Let us see an example on how to instantiate a component.

--(in a separate file topOne.vhd)

library ieee; -- added topOne entity/architecture

use ieee.std\_logic\_1164.all;entity topOne is port ( clk: in std\_logic; wavetopOne: out std\_logic );end entity;architecture foo of topOne is

signal wavelowTwo: std\_logic;

component lowTwo is

port (

clk: in std\_logic;

wavelowTwo: out std\_logic

);

end component;

begin

lowTwo

port map (

clk => clk,

wavelowTwo => wavelowTwo

);

wavetopOne <= wavelowTwo after 2 us;

end architecture;

--(in a new file lowTwo.vhd)

library ieee; -- added lowTwo entity/architecture

use ieee.std\_logic\_1164.all;

entity lowTwo is

port (

clk: in std\_logic;

wavelowTwo: out std\_logic

);

end entity;

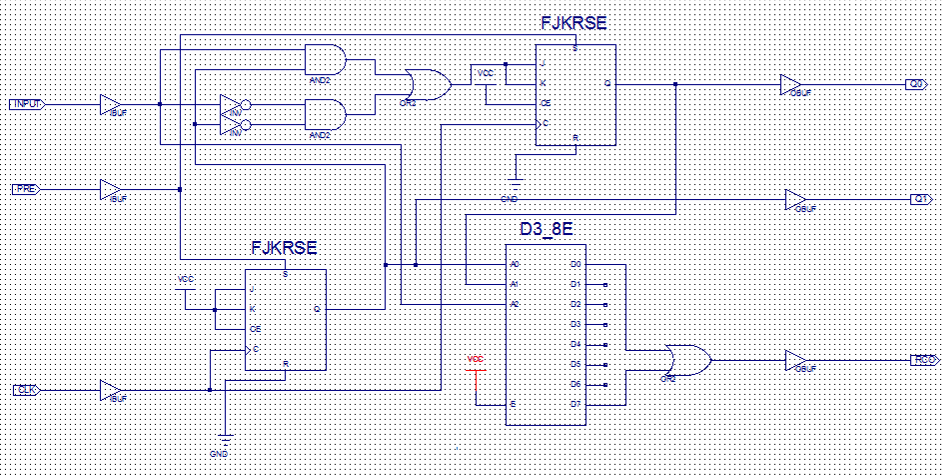
architecture foo of lowTwo is

begin

wavelowTwo <= clk after 1us;

end architecture;

Next, we work on project 2.

****

There are different ways to write the code. Below is one example. You are welcome and encouraged to copy this code into your project to use for Project 2 as long as you read it and understand what the pieces are doing. You must add additional descriptive comments to the code showing your understanding if you copy it.

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY sequential IS

PORT(  Q0           :             OUT      STD\_LOGIC;

           Q1 :              OUT       STD\_LOGIC;

RCO             :              OUT       STD\_LOGIC;

          CLK             :               IN  STD\_LOGIC;

          INPUT        :               IN  STD\_LOGIC;

          CLR             :               IN  STD\_LOGIC);

END sequential;

ARCHITECTURE behavioral OF sequential IS

                SIGNAL CE1        :               STD\_LOGIC := '1';

                SIGNAL CE0        :               STD\_LOGIC;

                SIGNAL IQ0  :     STD\_LOGIC := '0';

                SIGNAL IQ1  :     STD\_LOGIC := '0';

BEGIN

process (IQ0,IQ1,INPUT) -- output 1) the 3-8 decoder 2) Q0 3) Q1

                begin

                                if IQ0 = '1' and IQ1 = '1' and INPUT = '1' then

                                                RCO <= '1';

                                elsif IQ0 = '0' and IQ1 = '0' and INPUT = '0' then

                                                RCO <= '1';

                                else

                                                RCO <= '0';

                                end if;

                                Q0 <= IQ0;

                                Q1 <= IQ1;

                end process;

process (CLK, CLR) -- The first T flip-flop

                begin

                                if (CLK'event and CLK='1') then

                                                if CLR = '1' then

                                                                IQ1 <= '0';

                                                elsif CE1 = '1' then

                                                                IQ1 <= not(IQ1);

                                                end if;

                                end if;

                end process;

                process (CLK, CLR) -- The second T flip-flop

                begin

                                if INPUT = '1' and IQ1 = '1' then

                                                CE0 <= '1';

                                elsif INPUT = '0' and IQ1 = '0' then

                                                CE0 <= '1';

                                else

                                                CE0 <= '0';

                                end if;

                                if CLK'event and CLK = '1' then

                                                if CLR = '1' then

                                                                IQ0 <= '0';

                                                elsif CE0 ='1' then

                                                                IQ0 <= not(IQ0);

                                                end if;

                                end if;

                end process;

END;

How is the circuit simulated? The solution is to use the designed component in the test bench. We create a new file wave.vhd, from which we instantiate an instance of sequential we just defined for the simulation. You need both this test bench code on pages 5-6 in a wave.vhd file AND the behavioral description code on pages 3-4 in a sequential.vhd file.

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

ENTITY wave IS

- - there are no inputs and outputs

END wave;

ARCHITECTURE behavioral OF wave IS

   COMPONENT sequential

   PORT(  Q0         :               OUT       STD\_LOGIC;

           RCO            :               OUT       STD\_LOGIC;

           Q1 :               OUT       STD\_LOGIC;

           CLK              :               IN           STD\_LOGIC;

           INPUT         :               IN           STD\_LOGIC;

           CLR               :               IN           STD\_LOGIC);

   END COMPONENT;

   SIGNAL Q0       :               STD\_LOGIC := '0';

   SIGNAL CLK     :               STD\_LOGIC;

   SIGNAL INPUT:               STD\_LOGIC := '0';

   SIGNAL RCO    :               STD\_LOGIC;

   SIGNAL Q1       :               STD\_LOGIC := '0';

   SIGNAL CLR     :               STD\_LOGIC := '1';

                SIGNAL CE1        :               STD\_LOGIC := '1';

                SIGNAL CE0        :               STD\_LOGIC;

                SIGNAL TEMPORAL : STD\_LOGIC\_VECTOR(5 downto 0):="000000";

                SIGNAL IQ0  :     STD\_LOGIC := '0';

                SIGNAL IQ1  :     STD\_LOGIC := '0';

BEGIN

--Unit under test

uut: sequential PORT MAP(

INPUT => INPUT,

CLR => CLR,

CLK => CLK,

Q0 => Q0,

Q1 => Q1,

RCO=> RCO

);

                -- \*\*\* Test Bench - User Defined Section \*\*\*

               tb : PROCESS

                BEGIN

                   for i in 0 to 50 loop -- VHDL loop statement

                          CLK <= TEMPORAL(0); --assigning the LSB of TEMPORAL to CLK

                          TEMPORAL <= std\_logic\_vector(UNSIGNED(TEMPORAL) + 1);

                                                                -- increment TEMPORAL

                        if (TEMPORAL = "010111") then -- VHDL if statement

                                         INPUT <= '1';

            -- set INPUT=1 when TEMPORAL = "010110",

                          end if;

                          if (TEMPORAL = "000010") then -- VHDL if statement

                                         CLR <= '0';

            -- set CLR=0 after the first clock period

                           end if;

                           wait for 50 ns; -- CLK high/low for 50ns

                      end loop;

                      WAIT;

   END PROCESS;

-- \*\*\* End Test Bench - User Defined Section \*\*\*

END