SISTEMAS DIGITALES AVANZADOS

Clave del curso: TE-2030

Nombre del Profesor: Juan M. Hinojosa Olivares

Nombre del alumno: Edison Altamirano

Matricula: A00825234

Fecha de elaboración: 20/03/2020

Tecnológico de Monterrey

2.2 **(a)** Which of the following are legal VHDL identifiers? `123A`, `A_123`, `_A123`, `A123_`, `c1__c2`, `and`, `and1`

(b) Which of the following identifiers are equivalent? `aBC`, `ABC`, `Abc`, `abc`

A)

Legal	Ilegal	
A_123	**123A**	Empieza con un numero
and1	**_A123**	Empieza con guion bajo
	A123_	Termina en guion bajo
	c1_c2	Guion bajo adyacente
	and	Palabra reservada

B) Todos son equivalente puesto que VHDL no es case sensitive.

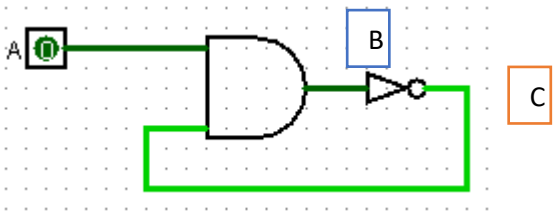
2.3 Given the concurrent VHDL statements:

```
B <= A and C after 3ns;
C <= not B after 2ns;
```

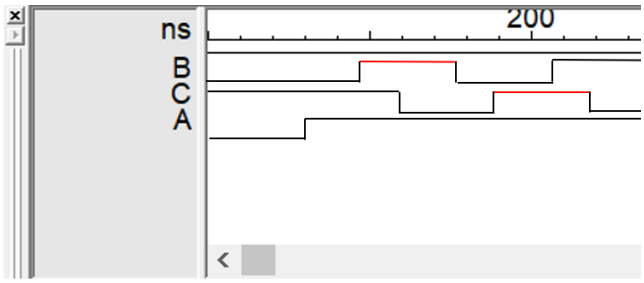
(a) Draw the circuit the statements represent.

(b) Draw a timing diagram if initially $A = B =$ '0' and $C =$ '1', and A changes to '1' at time 5 ns.

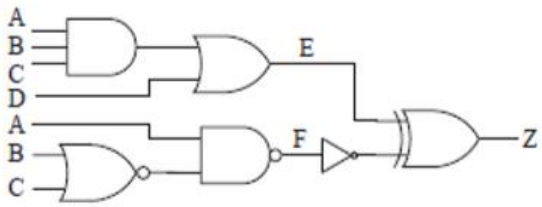
a)



b)

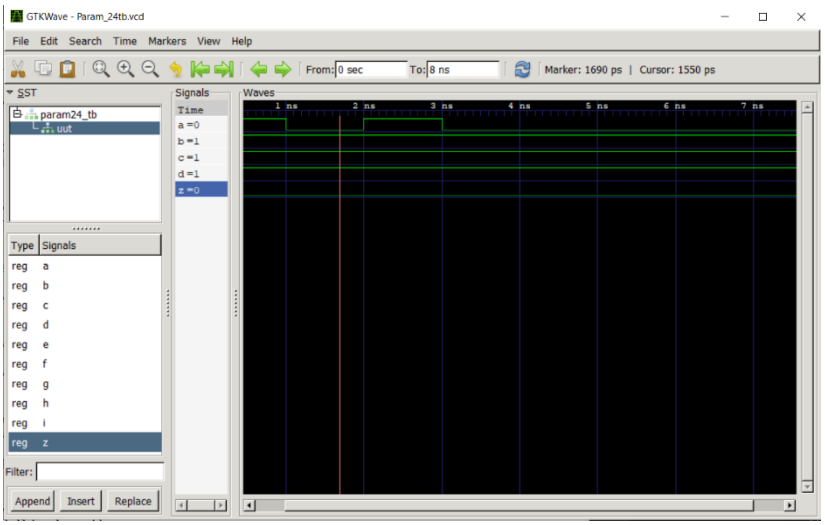


2.4 Write a VHDL description of the following combinational circuit using concurrent statements. Each gate has a 5-ns delay, excluding the inverter, which has a 2-ns delay.



