Part 1:

Function Code:

module freqDivider1(clk, reset, out);

input wire clk, reset;

output wire out;

reg [31:0] count\_r; // current value

wire [31:0] count\_n; // next value

always@(posedge clk)

begin

if(reset)

count\_r <= 0;

else

count\_r <= count\_n;

end

assign count\_n = count\_r + 1;

assign out = count\_r[9];

endmodule

Test Code:

//=======================================================

// This code is generated by Terasic System Builder

//=======================================================

module freqDividerTest(

//////////// CLOCK //////////

input CLOCK\_50,

input CLOCK2\_50,

input CLOCK3\_50,

//////////// LED //////////

output [8:0] LEDG,

output [17:0] LEDR,

//////////// KEY //////////

input [3:0] KEY,

//////////// GPIO, GPIO connect to GPIO Default //////////

inout [35:0] GPIO

);

//=======================================================

// REG/WIRE declarations

//=======================================================

wire reset, clk, my\_signal;

assign reset = 1'b0;

assign clk = CLOCK\_50;

assign GPIO[0] = my\_signal;

//=======================================================

// Structural coding

//=======================================================

freqDivider1 U1(.clk(clk), .reset(reset), .out(my\_signal));

endmodule

Output:



Part 2:

Function Code:

module freqDivider2(clk, reset, out);

input wire clk, reset;

output wire out;

reg [31:0] count\_r; // current value

wire [31:0] count\_n; // next value

always@(posedge clk)

begin

if(reset)

count\_r <= 0;

else

count\_r <= count\_n;

end

assign count\_n = count\_r + 1;

assign out = count\_r[18];

endmodule

Test Code:

//=======================================================

// This code is generated by Terasic System Builder

//=======================================================

module freqDividerTest2(

//////////// CLOCK //////////

input CLOCK\_50,

input CLOCK2\_50,

input CLOCK3\_50,

//////////// LED //////////

output [8:0] LEDG,

output [17:0] LEDR,

//////////// KEY //////////

input [3:0] KEY,

//////////// GPIO, GPIO connect to GPIO Default //////////

inout [35:0] GPIO

);

//=======================================================

// REG/WIRE declarations

//=======================================================

wire reset, clk, my\_signal;

assign reset = 1'b0;

assign clk = CLOCK\_50;

assign GPIO[0] = my\_signal;

//=======================================================

// Structural coding

//=======================================================

freqDivider2 U1(.clk(clk), .reset(reset), .out(my\_signal));

endmodule

Output:

