Saif Ba Madhaf - 527028

VHDL Week 4

Self-test chapter 9:

1. Which statements can easily generate inferred latches?

Statements like cases and nested ifs generate inferred latches easily.

2. What type of latch is generated by the syntheses tool if an inferred latch is generated?

It generates a level-triggered latch when an inferred latch is generated.

Self-test chapter 10:

1. Why do you need internal signals in the VHDL in the given 8-bits counter?

It makes the code less complex by breaking the functionalities into smaller pieces, which then makes it easier to read.

2. How can you give the counter a preset value?

You can give the counter a preset value by 2 ways:

1. By initializing it like this in the beginning.

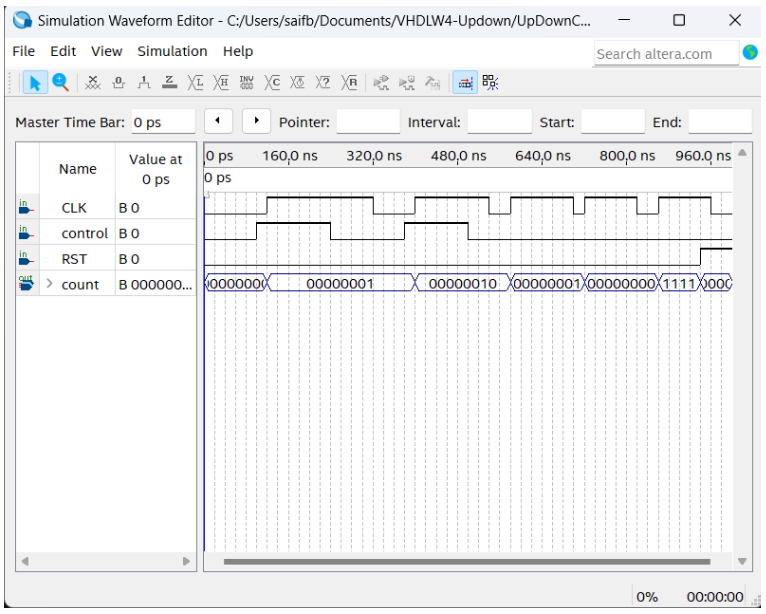
signal counter : STD_LOGIC_VECTOR (3 downto 0) := "0000";

2. By using the reset input and resesting the values to 0 when reses equals 1.

```
if (RST = '1') then
        counter <= "0000";
```

Exercise chapter 10:

1.Write in VHDL and simulate an 8 bit up/down counter.


Simulation Waveform Editor - C:/Users/saifb/Documents/VHDLW4-Updown/UpDownC...
File Edit View Simulation Help
Search altera.com
0 ps
160,0 ns
320,0 ns
480,0 ns
640,0 ns
800,0 ns
960,0 ns
0%
00:00:00

Master Time Bar: 0 ps Pointer: Interval: Start: End:

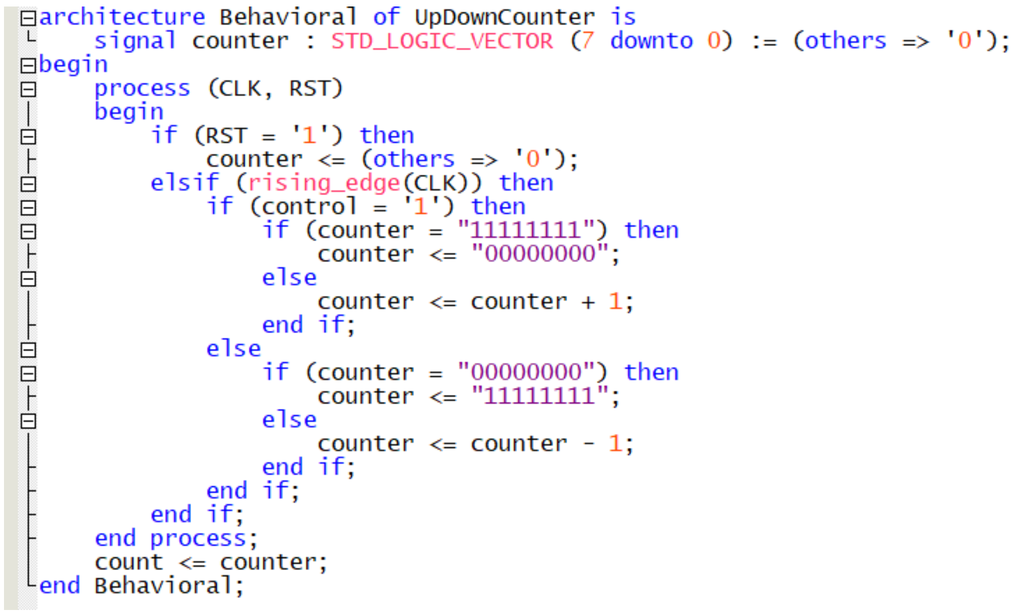
Name	Value at 0 ps
CLK	B 0
control	B 0
RST	B 0
count	B 000000...

