Cp319 Final Project

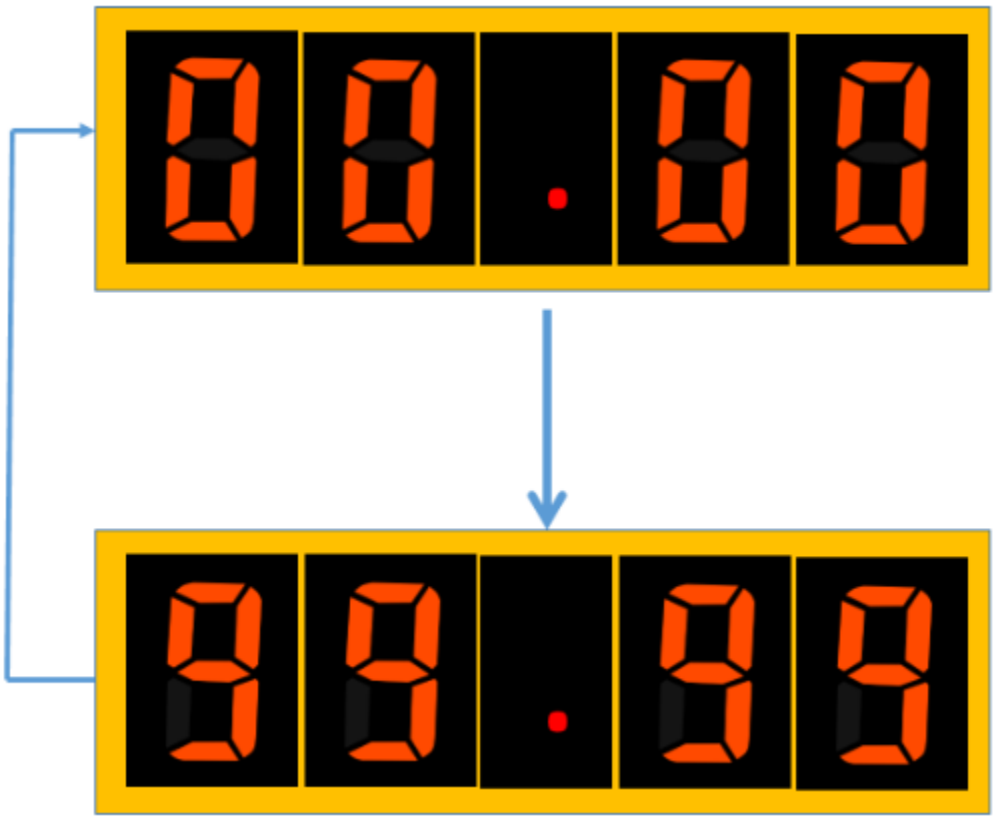
Instructor: Shaowen Song

2020/12/18

Hongshan Shang163042150

Dekai Meng186800570

In this project, we design a 5 digit 7-segment to display dollars and cents, from $00.00 to $99.99 and then go back to $00.00.



