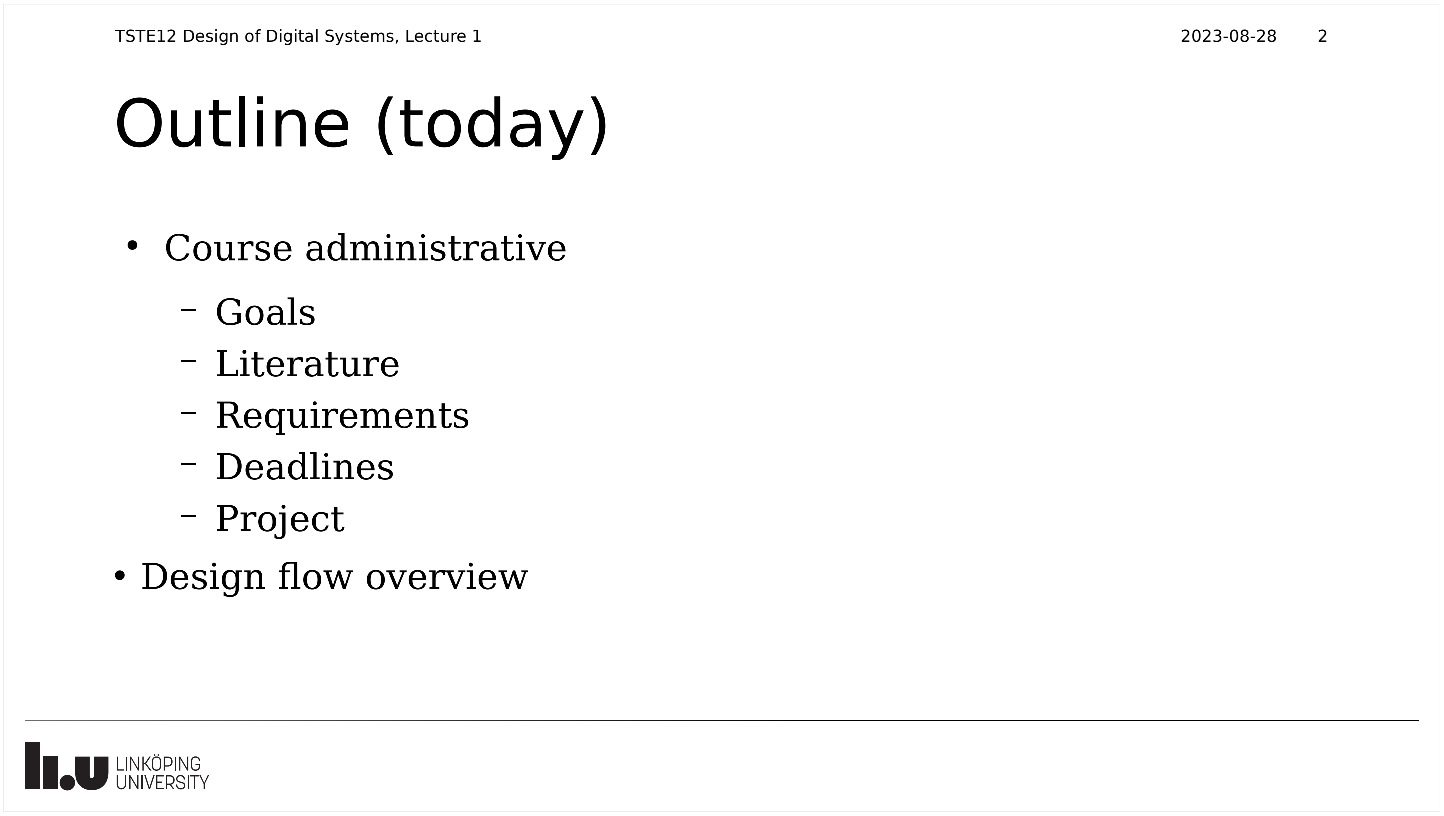
TSTE12 Design of Digital Systems

Kent Palmkvist



Outline (today)

- Course administrative
  - Goals
  - Literature
  - Requirements
  - Deadlines
  - Project
- Design flow overview

Course goals

- Structured design flow
- Design of larger digital systems
  - including fast prototyping using FPGA
- Learn VHDL (example of a Hardware Description Language)
- A project is included to learn and practice design
- Use of LIPS project model

Previous evaluation of the course

- Response rate 48% => Is it representative?
- Overall course grade: 4.56
  - Scale: 1:worst, 5:best
- Modifications from last year
  - No re-exam week => a little bit more calendar time for the project task for Y, D, and ED

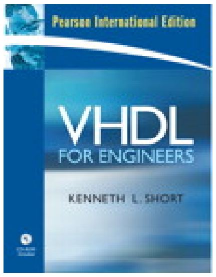
Course components

- 12 Lectures
- 3 handins
- 3 lab tasks
- 1 project task

- Web page: www.isy.liu.se/edu/kurs/TSTE12
- Lisam course room for handin/sign up etc.

