Read this!
All of this!
Carefully!

4 hours preparation time!

Digital Circuits

Lab session on Arithmetic and Counting

Adder

Counter

Prof. Dr.-Ing. Peter Schulz

Revision 1.3, May 25th, 2024

General Advices

- General information:
  - You are only allowed to take part in the laboratory if you have taken part in the safety instructions session and have signed on the list.
  - The laboratory instructions must be read and understood completely.
  - All preparatory tasks must be completed and documented and brought with you to the laboratory appointment.
- Colour-Scheme in this document
  - Requirements
  - Preparation
  - Lab execution
  - Report
- Execution of the laboratory:
  - Be on time.
    - The laboratory is a compulsory course.
    - You may miss important announcements and information.
    - It is also unfair to your team colleagues, because you also must have the preparation tasks checked together with your colleagues. Then the start for the entire team would be delayed.
  - Electronic experimental setups and measuring devices can easily be damaged by incorrect operation. Therefore, the following rules of conduct must be strictly adhered to:
    - Only connect the intended power supply to your experimental setup.
    - Do not touch the electrical connectors of the components used with your fingers.
    - Do not connect logical outputs to one another (short circuit), either by wire or through incorrect entries in constraints files.
  - Record all relevant steps and results, either on paper or by saving the appropriate files on your USB memory stick.

General Advices

- Report
  - The entire team must prepare a report. It is a community effort.
  - The report must be printed out within one week and placed in Prof. Schulz's mailbox (ground floor).
  - Report structure and content:
    - Introduction to the objective
      (with your own words, a copy of the lab task description wouldn't be acceptable)
    - A summary of the preparatory tasks and their results.
    - Description of the laboratory setup
    - Presentation and discussion of the results. Source code and circuit diagrams should only be shown in excerpts if they are necessary for explanation. Complete source code and large drawings hinder the flow of reading and belong in the appendix..
    - Summary of key findings at the end of the report.
    - List of literature and sources
    - Appendix with source codes and other documents that do not fit into the main body of the text.
  - References and Plagiarism
    - Any reference to external sources, be it texts, images or source code, must be correctly referenced. To do this, the generally recognized IEEE guidelines must be adhered to.
    - Unmarked quotations (including image quotations) will result in your report being classified as plagiarism.
    - Plagiarism will result in failure of the laboratory (for the entire semester and for the entire team!)

Part 1
Adder Circuit

Objectives

- Design an Adder circuit.
- Implement the circuit using FPGA.

Part 1

Adder

Requirements list

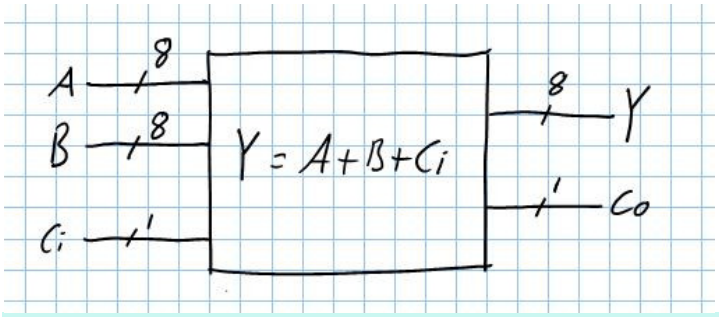


Fig. 1: Task 1.1

- For Task 1.1 (combinational adder)
  - The combinational adder circuit performs Y = A + B + Ci
    - A, B, Y are 8-bit vectors
  - There is a Carry Input signal Ci.
  - There is a Carry Output signal Co.
- For Task 1.2 (registered adder)
  - The Adder from Task 1.1 is extended by a result register.
  - The result register holds 8 bit of data.
  - The result register stores Y and Co on rising edge of clock.
  - The result register has a synchronous and LOW-active reset input signal RESETN.
  - The result register has a HIGH-active clock enable signal EN.

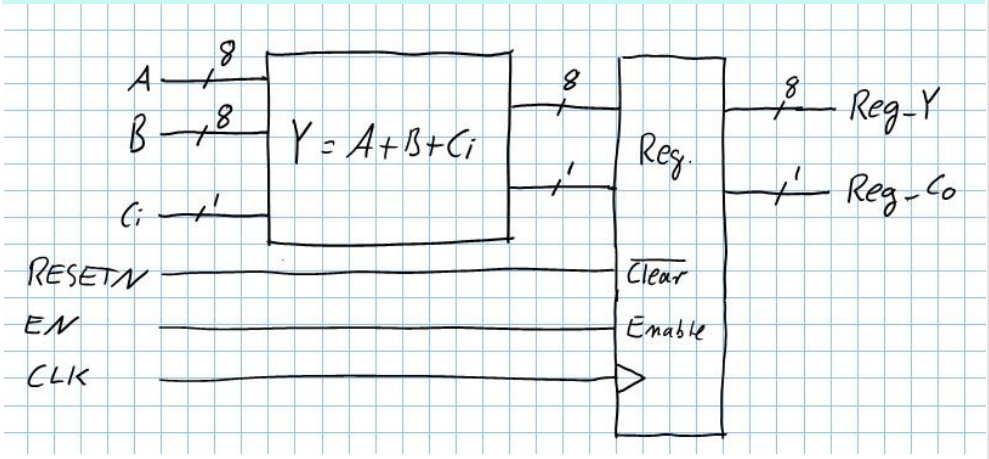


Fig. 2: Task 1.2

Part 1

Adder

Preparation

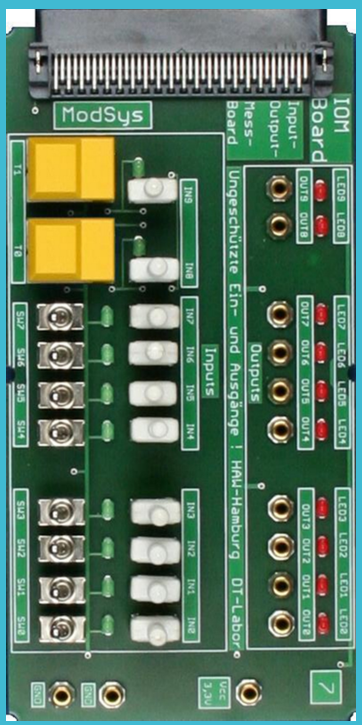


Fig. 3: IOM board

- Preparation Task 1.1 (for the combinational adder)
  - open this EDAplayground link:
    https://www.edaplayground.com/x/rHKX
    and make a copy of it:
  - Inspect Design and Testbench code (and simulate it)
  - Understand the concept of the stimuli process in conjunction with the TEST_DATA record.
  - Complement the TEST_DATA content by at least 20 meaningful test vectors (TODO-area in testbench.vhd) and document your simulation results. Every input bit and every carry chain connection must be triggered at least one time.
    - Each test case must have a unique identifier/number (like R1.1, R1.2, R2.1 etc. where the major number denotes a test category and the minor number several test vectors in that category.
      Use a test-plan table as shown on the next page.
- Preparation Task 1.2 (for the registered adder)
  - open this EDAplayground link:
    https://www.edaplayground.com/x/NmDe
    and make a copy of it:
  - Inspect Design and Testbench code
    - What is the difference in Test-Data timing compared to the combinational adder?
  - Complement the TEST_DATA content by the same test vectors already used for the combinational adder.
  - Check the source code for other TODO-entries and perform the required action
- Preparation Task 1.3
  - I/O specifications (2 IOM-Boards required)
    - Have the following xdc-files from the Moodle course with you on a USB memory stick:
      - Adder_IOM_Conn_3.xdc, Adder_IOM_Conn_4.xdc, Modsys_Artix7_Main_System.xdc
    - Please familiarize yourself with using the board resources.
      - The switches are for data
        (one IOM for Operand A, another one for Operand B)
      - The keys (shown in yellow here) are for carry input and clock enable.
      - Clock and RESETN are taken from the main board. (main xdc!)
      - The IOM board which is input for Operand A also uses the LED-outputs for indicating sum and carry_out .

Part 1

Adder

Test Plan

Print out this plan, fill in all test cases and bring it to the lab. During lab check all tests. The original, checked test plan must be added to your report.

