ENGG 125.03 Laboratory Activity 4: Counter

Activity Sheet; Submitted: 5 May 2021

Mallari, Kevin Luis T.
BS Computer Engineering
Electronics, Computer, and Communications Engineering
Ateneo de Manila University
Philippines

***Abstract*—The laboratory activity is about counters. A counter is a digital circuit that counts to a certain value. In the activity, a specific counter is simulated. The IC is the 74190. The counter has 3 inputs: the clock, direction, and reset. The clock is responsible for the clock input of the counter. The direction is responsible for telling the counter whether to count up or down. Lastly, the reset input allows the reset of the values in the counter.**

Keywords - Counter, ModelSim, Quartus, 74190

I. HDL Code

This section shows and explains the code made for the activity.

```
module counter(clock, reset, direction, q,
terminal_count);
parameter N=3; // variable used to
determine number of bits
input clock, reset, direction;
output terminal_count;
output [N-1:0] q;
reg [N-1:0] q;

assign terminal_count = direction ? &q :
~|q;

initial q = 0;
always @(posedge clock)
  if (reset == 0) q <= 0;
  else
    if (direction == 1) q <= q + 1;
    else q <= q - 1;
endmodule

// test bench code
module test_bench();
parameter N=3; // variable used to
determine number of bits
reg clock, reset, direction;
wire [N-1:0] q;
wire terminal_count;

counter dut(clock,reset, direction, q,
terminal_count);

always #10 clock = ~clock; // invert value
of clock every 10 time units

initial
begin
  // default values for clock, reset, and
direction
  clock=0;
  reset=0;
  direction=1;

  // set values for reset and direction at
a certain time
  #50        reset = 1;
  #100       reset = 0;
  #110 reset = 1;
  #150       direction = 0;
  #200       $stop; //stop the simulation
end

endmodule
```

Fig. 1. Code for the 3-to-8 Decoder

The parameter N is used to easily change the number of bits the output has. In the case of the code, it is set to 3 since the counter has to count up to 3 bits since it is a decade counter. The inputs of the program are clock, reset, and direction. The clock input serves as the clock of the counter, the reset input resets the counter, and the direction input dictates whether the counter counts up or down.

The value of terminal_count depends on the direction. If direction is equal to 1, then the value of the terminal count is

the AND reduction of q. Otherwise, the value would be the NOT OR reduction of q.

The always and posedge is used on the clock so that any transitions will only be done on the positive edge of the clock. If reset is equal to 0, the output will simply be 0. Otherwise, the program checks the value of direction. If it is 1, then it increments q by 1. Otherwise, it decrements q by 1.

For the test bench code, the initial values of the inputs were first instantiated. Afterwards, at certain times, these values are modified. The test bench ends at 500 time units.

II. Output Circuit

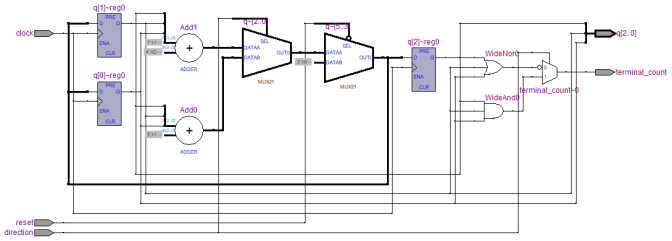

Fig. 2. Circuit for the Counter

The circuit in figure 2 is the output circuit of the entire code. Multiplexers, registers, AND gates, and OR gates were added to the circuit in their appropriate places.

III. Screenshot(s) of Simulations and VWF file

The screenshots of the simulations are presented below.

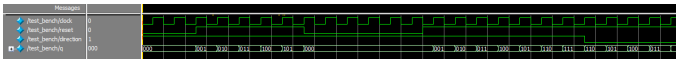

Fig. 3. Simulation for the Decoder

As seen in the screenshot, the value of the output, q, if the value of reset is 0. Otherwise, it enables the counter. If the value of direction is 1, the counter counts up and counts down otherwise.