count: 0000000 / 00000001 / 00000010 / 00000001 / 00000000 / 1111 / 000

As we can see in the simulation above the circuit works. There are 3 inputs clock, control, and reset. After every rising edge the counter starts counting, and if the control is 0 it means that it is a subtraction and if it is a 1 then it means that it is adding. When the rest is 1 the whole counter resets to 0. Additionally, there is an 8-bit output which is the count.

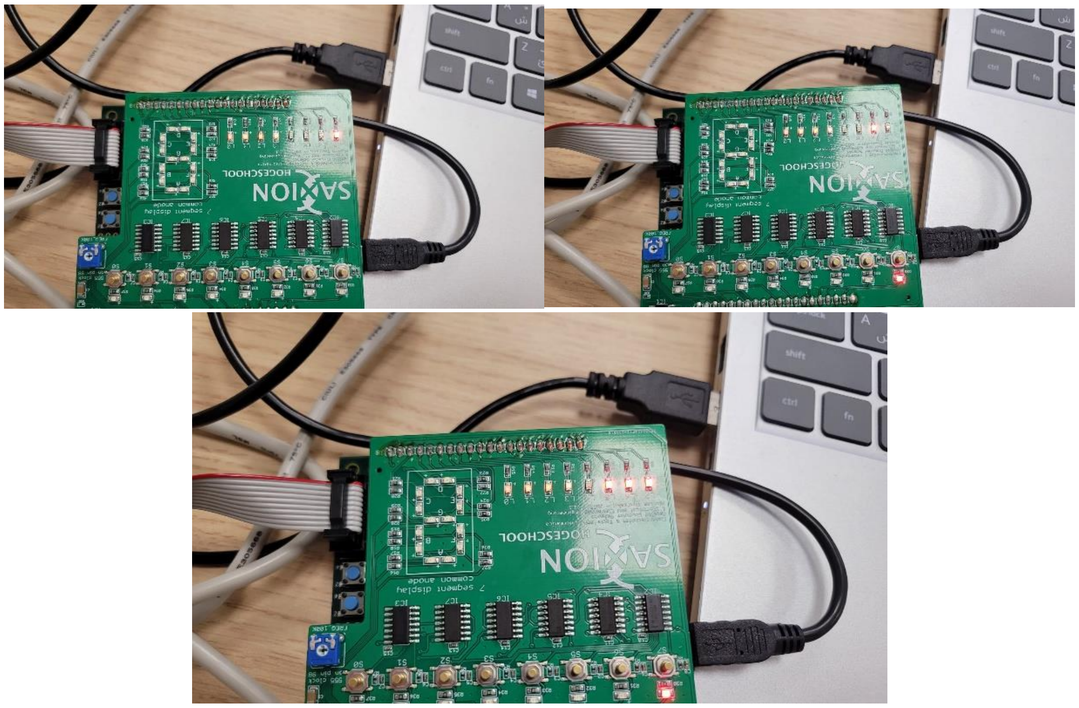
In this example, we can see that we added a 1 and another one after wards which made the counter equal 00000010 in binary which is 2 in decimal. Then we subtracted 3 which is -1 which makes the circuit go back to all ones. Finally, we made the reset 1 to reset the count to 0.