Tareas Sistemas Digitales > Tarea 3 > Tarea3_24.vhdl

```vhdl
entity Param24 is
    port(A, B, C, D : in bit;
        Z: out bit);
end Param24;

architecture Circuit of Param24 is
    signal E, F, G, H, I : bit;
    begin
        G <= (A and B and C) after 5 ns;
        E <= G or D after 5 ns;
        H <= B nor C after 5 ns;
        F <= H nand A after 5 ns;
        I <= not F after 2 ns;
        Z <= I xor E after 5 ns;
end Circuit;
```



Testbench para simulación

Tareas Sistemas Digitales > Tarea 3 > Tarea3_24tb.vhdl

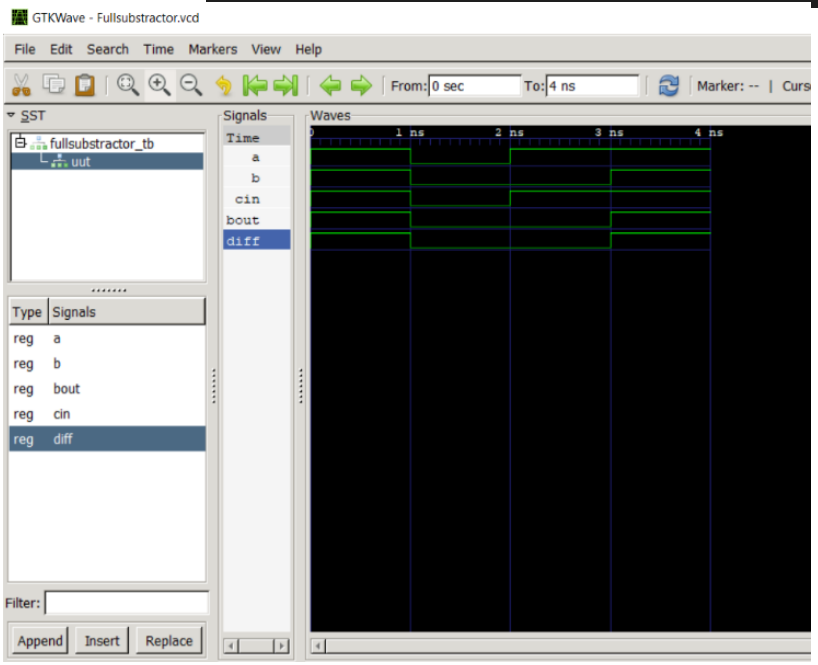
```vhdl
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;

entity Param24_tb is
end;

architecture bench of Param24_tb is

  component Param24
      port(A, B, C, D : in bit;
          Z: out bit);
  end component;

  signal A, B, C, D: bit;
  signal Z: bit;

begin

  uut: Param24 port map ( A => A,
                          B => B,
                          C => C,
                          D => D,
                          Z => Z );

  stimulus: process
  begin

    -- Put initialisation code here
    A <= '1';
    B <= '1';
    C <= '1';
    D <= '1';
    wait for 1 ns;

    A <= '0';
    B <= '1';
    C <= '1';
    D <= '1';
    wait for 1 ns;
    A <= '1';
    B <= '1';
    C <= '1';
    D <= '1';
    wait for 1 ns;
    A <= '0';
    B <= '1';
    C <= '1';
    D <= '1';
    wait for 1 ns;

    -- Put test bench stimulus code here

    wait;
  end process;


end;
```

2.5 **(a)** Write VHDL code for a full subtracter using logic equations.
(b) Write VHDL code for a 4-bit subtracter using the module defined in (a) as a component.

a)

Tarea3_25a.vhdl

```vhdl
entity fullsubstractor is
    port(A, B, CIN: in bit;
    DIFF, BOUT: out bit);
    end fullsubstractor;
architecture one_substractor of fullsubsctractor is
    begin
    DIFF <=  A xor B xor CIN;
    BOUT <= (not A and B) or (B and CIN) or (not A and CIN);
end one_substractor;
```