The code is:

```
LIBRARY ieee;
   USE ieee.std_logic_1164.all;
   USE ieee.std_logic_unsigned.all;
-- Cent, dollar accumulation control module
ENTITY data_cnt IS
   PORT (
      clk     : IN STD_LOGIC;-- clock
      reset     : IN STD_LOGIC;--reset
      dollars : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--dollar
      cents   : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)--cent
   );
END data_cnt;

ARCHITECTURE behave OF data_cnt IS
   SIGNAL data1_H : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
   SIGNAL data1_L : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
   SIGNAL data2_H   : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
   SIGNAL data2_L   : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
BEGIN

   -- Dollar control
   PROCESS (clk, reset)
   BEGIN
      IF (reset = '1') THEN--reset
         data1_H <= "0000";
         data1_L <= "0000";
      ELSIF (clk'EVENT AND clk = '1') THEN
         IF (data1_H = "1001" AND data1_L = "1001" AND data2_H = "1001" AND
data2_L = "1001") THEN-- The cents count to 99. The dollar counts to 99, back to 0
            data1_H <= "0000";--back to 0
            data1_L <= "0000";-- back to 0
         ELSIF (data1_L = "1001" AND data2_H = "1001" AND data2_L = "1001")
THEN
            data1_H <= data1_H + "0001";-- Tens digit of dollar plus 1
            data1_L <= "0000";--single digit return to 0
         ELSIF (data2_H = "1001" AND data2_L = "1001") THEN----cents count to 99
            data1_H <= data1_H;
            data1_L <= data1_L + "0001";-- single digit of dollar plus 1
         ELSE
            data1_H <= data1_H;
            data1_L <= data1_L;
         END IF;
      END IF;
   END PROCESS;

   --cents
   PROCESS (clk, reset)
   BEGIN
      IF (reset = '1') THEN--reset
```

```
         data2_H <= "0000";-- Tens digit of cents
         data2_L <= "0000";-- single digit of cents
      ELSIF (clk'EVENT AND clk = '1') THEN
         IF (data2_H = "1001" AND data2_L = "1001") THEN--cents count to 99
            data2_H <= "0000";--return 0
            data2_L <= "0000";--return 0
         ELSIF (data2_L = "1001") THEN--single digit of cents count to 9
            data2_H <= data2_H + "0001";--tens digit add 1
            data2_L <= "0000";-- single digit return to 0
         ELSE
            data2_H <= data2_H;
            data2_L <= data2_L + "0001";--single digit add 1
         END IF;
      END IF;
   END PROCESS;

   dollars <= data1_H & data1_L;--dollar
   cents <= data2_H & data2_L;--cents

END behave;
```

The segment.vhd file is our Digital tube control module.
we design five variables SEG1 to SEG5 to represent 5 digit in the 7-segment.

```
LIBRARY ieee;
   USE ieee.std_logic_1164.all;

ENTITY segment IS
   PORT (
      dollars  : IN STD_LOGIC_VECTOR(7 DOWNTO 0);--Dollar
      cents    : IN STD_LOGIC_VECTOR(7 DOWNTO 0);--Cent
      SEG1     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 1
      SEG2     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 2
      SEG3     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 3
      SEG4     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 4
      SEG5     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) --digit 5
   );
END segment;



ARCHITECTURE behave OF segment IS
BEGIN

   PROCESS (dollars)
   BEGIN
      CASE dollars(7 downto 4) IS--first digit of dollar
         WHEN "0000" =>SEG1 <= "11000000";--0
         WHEN "0001" =>SEG1 <= "11111001";--1
         WHEN "0010" =>SEG1 <= "10100100";--2
         WHEN "0011" =>SEG1 <= "10110000";--3
         WHEN "0100" =>SEG1 <= "10011001";--4
         WHEN "0101" =>SEG1 <= "10010010";--5
         WHEN "0110" =>SEG1 <= "10000010";--6
         WHEN "0111" =>SEG1 <= "11111000";--7
         WHEN "1000" =>SEG1 <= "10000000";--8
         WHEN "1001" =>SEG1 <= "10010000";--9
         WHEN OTHERS =>
      END CASE;
      CASE dollars(3 downto 0) IS--Second digit of dollar
         WHEN "0000" =>SEG1 <= "11000000";--0
         WHEN "0001" =>SEG1 <= "11111001";--1
         WHEN "0010" =>SEG1 <= "10100100";--2
         WHEN "0011" =>SEG1 <= "10110000";--3
         WHEN "0100" =>SEG1 <= "10011001";--4
         WHEN "0101" =>SEG1 <= "10010010";--5
         WHEN "0110" =>SEG1 <= "10000010";--6
         WHEN "0111" =>SEG1 <= "11111000";--7
         WHEN "1000" =>SEG1 <= "10000000";--8
         WHEN "1001" =>SEG1 <= "10010000";--9
         WHEN OTHERS =>
      END CASE;
   END PROCESS;
```

```
   SEG3 <= "01111111";-- Display decimal point

   PROCESS (cents)
   BEGIN
      CASE cents(7 downto 4) IS-- first digit of Cents
         WHEN "0000" =>SEG1 <= "11000000";--0
         WHEN "0001" =>SEG1 <= "11111001";--1
         WHEN "0010" =>SEG1 <= "10100100";--2
         WHEN "0011" =>SEG1 <= "10110000";--3
         WHEN "0100" =>SEG1 <= "10011001";--4
         WHEN "0101" =>SEG1 <= "10010010";--5
         WHEN "0110" =>SEG1 <= "10000010";--6
         WHEN "0111" =>SEG1 <= "11111000";--7
         WHEN "1000" =>SEG1 <= "10000000";--8
         WHEN "1001" =>SEG1 <= "10010000";--9
         WHEN OTHERS =>
      END CASE;
      CASE cents(3 downto 0) IS-- second digit of Cents
         WHEN "0000" =>SEG1 <= "11000000";--0
         WHEN "0001" =>SEG1 <= "11111001";--1
         WHEN "0010" =>SEG1 <= "10100100";--2
         WHEN "0011" =>SEG1 <= "10110000";--3
         WHEN "0100" =>SEG1 <= "10011001";--4
         WHEN "0101" =>SEG1 <= "10010010";--5
         WHEN "0110" =>SEG1 <= "10000010";--6
         WHEN "0111" =>SEG1 <= "11111000";--7
         WHEN "1000" =>SEG1 <= "10000000";--8
         WHEN "1001" =>SEG1 <= "10010000";--9
         WHEN OTHERS =>
      END CASE;
   END PROCESS;

END behave;


LIBRARY ieee;
   USE ieee.std_logic_1164.all;
ENTITY top IS
   PORT (
      clk   : IN STD_LOGIC;-- clock
      reset : IN STD_LOGIC;--reset
      SEG1  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 1
      SEG2  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 2
      SEG3  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 3
      SEG4  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 4
      SEG5  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) --digit 5
   );
END top;

ARCHITECTURE behave OF top IS
--Cent USD accumulation control module
COMPONENT data_cnt IS
   PORT (
```

