**Task -1**

**Verilog :**

/\*

**Name-Shadeeb Hossain**

**ID: sh7492**

**This is a Verilog file that generates a counter Up .**

It resets on the first digit of your N Number

The following were provided:

clk-1 bit: Design clock

rst-1 bit: Asynchronous active low reset

cnt-ent-1 bit: Enable for the counter

If 1 counter will be counting up or down.

0 it should be stopped at the most recent output.

cnt-6 bits: Counte output as a 6 bit unsigned vector

\*/

/\*

ChatGPT Prompt:

Hi, Can you please design a verilog file that generates a counter UP ,

resets to the first digit of your N number.

The following should be used clk- design clock, rst-1bit- Asynchronous active low reset.

cnt\_ent-2 bit to enable for the counter, If 1 counter will be counting up or down. 0 it should be stopped at its most recent output.

cnt-6 bits: Counter output as a 6 bit unsigned vector

\*/

module up\_counter (

input wire clk, // Clock input

input wire rst, // Asynchronous active low reset

input wire [1:0] cnt\_ent, // 2-bit enable for the counter (01: up, 10: down, 00: stop)

output wire [5:0] cnt // 6-bit counter output

);

reg [5:0] count; // 6-bit counter register

always @(posedge clk or negedge rst) begin

if (!rst) begin

count <= 6'b000000; // Reset the counter to 0

end else begin

case (cnt\_ent)

2'b01: count <= count + 1; // Count up

2'b10: count <= count - 1; // Count down

default: count <= count; // Hold the current value

endcase

end

end

assign cnt = count;

endmodule

/\*

In this Verilog module:

We have a 6-bit counter register called count to store the counter value.

The counter is sensitive to both the positive edge of the clock (clk) and the negative edge of the reset signal (rst).

When the reset signal is low (rst is active low), the counter is reset to 0.

Depending on the value of cnt\_ent (2-bit enable for the counter), the counter can either count up (01), count down (10), or stop (00).

The counter output is provided as a 6-bit unsigned vector through the cnt output wire.

You can instantiate this module in your larger Verilog design and connect the clock, reset, cnt\_ent, and cnt signals as needed.

\*/

**VHDL File**

--Name-Shadeeb Hossain

--ID: sh7492

--This is a VHDL code that generates a counter Down .

--It resets on the last digit of your N Number

--The following were provided:

--clk-1 bit: Design clock

--rst-1 bit: Asynchronous active low reset

--cnt-ent-1 bit: Enable for the counter

--If 1 counter will be counting up or down.

--0 it should be stopped at the most recent output.

--cnt-6 bits: Counte output as a 6 bit unsigned vector

--ChatGPT Prompt:

--Write a VHDL code that performs count down, resets to the last digit of your N number.

--The following should be used clk- design clock, rst-1bit- Asynchronous active low reset.

--cnt\_ent-2 bit to enable for the counter, If 1 counter will be counting up or down.

--0 it should be stopped at its most recent output.

--cnt-6 bits: Counter output as a 6 bit unsigned vector

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity down\_counter is

Port (

clk : in STD\_LOGIC; -- Clock input

rst : in STD\_LOGIC; -- Asynchronous active low reset

cnt\_ent : in STD\_LOGIC\_VECTOR(1 downto 0); -- 2-bit enable for the counter (01: down, 10: up, 00: stop)

cnt : out STD\_LOGIC\_VECTOR(5 downto 0) -- 6-bit counter output

);

end down\_counter;

architecture Behavioral of down\_counter is

signal count : STD\_LOGIC\_VECTOR(5 downto 0); -- 6-bit counter signal

begin

process (clk, rst)

begin

if rst = '0' then

count <= "000000"; -- Reset the counter to 0

elsif rising\_edge(clk) then

case cnt\_ent is

when "01" =>

if count = "000000" then

count <= "111111"; -- Reset to the last digit of N

else

count <= count - 1; -- Count down

end if;

when "10" =>

count <= count + 1; -- Count up

when others =>

null; -- Hold the current value

end case;

end if;

end process;

cnt <= count;

end Behavioral;

--We have an entity called down\_counter with the specified ports, including clk, rst, cnt\_ent, and cnt.

