LINKÖPING UNIVERSITY
Department of Electrical Engineering

# TSIU03, SYSTEM DESIGN

# LECTURE 1

Kent Palmkvist
Kent.Palmkvist@liu.se

Slides by: Mario Garrido Gálvez

Linköping, 2022



## TODAY

- Course description and organization:
  - What is the course about?
  - How is the course organized?
  - What do I have to do to pass?
  - …
- Introduction to hardware design.
- First designs in VHDL.

# PEOPLE

- Kent Palmkvist (Course responsible):Office 3B:502, kent.palmvist@liu.se
- Petter Källström: Office 3B:554, petter.kallstrom@liu.se
- Mikael Henriksson: Office 3B:528, mikael.henriksson@liu.se

Note: All Offices are at B-house, Campus Valla, Entrance B-25



# COURSE ORGANIZATION

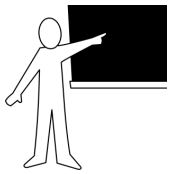
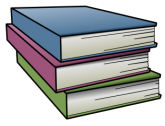
- Course Contents:
  - 10 Lectures (2 hours each).
  - 4 Laboratories.
  - 3 Assignments.
  - 1 Project (in groups)
- 8 ECTS credits:
  - LAB1 (3 ECTS): Laboratories.
  - PRA1 (5 ECTS): Assignments and Project.
- 213 scheduled hours: 52 scheduled + 161 self-study.
- Web page: http://www.isy.liu.se/edu/kurs/TSIU03/
- Lisam course room https://liuonline.sharepoint.com/sites/Lisam_TSIU03_2022HT_O9

# LAST YEAR COURSE EVALUATION

- Evaliuate result 2021:
  - 50% response rate (what did the rest of the students think?)
  - Average overall: 4.36 (1: worst, 5=best)
- Biggest change 2022 from 2021
  - All is now on site (no more remote teaching)



# COURSE OVERVIEW

Lectures
Books

Learn the Theory

Assignments

Apply the Theory

Labs 1, 2

Learn the Tools

Lab 3, 4

Apply the Tools and the Theory

# COURSE OVERVIEW (2)

Project Drive Alone

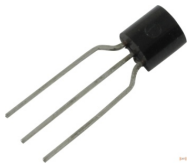


# BOOKS

- Andrew Rushton, “VHDL for Logic Synthesis”. John Wiley & Sons, 2011 (3rd edition).
- Peter J. Ashenden: “Digital Design: An Embedded Systems Approach Using VHDL” Morgan Kaufmann, 2007.
- Available:
  - 2 copies of each book at the library.
  - Online from the webpage of the library with unlimited access.
  - If you like any of them you can buy them at Bokab (in Kårallen).
- References to the books during the course:
  - [R] = Rushton. Example: [R2.3] = Rushton, chapter 2, section 3.
  - [A] = Ashenden.

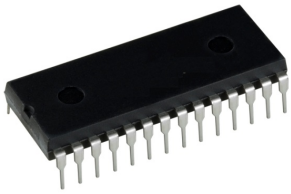
# INTEGRATED CIRCUITS

- An **integrated circuit (IC)** (also **chip** or **microchip**) is a set of electronic circuits on one small plate ("chip") of semiconductor material, normally silicon.
- **Very-large-scale integration (VLSI)** is the process of creating integrated circuits by combining thousands of transistors into a single chip.



Discrete component

(1 transistor)



Integrated Circuit

(millions of transistors)



# TYPES OF IC

- A **microprocessor** is an IC that incorporates the functions of a computer's central processing unit (CPU).
- A **microcontroller** is a small computer on a single integrated circuit (IC) that contains a CPU, memory, and programmable input/output peripherals.
- An **application-specific integrated circuit (ASIC)**, is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use.
- A **field-programmable gate array (FPGA)** is an IC designed to be configured by a hardware designer after manufacturing (this is why it is called "field-programmable“).

# ASICs & FPGAs

- Used for demanding digital signal processing applications (real-time, high throughput, low latency, low power consumption,…).
- We design the **hardware**.
- Hardware description languages: VHDL, Verilog.