Testbench para simulación

Tareas Sistemas Digitales > Tarea 3 > Tarea3_25a_tb.vhdl

```vhdl
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;

entity fullsubstractor_tb is
end;

architecture bench of fullsubstractor_tb is

  component fullsubstractor
      port(A, B, CIN: in bit;
      DIFF, BOUT: out bit);
      end component;

  signal A, B, CIN: bit;
  signal DIFF, BOUT: bit;

begin

  uut: fullsubstractor port map ( A    => A,
                                  B    => B,
                                  CIN  => CIN,
                                  DIFF => DIFF,
                                  BOUT => BOUT );

  stimulus: process
  begin

    -- Put initialisation code here
    A <= '1';
    B <= '1';
    CIN <= '1';
    wait for 1 ns;
    A <= '0';
    B <= '0';
    CIN <= '0';
    wait for 1 ns;
    A <= '1';
    B <= '0';
    CIN <= '1';
    wait for 1 ns;
    A <= '1';
    B <= '1';
    CIN <= '1';
    wait for 1 ns;
    -- Put test bench stimulus code here

    wait;
  end process;


end;
```

b)

Tarea3_25b.vhdl

```vhdl
entity full_four_substractor is
    port(A, B: in bit_vector(3 downto 0);
    CIN: in bit;
    D: out bit_vector(3 downto 0);
    BOUT: out bit);
    end full_four_substractor;
architecture four_substractor of full_four_substractor is
    signal BO: bit_vector(3 downto 0) := "0000";
    component fullsubstractor
        port(A, B, CIN: in bit;
            DIFF, BOUT: out bit);
    end component;
begin
    FS0: fullsubstractor port map(A(0), B(0), CIN, D(0), BO(1));
    FS1: fullsubstractor port map(A(1), B(1), BO(1), D(1), BO(2));
    FS2: fullsubstractor port map(A(2), B(2), BO(2), D(2), BO(3));
    FS3: fullsubstractor port map(A(3), B(3), BO(3), D(3), BOUT);
end four_substractor;
```

Testbench para simulación

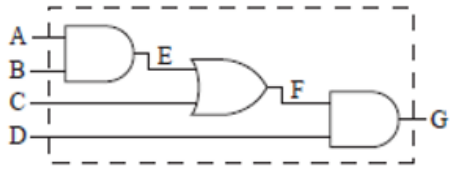
Tareas Sistemas Digitales > Tarea 3 > Tarea3_25b_tb.vhdl

```vhdl
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;

entity full_four_substractor_tb is
end;

architecture bench of full_four_substractor_tb is

  component full_four_substractor
      port(A, B: in bit_vector(3 downto 0);
      CIN: in bit;
      D: out bit_vector(3 downto 0);
      BOUT: out bit);
      end component;

  signal A, B: bit_vector(3 downto 0);
  signal CIN: bit;
  signal D: bit_vector(3 downto 0);
  signal BOUT: bit;

begin

  uut: full_four_substractor port map ( A    => A,
                                        B    => B,
                                        CIN  => CIN,
                                        D    => D,
                                        BOUT => BOUT );

  stimulus: process
  begin

    -- Put initialisation code here
    A <= "1111";
    B <= "1111";
    CIN <= '1';
    wait for 1 ns;
    A <= "0000";
    B <= "0000";
    CIN <= '0';
    wait for 1 ns;
    A <= "1101";
    B <= "1011";
    CIN <= '1';
    wait for 1 ns;
    A <= "1111";
    B <= "1001";
    CIN <= '1';
    wait for 1 ns;

    -- Put test bench stimulus code here

    wait;
  end process;


end;
```

a)

