

UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP
Ciência da Computação



ELETRÔNICA PARA COMPUTAÇÃO

LISTA

Aluno:

Marcus Vinícius Souza Fernandes

Matrícula: 19.1.4046

Ouro Preto

2019

1)

a) Equação: $S = ((\sim(A \cdot (B + C))) \cdot D) \cdot (\sim(A + B))$

- Com Teorema de Morgan, obtemos:

$$S = ((\sim(A + (\sim B \cdot \sim C))) \cdot D) \cdot (\sim A \cdot \sim B)$$

- Multiplicando, obtemos:

$$S = (\sim A \cdot D + \sim B \cdot \sim C \cdot D) \cdot (\sim A \cdot \sim B)$$

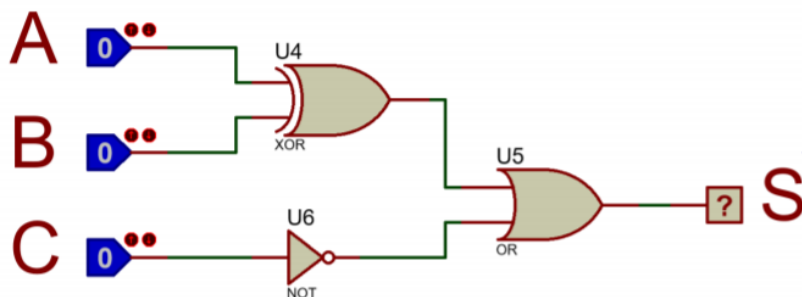
- Multiplicando mais uma vez e simplificando:

$$S = (\sim A \cdot \sim B \cdot D + \sim B \cdot \sim C \cdot D \cdot \sim A)$$

- Colocando o termo comum em evidência e simplificando, obtemos:

$$S = \sim A \cdot \sim B \cdot D$$

Veja a seguir o circuito equivalente:



b) Equação: $S = (\sim A + \sim B + \sim C) \cdot (A + B + \sim C)$

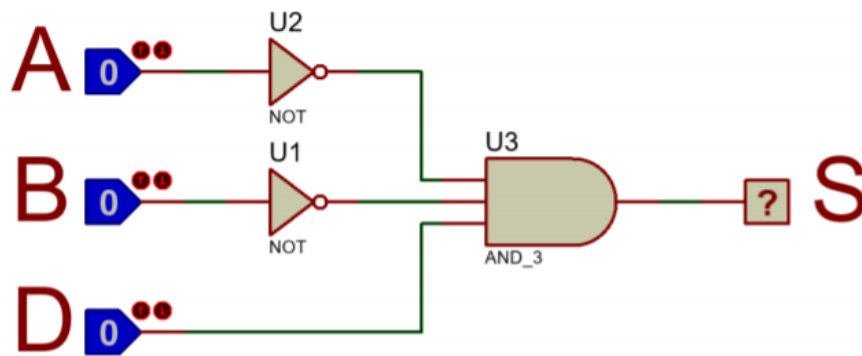
- Multiplicando e simplificando:

$$S = (\sim A \cdot B + \sim A \cdot \sim C + A \cdot \sim B + A \cdot \sim C + \sim B \cdot \sim C + B \cdot \sim C + \sim C)$$

- Colocando o termo comum em evidência e simplificando, temos:

$$S = (\sim C + \sim A \cdot B + A \cdot \sim B)$$

Veja a seguir o circuito equivalente:



c) Equação: $S = \{ \sim[(A + B) \cdot C] \} + \{ \sim[D \cdot (C + B)] \}$

- Teorema de Morgan:

