‘Work sheet2’

Ahmad Dar Khalil 1120443

Ehab Amriah 1120871

Rafat Ahmad 1120521

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL ;
ENTITY count is
        generic (n : positive :=5) ;
        port(
        clk : in std_logic ;
        reset : in std_logic ;
        q : inout  std_logic_vector (n-1 downto 0)
        );
        end ;
        architecture num1 of count is
        begin
                process(clk , reset)
                begin
                        if (reset = '1') then

                                q <=(others =>'0') ;
                        elsif (rising_edge(clk)) then
                                  q(n-1 downto 0) <= not q(0) &q(n-1 downto 1);
                                end if ;
                        end process;
                end architecture num1;
```

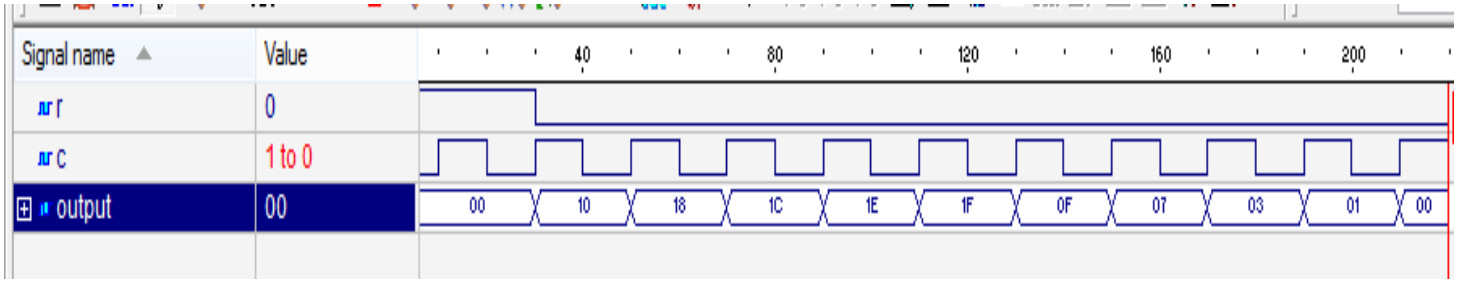
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL ;

entity test is
end ;

architecture testbench of test is
signal r,c: std_logic :='0' ;
signal output :std_logic_vector(4 downto 0);
begin
        g: entity work.count(n2)
                generic map(5)
                port map(clk =>c,reset=>r , q =>output) ;

                c <= not c after 10 ns;


                r <='1',
                '0' after 30 ns;



end     ;
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL ;

entity dff is
        port (clk , reset , d : in std_logic;
        q : out std_logic );
end  ;



ARCHITECTURE dffs OF dff IS
BEGIN
   PROCESS (clk, reset)
   BEGIN
      IF ( reset='1' ) THEN
         q <= '0';
      ELSIF ( rising_edge(clk) ) THEN
         q <= d;
      END IF;
   END PROCESS;

END ;

architecture n2 of count is
signal x : std_logic;
signal qe : std_logic_vector(n-1 downto 0 );
begin
          x <= not qe(0);
        gr:for i in 1 to n-1 generate
```

```
        g1: entity work.dff(dffs)
                port map (reset =>reset , clk =>clk ,d=> qe(n-i),q=>qe(n-i-1)) ;

        end generate ;

        g1: entity work.dff(dffs)
                port map (reset =>reset , clk =>clk ,d=>x,q=>qe(n-1)) ;

                q<= qe ;

end ;
```

Sample output :

- Structural :



- Behavioral: (same output)