

Trabalho Prático 1

**Computação Física**

**(C.F.)**

Curso de Licenciatura Informática e Multimédia (LEIM)

Ano Letivo 2017/2018

**Turma LEIM 24D**

**Docentes:**

Eng. Jorge Pais

Eng. Carlos Carvalho

**Grupo:** 1

**Membros:**

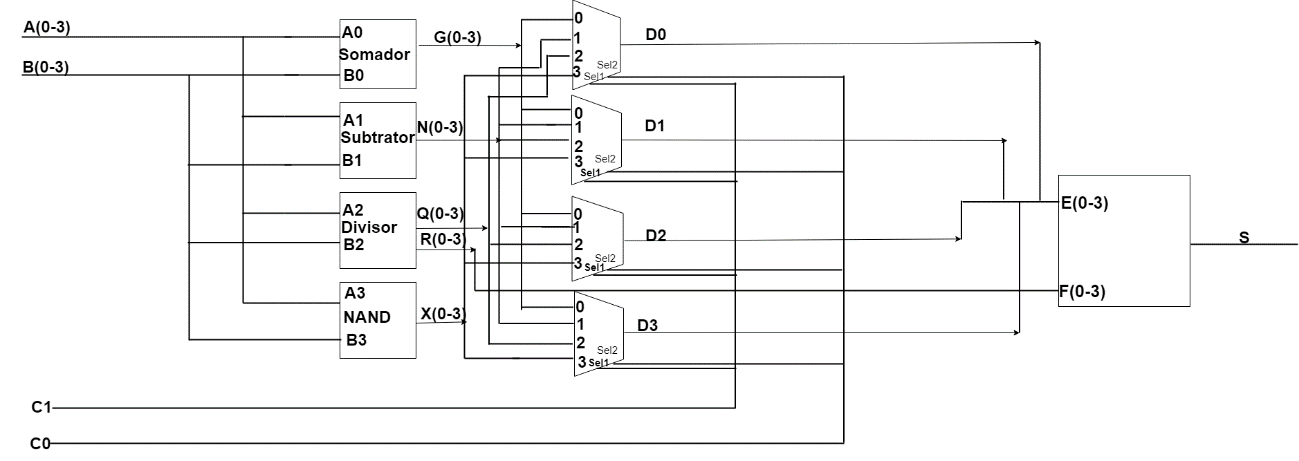
Luis Fonseca (A45125)

Philipp Al-Badavi(A45138)

**Introdução**

Para este trabalho foi pedido a realização e a representação de circuitos combinatórios e sequencias com as operações/operadores lógicas(os) apresentados no enunciado que são: soma, subtração, divisão e NAND. Neste relatório consta os processos que foram usados, assim como a explicação do procedimento adotado e uma respetiva conclusão.

**ALU(arithmetic logic unit)**



**Somador**

Com as entradas fornecidas, coube nos fazer uma tabela de verdade, a partir dai fazer um mapa de karnaugh, retirar as expressões para a implementação no arduino e desenhar o respetivo circuito. Encontra se os processos usados.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S | CyOut |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Tabela de Verdade: Mapa de Karnaugh:

|  |  |
| --- | --- |
| 0 | 1 |
| 1 | 0 |

S A

B

|  |  |
| --- | --- |
| 0 | 0 |
| 0 | 1 |

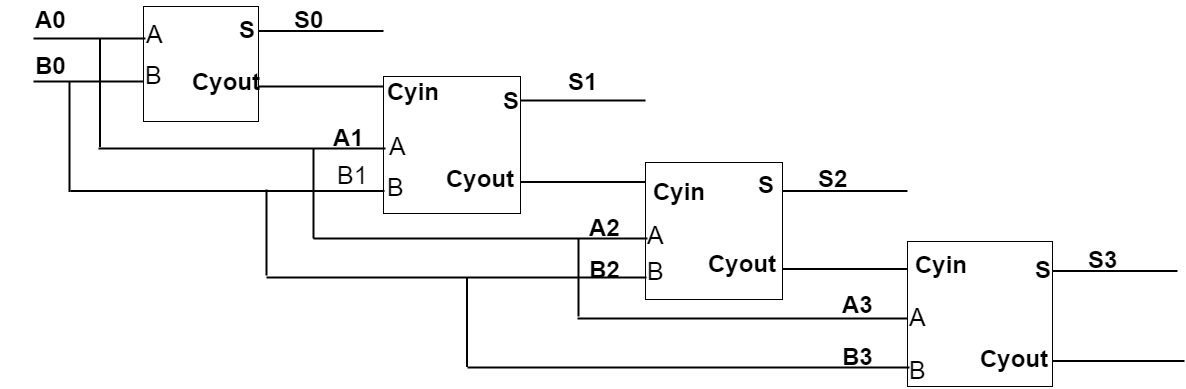
CyOut A

Expressões:

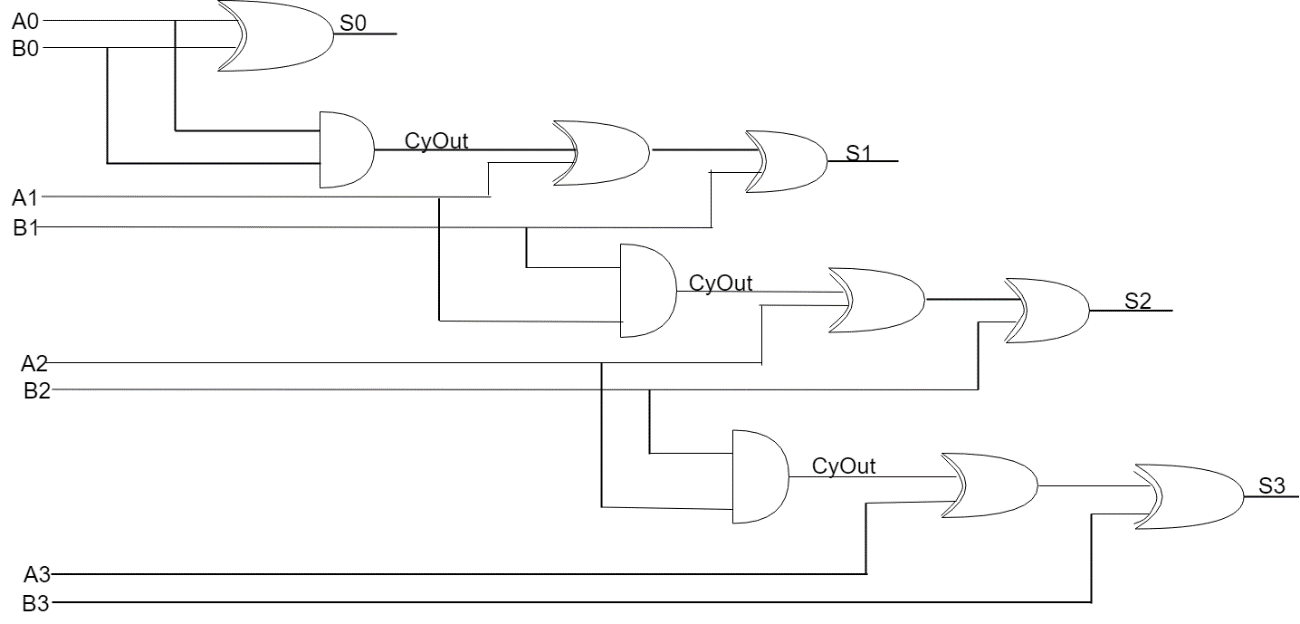
S = A/ . B + A.B/

CyOut = A.B.CyIn B

Modulo:



Circuito:



**Subtrator**

Com as entradas fornecidas, coube nos fazer uma tabela de verdade, a partir dai fazer um mapa de karnaugh, retirar as expressões para a implementação no arduino e desenhar o respetivo circuito. Encontra se os processos usados.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | S | Cyout |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

Tabela de Verdade: Mapas de Karnaugh:

|  |  |
| --- | --- |
| 0 | 0 |
| 1 | 0 |

Cyout A S A

|  |  |
| --- | --- |
| 1 | 0 |
| 1 | 0 |

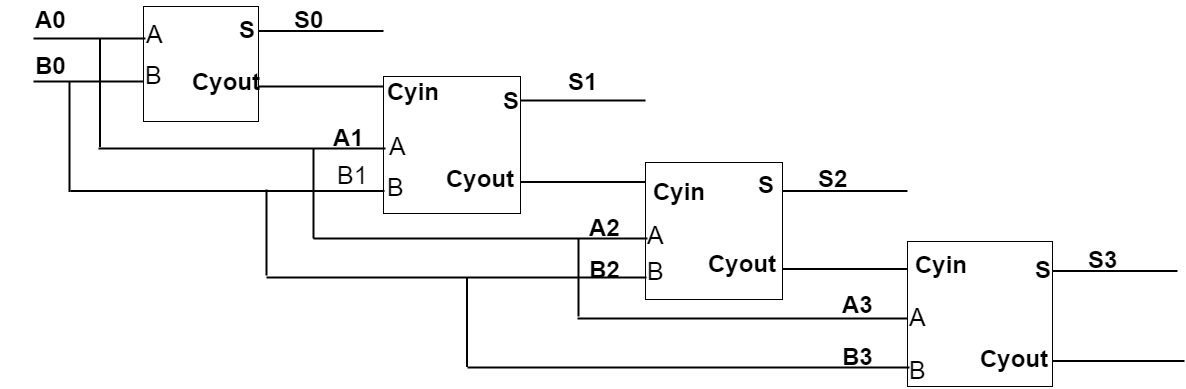
B B

Expressoes:

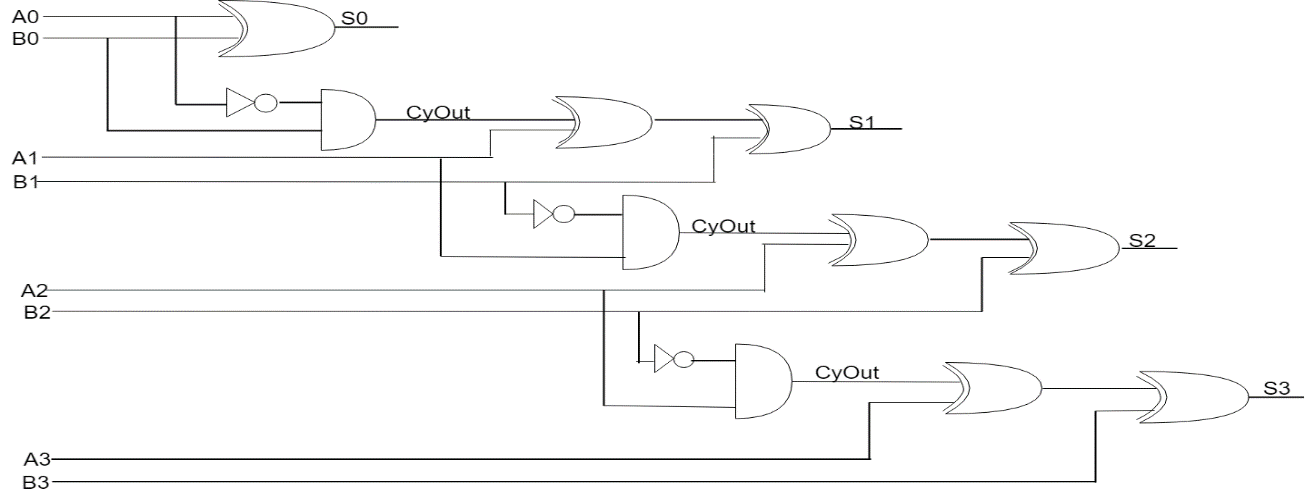
S = A/

CyOut = A/ . B/

Modulo:



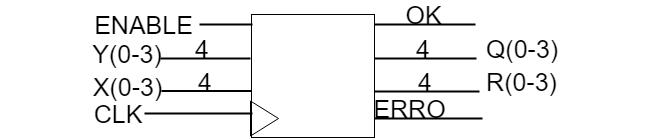
Circuito:



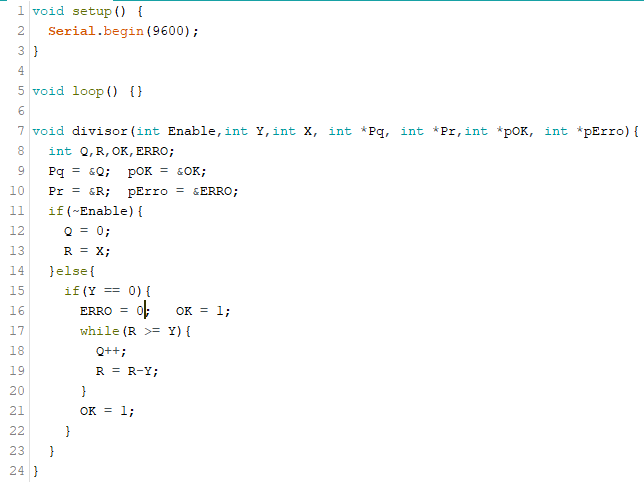
**Divisor**

Para esta operação logica o processo usado foi diferente dos que foi usado nas outras operações. Neste foram definidos vários passos para chegar ao objetivo pretendido, mas usando tabelas de verdade e mapas de Karnaugh.

1. **Definir as entradas e as saídas**



1. **Algoritmo da divisão**



1. **Tipos de Hardware**

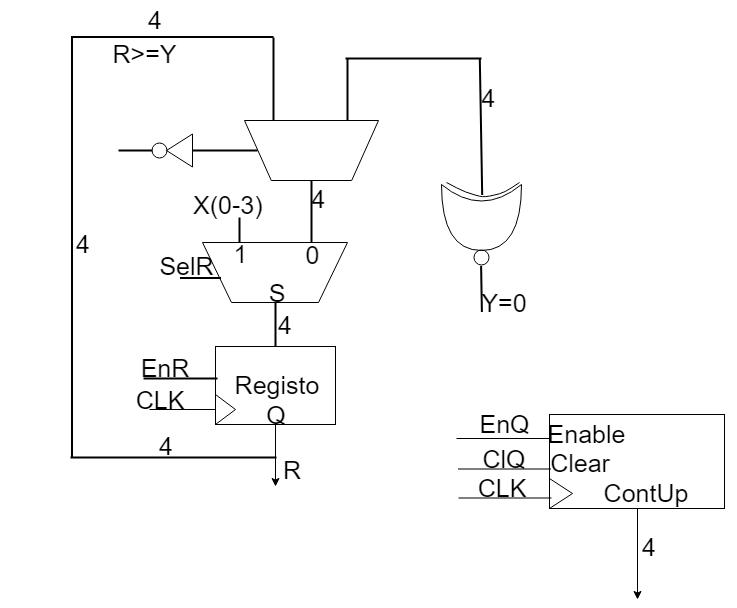
Q –ContadorUp (4bits)

R – Registo (4bits)

