Post Lab Questions by Ethan Zhou (yz69) & Qiran Pang (qpang2)

Question 1.

There are 126 digital IO pins on the XEM 7310-A75

Question 2.

The clock speed is 200MHz

Question 3.

|  |  |  |
| --- | --- | --- |
|  | XEM 6002 | XEM 7310-A75 |
| Logic gate count | 9152 | 75520 |
| Transfer speeds to PC | 34.38 MiB/s (36 MB/S) | 340 MiB/s |
| External memory | 30.52 MiB (32 Mb) SPI Flash | 1024 MiB DDR3 |
| Clock speed | 80kHz – 150MHz flexible | 200MHz Fixed |

Question 4.

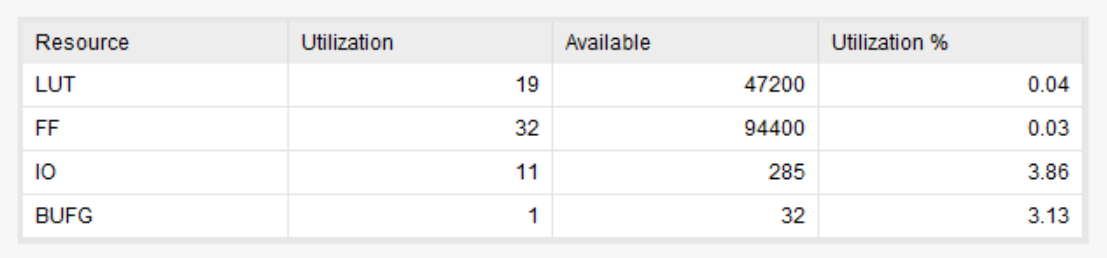
This is to facilitate division of 10,000,000, and 24 bit is the minimum bits length required

Question 5.

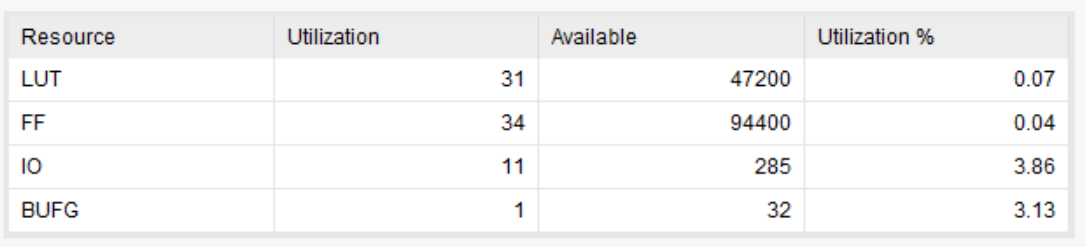
(200MHz/255)/2 = 392.16kHz

Question 6.

Check Point 1A



Check Point 1B



Question 7.

Check Point 1A

`timescale 1ns / 1ps

module intro(

input [3:0] button,

output [7:0] led,

input sys\_clkn,

input sys\_clkp

);

reg [23:0] clkdiv;

reg [7:0] counter;

reg slow\_clk;

// This section defines the main system clock from two

//differential clock signals: sys\_clkn and sys\_clkp

// Clk is a high speed clock signal running at ~200MHz

wire clk;

IBUFGDS osc\_clk(

.O(clk),

.I(sys\_clkp),

.IB(sys\_clkn)

);

initial begin

clkdiv = 0;

counter = 8'h00;

end

assign led = ~counter;

// This code creates a slow clock from the high speed Clk signal

// You will use the slow clock to run your finite state machine

// The slow clock is derived from the fast 20 MHz clock by dividing it 10,000,000 time

// Hence, the slow clock will run at 2 Hz

always @(posedge clk) begin

clkdiv <= clkdiv + 1'b1;

if (clkdiv == 10000000) begin

slow\_clk <= ~slow\_clk;

clkdiv <= 0;

end

end

//The main code will run fr0m the slow clock. The rest of the code will be in this section.

//The counter will decrement when button 0 is pressed and on the rising edge of the slow clk

//Otherwise the counter will increment

always @(posedge slow\_clk) begin

if (button [0] == 1'b0) begin

counter <= 0;

end

else begin

if (counter == 100) begin

counter <= 100;

end

else begin

counter <= counter + 10;

end

end

end

endmodule

Check Point 1B:

`timescale 1ns / 1ps

module intro(

input [3:0] button,

output [7:0] led,

input sys\_clkn,

input sys\_clkp

);

reg [23:0] clkdiv;

reg [7:0] counter;

reg slow\_clk;

reg flag;

// This section defines the main system clock from two

//differential clock signals: sys\_clkn and sys\_clkp

// Clk is a high speed clock signal running at ~200MHz

wire clk;

IBUFGDS osc\_clk(

.O(clk),

.I(sys\_clkp),

.IB(sys\_clkn)

);

initial begin

clkdiv = 0;

flag = 0;

counter = 8'h00;

end

assign led = ~counter;

// This code creates a slow clock from the high speed Clk signal

// You will use the slow clock to run your finite state machine

// The slow clock is derived from the fast 20 MHz clock by dividing it 10,000,000 time

// Hence, the slow clock will run at 2 Hz

always @(posedge clk) begin

clkdiv <= clkdiv + 1'b1;

if (clkdiv == 10000000) begin

slow\_clk <= ~slow\_clk;

clkdiv <= 0;

end

end

//The main code will run fr0m the slow clock. The rest of the code will be in this section.

//The counter will decrement when button 0 is pressed and on the rising edge of the slow clk

//Otherwise the counter will increment

always @(posedge slow\_clk) begin

if (button [0] == 1'b0) begin

counter <= counter - 1'b1;

end

else begin

if (counter == 100) begin

flag <= 1'b1;

counter <= counter - 10;

end

else if(counter == 0) begin

flag <= 1'b0;

counter <= counter + 10;

end

else begin

if (flag == 1'b0) begin

counter <= counter + 10;

end

else begin

counter <= counter - 10;

end

end

end

end

endmodule