Design of Digital Circuits: Lab Report

Lab 2: Mapping Your Circuit to an FPGA

Date		Grade
Names		
		Lab session / lab room

You have to submit this report via Moodle.

Use a zip file or tarball that contains the report and any other required material. Only one member from each group should submit the report. All members of the group will get the same grade.

The name of the submitted file should be *Lab2_LastName1_LastName2.zip* (or *.tar*), where *LastName1* and *LastName2* are the last names of the members of the group.

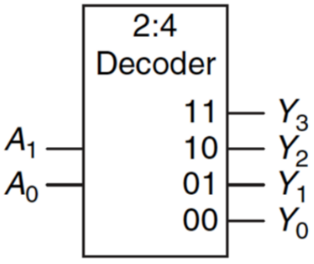
Note 1: Please include all the required material. No links/shortcuts are accepted.

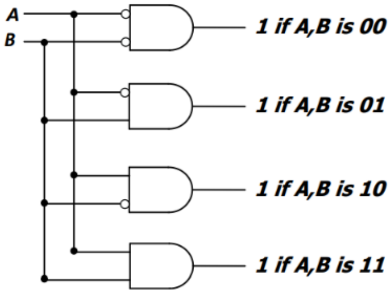
Note 2: The deadline for the report is a hard deadline and it will not be extended.

Exercise 1. Decoder

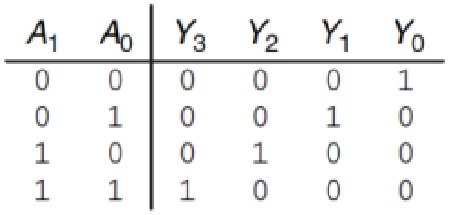
In this exercise, you will design a Decoder module and implement it using built-in logic gates (e.g., AND, OR, NOT, ...) in Verilog. Here is the description of a 2-input Decoder:

- The 2-input decoder has 2 inputs and 4 outputs.
- For any given input, exactly one of the decoder's outputs is 1; all other outputs are 0.
- The single output that is logically 1 is the output that corresponds to the input pattern applied to the circuit. For example, the first output of the decoder will be 1 when the input is '00'.

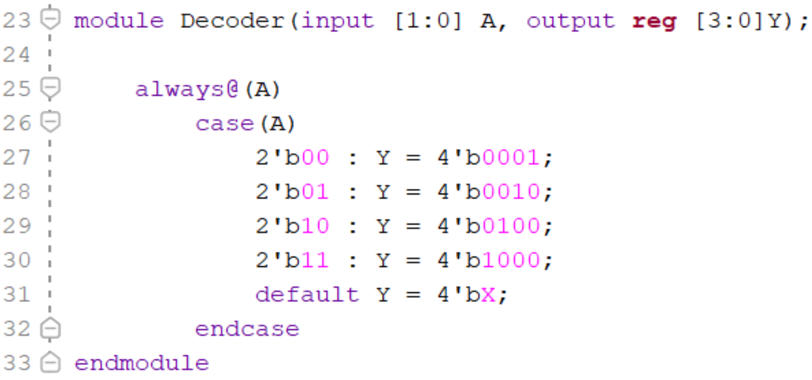




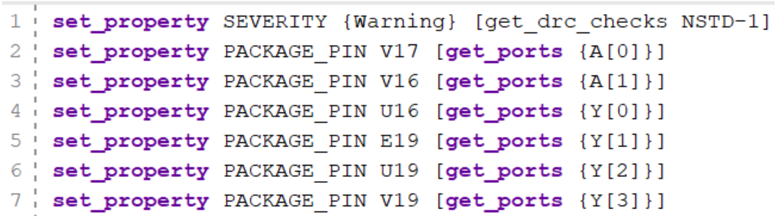
Map the inputs to two switches on the board, and map outputs to 4 LEDs on the board. Map your design to the FPGA and check the correctness of your design. Include your Verilog codes files and the constraint file in your report submission.



