

**UNIVERSIDADE FEDERAL DE OURO PRETO
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**CRISTIAN ANSELMO DE SOUZA PINTO - 21.2.4005
CÉSAR GABRIEL DE PAULA BATISTA - 21.2.4176
IVES HENRIQUE SENIBALDE DE OLIVEIRA- 21.2.4184
JULIANA APARECIDA BORGES - 21.2.4156
LUCAS DE OLIVEIRA BARBOZA- 21.2.4173
MARIA CLARA MIRANDA DE SÁ- 21.2.4187**

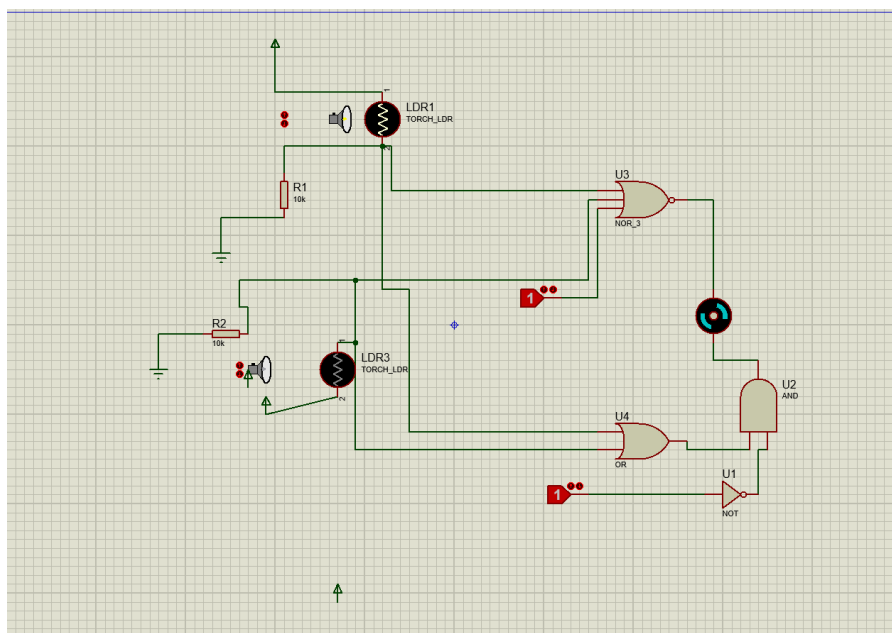
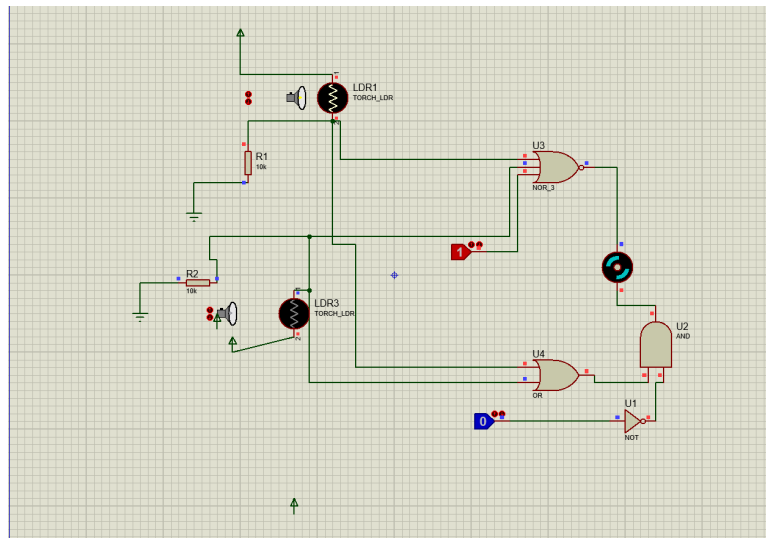
**ATIVIDADE ABERTA 01 DE ELETRÔNICA PARA COMPUTAÇÃO
BCC-265**

