1-point 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, 5, plus 1. The resulting number is the question that you should answer.

!!! (PN mod 5) + 1 !!!

For example, if your PN ends in 730, then (730 mod 5) + 1 = 1, meaning you must answer question 1.

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

A) Model using HDL a design that can be used to implement an arithmetic right shift. Your design has to be parametric and have N-bit input, where N ≥ 4. It should also allow to shift up to 3 positions. Do **not** use the '>>>, <<<' operators.

```
module ArithmeticRightShifter #(parameter N) (
  input  logic [N-1:0] input_data,
  input  logic [1:0] control,
  output logic [N-1:0] shifted_result
);
```

B) Draw a schematic of your design. Assume a value for the parameter N=5.

1.2 QUESTION 2

A) Model using HDL a signed N-bit multiplier. The design has to be parametric with 2 inputs of N-bit, where N>3. Your design should use half adders and/or full adders (given below).

B) Draw the schematic of your design. Assume a value for the parameter N=5.

```
module full_adder (
    input a,b,c_in,
    output c_out, s
  );
  logic s1,c1,c2;
  half_adder ha1(a,b,s1,c1);
  half_adder ha2(s1,c_in,c2,s);
  assign c_out = c1|c2;
  assign c_out = c1|c2;
endmodule
```

```
module half_adder (
    input a,b,
    output c_out, s
  );
  assign s = a^b;
  assign c_out = a&b;
endmodule
```

```
module multiplier #(parameter N) (
    input [N-1:0]a,b,
    output [2*N-1:0] product
  );
```

1.3 QUESTION 3

Model using HDL a generic unit that receives an 16-bit number in Q<I,F_{in}> to an 8-bit number with representation Q<I,F_{out}>. The design should have the following parameters: I, F_{in}, F_{out}. You have the following assumptions: 16=I+F_{in}, and 8=I+F_{out}, and F_{in}>F_{out}. The I represents the number of integer bits in the number, including the sign, and F_{in}, F_{out} represents the fraction bits. The design should be able to do both rounding up, or rounding down, based on a control input.

```
module saturation_round #(parameter I, parameter Fin, parameter Fout) (
    input logic [15:0] in,
    input logic up_down,
    output logic [8:0] out
  );
```

1.4 QUESTION 4

A) Model HDL, describe a unit that compares two signed numbers. The unit should be parametric and get two N-bit inputs, namely A and B. It should also have the following single-bit outputs: G, L, E. The single-bit outputs should be asserted to 1 under the following conditions.

G = 1'b1 if A>B

E = 1'b1 if A=B

L = 1’b1 if A<B

You are **not** allowed to use any of the following operands ‘<,<=,>,>=,==,!=’.

B) Draw a schematic of your design. Assume a value for the parameter N=4.

```
module comparator #(parameter N) (
    input [N-1:0] in,
    output G, E, L
  );
```

1.5 Question 5

A) Model using HDL a design that can multiply 6 N-bit numbers and add the results. The design should be parametric with parameter N. You must ensure that your design does not overflow or underflow and assign the correct bit width for all intermediate and output signals. The design should implement the following equation:

$$out = \sum_{k=0}^{5,k+2} X_k \cdot X_{k+1}$$

B) Draw a schematic of your design. Assume a value for the N=4 parameter.

```
module sum_prod #(parameter N) (
    input [N-1:0] X [5:0],
    output [?:0] result
  );
```