

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

GABRIEL VARGAS BENTO DE SOUZA

EXERCÍCIO PRÁTICO 02
Arquitetura de Computadores II

Belo Horizonte - MG
2023

Parte 1: Estudo da ULA, usando o Logisim

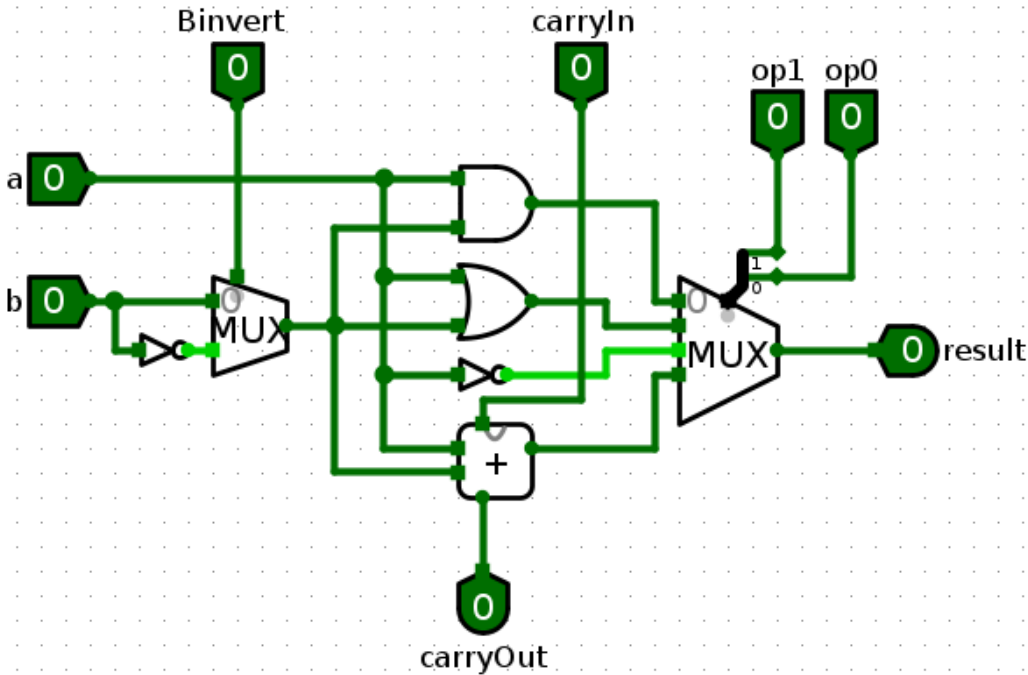


Figura 01: ULA

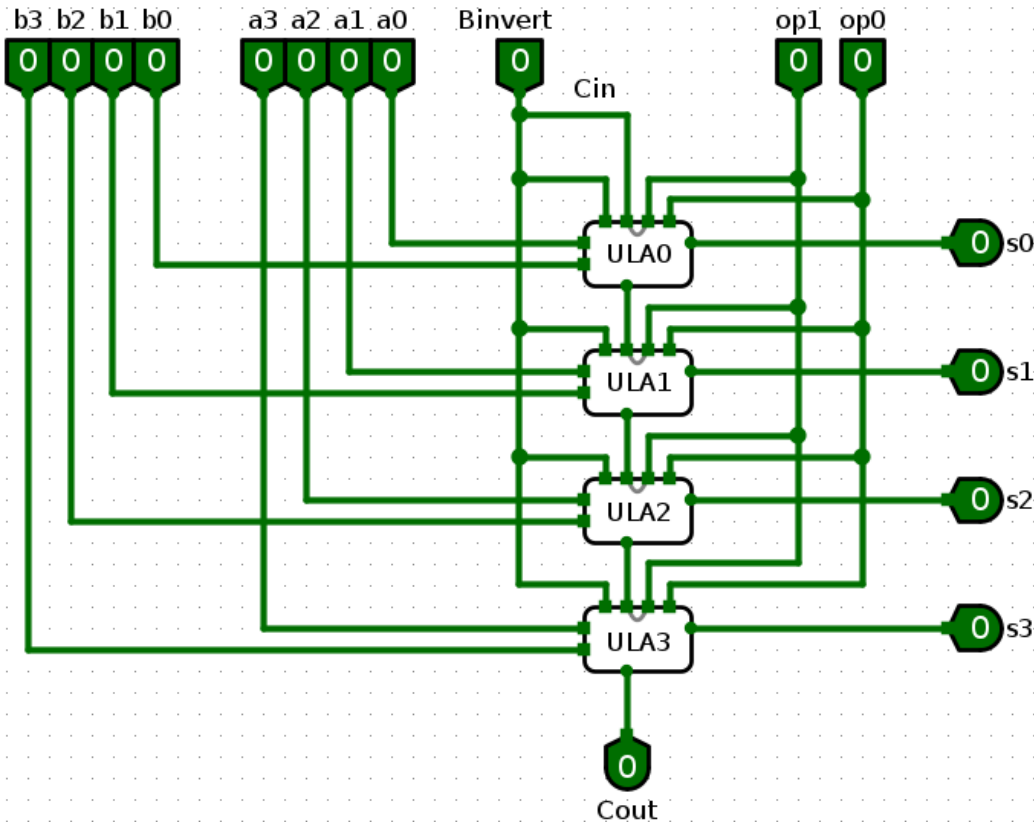
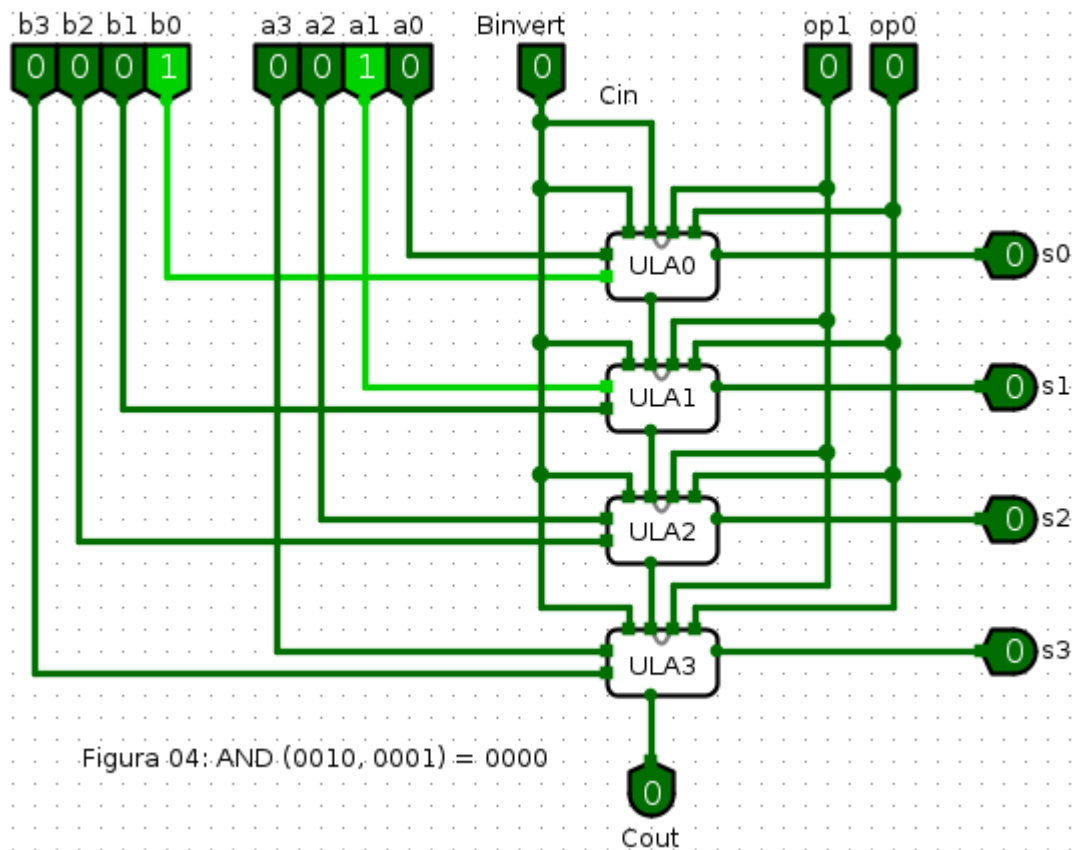
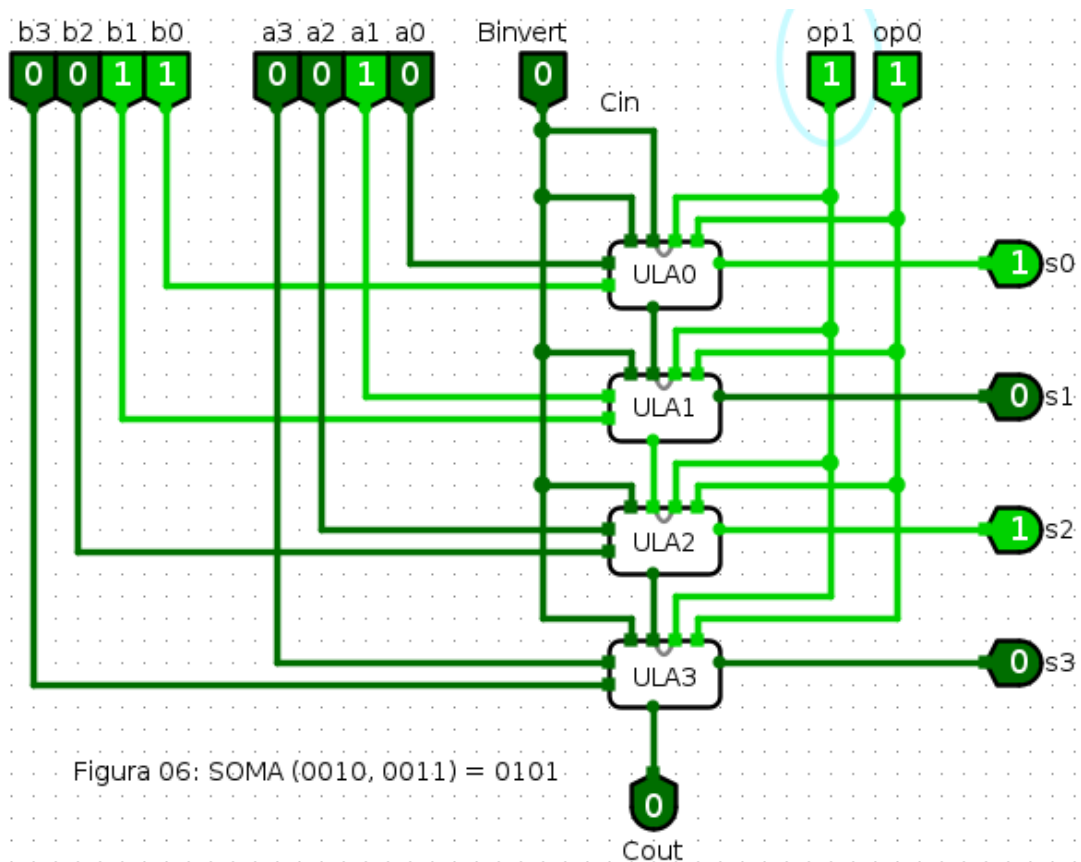
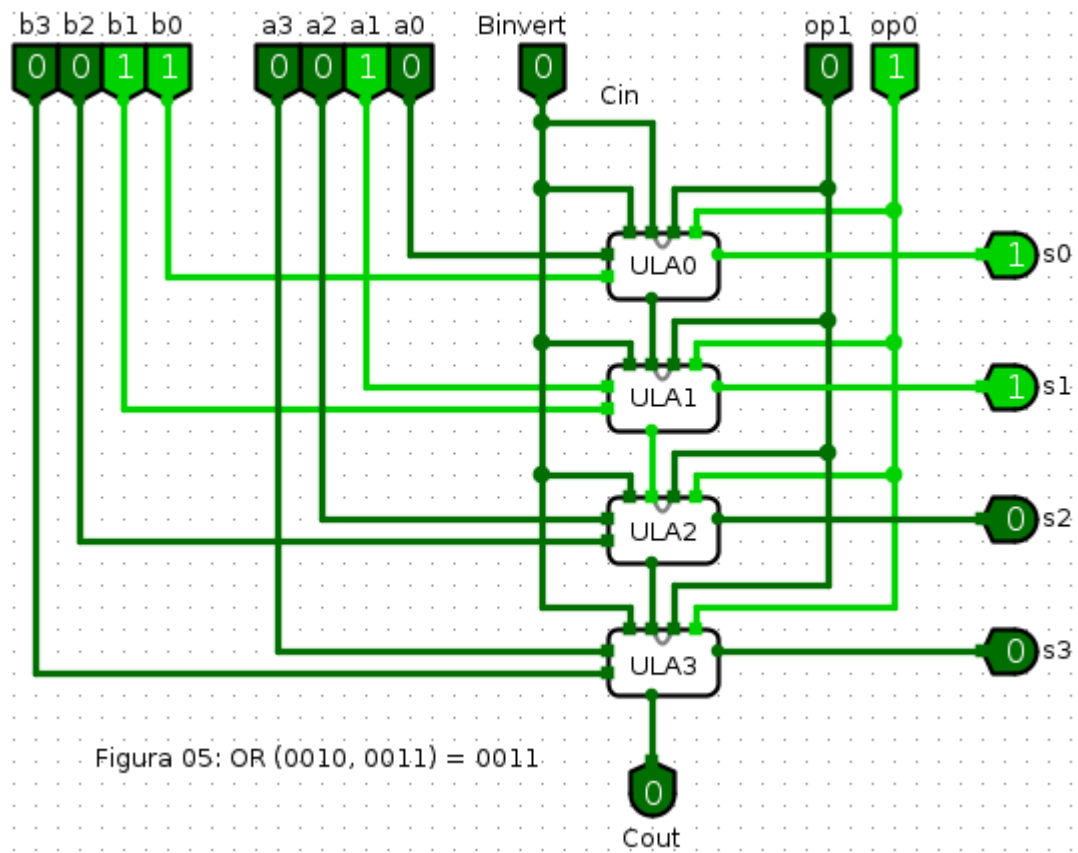


