EGC221: Digital Logic Lab

Experiment #9

Sequential Logic Design Using Verilog

Using Quartus Prime

Student's Name:	Reg. no.:
Student's Name:	Reg. no.:
Semester: Spring 2021	Date: 14 April 2021

Assessment:

Assessment Point	**Weight**	**Grade**
Methodology and correctness of results		
Discussion of results		
Participation		
Assessment Points' Grade:		

Comments:

Experiment #9:

Finite State Machines

Objectives:

The objective of this lab is to:

1. Design, simulate, and build a Divide-by-N circuit.
2. Design, simulate, and implement a sequential circuit with *D* flip-flops.

Procedure:

Exercise 1: Divide-by-N Circuit:

Design, simulate, and build a Divide-by-N circuit that will divide the on board clock from 50 MHz down to ~1 Hz. The basic principle is as follows:

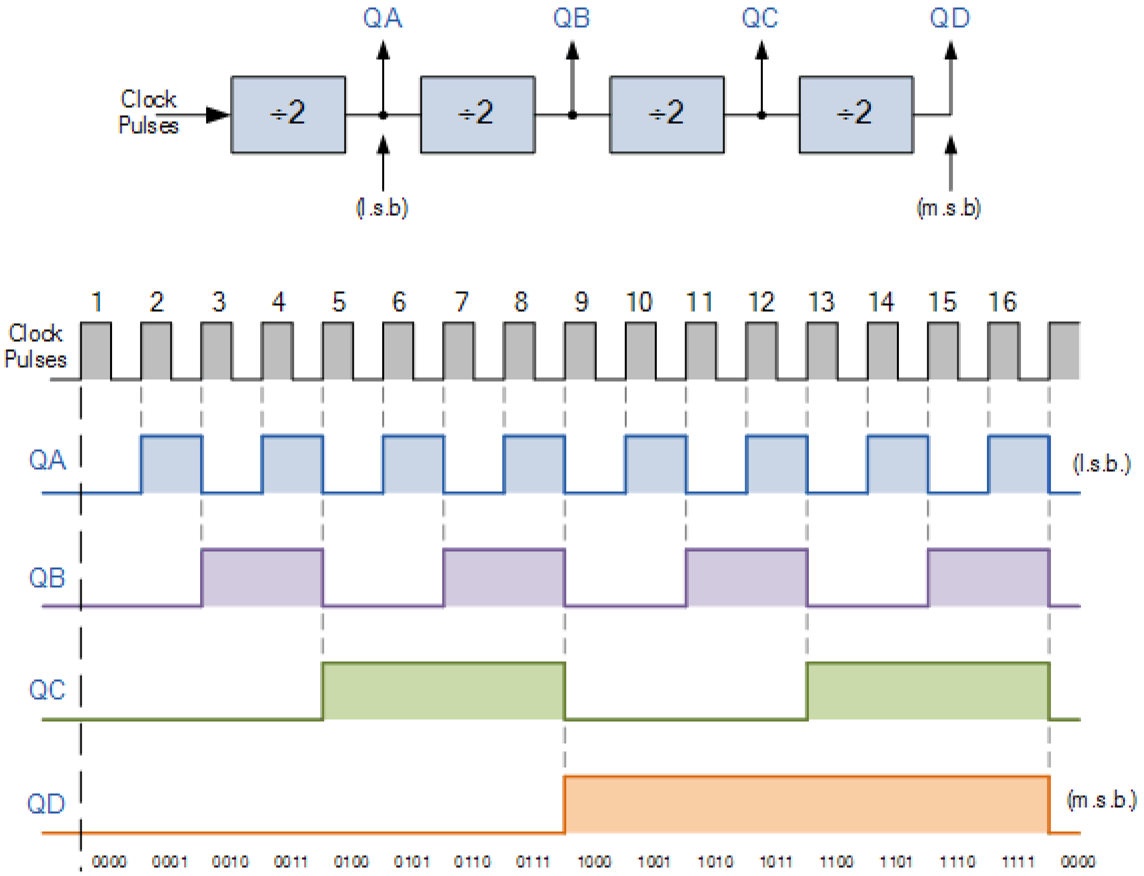

Figure 1. Divide-by-N (frequency divider) circuit

So, to convert a N clock cycles into one cycle, one needs to keep the output low for N/2 cycles and high for the other N/2 cycles. So, in converting 50 MHz (50,000,000 Hz) to 1 Hz, Clock output needs to be kept low for 25,000,000 cycles and then high for the other 25,000,000 cycles. Implement the following code and verify the circuit operation by blinking an LED at the divided rate.

```
//The goal of this always procedural block is to generate 1Hz clock from a
//50MHz clock that is used in the Altera FPGA board.
module Divide_by_50M_counter(clr,clk,clk_1Hz);
input clr,clk;
output clk_1Hz;
reg clk_1Hz =1'b0;
integer counter_50M =0;
always @(posedge clk, posedge clr)
begin
  if (clr)
      counter_50M <=0;
  else if (counter_50M <25000000)
    begin
        counter_50M <= counter_50M + 1;
    end
  else if (counter_50M ==25000000)
     begin
        clk_1Hz <= !clk_1Hz;
        counter_50M <=0;
    end
end
endmodule
```