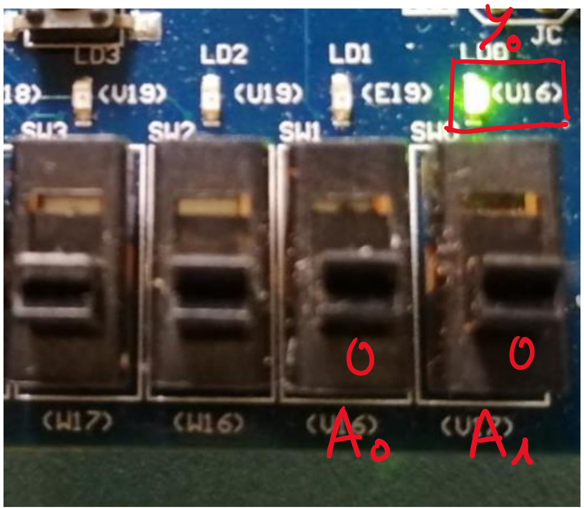
A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

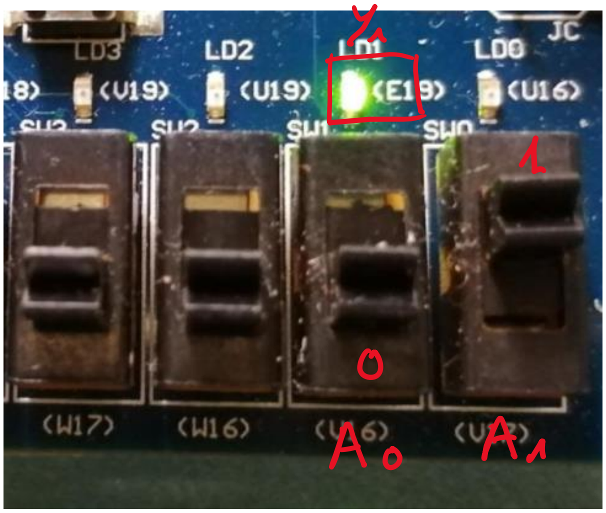


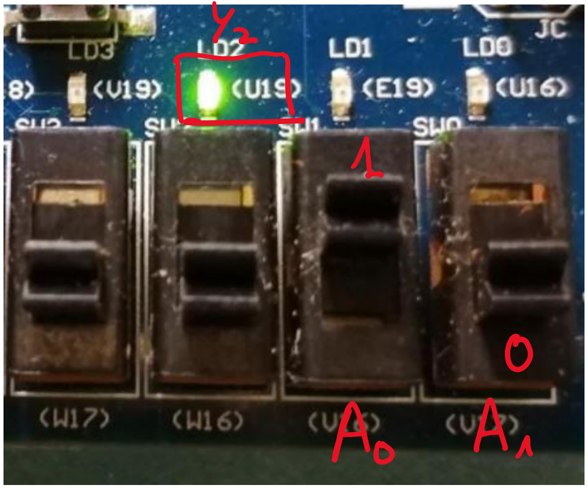
```
module Decoder(input [1:0] A, output reg [3:0]Y);

    always@(A)
        case(A)
            2'b00 : Y = 4'b0001;
            2'b01 : Y = 4'b0010;
            2'b10 : Y = 4'b0100;
            2'b11 : Y = 4'b1000;
            default Y = 4'bX;
        endcase
endmodule
```