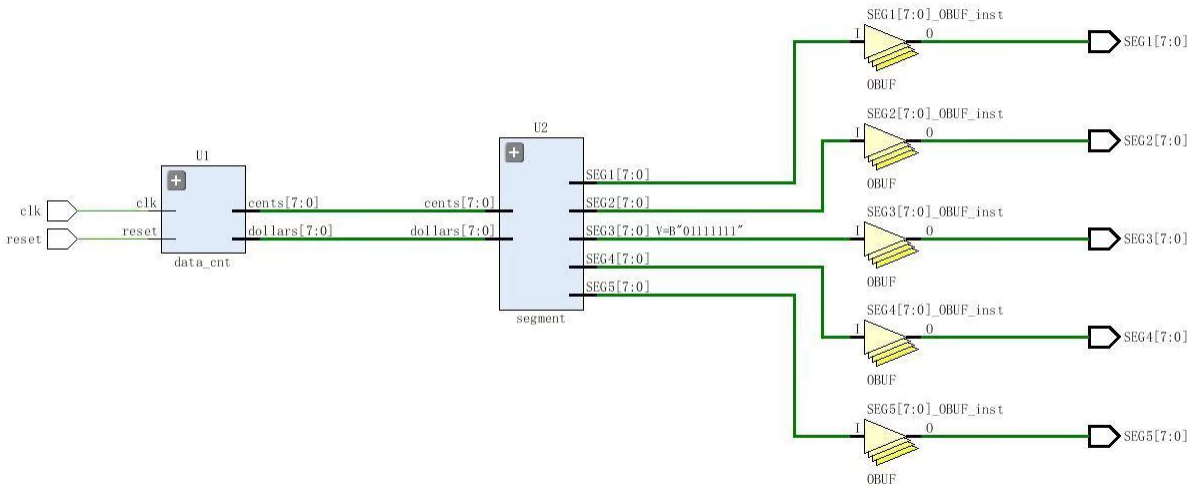
```
      clk     : IN STD_LOGIC;--clock
      reset     : IN STD_LOGIC;--reset
      dollars : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--dollar
      cents   : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)--cents
   );
END COMPONENT;

--digit 控制模块
COMPONENT segment IS
   PORT (
      dollars  : IN STD_LOGIC_VECTOR(7 DOWNTO 0);--dollar
      cents    : IN STD_LOGIC_VECTOR(7 DOWNTO 0);--cents
      SEG1     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 1
      SEG2     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 2
      SEG3     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 3
      SEG4     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 4
      SEG5     : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) --digit 5
   );
END COMPONENT;

   SIGNAL dollars : STD_LOGIC_VECTOR(7 DOWNTO 0);
   SIGNAL cents : STD_LOGIC_VECTOR(7 DOWNTO 0);

BEGIN
--cents, dollar accumulation
   U1 : data_cnt
      PORT MAP (
         clk    => clk,--clock
         reset  => reset,--reset
         dollars=> dollars,--dollar
         cents  => cents--cents
      );

--digit control
   U2 : segment
      PORT MAP (
         dollars  => dollars,--dollar
         cents  => cents,--cents
         SEG1   => SEG1,--digit 1
         SEG2   => SEG2,--digit 2
         SEG3   => SEG3,-- digit 3
         SEG4   => SEG4,-- digit 4
         SEG5   => SEG5 -- digit 5
      );

END behave;
```