CODE



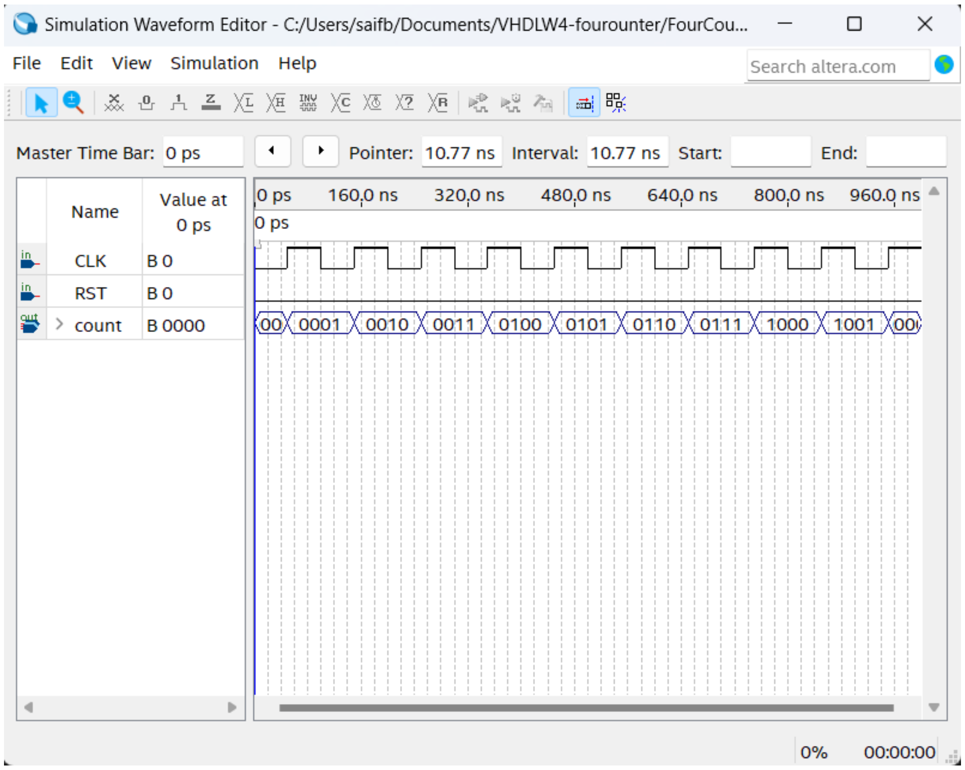
```
architecture Behavioral of UpDownCounter is
    signal counter : STD_LOGIC_VECTOR (7 downto 0) := (others => '0');
begin
    process (CLK, RST)
    begin
        if (RST = '1') then
            counter <= (others => '0');
        elsif (rising_edge(CLK)) then
            if (control = '1') then
                if (counter = "11111111") then
                    counter <= "00000000";
                else
                    counter <= counter + 1;
                end if;
            else
                if (counter = "00000000") then
                    counter <= "11111111";
                else
                    counter <= counter - 1;
                end if;
            end if;
        end if;
    end process;
    count <= counter;
end Behavioral;
```

2. Simulate your design and implement it on the MAX10 board



Here we can see an example of the up/down counter going from 1 (00000001) to 3 (00000011) using the clock (s7) at each rising edge.

3. Write in VHDL a 4 bit Synchronous Decade counter


Simulation Waveform Editor - C:/Users/saifb/Documents/VHDLW4-fourounter/FourCou...

Master Time Bar: 0 ps   Pointer: 10.77 ns   Interval: 10.77 ns   Start: End:

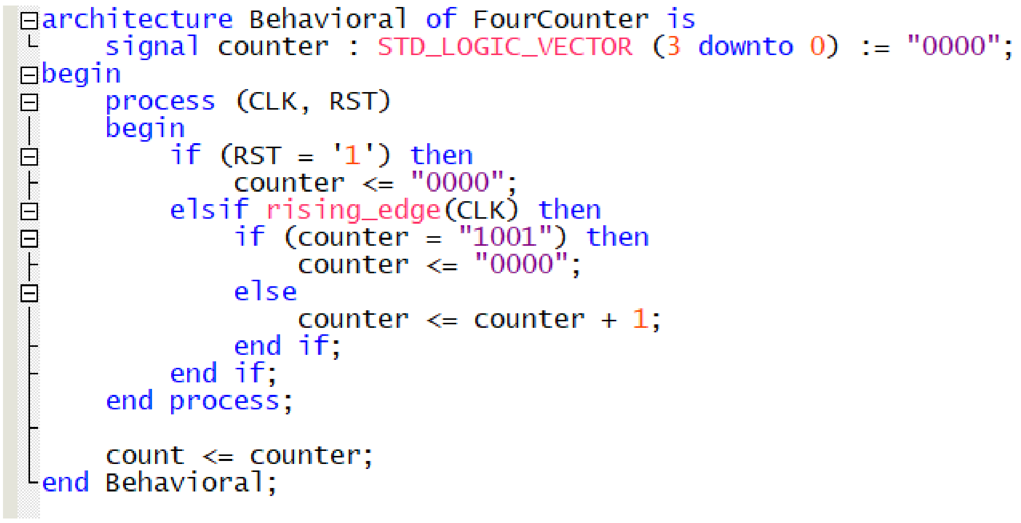
Name	Value at 0 ps
CLK	B 0
RST	B 0
count	B 0000

```
0 ps   160,0 ns   320,0 ns   480,0 ns   640,0 ns   800,0 ns   960,0 ns
count: 00 0001 0010 0011 0100 0101 0110 0111 1000 1001 00
```

