Part 2: 8-bit-ripple-carry-adder

# Task description:

For the preparations of this lab we were asked to run a ModelSim simulation of an 8bit adder with a given .vhd file (FULL\_ADDER.vhd). The testbench consisted of 5 testcases that we worked out ourselves.

In the lab it was our task to verify these testcases physically with the Xilinx and 2 IOM Boards as well as measure the time delay between a change at the most significant bit and the carry bit.

# 2.1 Verification of test cases

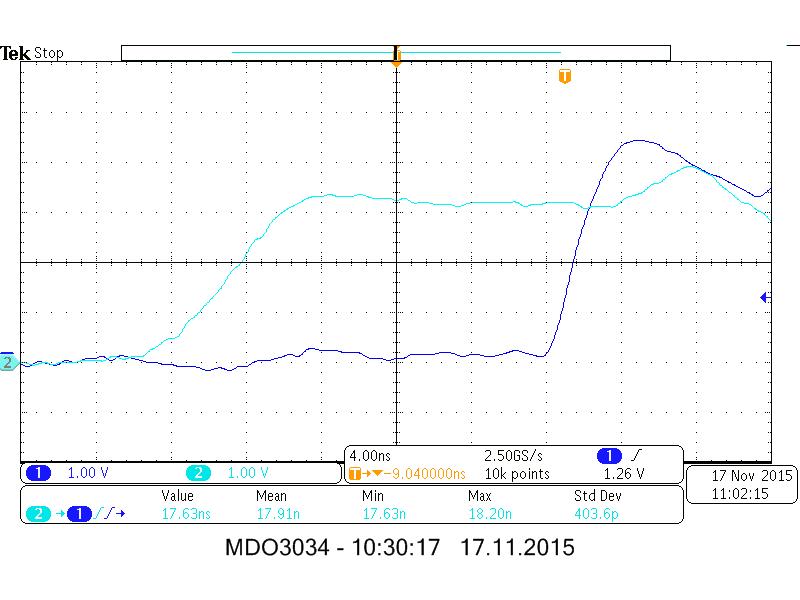
The following test cases were chosen to ensure proper behavior of the adder:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | Simulation Result | Lab Result |
| 05 | F3 | 0F8 | 0F8 |
| 3C | 9E | 0DA | 0DA |
| B5 | 97 | 14C | 14C |
| CD | EF | 1BC | 1BC |
| FF | FF | 1FE | 1FE |

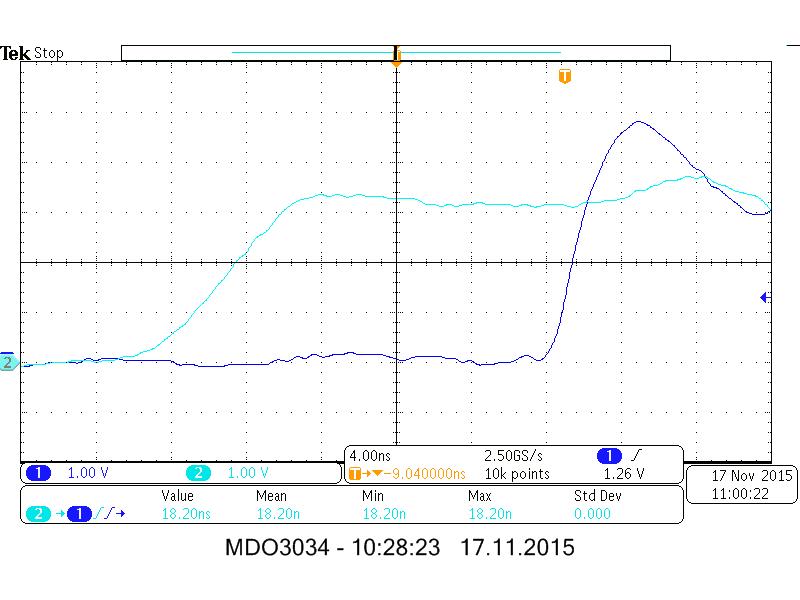
Our implemented adder on the CPLD gave us exactly the same results that we experienced in the simulation, thus it is safe to presume that the adder works in the desired way.

# 2.1 Time delay measurements

For the measurement of the time delay we used an oscilloscope where we connected 1 channel to the input of the carry and the second channel to one of the outputs (S[7] or C\_IN). With the use of the delay measurement of the oscilloscope as well the single shot method we were able to capture the following screenshots:



S[7] time delay



C\_IN time delay

The carry bit experienced a slightly higher delay time (18.2ns) compared to S[7] (17.63ns), although the difference is within the nano-range (0.57ns).

The delay-time is somewhat comparable to our ModelSim simulation, where the usual experienced delay we encountered was 15ns.

# 2.2 Resources

The synthesis of the .vhd and the .ucf file outputted us a fitter report and a timing report, where information about the implementation on the CPLD was stored.

Function blocks

Taken from this report, we merely used 2 out of the 8 available function blocks.

Macro cells

Each function block contains of 16 macrocells, of which the function blocks in use were occupying 8 each.

Product terms

One function block can make use of up to 56 product terms. For our implementation only 25 per active function block were used.

The CPLD is capable of realizing much more than just a simple adder, which can be seen from the usage of such few resources from the ones available.