--The counter is implemented using a process sensitive to the rising edge of the clock (clk) and the asynchronous active low reset signal (rst).

--Depending on the value of cnt\_ent (2-bit enable for the counter), the counter can count down (01), count up (10), or stop (00).

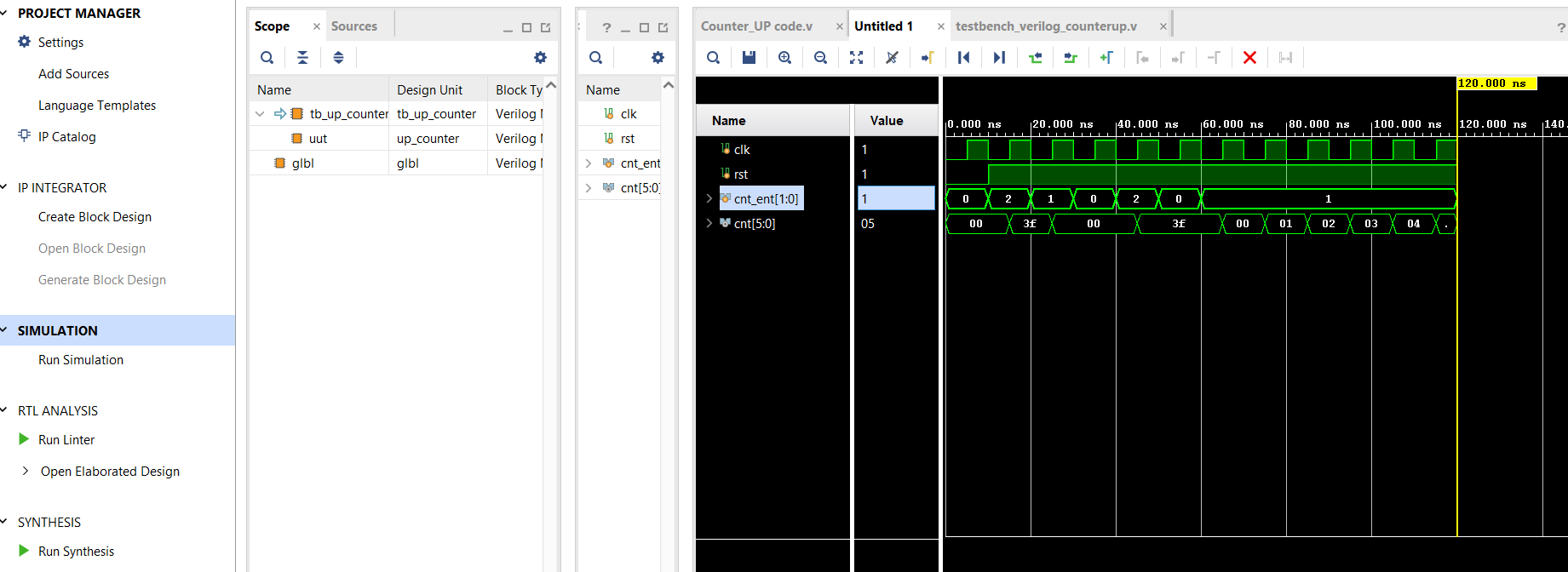
--When counting down, if the counter reaches 0, it resets to "111111," which corresponds to the last digit of N.

--The counter output is provided as a 6-bit unsigned vector through the cnt signal.