$$S = \{ \sim(A + B) + \sim C \} + \{ \sim D + \sim(C + B) \}$$

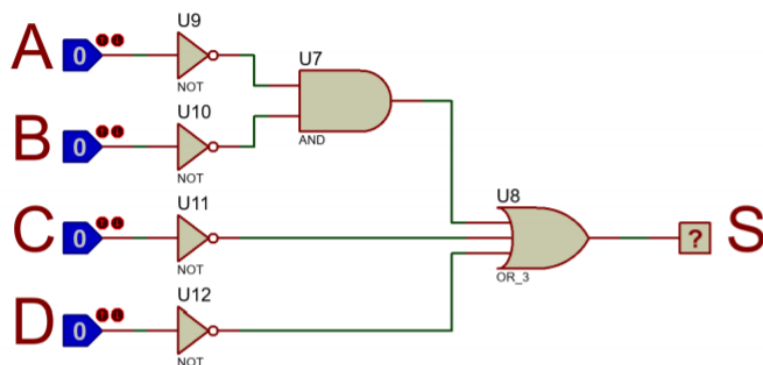
- Aplicando novamente:

$$S = \{ (\sim A \cdot \sim B) + \sim C \} + \{ \sim D + \sim C \cdot \sim B \}$$

- Colocando o termo comum em evidência e simplificando:

$$S = \sim A \cdot \sim B + \sim C + \sim D$$

Veja a seguir o circuito equivalente:



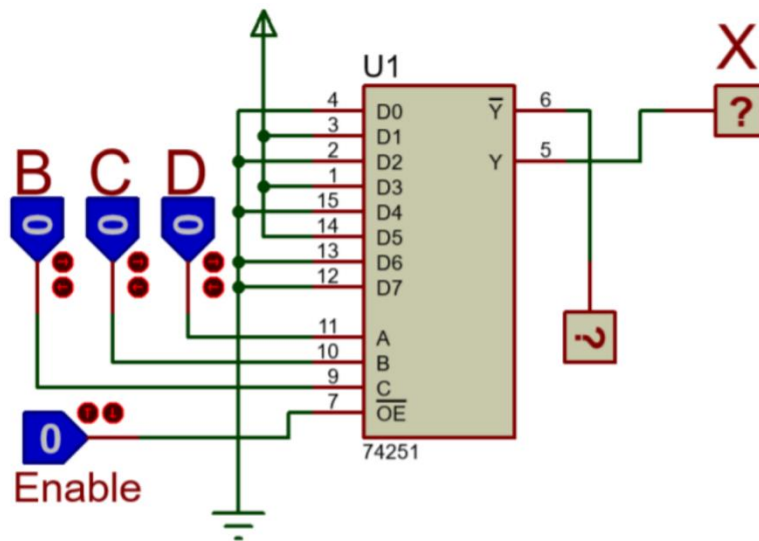
2)

As expressões equivalentes do circuito:

$$- X = (\sim B \cdot \sim C) \cdot D$$

$$- Y = \sim A \cdot B \cdot C \cdot D$$

Montando o circuito utilizando MUX, para X:



Implementação do Verilog:

```
module mux_8_1 (Vcc, GRD, B, C, D, Xis);

    input B, C, D, Vcc, GRD;
    output Xis;

    reg Xis;

    always @ (*) begin
        case ({B, C, D}) //mux_8_1
            3'b000: Xis = GRD;
            3'b001: Xis = Vcc;
            3'b010: Xis = GRD;
            3'b011: Xis = Vcc;
            3'b100: Xis = GRD;
            3'b101: Xis = Vcc;
            3'b110: Xis = GRD;
            3'b111: Xis = GRD;
        endcase
    end
endmodule

module quest2_mux_8_1;

    reg t_B;
    reg t_C;
    reg t_D;
    reg t_Vcc;
    reg t_GRD;

    wire t_Xis;

    initial begin: simul_stop
        #8 $stop;
    end

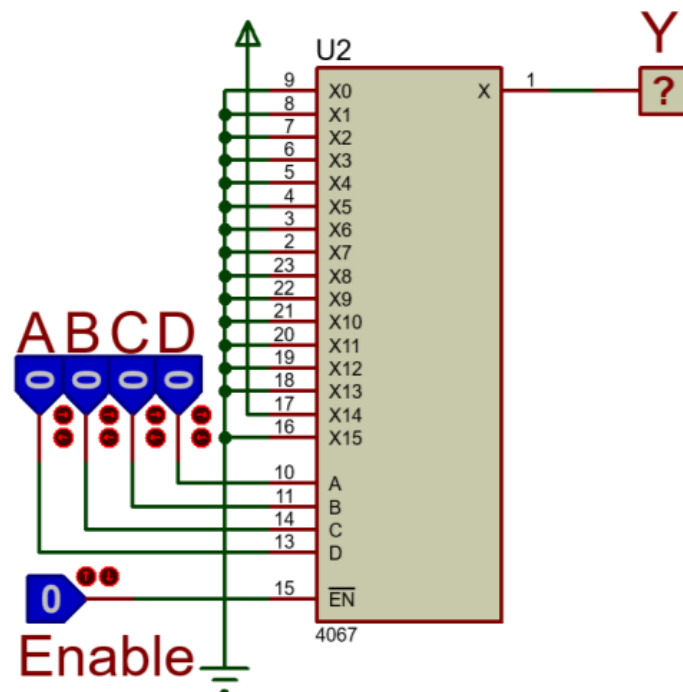
    initial begin
        $display("Time \t t_B \t t_C \t t_D \t X");
        $monitor(" %0d \t %b \t %b \t %b \t %b",
            $time, t_B, t_C, t_D, t_Xis);
    end

    initial begin: start
        t_B = 0;
        t_C = 0;
        t_D = 0;
        t_Vcc = 1;
        t_GRD = 0;
    end

    always begin: process
        #1 {t_B, t_C, t_D} = {t_B, t_C, t_D} + 1;
    end

    mux_8_1 simul (.B(t_B), .C(t_C), .D(t_D),
        .GRD(t_GRD), .Vcc(t_Vcc),
        .Xis(t_Xis));
endmodule
```

Montando o circuito utilizando MUX, para Y:



Implementação do Verilog:

```
module mux_16_1 (Vcc, GRD, A, B, C, D, Y);
    input A, B, C, D, Vcc, GRD;
    output reg Y;

    always @ (*) begin
        case ({A, B, C, D}) //mux_16_1
            4'b0000: Y = GRD;
            4'b0001: Y = GRD;
            4'b0010: Y = GRD;
            4'b0011: Y = GRD;
            4'b0100: Y = GRD;
            4'b0101: Y = GRD;
            4'b0110: Y = GRD;
            4'b0111: Y = GRD;
            4'b1000: Y = GRD;
            4'b1001: Y = GRD;
            4'b1010: Y = GRD;
            4'b1011: Y = GRD;
            4'b1100: Y = GRD;
            4'b1101: Y = GRD;
            4'b1110: Y = Vcc;
            4'b1111: Y = GRD;
        endcase
    end
endmodule
```

```

module quest2_mux_16_1;

    reg t_A;
    reg t_B;
    reg t_C;
    reg t_D;
    reg t_Vcc;
    reg t_GRD;

    wire t_Y;

    initial begin: simul_stop
        #16 $stop;
    end

    initial begin
        $display("Time \t t_A \t t_B \t t_C \t t_D \t Y");
        $monitor(" %0d \t %b \t %b \t %b \t %b \t %b",
            $time, t_A, t_B, t_C, t_D, t_Y);
    end

    initial begin: start
        t_A = 0;
        t_B = 0;
        t_C = 0;
        t_D = 0;
        t_Vcc = 1;
        t_GRD = 0;
    end

    always begin: process
        #1 {t_A, t_B, t_C, t_D} = {t_A, t_B, t_C, t_D} + 1;
    end

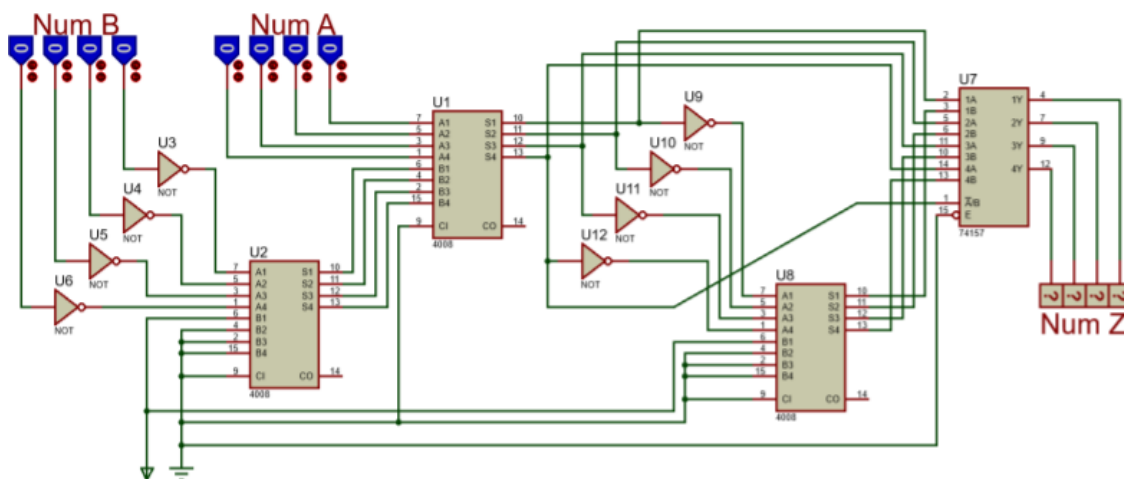
    mux_16_1 simul (.A(t_A), .B(t_B), .C(t_C), .D(t_D),
        .GRD(t_GRD), .Vcc(t_Vcc),
        .Y(t_Y));

endmodule

```

5)

Implementação do circuito no MUX:



Implementação do Verilog:

```
module somaSub(A, B, Ze);

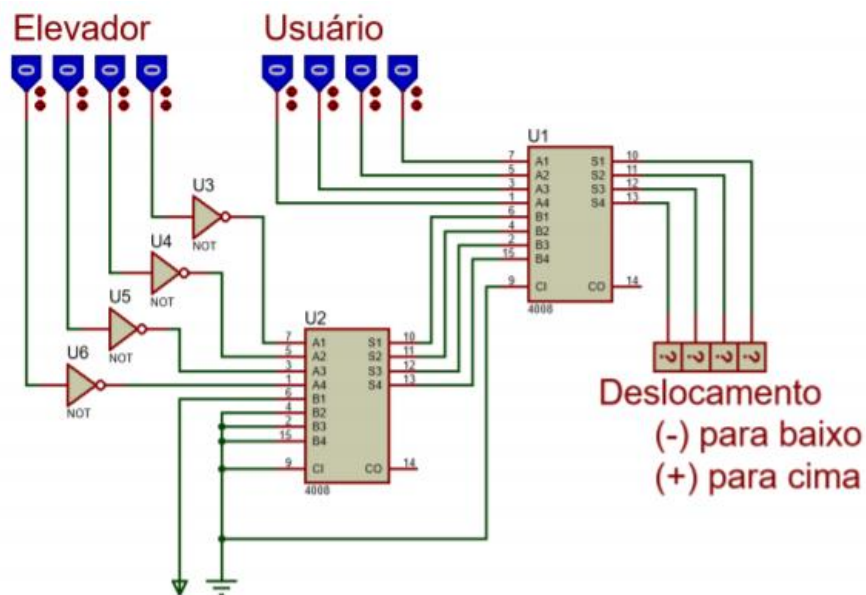
    input [3:0] A, B;
    output reg [3:0] Ze;

    always @ ( * ) begin
        if (A >= B)
            Ze = A - B;

        else
            Ze = B - A;
    end
endmodule
```

9)

Circuito MUX:



Obs: O bit de sinal indica o sentido de deslocamento e os outros 3 bits indicam o número de andares.

