# **COURSE CODE : : CSC 214**

# **COURSE TITLE : : COMPUTER HARDWARE GROUP NAME : : GROUP D**

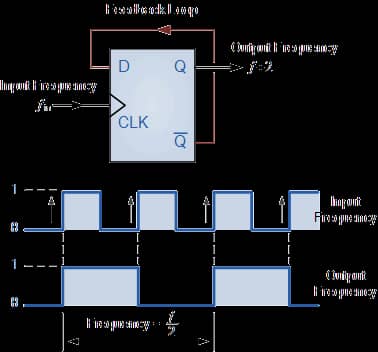
# **PROJECT TOPIC : :**

# **VHDL(VERY HIGH SPEED INTEGRATED CIRCUIT HARDWARE DESCRIPTION LANGUAGE).**

**LECTURER IN CHARGE::MR. ADEGBITE**

# **S/N GROUP MEMBERS MATRIC NO** 1 HASSAN OLUWATOBI ABIODUN 1804040472 OYINSUYI FAVOUR TAIWO 1804040483 YINUZA RILIWAN TEMITOPE 1804040494 SHITTU AYODEJI EMMANUEL 1804040505 BAYO-AWE INIOLUWA MARY 1804040516 OLOFE SHADRACH OLAMIGOKE 1804040527 AWOJULU AYOMIDE MIGHTY 1804040538 OYINLADE DAMILOLA CHARLES 1804040549 ERUE EARNEST AYOWOLE 18040405510 ADEBORO ADEDAMOLA 18040405611 AJAYI DAMILOLA VICTORIA 18040405712 BELLO ABDULRAUF OLATUBOSUN 190404065(DE)13 OLADOYINBO VINCENT BAMIDELE 180404058 14 AKINTOLA TOPE AKINFOLARIN 18040405915 OLUWOLE TOLULOPE DUNMININU 180404060





**Csc 214 presentation for group D**

Question :

The task is to write a VHDL module that implements a circuit which can generate an output clock signal that has a period that is either 3 times or 2 times as much as the period of the input clock signal. The multiplying factor is controlled by the signal F. The doubling of the clock period can be accomplished by defining a signal and then inverting that signal on every rising edge of the input clock. The tripling of the clock period can be accomplished by defining another signal and using a counter. The counter can be incremented by one on every rising and falling edge of the input. clock. When the counter has been incremented enough times to determine 1 and a half clock signals then the signal defined should be inverted and the counter should be set back to zero. The counter also has an asynchronous active high reset signal. Now the period of the first signal will be twice the input clock period and the period of the second signal will be three times the input clock period. These signals can be assigned to the output clock by using an if-statement to check the value of the control signal F.

Solution

Write the VHDL code:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD LOGIC UNSIGNED.ALL;

entity period is port (CLK, F, ClrN: in std\_logic;

CLKout: out std\_logic);

end period;

architecture beh of period is

signal countcur: std\_logic\_vector(1 downto 0) :="00";

signal clkint2: std\_logic :='0';

signal clkint3: std\_logic :=0^ prime ;

begin

process (CLK, ClrN, F)

begin

if ( (ClrN="1) then

countcur <= "00";

elseif CLK'event and CLK = '1' then

countcur <= countcur +1:

clkint2 <= not clkint2;

elsif CLK'event and CLK = '0' then

countcur <= countcur+1;

end if;

end beh;

In order to simulate the VHDL code, first click the simulate button and use the following force commands to set the clock signal and other inputs.

force CLK 0 0,1 10 -repeat 20

force CirN 0 0

force CLK 0 0, 1 250

Comment

Now run the simulation and get the waveform shown in Figure 1.

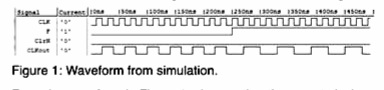


Figure 1: Waveform from simulation.

From the waveform in Figure 1, observe that the output clock signal does indeed have a period that is controlled by F and is a multiple of the input clock's period. When F is 0 the output clock has a period of 40 ns which is twice the input clock period. When Fis 1 the output clock has a period of 60 ns which is three times the input clock period. Therefore, the circuit is functioning properly. Hence, the VHDL code is written.

SIMULATION FROM THE COMPILER BEEN USED

COMPILER USED **Aldec Active-HDL 2021**

