TUT 5 - EEE3096S
08/10/2021
BRMKEA001, CLLSTE009

---

1. How do FPGAs differ from microcontrollers? Give two advantages of FPGAs, and 2 disadvantages.

The main difference between FPGAs and microcontrollers is that FPGAs have flexible hardware structure while microcontrollers have fixed hardware structure. FPGAs are more application specific while microcontrollers are more general solutions.

The main advantages of FPGAs is: the hardware is programmable and flexible; the instruction set is not fixed and instructions can be executed in parallel.

The main disadvantages of FPGAs is that they are comparatively more complex and expensive than microcontrollers as well as draw more power.

2. What is the difference between blocking and non-blocking assignments? Support your answer by comparing the two code snippets below and how their outputs may differ.

Blocking assignments block the execution of the next assignment resulting in order execution. While non-blocking assignments do not block the execution of the next assignment allowing for simultaneous or parallel execution.

```
module blocking(in, clk, out);
  input in, clk;
  output out;
  reg q1, q2, out;
  always @ (posedge clk)
  begin
    q1 = in;
    q2 = q1;
    out = q2;
  end
endmodule
```

```
module nonblocking(in, clk, out);
  input in, clk;
  output out;
  reg q1, q2, out;
  always @ (posedge clk)
  begin
    q1 <= in;
    q2 <= q1;
    out <= q2;
  end
endmodule
```

Blocking	in	out	q1	q2
Initial values	1	0	0	0
q1=in	1	0	1	0
q2 =q1	1	0	1	1
out=q2	1	1	1	1

A table was created where each line is the code that is being run in order. As can be seen, out will be set to 1 as it is sequentially equal to in.

A similar table was created below for the non blocking code.

Non-blocking	in	out	q1	q2
Initial values	1	0	0	0
q1=in	1	0	1	0
q2 =q1	1	0	1	0
out=q2	1	0	1	0

As can be seen above the non blocking code will output out as 0. This is because if all the lines of code are run in parallel, q1 is set to 1, q2 will be set to q1 where q1 is still 0 and out is set to q2 where q2 is still 0.

3. Port Mapping in module instantiation can be done in two different ways; 1. Port Mapping by order 2. Port Mapping by name For the module above; what is the output of the following testbenches done using port mapping by order (on the left) and mapping by name (on the right): Comment on the results from each testbench. Which port mapping method is better to use when instantiating a module with many ports?

Named mapping is easier to read and less error prone when working with many ports as you can verify your mappings while writing them e.g. .clk(clk) you can easily see that you are mapping the correct ports. Whereas with port mapping by order it would be easier to make a mistake if you have to enter many ports in the instantiation of a module because you do not have that visual verification that you have with named port mapping.

The code used below outputted: `0 1 x`

```
module sum_tb();
reg A, B, clk;
wire out;
sum ut(A,B,clk,out);
 initial begin
  $display("A B Out");
  $monitor("%b %b %b", A, B, out); clk = 1'b1; A = 1'b0; B = 1'b1;
 end
endmodule
```

The code below outputted: `0 1 1`

```
module sum_tb();
  reg A, B, clk;
  wire out;
  sum ut(.A(A),.B(B),.clk(clk),.out(out));

  initial begin

    $display("A B Out");
    $monitor("%b %b %b", A, B, out);
       clk = 1'b1;
    A = 1'b0;
    B = 1'b1;

  end
endmodule
```