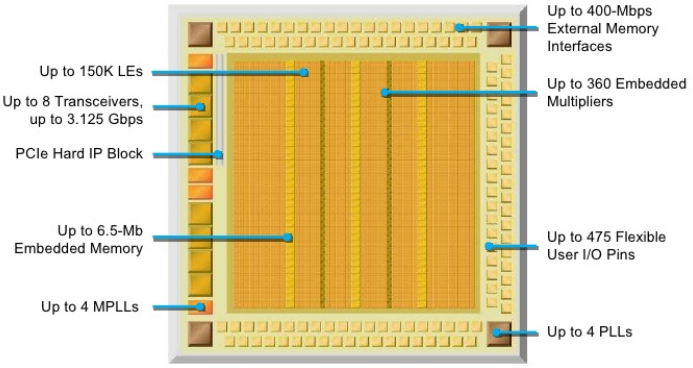
| | **ASICs** | **FPGAs** |
|---|---|---|
| Reprogrammable/ upgradable | No | Yes |
| Performance | Highest | High |
| Unit Cost | Very high (~ $10^6$) | Low (~ $100) |
| **Mass Production** | Low price per unit | **Does not reduce the prices per unit.** |

11

# FPGAs

- Configurable hardware. No software is running on an FPGA!!
- Main vendors: Intel (Altera) and Xilinx.

**Cyclone IV GX Key Architectural Features**





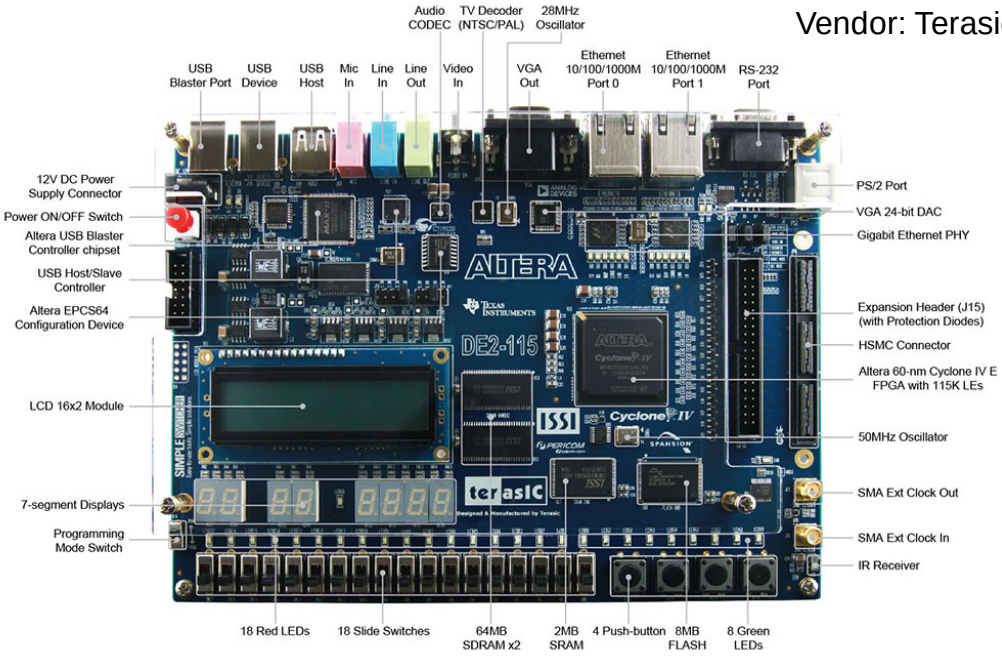
*MPLL – Multi-purpose phase-locked loop for transceivers*




12

# DE2-115 DEVELOPMENT BOARD

Vendor: Terasic




# LECTURES

- The course includes 10 lectures.
- The general approach of the course is to explain different types of circuits and show how to describe them in VHDL. The main topics that are covered in the course are:
  1. Combinational circuits.
  2. Sequential circuits.
  3. Debugging and test.
  4. Memories.
  5. State machines.
  6. Other important commands in VHDL.

# LABORATORIES

| Laboratory | Title | Preparation | Time | Deadline |
|---|---|---|---|---|
| Laboratory 1 | Introduction and Lab Tools | No | 4 h | Sep. 12th (*) |
| Laboratory 2 | Keyboard | Some | 6 h | Sep. 12th (*) |
| Laboratory 3 | VGA | Yes | 12 h | Sep. 28th |
| **Laboratory 4** | Audio CODEC | Yes | 8 h | **Sep. 28th** |

(*) Labs 1 and 2 must be passed in time in order to take part in the project.

- Lab room(s): MUXEN 4. Access at any time + meeting room.
- Extra Labs.



# ASSIGNMENTS

- The course includes 3 assignments to reinforce the knowledge from the lectures.
- **IMP: The assignments have to be solved and submitted individually. Submit answers using the Lisam course room submission function before the deadline.**

| Assignment | Corresponds to the theory in lecture | Deadline (lecture) |
|---|---|---|
| Assignment 1 | 1 | 4 |
| Assignment 2 | 2 and 3 | 6 |
| **Assignment 3** | 4, 5 and 6 | **8** |

- While you solve the assignment, you can ask any doubts to the teachers.
- The assignments will be discussed during the lecture when it is collected.
- After the grading of the assignments, students can come to the office to review it at the revision hours indicated in the assignment.
- Each assignments is graded as 100,90,80,... and the total grade is the average.

# PROJECT PART

- The students in groups of 6 people develop a complete system. The lab work can be integrated in the project.
- Project phases:
  - Definition of Requirements: Describe the functionality expected by the system. You can choose among different alternatives.
  - Hardware Design: Do the design of the system.
  - Implementation in VHDL: Describe the system in VHDL.
- Documents: Requirement Specification, Design Specification, Project Plan, Timing Reports, Project Report.
- Presentations:
  - First Presentation after the HW Design phase.
  - Final Presentation at the end.



# DEADLINES

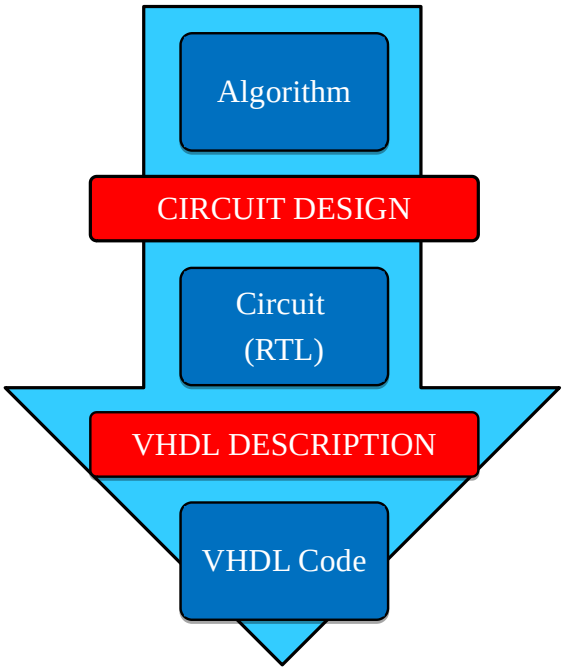
- The deadlines appear in the web page of the course.
- Any change of a deadline will be communicated to the students by mail (require that you are signed up to the course!).
- Assignments must be submitted in time to be graded.
- First deadline: Assignment 1, next Tuesday 8.15 at the beginning of the class (submitted into Lisam)
- Handin of assignments done using Lisam
- Suggestion: Write answers on paper, and then create a pdf-file using an app i a mobile phone.
  Example: Office Lens

# TO PASS

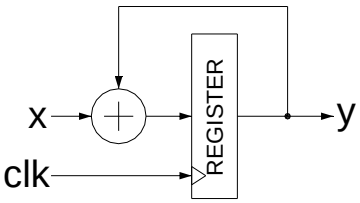
- LAB1 part:
  - To pass: Get a pass in all the Labs.
  - To get a pass in a Lab: Finish the task and show that you have learned and understood the Lab.
- PRA1 part:
  - To pass: Get a pass in the Assignments and in the Project.
  - Assignments: Solve the Assignments and submit them in time.
  - Finish the project so that it works according to the specifications.
  - Produce project documents with technical quality.
  - Do the presentations and the demonstration of the project.
- Final grades are given as 'Fail' or 'Pass‘.



# HARDWARE DESIGN PROCESS



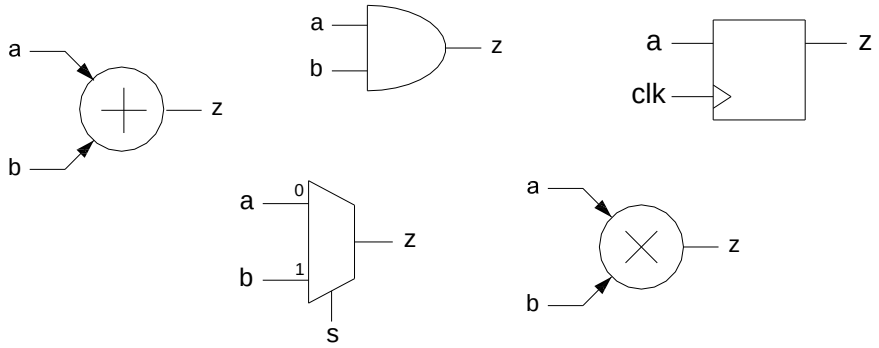

$$y = \sum_{n=0}^{N} x[n]$$




```
entity accumulator is
    port(x: in std_logic;
...
architecture acc of accumulator is
...
end acc;
```

# 1. CIRCUIT DESIGN

- Consists in combining hardware components in such a way that they fulfill the desired function. It is like making a puzzle. The components are like pieces of the puzzle.
- Typical hardware components are logic gates, multiplexers, adders, multipliers, registers, shift registers, memories, …

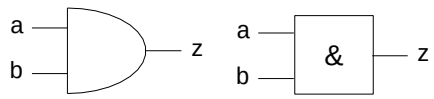





# LOGIC GATES

- Calculate a logic function.

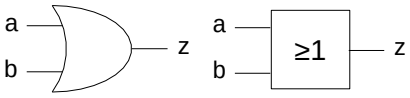
AND




$z = a \cdot b$

| a | b | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| **1** | 1 | **1** |

OR



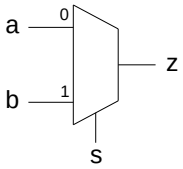

$z = a + b$

| a | b | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| **1** | 1 | **1** |

- To refresh Boolean Algebra and Logic Gates [A2.1]

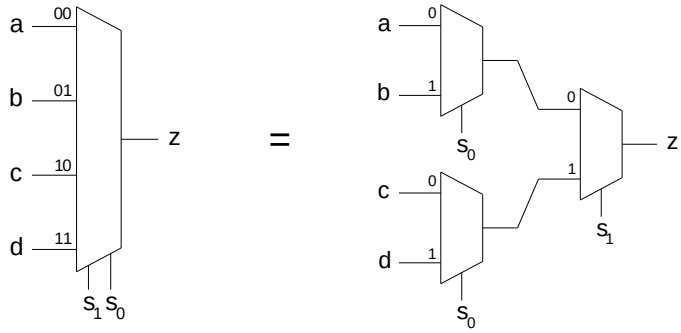
# MULTIPLEXER

- Allows to select between two inputs, a and b, depending on a control signal, s.




$$z = \begin{cases} a & if \quad s = 0 \\ b & if \quad s = 1 \end{cases}$$

- Multiple inputs and multiple select signals:






# CIRCUIT DESIGN EXAMPLE

- Now you know the components. By using only logic gates and multiplexer, design a circuit that calculates the following function. Draw the circuit.

$$z = \begin{cases} a \cdot b & if \quad s = 1 \\ 0 & if \quad s = 0 \end{cases}$$

# 2. VHDL DESCRIPTION

- VHDL = Very High Speed Integrated Circuit (VHSIC) Hardware Description Language.
- Just describes the circuit!

  - If you do not know how to describe it, either you have not done the circuit design yet (you are thinking about the mathematical function, but not about a circuit), or it is a new circuit for you and you need to learn how to describe it in VHDL.

- To describe a circuit in VHDL you have to describe:

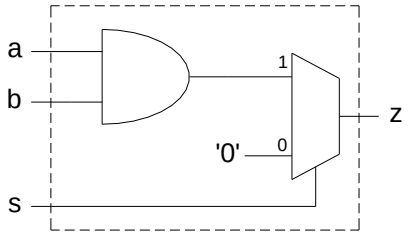
  - The external interface of the circuit (**ENTITY**).
  - The internal blocks (**ARCHITECTURE**).



