**Verilog Code:** lab2\_example.v

`timescale 1ns / 1ps

module lab2\_example(

input wire [4:0] okUH,

output wire [2:0] okHU,

inout wire [31:0] okUHU,

inout wire okAA,

input wire sys\_clkn,

input wire sys\_clkp,

input wire reset,

// Your signals go here

input [3:0] button,

output [7:0] led

);

wire okClk; //These are FrontPanel wires needed to IO communication

wire [112:0] okHE; //These are FrontPanel wires needed to IO communication

wire [64:0] okEH; //These are FrontPanel wires needed to IO communication

//Declare your registers or wires to send or recieve data

wire [31:0] clkdivFreq; //signals that are outputs from a module must be wires

wire counter\_reset;

//This is the OK host that allows data to be sent or recived

okHost hostIF (

.okUH(okUH),

.okHU(okHU),

.okUHU(okUHU),

.okClk(okClk),

.okAA(okAA),

.okHE(okHE),

.okEH(okEH)

);

//Depending on the number of outgoing endpoints, adjust endPt\_count accordingly.

//In this example, we have 2 output endpoints, hence endPt\_count = 2.

localparam endPt\_count = 2;

wire [endPt\_count\*65-1:0] okEHx;

okWireOR # (.N(endPt\_count)) wireOR (okEH, okEHx);

// Clock

wire clk;

reg [31:0] clkdiv;

reg slow\_clk;

reg [7:0] counter;

IBUFGDS osc\_clk(

.O(clk),

.I(sys\_clkp),

.IB(sys\_clkn)

);

initial begin

clkdiv = 0;

slow\_clk = 0;

end

// variable\_1 is a wire that contains data sent from the PC to FPGA.

// The data is communicated via memeory location 0x00

okWireIn wire10 ( .okHE(okHE),

.ep\_addr(8'h00),

.ep\_dataout(clkdivFreq));

okWireIn wire11 ( .okHE(okHE),

.ep\_addr(8'h01),

.ep\_dataout(counter\_reset));

// 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 200 MHz clock by dividing it 10,000,000 time and another 2x

// Hence, the slow clock will run at 10 Hz

always @(posedge clk) begin

clkdiv <= clkdiv + 1'b1;

if (clkdiv == clkdivFreq) begin

slow\_clk <= ~slow\_clk;

clkdiv <= 0;

end

end

assign led = ~counter;

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

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

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

always @(posedge slow\_clk) begin

if ((button [0] == 1'b0) && (button[1]==1'b0 || button[2]==1'b0 || button[3]==1'b0)) begin

counter <= counter + 1'b0;

end

else if ((button [1] == 1'b0) && (button[0]==1'b0 || button[2]==1'b0 || button[3]==1'b0)) begin

counter <= counter + 1'b0;

end

else if ((button [2] == 1'b0) && (button[1]==1'b0 || button[0]==1'b0 || button[3]==1'b0)) begin

counter <= counter + 1'b0;

end

else if ((button [3] == 1'b0) && (button[1]==1'b0 || button[2]==1'b0 || button[0]==1'b0)) begin

counter <= counter + 1'b0;

end

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

counter <= 8'hFF;

end

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

counter <= 8'h00;

end

else if (button [2] == 1'b0) begin

if (counter > 8'hFD) begin

counter <= 8'h00;

end

else

counter <= counter + 2'b10;

end

else if (button [3] == 1'b0) begin

counter <= counter - 2'b10;

if (counter < 8'h02) begin

counter <= 8'h00;

end

else

counter <= counter - 2'b10;

end

if (counter\_reset == 1) begin

counter = 8'h00;

end

end

// result\_wire is transmited to the PC via address 0x20

okWireOut wire20 ( .okHE(okHE),

.okEH(okEHx[ 0\*65 +: 65 ]),

.ep\_addr(8'h20),

.ep\_datain(counter));

endmodule

**Python Code:** lab2\_example\_python.py

# -\*- coding: utf-8 -\*-

#%%

# import various libraries necessery to run your Python code

import time # time related library

import sys # system related library

ok\_loc = 'C:\\Program Files\\Opal Kelly\\FrontPanelUSB\\API\\Python\\3.6\\Win32'

sys.path.append(ok\_loc) # add the path of the OK library

import ok # OpalKelly library

#%%

# Define FrontPanel device variable, open USB communication and

# load the bit file in the FPGA

dev = ok.okCFrontPanel() # define a device for FrontPanel communication

SerialStatus=dev.OpenBySerial("") # open USB communicaiton with the OK board

ConfigStatus=dev.ConfigureFPGA("C:\\Users\\nlao2\\Lab2\\Lab2.runs\\impl\_1\\lab2\_example.bit"); # Configure the FPGA with this bit file

# Check if FrontPanel is initialized correctly and if the bit file is loaded.

# Otherwise terminate the program

print("----------------------------------------------------")

if SerialStatus == 0:

print ("FrontPanel host interface was successfully initialized.")

else:

print ("FrontPanel host interface not detected. The error code number is:" + str(int(SerialStatus)))

print("Exiting the program.")

sys.exit ()

if ConfigStatus == 0:

print ("Your bit file is successfully loaded in the FPGA.")

else:

print ("Your bit file did not load. The error code number is:" + str(int(ConfigStatus)))

print ("Exiting the progam.")

sys.exit ()

print("----------------------------------------------------")

print("----------------------------------------------------")

#%%

while (True):

clkdivFreq = int(input("What value do you want the clock divider?\n"))

if (clkdivFreq > 0 and clkdivFreq <= 4000000000):

break

print("\nInvalid value\n")

dev.SetWireInValue(0x00, clkdivFreq)

dev.UpdateWireIns()

while(True):

time.sleep(1)

dev.UpdateWireOuts()

counter = dev.GetWireOutValue(0x20)

print("The value of the counter is " + str(int(counter)))

if (counter > 100):

dev.SetWireInValue(0x01, 1)

dev.UpdateWireIns()

print("Resetting counter")

time.sleep(clkdivFreq/200000000 + 0.0001)

dev.SetWireInValue(0x01, 0)

dev.UpdateWireIns()

dev.Close

#%%