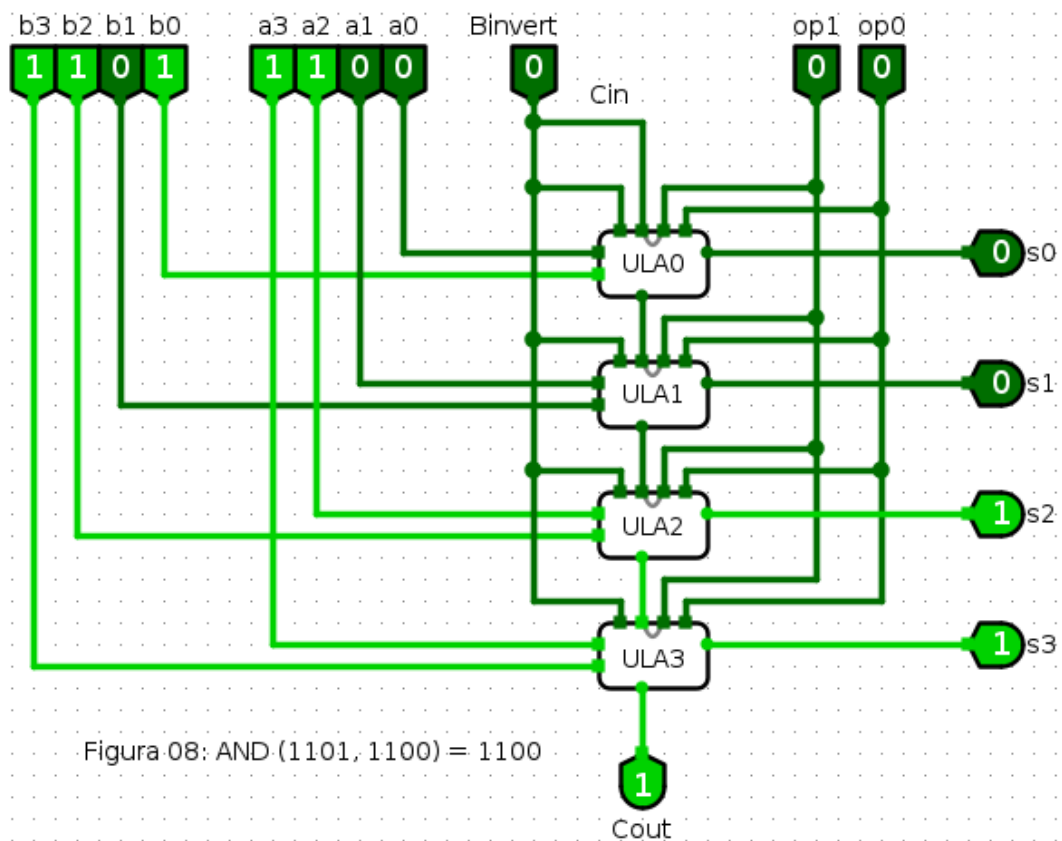
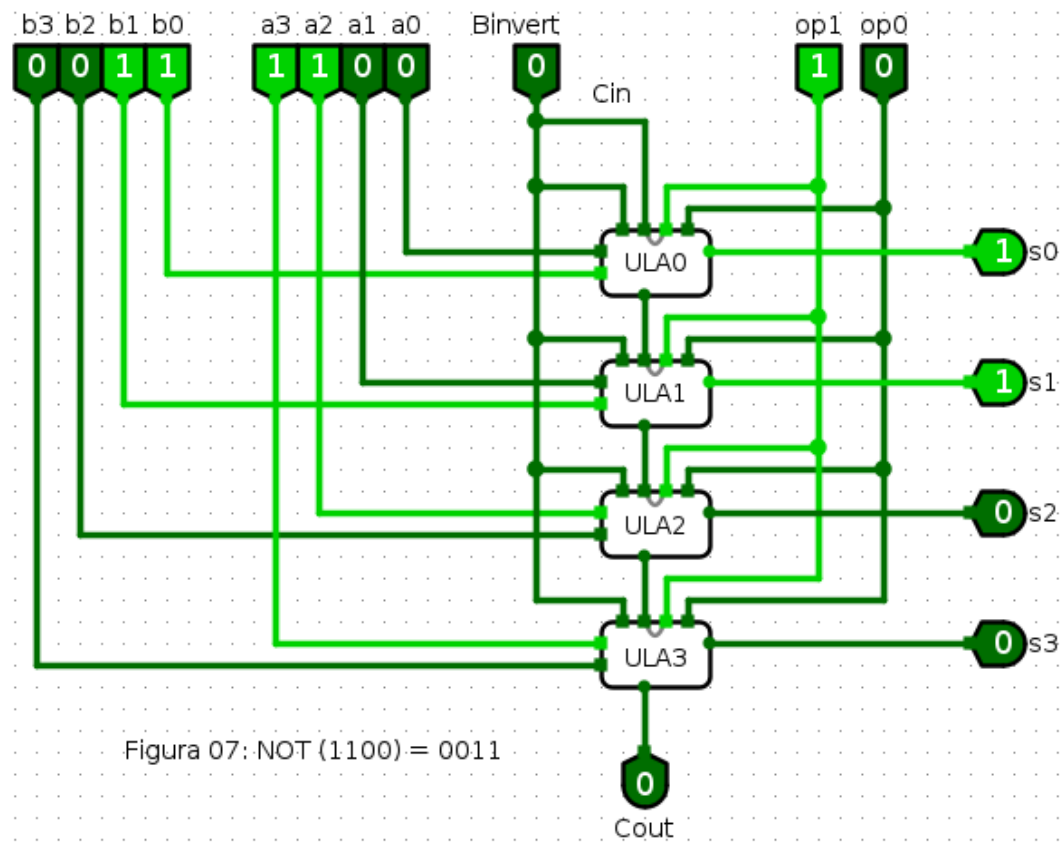
Figura 02: ULA para 4 bits

Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x ...)	Resultado em binário
AND(A,B)	0010 0001 00	(0000 1000 0100) = 0x084	0000
OR(A,B)	0010 0011 01	(0000 1000 1101) = 0x08D	0011
SOMA(A,B)	0010 0011 11	(0000 1000 1111) = 0x08F	0101
NOT(A)	1100 0011 10	(0011 0000 1110) = 0x30E	0011
AND(B,A)	1100 1101 00	(0011 0011 0100) = 0x334	1100

Figura 03: Tabela de resultados







Parte

2: 0

Arduíno

Exercício 01:

Link projeto: [Semaforo](#)

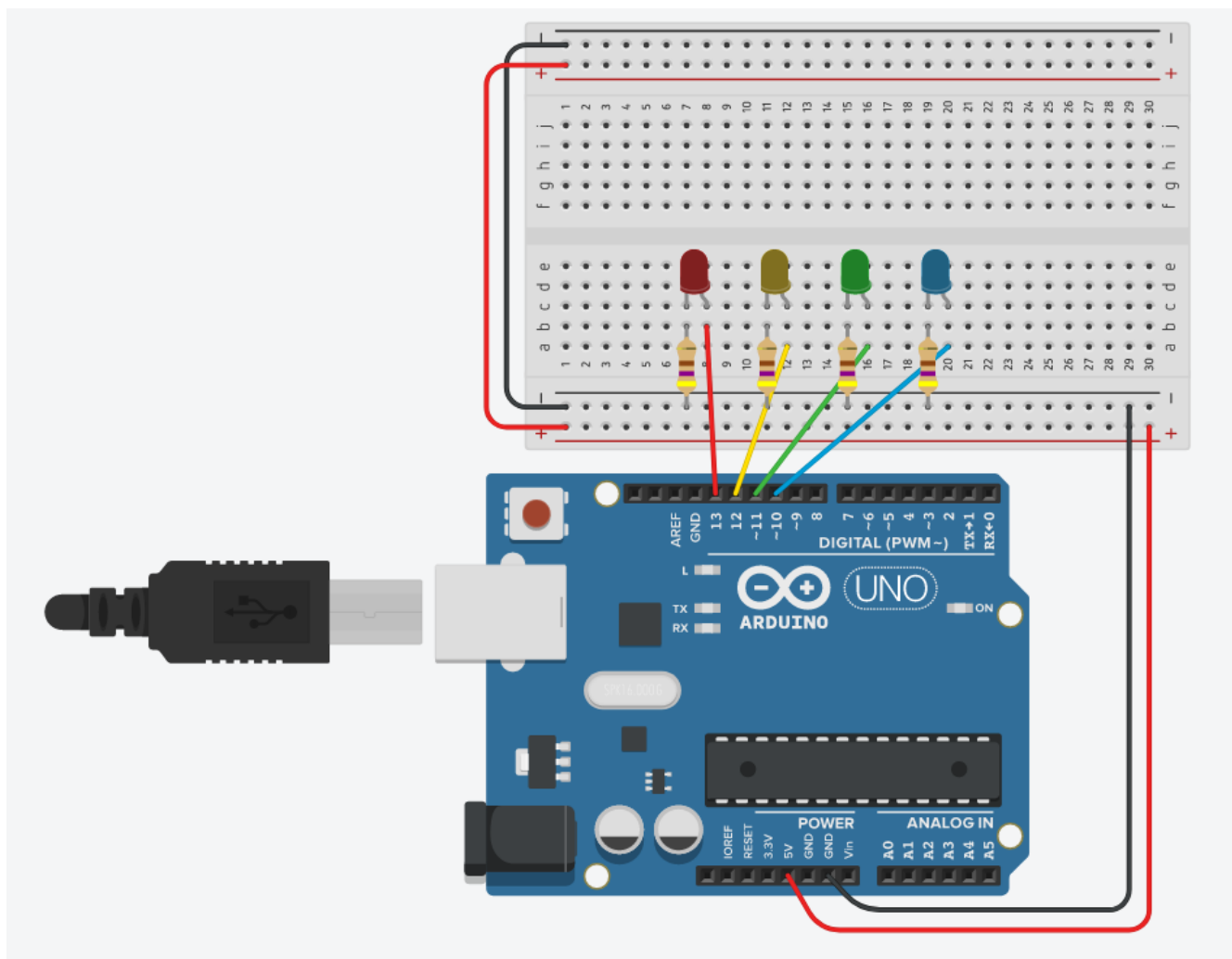


Figura 09: Montagem tinkercad

```
1  int azul = 10;
2  int verde = 11;
3  int amarelo = 12;
4  int vermelho = 13;
5
6  int tempoVermelho = 3;
7  int tempoVerde = 4;
8  int tempoAmarelo = 2;
9
10 int tempoPiscada = 1000; // 1 segundo
11
12 void setup() {
13     Serial.begin (9600);
14     pinMode (azul, OUTPUT);
15     pinMode (verde, OUTPUT);
16     pinMode (amarelo, OUTPUT);
17     pinMode (vermelho, OUTPUT);
18 }
19
```

Figura 10: Código programa [1/2] – Definição de variáveis e setup

```

20 void loop() {
21
22     // Ligar led vermelho
23     digitalWrite (vermelho, HIGH);
24
25     // Piscar led azul com vermelho
26     for(int i = 0; i < tempoVermelho; i++) {
27         digitalWrite (azul, HIGH);
28         delay (tempoPiscada/2);
29         digitalWrite (azul, LOW);
30         delay (tempoPiscada/2);
31     }
32
33     // Desligar vermelho
34     digitalWrite (vermelho, LOW);
35
36     // Ligar verde
37     digitalWrite (verde, HIGH);
38
39     // Piscar led azul com verde
40     for(int i = 0; i < tempoVerde; i++) {
41         digitalWrite (azul, HIGH);
42         delay (tempoPiscada/2);
43         digitalWrite (azul, LOW);
44         delay (tempoPiscada/2);
45     }
46
47     // Desligar verde
48     digitalWrite (verde, LOW);
49
50     // Ligar amarelo
51     digitalWrite (amarelo, HIGH);
52
53     // Piscar led azul com amarelo
54     for(int i = 0; i < tempoAmarelo; i++) {
55         digitalWrite (azul, HIGH);
56         delay (tempoPiscada/2);
57         digitalWrite (azul, LOW);
58         delay (tempoPiscada/2);
59     }
60
61     // Desligar amarelo
62     digitalWrite (amarelo, LOW);
63 }
64

```

Figura 11: Código programa [2/2] – Loop (“main”)

Exercício 02:

Link projeto: [ULA](#)

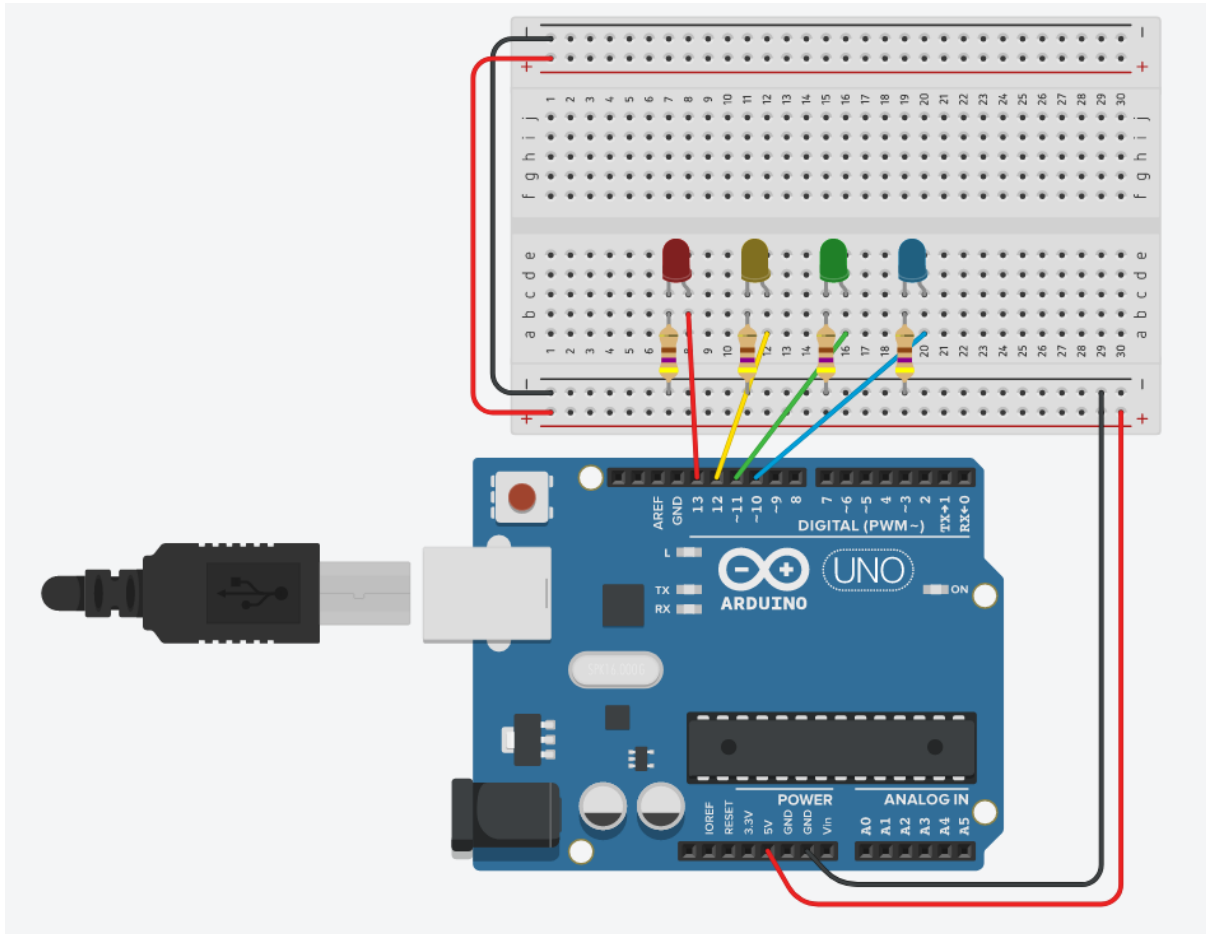


Figura 12: Montagem tinkercad

```
1 int azul = 10;    // valor de a
2 int verde = 11;   // valor de b
3 int amarelo = 12; // resultado ULA
4 int vermelho = 13; // carry out
5
6 int a = -1;
7 int b = -1;
8 int op = -1;
9
10 char entrada = ' ';
11 int tmp[3] = {-1, -1, -1};
12
```

Figura 13: Código programa [1/5] – Declaração variáveis

```

13 void setup () {
14     Serial.begin (9600);
15     pinMode (azul, OUTPUT);
16     pinMode (verde, OUTPUT);
17     pinMode (amarelo, OUTPUT);
18     pinMode (vermelho, OUTPUT);
19 }
20
21 void loop () {
22
23     if (Serial.available() > 0) {
24
25         for(int i = 0; i < 3; i++) {
26             entrada = Serial.read();
27             tmp[i] = entrada - 48;
28             delay(3);
29         }
30
31         a = tmp[0];
32         b = tmp[1];
33         op = tmp[2];
34
35         if (isValido()) {
36
37             digitalWrite (vermelho, a);
38             digitalWrite (amarelo, b);
39
40             switch (op) {
41                 case 0: portaAnd (); break;
42                 case 1: portaOr (); break;
43                 case 2: portaNot (); break;
44                 case 3: somador (); break;
45             }
46         } else {
47             mostrarErro();
48         }
49
50         delay (2000);
51         reset();
52     }
53 }
54 }

```

Figura 14: Código programa [2/5] – setup e loop (“main”)

```

56 bool isValido () {
57     return ((a == 0 || a == 1) &&
58             (b == 0 || b == 1) &&
59             (op >= 0 && op <= 3));
60 }
61
62 void portaAnd () {
63     Serial.print("AND  ");
64     mostrar(a&b);
65     digitalWrite (verde, a&b);
66 }
67
68 void portaOr () {
69     Serial.print("OR   ");
70     mostrar(a|b);
71     digitalWrite (verde, a|b);
72 }
73
74 void portaNot () {
75     Serial.print("NOT  ");
76     mostrar(~a&1);
77     digitalWrite (verde, (~a&1));
78 }
79
80 void somador () {
81     Serial.print("SOMA ");
82     mostrar(a^b, a&b);
83     digitalWrite (verde, a^b);
84     digitalWrite (azul, a&b);
85 }
86

```

Figura 15: Código programa [3/5] – Teste e métodos da ULA

```

87 void mostrarErro () {
88     Serial.println("ERRO: entrada invalida!");
89     for(int i = 0; i < 3; i++) {
90         digitalWrite (azul, HIGH);
91         digitalWrite (verde, HIGH);
92         digitalWrite (amarelo, HIGH);
93         digitalWrite (vermelho, HIGH);
94         delay (500);
95
96         digitalWrite (azul, LOW);
97         digitalWrite (verde, LOW);
98         digitalWrite (amarelo, LOW);
99         digitalWrite (vermelho, LOW);
100        delay (500);
101    }
102 }
103
104 void mostrar (int value) {
105     Serial.print(a);
106     Serial.print(",");
107     Serial.print(b);
108     Serial.print(" = ");
109     Serial.println(value);
110 }
111
112 void mostrar (int value0, int value1) {
113     Serial.print(a);
114     Serial.print(",");
115     Serial.print(b);
116     Serial.print(" = ");
117     Serial.print(value1);
118     Serial.println(value0);
119 }

```

Figura 16: Código programa [4/5] – Mensagens de erro e de sucesso

```

121 void reset () {
122     digitalWrite (azul, LOW);
123     digitalWrite (verde, LOW);
124     digitalWrite (amarelo, LOW);
125     digitalWrite (vermelho, LOW);
126
127     a = -1;
128     b = -1;
129     op = -1;
130
131     entrada = ' ';
132     for(int i = 0; i < 3; i++) tmp[i] = -1;
133 }

```

Figura 17: Código programa [5/5] – Resetar configurações iniciais

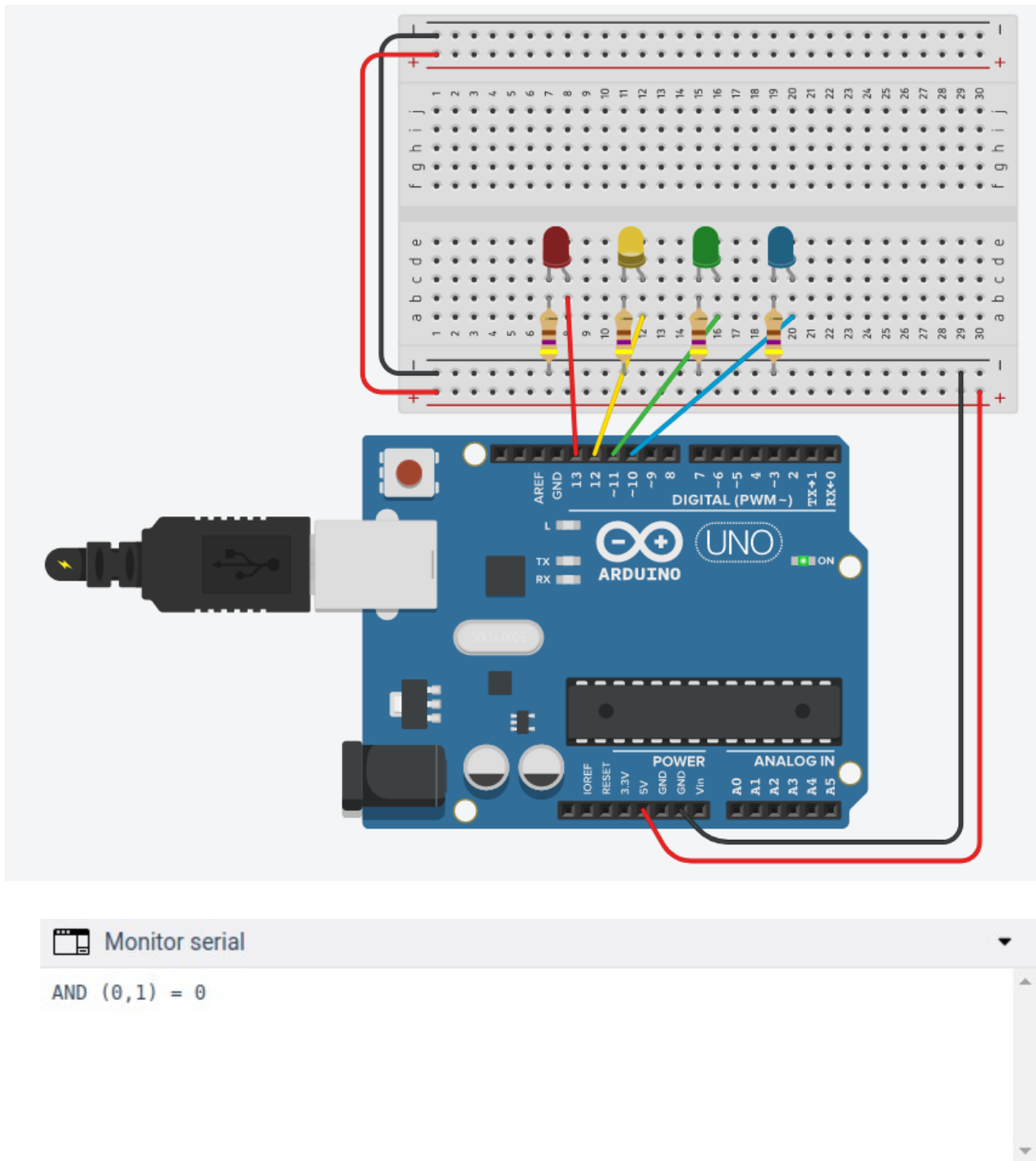


Figura 18: Teste AND (0, 1)