# LET'S DESCRIBE A CIRCUIT!

- Taking the previous example:

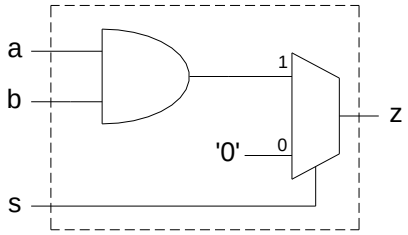
$$z = \begin{cases} a \cdot b & if \quad s = 1 \\ 0 & if \quad s = 0 \end{cases}$$



# LET'S DESCRIBE A CIRCUIT!

- Taking the previous example:

$$z = \begin{cases} a \cdot b & if \quad s = 1 \\ 0 & if \quad s = 0 \end{cases}$$




- Interface (ENTITY): a, b and s are input signals. All of them have one bit; z is an output signal that also has one bit.
- Internal blocks (ARCHITECTURE): a and b are connected to the inputs of an AND gate. Its output is connected to a multiplexer controlled by s and it is selected when s = '1'. When s='0', the output is '0'. The output of the multiplexer is connected to z.
- Learning VHDL is learning how to do this description in a way that the synthesizer understands!



# VHDL: ENTITY

- Describes the external interface of the circuit (no need to know how data is processed internally):

  - **PORT**: The ports are the input and output signals of the block. For each of them, we must specify:

    - Name of the signal.

    - If the signal is an input, output, etc.

    - Type of signal (we will use **std_logic** if the signal has 1 bit and **std_logic_vector, signed, unsigned**, etc if it has several bits).

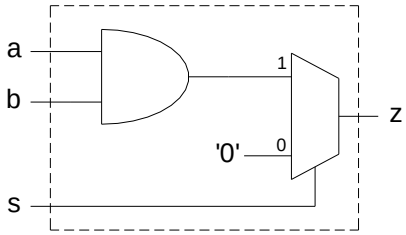
  - **GENERIC**:

    - Allows for defining parameters for the circuit.

    - It will be explained in the last lectures.

# EXAMPLE OF ENTITY

```
entity SimpleCircuit is
   port (a,b,s: in std_logic;
         z : out std_logic);
end SimpleCircuit;
```




- Name of the circuit: SimpleCircuit
- **entity**: to specify that we are describing the interface.
- **ports** of the circuit: a, b, s and z.
- a, b, s and z have only one bit (type **std_logic**).
- a, b and s are inputs (**in**) and z is an output (**out**).



# VHDL: ARCHITECTURE

- Describes the internal blocks of the circuit and how the signals are connected.
- First part of the architecture (declaration part):
  - **SIGNAL** declarations (to define internal signals).
  - **CONSTANT** declarations (for constant values).
- **BEGIN**: to separate the two parts of the architecture.
- Second part (description of the circuit): We will study how to describe different circuits during the course. For the circuit of the example we need to know **<=** (assignment of signals) and **WHEN…ELSE**.

# EXAMPLE OF ARCHITECTURE

```
architecture arch of SimpleCircuit is
   constant logicZero: std_logic:= '0';
   signal p: std_logic;
begin
   p <= a AND b;
   z <= p WHEN s = '1' ELSE logicZero;
end arch;
```

- Name of the architecture: arch
- **constant:** logicZero is a constant whose value is a logic 0.
- p: internal signal to define the connection between the AND gate and the multiplexer.



# COMPLETE DESCRIPTION IN VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity SimpleCircuit is
   port (a,b,s: in std_logic;
         z : out std_logic);
end SimpleCircuit;
architecture arch of SimpleCircuit is
begin
   z <= a AND b WHEN s = '1' ELSE '0';
end arch;
```

- We need to add the package **std_logic_1164** from the library **ieee.** It is the most basic package that includes the type **std_logic**.

# CHECKLIST FOR LECTURE 1

- Course description.
- Hardware design process: circuit design + VHDL description.
- Combinational circuits: logic gates and multiplexers.
- VHDL language: entity, port, in, out, architecture, signal, constant, begin, library, package std_logic_1164, std_logic, AND, OR, <=, when..else.



# AT HOME

- Register for the course if you have not done it yet.
- Visit the web page of the course:

  http://www.isy.liu.se/edu/kurs/TSIU03/
- If you have not done it yet, read carefully the Course Description document that you can find on the web page.
- Review the checklist for lecture 1 and check that you understand all the concepts and you know how to use them.
- Do the Assignment 1. It has to be submitted before the beginning of lecture 4!!