ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

Exercícios Referentes à Prova P1

prof. Dr. César Augusto M. Marcon
prof. Dr. Edson Ifarraguirre Moreno

Exercícios

1. Assinale com **V** se a sentença estiver correta, caso contrário preencha com **F**

( ) A ocorrência de overflow, em operações de soma de números sem sinal, indica que o resultado da soma não pode ser representado corretamente

( ) Não pode ocorrer overflow na soma de números inteiros de sinais opostos

( ) Carry e negativo são sinais usados para informar o resultado de operações aritméticas. Estes servem como informação útil para alterar o fluxo de execução de programas

( ) Descrição comportamental, por ser mais detalhada que descrição estrutural, permite ao projetista informar mais características de projeto, de forma a melhor seu desempenho

( ) A sentença VHDL ***"y <= x + z after 5ns"*** significa que o valor do tempo atual de ***x*** é somado com o valor que estava em ***z*** 5ns atrás para gerar o novo valor de ***y***.

Exercícios

2. Diga duas características de VHDL, não existentes nas linguagens de programação, que a capacita para descrever hardware. Descreva e exemplifique estas características

3. Considere o projeto de um circuito digital que implementa a função **f** com três variáveis de entrada (**x**, **y**, **z**) e satisfazendo as seguintes propriedades:

$$f(x,y,z) = \begin{cases} 1 & \text{se } x = y \\ 0 & \text{caso contrário} \end{cases}$$

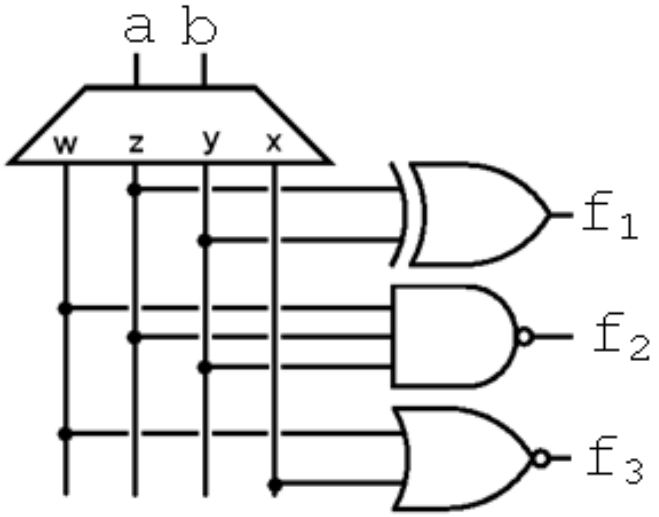
- Qual das expressões abaixo representa corretamente a função **f**?

1. $x + \bar{y}.z$
2. $\overline{x.y.z} + x.\bar{y}.z$
3. $\bar{x}.y + x.\bar{y}$
4. $x.y + \bar{y}.z + \bar{z}$
5. $\bar{x}.\bar{y} + x.y$

Exercícios

4. Dado o decodificador binário 2x4 e as portas lógicas da figura - ambos ilustrados abaixo, obtenha as funções **f1**, **f2** e **f3** em função das entradas **a** e **b**.

a	b	w	z	y	x
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



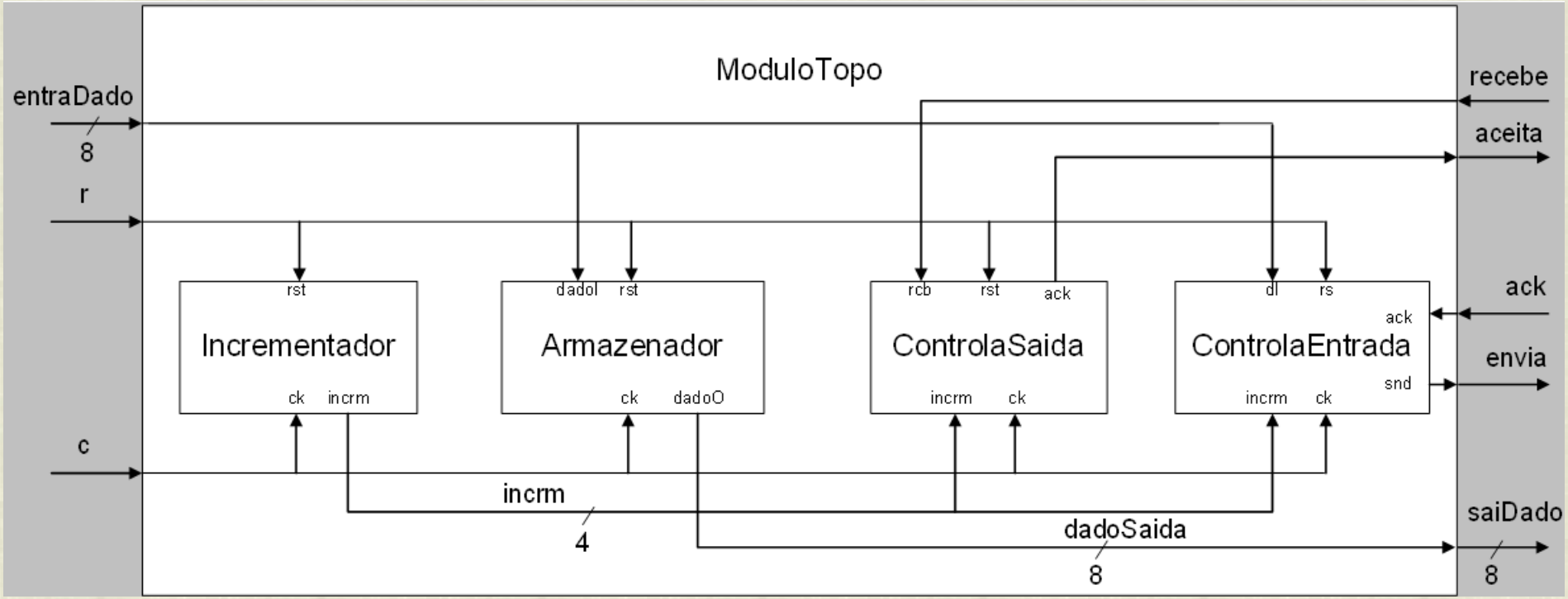
f_1
1) a . b
2) a ⊕ b
3) a + b
4) 1
5) 0

f_2
1) a . b
2) a ⊕ b
3) a + b
4) 1
5) 0

f_3
1) a . b
2) a ⊕ b
3) a + b
4) 1
5) 0

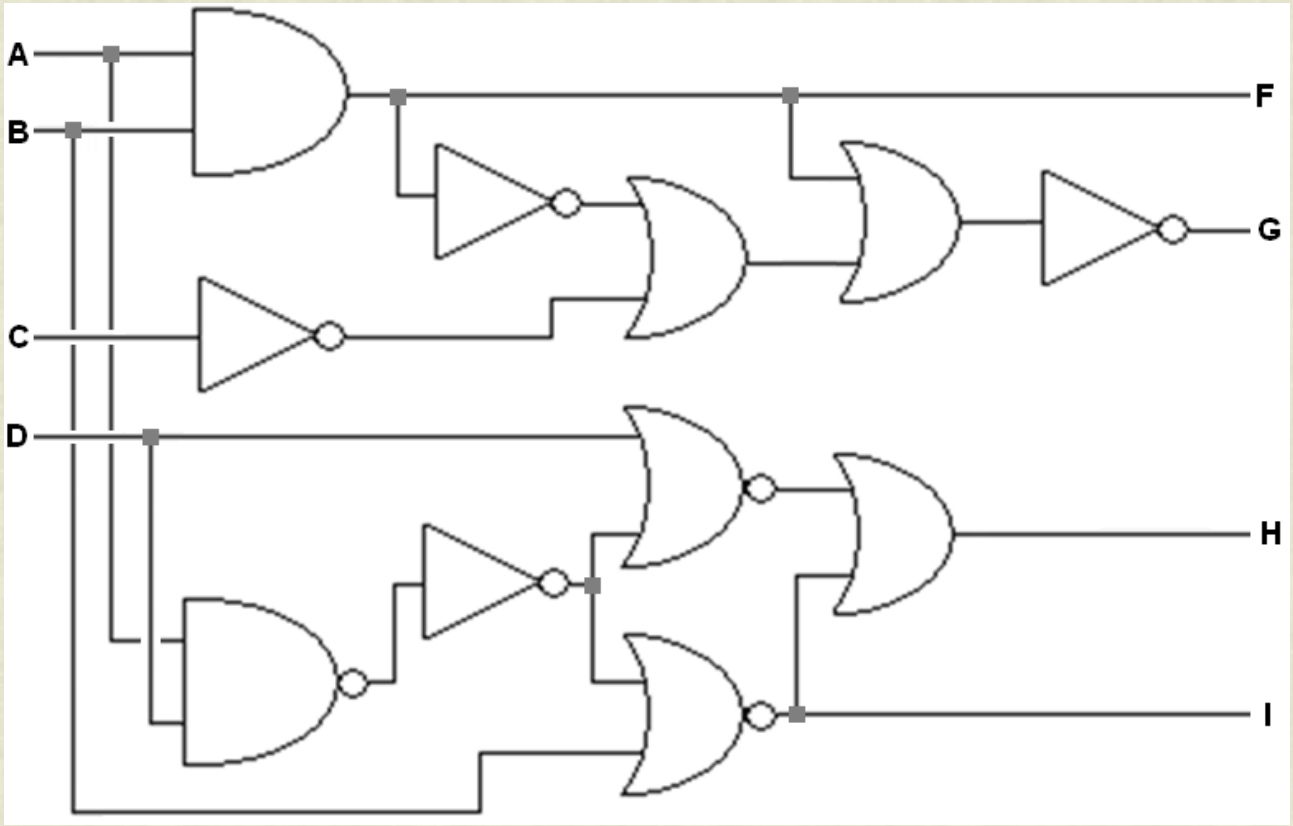
Exercícios

5. Descreve a entidade e a arquitetura do módulo ModuloTopo. Considere que os módulos: Incrementador, Armazenador, ControlaSaida e ControlaEntrada devam apenas ser mapeados (port map) na arquitetura do ModuloTopo



Exercícios

6. Determine o maior caminho do circuito abaixo. Considerando que portas NOT (inversores) atrasam 1ns, portas NAND (não e lógico) e NOR (não ou lógico) atrasam 3ns, portas AND (e lógico) e portas OR (ou lógico) atrasam 5ns.



Exercícios

7. Jogo dos 8 erros em VHDL. O código VHDL abaixo contém exatamente 8 falhas, que você deve identificar, deixando claro qual o erro. Os erros são de diversas naturezas, sintaxe, contexto, semântica.

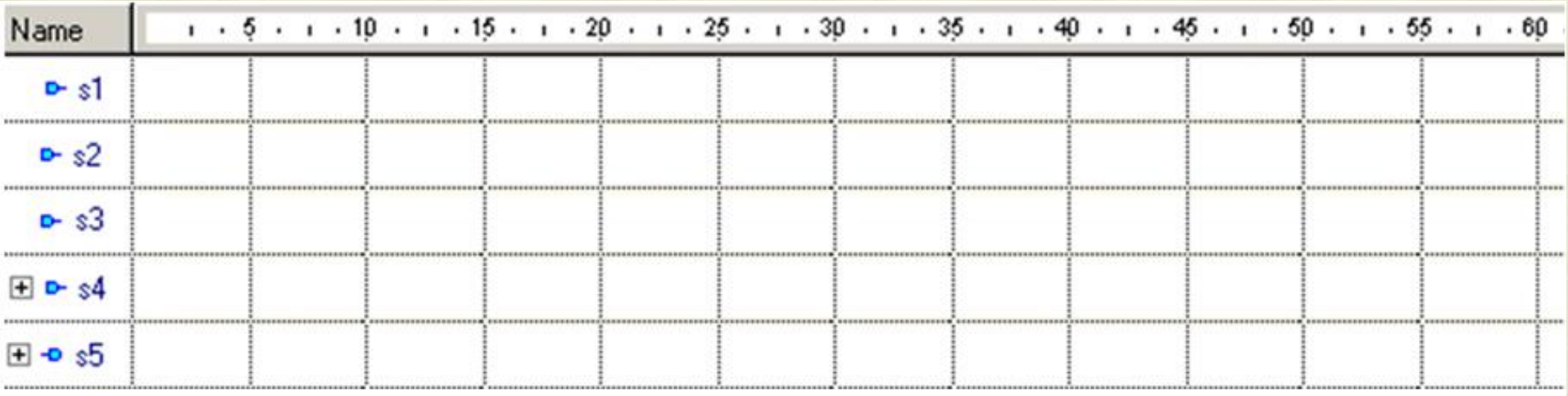
```
1. library IEEE;
2. entity p1 is
3.       port(  A1; B2; C3: in STD_LOGIC;
4.              D1; E2; F3; G4: out STD_LOGIC);
5. end p1;
6. architecture Prova1 of p1 is
7.      signal iD1, iE2, iF3, iG4: out std_logic;
8.      signal S: std_logic_vector(3 downto 0);
9. begin
10.     D1 <= iD1; E2 <= iE2; F3 <= iF3; G4 <= iG4;
11.     A1 <=  C3 and (B2 or iD1);
12.     iD1 <= A1 and B2; iE2 <= A1 xor C3; iF3 <= A1 or C3;
13.     iG4 <= iF3 nand (iE2 xor (B2 and C3));
14.     S(1 downto 0) <= iE2 & iF3 & iD1;
15.     with (D1 & E2 & F3) select
16.             S <=    "1011" when "111",
17.                     "0101" when "110",
18.                     "0101" when "011"| "100" | "000",
19.                     "1100" when "010",
20.                     "1011" when others;
21.end p1;
```

Exercícios

8. Faça uma figura com os primeiros 100ns das formas de onda geradas pelo testbench descrito a seguir. Considere os sinais **s1, s2, s3, s4** e **s5**:

```
entity E_P is
  port
  (
    s1, s2: in STD_LOGIC;
    s3: out STD_LOGIC;
    s4: in STD_LOGIC_VECTOR(3 downto 0);
    s5: out STD_LOGIC_VECTOR(3 downto 0)
  );
end E_P;
```

```
architecture P of E_P is
begin
  s3 <= not (s1 and s2) after 10ns;
  s5 <= not s4 after 5ns;
end P;
```



Exercícios

8. (continuação)

```
entity e_p_tb is
end e_p_tb;

architecture TB of e_p_tb is
  signal s1, s2, s3: std_logic;
  signal s4: std_logic_vector(3 downto 0) := (others=>'0');
  signal s5: std_logic_vector(3 downto 0);
begin
  U: entity work.E_P port map(s1=>s1, s2=>s2, s3=>s3, s4=>s4, s5=>s5);
  process
  begin
    s1 <= '0', '1' after 20ns;
    wait for 50ns;
  end process;
  process
  begin
    s2 <= '1', '0' after 15ns, '1' after 25ns;
    wait for 30ns;
  end process;
  process
  begin
    s4 <= s4 + 1 after 25ns;
    wait for 60ns;
  end process;
end TB;
```

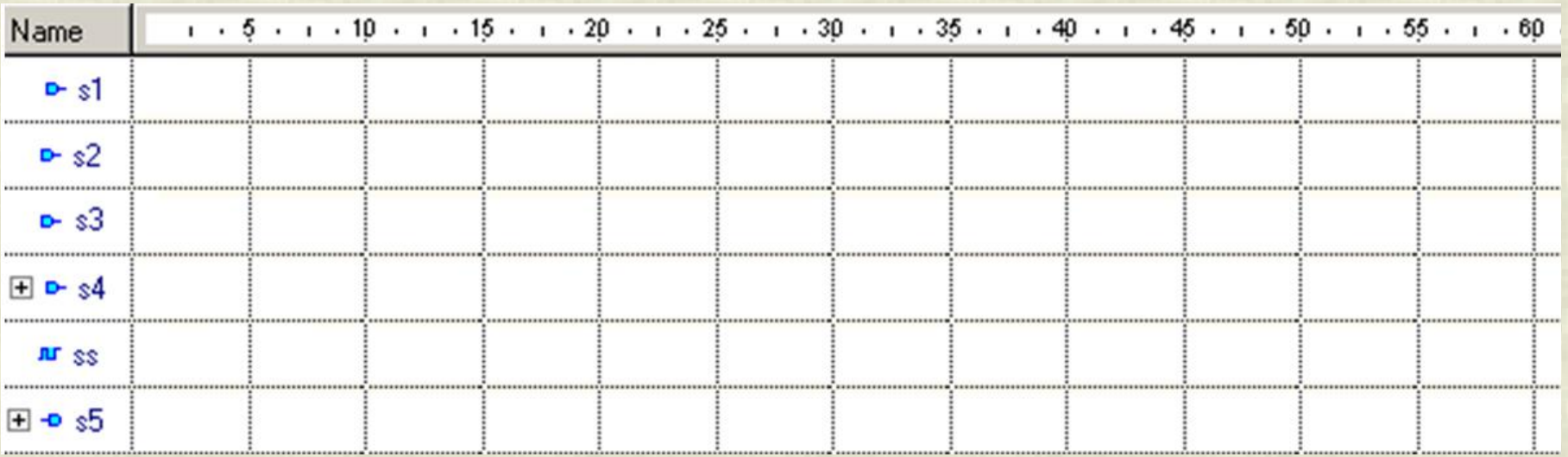
Exercícios

9. Faça as formas de onda geradas pela execução dos **60ns** iniciais do testbench descrito a seguir:

```
architecture TB of TbAtraso is
    signal s1, s2, s3, ss: std_logic;
    signal s4, s5: std_logic_vector(3 downto 0) := (others=>'0');
begin
    s3 <= not ss;
    ss <= s1 xor s2;
    s5 <= s4 + ss;
    process
    begin
        s1 <= '1', '0' after 10ns;
        wait for 50ns;
    end process;
    process
    begin
        s2 <= '1', '0' after 10ns, '1' after 15ns, '0' after 25ns;
        wait for 60ns;
    end process;
    process
    begin
        s4 <= s4 + "0101" after 30ns;
        wait for 45ns;
    end process;
end TB;
```

Exercícios

9. (continuação)



Exercícios

1. Assinale com **V** se a sentença estiver correta, caso contrário preencha com **F**

(**F**) A ocorrência de overflow, em operações de soma de números sem sinal, indica que o resultado da soma não pode ser representado corretamente

(**V**) Não pode ocorrer overflow na soma de números inteiros de sinais opostos

(**V**) Carry e negativo são sinais usados para informar o resultado de operações aritméticas. Estes servem como informação útil para alterar o fluxo de execução de programas

(**F**) Descrição comportamental, por ser mais detalhada que descrição estrutural, permite ao projetista informar mais características de projeto, de forma a melhor seu desempenho

(**F**) A sentença VHDL ***"y <= x + z after 5ns"*** significa que o valor do tempo atual de ***x*** é somado com o valor que estava em ***z*** 5ns atrás para gerar o novo valor de ***y***.

Exercícios

3. Considere o projeto de um circuito digital que implementa a função **f** com três variáveis de entrada (**x**, **y**, **z**) e satisfazendo as seguintes propriedades:

$$f(x,y,z) = \begin{cases} 1 & \text{se } x = y \\ 0 & \text{caso contrário} \end{cases}$$

- Qual das expressões abaixo representa corretamente a função **f**?

x	y	Z	f(x,y,z)	
0	0	0	1	$\bar{x} . \bar{y} . \bar{z}$
0	0	1	1	$\bar{x} . \bar{y} . z$
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$x . y . \bar{z}$
1	1	1	1	$x . y . z$

$\bar{x}.\bar{y}.\bar{z} + \bar{x}.\bar{y}.z + x.y.\bar{z} + x.y.z$

$\bar{x}.\bar{y}. (z+\bar{z}) + x.y. (z+\bar{z})$

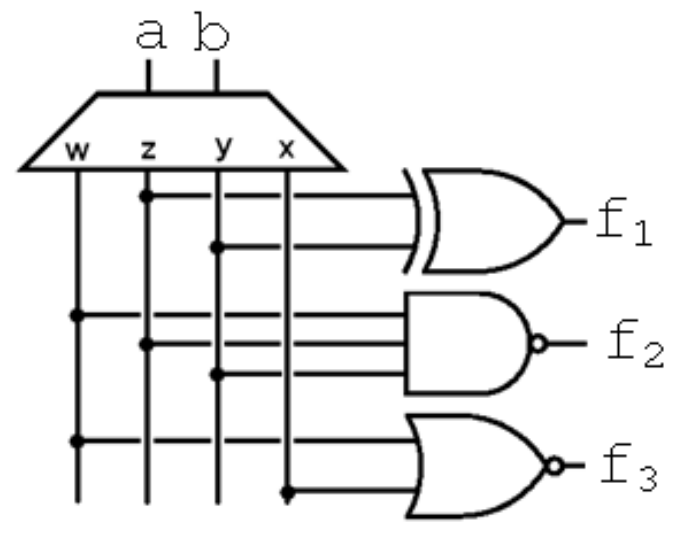
$\bar{x}.\bar{y} + x.y$

1. $x + \bar{y}.z$
2. $\overline{x.y.z} + x.\bar{y}.z$
3. $\bar{x}.y + x.\bar{y}$
4. $x.y + \bar{y}.z + \bar{z}$
5. $\bar{x}.\bar{y} + x.y$

Exercícios

4. Dado o decodificador binário 2x4 e as portas lógicas da figura - ambos ilustrados abaixo, obtenha as funções **f1**, **f2** e **f3** em função das entradas **a** e **b**.

a	b	w	z	y	x
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



f1 = (z ^ y)	f2=$\overline{(w.z.y)}$	f3=$\overline{(w+x)}$
0	1	0
1	1	1
1	1	1
0	1	0

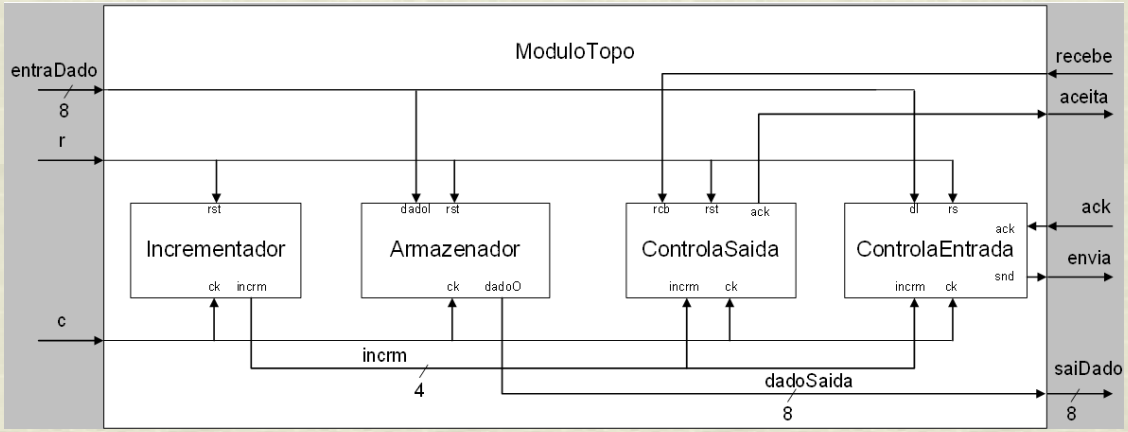
f_1	
1)	a . b
2)	a ⊕ b
3)	a + b
4)	1
5)	0

f_2	
1)	a . b
2)	a ⊕ b
3)	a + b
4)	1
5)	0

f_3	
1)	a . b
2)	a ⊕ b
3)	a + b
4)	1
5)	0

Exercícios

5. Descreve a entidade e a arquitetura do módulo ModuloTopo. Considere que os módulos: Incrementador, Armazenador, ControlaSaida e ControlaEntrada devam apenas ser mapeados (port map) na arquitetura do ModuloTopo



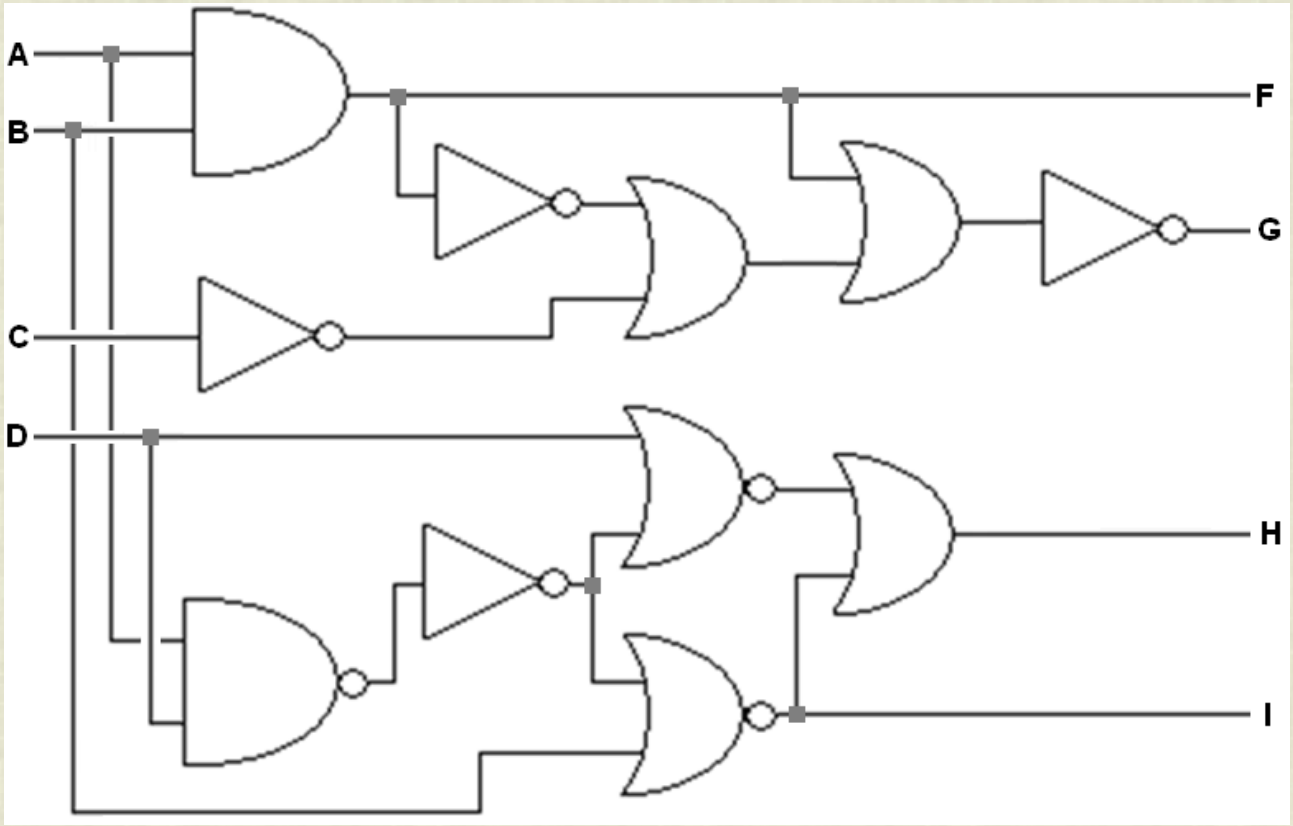

```
entity InterfaceExterna is
  port
    (
      entraDado: in std_logic_vector(7 downto 0);
      c, r: in std_logic;
      recebe: in std_logic; aceita: out std_logic;
      ack: in std_logic; envia: out std_logic;
      saiDado: out std_logic_vector(7 downto 0)
    );
end InterfaceExterna;
```

```
architecture InterfaceExterna of InterfaceExterna is
        signal incrm: std_logic_vector(3 downto 0);
        signal dadoSaida: std_logic_vector(7 downto 0);
begin
  T1: entity Incrementador port map(ck=>c, rst=>r, incrm=>incrm);
  T2: entity controlaEntrada port map(ck=>c, rst=>r, incrm=>incrm,
                                      dI=>entraDado, snd=>envia, ack=>ack);
  T3: entity Armazenador port map(ck=>c, rst=>r,
                                  dadoI=>entraDado, dadoO=>dadoSaida);
  T4: entity controlaSaida port map(ck=>c, rst=>r,
                                    incrm=>incrm, rcb=>recebr, ack=>aceita);

  saiDado<= dadoSaida;
end InterfaceExterna;
```

Exercícios

6. Determine o maior caminho do circuito abaixo. Considerando que portas NOT (inversores) atrasam 1ns, portas NAND (não e lógico) e NOR (não ou lógico) atrasam 3ns, portas AND (e lógico) e portas OR (ou lógico) atrasam 5ns.



Exercícios

7. Jogo dos 8 erros em VHDL. O código VHDL abaixo contém exatamente 8 falhas, que você deve identificar, deixando claro qual o erro. Os erros são de diversas naturezas, sintaxe, contexto, semântica.

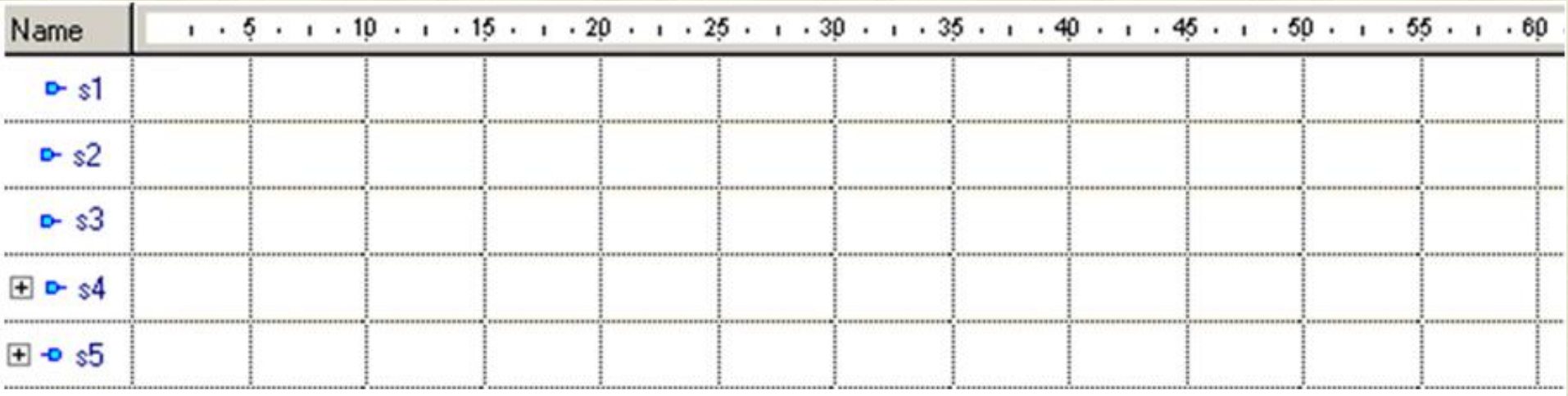
```
1. library IEEE;
2. entity p1 is
3.       port(  A1; B2; C3: in STD_LOGIC;
4.              D1; E2; F3; G4: out STD_LOGIC);
5. end p1;
6. architecture Prova1 of p1 is
7.      signal iD1, iE2, iF3, iG4: out std_logic;
8.      signal S: std_logic_vector(3 downto 0);
9. begin
10.     D1 <= iD1; E2 <= iE2; F3 <= iF3; G4 <= iG4;
11.     A1 <=  C3 and (B2 or iD1);
12.     iD1 <= A1 and B2; iE2 <= A1 xor C3; iF3 <= A1 or C3;
13.     iG4 <= iF3 nand (iE2 xor (B2 and C3));
14.     S(1 downto 0) <= iE2 & iF3 & iD1;
15.     with (D1 & E2 & F3) select
16.             S <=    "1011" when "111",
17.                     "0101" when "110",
18.                     "0101" when "011"| "100" | "000",
19.                     "1100" when "010",
20.                     "1011" when others;
21.end p1;
```

Exercícios

8. Faça uma figura com os primeiros 100ns das formas de onda geradas pelo testbench descrito a seguir. Considere os sinais **s1, s2, s3, s4** e **s5**:

```
entity E_P is
  port
  (
    s1, s2: in STD_LOGIC;
    s3: out STD_LOGIC;
    s4: in STD_LOGIC_VECTOR(3 downto 0);
    s5: out STD_LOGIC_VECTOR(3 downto 0)
  );
end E_P;
```

```
architecture P of E_P is
begin
  s3 <= not (s1 and s2) after 10ns;
  s5 <= not s4 after 5ns;
end P;
```



Exercícios

8. (continuação)

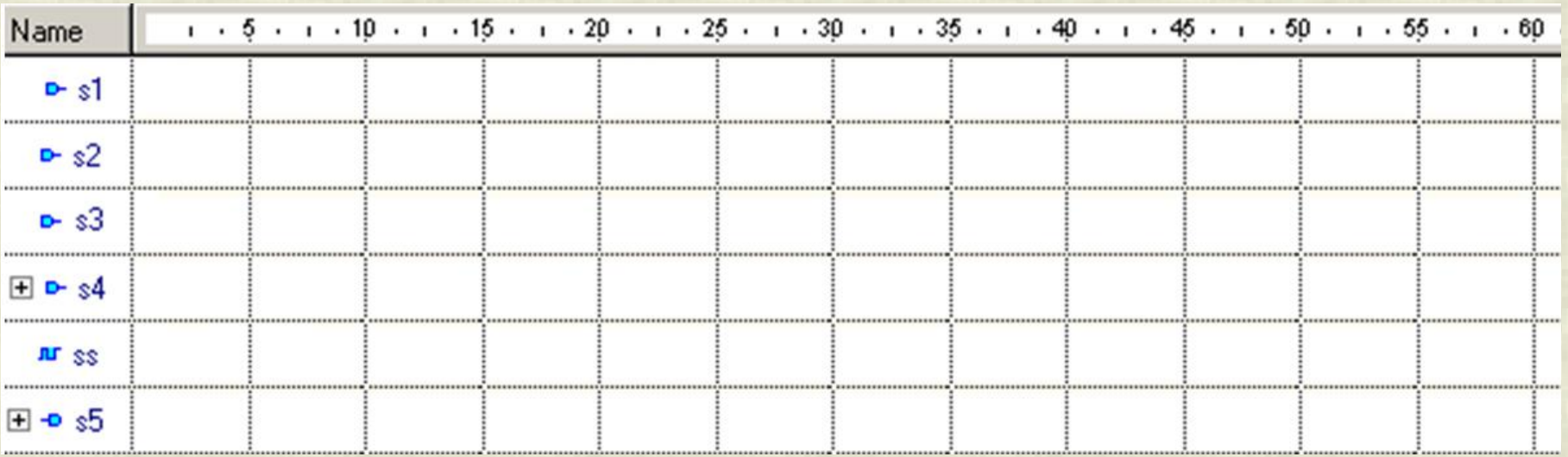
```
entity e_p_tb is
end e_p_tb;

architecture TB of e_p_tb is
  signal s1, s2, s3: std_logic;
  signal s4: std_logic_vector(3 downto 0) := (others=>'0');
  signal s5: std_logic_vector(3 downto 0);
begin
  U: entity work.E_P port map(s1=>s1, s2=>s2, s3=>s3, s4=>s4, s5=>s5);
  process
  begin
    s1 <= '0', '1' after 20ns;
    wait for 50ns;
  end process;
  process
  begin
    s2 <= '1', '0' after 15ns, '1' after 25ns;
    wait for 30ns;
  end process;
  process
  begin
    s4 <= s4 + 1 after 25ns;
    wait for 60ns;
  end process;
end TB;
```

Exercícios

9. Faça as formas de onda geradas pela execução dos **60ns** iniciais do testbench descrito a seguir:

```
architecture TB of TbAtraso is
    signal s1, s2, s3, ss: std_logic;
    signal s4, s5: std_logic_vector(3 downto 0) := (others=>'0');
begin
    s3 <= not ss;
    ss <= s1 xor s2;
    s5 <= s4 + ss;
    process
    begin
        s1 <= '1', '0' after 10ns;
        wait for 50ns;
    end process;
    process
    begin
        s2 <= '1', '0' after 10ns, '1' after 15ns, '0' after 25ns;
        wait for 60ns;
    end process;
    process
    begin
        s4 <= s4 + "0101" after 30ns;
        wait for 45ns;
    end process;
end TB;
```

Exercícios

9. (continuação)