Book suggestion (not mandatory to get)

- "VHDL for Engineers",
  Kenneth L. Short, Pearson Education
  - ISBN 9780135018101, 2008, 720 pages
- Most of the material can also be found by combining other sources
  - Alternative books
  - FPGA and ASIC vendor web sites
- Other web sites

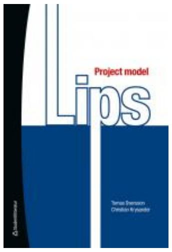


Example of alternative books

- “VHDL Designer”, Stefan Sjöholm, Lennart Lindh. Prentice Hall
  - ISBN: 0134734149; 1997, 496 pages
- “VHDL, Analysis and Modeling of Digital Systems”, Zainalabedin Navabi, McGraw-Hill
  - ISBN 0-07-046479-0
- “FPGA-Based System Design”, Wayne Wolf, Prentice Hall
  - ISBN 0-13-142461-012 Lectures

LIPS Project model

- “Project model LIPS”, T. Svensson, C. Krysander, Studentlitteratur
  - ISBN 9789144075266 (Out of print)
- “Projektmodellen LIPS”, T. Svensson, C. Krysander, Studentlitteratur
  - ISBN 9789144075259
- Describes the project model used in this and other CDIO courses.
- Also a website available: https://lips.isy.liu.se



Additional course material

- Lab Notes
  - Downloadable from the web page
    http://www.isy.liu.se/en/edu/kurs/TSTE12/laboration
- Tools tutorial
  - Downloadable from the web page
    http://www.isy.liu.se/en/edu/kurs/TSTE12/kursmaterial
- Plenty of additional web material on VHDL and logic synthesis

Lectures

- Introduce material
- Give an overview of language and usage of the language (not only for synthesis)
- Describe a design flow, from behavioral level down to layout
- Give hints
  - Hardware
  - Language usage
  - Tool usage
- Some material not in the book

Note about lecture 5

- Lecture 5 (Thursday 7/9) will discuss the LIPS project model
- Y and D program students may have seen this already
  - TSEA27 Elektronikprojekt Y
  - TSEA56 Elektronik kandidatprojekt
  - TSEA29 Konstruktion med mikrodator (KMM)
  - You are welcome to attend this anyway

Note about lectures 11-12

- Lecture 11 and 12 will discuss microprogramming and low level programming
- Y and D program students may have seen this already
  - TSEA83 Datorkonstruktion
  - TSEA28 Datorteknik Y
  - You are welcome anyway

Tutorial tasks

- Test language and simulation
  - Help understand VHDL
  - Make it possible to carry out hand-in tasks
- Tutorial material on the web
  - https://www.isy.liu.se/en/edu/kurs/TSTE12/kursmaterial
- Intended to be used as a startup task before handin and lab

Handin task (homework)

- Three separate hand-ins, each solved **individually**
- One week available to complete each one
  - v37 Monday 11 Sept. – v38 Monday 18 Sept.
  - v39 Monday 25 Sept. – v40 Monday 2 Oct.
  - v41 Monday 9 Oct. – v42 Monday 16 Oct.
- All info will be available on the web (homework)
- Theory and programming based on lectures and on book
  - Increasing difficulty level.
- 4th set v43 Monday 23 Oct. – v44 Monday 30 Oct.

Lab task 1-2

- Exercise use of design tools
  - Design flow used in project
- Lab1: PS/2 Keyboard decoder
- Lab2: VGA Graphic Controller
- Carried out in groups of 1-2 persons
  - Lab sign up on lisam pages
- Lab group sign up required to get access to CAD tools

Lab task 1-2, cont.

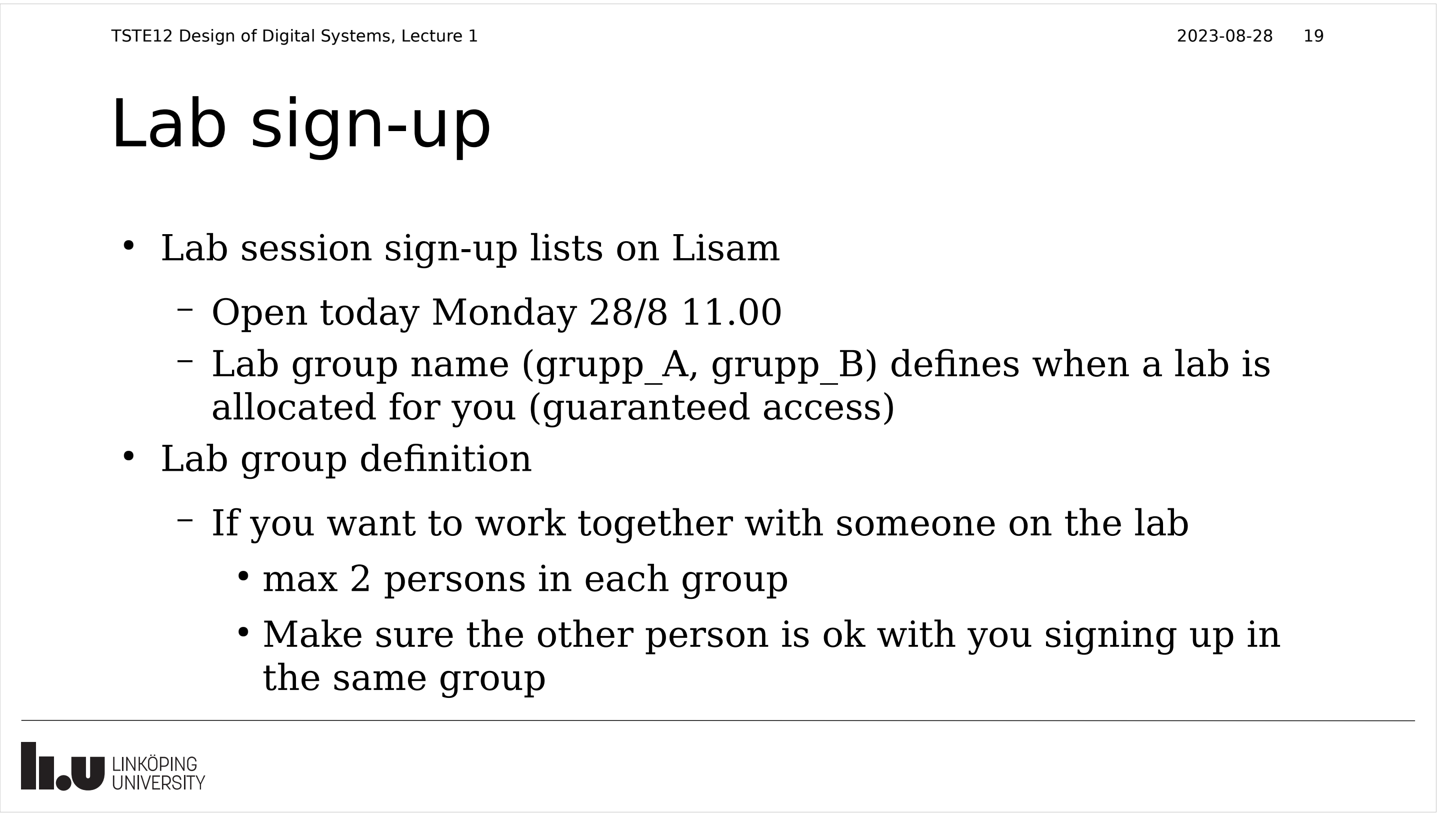
- Deadlines Y, D and ED
  - Lab 1: week 36, Wednesday 6 September, 21:00
  - Lab 2: week 37, Wednesday 13 September, 21:00
- Deadlines all other students
  - Lab 1: week 37, Wednesday 13 September, 21:00
  - Lab 2: week 38, Wednesday 20 September, 21:00
- Lab 1 deadline is a hard deadline
  - Project participation require lab 1 complete before deadline
- Lab 2 deadline is soft deadline, lab results will be checked after the project finished

Lab task 3

- Program an existing design
  - Microprogrammed VGA Controller
- Introduce low level programming
  - Microprogramming (lower abstraction than assembly language and C)
- Automatic pass if you got a pass on
  - TSEA28 Datorteknik Y
  - TSEA83 Datorkonstruktion
- Deadline lab 3: week 43, Friday 27 Sept.

