Pass/Fail Homework 1

Use the last 3 digits of your personal number (PN) to select which question you have to answer. Modulo divide your 3 digits with the number of questions, in this case, 8, plus 1. The resulting number is the question that you should answer.

!!! (PN mod 8) + 1 !!!

For example, if your PN ends in 730, then (730 mod 8) + 1 = 3, meaning you must answer question 3.

Make sure you comment on the top of each file you submit, your PN, name and the question you are answering.

!!! Important !!!

If you answer the wrong question, your submission will be invalid.

We can call you to explain your solution. If you cannot explain your answer, the homework will be invalid!

KTH has a zero-tolerance policy against cheating.

https://www.kth.se/en/student/stod/studier/fusk-1.997287

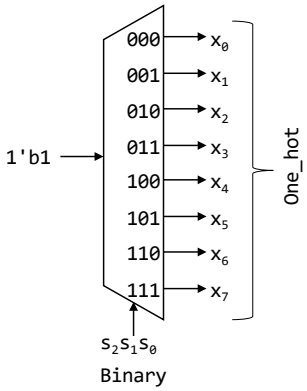
If you have questions or need clarifications, you can ask in the discussion forum in Canvas.

Questions

1.1 Question 1

Model using HDL a look-up table **using a 1-to-16 de-multiplexer** that can act as a 4-bit one-hot decoder. The design should take as input a 4-bit binary value and output the one-hot equivalent. The definition of the module is given below.

```
module decoder (
  input  logic [3:0] binary,
  output logic [15:0] one_hot
);
  // …
  // Add your description here
  // …
endmodule
```

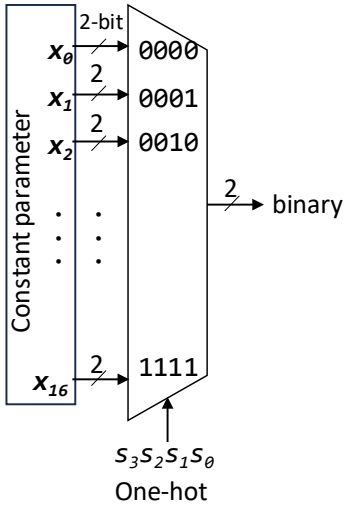


1.2 QUESTION 2

Model using HDL a design that can act as a 4-to-2-bit encoder. The design should take as input a 4-bit one-hot number and output the equivalent binary value. The table below shows the functionality of a simple 4-to-2-bit encoder. Note that the '-' in the table signifies don't-care values. The don't care values can be either 0 or 1. The definition of the module is given below.

4 to 2 Simple Encoder

I_3	I_2	I_1	I_0	O_1	O_0	Valid
-	-	-	-	-	-	0
0	0	0	0	-	-	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1



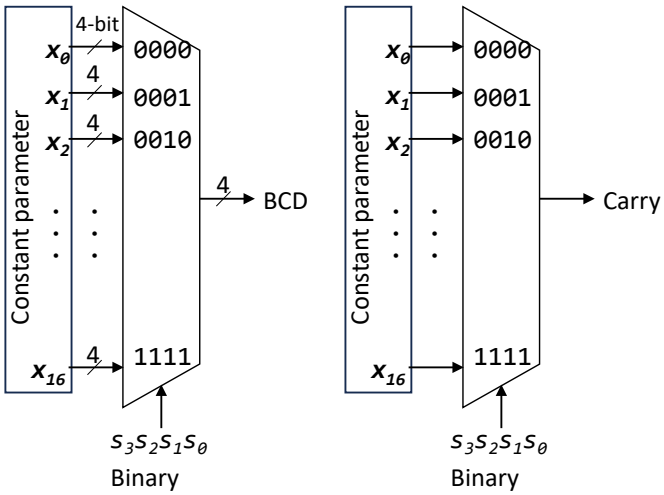

```
module one_hot_encoder (
  input  logic [3:0] one_hot,
  output logic [1:0] binary,
  output valid
);
  // …
  // Add your description here
  // …
endmodule
```

1.3 QUESTION 3

Model using HDL a design that uses **16-to-1 multiplexers** and implements a 4-bit binary to BCD (Binary-coded decimal) encoder. The BCD encoding can be seen in the following table. The module definition is also seen below.

Decimal	Binary	BCD				Carry
		8	**4**	**2**	**1**	
0	0000	0	0	0	0	0
1	0001	0	0	0	1	0
2	0010	0	0	1	0	0
3	0011	0	0	1	1	0

4	0100	0	1	0	0	0
5	0101	0	1	0	1	0
6	0110	0	1	1	0	0
7	0111	0	1	1	1	0
8	1000	1	0	0	0	0
9	1001	1	0	0	1	0
10	1010	0	0	0	0	1
11	1011	0	0	0	1	1
12	1100	0	0	1	0	1
13	1101	0	0	1	1	1
14	1110	0	1	0	0	1
15	1111	0	1	0	1	1

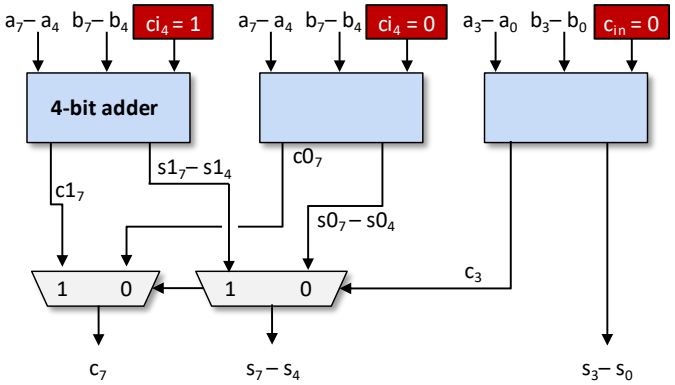



```
module bin2bcd (
  input  logic [3:0] binary,
  output logic [3:0] bcd,
  output logic carry
);
  // …
  // Add your description here
  // …
endmodule
```

1.4 QUESTION 4

Model a design using HDL that implements an 8-bit unsigned carry select adder (CSA). The CSA should be built using 4-bit adders. The design of the adder is given to you. The definition of the module can be seen below.

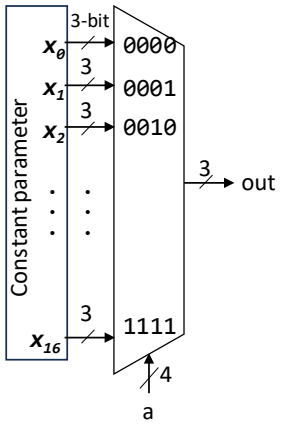
```
module CSA_8 (
  input  logic [7:0] A, B,
  output logic [7:0] sum,
  output logic carry
);
  // …
  // Add your description here
  // …
endmodule
module adder_4 (
    input logic [3:0] A, B,
    output logic [3:0] sum,
    output logic carry
  );
  assign {carry, sum} = A+B;
endmodule
```



1.5 Question 5

Model a design using HDL that can count the number of 1s in an 4-bit binary number. For example, the `4'b0101` should output `3'b010`. The definition of the module is given below.

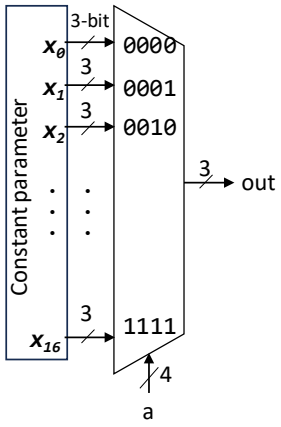
```
module count_1 (
  input  logic [3:0] a,
  output logic [2:0] out
);
  // …
  // Add your description here
  // …
endmodule
```



1.6 Question 5

Model a design using HDL that can count the number of 0s in an 4-bit binary number. For example, the `4'b0001` should output `3'b011`. The definition of the module is given below.

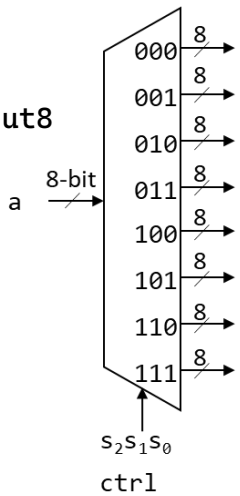
```
module count_0 (
  input  logic [3:0] a,
  output logic [2:0] out
);
  // …
  // Add your description here
  // …
endmodule
```



1.7 Question 7

Model a design in HDL that has one 8-bit input, a 3-bit control input, and eight 8-bit outputs. The value from the input should be assigned to the corresponding output based on the 2-bit control value, while the other three outputs are set to zero. The definition of the module is given below.

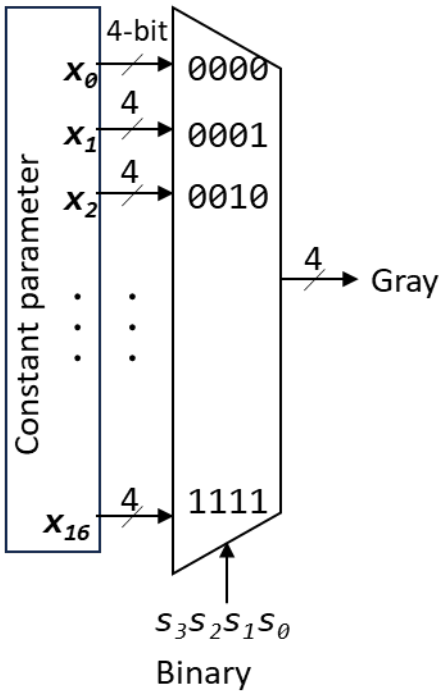
```
module switchbox (
  input  logic [7:0] a,
  input  logic [2:0] ctrl,
  output logic [7:0] out1,out2,out3,out4,out5,out6,out7,out8
);
  // …
  // Add your description here
  // …

endmodule
```



1.8 QUESTION 8

Model using HDL a design that uses a **16-to-1 multiplexer** and implements a 4-bit binary to gray encoder. The gray encoding can be seen in the following table. The module definition is also seen below.

Decimal	Binary	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000




```
module bin2gray (
  input  logic [3:0] binary,
  output logic [3:0] bcd
);

  // …
  // Add your description here
  // …

endmodule
```