Task 1

<StudentName1>, <StudNumb1> & <StudentName2>, <StudNumb2>

1 4-line 2-to-1 MUX

1.1 Implementation

To implement the 2-to-1 4-line multiplexer we took four 2-to-1 multiplexers without enable (M2_1) from the existing component library. All the select inputs were connected to a common select line. The two data inputs for the multiplexers are connect to two 4 bit wide busses D0 and D1 via bus-taps. Each D0(x) D1(x) pair is connected to one of the four multiplexers resulting in outputs that are bundled in the output bus O[3:0].

1.2 Simulation

To test the the implementation described we made a vhdl testbench. The testbench can be run within Xilinx ise by letting Isim run the MUX4_2to1_Test .vhd vhdl tesbench file. The file tests if the correct input is propagated to the output according to the current select input on the 4-line multiplexer (meaning either D0[3:0] or D1[3:0]). It also tests if every input bit is able to propagate to the output gate properly. It achieves this by alternating the select line every 125 ns and letting D0 and D1 be 0101 and 1010 respectively and switching this after 500 ns. This way we can check that every possible state (meaning 1/0) of every possible input pin (D0[x], D1[x]) can propagated properly to the output O[x].

2 Registers A & B

2.1 Implementation

Register A and B have the exact same function and can therefor be implement by the exact same hardware component. To implement registers A and B a register with parallel load capability and reset needs to be designed. Such a registers is capable of storing a value for multiple clock cycles and storing a new value on command via a load input. A component with this capability already existed within the standard component library (FD4RE). We only connected an input and output bus via bus-taps to this register component (FD4RE) to make future integration easier.

2.2 Simulation

To test the the implementation described we made a vhdl testbench. The testbench can be run within Xilinx ise by letting Isim run the RegisterTest .vhd vhdl testbench file. This testbench tests every combination of the control inputs LOAD and RST so we can check if it functions as expected. It also tests if every possible state of the inputs (1 or 0) can propagate into the storage an if all possible states of every storage elements (1 or 0) can be retained and can propagate to the output. This is all done by alternating RST and LOAD at certain moments (and CLK of course) and alternating The input bus I[3:0] between 0101 and 1010 and then checking via the output bus O[3:0] if all is correct.