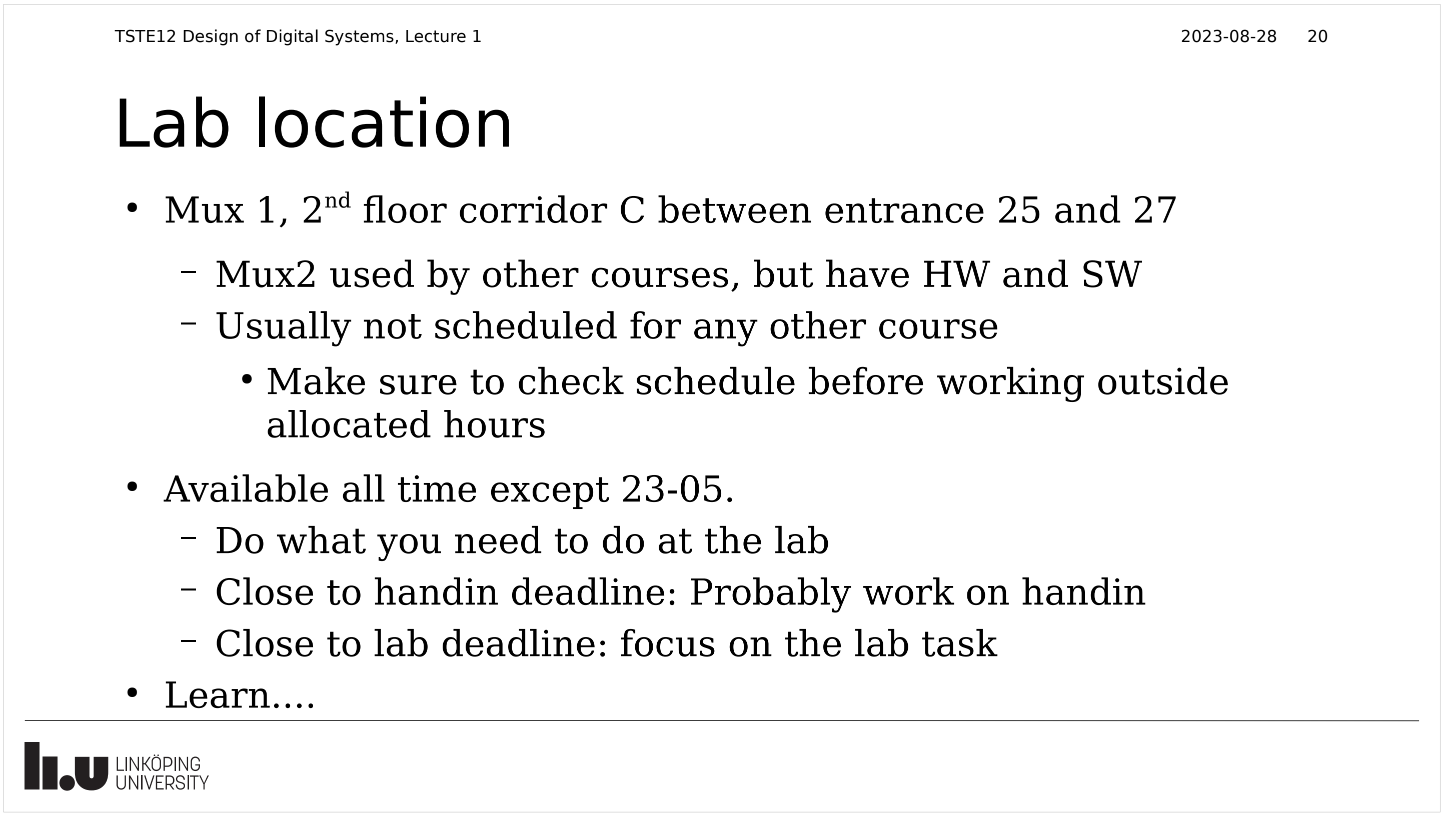
Lab tasks, cont.

- Labs group list
  - Must be decided at least one day before first exercise/lab to allow for group setup
  - Each group will have their own directory for the lab
  - If you want to work together (max 2 persons/group) sign up on lab list and send email to me
- You are encouraged to work on your own outside schedule! Use scheduled times for questions and demonstration of results.



Lab sign-up

- Lab session sign-up lists on Lisam
  - Open today Monday 28/8 11.00
  - Lab group name (grupp_A, grupp_B) defines when a lab is allocated for you (guaranteed access)
- Lab group definition
  - If you want to work together with someone on the lab
    - max 2 persons in each group
    - Make sure the other person is ok with you signing up in the same group



Lab location

- Mux 1, 2nd floor corridor C between entrance 25 and 27
  - Mux2 used by other courses, but have HW and SW
  - Usually not scheduled for any other course
    - Make sure to check schedule before working outside allocated hours
- Available all time except 23-05.
  - Do what you need to do at the lab
  - Close to handin deadline: Probably work on handin
  - Close to lab deadline: focus on the lab task
- Learn….

Make sure you are registered for the course

- Sign up on the lab list in lisam for access to lab and handin
- Email me (kent.palmkvist@liu.se) if you do NOT want to participate in the project.
- Non-registered participants can not get access to the tools and lab/project directories
- May take a day or two to setup login/tools

Project task

- Separate scheduled lab and project time: Use for whats needed at the moment: lab, handin, or project
- Project groups are determined by me!
  - Matlab script used
  - Make you meet other people
  - 5-6 persons / group
  - 4 credits (högskolepoäng) for project => project work ~ 80h per person (at least)!
- Few basic project tasks available
- Idea is to reduce tension in group

Project task, cont.

- Supervisors
  - Olov Andersson
  - Kent Palmkvist
- Meetings agreed on at least one day in advance
  - Sign up on paper outside the supervisors office (or whatever the supervisor specifies)
- Documents to discuss must be submitted at least 24 h before meeting!

