

Av2 - 1º Semestre de 2025

Avaliação 2 - Elementos de Sistemas

Pontos HW	Pontos SW
60	0

- Avaliação individual.
- 120 min total.
- Ficar no blackboard durante a prova.
- Fazer commit ao final de cada questão.
- Lembre de dar **push** ao final.

architecture rtl of mult2 is

1. Multiplicador

begin

C <= "0001" when (A="01" and B="01") else

"0010" when ((A="01" and B="10") or (B="01" and A="10")) else

"0011" when ((A="01" and B="11") or (B="01" and A="11")) else

"0100" when (A="10" and B="10") else

"0110" when ((A="10" and B="11") or (B="10" and A="11")) else

"1001" when (A="11" and B="11") else

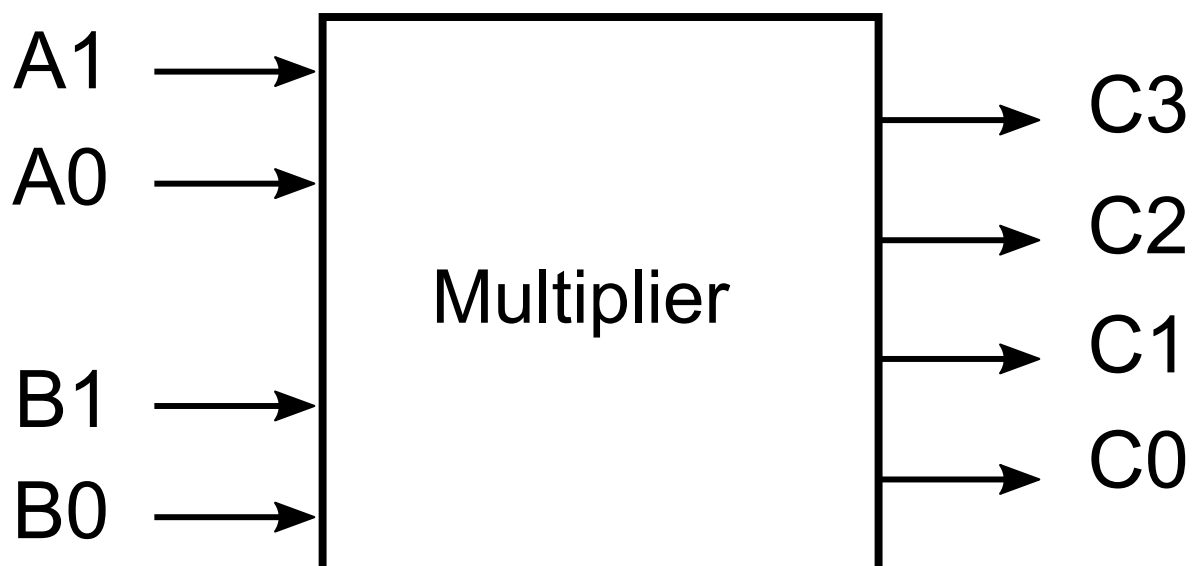
"0000";

end;

Queremos fazer um código VHDL para um bloco lógico que realize a multiplicação de bits.

Considere que o bloco possua duas entradas de 2 bits cada e a saída possua 4 bits, como ilustrado na figura abaixo. O valor binário representado na saída deve ser numericamente igual a multiplicação dos números em binários da entrada.

Considere que todos os números são positivos.



{width=300}

```
outra opção:  
C(0) <= A(0) and B(0);  
C(1) <= (A(1) and B(0)) xor (A(0) and B(1));  
C(2) <= (A(1) and B(1)) xor ((A(1) and B(0)) and (A(0) and B(1)));  
C(3) <= (A(1) and B(1)) and ((A(1) and B(0)) and (A(0) and B(1)));
```

Por exemplo, se as entradas forem A="10" e B="01", a saída deve ser C="0010".

Implementação

Implemente o código em VHDL no arquivo src/mult2.vhd .

Lembre de testar com `pytest -k mult2

Rubrica para avaliação:

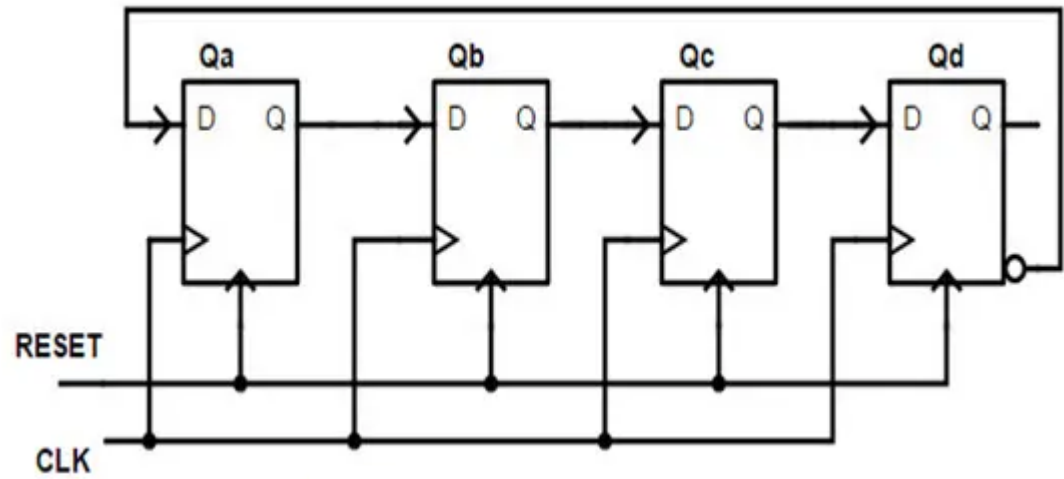
Pontos HW	Descritivo
15	Implementação correta
?	Implementações incorretas serão analisadas caso a caso

2. Circuito sequencial

Pontos HW	Pontos SW
15	0

t	Qa	Qb	Qc	Qd
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0
9	1	0	0	0
10	1	1	0	0

Considere o circuito da figura abaixo



{width=300}

Questões:

- 1. Qual as saídas (Qa, Qb, Qc , Qd) para as próximas 10 bordas de subida, considerando que até o instante t=0 o reset estava ativado?

Responda no arquivo src/circuito.txt .

Rubrica para avaliação:

Pontos HW	Descritivo
1.5	Para a saída certa em cada instante de tempo

3. Lógica combinacional/sequencial

Pontos HW	Pontos SW
15	0

Considere o código VHDL a seguir:

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

entity X is
port (
    clock: in std_logic;
    A,B: in std_logic_vector(7 downto 0);
    IAB: in std_logic;
    Output: out std_logic
);
end comparator;

architecture Behavioral of X is
    signal AB: std_logic_vector(7 downto 0);
    signal Result: std_logic;

begin

    AB(0) <= (not A(0)) xnor (not B(0));
    AB(1) <= (not A(1)) xnor (not B(1));
    AB(2) <= (not A(2)) xnor (not B(2));
    AB(3) <= (not A(3)) xnor (not B(3));
    AB(4) <= (not A(4)) xnor (not B(4));
    AB(5) <= (not A(5)) xnor (not B(5));
    AB(6) <= (not A(6)) xnor (not B(6));
    AB(7) <= (not A(7)) xnor (not B(7));

    Result <= '1' when (AB = x"FF" and IAB = '1') else '0';
    Output <= Result;

end Behavioral;
```

Questões:

1. Explique o que o circuito faz no arquivo `src/X.txt` . Comparador, sendo que IAB faz uma "habilitação"
2. Altere o arquivo `src/X.vhd` para que o circuito tenha um funcionamento síncrono com a borda de subida do clock (não tem teste).

Rubrica para avaliação:

Pontos HW	Descritivo
10	Questão 1
5	Questão 2

```
process(clock)
begin
    if (rising_edge(CLOCK)) then
        Result <= '1' when (AB = x"FF" and IAB = '1') else '0';
    end if;
end process;
Output <= Result;
```

4. Decodificação

A figura a seguir ilustra a tabela verdade de um decodificador 3-8. Para cada combinação de entrada possível, há sete saídas que são iguais a 0 e apenas uma delas é igual a 1. A saída cujo valor é igual a 1 representa o equivalente em mintermo do número binário atualmente disponível nas linhas de entrada.

INPUTS			OUTPUTS							
X	Y	Z	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

{width=300}

Questões

A partir da tabela, pede-se:

1. Implemente o decodificador no arquivo `src/decoder3to8.vhd` .
2. Implemente um decodificador 4-16 no arquivo `src/decoder4to16.vhd` utilizando **necessariamente** decodificadores 3-8 através de port map. Considere que a quarta entrada seja W.
3. Implemente no arquivo `src/decoder_ascii.vhd` uma saída (codigo_ascii de 8 bits) que retorna o código ASCII correspondente ao número da entrada XYZ em binário (caracteres de 0 a 7).

Lembre de testar com ``pytest -k decoder`

Rubrica para avaliação:

Pontos HW	Descritivo
5	Implementação correta de cada item
?	Implementações incorretas serão analisadas caso a caso

architecture rtl of decoder3to8 is

signal XYZ: STD_LOGIC_VECTOR(2 downto 0);

begin

XYZ <= X & Y & Z;

D <= "00000001" when XYZ="000" else
 "00000010" when XYZ="001" else
 "00000100" when XYZ="010" else
 "00001000" when XYZ="011" else
 "00010000" when XYZ="100" else
 "00100000" when XYZ="101" else
 "01000000" when XYZ="110" else
 "10000000" when XYZ="111" else
 "00000000";

end architecture;

architecture rtl of decoder4to16 is

component decoder3to8 is

port(

X,Y,Z : in STD_LOGIC;

D : out STD_LOGIC_VECTOR(7 downto 0)

);

end component;

signal inter: STD_LOGIC_VECTOR(7 downto 0);

begin

u1: decoder3to8 port map(X,Y,Z,inter);

D(7 downto 0) <= inter when W='0' else (others=>'0');

D(15 downto 8) <= inter when W='1' else (others=>'0');

end architecture;

architecture rtl of decoder_ascii is

signal num: integer range 0 to 57;

signal num_concat: STD_LOGIC_VECTOR(2 downto 0);

begin

num_concat <= X & Y & Z;

num <= to_integer(unsigned(num_concat));

codigo_ascii <= STD_LOGIC_VECTOR(to_unsigned(num+48,8));

end architecture;