Motivation

- **MOOC:** Massive Open Online Course
- Flexible learning: Learn anytime – anywhere
- Work with real hardware without needing to sit in a dedicated lab

Teaching material including FPGA setups
Earlier work:
Remote FPGA Lab: eDiViDe (2011-2014)

• European Digital Virtual Design Lab
• Developed a number of web-accessible digital design labs together with four European universities.
• Goal: Learn digital design for FPGA from anywhere through Internet
• Funding eDiViDe: ERASMUS Lifelong Learning Programme (LLP).

• Continued funding from Norgesuniversitetet (2014-2016)
Missing in eDiViDe – Now Realized in SPLUF

• **eDiViDe**: Only separate design assignments
• **SPLUF**: Start with simple designs that gradually becomes more complex
• **eDiViDe**: No teaching material prepared
• **SPLUF**: Slides, videos and quizzes provided
• **eDiViDe**: Xilinx ISE (old design tool)
• **SPLUF**: Xilinx Vivado with Xilinx Zynq®-7000
Two Courses

• **INF3430: Digital hardware system design**
  - *Introduction to digital design* including VHDL, FPGA and system-on-chip
  - *Bachelor level* course (Master-level option)

• **INF5430: Advanced digital systems design**
  - *Advanced digital hardware* system design, digital arithmetic, dynamic partial reconfiguration of FPGA, high level synthesis and functional verification
  - *Master level* course (PhD-level option)
edX MOOC platform

• **edX** is widely used internationally by leading universities. (Created by Massachusetts Institute of Technology and Harvard University in 2012.)

• an **open-source framework**, which one may host oneself.

• In contrast to using a regular course web page, edX includes functionality for quizzes and follow up of student assignments for the course manager.

• We have used the platform to give students in INF3430 access to **lecture slides, videos, quizzes and lab assignments**. Access here. [http://openedx-test.bibsys.no](http://openedx-test.bibsys.no) Select INF3430
About the course

Design of advanced digital systems. This includes programmable logic circuits, the design language VHDL and design of system-on-chip (processor, memory and logic on one chip). The labs gives practical experience on how to make a design.

What do you learn?

After completing this course you should be able to describe advanced digital systems on different levels of detail, explain important principals of design and testing of digital systems, understand the relation between behavior and different construction criterias and perform simulation and synthesis of digital systems.
How easy was it to start using the content of the edX-platform?
Quizzes

- Provide quizzes to most of the syllabus to improve the learning outcome when videos are watched or slides are studied on your own.
- Allow the students to become aware of their level of understanding.
- Taken from multiple choice questions in earlier regular course exams.
- The quizzes related to the textbook are used for preparation right before the exam (rather than during the semester).
Videos

• Have made videos to **some parts of the syllabus** (lecture slides provided for the rest).
• Videos have both been shown in regular lectures and been made available on the course website or in edX.
• Made the videos according to what we found were the guidelines many applied for making MOOC videos:
  – Keep the videos short and normally not longer than 10 minutes
  – The speaker should keep a proper speed
  – Picture of the speaker is not needed in the recording
What do you think about the length of the videos?

- Too short
- Could want longer
- Appropriate
- A bit long
- Much too long

Percentage

- INF5430
- INF3430
I have watched the videos

<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>In lectures only</td>
<td>INF5450: 10%, INF3430: 15%</td>
</tr>
<tr>
<td>On the web page only</td>
<td>INF5450: 35%, INF3430: 45%</td>
</tr>
<tr>
<td>In both lectures and on</td>
<td>INF5450: 55%, INF3430: 80%</td>
</tr>
</tbody>
</table>
How much did you feel you learned from the videos?

- None
- Some
- OK
- Quite much
- A lot

Percentage

INF5430
INF3430
Future preferences for videos

<table>
<thead>
<tr>
<th>No videos at all (only regular lectures)</th>
<th>In lectures only</th>
<th>On the web page only</th>
<th>In both lectures and on the web page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INF5430</td>
<td></td>
<td>INF3430</td>
<td></td>
</tr>
</tbody>
</table>
Interaction Between the Student and the Remote Lab

1. The student selects a lab assignment and gets files downloaded to own computer from edX.
2. The student implements the solution of the assignment locally using the Xilinx WebPack and verifies the design with simulation using the Modelsim PE Student Edition (or the Xilinx WebPack internal simulator).
3. The student logs in to the remote lab and files including the student solution is uploaded to the local server (tar.gz format).
4. The compilation is undertaken on the local server and a FPGA configuration file is generated.
5. If the design compiles without errors, a time slot is allocated for testing the design on the FPGA-board together with observing the performance through the camera.
Get Involved: Student