Project task, cont.

- Weekly group meetings (without supervisor)
  - Report progress + spent time to supervisor by email (use template)
  - Must have at least one group meeting each week
  - Email must be sent no later than 2 days after the meeting
- You must keep track of the amount of time spent on the project (and on what)!
- Keep a personal diary where you note project work (hours spent + what type of work)

Project task, cont.

- First meeting of group
  - Meet all group members and supervisor
    - Deadline Y,D program students: Friday 1 Sept.
    - Deadline all other students: Friday 8 Sept.
- Different deadlines for Y, D and ED compared to other students
  - Y, D and ED have prior experience with project management, tools, and computer system

Project documentation

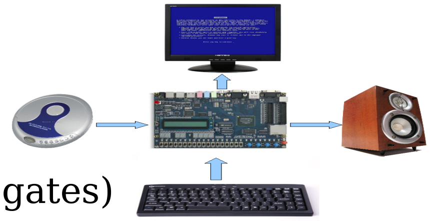
- Requirement specification (what, not how)
  - Deadline initial version (meeting with supervisor)
    Y, D, IEN: Tuesday 5 Sept.
    Others: Tuesday 12 Sept.
  - Final version Friday 8 Sept. / Friday 15 Sept.
- Project plan, Design specification
  - Deadline initial version (meeting with supervisor)
    Tuesday 12 Sept. / Tuesday 19 Sept.
  - Final version Friday 15 Sept. / Friday 22 Sept.
- Final Report
  - Deadline Friday 20 Oct. / Friday 27 Oct.

Project task, cont.

- All group members must attend the meetings where the documents are discussed
  - Meeting held together with the project supervisor
- Presentation
  - One oral presentation (ALL project members participate)
  - Include a demonstration of the project result

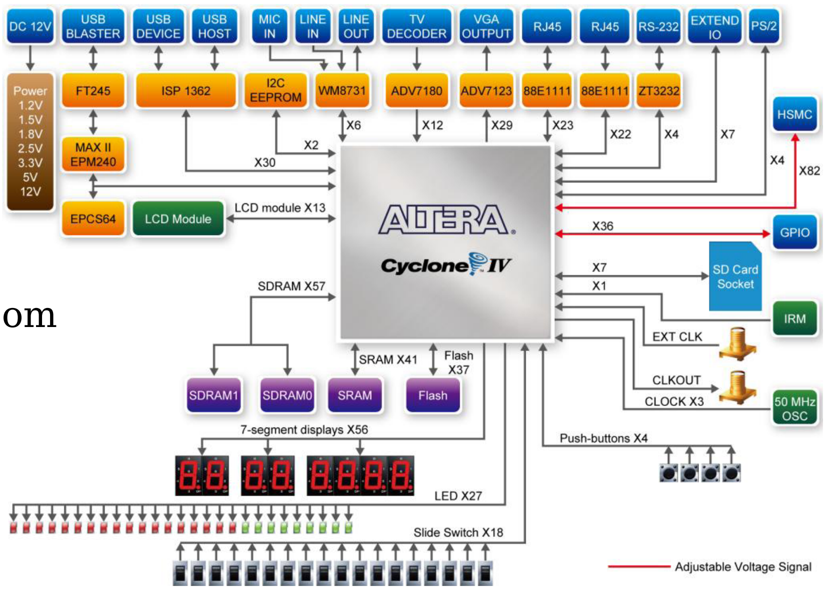
Project task description

- Sound, screen, keyboard
- Available hardware (DE2-115)
  - FPGA (> 3Mbit memory, 114 480 Logic element ~ 1 million logic gates)
  - Memory (8 MByte Flash, 2 MByte SRAM)
  - A/D, D/A (44 kHz 20 bit stereo)
  - Push-buttons, Switches
  - 7-segment LEDs
  - 16x2 LCD display, 24 bit VGA
- SD-card interface, IR remote control



FPGA board

- DE2-115
- FPGA + peripherals
- https://www.terasic.com



Project task definition

- Common requirements
  - Audio in and out (at least 20 bit, 44 kHz sample rate, stereo)
  - Volume and balance control (minimum 10 steps each)
  - Use VGA screen to indicate current settings/status
- Additional features possible
  - SD-card for audio recording/playback
- IR remote control

Project task definition, cont.

- Different major choices
  - Signal presentation (oscilloscope or signal level average)
  - Signal modification (echo, various sound effects, loudness control, or mono supression)
- Little more details available at the project web page of the course (including hardware descriptions)

Software to be used

- FPGA Advantage (Mentor Graphics)
  - Complete toolset for digital system development
    - HDL Designer design entry
    - Modelsim HDL Simulator
    - Precision HDL Synthesis
  - State of the art software (commercial high-end)
- Altera Quartus
  - Backend CAD tool for the used FPGA (used by Precision)
- Only available in labs Mux1 and Mux2

Software, cont.

