Exercise 3

7-segment display and FSM

Objectives

- 7-segment digit encoding and FSM-based selection of 4 digit 7-segment display
- Counter/Timer/IO-based control of the output
- Functional test on an FPGA (Device **XC7A35TCPG236-1**)
- Performance Analysis

General Description

In this exercise, the 7-segment display of the Basys 3 board will be put into operation by a VHDL-based implementation of an FSM and some additional functionality.

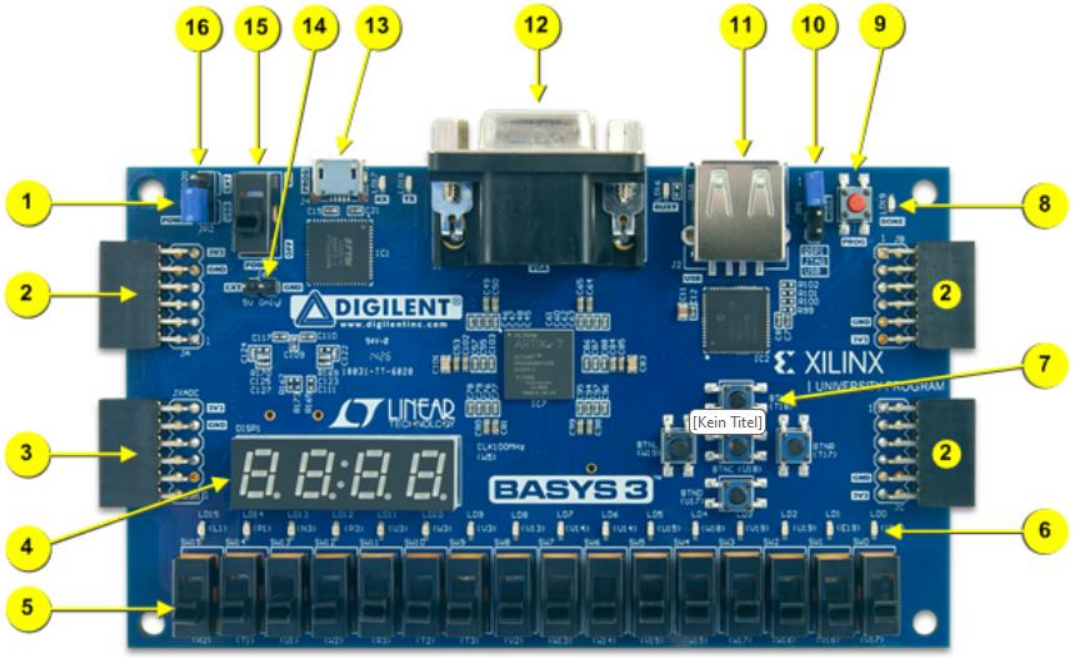
Remarks:

- This exercise is conducted as "Come-As-You-Are". This means a general overview of the hardware architecture, the setup of the experiment and an initial preparation task are described in this document. The task descriptions of this exercise will be provided to you at the lab session.
- Each team has to work all tasks on its own and manually implement the VHDL code. The use of code generators, i.e. generative AI tools, is strictly prohibited. Any misconduct will result in a score of 0 points.
- After the session, each group has to write and submit a lab report (see `General Advises and Rules` in TEAMs) as pdf via TEAMs no later than one week after the session: Deadline: Thursday 11.59 pm (hard deadline, no extensions possible). If no report has been submitted, this will result in a score of 0 points for the exercise and team.

The Basys 3 board

The Basys 3 board is a simple development platform for testing digital circuit designs. It provides a range of peripheral connectors (e.g. USB, VGA) as well as a large set of switches, LEDs, and other, I/O devices and pins. Core of this board is an Artix-7 FPGA (**XC7A35T-1CPG236C**), that can control all the components mentioned by operating hardware architectures on its fabric, commonly implemented in VHDL and synthesized by exploiting the Vivado Designtools (Synthesis, Implementation, Generate Bitstream and Program Device).

A more comprehensive overview of the Basys 3 board functionallity, the configuration possibilities and the use cases can be found in the reference manual (`basys3_reference_manual.pdf`) e.g. given in TEAMs



Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod port(s)	10	Programming mode jumper
3	Analog signal Pmod port (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Fig. 1: Basys 3 board and peripherals

The 7-segment display

One component of the Basys 3 board is the 7-segment display (see Fig 1, No. 4). In general, each digit of a 7-segment component consists of 7 LEDs segments (CA-CG) that form the pattern to be displayed (and an additional LED for the decimal point (DP)). Each LED can be controlled via an output pin of the FPGA (see Fig. 2). As shown in Fig. 1, the Basys 3 board possesses 4 7-segment blocks, which will be called (decimal) digits in the following. To keep the overall number of pins that control the 4 digits reasonable, one output pin can control the same LED of all 4 digits. In order to select between the different digits a one-hot encoded multiplexing scheme is provided by the Basys 3 board (AN0-AN3). A graphical overview of the 7-segment display, its digits and LEDs as well as the corresponding FPGA pins is given in Fig. 2.

Note, that both the LED and digit control (`DIGIT_select`) are **low-active**. Hence, switching an LED on or off requires the related FPGA-pin to be driven with a 0 and 1, respectively. For instance, setting the pins related to CA-CG to ‘0’ as well as AN1/AN2 to ’0’ and AN0/AN3 to ‘1’ will cause the two digits in the middle to display an eight (all LEDs and the decimal point (DP) are switched on).

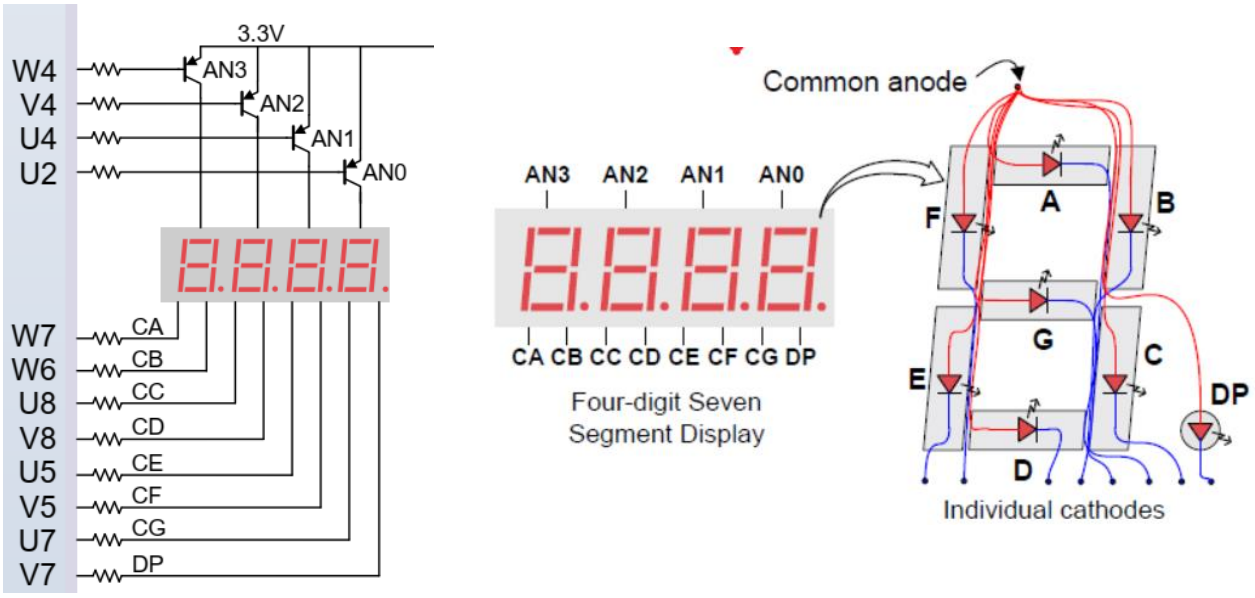


Fig. 2: Pin mapping and electrical setup of the 7-segment display of the Basys 3 board.

Things you should know when attending the lab exercise:

- Design-Flow to map VHDL code onto the FPGA
- Rudimentary debugging when synthesis/implementation fails
- How to read, set up and modify constraints
- How to implement in VHDL
  - Register
  - Counter/Basic Arithmetic
  - State machines
  - Combinatorial/sequential processes
  - Multiplexer
  - Extraction of subvectors and single bits
  - …

Preparation

P1 – LED signal assignment

- Familiarize yourself with the VHDL template `seven_segment.vhd`
- What are the output ports `LEDs`, `DIGIT_select` and `SW` used for?
- Implement the bit patterns in for the ZERO – NINE constants (line 45 - 54) so that it can be used to display decimal numbers on the 7-segment display (see Fig. 3).
  Hints:
  - You can investigate the constraint file (do not modify it!) and Fig. 2 get to know the assignment between the VHDL output ports and the FPGA pins.
  - The decimal point (DP) can be neglected (e.g. set to `'1'`)

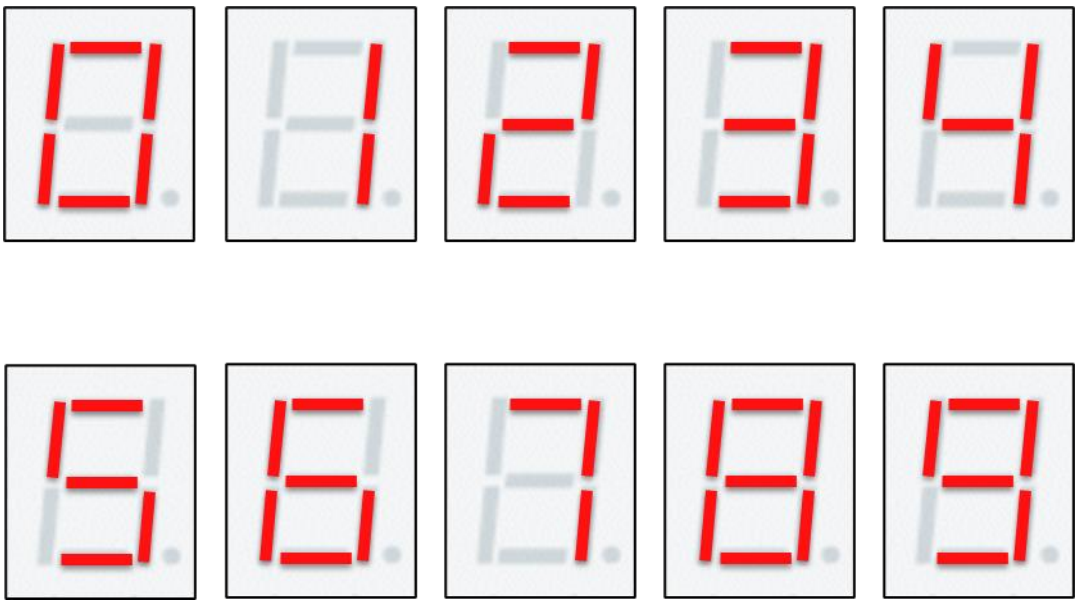


Fig. 3: LED configuration of the numbers ZERO to NINE. The decimal point (DP) can be neglected (e.g. set to `'1'`)

Tasks

T0 – Create new project

- Open Vivado and create a new project for the Basys 3 board (Device **XC7A35TCPG236-1**)
- Copy the source files with your prepared bit patterns to your project

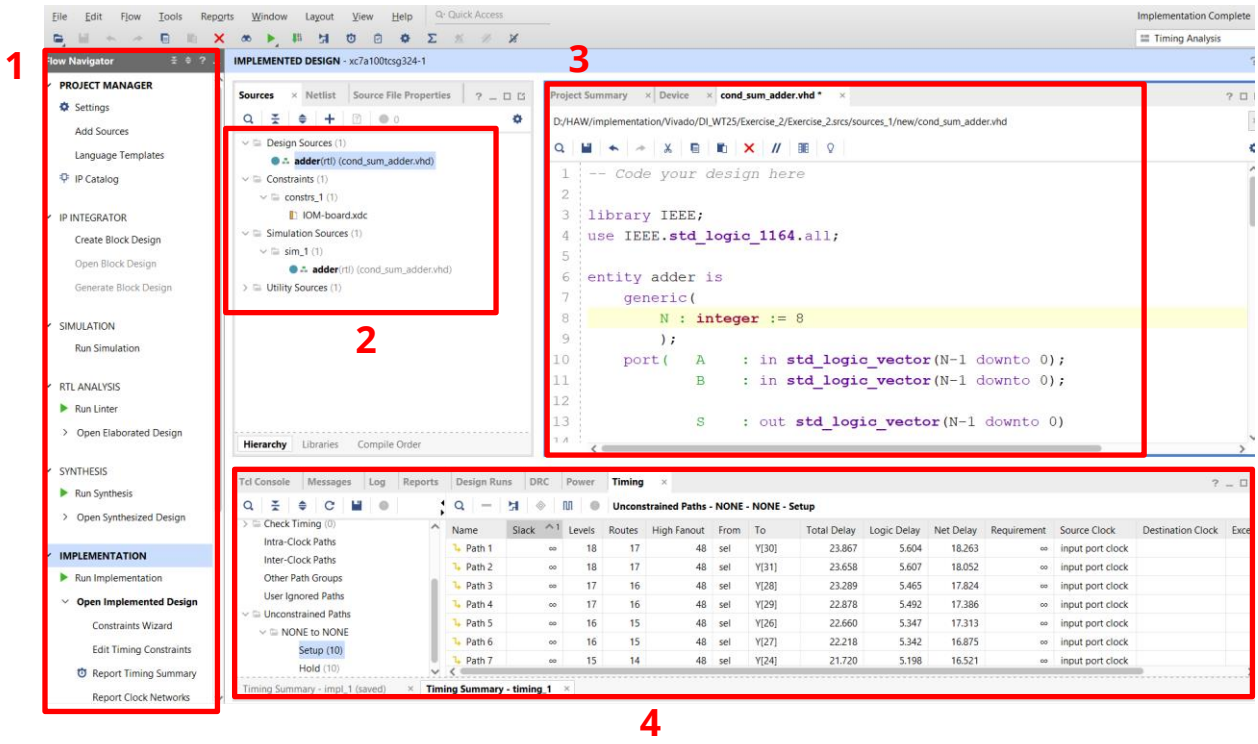
T1 – T5

- Will be provided to you during the lab session

T6 – Timing Analysis (Implementation)

- Open the results of the physical synthesis, (Open Implemented Design -> Report Timing Analysis)
- Inspect the timing and reflect the results in the report.

Appendix –Overview Vivado



1. Flow navigator:
    a. For starting the synthesis steps, bitstream generation, etc.
    b. For opening designs and analyse timing
2. Design sources
    a. Design, cornstraints, simulation/testbench
3. Editor
    a. VHDL files, summary, etc
4. Console
    a. Design flow outputs (Logs)
    b. Design analysis outputs: Timing, Power, etc.