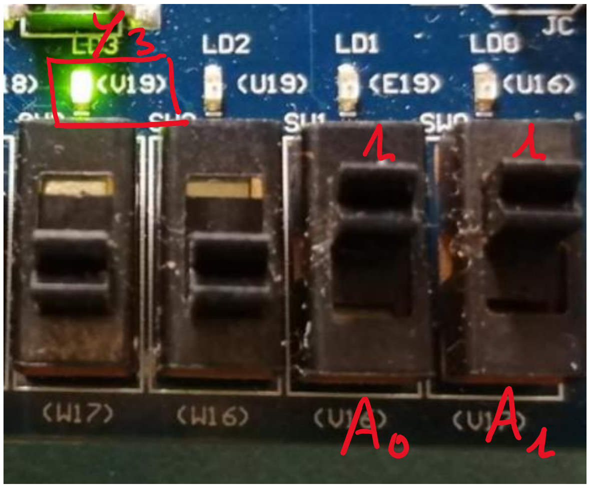


```
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property PACKAGE_PIN V17 [get_ports {A[0]}]
set_property PACKAGE_PIN V16 [get_ports {A[1]}]
set_property PACKAGE_PIN U16 [get_ports {Y[0]}]
set_property PACKAGE_PIN E19 [get_ports {Y[1]}]
set_property PACKAGE_PIN U19 [get_ports {Y[2]}]
set_property PACKAGE_PIN V19 [get_ports {Y[3]}]
```







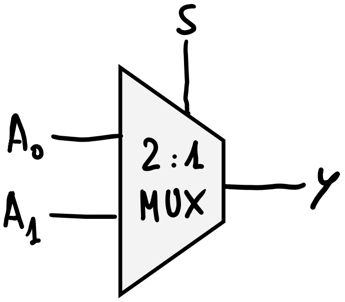


Exercise 2. Multiplexer

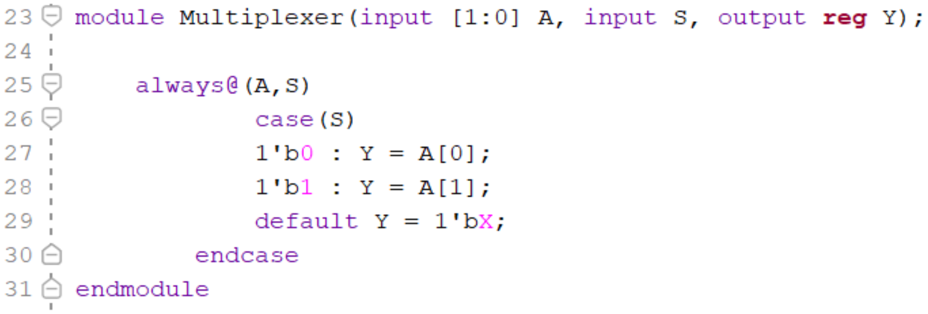
(a) Design a multiplexer with these features:

- The multiplexer should select one of *2* inputs and connect it to its output.
- The selection of which input is routed to the output is controlled by a 1-bit control input.