- Openoffice/Libreoffice
  - Complete Microsoft Office replacement
    - Text editor
    - Drawing
    - Spreadsheet
    - Presentation
  - Free Software (download from www.openoffice.org and/or www.libreoffice.org)
  - Multi-platform (Unix, Windows, MacOS)
- Available in all computer labs at ISY

Introduction to the design flow

Background

Description

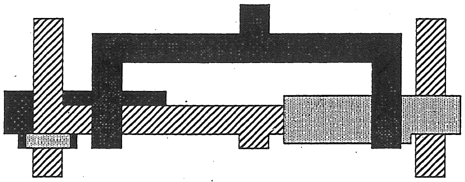
Design flow

System description

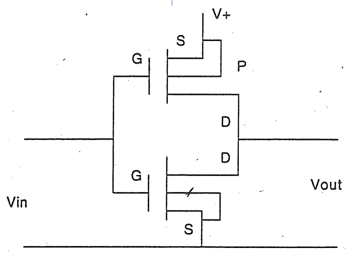
- Many possible systems (various application areas)
  - Electronics (TV, Computers, MP3 players)
  - Communication (Mobile phones, WLAN)
- Different levels
  - Baseband vs mobile phone vs cell vs country vs world
- Realtime vs non-realtime
  - Phone vs Internet
    - Subsystems in internet still real-time
- VGA control signals vs LED control signal

Description domains

- Physical (only material and geometry)
  - Inverter layout (does not define functionality on its own)
- Behavoural (black box)
  - Inverter behavoural (truth table)
- Functional (data flow)
  - Boolean expression
  - Y = not X
- Structural
  - Transistor schematic



X	Y
0	1
1	0



Abstraction levels

- Different levels of detail
  - System
  - Chip
  - Register
  - Logical
  - Transistor
  - Silicon
- All levels has both structural and behavoural descriptions

Abstraction transformations

- Synthesis: high abstraction level -> low abstraction level
  - Adds details
- Analysis
  - performance/property evaluation
- Optimization
  - not changing abstraction level
- Abstraction: low abstraction level -> high abstraction level
  - Hide details
- Validation/Verification
  - compare different abstraction levels

Design approaches

- Top-down
  - Start with high abstraction level, divide
- Bottom-up
  - Build more and more complex modules
- Meet-in-the-middle
  - Simultaneous bottom-up and top-down
- Edge-in
  - Follow data-flow from input and output

Representation formats

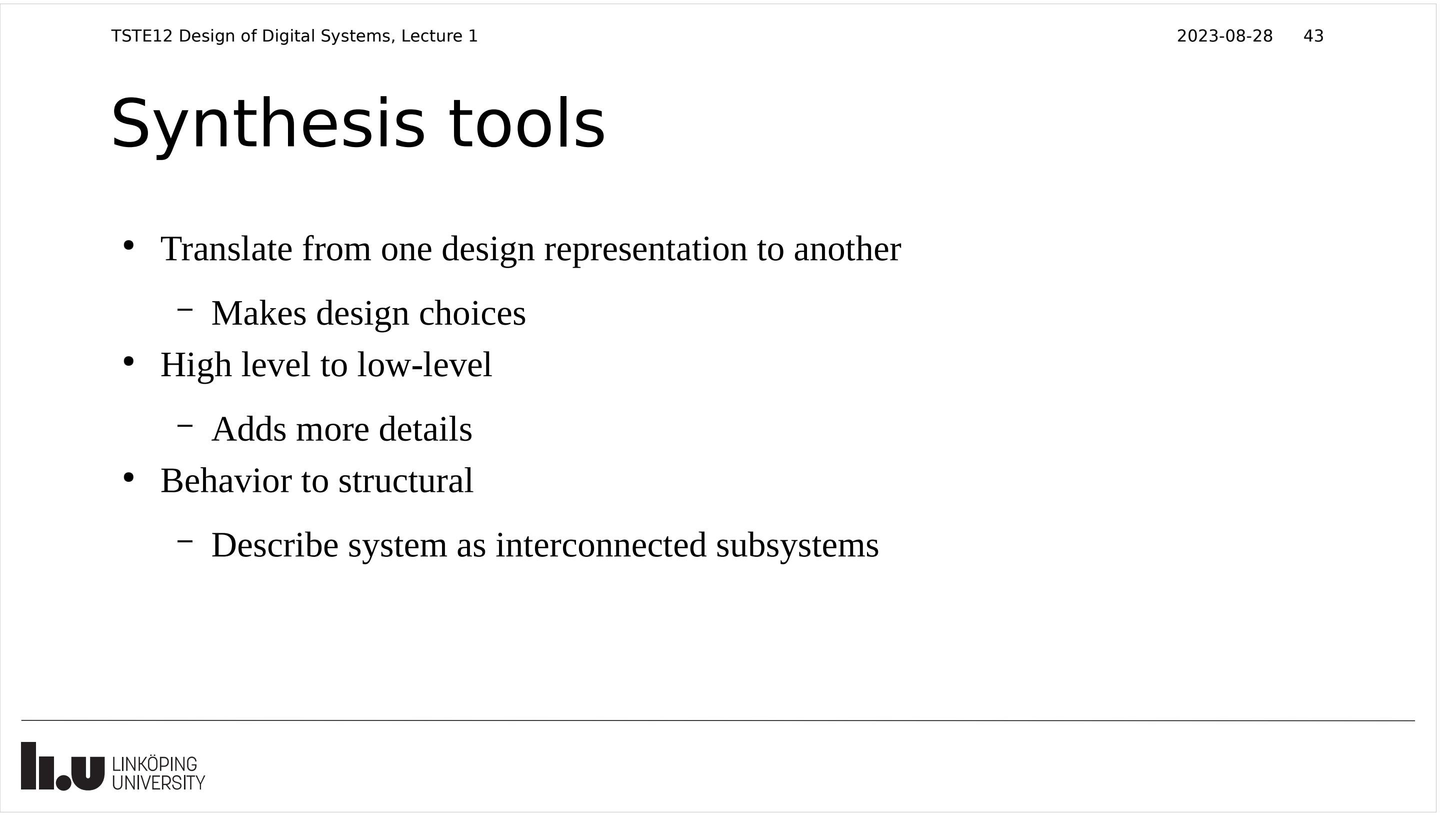
- Textual
  - Human language text
  - Mathematical equations
  - Computer languages
- Graphical
  - Block diagrams
  - Karnaugh maps
  - State diagrams

