

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	Illegal	
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

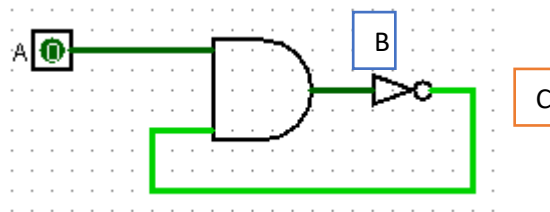
B) Todos son equivalente puesto que VHDL no es case sensitive.

2.3 Given the concurrent VHDL statements:

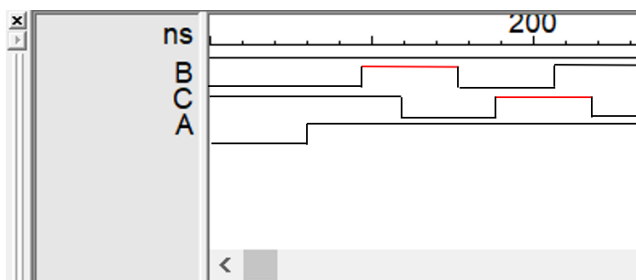
```
B <= A and C after 3ns;
C <= not B after 2ns;
```

- (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.

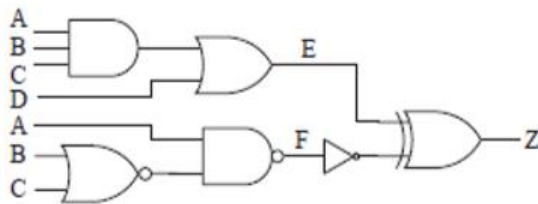
a)



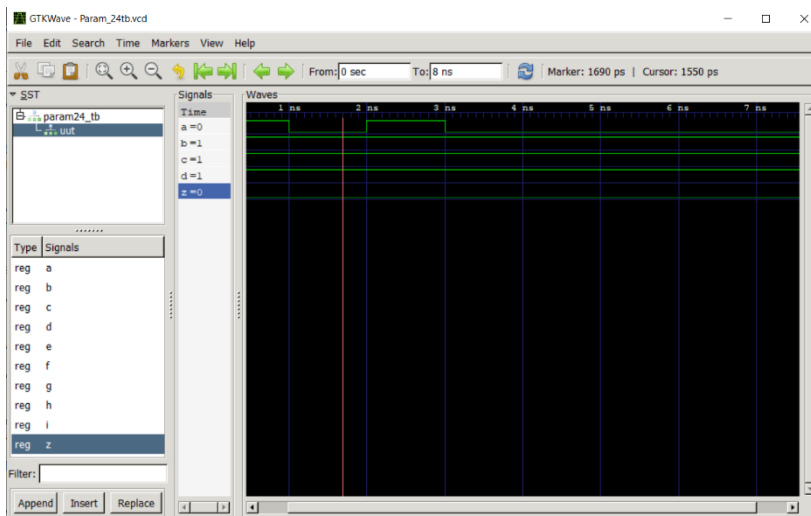
b)



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
1 entity Param24 is
2   port(A, B, C, D : in bit;
3     Z: out bit);
4 end Param24;
5
6 architecture Circuit of Param24 is
7   signal E, F, G, H, I : bit;
8   begin
9     G <= (A and B and C) after 5 ns;
10    E <= G or D after 5 ns;
11    H <= B nor C after 5 ns;
12    F <= H nand A after 5 ns;
13    I <= not F after 2 ns;
14    Z <= I xor E after 5 ns;
15 end Circuit;
```



Testbench para simulación

