**LAB TASK 2.1**

Synthesize the VHDL model of the mod-16 counter with load input and implement

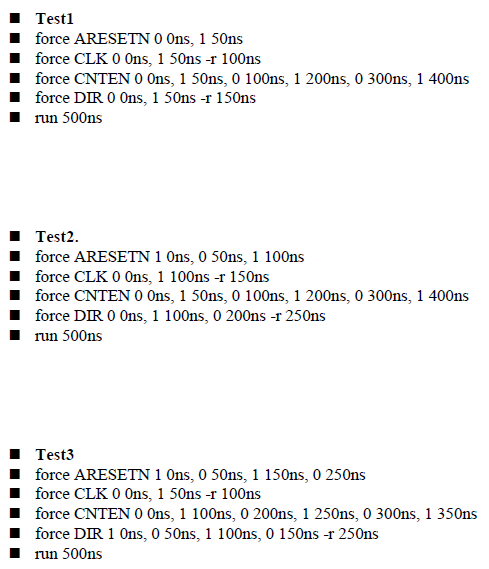
it on the CPLD.

* Verify the correct operation of the CPLDwith your test cases (see prep) and

document the results in the report.

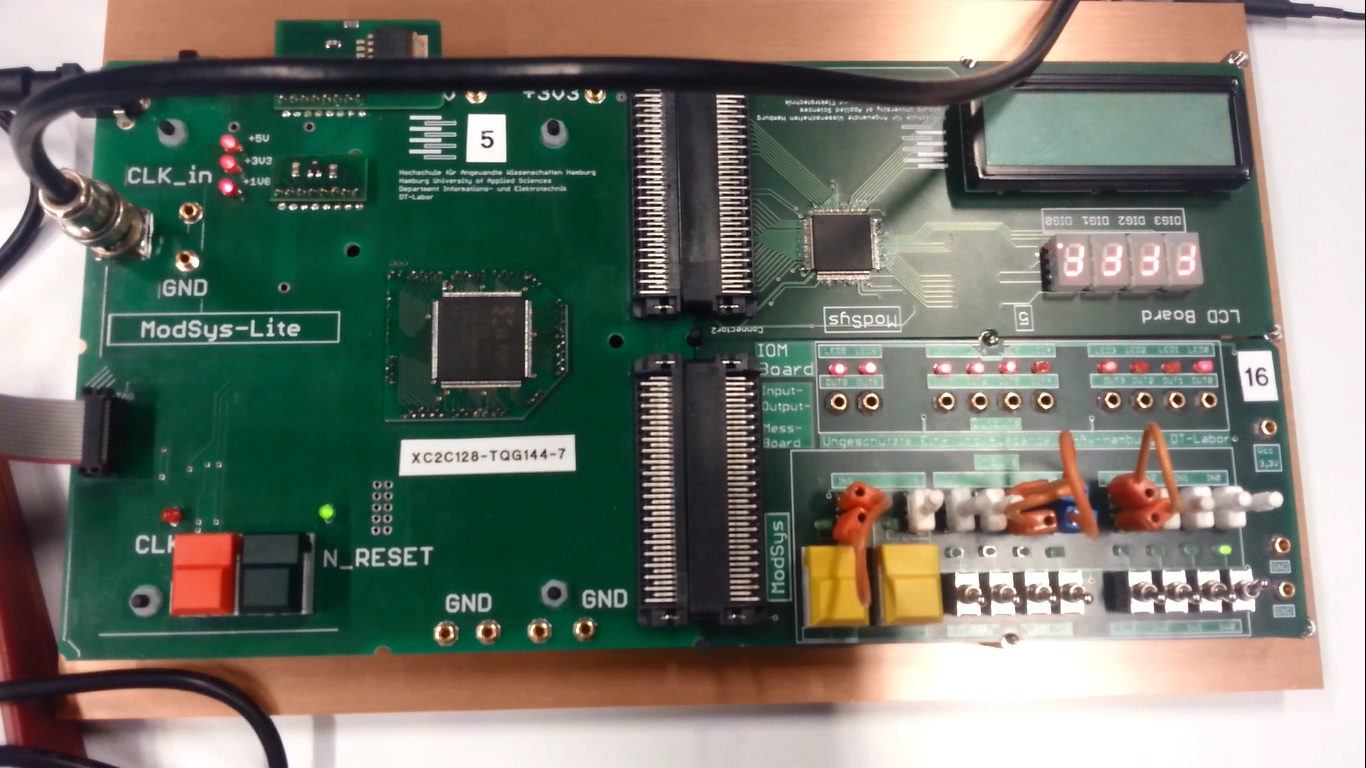
**Solution:**

We could use as clock signal either frequency generator or clock button. We used frequency generator so we saw all test cases required to test program. We used test cases which were provided in lab preparation. That test cases show normal operation, corner and edge cases.



(Source: Lab prepsheet of our group)

All test cases worked properly, as expected.



