Digital System Integration and Programming

Introduction/Kick-Off

Barbara Gigerl, Rishub Nagpal

October 6th, 2021

IAIK – Graz University of Technology

Outline

1. Digital system integration and programming
2. About this course
3. Outlook: Projects

Digital system integration and programming

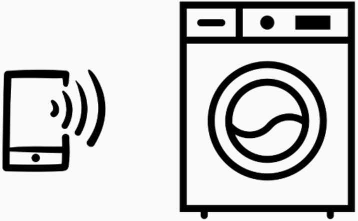
What is digital system integration and programming?

Digital system integration

- Digital systems: very complex
- System integration: connect multiple complex systems to achieve a certain goal

...and programming

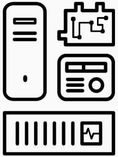
- Hardware and Software



What is a System-on-a-Chip?

A **System-on-a-Chip (SoC)** is a complex system which:

- consists of several components.
- Each component itself is a complex system.
- Components include:
  - CPU
  - Memory
  - Bus architectures
  - I/O modules
  - Co-processors
  - Analog circuits
  - ...



What is a System-on-a-Chip?

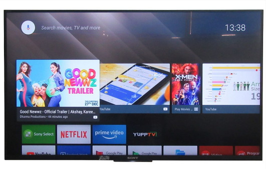
A quick history

- 1970s: VLSI design
  - VLSI = Very large-scale integration
  - Combining millions of MOS transistors into an integrated circuit
- 1990s: System-on-a-chip
  - System integration: integration of a complete system, that until recently consisted of multiple ICs, onto a single IC (a SoC)
- Today: SoC is the state-of-the-art principle for designing chips.

SoCs are everywhere

Smartphones

Tablets



Smart TVs

Cars

Example: Apple A12 Bionic

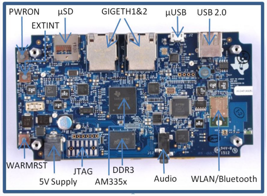
- Used in iPhone XS, XS Max, XR
- 7nm CMOS, 6.9 billion transistors
- Components:
    - 64-bit ARMv8.3A (6 performance CPUs, 4 energy-efficient CPUs)
    - Four-core GPU
    - *Neural Engine* with 8 cores
    - ...

Example: Qualcomm Snapdragon 865

- Used in smartphones by ZTE, Sony, OnePlus, LG, ...
- 7nm CMOS
- Components:
  - Several ARM Cortex-A77 and Cortex-A55-based CPUs
  - Deidcated processor for ISP for photos and videos
  - Wi-Fi
  - SPU: dedicated subsystem for boot-loader, key management unit, crypto accelerators, ...

Example: Sitara Processors of TI

- SoC for industrial applications
- Used in thermostats, firewalls, Lego Mindstorms
- 1 GHz ARM CPU
- On-chip quad-core PRU (Programmable Realtime Unit)



A Typical SoC

A **typical SoC** consists of:

- Processor(s): mostly ARM cores
- GPU: depending on the field of application, ranging from simple cores for small LCDs to 4k screens
- Co-processors: for security, real-time signal processing, ...
- I/O interfaces: Ethernet, SPI, USB, ADC, ...
- A bus connecting all components: AMBA, AXI, CoreConnect, ...

Why use SoCs?

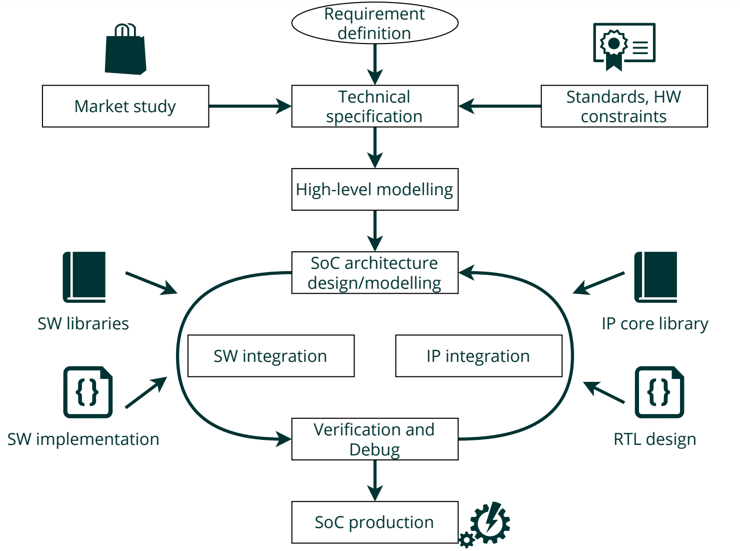
Advantages

- Low silicon area
- Power efficiency (no need for complex component **wiring**)
- Low manufacturing costs
- Smaller power supply unit