Req. ID	Test description	stimuli operand A	stimuli operand B	stimuli carry in	expected response Y	expected response carry out	Testbench Simulation (check)	FPGA circuit (check)
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐
							☐	☐

Part 1

Adder

Execution in the Laboratory

- Lab Task 1.1 (combinational adder)
  - Create a new RTL project in Vivado.
  - Add the design VHDL file and the xdc-files for the combinational adder.
  - Perform the Implementation Design Flow. (no simulation, just generate Bitstream)
  - Inspect schematic files if they show reasonable results.
  - Test the adder by applying your test vectors through the IOM-Boards.
  - Measure propagation delay from carry input to carry output with the oscilloscope.
    (test vector must be conditioned for this special case)
- Lab Task 1.2 (registered adder)
  - Create a new RTL project in Vivado.
  - Add the design VHDL file and the xdc-files for the registered adder.
  - Perform the Implementation Design Flow. (no simulation, just generate Bitstream)
  - Inspect schematic files if they show reasonable results.
  - Test the adder by applying your test vectors. Use clock in manual mode.
  - Switch the clock from manual to intern mode and set it to 1 MHz.
  - Measure propagation delay from carry input to carry output with the oscilloscope.
    (test vector must be conditioned for this special case)
    - Is there a difference to the propagation delay of Lab Task 1.1?
    - If yes, explain the difference!
- Lab Task 1.3 (Accumulator, Fig. 4)
  - Detach the switches from the FPGA inputs on the IOM board which also uses the LED outputs (white bridging connectors).
  - Feedback the output vector to the input vector using lab cables. Preserve the bit ordering.
  - Apply the following number sequence: 3, 5, 9, 23 (manual clocking)
    - What is the expected result?
    - Compare the accumulated value with the expected result.
- **Don't disassemble your setup. You need it for Part 2 as well.**

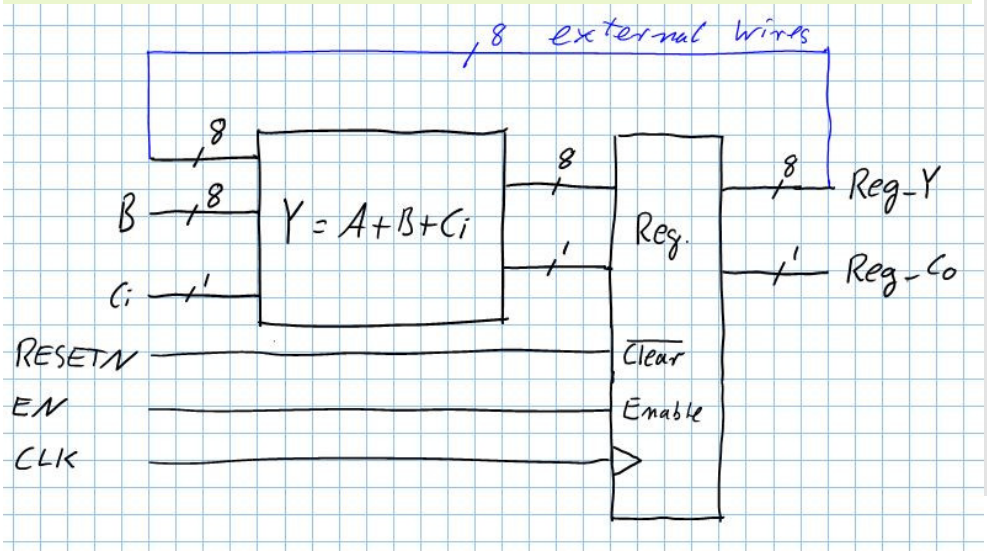
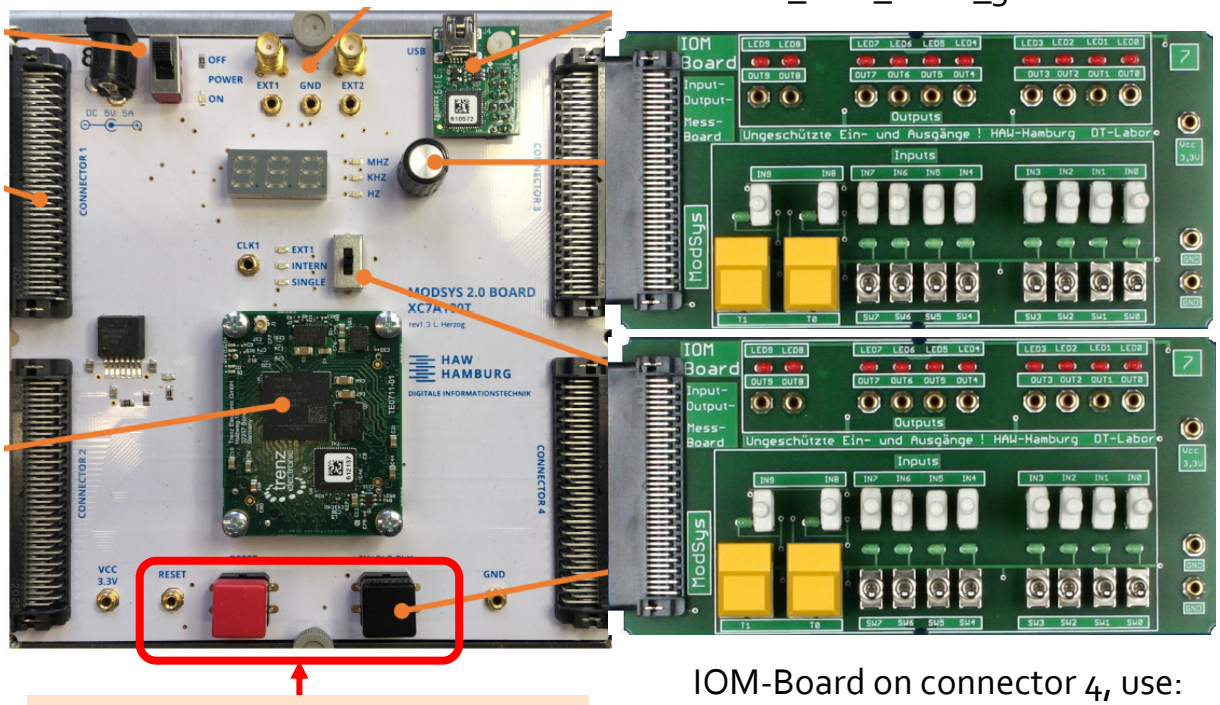