```
Tareas Sistemas Digitales > Tarea 3 > Tarea3_24tb.vhdl
1 library IEEE;
2 use IEEE.Std_logic_1164.all;
3 use IEEE.Numeric_Std.all;
4
5 entity Param24_tb is
6 end;
7
8 architecture bench of Param24_tb is
9
10  component Param24
11    port(A, B, C, D : in bit;
12      Z: out bit);
13  end component;
14
15  signal A, B, C, D: bit;
16  signal Z: bit;
17
18  begin
19
20    uut: Param24 port map ( A => A,
21      B => B,
22      C => C,
23      D => D,
24      Z => Z );
25
26    stimulus: process
27    begin
28
29      -- Put initialisation code here
30      A <= '1';
31      B <= '1';
32      C <= '1';
33      D <= '1';
34      wait for 1 ns;
35
36      A <= '0';
37      B <= '1';
38      C <= '1';
39      D <= '1';
40      wait for 1 ns;
41
42      A <= '1';
43      B <= '1';
44      C <= '1';
45      D <= '1';
46      wait for 1 ns;
47      A <= '0';
48      B <= '1';
49      C <= '1';
50      D <= '1';
51      wait for 1 ns;
52
53      -- Put test bench stimulus code here
54      wait;
55    end process;
56
57  end;
```

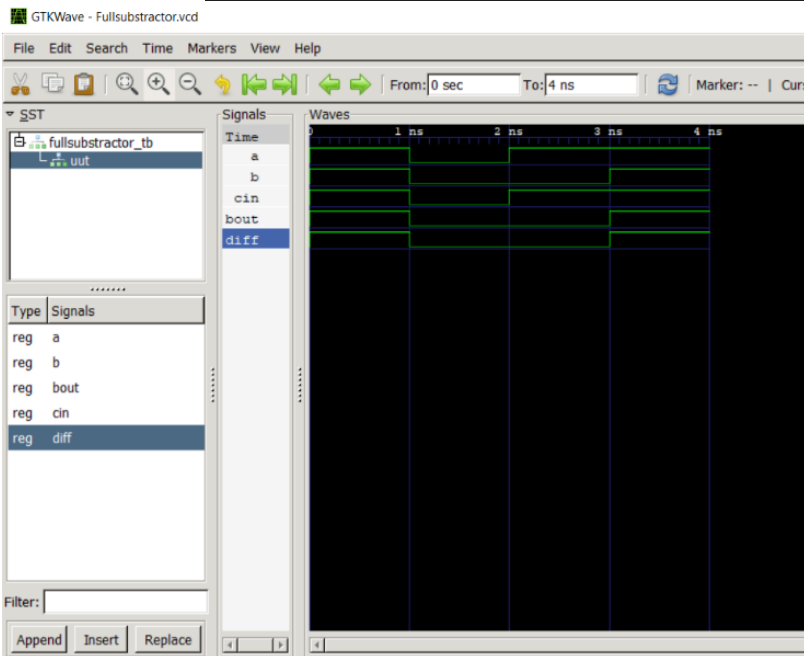
- 2.5 (a) Write VHDL code for a full subtractor using logic equations.
 (b) Write VHDL code for a 4-bit subtractor using the module defined in (a) as a component.

a)

```

Tarea3_25a.vhdl
1 entity fullsubtractor is
2     port(A, B, CIN: in bit;
3         DIFF, BOUT: out bit);
4 end fullsubtractor;
5 architecture one_subtractor of fullsubtractor is
6     begin
7         DIFF <= A xor B xor CIN;
8         BOUT <= (not A and B) or (B and CIN) or (not A and CIN);
9     end one_subtractor;
  
```

Testbench para simulación



```

Tareas Sistemas Digitales > Tarea 3 > Tarea3_25a_tb.vhdl
1 library IEEE;
2 use IEEE.Std_logic_1164.all;
3 use IEEE.Numeric_Std.all;
4
5 entity fullsubtractor_tb is
6 end;
7
8 architecture bench of fullsubtractor_tb is
9
10     component fullsubtractor
11         port(A, B, CIN: in bit;
12             DIFF, BOUT: out bit);
13     end component;
14
15     signal A, B, CIN: bit;
16     signal DIFF, BOUT: bit;
17
18 begin
19
20     uut: fullsubtractor port map ( A    => A,
21                                   B    => B,
22                                   CIN  => CIN,
23                                   DIFF => DIFF,
24                                   BOUT => BOUT );
25
26     stimulus: process
27     begin
28
29         -- Put initialisation code here
30         A <= '1';
31         B <= '1';
32         CIN <= '1';
33         wait for 1 ns;
34         A <= '0';
35         B <= '0';
36         CIN <= '0';
37         wait for 1 ns;
38         A <= '1';
39         B <= '0';
40         CIN <= '1';
41         wait for 1 ns;
42         A <= '1';
43         B <= '1';
44         CIN <= '1';
45         wait for 1 ns;
46         -- Put test bench stimulus code here
47
48         wait;
49     end process;
50
51
52 end;
  
```

b)

```
Tarea3_25b.vhdl
1  entity full_four_subtractor is
2      port(A, B: in bit_vector(3 downto 0);
3          CIN: in bit;
4          D: out bit_vector(3 downto 0);
5          BOUT: out bit);
6  end full_four_subtractor;
7  architecture four_subtractor of full_four_subtractor is
8      signal B0: bit_vector(3 downto 0) := "0000";
9      component fullsubtractor
10         port(A, B, CIN: in bit;
11             DIFF, BOUT: out bit);
12     end component;
13 begin
14     FS0: fullsubtractor port map(A(0), B(0), CIN, D(0), B0(1));
15     FS1: fullsubtractor port map(A(1), B(1), B0(1), D(1), B0(2));
16     FS2: fullsubtractor port map(A(2), B(2), B0(2), D(2), B0(3));
17     FS3: fullsubtractor port map(A(3), B(3), B0(3), D(3), BOUT);
18 end four_subtractor;
```

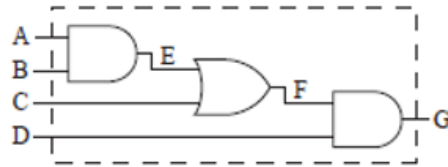
Testbench para simulación