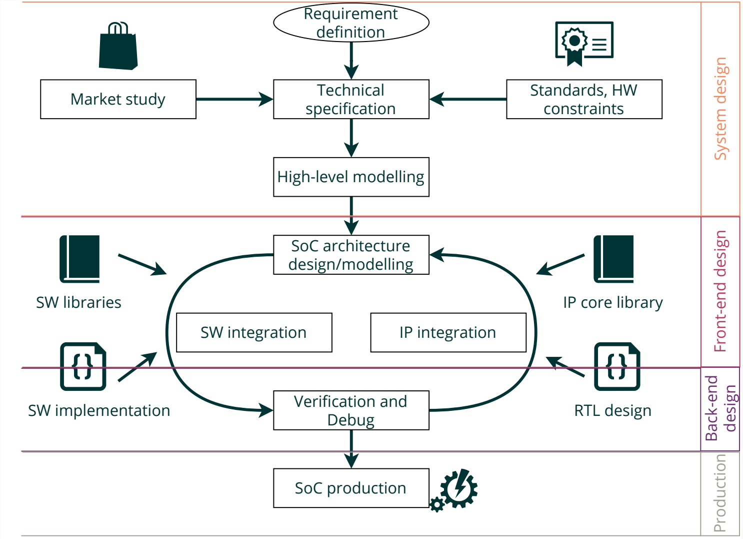
Disadvantage: resulting system is very complex

- High design and development costs

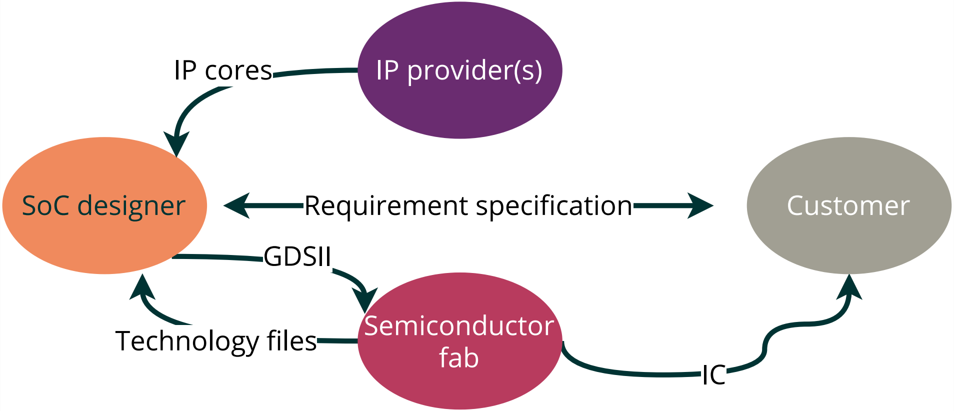
SoC Design Methodology



SoC Design Methodology



SoC Players



- GDSII: data format to describe ICs
- Technology file: information about manufacturing (metals, IC layers, ...)

About this course

Who are we?

Barbara Gigerl

PhD student @ Graz University of Technology

✉ barbara.gigerl@iaik.tugraz.at

✉ sip-team@iaik.tugraz.at

Who are we?

Rishub Nagpal

PhD student @ Graz University of Technology

✉ rishub.nagpal@lamarr.at

✉ sip-team@iaik.tugraz.at

Contact

- General information: `https://www.iaik.tugraz.at/course/digital-system-integration-and-programming-705007-wintersemester-2021-`
- Questions and concerns via Discord `https://discord.gg/9KKGfndsD5`
- Questions and concerns by E-Mail `mailto:sip-team@iaik.tugraz.at`

We Build our own SoC

- We focus on the front-end design
- We use an FPGA in order to build a prototype of our SoC
- Our Platform: Zybo Zynq Boards
    - 650Mhz dual-core Cortex-A9 processor
    - HDMI, VGA, USB, SPI, Ethernet, Audio, ...

