VHDL SYNCHRONUS COUNTER ASSIGNMENT

Name: Shreya Mamadapur Student ID: C0774035

File 1: Code for T Flip Flop

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity ichaya is
   Port ( T : in  STD_LOGIC; Clk : in  STD_LOGIC;CLR : in  STD_LOGIC; Q : out  STD_LOGIC); --variable declaration
end ichaya;
architecture Behavioral of ichaya is
signal Qx: STD_LOGIC := '0';
begin
process (Clk, CLR)
begin
 if (CLR='1') then
        Qx <= '0'; -- output is 0 if CLR is activated
elsif (rising_edge(Clk)) then -- if CLR is not activated, look at the rising edge of clock
        if (T='1') then – If T is high then toggle the output
                Qx <= not Qx;
        end if;
end if;
end process;
Q <= Qx;
end Behavioral;
```

File 2: Code for Parallel Counter using T flipflops (Structural method)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity parallel_c is
   Port ( Qa,Qb,Qc,Qd : out  STD_LOGIC; --variable declaration
        Ck, C : in  STD_LOGIC);
end parallel_c;
```

```
architecture Behavioral of parallel_c is
component ichaya is
port(T,Clk,CLR: in STD_LOGIC; Q: out STD_LOGIC); --using the previous written code as a component
end component;
signal Qax,Qbx,Qcx,Qdx: STD_LOGIC; --declaring signals to use as inouts
signal Ta: STD_LOGIC;

begin
Ta <= '1';
G1: ichaya port map(T=>Ta,Clk=> Ck, CLR=> C, Q=> Qax ); --instantiation
G2: ichaya port map(T=>Qax,Clk=> Ck, CLR=> C, Q=> Qbx );
G3: ichaya port map(T=>(Qax and Qbx),Clk=> Ck, CLR=> C, Q=> Qcx );
G4: ichaya port map(T=>(Qcx and Qax and Qbx),Clk=> Ck, CLR=> C, Q=> Qdx );
Qa <= Qax; Qb <= Qbx; Qc <= Qcx; Qd <= Qdx;
end Behavioral;
```

File 3: Code for 1-bit Comparator

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Comp_1bit is
   Port ( A, B : in  STD_LOGIC;
          AgB : out  STD_LOGIC; -- A greater than B output
          AeB : out  STD_LOGIC; -- A equal to B output
          AlB : out  STD_LOGIC); -- A less than B output
end Comp_1bit;
architecture Behavioral of Comp_1bit is

begin
AgB <= (A and (not B));
AlB <= ((not A) and B);
AeB <= A xnor B;
end Behavioral;
```

File 4: Code for comparing two 4-bit numbers

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Comp_4bit is
   Port ( A0,A1,A2,A3 : in  STD_LOGIC;
          B0,B1,B2,B3 : in  STD_LOGIC;
          G, L, E : out  STD_LOGIC);
end Comp_4bit;

architecture Behavioral of Comp_4bit is
component Comp_1bit is
port(A, B : in  STD_LOGIC; AgB, AeB, AlB: out  STD_LOGIC); -- using the previous code as a component
end component;
signal E1,E2,E3,E4: STD_LOGIC; -- declaring 4 inout signals
begin
G1: Comp_1bit port map(A=>A3, B=>B3, AeB=>E1);
G2: Comp_1bit port map(A=>A2, B=>B2, AeB=>E2);
G3: Comp_1bit port map(A=>A1, B=>B1, AeB=>E3);
G4: Comp_1bit port map(A=>A0, B=>B0, AeB=>E4);
E <= E1 and E2 and E3 and E4; -- if all four bits are equal then E is set high.
end Behavioral;
```

File 5: The main code of counter whose modulus is controlled by a comparator

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity SyncCount_ModbyComp is
   Port ( A3,A2,A1,A0 : in  STD_LOGIC;
          Q3,Q2,Q1,Q0 : out  STD_LOGIC;
          CLK : in  STD_LOGIC);
end SyncCount_ModbyComp;
```

```
architecture Behavioral of SyncCount_ModbyComp is

component Comp_4bit is

port(A3,A2,A1,A0, B3,B2,B1,B0 : in  STD_LOGIC; G, L, E : out  STD_LOGIC); --using 4bit comparator code as a component

end component;


component parallel_c is

port(Qd,Qc,Qb,Qa : out  STD_LOGIC; Ck, C : in  STD_LOGIC);  --using parallel counter code as a component

end component;

signal temp,Qd3,Qc2,Qb1,Qa0: STD_LOGIC; -- declaring few signals which are inouts

begin

G1: parallel_c port map (Qd=>Qd3, Qc=>Qc2, Qb=>Qb1, Qa=>Qa0, Ck=>CLK, C=> temp); --instatiation

G2: Comp_4bit port map (A3=>A3,A2=>A2,A1=>A1,A0=>A0,B3=>Qd3,B2=>Qc2,B1=>Qb1,B0=>Qa0,E=>temp); --Clear of parallel counter is set by “A equal B” output of 4-bit comparator.

Q3<=Qd3; Q2<=Qc2; Q1<=Qb1; Q0<=Qa0;

end Behavioral;
```

Waveform: Comparator input(a3,a2,a1,a0) value is set to 1100. Counter counts till 1010 and resets at 1100.