FPGA platform

test / perf. measure on board

Canonical embedded systems design flow

Design flow in lab
FPGA platform

Initial try: ARM 7/9 + FPGA

Xilinx Univ. Programme boards

old: XUPV2P Virtex-II

new: XUPV5 Virtex-5
Many students got lost in part 2 due to
- lack of overview, unstructured working style
- lack of basic knowledge, in particular in VHDL
Admission requires entry test in C and VHDL

<table>
<thead>
<tr>
<th>Part</th>
<th>Session</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session 1</td>
<td>XPS Tutorial (Platform building + basic I/O)</td>
</tr>
<tr>
<td></td>
<td>Session 2</td>
<td>Software impl. of algorithm + performance measurement</td>
</tr>
<tr>
<td></td>
<td>Session 3</td>
<td>VHDL crash course (simple traffic light)</td>
</tr>
<tr>
<td><strong>Part 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session 4,5</td>
<td>Dataflow graph of algorithm, unscheduled implementation + datapath testbench</td>
</tr>
<tr>
<td></td>
<td>Session 6</td>
<td>Scheduling of operators e.g. cross product</td>
</tr>
<tr>
<td></td>
<td>Session 7</td>
<td>Datapath with scheduled operators, simulation</td>
</tr>
<tr>
<td></td>
<td>Session 8,9</td>
<td>FSL-interface design (state machine) + integration of coprocessor into platform</td>
</tr>
<tr>
<td></td>
<td>Session 10</td>
<td>Driver implementation + performance measurement</td>
</tr>
<tr>
<td><strong>Part 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Session 11</td>
<td>System simulation + software enhancement (assembler)</td>
</tr>
<tr>
<td></td>
<td>Session 12</td>
<td>Pipelining of FSL communication</td>
</tr>
</tbody>
</table>
Deliverables

- Software code, structured and with comments
- VHDL code, structured and with comments
- Demo
  - “sales presentation“ of the results: speedup, FPGA area
  - verbal examination, questioning of design decisions
- Written documentation (datasheet)
  - data path documentation: IFs, structure, schedule
  - controller documentation: IFs, FSM diagram
  - SW integration documentation: FSL interface, register map
  - measurements documentation
Research: embedded architectures

- Hardware/software architecture
- Many-core computing systems
- System communication
  - Bus subsystems
  - Networks-on-Chip
- Reconfigurable architectures
- Fault-tolerant architectures

Sample design (left): MPEG-4 H.264 video codec with ARM9 CPU and Xilinx Virtex5 FPGA to implement hardware accelerators
Research: design methods

- Modelling and simulation
- System-level fault simulation
- Analysis of properties such as reliability and robustness
Conclusion

- Embedded HW/SW design fundamentals, valid for any application
- Heterogeneity of students’ capabilities and backgrounds
- Monitoring results by qualitative and quantitative evaluation
- Monitoring the embedded market
- Continuous improvement of syllabus, courses