Figure 2 shows the clock circuit of DE10 Board, the crystal 24 MHz buffered to four input clocks. The two 50MHz clock signals connected to the FPGA are used as clock sources for user logic. One 24MHz clock signal is connected to the clock inputs of USB microcontroller of USB Blaster. One 10MHz clock signal is connected to the PLL1 and PLL3 of FPGA, the outputs of these two PLLs can drive ADC clock.

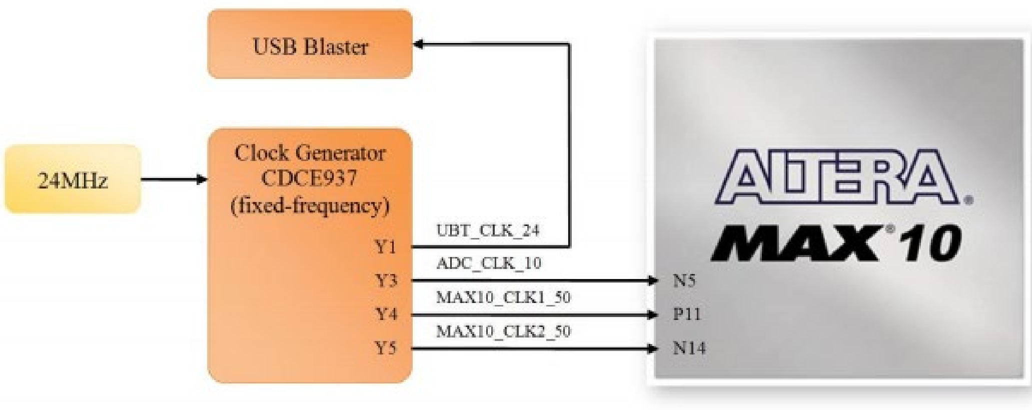

Figure 2: Clock Circuit of the Altera FPGA Board

The associated pin assignment for clock inputs to FPGA I/O pins is listed in Table 1.

Table 1: Pin Assignment of Clock Inputs

Signal Name	*FPGA Pin No.*	*Description*
ADC_CLk_50	PIN_N5	10 MHz clock input for ADC (Bank 3B)
MAX10_CLK1_50	PIN_P11	50 MHz clock input (Bank 3B)
MAX10_CLK2_50	PIN_N14	50 MHz clock input (Bank 3B)

Exercise 2:
You are to design a 4-bit counter with the following inputs and functionality:

- Load (ld): if activated, count will be loaded from D_in [3:0]
- Mode: if 0, counter counts up. Otherwise, it will count down.
- Clear (clr): If activated, count will be 0
- Clock 1 Hz is generated from the previous exercise.

Complete the following code and implement it on the FPGA board.

```
module up_down_counter(mode,clr,ld,D_in,clk,count,clk_1Hz);
input mode,clr,ld,clk;
input [3:0] D_in;
output clk_1Hz;
output [3:0] count;
reg [3:0] count;
reg clk_1Hz =1'b0;
integer counter_50M =0;

//The goal of this always procedural block is to generate 1Hz clock from a
//50MHz clock that is used in the Altera FPGA board.

always @(posedge clk)
begin
 if (clr)
    counter_50M <=0;
 else if (counter_50M <25000000)
   begin
      counter_50M <= counter_50M + 1;
   end
 else if (counter_50M ==25000000)
    begin
      clk_1Hz <= !clk_1Hz;
      counter_50M <=0;
   end
 end
```

```
//If "ld =1" we load the external data through D_in[3:0], if mode is active
// it will be counting up and if mode is inactive it will count down.

always @(posedge clk_1Hz, posedge clr)

.
.
.

endmodule
```

Conclusions: