



## **Laboratório 4: Análise e síntese de circuitos sequenciais assíncronos no modo fundamental e Descrição de Circuitos Combinacionais em VHDL .**

Bernardo Hoffmann da Silva  
Marcos Vinicius Pereira Veloso

### **4. Montagem**

#### **4.1. Projeto e montagem de uma máquina sequencial assíncrona (Máquina de Huffman)**

Conforme discutido na metodologia adotada, após a obtenção da tabela de estados do circuito sequencial analisado, pode-se obter a função lógica de saída  $Z$  para a Máquina de Huffman como especificado:

$$Y_0(t+1) = \overline{X_0}X_1 + X_1Y_0(t) \quad (1)$$

$$Y_1(t+1) = X_1 + X_0Y_1(t) \quad (2)$$

$$Z = X_0\overline{Y_0(t)}Y_1(t) \quad (3)$$

Para a implementação, considere a representação lógica da função dada em (3) com a nomenclatura dos sinais internos especificados conforme denotado na Figura 1. Esses serão relevantes para a montagem com os circuitos integrados requisitados.

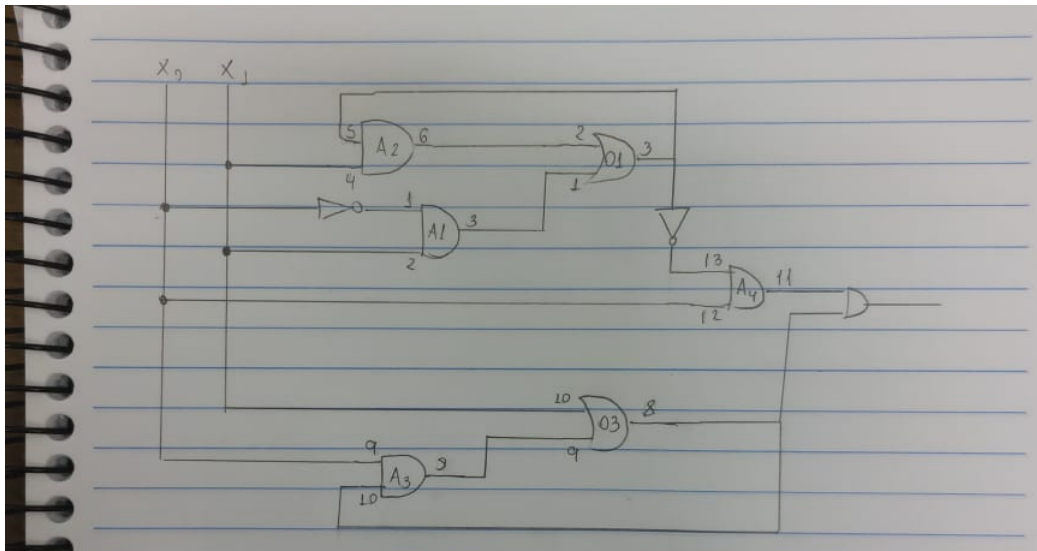
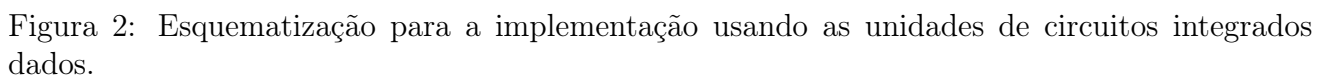


Figura 1: Esquematização da saída especificada na Relação (3), com a numeração especificada para implementação em protoboard.



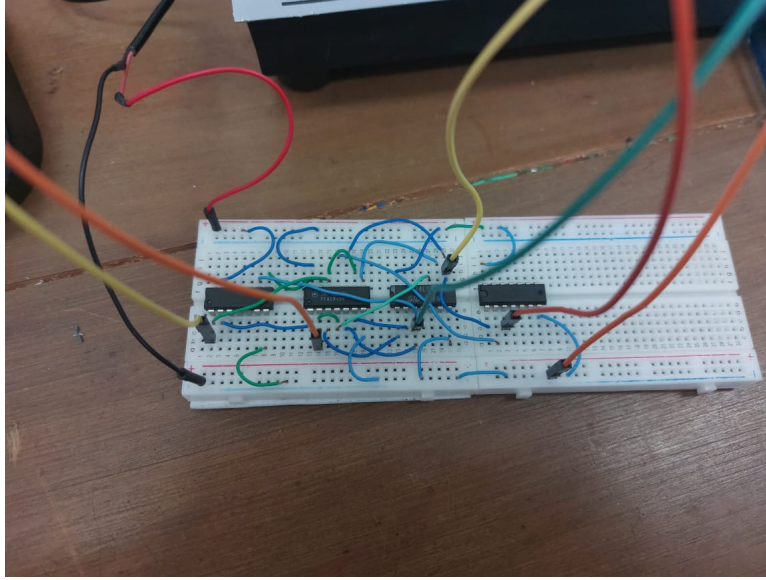


Figura 3: Construção prática do circuito requisitado usando unidades integradas, conforme esquema elétrico denotado na Figura 2.

Feito isso, prossegue-se com a associação do circuito da Figura 1 nas respectivas entradas dos circuitos integrados utilizados neste projeto, que consistem nos modelos das famílias TTL7400, a saber: CI7404 (portas NOT), CI7408 (portas AND) e CI7432 (portas OR). A síntese de tal circuito é denotado na Figura 3, feita em laboratório.

Por fim, pode-se então prosseguir com a verificação do comportamento do circuito, cujo resultado é condizente com o previsto das especificações de Máquina de Huffman. Para fins de consulta, segue na seção de Referências, em [1], em que há registro em nuvem dos resultados de saídas obtidos.

## 5. Simulação em VHDL

### 5.1 Descrição VHDL e Simulação de uma ULA de 8 bits

Para a implementação da ULA de 8 bits especificado, pode-se utilizar do Mapa de Karnaugh reduzido considerando as variáveis  $M$ ,  $S_1$  e  $S_0$ . Com isso, as saídas são denotadas por

$$F_i = ((S_i B_i) \oplus (M C_i)) \oplus (A_i \oplus S_0) \quad (4)$$

$$C_{i+1} = (A_i \oplus S_0)[(S_i B_i) \oplus (M C_i)] + (S_i B_i)(M C_i) \quad (5)$$

Desse modo, pode-se então implementar a descrição da ULA em VHDL tal como presente nas relações (4) e (5), de acordo como segue no script abaixo:

```

library IEEE;
use IEEE.std_logic_1164.all;

--CELULA

entity cell is
port (Ai,Bi,S0,S1,M,Ci : in std_logic;
      F, Cout : out std_logic );
end cell;

architecture cellArc of cell is
  signal aux1, aux2,aux3, aux4, aux5, aux6 : std_logic;
begin
  aux1 <= Ci and M;
  aux2 <= Bi and S1;
  aux3 <= Ai xor S0;
  aux4 <= aux1 xor aux2;
  F <= aux4 xor aux3;
  aux5 <= aux3 and aux4;
  aux6 <= aux1 and aux2;
  Cout <= aux5 or aux6;
end cellArc;

-- ULA

library IEEE;
use IEEE.std_logic_1164.all;

entity lab5VHDL is
port(A: in std_logic_vector (7 downto 0);
      B: in std_logic_vector (7 downto 0);
      M,S_0,S_1,Cin : in std_logic;
      F: out std_logic_vector (7 downto 0) );
end lab5VHDL;

architecture ULAarc of lab5VHDL is
  component cell
    port (Ai,Bi,S0,S1,M,Ci : in std_logic;
          F, Cout : out std_logic );
  end component;
  signal C: std_logic_vector (7 downto 0);
begin
  B1 : cell port map (Ai => A(0), Bi => B(0),S0 => S_0, S1 => S_1, M => M,
    Ci => Cin, F => F(0), Cout => C(0));

```

```

B2 : cell port map (Ai => A(1), Bi => B(1),S0 => S_0, S1 => S_1, M => M,
Ci => C(0), F => F(1), Cout => C(1));

B3 : cell port map (Ai => A(2), Bi => B(2),S0 => S_0, S1 => S_1, M => M,
Ci => C(1), F => F(2), Cout => C(2));

B4 : cell port map (Ai => A(3), Bi => B(3),S0 => S_0, S1 => S_1, M => M,
Ci => C(2), F => F(3), Cout => C(3));

B5 : cell port map (Ai => A(4), Bi => B(4),S0 => S_0,S1 => S_1, M => M,
Ci => C(3), F => F(4), Cout => C(4));

B6 : cell port map (Ai => A(5), Bi => B(5),S0 => S_0,S1 => S_1, M => M,
Ci => C(4), F => F(5), Cout => C(5));

B7 : cell port map (Ai => A(6), Bi => B(6),S0 => S_0,S1 => S_1, M => M,
Ci => C(5), F => F(6), Cout => C(6));

B8 : cell port map (Ai => A(7), Bi => B(7),S0 => S_0,S1 => S_1, M => M,
Ci => C(6), F => F(7), Cout => C(7));

end architecture;

```

Feito isso, prossegue-se com a simulação funcional do circuito com o uso do software *Quartus II*. Por fim, implementa-se a arquitetura da ULA completa conforme Figura 4, em que cada célula unitária é denotada na Figura 5; o resultado do diagrama de temporização é denotado na Figura 6.

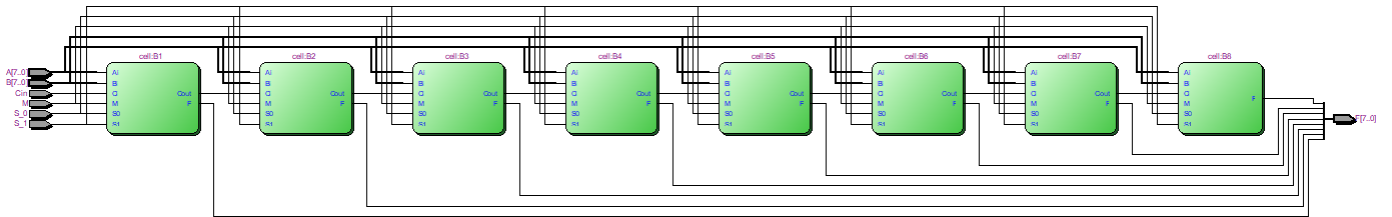


Figura 4: Esquematização da ULA completa.

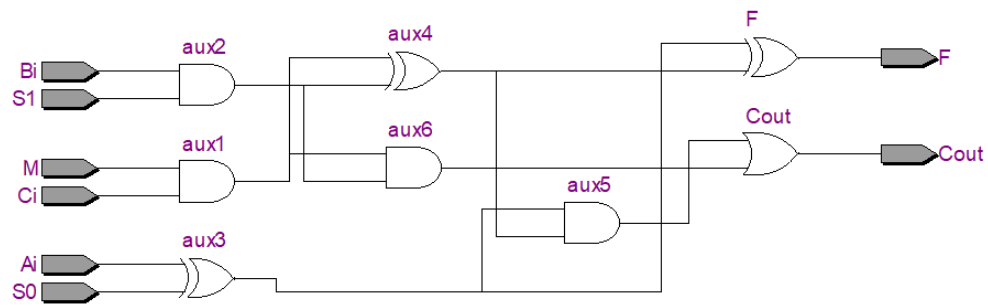


Figura 5: Esquematização da célula básica da ULA completa.

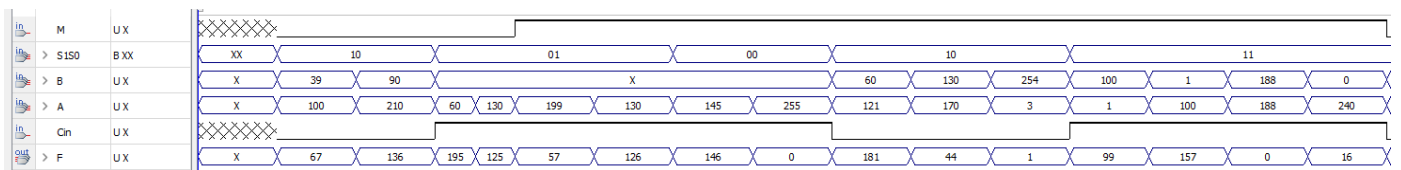


Figura 6: Simulação temporal obtida para algumas entradas da ULA completa implementada.

## 1 6. Referências

[1] Arquivos de imagem do resultado de saída do Projeto especificado. Disponível em <<https://drive.google.com/drive/folders/1EePUTsyRpyUr1jB0hJ3HmJjNGCIh-ogo>>