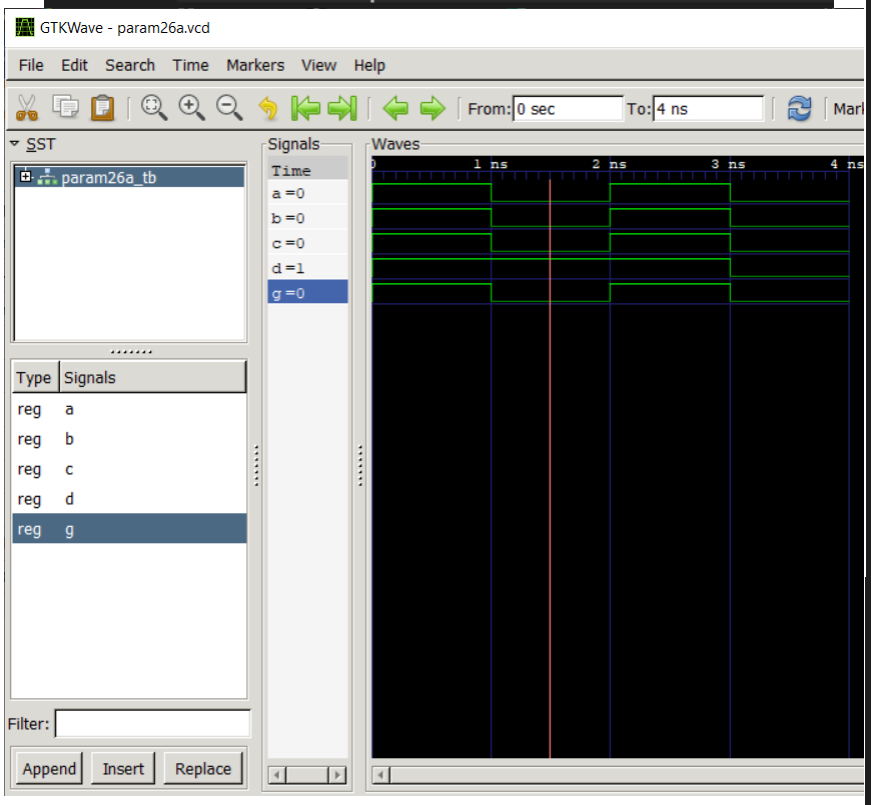
2.6 Write VHDL code for the following circuit. Assume that the gate delays are negligible.

(a) Use concurrent statements.
(b) Use a process with sequential statements.



Tareas Sistemas Digitales > Tarea 3 > Tarea3_26a.vhdl

```vhdl
entity param26a is
port(A, B, C, D: in bit;
    G: out bit);
end param26a;
architecture circuit of param26a is
    signal E, F: bit;
    begin
        E <= A and B;
        F <= E or C;
        G <= F and D;
end circuit;
```



Testbench para simulación

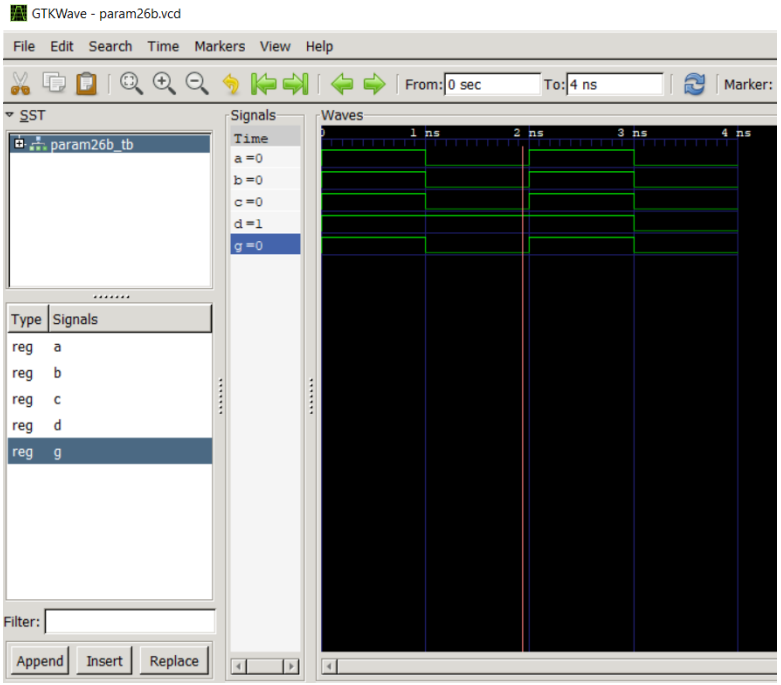
Tareas Sistemas Digitales > Tarea 3 > Tarea3_26a_tb.vhdl

```vhdl
entity param26a_tb is
    end;

    architecture bench of param26a_tb is

      component param26a
      port(A, B, C, D: in bit;
          G: out bit);
      end component;

      signal A, B, C, D: bit;
      signal G: bit;

    begin

      uut: param26a port map ( A => A,
                               B => B,
                               C => C,
                               D => D,
                               G => G );

      stimulus: process
      begin

        -- Put initialisation code here
        A <= '1';
        B <= '1';
        C <= '1';
        D <= '1';
        wait for 1 ns;
        A <= '0';
        B <= '0';
        C <= '0';
        D <= '1';
        wait for 1 ns;
        A <= '1';
        B <= '1';
        C <= '1';
        D <= '1';
        wait for 1 ns;
        A <= '0';
        B <= '0';
        C <= '0';
        D <= '0';
        wait for 1 ns;
        -- Put test bench stimulus code here

        wait;
      end process;


    end;
```