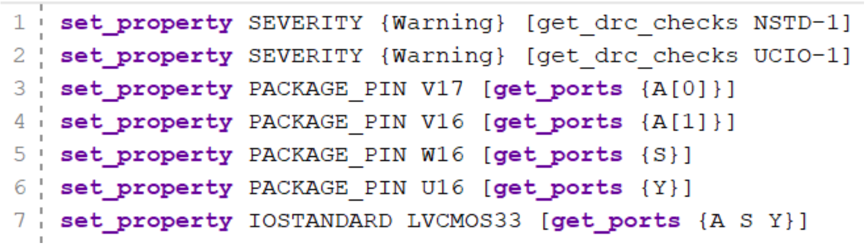
We call this multiplexer a 2:1 MUX



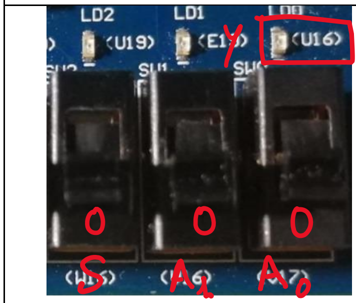
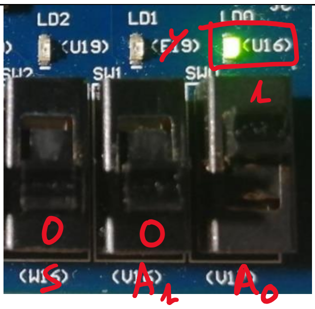
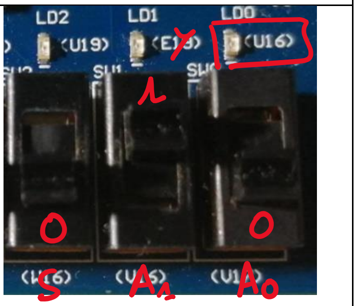
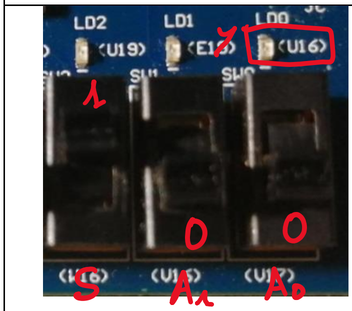
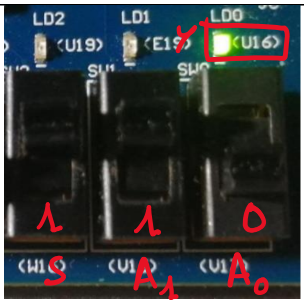
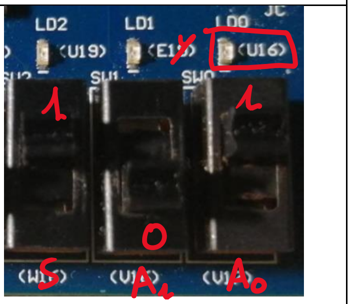
S	Y
0	A_0
1	A_1



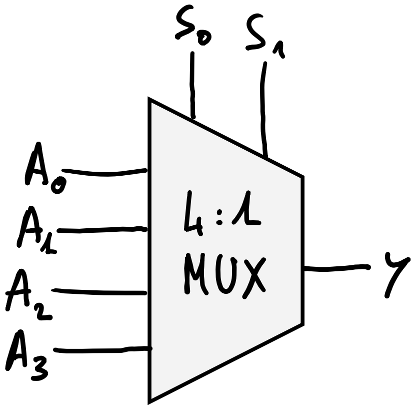
```
module Multiplexer(input [1:0] A, input S, output reg Y);

    always@(A,S)
            case(S)
            1'b0 : Y = A[0];
            1'b1 : Y = A[1];
            default Y = 1'bX;
        endcase
endmodule
```



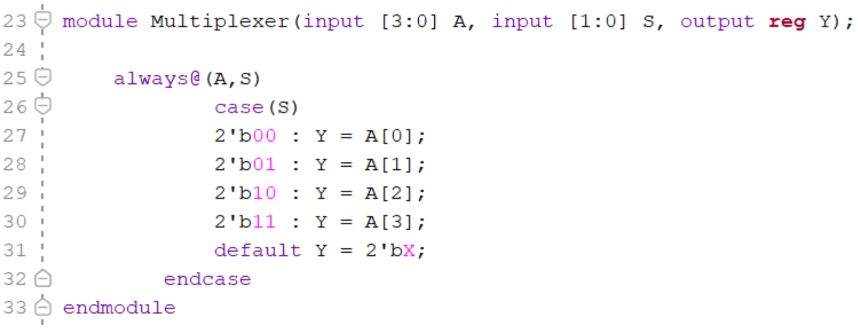
```
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property SEVERITY {Warning} [get_drc_checks UCIO-1]
set_property PACKAGE_PIN V17 [get_ports {A[0]}]
set_property PACKAGE_PIN V16 [get_ports {A[1]}]
set_property PACKAGE_PIN W16 [get_ports {S}]
set_property PACKAGE_PIN U16 [get_ports {Y}]
set_property IOSTANDARD LVCMOS33 [get_ports {A S Y}]
```