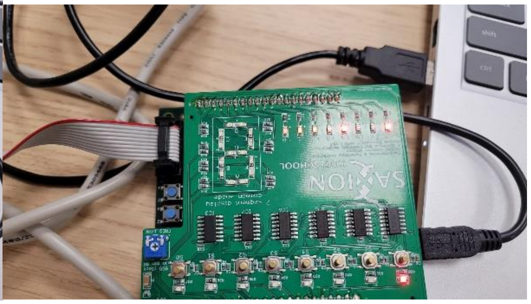
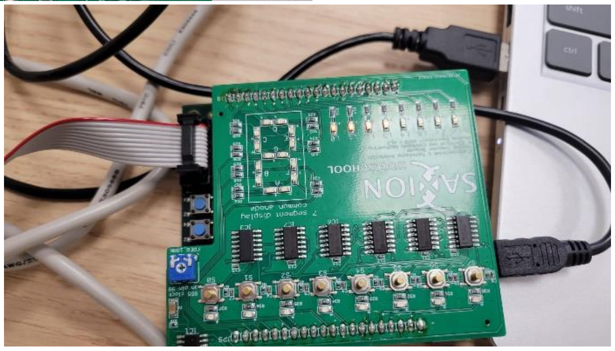
0%   00:00:00

As we can see there are three inputs (clock, reset, and count (4 bits)). When there is a rising edge in the clock the counter starts counting to 9 and resets to 0 when exceeding 9. The reset is used to reset the count to 0 when its 1.

CODE

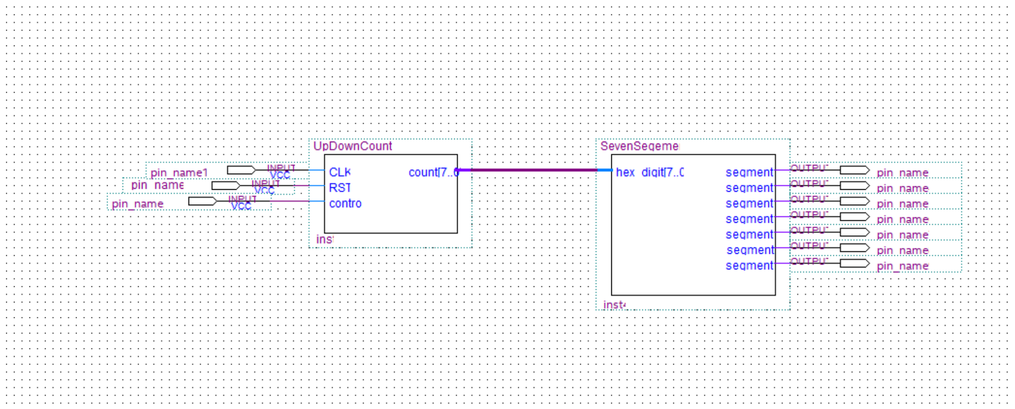

```
architecture Behavioral of FourCounter is
    signal counter : STD_LOGIC_VECTOR (3 downto 0) := "0000";
begin
    process (CLK, RST)
    begin
        if (RST = '1') then
            counter <= "0000";
        elsif rising_edge(CLK) then
            if (counter = "1001") then
                counter <= "0000";
            else
                counter <= counter + 1;
            end if;
        end if;
    end process;

    count <= counter;
end Behavioral;
```

4. Simulate your design and implement it on the MAX10 board

In this example, we made the clock up to 9 then back to zero using the clock (s7) at each rising edge. As shown above there are three images one 8 (1000) then to 9 (1001) and then back to 0 (0000).

5. Create a symbol for the counter (up/down) and for the 7-segment decoder.


UpDownCount
pin_name1
pin_name
pin_name
INPUT VCC
CLK
RST
contro
count[7..0]
ins
SevenSegeme
hex_digit[7..0]
segment
OUTPU
pin_name
inst