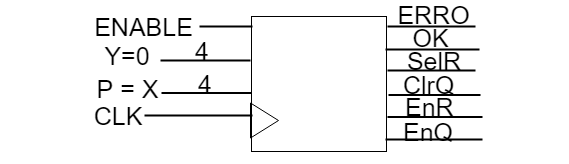
Erro,OK,Enable – Sinais de 1bit

X,Y – Entradas a 4 bits

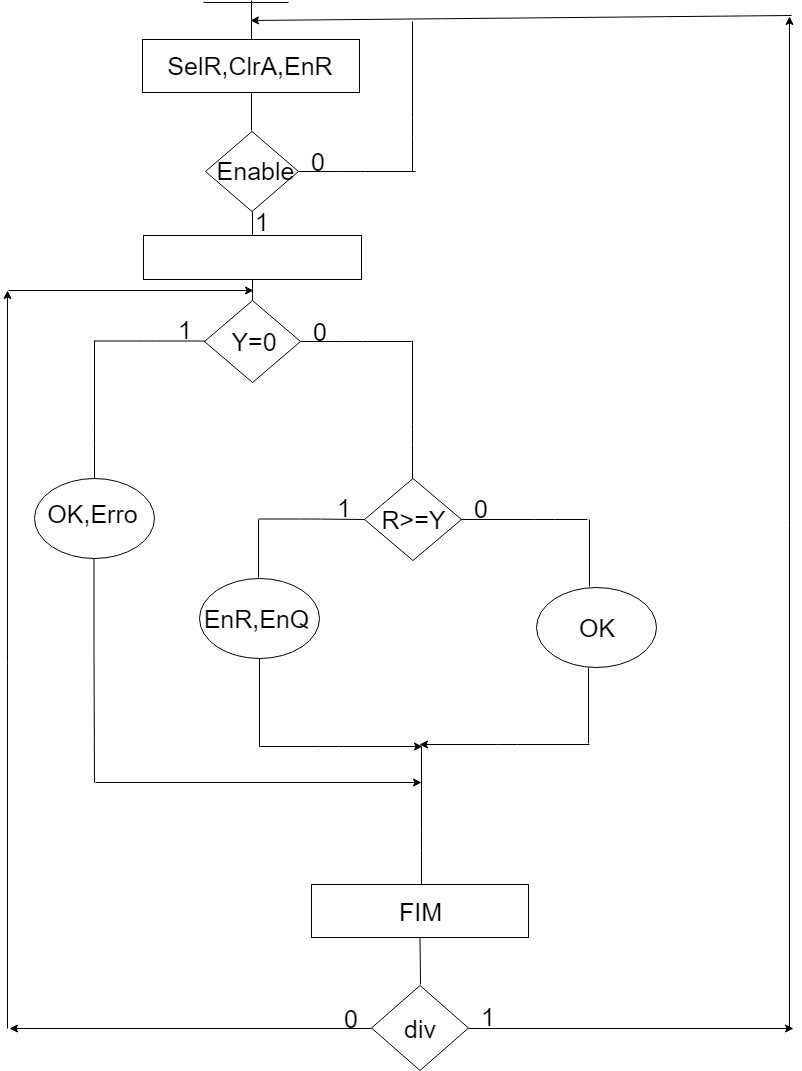
1. **Diagrama de blocos do modulo funcional**



1. **Especificar as entradas e saídas do modulo de controlo**



1. **ASM – Modulo de Controlo**

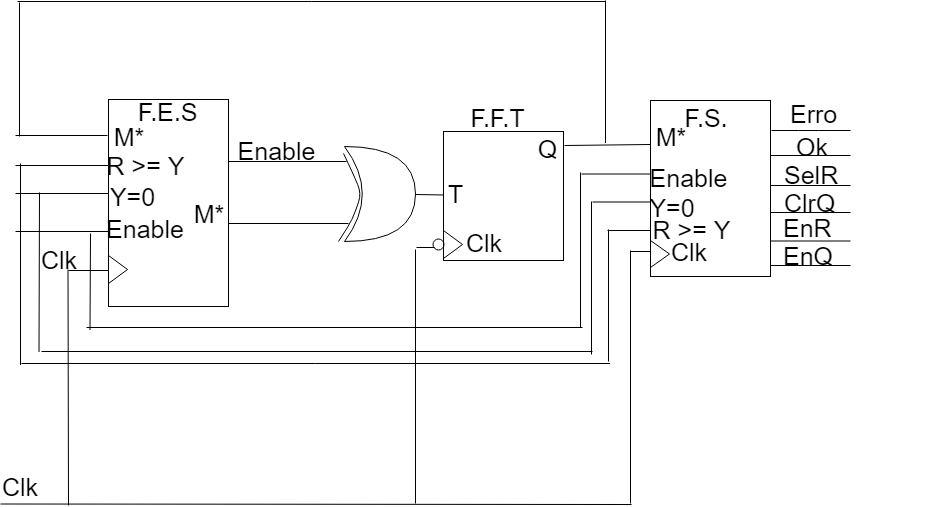


1. **Escolher o tipo de célula de Memoria**

R: Flip-Flop-Edge Triggered T

|  |  |
| --- | --- |
| T | Q |
| 0 | Q\* |
| 1 | Q\*/ |

1. **Esquema Mealy-Moore**



**9.Tabela de Verdade e Mapa de Karnaugh**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Enable | Y = 0 | R >=Y | M\* | M | Q |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