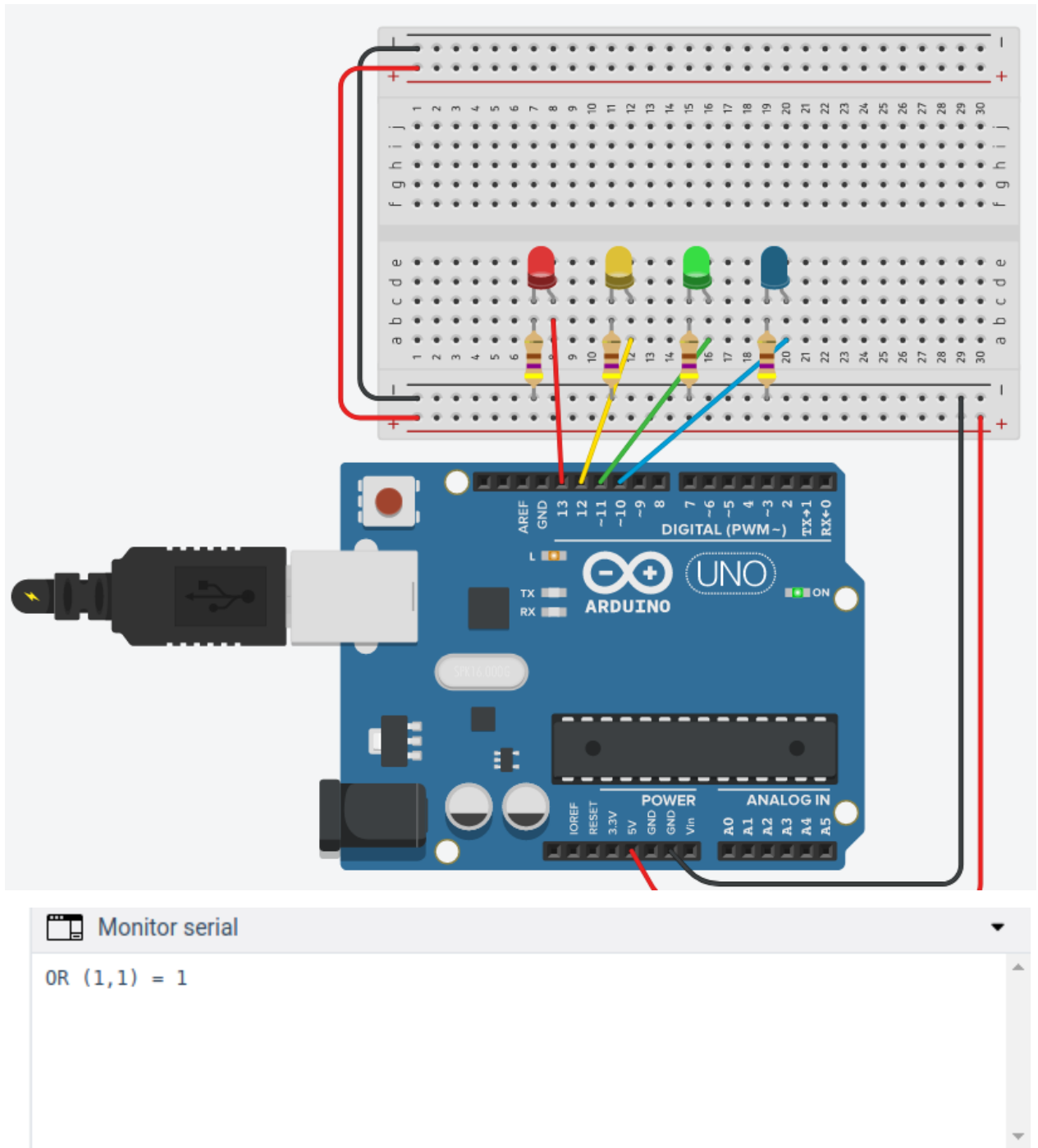


Figura 19: Teste OR (1, 1)

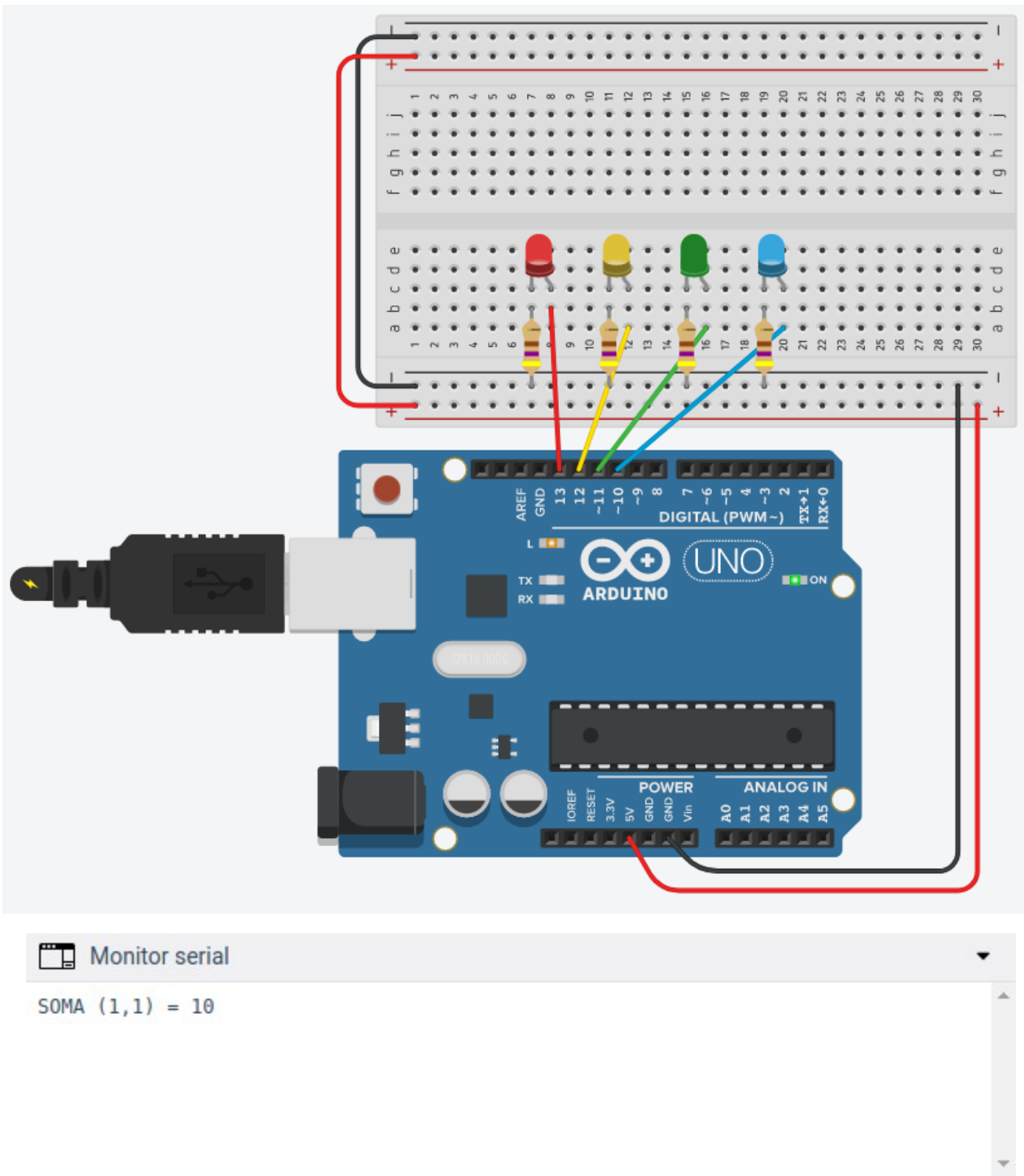


Figura 20: Teste SOMA (1, 1)

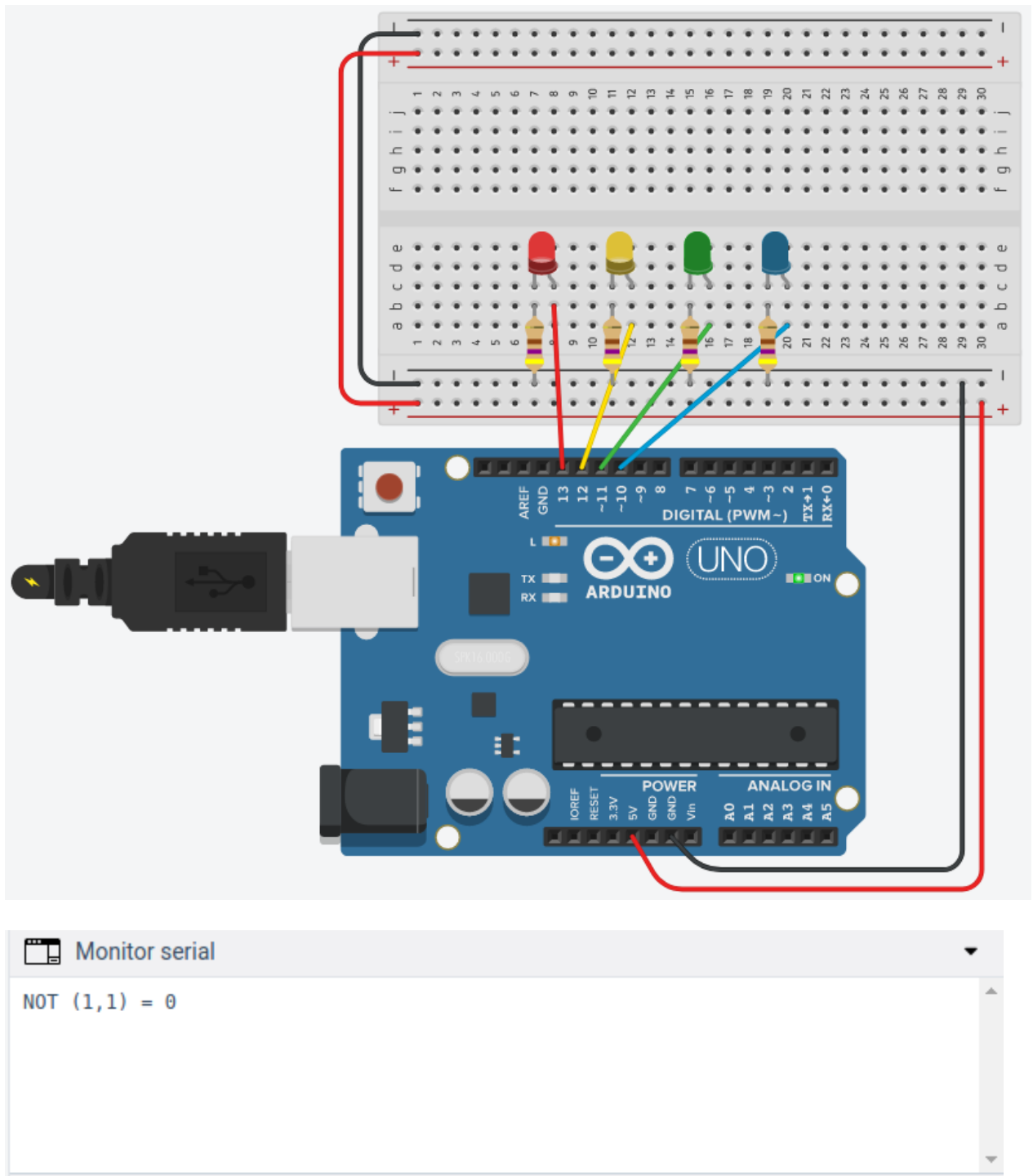


Figura 21: Teste NOT (1, 1)