Digital System Design
Assignment Presentation

Robert Schilling, Pascal Nasahl

10.03.2020

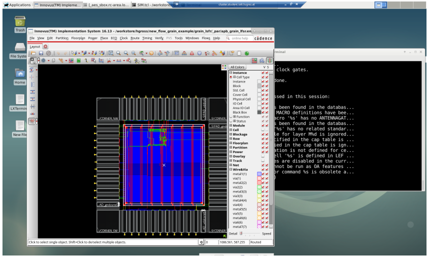
Digital System Design
Graz University of Technology

Reminder: Group Registration

- Groups
  - Form groups of **two** students
    - Register via mail to robert.schilling@iaik.tugraz.at and pascal.nasahl@iaik.tugraz.at
    - Subject: [DSD KU GroupReg] *Surname1 Surname2*
    - Body:
      Firstname1 Surname1 eMail(student.tugraz)
      Firstname2 Surname2 eMail(student.tugraz)
  - Sign “Cadence Software Usage Agreement” and send together with registration mail
  - No group? (1) Use newsgroup. (2) Talk to others. (3) Contact us!
  - Deadline: **Today!** (March 10)

Groups and Accounts

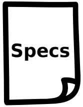
- Remote access
  - Via SSH
    - cluster.student.iaik.tugraz.at
    - Enable x11 forwarding for graphical output
  - Via Remote Desktop
    - cluster.student.iaik.tugraz.at:3389
    - MSTSC (Windows)
    - Rdesktop, Remmina (Linux)



```
 clock gates.

done.

ssed in this session:

s been found in the databas...
 MACRO definitions have bee...
acro '%s' has no ANTENNAGAT...
s been found in the databas...
'%s' has no related standar...
le for layer M%d is ignored...
cified in the cap table is ...
ied in the cap table is ign...
ation is not defined for ce...
ell '%s' is defined in LEF ...
es are disabled in the curr...
nnot be run as OA features ...
or command %s is obsolete a...
```

Goal of the Practicals

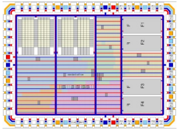
- Learn how to design digital hardware!



Starting with the specification

**Constraints, Algorithm Selection,
High-Level Model, Architecture, Partitioning,
Test and Verification, RTL Design, VHDL,
Synthesis, Place-and-Route, Layout**

To the final chip layout



Overview of the Assignment

3 Tasks:

1. Get in the flow (PRNG example)
2. Cipher implementation
3. Cipher integration

Grading

- Assignment 0: 0 points
  - Submission required!
- Assignment 1: 60 points
  - Tests (20)
  - Design document (5)
  - Design (35)
- Assignment 2: 40 points
  - Tests (5)
  - Design document (10)
  - Design (25)
- Deductions:
  - Coding Standard (-5)
  - Errors and Warnings (e.g. Latches) (-5)

Assignment 0

- Get in the flow - PRNG Example
  - Helps you to get to known our digital design flow
  - Per person **NOT** per group
  - Sign “Cadence Software Usage Agreement”
    - `https://www.iaik.tugraz.at/course/digital-system-design-705044-sommersemester-2020/`
    - send signed form to robert.schilling@iaik.tugraz.at and pascal.nasahl@iaik.tugraz.at
    - you get access to tools and your GIT group repo

Assignment 0

- “Get in the flow” Cont’d.
    - Repository contains:
        - Ibex processor
        - LFSR peripheral
        - LFSR module
        - Software
    - Goal: get random number in the software
    - Follow instructions in the repo README
    - Individual task, do not push solution!
    - Compress the cleaned directory and send it us per email
    - **Deadline: March 24**

Assignment 1

- Cipher implementation
  - Every group gets to design one symmetric primitive
    - e.g. Ascon, Keccak, . . .
  - Design only one variant
    - e.g. Ascon-128 not Ascon-128a
  - Encryption only (if applicable)
  - Submit solution using tag **ex1**
  - **Deadline: May 12**

Assignment 1

- First step: High-level modeling
  - Read the paper containing the cipher’s design description
  - Implement first rough model of the cipher
    - e.g. in C, Python, JAVA, . . .
    - you can use available reference implementations as basis and for comparison
    - better to start from scratch
  - After refinement in next step, create test vectors

Assignment 1

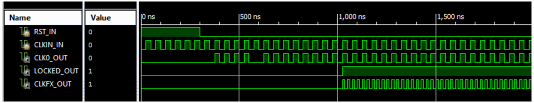
- Second step: Architecture design
  - Design the architecture of you cipher using pen & paper
    - keep bus interface in mind
    - split into modules and submodules
  - Refine your high-level model according to architecture design
    - e.g. to create test data for submodules (S-box, round transformation)
  - Before going to the next step, ensure that the overall system is understood well enough!

Assignment 1

- Third step: HDL implementation
  - In SystemVerilog, Verilog, or VHDL
  - Adopt introduced coding standard (part of grading!)
  - Best practice:
    - Stick to synthesizable constructs (see this lecture)
    - module wise implementation
    - continuous testing and verification against HL model

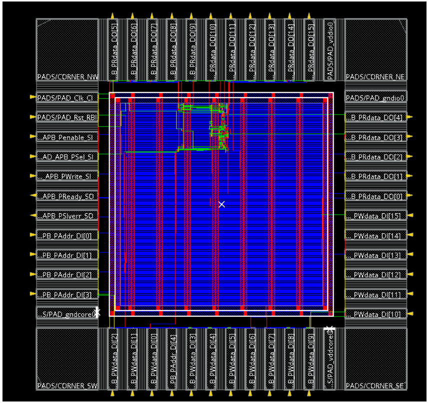
Assignment 1

- Fourth step: Synthesis
  - Automated process, but things can and will go wrong
    - check *.log files (in _synth) for errors and warnings
    - final design should not contain: “latch inference” or “signal not in sensitivity list” warnings or other errors (grading!)
  - Verify against high-level model
    - make sure to not have setup/hold-time violations or undefined signals
  - Extract area numbers and timing information from log files



Assignment 1

- Fifth step: Create layout data
  - Run place & route
  - Again verify against high-level module
  - Look at ”final” chip layout



Deliverables - Assignment 1

- Design document
    - Draft available at DSD course website
    - Short description of design
    - Goal: briefly document outcome of all tasks
    - Overview and description of architecture
    - High level model overview, how is the test data generated?
    - Hardware numbers for area, timing, max. throughput, power estimate after synthesis
    - Picture of chip layout after place & route
    - …
- Use tag **ex1** to submit

Assignment 2

- Cipher integration
  - Use your cipher implementation from **ex1**
  - Create a new Ibex peripheral
  - Make it accessibly by software
  - Submit solution using tag **ex2**
  - Deadline: **June 09**

Assignment 2

- Direct memory access
  - Design the new peripheral as a DMA device
  - Peripheral implements a slave and a master interface
  - Slave receives configuration command from CPU
  - Master fetches the plaintext from memory, encrypts it and stores the ciphertext to memory

Assignment 2

- Configuration command issued by software running on the CPU:
  - Start and end address of plaintext in memory
  - Start and end address of ciphertext in memory
  - Start signal
- Peripheral
  - Starts to encrypt data when receiving the start signal
  - Issues the end signal when finished
- Software
  - Polls the end signal

Grading Summary

- Design Document
- Hardware design(s)
  - functionality
  - high-level model
  - clean synthesis + place & route
  - coding style
- Colloquium in form of interviews at the end of semester
- Typically **one mark per group**

Schedule and Deadlines

Date	Action
Today	Assignment presentation + Ibex introduction
Next week (17.3)	HDL introduction + Design flow presentation
March 24	**Deadline assignment 0 (PRNG example)**
May 12	**Deadline assignment 1 (Cipher design)**
June 09	**Deadline assignment 2 (Cipher integration)**
10-12 June	Colloquium (interviews)
June 16	First exam

Next steps

- Sign Cadence Software Agreement
- Organize and register groups
  - You will then receive your accounts
- Make your self familiar with design flow in the first assignment
- Start as soon as possible with Assignment 1 (Cipher)

Digital System Design
Assignment Presentation

Robert Schilling, Pascal Nasahl

10.03.2020

Digital System Design
Graz University of Technology