Y=0

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

T

M\*

Enable

R >=Y

T = Enable  M\*

**NAND**

Com as entradas fornecidas, coube nos fazer uma tabela de verdade, a partir dai fazer um mapa de karnaugh, retirar as expressões para a implementação no Arduíno e desenhar o respetivo circuito. Encontra se os processos usados. Esta expressão consiste na negação da operação lógica AND e como é possível observar na tabela de verdade, NAND só toma o valor 1 quando AND for igual a 0.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | A.B | (A.B)/ |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Tabela de Verdade: Mapas de Karnaugh:

|  |  |
| --- | --- |
| 1 | 0 |
| 0 | 0 |

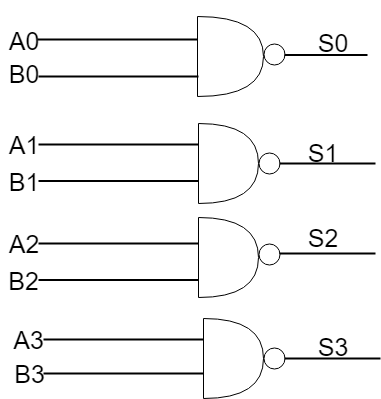
S A

B

Expressoes:

S = (A.B)/

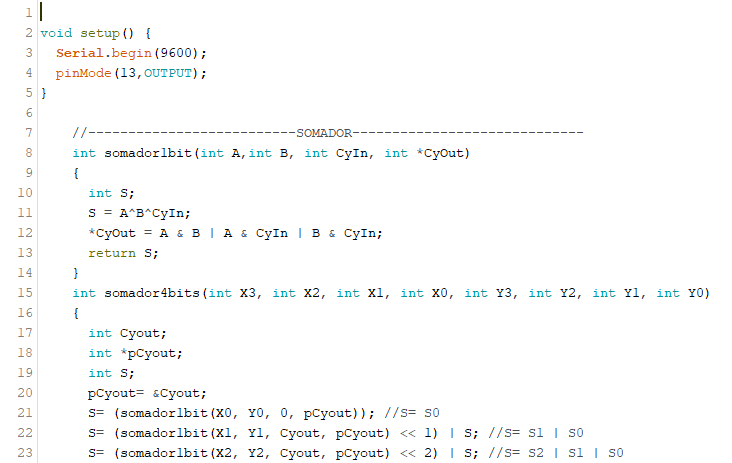
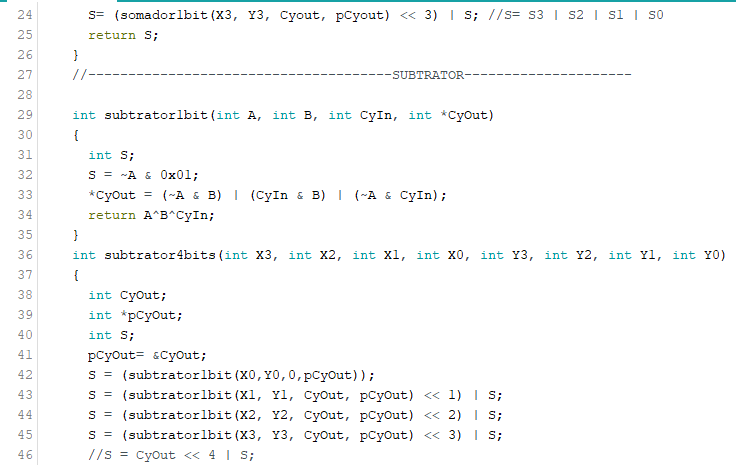
Circuito:

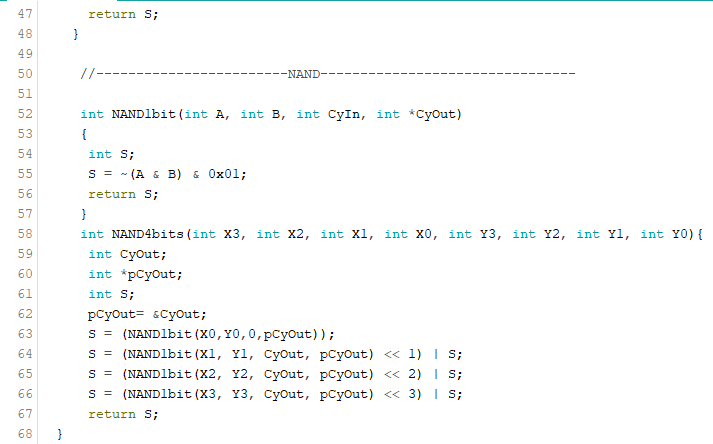


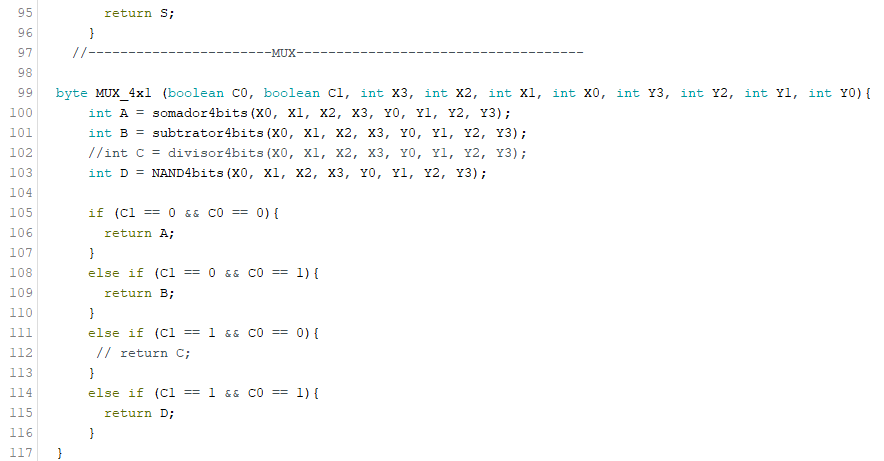
**No Arduíno…**

Com as expressões retiradas pelos processos mencionados, coube-nos correr no Arduíno para verificar os resultados. Através das varias funções conseguimos obter os resultados que esperávamos. Também foi feito um multiplexe de 4x1(4 entradas, 2 entradas de seleção e 1 saída) caso o utilizador pretenda efetuar outras operações. O código e o respetivo output encontram se em baixo.

* Código Arduíno:

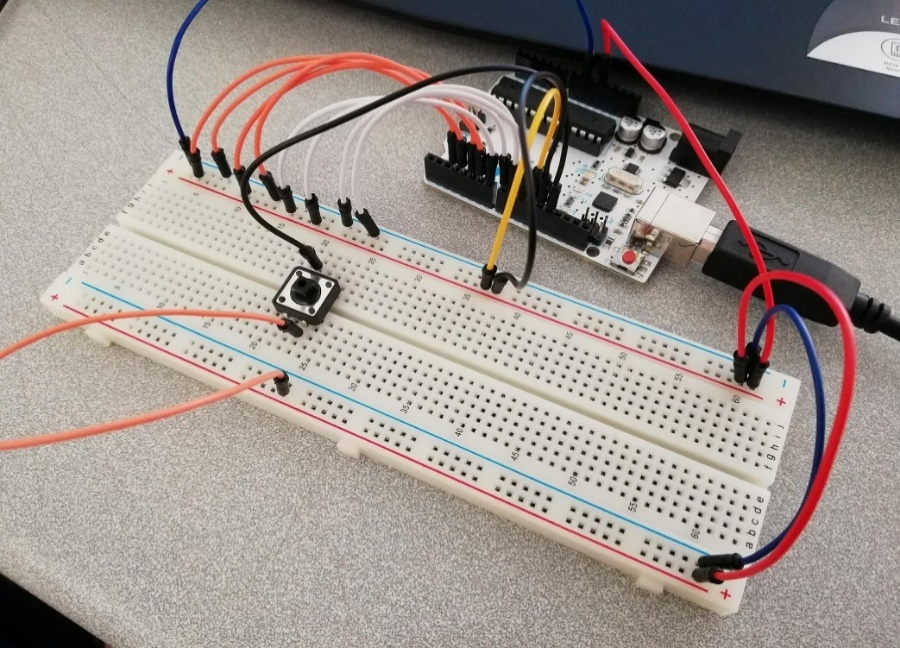


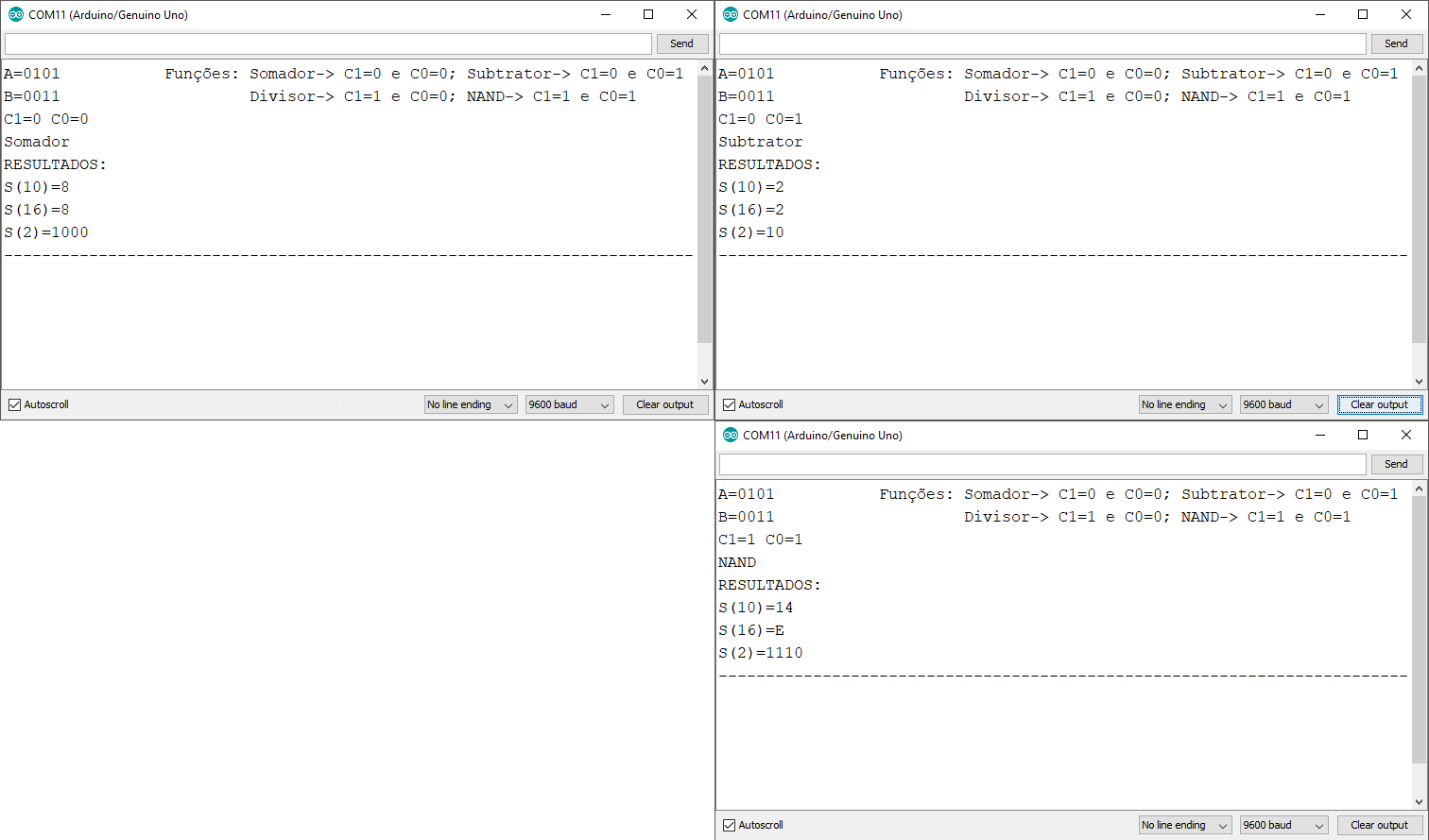








* Modulo do Utilizador:
* 
* Outputs:



**Conclusão**

Com este trabalho foi permitido uma profunda abordagem em relação a este tópico, como se atua e extrair as expressões através de vários processos.