**Source**: one of the lab group member’s camera.

**LAB TASK 2.2**

* Measure the propagation delay between a rising clock edge and the Moore

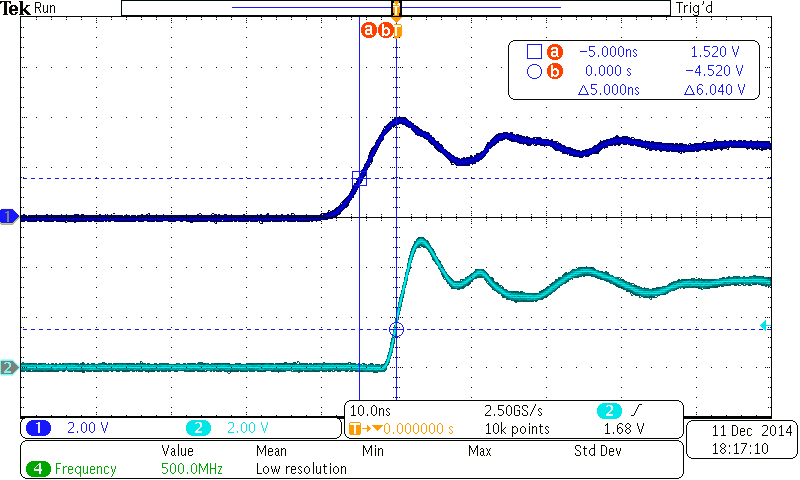
output CNT.

* Measure the propagation delay between a change of an input and the

asynchronous change of the output TC.

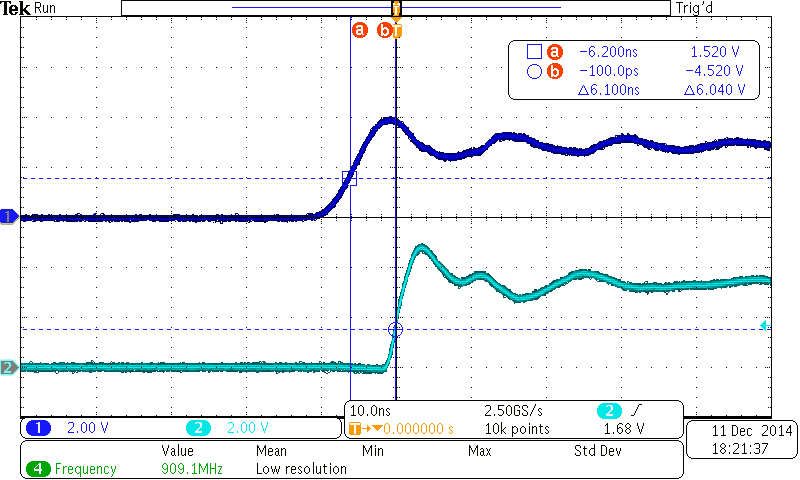
* Compare the results to the values in the Timing Report.

**Propagation delay between a rising clock edge and the Moore output CNT.**

****

Propagation delay between a rising clock edge and the Moore output CNT is 5.00ns.

**Propagation delay between a change of an input and the asynchronous change of the output TC.**

****

Propagation delay between a change of an input and the asynchronous change of the output TC is 6.100ns.

**Compare the results to the values in the Timing Report.**

|  |  |  |
| --- | --- | --- |
| **Source of Pad** | **Destination of Pad** | **Delay** |
| Rising clock edge | Moore output CNT | 5.00 ns |
| Asynchronous change of the output TC | Change of an input | 6.100 ns |

**LAB TASK 2.3**

Create a new project in Xilinx ISE Design Suite and add the VHDL model of LAB TASK 2.1 to it. Modify the VHDL model and add the signals required for usage of the 7-seg display to the entity ports. Make the necessary changes in process P\_OUT that the decimal count value is correctly displayed (not as a hexadecimal number!).

The UCF file for the LCD board will be provided in the lab.

Hint: the 4-bit value S has to be split into two decimal digits if the count value is greater than 9. In VHDL the “>”-operator is available.

**Modify VHDL file:**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity MOD16\_UD\_CNT is

port ( ARESETN, CLK, DIR, CNTEN, LOAD, TCEN : in bit;

PRE : in unsigned (3 downto 0);

CNT : out bit\_vector (3 downto 0);

NUM1: out bit\_vector (3 downto 0);

NUM2: out bit\_vector (3 downto 0);

TC, TYP : out bit);

end entity MOD16\_UD\_CNT;

architecture BEHAVIORAL of MOD16\_UD\_CNT is

signal S : unsigned(3 downto 0);

begin

P\_SYNC : process (ARESETN, CLK)

begin

TYP <= '0';

if ARESETN = '0' then S <= "0000" after 5 ns;

elsif LOAD = '1' then

if CLK'event and CLK='1' then S <= PRE after 5 ns;

end if;

else

if CLK'event and CLK='1' then

if CNTEN='0' then

if DIR = '0' then

S <= S - 1 after 5 ns;

else

S <= S + 1 after 5 ns;

end if;

end if;

end if;

end if;

end process P\_SYNC;

P\_OUT : process (S)

begin

NUM1 <= to\_bitvector(std\_logic\_vector(S));

NUM2 <= "0000";

if S > 9 then

NUM1 <= to\_bitvector(std\_logic\_vector(S - 10));

NUM2 <= "0001";

end if;

end process P\_OUT;

end architecture BEHAVIORAL;

The signals required for usage of the 7-seg display are added to the entity ports.

Necessary changes in process P\_OUT have been made. The decimal count value is correctly displayed, since S was splitted into 2 decimal digits, using simple VHDL operators.