Text vs Graphical representation

- No representation is best in all situations
- Graphics
  - Interrelations
  - Dependencies
- Textual
  - Complex behavior
  - Control structures

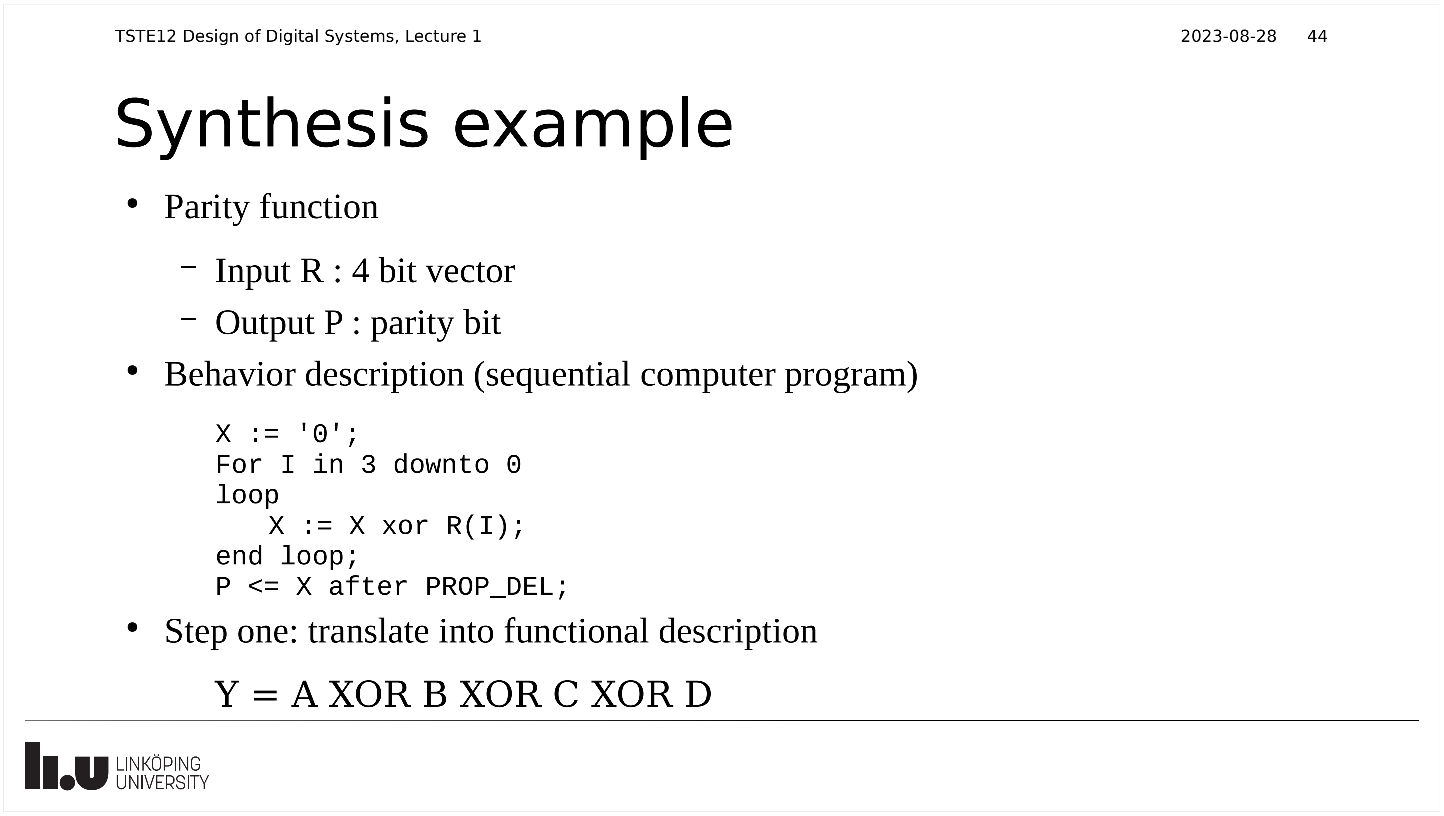
Design optimization

- Mandatory functionality
  - Functionality
  - Speed (clock cycle period)
- Cost
  - Power consumption
  - Size
  - Speed
- Design should fulfill mandatory requirements while optimizing for the cost



Synthesis tools

- Translate from one design representation to another
  - Makes design choices
- High level to low-level
  - Adds more details
- Behavior to structural
  - Describe system as interconnected subsystems



Synthesis example

- Parity function
  - Input R : 4 bit vector
  - Output P : parity bit
- Behavior description (sequential computer program)

```
X := '0';
For I in 3 downto 0
loop
   X := X xor R(I);
end loop;
P <= X after PROP_DEL;
```

- Step one: translate into functional description

Y = A XOR B XOR C XOR D

Example of synthesis and optimization

- Synthesis: Implemented using XOR gates

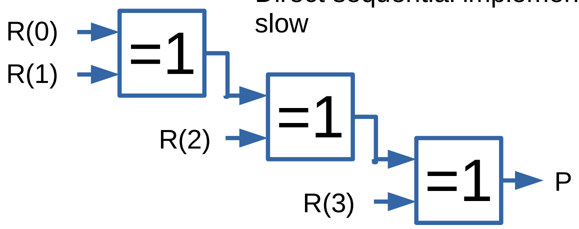
  Y = ((A XOR B) XOR C) XOR D

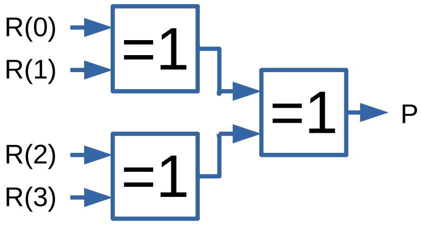
- Optimization: Rearrange gate structure

  Y = (A XOR B) XOR (C XOR D)

- Gives shorter critical path (2 gate delays instead of 3)

Direct sequential implementation slow





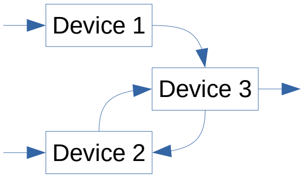
Optimized faster

Design tools

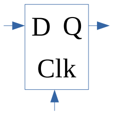
- Editors (design entry)
  - Text and graphics
- Simulators