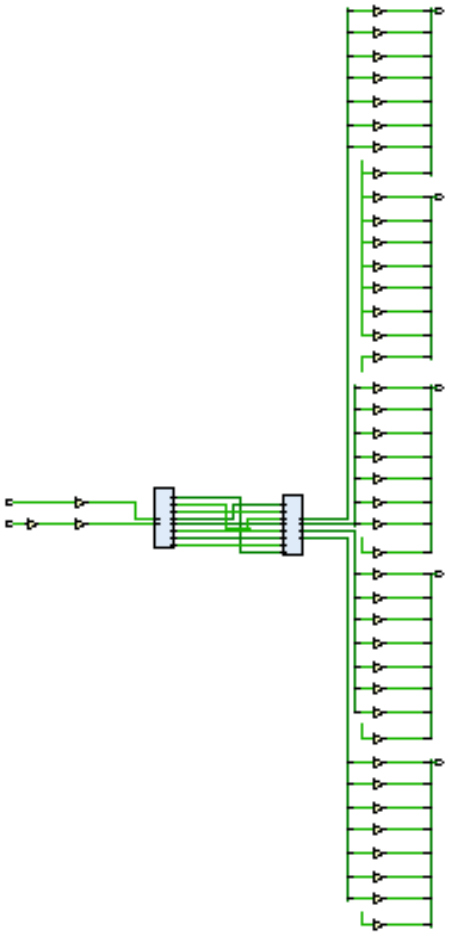
The test function is:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--testbench
entity testbench is
end testbench;

architecture behave of testbench is

   COMPONENT top IS
      PORT (
      clk   : IN STD_LOGIC;--clock
      reset : IN STD_LOGIC;--reset
      SEG1  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);-- digit1
      SEG2  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);-- digit 2
      SEG3  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 3
      SEG4  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);--digit 4
      SEG5  : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)--digit 5
      );
   END COMPONENT;

signal   clk   : STD_LOGIC;
signal   reset : STD_LOGIC;
signal   SEG1  : STD_LOGIC_VECTOR(7 DOWNTO 0);
signal   SEG2  : STD_LOGIC_VECTOR(7 DOWNTO 0);
signal   SEG3  : STD_LOGIC_VECTOR(7 DOWNTO 0);
signal   SEG4  : STD_LOGIC_VECTOR(7 DOWNTO 0);
signal   SEG5  : STD_LOGIC_VECTOR(7 DOWNTO 0);
begin

   UUT : top
      PORT MAP (
         clk    => clk   ,--clock
         reset  => reset ,--reset
         SEG1   => SEG1  ,--digit 1
         SEG2   => SEG2  ,--digit 2
         SEG3   => SEG3  ,--digit 3
         SEG4   => SEG4  ,--digit 4
         SEG5   => SEG5   --digit 5
      );

   --reset
   PROCESS
   BEGIN
    reset<='1';--reset
    wait for 100 ns;
    reset<='0';
    wait;
   END PROCESS;

   --clock
   PROCESS
```

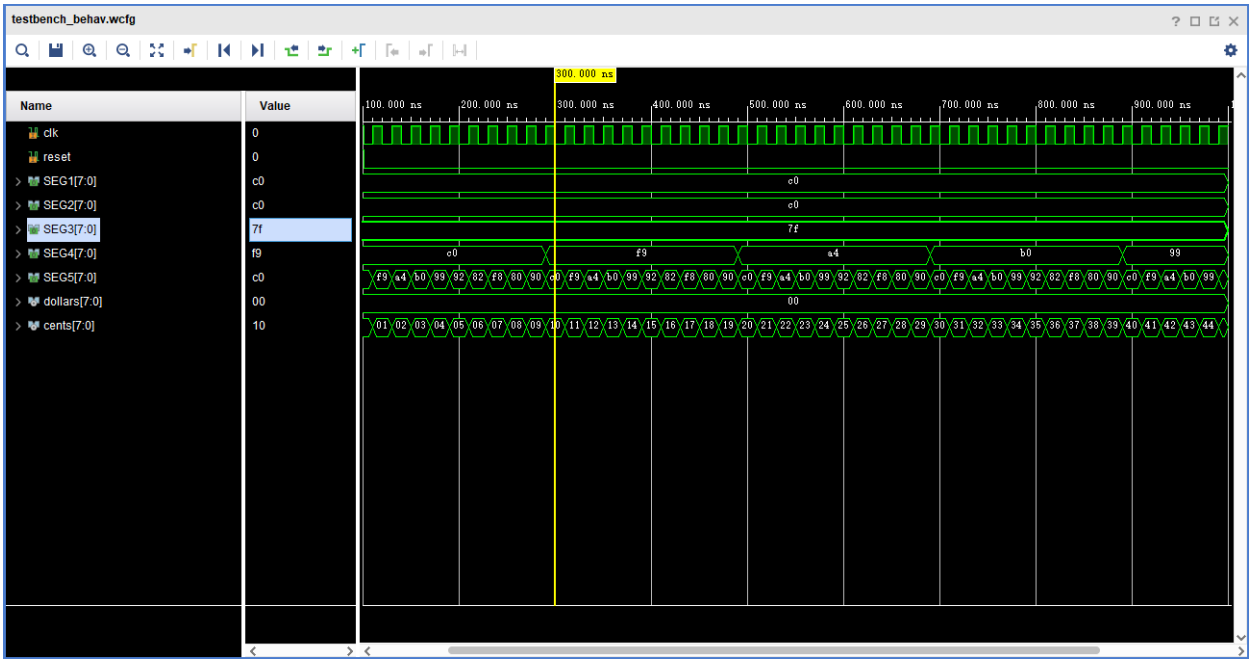
```
    BEGIN
     clk<='0';
     wait for 10 ns;
     clk<='1';
     wait for 10 ns;
    END PROCESS;

end behave;
```

RTL diagram is:

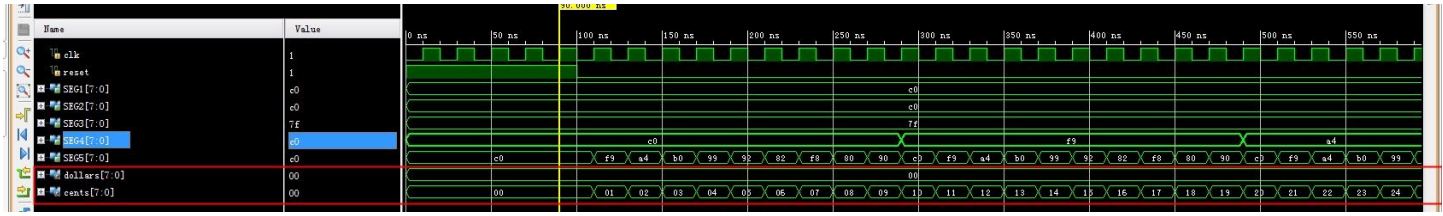




The simulation diagram is:
From the diagram you can see 5 digital tube display, Signals representing cents and dollars.



The red box shows the accumulation of cents.



This red box shows that when the cents are added to 99, it becomes 1 dollar. The signal representing cents changed from 99 to 00, and the signal representing US dollars changed from 00 to 01.