Implementação Verilog:

```
module elevador(A, B, S);

    input [3:0] A, B;
    output reg [3:0] S;

    always @ ( * ) begin
        S = A - B;
    end
endmodule
```

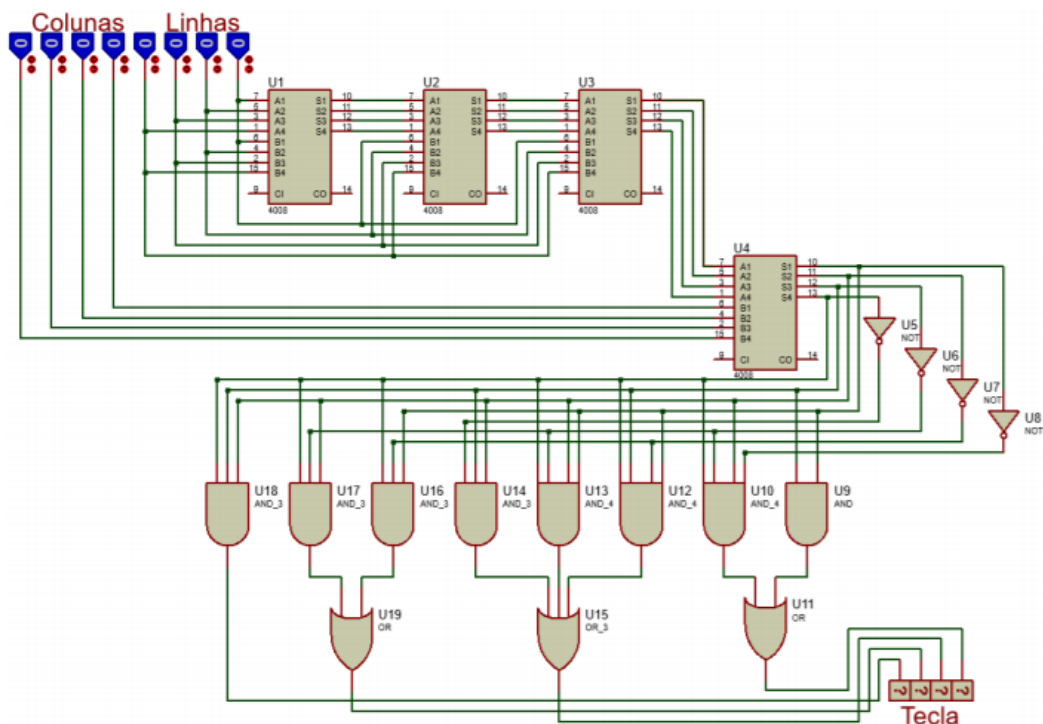

11)

Seja a matriz abaixo um teclado numérico, organizado com linhas e colunas da seguinte forma:

1	2	3	0001
4	5	6	0010
7	8	9	0011
			0001 0010 0011

A tecla zero é um número separado da matriz que corresponde a palavra binária 0000.

Sendo assim o circuito abaixo foi montado utilizando somadores para encontrar a combinação de linhas e colunas correspondente ao número pressionado, bem como um decodificador para transformar o código correspondente a posição do número na matriz para o seu valor numérico. O circuito poderia ser melhorado colocando um sensor quando a tecla fosse pressionada para não emitir o número zero quando nada estivesse sendo pressionado



12)

Pela tabela abaixo, podemos extrair as seguintes equações:

				<u>W</u>		<u>X</u>
<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>Z</u>		<u>Y</u>
0	0	0	0	.		.
0	0	0	1	.		.
0	0	1	0	.		.
0	0	1	1	.		.
0	1	0	0	.		.
0	1	0	1	.		.
0	1	1	0	.		.
0	1	1	1	.		.
1	0	0	0	.		.
1	0	0	1	.		.

$$W = \sim A \cdot (D + C + B) + A \cdot \sim B \cdot \sim C \cdot \sim D$$

$$X = A \cdot \sim B \cdot \sim C \cdot D + \sim A \cdot \sim (C \oplus D) + \sim A \cdot B \cdot C$$

$$Y = \sim A \cdot \sim C \cdot \sim D + \sim B \cdot \sim C \cdot \sim D + \sim A \cdot B \cdot D$$

$$Z = \sim A \cdot \sim B \cdot \sim D + \sim A \cdot B \cdot C + A \cdot \sim B \cdot \sim C$$

Montando o circuito equivalente, temos:

