EASTERN MEDITERRANEAN UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
CMPE224 **DIGITAL LOGIC SYSTEMS**
VHDL EXPERIMENT VI

TITLE: VHDL IMPLEMENTATION OF REGISTERS AND COUNTERS

OBJECTIVES: Implementation of counters in VHDL will be studied. The students are expected to learn the implementation issues of synchronous and asynchronous counters.

Below, you will find detailed examples on the implementation of different types of counters in VHDL. Examine these VHDL codes and implement them in Quartus II environment to understand the underlying ideas better.

Example 1: 4-bit binary up-counter with Reset and Enable.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;   -- required for unsigned arithmetic

entity counter_4Bit_RE is
port (Clock, Reset, Enable : in std_logic;
      Q                    : out std_logic_vector (3 downto 0);
end Counter_4Bit_RE;

architecture Behavioral of counter_4Bit_RE is
signal count: std_logic_vector (3 downto 0);

begin
process (Reset, Clock) is
  begin
      if (Reset='1') then
            Count <= "0000";
      elseif  (Clcok'event and Clock = '1') then
            if Enable = '1'
                  Count<=Count+'1';
            else
                  Count <=Count;
            end if;
      end if;
  end process;
  Q<= Count;
end Behavioral;
```

Example 2: The behavioral code that represents an 8-bit up/down counter with parallel load and asynchronous reset is as follows.

```
LIBRARY ieee ;
USE ieee.std logic 1164.all ;
USE ieee.std logic unsigned.all ;

ENTITY UpDown_Counter8 IS
PORT ( R : IN STD_LOGIC_VECTOR(7 DOWNTO 0) ;  -- Parallel input register
        Clock, Resetn, L, U : IN STD_LOGIC ;
                    -- L: Parallel load
                    -- U=1: count up, U=0: Count down
        Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) ) ;  -- count register
END UpDown_Counter8;

ARCHITECTURE Behavior OF UpDown_Counter8 IS
signal Count: std_logic_vector (7 downto 0);

BEGIN
PROCESS ( Clock, Resetn )
BEGIN
        IF Resetn = '0' THEN
                Count <= (OTHERS => '0') ;
        ELSIF (Clock'EVENT AND Clock = '1') THEN
                IF L = '1' THEN                 -- parallel load
                        Count <= R ;
                ELSIF U = '1' THEN
                        Count <= Count+1 ;              -- count up
                ELSE
                        Count <= Count-1 ;              -- count down.
                END IF ;
        END IF ;
END PROCESS ;
Q<=Count;
END Behavior ;
```

Example 3: The behavioral code that represents a modulo-12 up/down counter with synchronous reset is as follows.

```
LIBRARY ieee ;
USE ieee.std logic 1164.all ;

ENTITY Modulo12_Counter IS
PORT ( R : IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;  -- Parallel input register
        Clock, Resetn, L: IN STD_LOGIC ;
                -- L: Parallel load
        Q : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) ) ;  -- count register
END Modulo12_Counter;

ARCHITECTURE Behavior OF Modulo12_Counter IS
signal Count: std_logic_vector (3 downto 0);

BEGIN
PROCESS ( Clock, Resetn )
BEGIN
        IF Resetn = '0' THEN
                Count <= (OTHERS => '0') ;
        ELSIF (Clock'EVENT AND Clock = '1') THEN
                IF L = '1' THEN                 -- parallel load
                        Count <= R ;
                ELSIF Count = "1011" THEN
                        Count <= "0000" ;               -- synchronous reset
                ELSE
                        Count <= Count+1 ;              -- count down.
                END IF ;
        END IF ;
END PROCESS ;
Q<=Count;
END Behavior ;
```

Example 4: VHDL code that represents an 8-bit Johnson counter is given below.

```
LIBRARY ieee ;
USE ieee.std logic 1164.all ;

ENTITY Johnson_Count8 IS
PORT ( Clock, Resetn : IN STD_LOGIC ;
        Q : Out STD_LOGIC_VECTOR(0 TO 7) ) ;
END Johnson_Count8;

ARCHITECTURE Behavior OF Johnson_Count8 IS
signal Count: std_logic_vector (0 to 7);

BEGIN
PROCESS ( Clock, Resetn )
BEGIN
        IF Resetn = '0' THEN
                Count <= "00000000" ;
        ELSIF Clock'EVENT AND Clock = '1' THEN
                Count<= (NOT Count(7)) & Count(0 TO 6) ;
        END IF ;
END PROCESS ;
Q<=Count;
END Behavior ;
```

PRELIMINARY WORK:

Write down the VHDL code for an 8-bit down counter with parallel load. Perform all experimental steps including VHDL coding, waveform preparation, and simulations; make them ready before you come to the laboratory.

EXPERIMENTAL WORK:

Demonstrate your preliminary work to the laboratory assistants using Quartus II environment. Be ready for detailed questions on your work.

Good Luck.

Dr. Adnan Acan
Dr. Evgueni Doukhnitch
Dr. Muhammed Salamah.