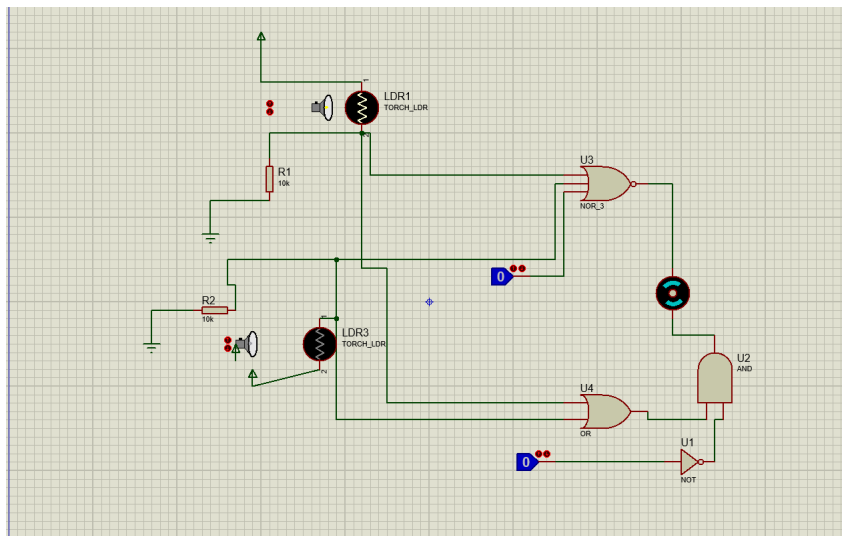
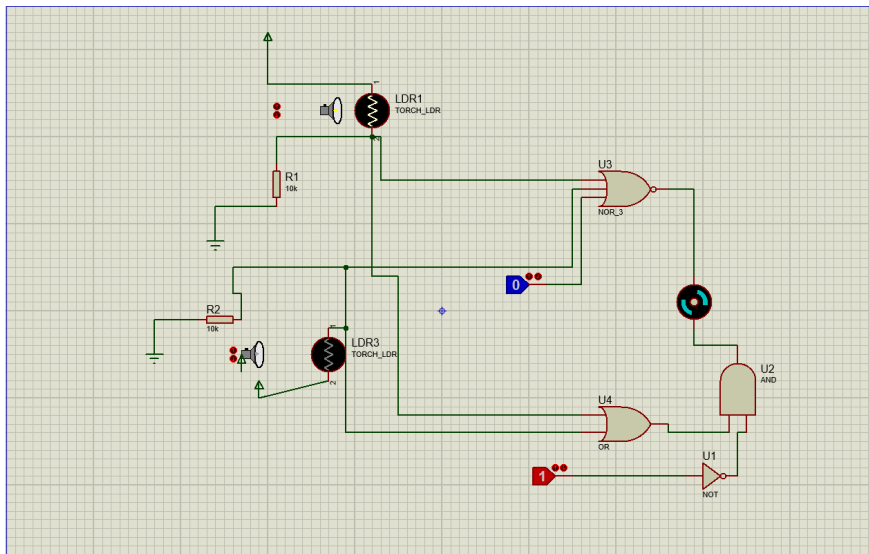
**OURO PRETO
2022**

Nessa atividade aberta tivemos que criar um circuito no Proteus, onde teria que supor a existência de um lugar que deveria ser coberto e descoberto dependendo da ocorrência do sol ou da chuva. Quando estivesse com luminosidade em excesso ou chuva em excesso, o lugar deveria ser coberto, caso contrário, o lugar deveria permanecer descoberto.

Portanto, a cobertura seria acionada através de um motor elétrico que giraria no sentido horário ou no sentido anti horário. Com esse objetivo, o circuito montado deveria ler o status de um sensor de luminosidade (L) e de um sensor de umidade (U).

As saídas do circuito deveriam ser compatíveis com as entradas de uma “ponte H”. Então, para sabermos se a cobertura está totalmente fechada ou aberta, acha-se a presença de dois interruptores para determinar o “fim de curso”. No Proteus, para representar os sensores L e U, utilizamos “LDRs”. Usamos o tipo “LogicState” para representar os interruptores Fd (Fim de curso direito) e Fe (Fim de curso esquerdo).





VERILOG

Código:

```
module controla_motor(x, y, fe, fd, a, b);
```

```
    input x, y, fe, fd;
```

```
    output a, b;
```

```
    reg a, b;
```

```
    always@(*)
    begin
```

```
        a = ~fd & ~x & ~y;
```

```
        b = ~fe & (y | x);
```

```
    end
```

```
endmodule
```

```

module simulacao;
    reg teste_x, teste_y, teste_fe, teste_fd;
    wire teste_a, teste_b;

    initial
        begin
            teste_x = 0;
            teste_y = 0;
            teste_fe = 0;
            teste_fd = 0;
        end

    initial
        begin
            #16 $stop;
        end

    initial
        begin
            $display("\t\t TEMPO fd fe x y a b");
            $monitor("%d %b %b %b %b %b %b", $time, teste_fd,
teste_fe, teste_x, teste_y, teste_a, teste_b);
        end

    always
        begin
            #1 {teste_fd, teste_fe, teste_x, teste_y} = {teste_fd, teste_fe, teste_x,
teste_y} + 1;
        end

    controla_motor(.x(teste_x), .y(teste_y), .fe(teste_fe), .fd(teste_fd), .a(teste_a),
.b(teste_b));

endmodule

```

Compilação:

TEMPO	fd	fe	x	y	a	b
0	0	0	0	0	1	0
1	0	0	0	1	0	1
2	0	0	1	0	0	1
3	0	0	1	1	0	1
4	0	1	0	0	1	0
5	0	1	0	1	0	0
6	0	1	1	0	0	0
7	0	1	1	1	0	0
8	1	0	0	0	0	0
9	1	0	0	1	0	1
10	1	0	1	0	0	1
11	1	0	1	1	0	1
12	1	1	0	0	0	0
13	1	1	0	1	0	0
14	1	1	1	0	0	0
15	1	1	1	1	0	0