b)

Tareas Sistemas Digitales > Tarea 3 > Tarea3_26b.vhdl

```vhdl
entity param26b is
    port(A, B, C, D: in bit;
    G: out bit);
end param26b;
architecture circuit of param26b is
    signal E, F: bit;
begin
    process(A, B, C, E, F)
    begin
        E <= A and B;
        F <= C or E;
        G <= F and D;
    end process;
end circuit;
```



Testbench para simulación

```vhdl
library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;

entity param26b_tb is
end;

architecture bench of param26b_tb is

  component param26b
      port(A, B, C, D: in bit;
      G: out bit);
  end component;

  signal A, B, C, D: bit;
  signal G: bit;

begin

  uut: param26b port map ( A => A,
                           B => B,
                           C => C,
                           D => D,
                           G => G );

  stimulus: process
  begin

    -- Put initialisation code here
    A <= '1';
    B <= '1';
    C <= '1';
    D <= '1';
    wait for 1 ns;
    A <= '0';
    B <= '0';
    C <= '0';
    D <= '1';
    wait for 1 ns;
    A <= '1';
    B <= '1';
    C <= '1';
    D <= '1';
    wait for 1 ns;
    A <= '0';
    B <= '0';
    C <= '0';
    D <= '0';
    wait for 1 ns;

    -- Put test bench stimulus code here

    wait;
  end process;


end;
```

2.7 In the following VHDL code, A, B, C, and D are integers that are 0 at time 10 ns. If D changes to 1 at 20 ns, specify the times at which A, B, and C will change and the values they will take.

```
process(D)
begin
  A <= 1 after 5 ns;
  B <= A + 1;           -- executes before A changes
  C <= B after 10 ns;   -- executes before B changes
end process;
```

A cambia a 1 en 25 ns, B cambia a 1 en 20, y C no cambia.

2.8 **(a)** What device does the following VHDL code represent?

```
process(CLK, Clr, Set)
begin
  if Clr = '1' then Q <= '0';
  elsif Set = '1' then Q <= '1';
  elsif CLK'event and CLK <= '0' then
    Q <= D;
  end if;
end process;
```

(b) What happens if $Clr = Set =$ '1' in the device in part (a)?

A) Flipflip D con salida activa asíncrono clear y set.

B) Q = 'O', porque Clear = 1 tiene prioridad.

2.10 An M-N flip-flop responds to the falling clock edge as follows:

If $M = N =$ '0', the flip-flop changes state.
If $M =$ '0' and $N =$ '1', the flip-flop output is set to '1'.
If $M =$ '1' and $N =$ '0', the flip-flop output is set to '0'.
If $M = N =$ '1', no change of flip-flop state occurs.
The flip-flop is cleared asynchronously if $CLRn =$ '0'.

Write a complete VHDL module that implements an M-N flip-flop.

Tareas Sistemas Digitales > Tarea 3 > Tarea3_210.vhdl

```
1  entity Param is
2      port(M, N, Clk, CLR: in bit; Q: inout bit; Qn: out bit);
3  end Param;
4  architecture Circuit of Param is
5  begin
6      process(Clk, CLR)
7      begin
8          if CLR = '0' then Q <= '0';
9          elsif Clk = '0' and Clk'event then
10             if M = '0' and N = '0' then Q <= not Q;
11             elsif M = '0' and N = '1' then Q <= '1';
12             elsif M = '1' and N = '0' then Q <= '0';
13             elsif M = '1' and N = '1' then Q <= Q;
14             end if;
15         end if;
16     end process;
17     Qn <= not Q;
18 end Circuit;
```