1. Explain the following hierarchical design’s function. The purpose of this exercise is to make sure you can follow the hierarchical design so don’t panic if the code is too long!!

library ieee;
use ieee.std_logic_1164.all;

entity TCOUNT is
  port (Rst: in std_logic;
        Clk: in std_logic;
        Count: out std_logic_vector(4 downto 0));
end TCOUNT;

architecture STRUCTURE of TCOUNT is

  component tff
    port(Rst,Clk,T: in std_logic;
         Q: out std_logic);
  end component;

  component andgate
    port(A,B,C,D: in std_logic := '1';
         Y: out std_logic);
  end component;

  constant VCC: std_logic := '1';
signal T,Q: std_logic_vector(4 downto 0);
begin
  T(0) <= VCC;
  T0: tff port map (Rst=>Rst, Clk=>Clk, T=>T(0), Q=>Q(0));
  T(1) <= Q(0);
  T1: tff port map (Rst=>Rst, Clk=>Clk, T=>T(1), Q=>Q(1));
  A1: andgate port map(A=>Q(0), B=>Q(1), C => '1', D => '1', Y=>T(2));
  T2: tff port map (Rst=>Rst, Clk=>Clk, T=>T(2), Q=>Q(2));
  A2: andgate port map(A=>Q(0), B=>Q(1), C=>Q(2), D => '1', Y=>T(3));
  T3: tff port map (Rst=>Rst, Clk=>Clk, T=>T(3), Q=>Q(3));
  A3: andgate port map(A=>Q(0), B=>Q(1), C=>Q(2), D=>Q(3), Y=>T(4));
  T4: tff port map (Rst=>Rst, Clk=>Clk, T=>T(4), Q=>Q(4));

  Count <= Q;
end STRUCTURE;

entity andgate is
  port(A,B,C,D: in std_logic := '1';
       Y: out std_logic);
end andgate;

architecture gate of andgate is
begin
  Y <= A and B and C and D;
end gate;

entity tff is
  port(Rst,Clk,T: in std_logic;
       Q: out std_logic);
end tff;

architecture behavior of tff is
begin
  process(Rst,Clk)
  variable Qtmp: std_logic;
  begin
    if (Rst = '1') then
      Qtmp := '0';
    elsif rising_edge(Clk) then
      if T = '1' then
        Qtmp := not Qtmp;
    end if;
  end process;
end;
end if;
end if;
Q <= Qtmp;
end process;
end behavior;

2. Write a process that functions as a clock doubler. The sensitivity list should trigger on a signal named ‘clk’. You can assume that the input signal is of type ‘std_logic’ and that it will only have values ‘1’ and ‘0’. You do not know the frequency of ‘clk’ ahead of time.

3. Given an std_logic_vector, write a VHDL code fragment that will return the number of bits in the vector.

4. What is an advantage of placing a constant assignment within a package body?

5. What is the purpose of VHDL configurations?

6. What is the function of the following piece of code?

```vhdl
--      D_IN:  in STD_LOGIC;
--      RESET:  in STD_LOGIC;
--      CLK: in STD_LOGIC;
--      Q_OUT:  out STD_LOGIC;

signal Q1, Q2, Q3 : std_logic;

process(CLK, RESET)
begin
    if (RESET = '1') then
        Q1 <= '0';
        Q2 <= '0';
        Q3 <= '0';
    elsif (CLK'event and CLK = '1') then
        Q1 <= D_IN;
        Q2 <= Q1;
        Q3 <= Q2;
    end if;
end process;
```
Q_OUT <= Q1 and Q2 and (not Q3);

7. What is the logic element that best describes the following piece of code?

```vhdl
--     GATE:  in STD_LOGIC;
--     DIN:   in STD_LOGIC;
--     DOUT:  out STD_LOGIC;

process (GATE, DIN)
begin
  if GATE='1' then --GATE active High
    DOUT <= DIN;
  end if;
end process;
```

8. What is the logic element that best describes the following piece of code?

```vhdl
--     ENABLE:  in STD_LOGIC;
--     DIN:   in STD_LOGIC;
--     DOUT:  out STD_LOGIC;

process (ENABLE, DIN)
begin
  if (ENABLE='1') then
    DOUT <= DIN;
  else
    DOUT <= 'Z';
  end if;
end process;
```

9. Draw the gate level diagram of a 2 bit comparator for the following piece of code:

```
EQ <= '1' when (a = b) else '0';
```

10. Explain five main advantages of VHDL over schematic capture method to describe the logic circuits.
11. What logic building block does the following VHDL code describe?

A <= B when (X = '1') else C;

12. You are asked to design a state machine with 28 states. The synthesis tools that you use has the capability of constraining the state machine synthesis. How many flip flops would you need in each one of the following cases? Why?

   a) Synthesis tools applies One hot encoding
   b) Synthesis tools applies Binary encoding
   c) Synthesis tools applies Johnson encoding
   d) Synthesis tools applies Gray encoding

13. What would be the value of variable “count” at t = t1 for the following clock pulse?

   process (clk)
   variable count : integer := 0;
   begin
       count := count + 1;
   end process;

14. Draw Y and Z waveforms for the following piece of code. The waveform for X is given. Assume each segment is 1 ns.

Y <= transport X after 3 ns;
Z <= X after 3 ns;
15. Write a VHDL process that will generate a fixed number of clock cycles based upon a constant called 'clk_num'. The clock is a 50% duty cycle clock whose period is a constant called 'clk_per'. The initial value of the clock signal should '0' and the clock signal name is 'clk'.

16. Write a hierarchical VHDL code for a

17. Write a VHDL code for a mod-12 counter (counter that counts from 0 to 11 using state machine technique. Here is the entity declaration for your reference:

   ```vhdl
   entity mod12_cnt is
       port (clk: in std_logic;
             Q: out std_logic_vector);
   end entity;
   
   Hint: Use TYPE to define 12 states and write a CASE to perform the transitions.
   
18. Write a VHDL code for a quad 8x1 multiplexer.

19. Can you use wait statement in a process that has a sensitivity list? Why?
20. Wait statement is used to temporarily suspend a process. Which one of the wait statement combinations you would use for each one of the following cases:

a) a specified time has passed  
b) a specified condition is met  
c) an event occurs which affects one or more signals

21. What is the difference between a signal and variable assignment?

22. What is the function of the following piece of code?

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity unknown is
  port(D: in std_logic_vector(0 to 9);
       OUTPUT: out std_logic);
  constant WIDTH: integer := 10;
end unknown;

architecture behavior of unknown is begin

process(D)
  variable tmp: Boolean;
begin
  tmp := false;
  for i in 0 to D'length - 1 loop
    if D(i) = '1' then
      tmp := not tmp;
    end if;
  end loop;
  if tmp then
    OUTPUT <= '1';
  else
    OUTPUT <= '0';
  end if;
end process;
```
23. What is the function of the following function?

```vhdl
function val (arg1, arg2: integer) return integer is
    variable result: integer;
begin
    if arg1 > arg2 then
        result := arg1;
    else
        result := arg2;
    end if;
    return result;
end val;
```

24. If you have multiple architectures for an entity, how would the compiler know which one to consider? Is there any VHDL construct that binds the architecture to the entity?