Fig. 4: Accumulating circuit

Board setup, all lab tasks

IOM-Board on connector 3 use:
Adder_IOM_Conn_3.xdc

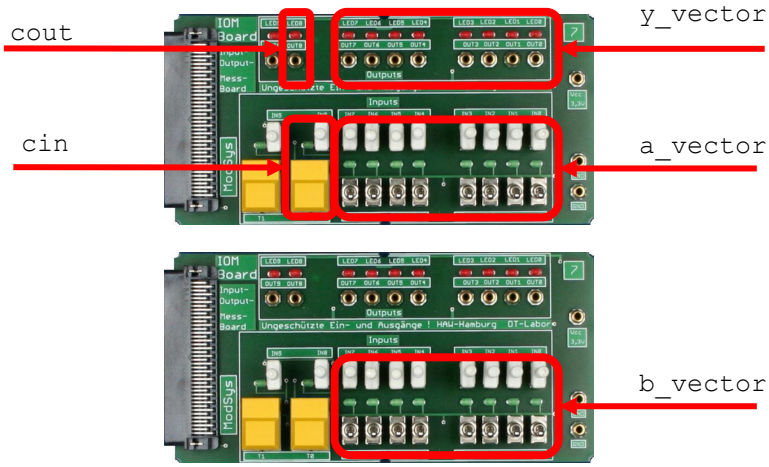


IOM-Board on connector 4, use:
Adder_IOM_Conn_4.xdc

clk and *resetn* from lab task 1.2 on

Lab task 1.1, link to entity signals

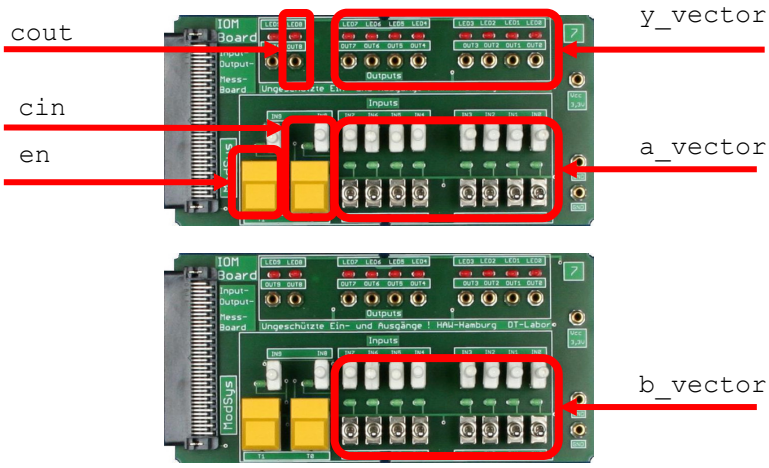
```
entity adder is
       generic (N: natural := 8);
       port(  a_vector: in std_logic_vector(N-1 downto 0);
              b_vector: in std_logic_vector(N-1 downto 0);
              y_vector: out std_logic_vector(N-1 downto 0);
              cin: in std_logic;
              cout: out std_logic);
end entity;
```