A_0 selected		Check with A_1
		
A1 selected		Check
		

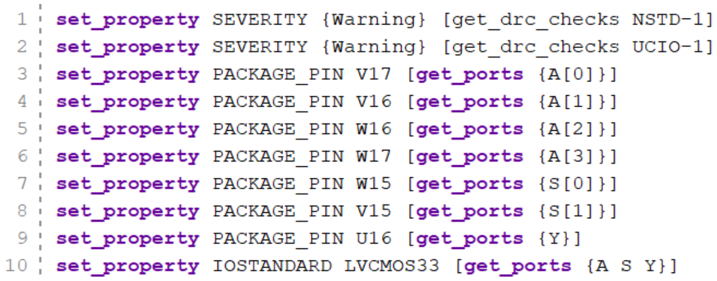
(b) Use several instances of your 2:1 MUX to design a 4:1 MUX. Map your design to your FPGA board, with input and output ports of your choice, and check the functionality of your circuit.



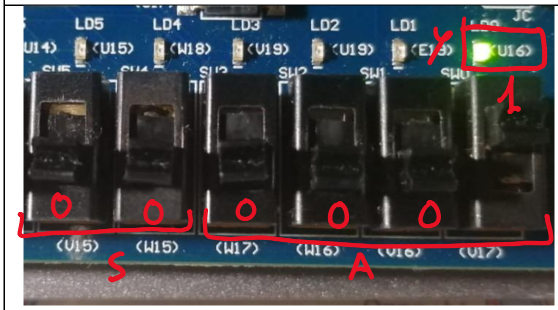
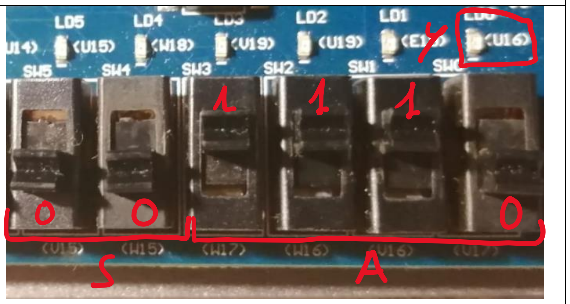
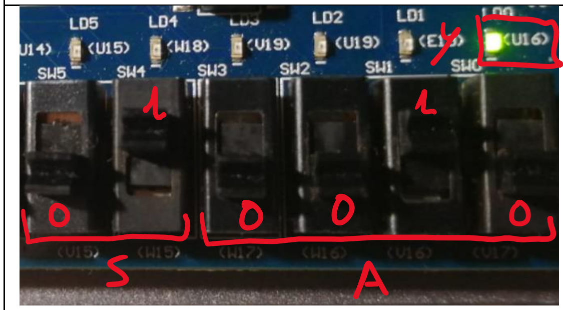
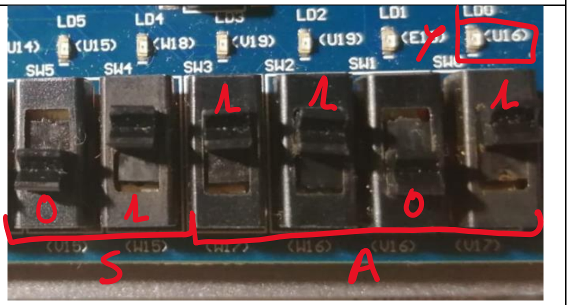
S_0	S_1	Y
0	0	A_0
1	0	A_1
0	1	A_2
1	1	A_3

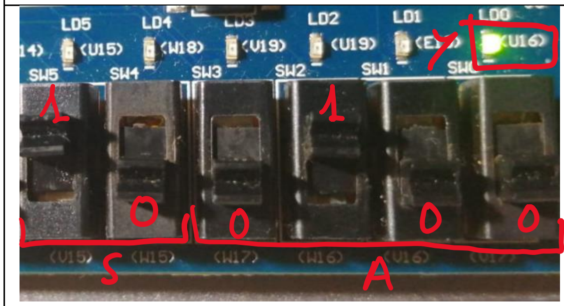
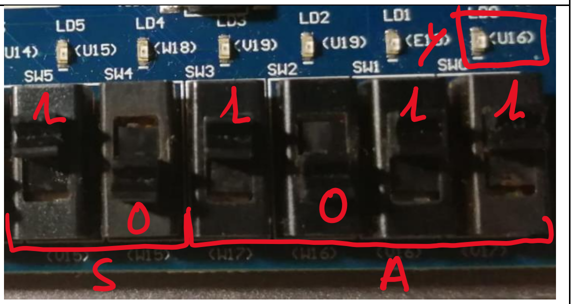
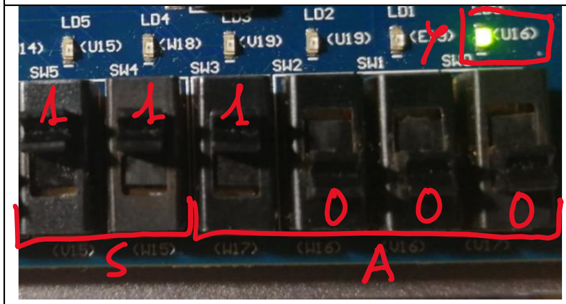
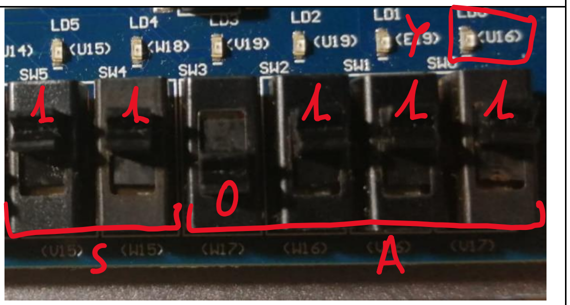


```
module Multiplexer(input [3:0] A, input [1:0] S, output reg Y);

    always@(A,S)
            case(S)
            2'b00 : Y = A[0];
            2'b01 : Y = A[1];
            2'b10 : Y = A[2];
            2'b11 : Y = A[3];
            default Y = 2'bX;
        endcase
endmodule
```



```
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property SEVERITY {Warning} [get_drc_checks UCIO-1]
set_property PACKAGE_PIN V17 [get_ports {A[0]}]
set_property PACKAGE_PIN V16 [get_ports {A[1]}]
set_property PACKAGE_PIN W16 [get_ports {A[2]}]
set_property PACKAGE_PIN W17 [get_ports {A[3]}]
set_property PACKAGE_PIN W15 [get_ports {S[0]}]
set_property PACKAGE_PIN V15 [get_ports {S[1]}]
set_property PACKAGE_PIN U16 [get_ports {Y}]
set_property IOSTANDARD LVCMOS33 [get_ports {A S Y}]
```

A_0 selected	**Check with others**
	
A_1 selected	**Check with others**
	

A_2 selected	Check with others
 LD5 LD4 LD2 LD1 LD0 (U15) (W18) (V19) (U19) (E19) (U16) SW5 SW4 SW3 SW2 SW1 SW0 Y 1 0 0 1 0 0 S A	 LD5 LD4 LD2 LD1 (U15) (W18) (V19) (U19) (E19) (U16) SW5 SW4 SW3 SW2 SW1 Y 1 0 1 0 1 1 S A
A_3 selected	Check with others
 LD5 LD4 LD2 LD1 (U15) (W18) (V19) (U19) (E19) (U16) SW5 SW4 SW3 SW2 SW1 Y 1 1 1 0 0 0 S A	 LD5 LD4 LD2 LD1 (U15) (W18) (V19) (U19) (E19) (U16) SW5 SW4 SW3 SW2 SW1 Y 1 1 0 1 1 1 S A

Include your Verilog files and constraint file in your submission.

Feedback

If you have any comments about the exercise please add them here: mistakes in the text, difficulty level of the exercise, or anything that will help us improve it for the next time.