  - Stochastic or deterministic
- Analyzers
  - DRC, Timing analyzers
- Synthesizers
- Optimizers

Simulators

- Important: Need to understand the behavior of the simulator to understand the description language
- Model hardware as set of hardware processes communicating through signals.
- Each process models function and delay of a subsystem

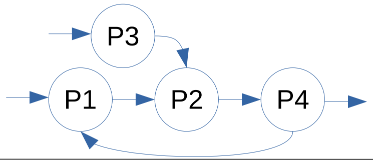


Network of devices modeled as a network of processes



Device modeled as a process

```
Flipflop:  Process(Clk)
begin
        if Clk = '1' and Clk'Event then
                Q <= D;
        end if;
end process;
```



Simulator, cont.

- Discrete event simulator (compared to a fixed time sampled)
  - No fixed time-step or maximal time step
  - Only execute models when activity on the signals
- Signals are sampled or triggering
  - Triggering signal starts processes
- Two types of signal update:
  - Transactions: a value is assigned to a signal
  - Events: new value different from current signal value
- All transactions are stored in a queue

Basic simulator execution

- Run processes with inputs that have changed
- Any signal that the process update is stored in a queue (sorted by update time)
- Once all processes run (that needed to be run) then increase simulation time and start over

Next lecture

- Practical issues
  - Lab setup, access
- VHDL intro
  - Hardware description language
- Syntax, details regarding the language
  - What can be described
  - How to describe
  - How it behaves (when simulated)

LINKÖPING
UNIVERSITY