```
Tareas Sistemas Digitales > Tarea 3 > Tarea3_25b.tb.vhdl
1  library IEEE;
2  use IEEE.Std_logic_1164.all;
3  use IEEE.Numeric_Std.all;
4
5  entity full_four_subtractor_tb is
6  end;
7
8  architecture bench of full_four_subtractor_tb is
9
10     component full_four_subtractor
11         port(A, B: in bit_vector(3 downto 0);
12             CIN: in bit;
13             D: out bit_vector(3 downto 0);
14             BOUT: out bit);
15     end component;
16
17     signal A, B: bit_vector(3 downto 0);
18     signal CIN: bit;
19     signal D: bit_vector(3 downto 0);
20     signal BOUT: bit;
21
22 begin
23
24     uut: full_four_subtractor port map ( A => A,
25                                         B => B,
26                                         CIN => CIN,
27                                         D => D,
28                                         BOUT => BOUT );
29
30     stimulus: process
31     begin
32
33         -- Put initialisation code here
34         A <= "1111";
35         B <= "1111";
36         CIN <= '1';
37         wait for 1 ns;
38         A <= "0000";
39         B <= "0000";
40         CIN <= '0';
41         wait for 1 ns;
42         A <= "1101";
43         B <= "1011";
44         CIN <= '1';
45         wait for 1 ns;
46         A <= "1111";
47         B <= "1001";
48         CIN <= '1';
49         wait for 1 ns;
50
51         -- Put test bench stimulus code here
52
53         wait;
54     end process;
55
56
57 end;
```

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

a)

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



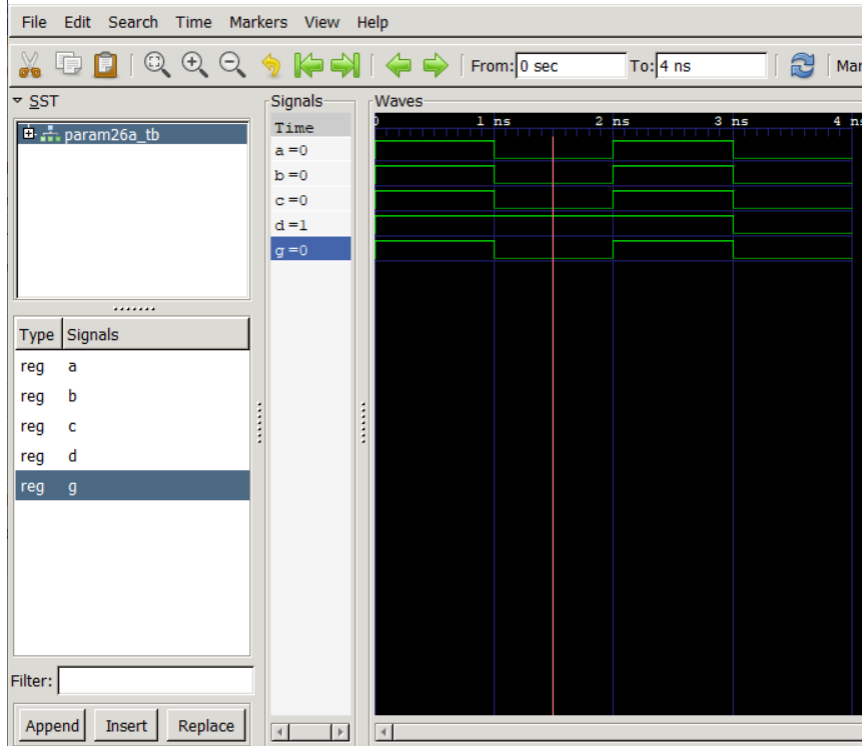
Testbench para simulación

Tareas Sistemas Digitales > Tarea 3 > Tarea3_26a.vhdl