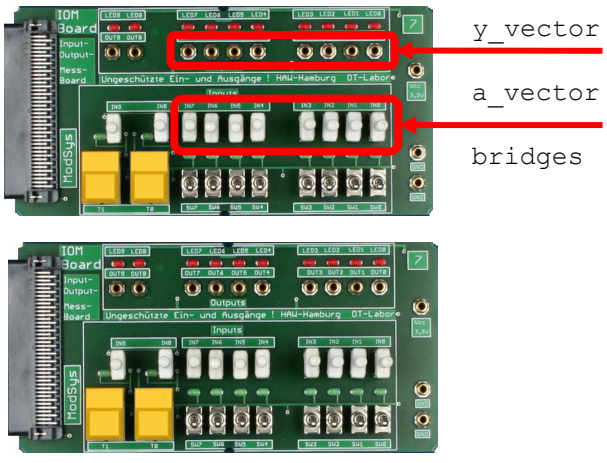
Lab task 1.2, link to entity signals

```
entity registered_adder is
       generic (N: natural := 8);
       port(  a_vector: in std_logic_vector(N-1 downto 0);
              b_vector: in std_logic_vector(N-1 downto 0);
              y_vector: out std_logic_vector(N-1 downto 0);
              cin: in std_logic;
              cout: out std_logic;
              clk: in std_logic;
              resetn: in std_logic;
              en: in std_logic);
end entity;
```

for *clk* and *resetn* use master xdc file
for *en* uncomment entry in Adder_IOM_Conn_3.xdc
leave Adder_IOM_Conn_4.xdc unmodified



Lab task 1.3: remove bridges of a_vector and install wires from y to a



Part 2
Counter Circuit

Objectives

- Use the registered Adder as a counter
- Implement the circuit using FPGA.

Part 2

Counter

Preparation

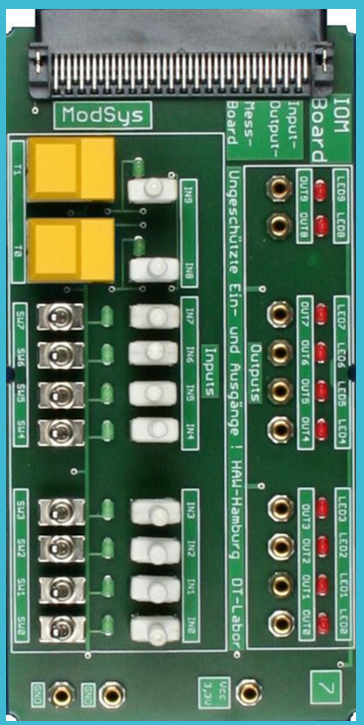


Fig. 5: IOM board

- Preparation Task 2.1 (accumulator as counter)
  - Calculate for the Accumulator of Lab task 1.3 the setting of the B input, so that
    - the accumulator acts as a counter which counts up.
    - the accumulator acts as a counter which counts down.
  - open your registered adder EDAplayground link and make a new (separate) copy of it:
  - Modify the testbench.vhd code for the DUT instantiation so that result Y is fed back to operand A input.
  - Write new TEST_DATA content with the above calculated values for B and check by simulation that a counting sequence appears on the adder output.
- Preparation Task 2.2 (counter with internal feedback)
  - Make a new copy on EDAplayground of your registered adder.
  - Modify the design.vhd, so that the feedback from Y to A is performed internally.
    - This can be done with or without changing the entity port list. But even if input for operand A is still present, it shouldn't be used anymore. Consider, that entity port signal names are always visible from inside the architecture body.
  - Modify the Testbench code accordingly. Reuse TEST_DATA from preparation task 2.1.
  - Perform simulations.
- Preparation Task 2.3
  - I/O specifications (1 IOM-Board required)
  - Copy the xdc file Adder_IOM_Conn_3.xdc under a different name and edit the copy (replace a_vector by b_vector).
  - Please familiarize yourself with using the board resources.
    - The switches are for Operand B
    - The keys (shown in yellow here) are for carry input and clock enable.
    - The LEDs are for the sum and the carry_out.
    - Clock and RESETN are taken from the main board. (main xdc!)