• Teach you how to do digital hardware design in an environment that offers real-life setups.
• Help you practicing your VHDL skills at any time anywhere.
• Offer a discussion forum for problems.
Get Involved: Lecturer

• Inspire your students by offering them real-life experiments by the available labs
• Include our setups to become a part of your digital design courses.
• Reduce the need for developing your own up-to-date setups and exercises.
Get Involved: Researcher

• Visualize your research by adding a setup.
• Demonstrate and test your novel designs on a readymade platform.
• Let you cooperate with your peers using a remote setup
Get Involved: Company

- Increase your competences in digital design.
- Include labs in your recruiting process.
- Promote your products by providing a remote lab access.
Remote lab - overview

- One central server
  - with web-interface
- Three local servers
  - connected through a local network to the central server
  - Web camera from each server to the FPGA-board
  - configuration-interface to the FPGA-board
  - USB interface to switchsim board
Remote lab - overview

• Web user interface
  – Implemented using Django
    • High-Level Python based web framework
  – File dialog
  – Output from Xilinx Vivado consol window
  – HTML5 video from a web-camera showing the lab HW setup
  – Simulated Switch and push buttons
Remote lab - overview

- Task control
  - Implemented using Celery
    - Distributed Task Queue (Python based)
- Each local server can handle up to ten background tasks simultaneously
  - ten Vivado user tasks
  - thus, 30 users can compile simultaneously
- One video task per local server
  - shows a hardware setup
  - based on time slots
  - round robin priority
- If one server is occupied then the video task is dispatched to the next free local server
Remote lab – HW overview

- USB switch simulator board
- 7-segment display board
- DC motor
- Motor control board w/power electronics
- Zedboard (Zync 7000)
- Double PMOD connectors 2*(VCC+GND+4 IO)
- Shaft encoder used for position measurements
- Shaft encoder used for position measurements
- Shaft encoder used for position measurements
- Shaft encoder used for position measurements
Remote lab – HW overview

- LEDs, 7-segments and Motor shaft movement observed through a web-camera.
  - Microsoft web-cam (Model1425)
- A mechanical frame was made to
  - stabilize the camera
  - anchor for and stabilize a light source
Remote Lab – The HW boards made USB Switch simulation board

- Was made to simulate the slide switches and push buttons on the Zedboard
- USB interface which communicate through Python's scripts (PY-serial)
- Small ARM micro controller for serial/parallel conversion
- Defined commands for bit-, nibble- and byte-level pattern generation
- LEDs so that the state of each switch-bit can be observed through the web-camera
- Interfaces to three double PMOD-connectors on the Zedboard => 24 bits available for input
Remote lab – The HW boards made 7-segment board

- The Zedboard has a built-in OLED display module
  - Too complicated to control at this knowledge level
  - We decided to make a 7-segment board because we had good experience from earlier courses
- Four 7-segments in one module with common segment signals a, b, c, d, e, f, g and dec
- Each segment has a common anode
- Interfaces to two double PMOD-connectors on the Zedboard => one free PMOD to interface the Motor control board
Live demonstration

- In lab3 we make a position controller (servo) for a robot arm
- It is a closed loop control with
  - Setpoint taken from the switch sim board
  - Measured position from a shaft encoder as pulses a and b which are 90 degrees out of phase with each other, and which the students convert to an up/down counter.
- The students implement all the elements in regulator as individual modules which are well simulated in the RTL-level
- The 7-segment module (made in lab2) is reused and used to show the set-point and the position
Lab3 – top-level block schematic
User Experiences with Remote Lab

The remote lab has not yet been much used since:

– Although positive attitude for using it, it was regarded as additional work to get into using it
– Good access to the physical lab
– Lab assistants supervising in the physical lab
  (an active discussion forum related to the remote lab could compensate some for that)
Lecture attendance with teaching material on the web

- Attend less lectures
- No change
- Attend more lectures
Flexible teaching: SPLUF (2014-2016)

• Studietilbud for Programmerbar Logikk-Utvikling på Fjernlab (SPLUF)

• Web: [http://www.mn.uio.no/ifi/forskning/prosjekter/spluf](http://www.mn.uio.no/ifi/forskning/prosjekter/spluf)
  – [http://openedx-test.bibsys.no](http://openedx-test.bibsys.no)

• Goal: Learn digital design for FPGA from anywhere through the Internet

• Get in contact and get involved:
  – Jørgen Norendal: e-mail: joen@ifi.uio.no
  – Jim Tørresen: e-mail: jimtoer@ifi.uio.no
    Mobile: 92846669