```

1  entity param26a is
2  port(A, B, C, D: in bit;
3      G: out bit);
4  end param26a;
5  architecture circuit of param26a is
6      signal E, F: bit;
7  begin
8      E <= A and B;
9      F <= E or C;
10     G <= F and D;
11 end circuit;
```

GTKWave - param26a.vcd



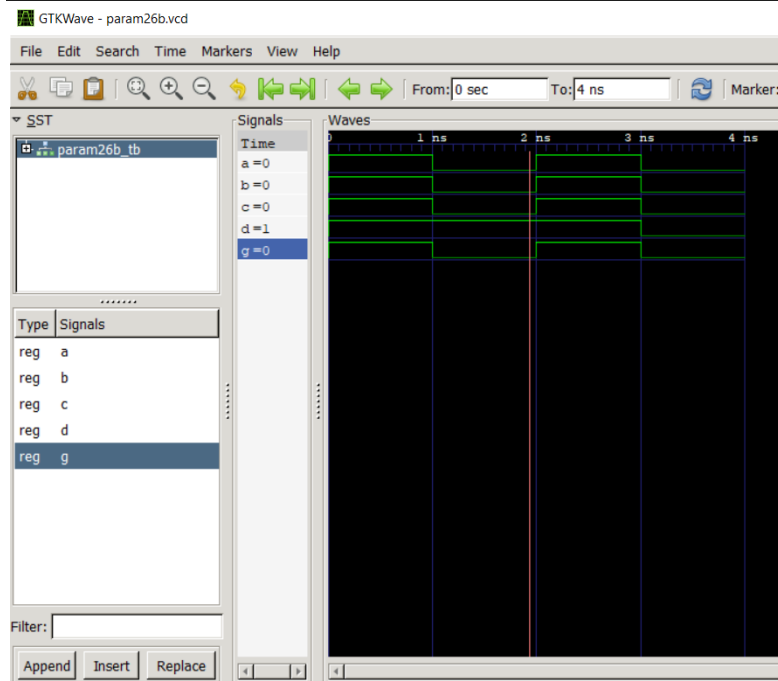
Tareas Sistemas Digitales > Tarea 3 > Tarea3_26a_tb.vhdl

```

1  entity param26a_tb is
2  end;
3
4  architecture bench of param26a_tb is
5
6      component param26a
7      port(A, B, C, D: in bit;
8          G: out bit);
9      end component;
10
11     signal A, B, C, D: bit;
12     signal G: bit;
13
14     begin
15
16     uut: param26a port map ( A => A,
17                             B => B,
18                             C => C,
19                             D => D,
20                             G => G );
21
22     stimulus: process
23     begin
24
25         -- Put initialisation code here
26         A <= '1';
27         B <= '1';
28         C <= '1';
29         D <= '1';
30         wait for 1 ns;
31         A <= '0';
32         B <= '0';
33         C <= '0';
34         D <= '1';
35         wait for 1 ns;
36
37         A <= '1';
38         B <= '1';
39         C <= '1';
40         D <= '1';
41         wait for 1 ns;
42         A <= '0';
43         B <= '0';
44         C <= '0';
45         D <= '0';
46         wait for 1 ns;
47         -- Put test bench stimulus code here
48
49         wait;
50     end process;
51
52 end;
```

b)

```
Tareas Sistemas Digitales > Tarea 3 > Tarea3_26b.vhdl
1  entity param26b is
2      port(A, B, C, D: in bit;
3          G: out bit);
4  end param26b;
5  architecture circuit of param26b is
6      signal E, F: bit;
7  begin
8      process(A, B, C, E, F)
9      begin
10         E <= A and B;
11         F <= C or E;
12         G <= F and D;
13     end process;
14 end circuit;
```



Testbench para simulación

```
2  library IEEE;
3  use IEEE.Std_logic_1164.all;
4  use IEEE.Numeric_Std.all;
5
6  entity param26b_tb is
7  end;
8
9  architecture bench of param26b_tb is
10
11      component param26b
12          port(A, B, C, D: in bit;
13              G: out bit);
14      end component;
15
16      signal A, B, C, D: bit;
17      signal G: bit;
18
19  begin
20
21      uut: param26b port map ( A => A,
22                              B => B,
23                              C => C,
24                              D => D,
25                              G => G );
26
27      stimulus: process
28      begin
29
30          -- Put initialisation code here
31          A <= '1';
32          B <= '1';
33          C <= '1';
34          D <= '1';
35          wait for 1 ns;
36          A <= '0';
37          B <= '0';
38          C <= '0';
39          D <= '1';
40          wait for 1 ns;
41          A <= '1';
42          B <= '1';
43          C <= '1';
44          D <= '1';
45          wait for 1 ns;
46          A <= '0';
47          B <= '0';
48          C <= '0';
49          D <= '0';
50          wait for 1 ns;
51
52          -- Put test bench stimulus code here
53
54          wait;
55      end process;
56
57
58  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) Flipflop D con salida activa asíncrono clear y set.

B) $Q = '0'$, 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 $CLR_n = '0'$.

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;
```