Part 2

Counter

Execution in the Laboratory

- Lab Task 2.1 (accumulator as counter)
  - **Reuse the non-modified setup from Lab Task 1.3!**
  - Configure the B inputs as calculated in Preparation task 2.1.
  - Use manual clocking first and then low frequency clock
  - Check both: up-counting and down-counting.
  - Increase clock frequency.
  - Measure clock-to-output delay with the oscilloscope (one channel on clock, 3 channels on the 3 lowest counter bits)
- Lab Task 2.2 (counter with internal feedback)
  - Create a new RTL project in Vivado.
  - From preparation task 2.2: Add the design VHDL file and the xdc-files for the counter.
  - Perform the Implementation Design Flow. (no simulation, just generate Bitstream)
  - Inspect schematic files if they show reasonable results.
  - Configure the B inputs as calculated in Preparation task 2.1.
  - Use manual clocking first and then switch to low frequency intern clock
  - Check both: up-counting and down-counting.
  - Increase clock frequency.
  - Measure clock-to-output delay with the oscilloscope (one channel on clock, 3 channels on the 3 least significant counter bits)

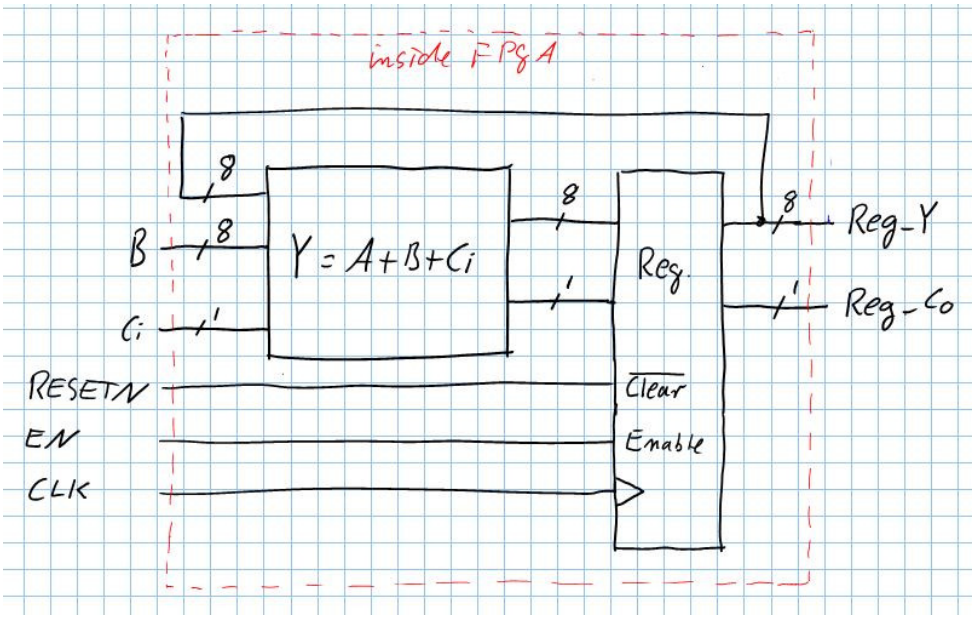


Fig. 6: Counter circuit

- References:

[1] L. Leutelt, T. Link, D. Palme "LAB Digital Circuits", HAW Hamburg Teaching Material

Lab task 2.2 link to entity signals

```
entity registered_adder is
       generic (N: natural := 8);
       port(  b_vector: in std_logic_vector(N-1 downto 0);
              y_vector: out std_logic_vector(N-1 downto 0);
              cin: in std_logic;
              cout: out std_logic;
              clk: in std_logic;
              resetn: in std_logic;
              en: in std_logic);
end entity;
```

For the IOM-Board on connector 3 use your edited copy of Adder_IOM_Conn_3.xdc