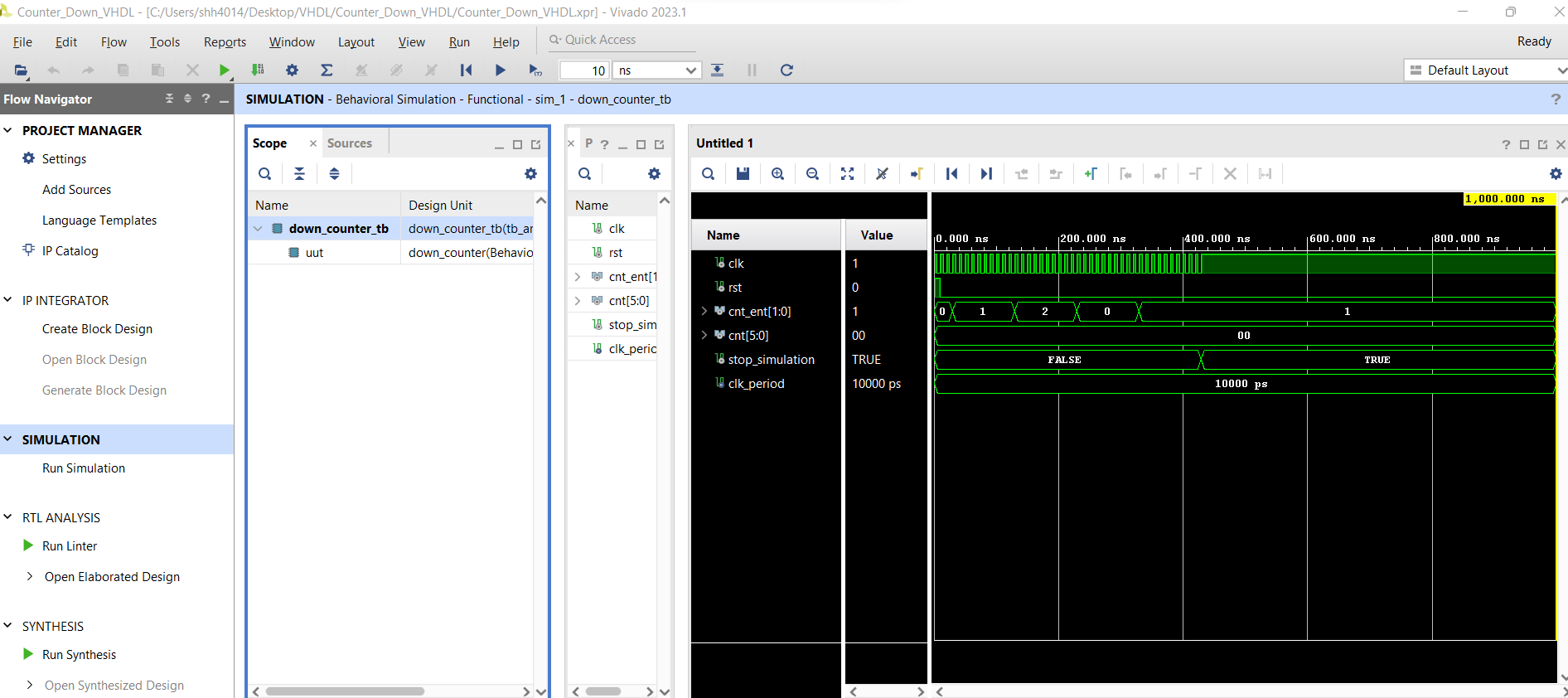
--You can instantiate this module in your larger VHDL design and connect the clock, reset, cnt\_ent, and cnt signals as needed.

