

Estudos sobre circuitos combinacionais e sequenciais e implementação de dois circuitos digitais no Circuit Maker

Diogo Alves de Almeida RA: 95108¹

¹Centro de Tecnologia

Departamento de Informática – Universidade Estadual de Maringá (UEM)

CEP 87020-900 – Maringá – Paraná – Brasil

1. Introdução

Para a primeira parte do trabalho foi proposto a execução de uma Unidade Lógica e Aritmética (ULA) de 1 bit. Muitas das operações de um computador são feitas pela ULA, ela recebe os dados dos registradores, realiza as operações e devolve esses dados para os registradores de saída. Então, outras estruturas movem esses dados para a memória e outros registradores. A ULA deverá a partir de um circuito digital fazer operações lógicas e aritméticas. As operações propostas são: O objetivo do documento é a apresentação do desenvolvimento de dois projetos utilizando circuitos lógicos para que os alunos possam de adquirir conhecimento a partir de um experimento prático possibilitando o melhor entendimento da funcionalidade de algumas partes da arquitetura de um computador.

Lógicas:

AND

NAND

OR

NOR

XOR

XNOR

Aritméticas:

Adição Binária Completa

Subtração Binária Completa

Um sistema digital é capaz de armazenar uma grande quantidade de dados em um período de tempo curto ou longo, essa é uma das principais vantagens em relação aos sistemas analógicos pois é possível ser adaptada de várias maneiras, para nossa situação desejada. O flip flop é um circuito capaz de armazenar 1 bit, um conjunto de flip flops denominamos registradores que são capazes de armazenar informações estruturadas, composta por dados ou instruções e tais informações podem ser recebidas ou transferidas para outros dispositivos de armazenamento. [Junior 2011]

Com isso, proposta como segunda parte do trabalho a execução de uma memória semicondutora de duas palavras de 3 bits. Cada operação devera ler ou escreve uma

palavra completa de 3 bits. O projeto deverá ter 7 linhas de entrada e 3 linhas de saída. Na entrada serão: 3 linhas de entrada de dado, 1 linha de entrada de endereço e 3 linhas de entrada de controle para habilitar a pastilha (Chip Select), distinguir leitura de escrita e para habilitar a saída, e por fim, 3 linhas de saída de dado. Neste projeto, cada palavra de 3 bits necessitará de 3 flip-flops tipo D para o seu armazenamento.

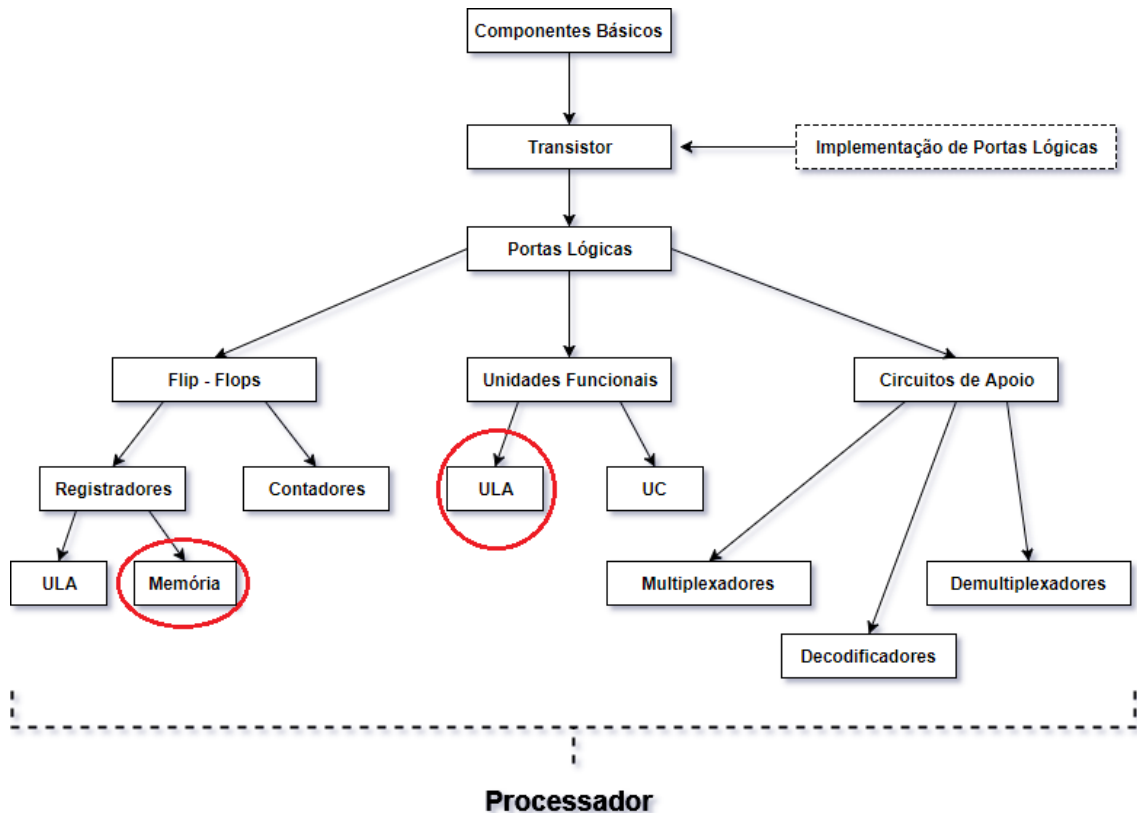


Figura 1. Representação de um processador.

2. Componentes utilizados

Nesta seção vamos apresentar os componentes utilizados no projeto, as portas lógicas que serão necessárias, suas tabelas verdades, sua simbologia e sua expressão booleana como vimos durante as aulas. Os dispositivos representativos terão como valores de entrada e saída os bits 1 ou 0 que são a representação em valor lógico de verdadeiro e falso, respectivamente.

2.1. Logic Switch



Figura 2. Logic switch desligado (bit = 0).



Figura 3. Logic switch ligado (bit = 1).

2.2. Logic Display



Figura 4. Logic display desligado (bit = 0).



Figura 5. Logic display Ligado (bit = 1).

2.3. NOT

Função NOT Representação: $S = \overline{A}$

Símbolo da Porta NOT

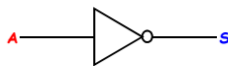


Figura 6. Simbologia e álgebra de boole da porta NOT.

A	S
0	1
1	0

Tabela 1. Tabela verdade para a porta NOT.

2.4. AND

Função AND Representação: $S = A \cdot B$

Símbolo da Porta AND



Figura 7. Simbologia e álgebra de boole da porta AND.

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 2. Tabela verdade para a porta AND.

2.5. NAND

Função NAND Representação: $S = \overline{A \cdot B}$

Símbolo da Porta NAND



Figura 8. Simbologia e álgebra de boole da porta NAND.

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Tabela 3. Tabela verdade para a porta NAND.

2.6. OR

Função OR Representação: $S = A + B$

Símbolo da Porta OR



Figura 9. Simbologia e álgebra de boole da porta OR.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 4. Tabela verdade para a porta OR.

2.7. NOR

Função NOR Representação: $S = \overline{A+B}$

Símbolo da Porta NOR



Figura 10. Simbologia e álgebra de boole da porta NOR.

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Tabela 5. Tabela verdade para a porta NOR.

2.8. XOR

Função XOR Representação: $S = A \oplus B$

Símbolo da Porta XOR



Figura 11. Simbologia e álgebra de boole da porta XOR.

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 6. Tabela verdade para a porta XOR.

2.9. XNOR

Função XNOR Representação: $S = \overline{A \oplus B} = A \odot B$

Símbolo da Porta XNOR



Figura 12. Simbologia e álgebra de boole da porta XNOR.

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Tabela 7. Tabela verdade para a porta XNOR.

Observações: As portas lógicas XOR e XNOR sempre possuem duas entradas e uma única saída.

2.10. Flip-Flop D

O Flip-Flop tipo D é obtido a partir do Flip-Flop JK com a porta K conectada à porta J e invertida, portanto, é impossível termos uma situação em que J e K sejam iguais (J = 0 e K = 0, J = 1 e K = 1).

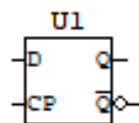


Figura 13. Simbologia do flip-flop D. (Circuit Maker)

D	Q
0	0
1	1

Tabela 8. Tabela verdade do flip-flop D.

2.11. Pulser

O pulser é um dispositivo que gera pulsos elétricos variando entre 0 e 1, será o dispositivo que iremos utilizar para a ativação dos flip-flops.

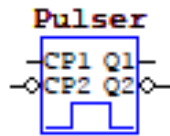


Figura 14. Simbologia do pulser. (Circuit Maker)

2.12. Buffer 3S E

É um buffer comum, porém com uma entrada de dados auxiliar, que permite a saída ou não de Dados. Será usado quando o Chip Select (CS) estiver desligado (para assim desligar a entrada e saída de dados), e quando a memória estiver no modo de leitura.

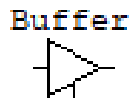


Figura 15. Simbologia buffer 3S E. (Circuit Maker)

3. Desenvolvimento

3.1. Projeto I : Unidade Lógica e Aritmética (ULA)

Como foi proposto, a ULA deverá fazer 8 operações (6 lógicas e 2 aritméticas), portanto há 8 possibilidades de escolha de operações. Utilizando a fórmula 2^n , onde n é o número de variáveis nós obtemos o número necessário para criarmos o codificador que irá selecionar as operações:

$$2^n = 8 \text{ operações} \quad (1)$$

$$2^3 = 8 \text{ operações} \quad (2)$$

$$n = 3 \text{ variáveis} \quad (3)$$

Após definirmos quantas variáveis são necessárias nós construímos a tabela verdade para o decodificador:

A	B	C	S
0	0	0	AND
0	0	1	NAND
0	1	0	OR
0	1	1	NOR
1	0	0	XOR
1	0	1	XNOR
1	1	0	SOM
1	1	1	SUB

Tabela 9. Tabela verdade para o decodificador.

Utilizando o método Soma de Produtos podemos tirar a expressão booleana de S:

$$S = \text{AND} + \text{NAND} + \text{OR} + \text{NOR} + \text{XOR} + \text{XNOR} + \text{SOM} + \text{SUB}$$

$$S = \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{A}.C + \overline{A}.B.\overline{C} + \overline{A}.B.C + A.\overline{B}.\overline{C} + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

A seguir o circuito montado a partir da expressão booleana que servirá como nosso decodificador:

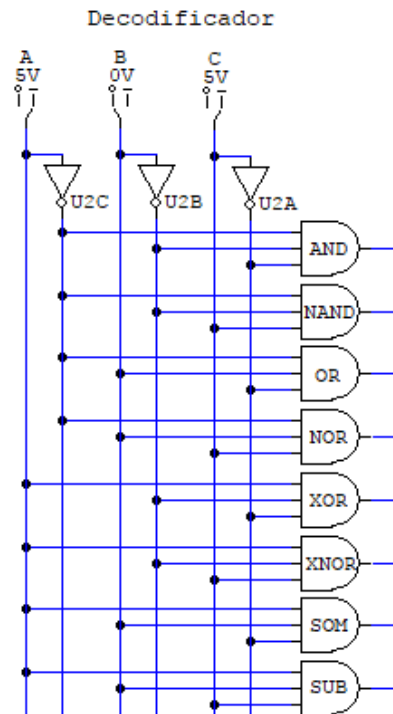


Figura 16. Circuito do decodificador. (CircuitMaker)

Após a produção do decodificador, foram conectadas as portas lógicas de suas respectivas saídas, tem-se a seguir o circuito das operações lógicas:

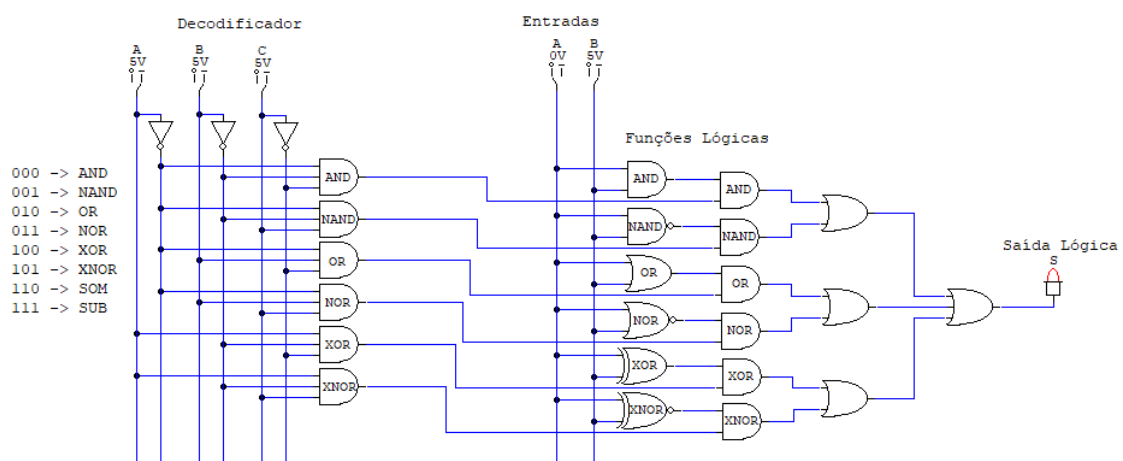


Figura 17. Operações lógicas conectadas ao decodificador e as entradas da ULA. (CircuitMaker)

Por fim, foi implementado o circuito somador e subtrator para finalizar a ULA com as operações lógicas e aritméticas completas. Para a execução do somador e do subtrator, foi utilizado o material de apoio das aulas dadas pelo professor Nardênio no qual explica como funciona um somador e um subtrator completo, suas tabelas verdades e a partir delas retirar os resultados para produzir o circuito lógico através do Mapa de Karnaugh.

Para este projeto foi utilizado o Karnaugh de 3 variáveis como veremos a seguir:

A	B	C	S
0	0	0	S0
0	0	1	S1
0	1	0	S2
0	1	1	S3
1	0	0	S4
1	0	1	S5
1	1	0	S6
1	1	1	S7

Tabela 10. Tabela verdade do mapa de karnaugh de 3 variáveis.

	\overline{B}	B
\overline{A}	S0	S1
A	S4	S5
	\overline{C}	C

Tabela 11. Mapa de Karnaugh de 3 variáveis.

A seguir as respectivas tabelas verdades e os mapas do Somador e do Subtrator:

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabela 12. Tabela verdade para o somador completo.

	\overline{B}	B
\overline{A}	0	1
A	1	0
	\overline{C}	C

Tabela 13. Mapa de karnaugh da saída S do somador completo.

	\overline{B}	B
\overline{A}	0	1
A	0	0
	\overline{C}	C

Tabela 14. Mapa de karnaugh da saída Cout do somador completo.

Retirando os dados do mapa podemos perceber que a expressão booleana de S é:

$$S = \overline{A}.\overline{B}.Cin + \overline{A}.B.\overline{Cin} + A.\overline{B}.\overline{Cin} + A.B.Cin$$

$$S = \overline{A} (\overline{B}.Cin + B.\overline{Cin}) + A (\overline{B}.\overline{Cin} + B.C)$$

$$\text{Como } B \oplus Cin = \overline{B}.Cin + B.\overline{Cin} \text{ e } B \odot Cin = \overline{B}.\overline{Cin} + B.Cin$$

$$S = \overline{A} (B \oplus Cin) + A (B \odot Cin)$$

$$\text{Fazendo } X = B \oplus Cin \text{ e } \overline{X} = B \odot Cin$$

$$S = \overline{A}.X + A.\overline{X}$$

$$S = A \oplus X$$

$$\mathbf{S = A \oplus B \oplus C}$$

Retirando os dados do mapa podemos perceber que a expressão booleana de Cout é:

$$\mathbf{Cout = A.B + B.Cin + A.Cin}$$

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Tabela 15. Tabela verdade para o subtrator.

	\overline{B}	B	
\overline{A}	0	1	0
A	1	0	1
	\overline{C}	C	\overline{C}

Tabela 16. Mapa de karnaugh da saída S do subtrator.

	\overline{B}	B	
\overline{A}	0	1	1
A	0	0	1
	\overline{C}	C	\overline{C}

Tabela 17. Mapa de karnaugh da saída Cout do subtrator.

Retirando os dados do mapa podemos perceber que a expressão booleana de S é:

$$S = \overline{A}.\overline{B}.Cin + \overline{A}.B.\overline{Cin} + A.\overline{B}.\overline{Cin} + A.B.Cin$$

$$S = \overline{A} (\overline{B}.Cin + B.\overline{Cin}) + A (\overline{B}.\overline{Cin} + B.C)$$

$$\text{Como } B \oplus Cin = \overline{B}.Cin + B.\overline{Cin} \text{ e } B \odot Cin = \overline{B}.\overline{Cin} + B.Cin$$

$$S = \overline{A} (B \oplus Cin) + A (B \odot Cin)$$

$$\text{Fazendo } X = B \oplus Cin \text{ e } \overline{X} = B \odot Cin$$

$$S = \overline{A}.X + A.\overline{X}$$

$$S = A \oplus X$$

$$\mathbf{S = A \oplus B \oplus C}$$

Retirando os dados do mapa podemos perceber que a expressão booleana de Cout é:

$$\mathbf{Cout = \overline{A}.B + B.Cin + \overline{A}.Cin}$$

Após retirar as expressões booleanas do Mapa de Karnaugh conseguimos chegar nos circuitos lógicos desejados:

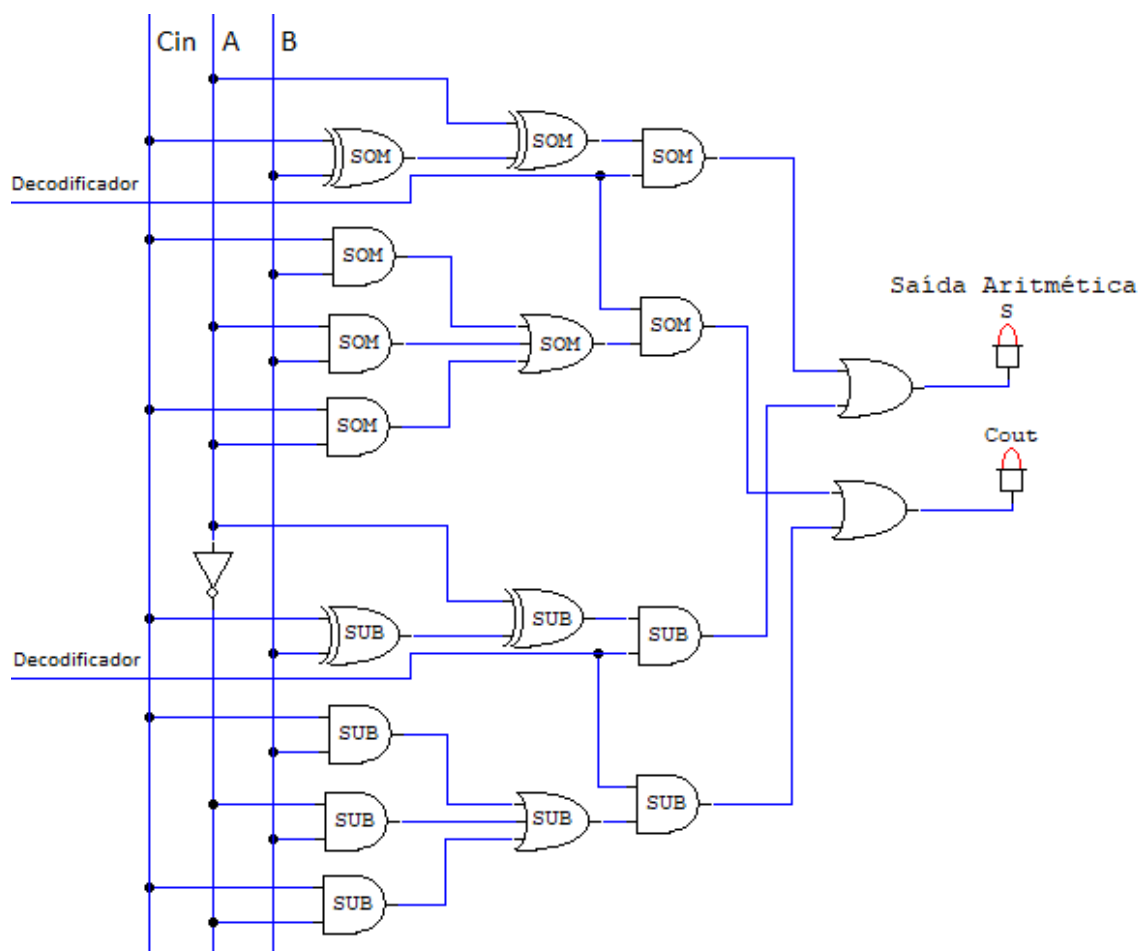


Figura 18. Circuito do somador e subtrator. (CircuitMaker)

A seguir o circuito da Unidade Lógica e Aritmética (ULA) de 1 bit finalizada:

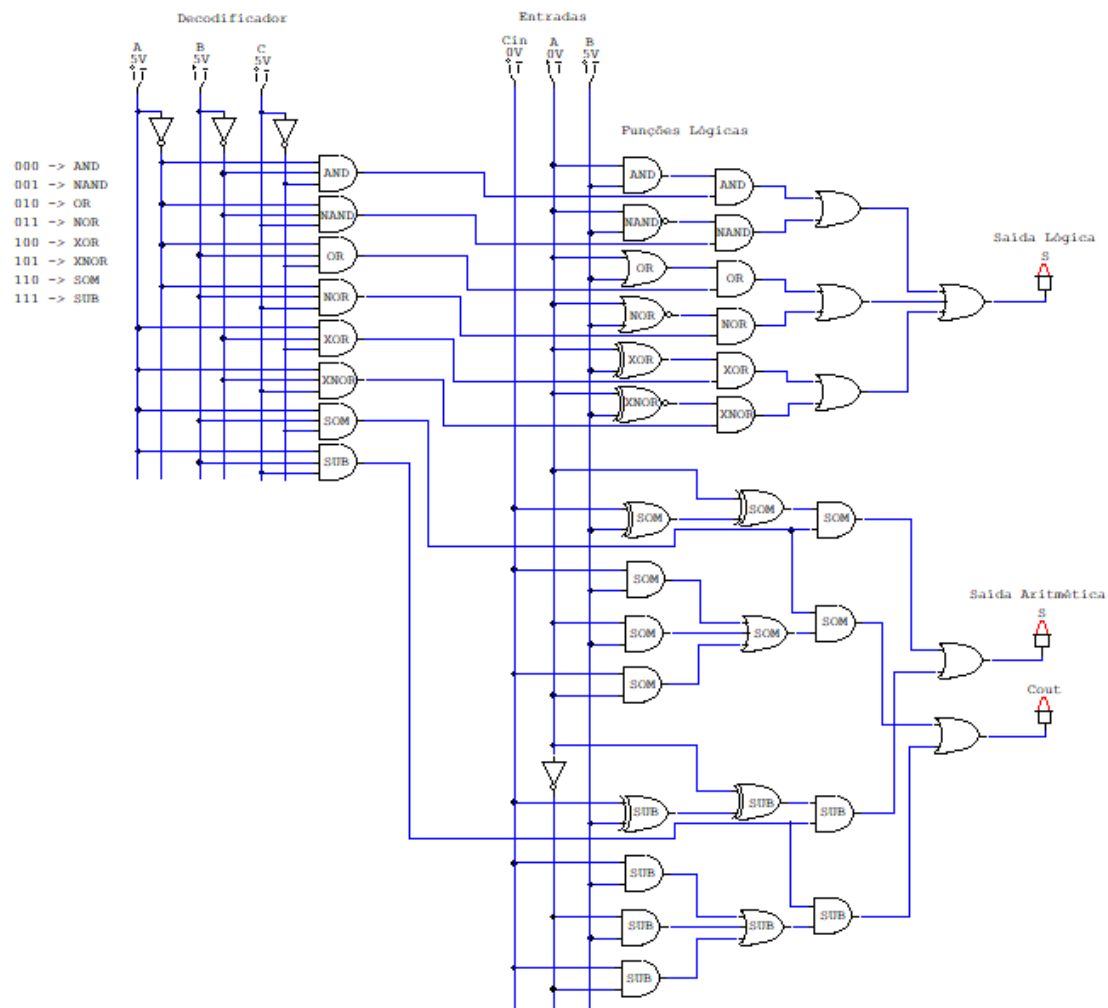


Figura 19. Circuito completo da ULA. (CircuitMaker)

3.2. Projeto II : Memória Semicondutora de 3 Bits

Para a realização deste projeto foram realizados os seguintes passos:

1. Foi utilizado 6 Flip-Flops tipo D, sendo 3 para cada palavra, para energizá-los foi instalado um pulser. Em seguida foi instalado 3 logic switch para a entrada de dados, que foram conectados duas vezes, um no Flip-Flop da primeira palavra e o outro no da segunda palavra.
2. Após isso foi instalado 4 logic switch, uma para a função de leitura/escrita, outra para o Chip Select, outra chamada A0 para escolher qual palavra fará a leitura/escrita e a ultima para habilitar o a saída de dados, esse chamaremos de OE.
3. Para o controle de leitura e escrita foi conectado o Chip Select (CS), Read/Write (RD) e o Pulser em uma porta AND (U7B) de 3 entradas. Portanto, quando o RD estiver desativado (valor lógico 0 no logic switch e 1 na entrada da porta AND (U7B)), a memória fará a leitura da palavra. E quando RD estiver ativado (valor lógico 1 no logic switch e 0 na entrada da porta AND (U7B)), a memória fará a escrita da palavra.

4. A porta AND (U7B) citada acima foi conectada a uma nova AND (CC), juntamente com a entrada A0 Negada, e conectou-se novamente a saída da mesma porta AND (U7B) a uma AND (CB) com A0 (não negada). Assim, Ligamos essas ANDs (CC e CB) cada uma no clock de cada flip-flop. A saída da AND com A0 negado (CC), foi conectada nos Flipflops que armazenam o valor da primeira palavra, e a saída da AND com A0 não negado (CB), foi conectada aos flip-flops que armazenam o valor da segunda palavra. Assim, a partir do valor de A0 selecionado é possível selecionar quais flip-flops receberam energia.
5. Então, foram feitas as ligações das entradas OE negada, CS negada e RD não negada a uma AND (U7A) de 3 entradas. A saída desta porta AND (U7A) foi conectada nas entradas auxiliares dos Buffers (BufeB, BufeC, BufeD). Assim, ao ativar a entrada RD (nível lógico 1) a memória realiza a leitura da palavra, se a memória estiver desligada significa que a entrada CS está ativada (nível lógico 1). Por fim, caso a entrada OE estiver ativada (nível lógico 1) a saída de dados é desabilitada.

A seguir o circuito da Memória Semicondutora de 3 bits finalizado.

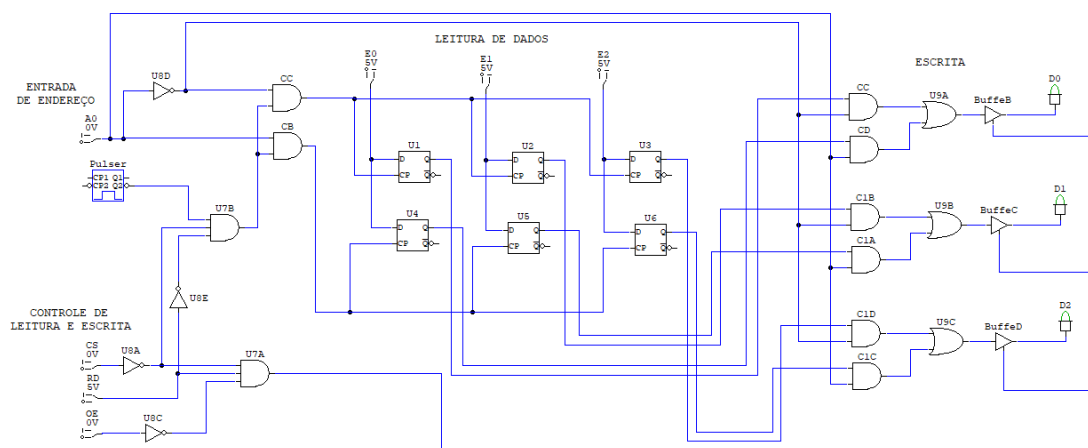


Figura 20. Circuito completo da memória semicondutora de 3 bits. (Circuit Maker)

4. Experimentos realizados

4.1. Unidade Lógica e Aritmética (ULA)

Nesta seção iremos apresentar alguns testes realizados com a ULA.

4.1.1. AND

Neste teste foram setados os valores lógicos das entradas do decodificador como $A = 0$, $B = 0$ e $C = 0$ e os valores lógicos das entradas para as operações com $A = 1$ e $B = 1$.

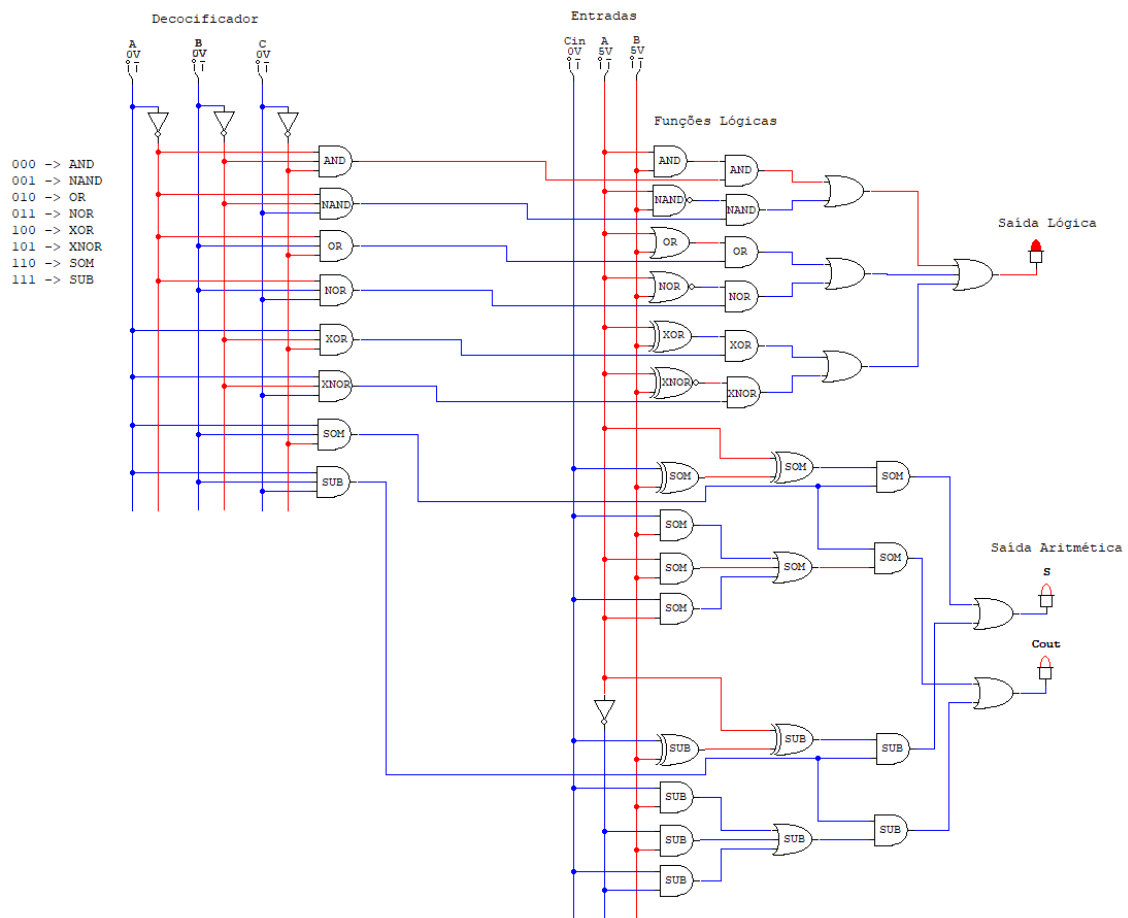


Figura 21. ULA - Teste da operação AND. (CircuitMaker)

4.1.2. NAND

Neste teste foram setados os valores lógicos das entradas do decodificador como A = 0, B = 0 e C = 1 e os valores lógicos das entradas para as operações com A = 0 e B = 1.

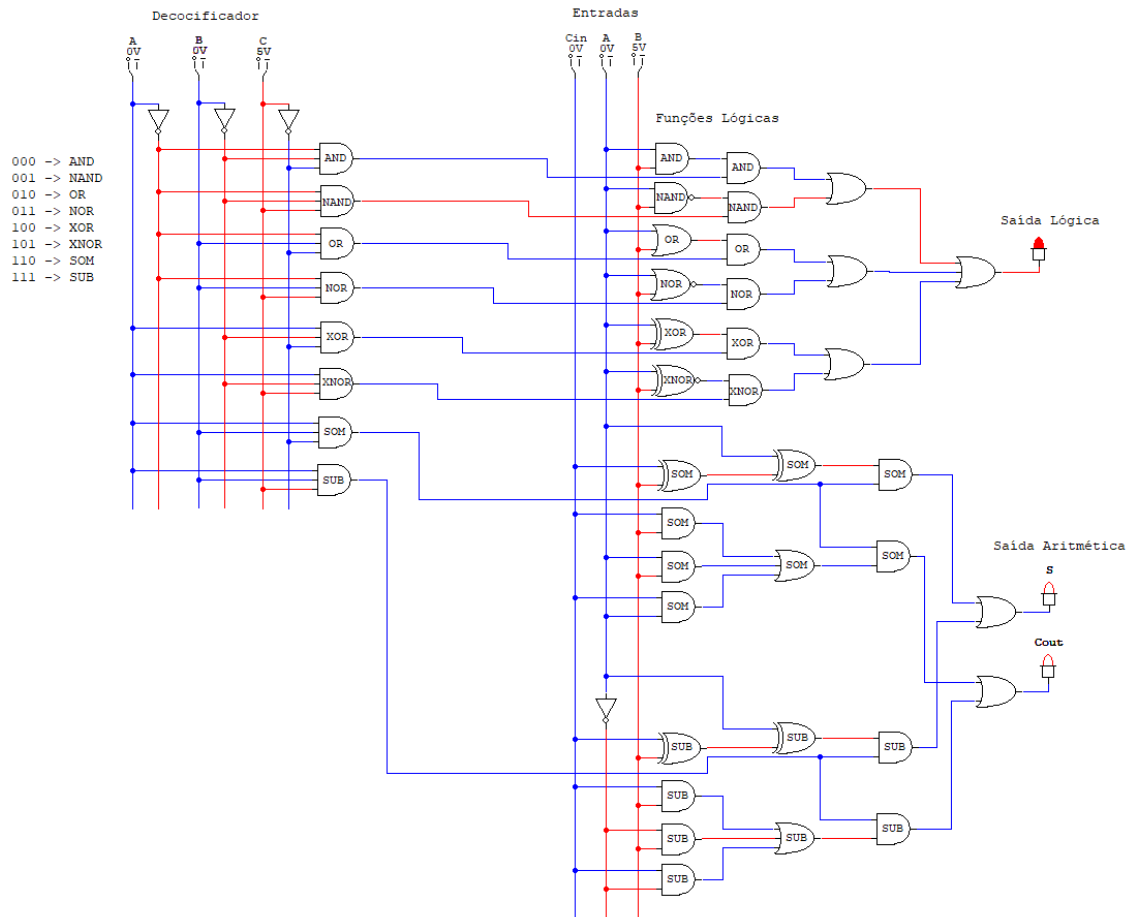


Figura 22. ULA - Teste da operação NAND. (CircuitMaker)

4.1.3. OR

Neste teste foram setados os valores lógicos das entradas do decodificador como $A = 0$, $B = 1$ e $C = 0$ e os valores lógicos das entradas para as operações com $A = 0$ e $B = 1$.

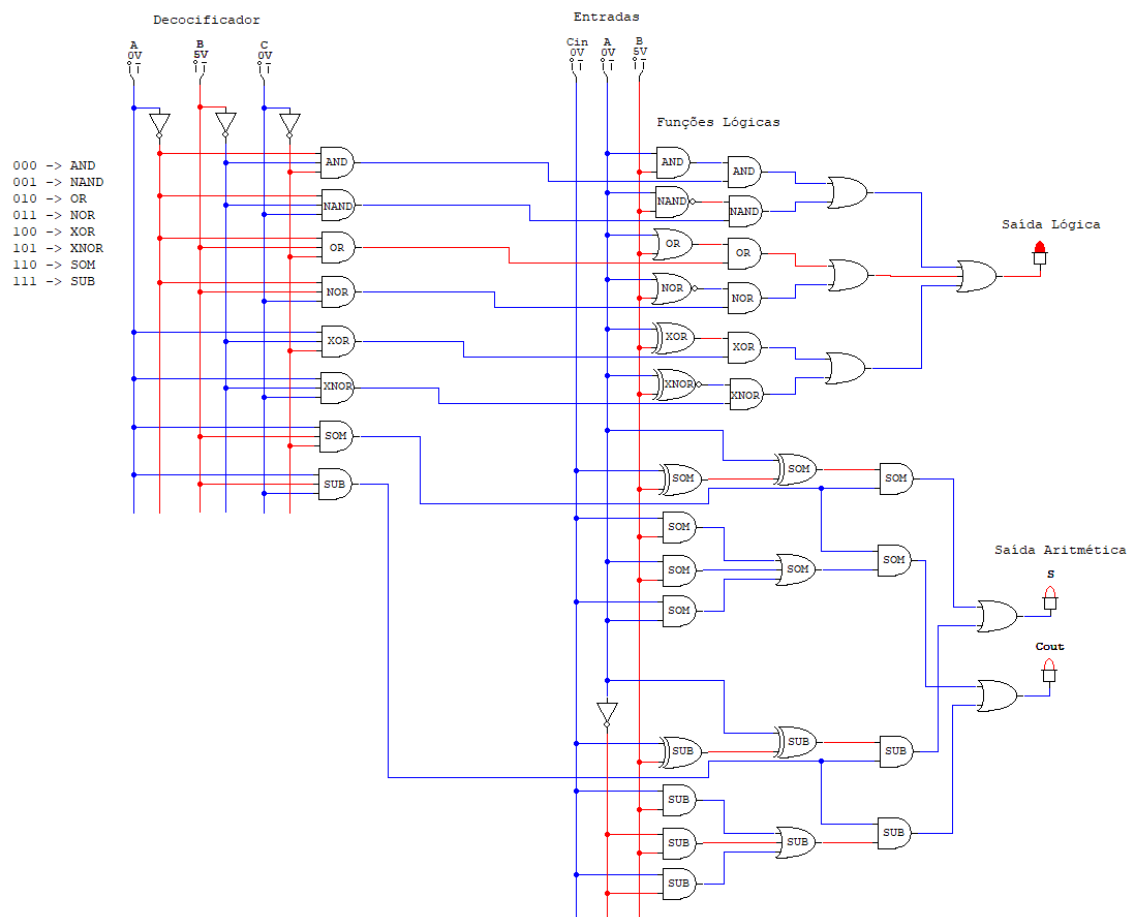


Figura 23. ULA - Teste da operação OR. (CircuitMaker)

4.1.4. NOR

Neste teste foram setados os valores lógicos das entradas do decodificador como $A = 0$, $B = 1$ e $C = 1$ e os valores lógicos das entradas para as operações com $A = 0$ e $B = 0$.

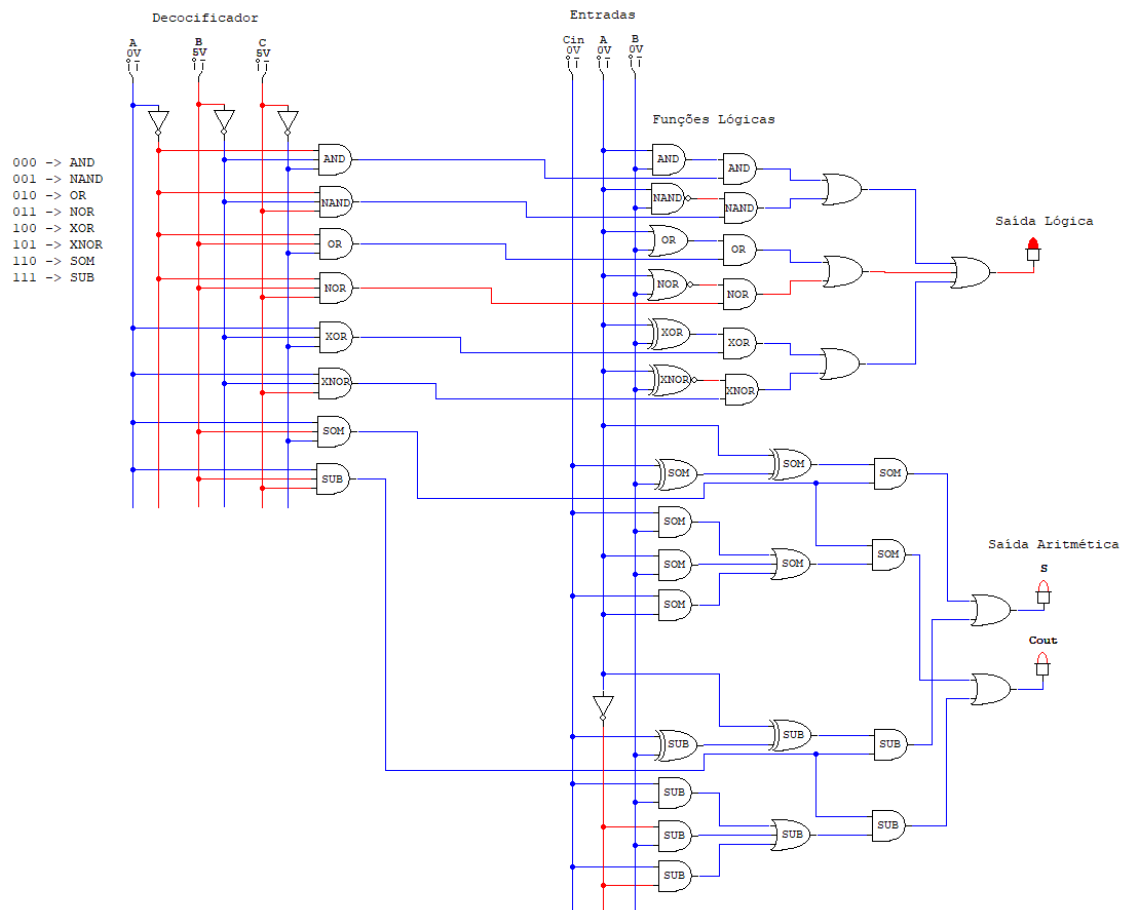


Figura 24. ULA - Teste da operação NOR. (CircuitMaker)

4.1.5. XOR

Neste teste foram setados os valores lógicos das entradas do decodificador como A = 1, B = 0 e C = 0 e os valores lógicos das entradas para as operações com A = 0 e B = 1.

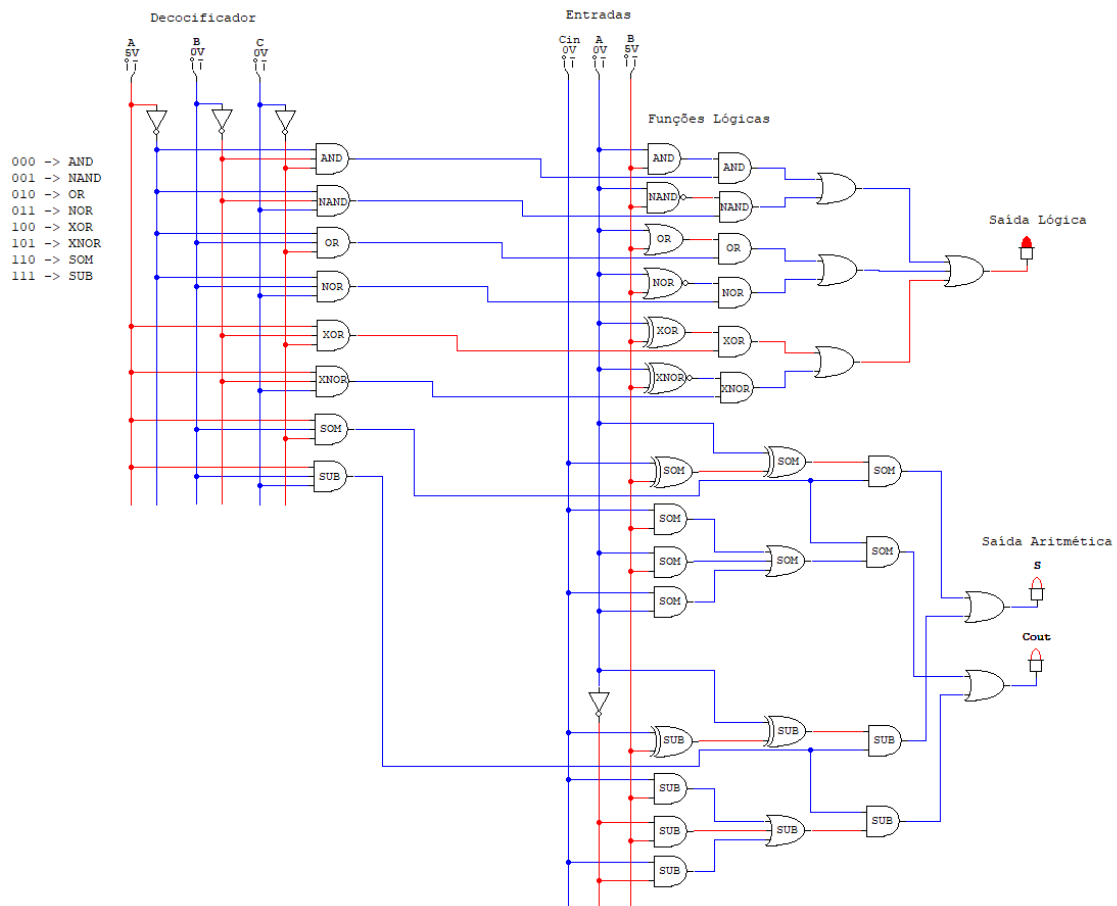


Figura 25. ULA - Teste da operação XOR. (CircuitMaker)

4.1.6. XNOR

Neste teste foram setados os valores lógicos das entradas do decodificador como $A = 1$, $B = 0$ e $C = 1$ e os valores lógicos das entradas para as operações com $A = 1$ e $B = 1$.

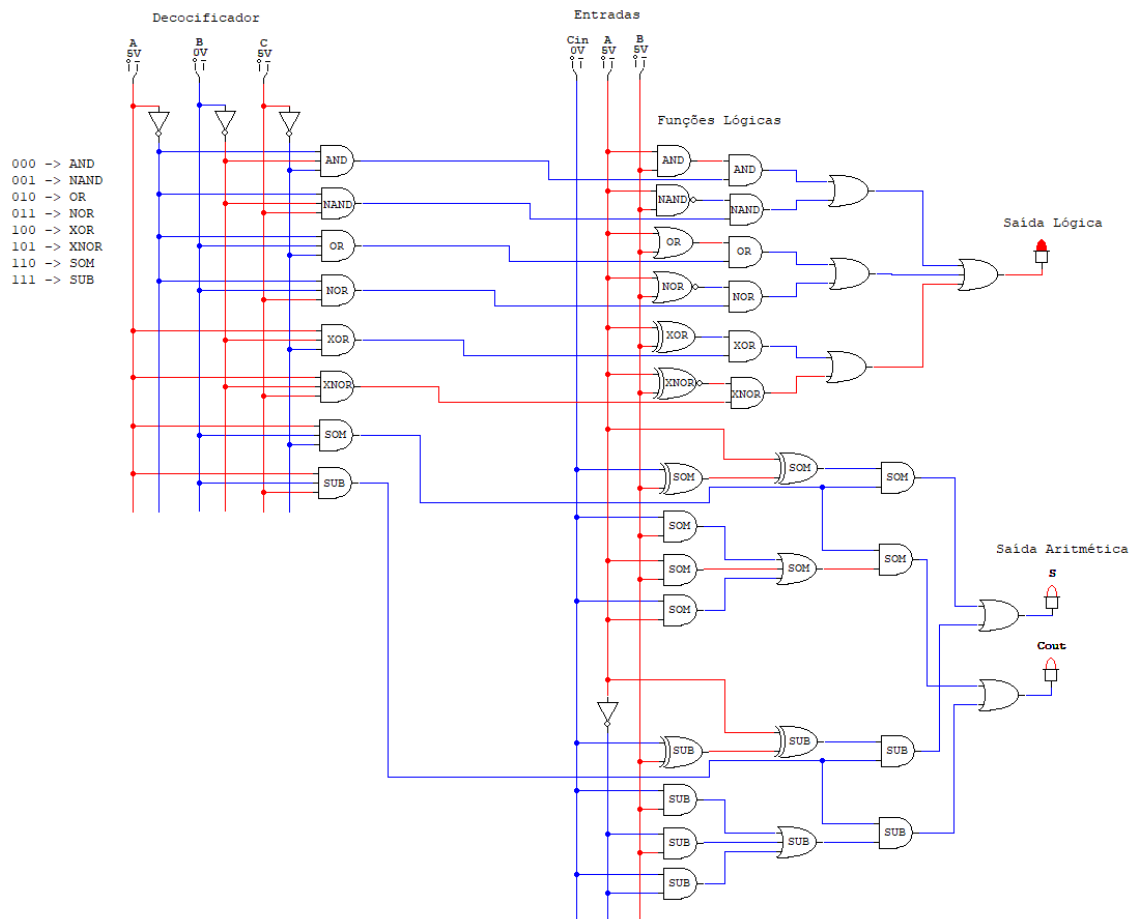


Figura 26. ULA - Teste da operação XNOR. (CircuitMaker)

4.1.7. Adição Binária Completa

Neste teste foram setados os valores lógicos das entradas do decodificador como $A = 1$, $B = 1$ e $C = 0$ e os valores lógicos das entradas para as operações com $A = 0$, $B = 1$ e $Cin = 0$.

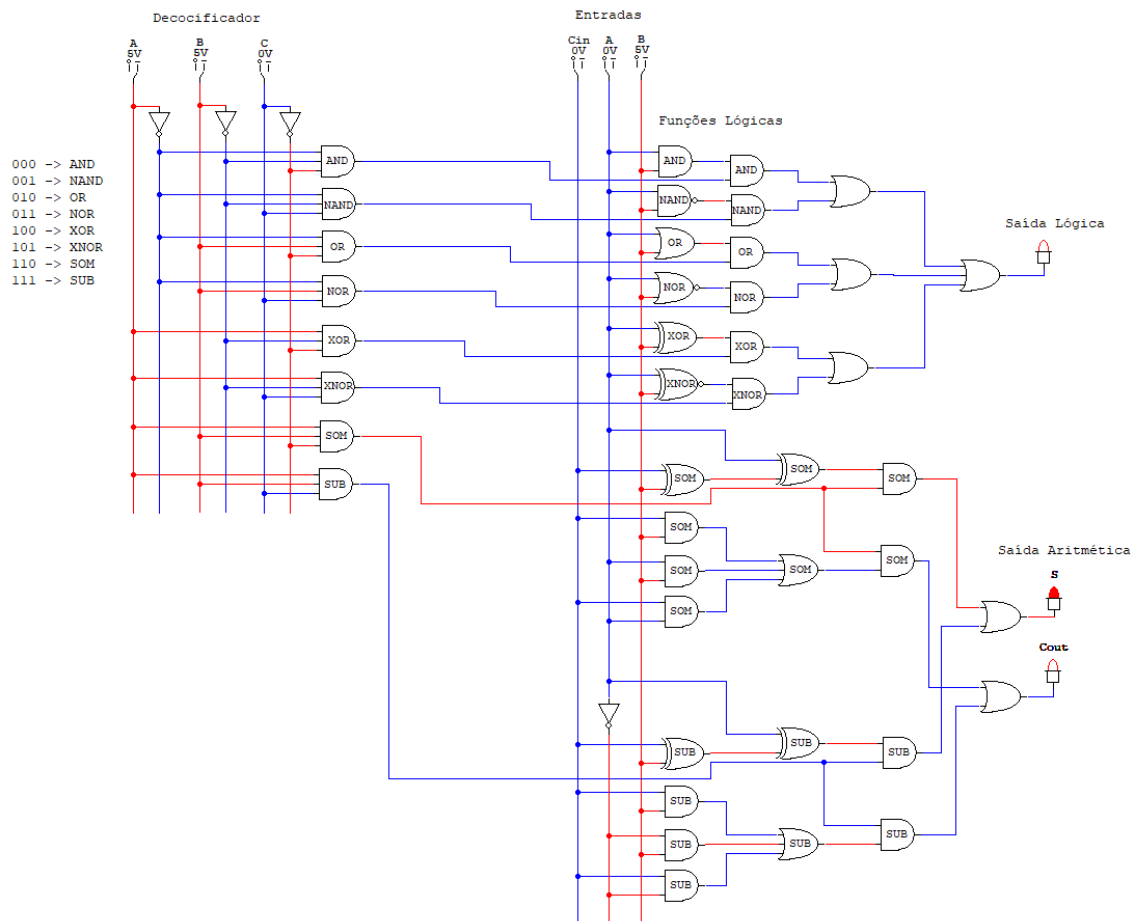


Figura 27. ULA - Teste do SOMADOR. (CircuitMaker)

4.1.8. Subtração Binária Completa

Neste teste foram setados os valores lógicos das entradas do decodificador como $A = 1$, $B = 1$ e $C = 1$ e os valores lógicos das entradas para as operações com $A = 0$, $B = 1$ e $C_{in} = 0$.

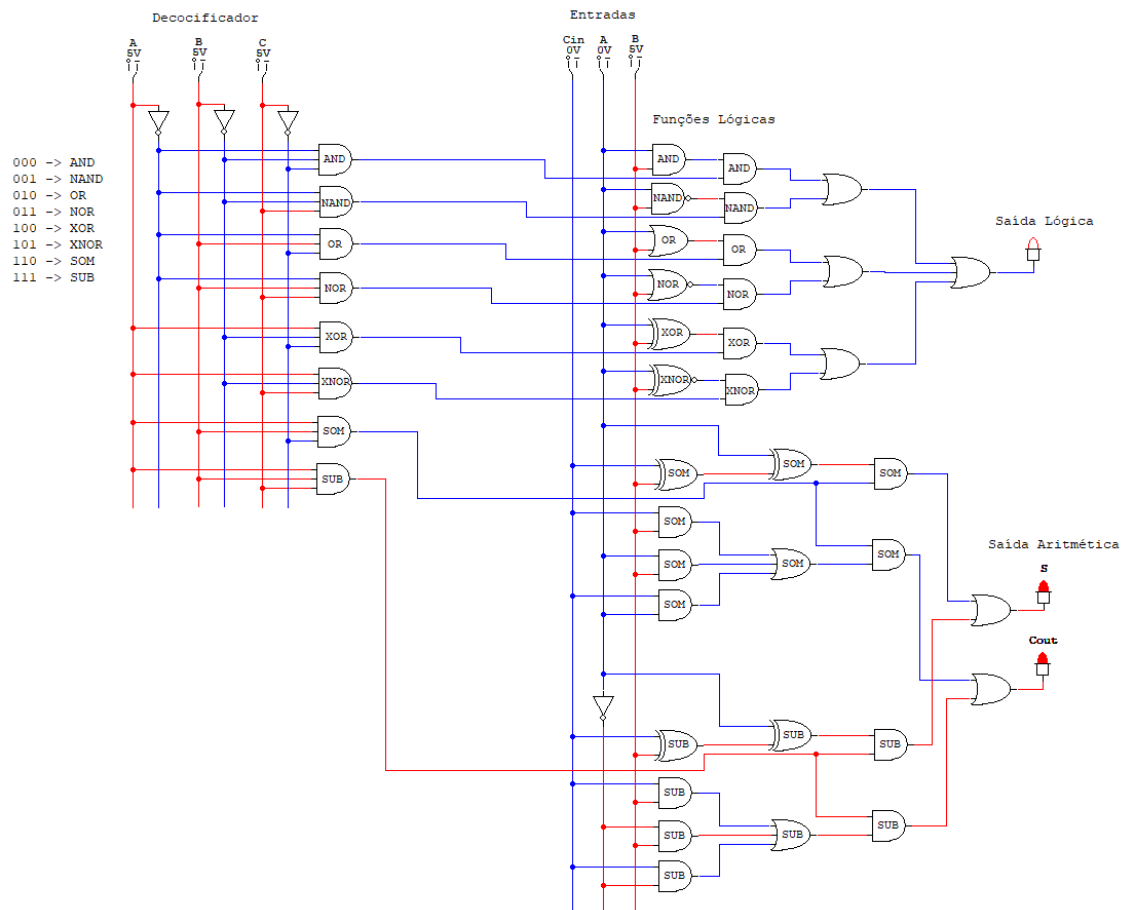


Figura 28. ULA - Teste do SUBTRATOR. (CircuitMaker)

4.2. Memória Semicondutora de 3 Bits

Para os testes foi proposto a leitura e a escrita das seguintes palavras de 3 bits:

- 111
- 101
- 010
- 000

4.2.1. 111

Leitura:

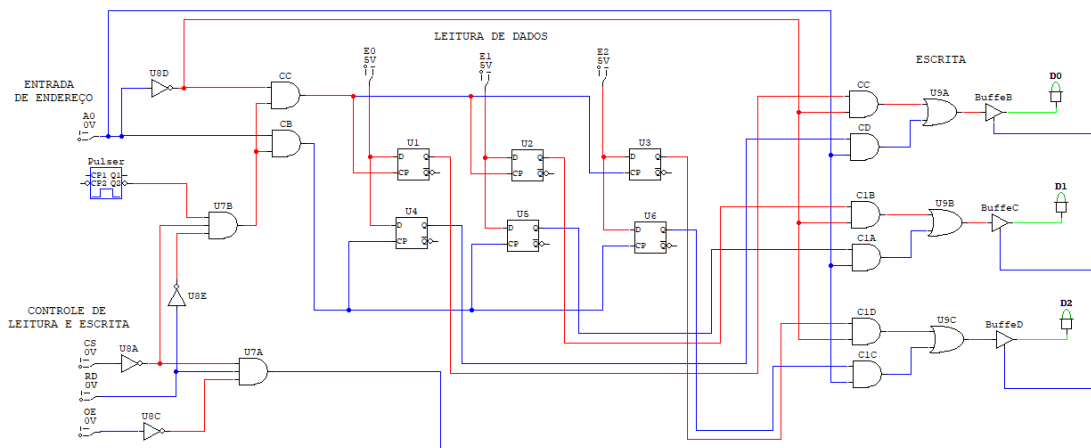


Figura 29. Leitura da palavra de 3 bits 111. (Circuit Maker)

Escrita:

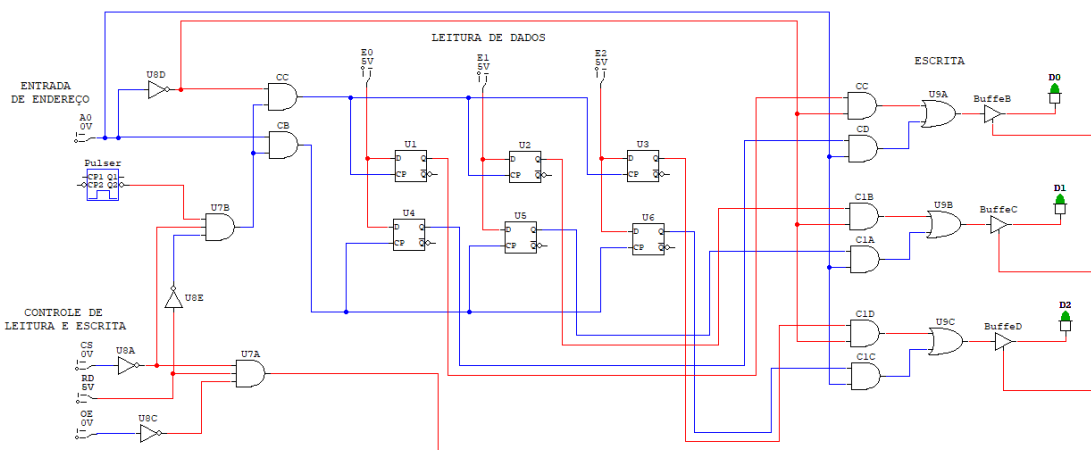


Figura 30. Escrita 1 (111). (Circuit Maker).

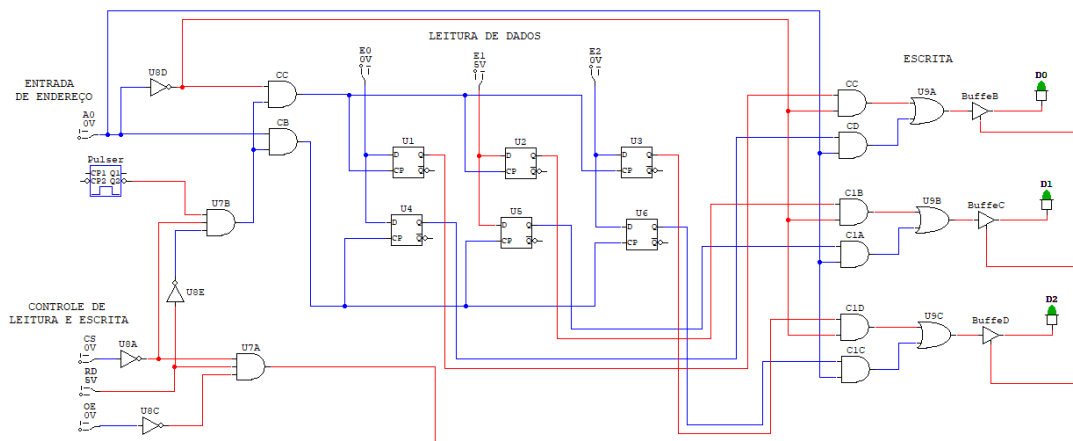


Figura 31. Escrita 2 (111). (Circuit Maker).

4.2.2. 101

Leitura:

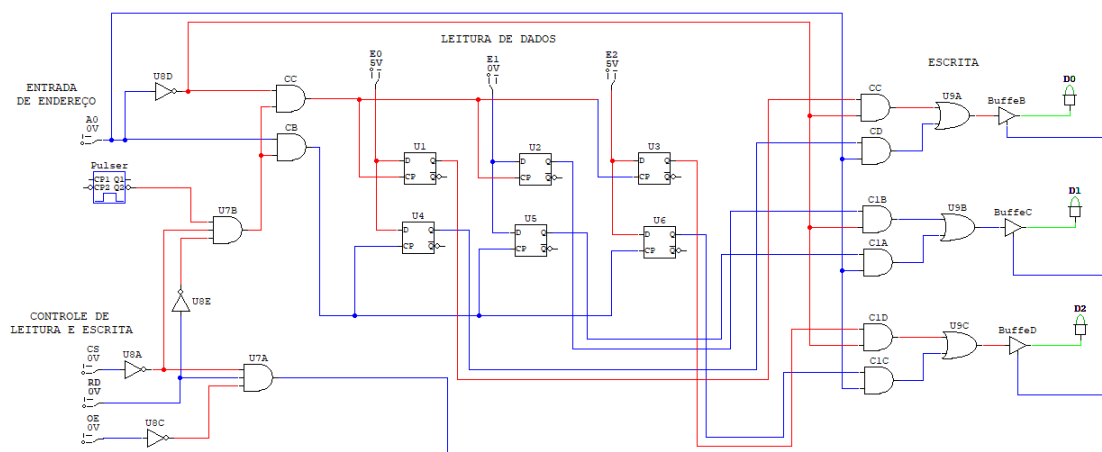


Figura 32. Leitura da palavra de 3 bits 101. (Circuit Maker)

Escrita:

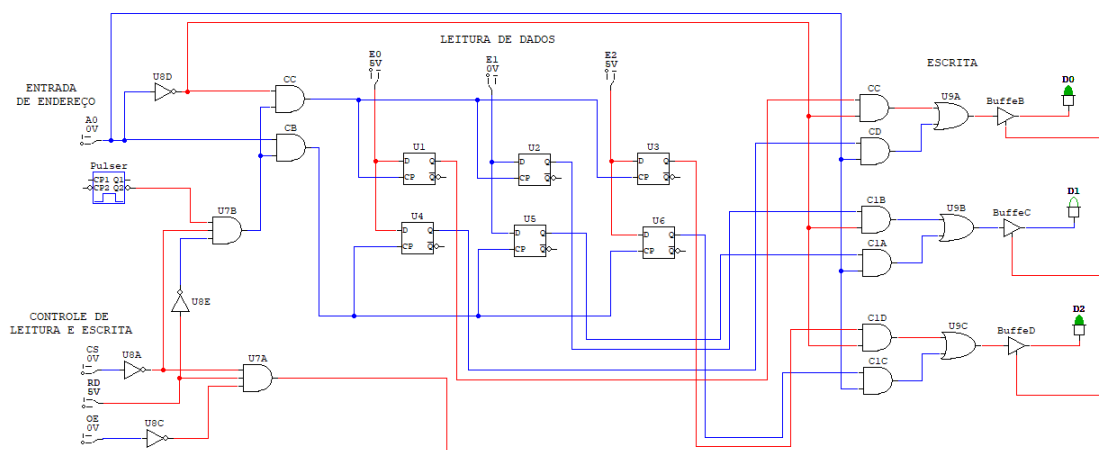


Figura 33. Escrita da palavra de 3 bits 101. (Circuit Maker)

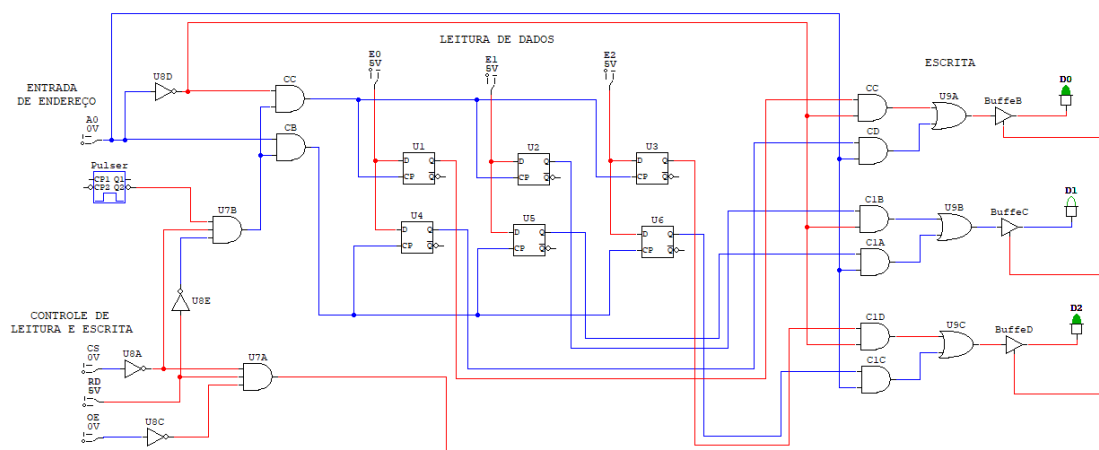


Figura 34. Confirmação da escrita da palavra de 3 bits 101. (Circuit Maker)

4.2.3. 010

Leitura:

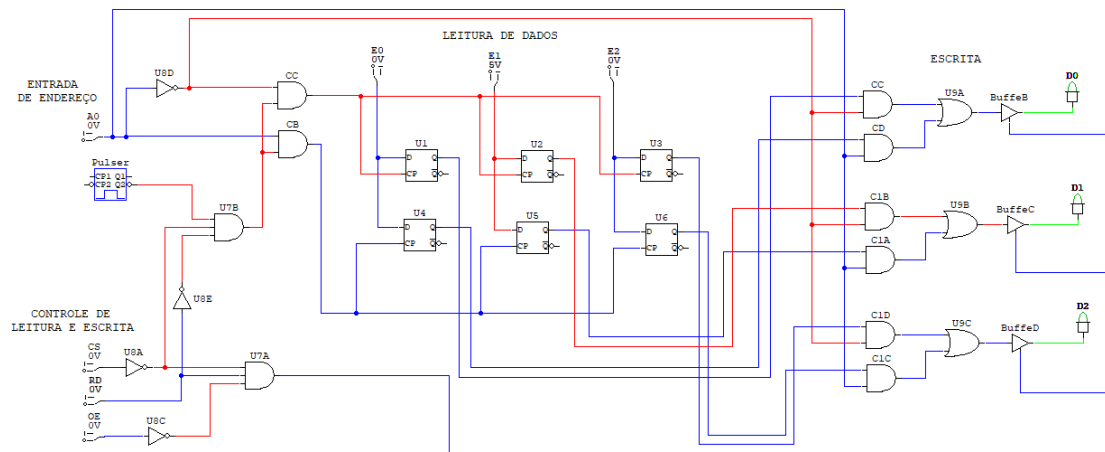


Figura 35. Leitura da palavra de 3 bits 010. (Circuit Maker)

Escrita:

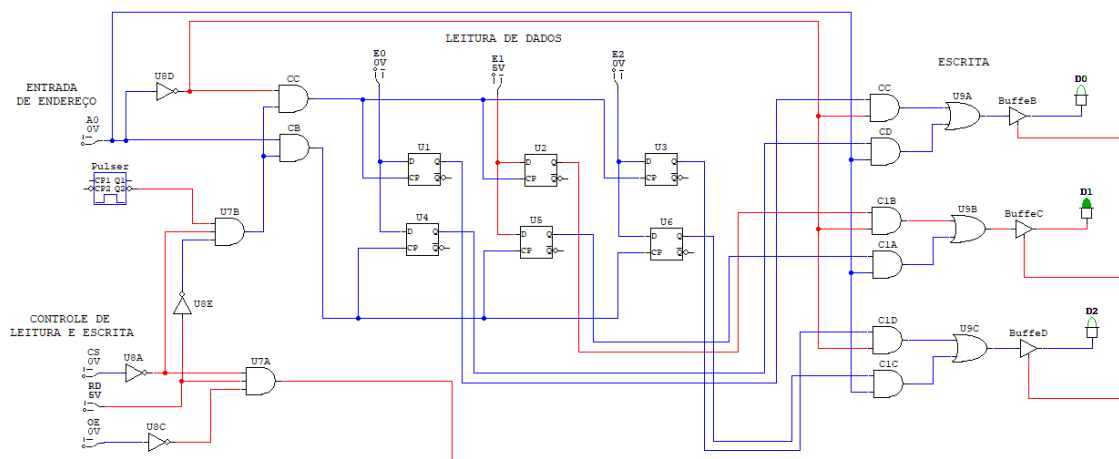


Figura 36. Escrita da palavra de 3 bits 010. (Circuit Maker)

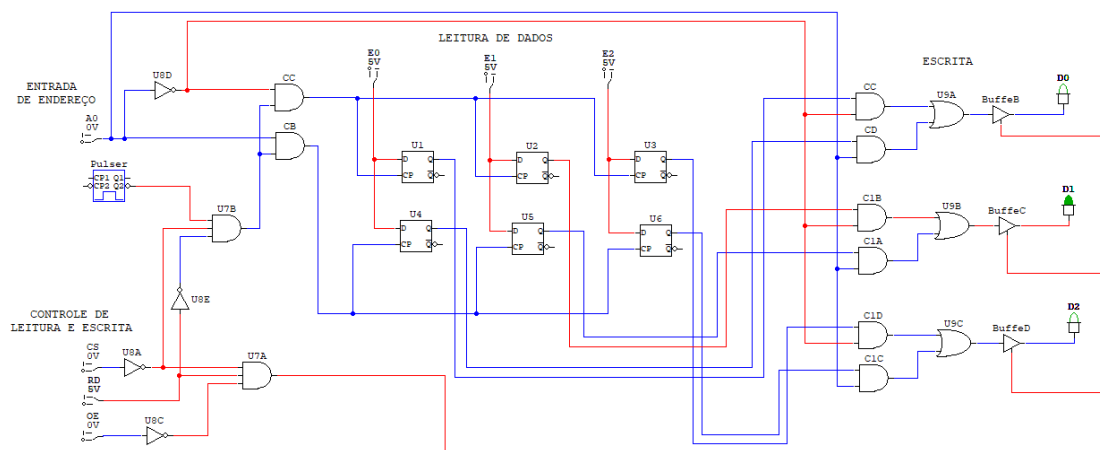


Figura 37. Confirmação da escrita da palavra de 3 bits 010. (Circuit Maker)

4.2.4. 000

Leitura:

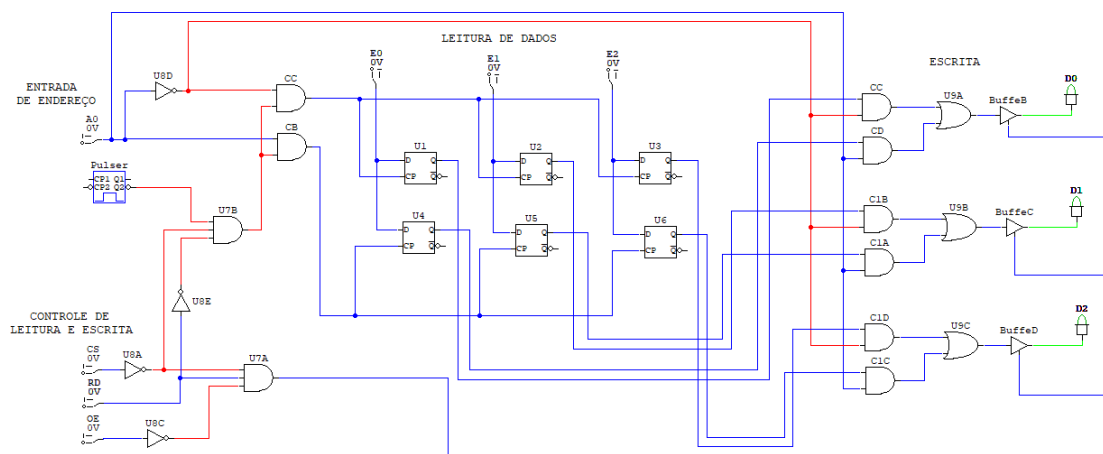


Figura 38. Leitura da palavra de 3 bits 000. (Circuit Maker)

Escrita:

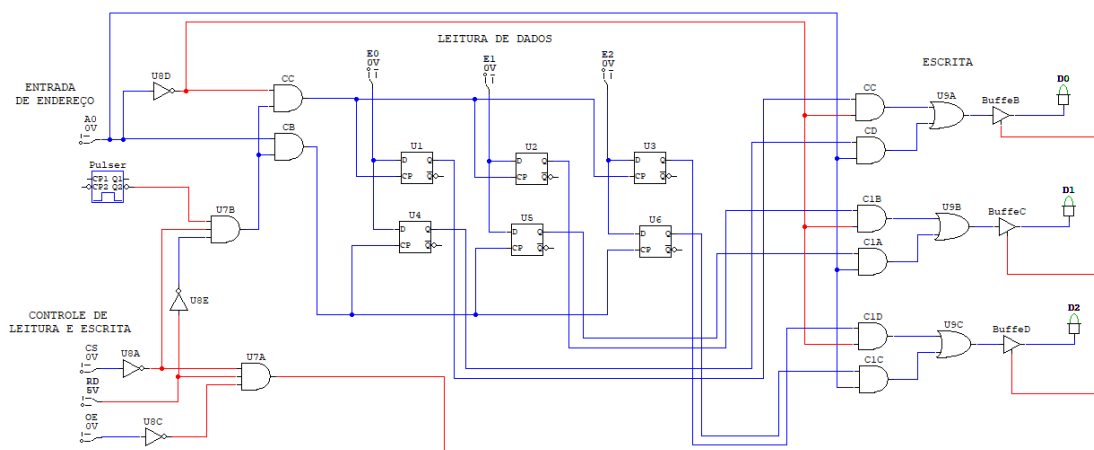


Figura 39. Escrita da palavra de 3 bits 000. (Circuit Maker)

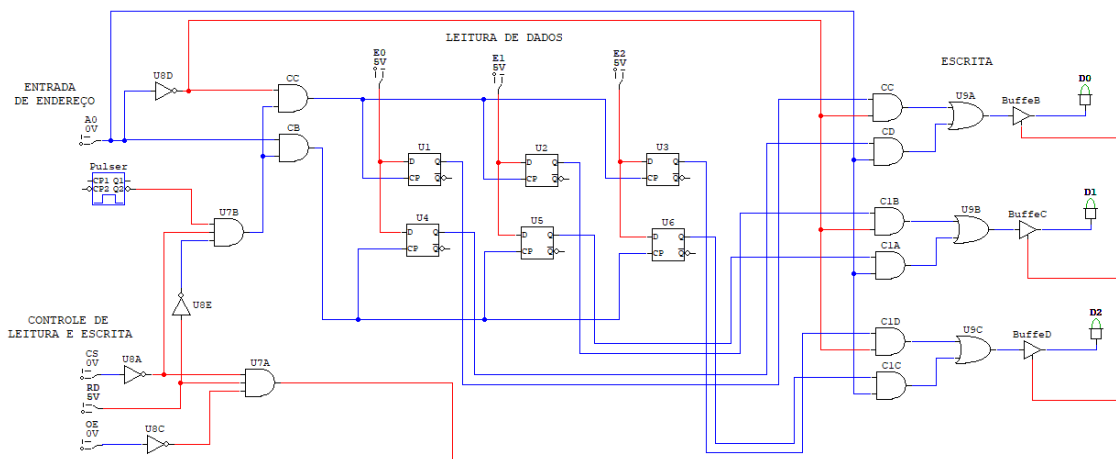


Figura 40. Confirmação da escrita da palavra de 3 bits 000. (Circuit Maker)

5. Conclusão

O objetivo desse trabalho era apresentar os conceitos básicos da ULA de 1 bit e suas operações de maneira teórica e prática, junto com a memória semicondutora de 3 bits. Vimos que é possível representar a ULA e realizar todas as suas operações comprovando a veracidade de sua tabela verdade utilizando o circuito que construímos como também a leitura e escrita da memória.

Referências

Junior, R. C. (2011). Memórias semicondutoras. <http://www.corradi.junior.nom.br/memorias1.pdf>.