Goals

- Build a working prototype
- Project management and self-organization
- Presentation of: ideas, results, technology in English
- Preparation for project/thesis

Previous Knowledge

Digital System Integration and Programming addresses **advanced-level students**. You need:

- Advanced knowledge about hardware including HDL (Verilog/VHDL) (Computer Organization and Networks is not enough!)
- Very good C/C++ skills
- Knowledge about Linux
  - Buildroot/Yocto, kernel modules, drivers, device trees, GPIO
- Knowledge about FPGAs and how to program them
- Knowledge about bus protocols, CPUs and networks
- Very good time-management skills
- Good presentation skills

Teaching method

We offer:

- Project driven work
  - Group oriented
  - Project centric
- Hands-on project with real hardware
- Upgrading of soft skills
  - Presentations
  - Speaking English
  - Group communication

Teaching method

We expect:

- Investment of time
  - SIP: 3 VU (5 ECTS)
  - $5 \times 25 = 125$ hours work $=$ 28 days of 8 hours
- Active communication within your group
- Active participation, presence during lectures

Grading

Your grade consists of:

- Project 1: 20%
  - Individual work, independent submissions
- Project 2: 40%
  - Team work in groups of 2 or 3 students
- Seminar presentation: 40%
  - Selection from course catalog
  - Slides are reviewed by us
  - Submission until Monday evening

Team work

- Team Size for Project 1: 1
- Team Size for Project 2: 3
- Team Size for Seminar Presentation: 1

Registration Process

1. Find a group
2. Register your group: `mailto:sip-team@iaik.tugraz.at`
3. Wait for the confirmation mail to get your group number
4. Decide when to pick up the HW (IF01052, Mo-Fr 10:00-16:00)
5. Choose a seminar topic
6. Register for a seminar topic: `https://bit.ly/2Ys9Cvi`

Deadline: 12.10., 23:59

Schedule of SIP 2021

Project meetings

- Regular weekly meetings: Wednesday 10:00 - 12:00, online
  - Discord: `https://discord.gg/9KKGfndsD5`
  - Register in #getting-started to see channel
  - Questions: #sip
  - Finding team mates: #sip-groupsearch
  - Lecture: #sip-lecture
- Content
  - Student talks
  - Questions, problems and challenges about the current project

Timetable (Preliminary!)

Date	Topic
06.10.2021	Kick-off / Introduction to Seminar Topics / SoC Design Flow Tutorial
13.10.2021	Embedded Linux Tutorial / Introduction Project 1
20.10.2021	Debugging Tutorial / Seminar talks
27.10.2021	Seminar talks
03.11.2021	Seminar talks
10.11.2021	Introduction Project 2 / Seminar talks
17.11.2021	Seminar talks
24.11.2021	Seminar talks
01.12.2021	Seminar talks
15.12.2021	Seminar talks
12.01.2021	Seminar talks
19.01.2021	Seminar talks
26.01.2021	Seminar talks

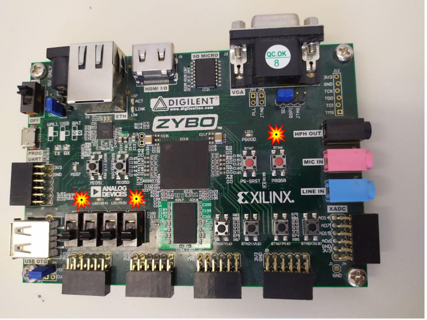
Important Dates and Deadlines (Preliminary!)

- 12.10., 23:59: Deadline Group Registration
- 09.11., 23:59: Deadline Project 1
- 10.11. - 17.11.: Interviews Project 1
- 09.12.,23:59: Deadline Project 2a
- ~~18.01.~~ 25.1., 23:59: Deadline Project 2b
- ~~19.01. - 26.01.~~ 26.1.-28.1.: Interviews Project 2a+2b

Outlook: Projects

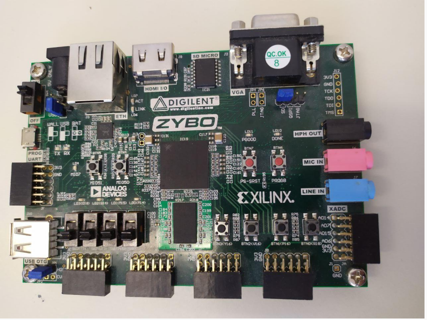
Project 1: Fancy Lights

- Get to know the board and run through all steps
- Design hardware, build a driver, write an application
- Access the LEDs from a bare-metal application and from within Linux
- No team work; everybody should do all steps (share your board within group)
- Aim: Everybody should have the same basic knowledge.



Project 2: Display for Encrypted Images

- Use knowledge from Project 1 to build larger system
- Receive encrypted image via Ethernet, decrypt it in hardware and display it via HDMI
- Team work
- Aim: Get some deeper understanding of the topic



Looking for a master project/thesis?

`https://www.iaik.tugraz.at/teaching/master-thesis/`