**Task-2**

**(a)**

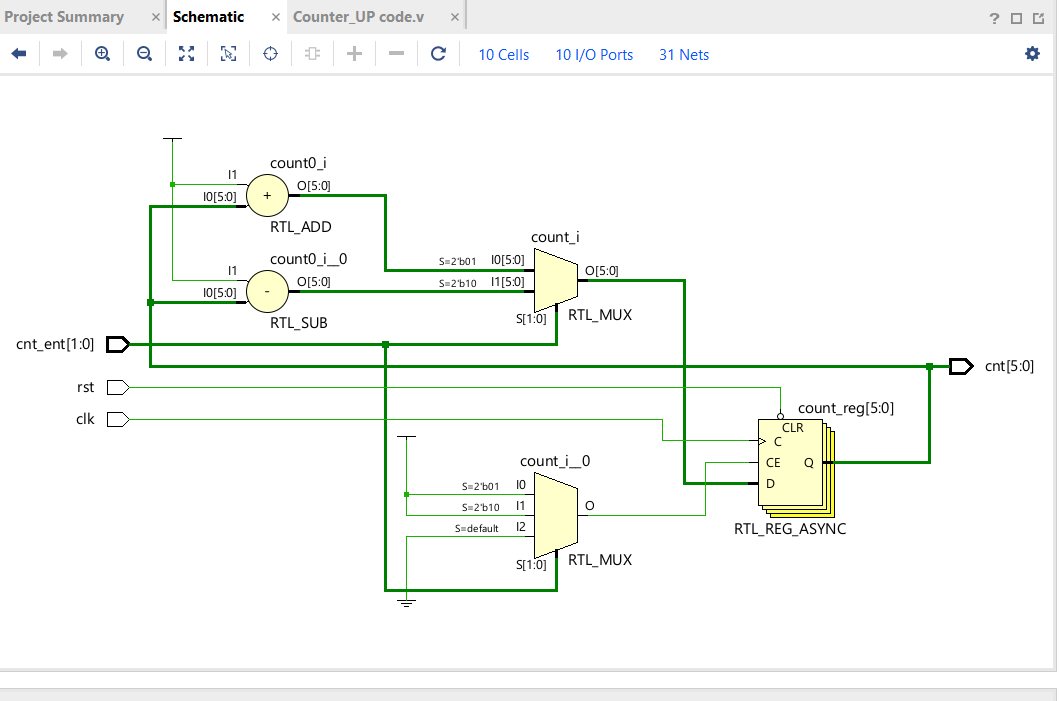
****

**Fig. 2(a) : Functional Simulation model for Up counter (Verilog)**

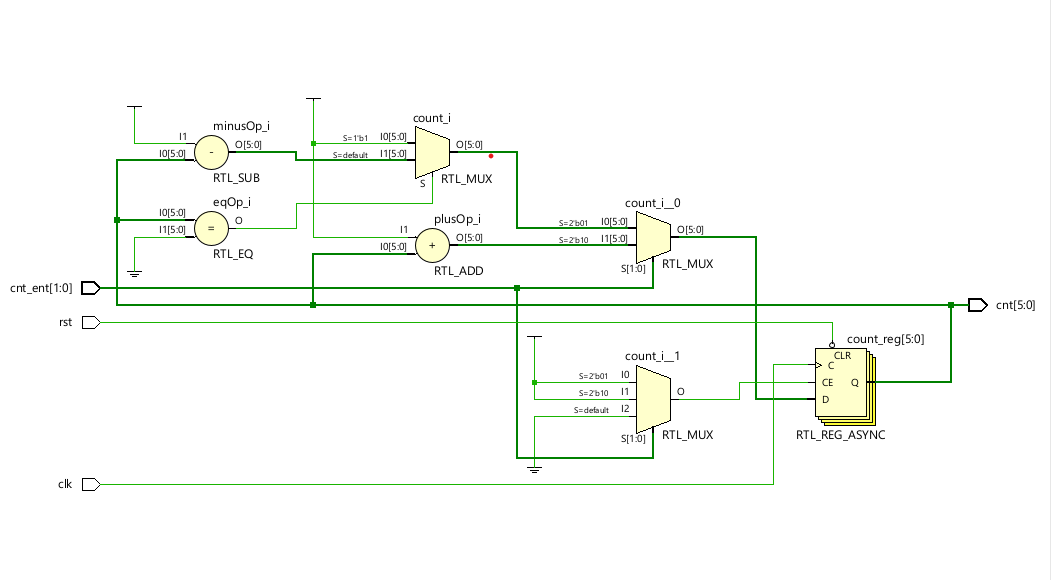
****

**Fig.2 (b) : Functional Simulation model for Down counter (VHDL)**

**Task-3**

****

**Fig. 3 (a) : Schematic for elaborated design for Up counter (Verilog)**

****

**Fig. 3 (b) : Schematic for elaborated design for Down counter (VHDL)**