# Lab 10 – Worksheet

|  |  |  |
| --- | --- | --- |
| Name: Ali Muhammad Asad  Shayan Shoaib Patel | ID: aa07190  sp0smthng0 | Section: T6 |

## Task a. h\_counter

*Provide appropriately commented code for your h\_count design module*

|  |
| --- |
| `timescale 1ns / 1ps  module h\_counter(  input clk,  output [9:0] h\_count,  output trig\_v  );  reg[9:0] h\_count;  reg trig\_v;  initial h\_count = 0;  always @ (posedge clk)  begin  if (h\_count < 799)  begin  h\_count <= h\_count + 1; trig\_v <= 0;  end  else  begin  trig\_v <= 1;  h\_count <= 0;  end  end  endmodule |

Modify the testbench provided in Figure 10. 9(b) to verify the functionality of your designed h\_counter module.

*Attach your testbench here.*

|  |
| --- |
| `timescale 1ns / 1ps  module testbench\_h\_counter();  reg clk; wire [9:0] h\_count; wire trig\_v;  h\_counter h\_c1(.clk(clk), .h\_count(h\_count), .trig\_v(trig\_v));  initial  clk = 1'b0;  always  #5 clk = ~clk;  endmodule |

*Attach screenshot of waveform here*

|  |
| --- |
|  |

### Concept check:

Assuming that a time unit, in above mentioned testbench, is defined in nano seconds; what is the frequency at which the designed h\_counter module would operate?

|  |
| --- |
|  |

## Task b. v\_counter

*Provide appropriately commented code for your v\_counter design module*

|  |
| --- |
| `timescale 1ns / 1ps  module v\_counter(  input clk,  input enable\_v,  output reg [9:0] v\_counter  );  initial v\_counter = 0;  always @ (posedge clk)  begin  if (v\_counter < 524)  begin  if(enable\_v == 1)  begin  v\_counter <= v\_counter + 1;  end  else  begin  v\_counter <= v\_counter;  end  end  else  begin  v\_counter <= 0;  end  end  endmodule |

Modify the testbench provided in Figure 10. 9(b) to verify the functionality of your designed v\_counter module.

*Attach your testbench here.*

|  |
| --- |
| `timescale 1ns / 1ps  module testbench\_v\_counter();  reg clk; reg enable\_v; wire[9:0] v\_counter;  v\_counter vc1(.clk(clk), .enable\_v(enable\_v), .v\_counter(v\_counter));  initial  begin  clk = 1'b0;  enable\_v = 1'b0;  end  always #5 clk = ~clk;  always #10 enable\_v = ~enable\_v;  endmodule |

*Attach screenshot of waveform here*

|  |
| --- |
|  |

## Task c: Clock divider

*Show working to calculate div\_value*

|  |
| --- |
| div\_value = [input clock frequency / 2(desired clock frequency) ]- 1 |

*Provide appropriately commented code for your clk\_div design module*

|  |
| --- |
| `timescale 1ns / 1ps  module clk\_div (clk, clk\_d);  parameter div\_value = 1;  input clk;  output clk\_d;  reg clk\_d;  reg count;  initial  begin  clk\_d = 0;  count = 0;  end  always @(posedge clk)  begin  if (count == div\_value)  count <= 0; // reset count  else count <= count + 1; // count up  end  always @(posedge clk)  begin  if (count == div\_value)  clk\_d <= ~clk\_d;  end  endmodule |

Modify the testbench provided in Figure 10. 9(b) to verify the functionality of your designed clk\_div module.

*Attach your testbench here.*

|  |
| --- |
| `timescale 1ns / 1ps  module clk\_div\_testbench();  reg clk;  wire clk\_d;  clk\_div cd1(.clk(clk), .clk\_d(clk\_d));  initial  begin  clk = 1'b0;  end    always  #5 clk = ~clk;  endmodule |

*Attach screenshot of waveform here*

|  |
| --- |
|  |

## Exercise 1

*Provide appropriately commented code for your top design module*

|  |
| --- |
| `timescale 1ns / 1ps  module clk\_TopLeveLModule(  input clk,  output [9:0] h\_count,  output [9:0] v\_count  );  wire clk\_div\_out;  wire trig\_v;  clk\_div cd1(clk, clk\_div\_out);  h\_counter(clk\_div\_out, h\_count, trig\_v);  v\_counter(clk, trig\_v, v\_count);  endmodule |

Modify the testbench provided in Figure 10. 9(b) to verify the functionality of your designed top module.

*Attach your testbench here.*

|  |
| --- |
| `timescale 1ns / 1ps  module testbench\_clk\_TopLevelModule();  reg clk;  wire [9:0] h\_count;  wire [9:0] v\_count;  clk\_TopLevelModule clkTLM(clk, h\_count, v\_count);  initial  clk = 1'b0;  always  #1 clk = ~clk;  endmodule |

*Attach screenshot of waveform here*

|  |
| --- |
|  |

## Exercise 2

*Provide appropriately commented code for your top-level design module*

|  |
| --- |
|  |

*Attach picture of FPGA here*

|  |
| --- |
|  |

## 

## Assessment Rubrics

**Marks Distribution:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **LR2**  **Code** | **LR5**  **Results** | **LR7**  **Viva** |
| **In-lab** | **Task a** | 10 points | 5 points | 10 points |
| **Task b** | 10 points | 10 points |
| **Task c** | - | 5 points |
| **Exercise 1** |  | 20 points | 10 points |
| **Exercise 2** |  | 10 points | 10 points |  |
| **Total** |  |  |  |  |

**Marks Obtained:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **LR2**  **Code** | **LR5**  **Results** | **LR7**  **Viva** |
| **In-lab** | **Task a** |  |  |  |
| **Task b** |  |  |
| **Task c** | - |  |
| **Exercise 1** |  |  |  |
| **Exercise 2** |  |  |  |  |
| **Obtained** |  |  |  |  |