



Universidade Estadual de Maringá



Centro de Tecnologia

Departamento de Informática

Informática

Circuitos Digitais (5174)

Relatório do Trabalho 1: Unidade Lógica e Aritmética de 1 BIT (ULA)

16 de novembro de 2017

Professor: Carlos Fransley Scatambulo Costa

Discente:

RA:	Nome:
98336	Raphael Franco de Lima

Lista de Tabelas

TABELA 1 SAÍDAS DO DECODIFICADOR E RESPECTIVAS FUNÇÕES	5
TABELA 2 TABELA VERDADE DO DECODIFICADOR 4:16	5
TABELA 3 TABELA VERDADE DO SOMADOR COMPLETO	9
TABELA 4 TABELA VERDADE DO SUBTRATOR COMPLETO	10
TABELA 5 TABELA VERDADE DO SOMADOR E SUBTRATOR COMPLETO	11

Lista de Figuras

FIGURA 1 CIRCUITO DO DECODIFICADOR -----	7
FIGURA 2 CIRCUITO DO BLOCO LÓGICO -----	8
FIGURA 3 CIRCUITO DO SOMADOR COMPLETO -----	9
FIGURA 4 MAPA DE KARNAUGH PARA SAÍDA COUT -----	10
FIGURA 5 CIRCUITO DIGITAL DO SUBTRATOR COMPLETO -----	11
FIGURA 6 MAPA DE KARNAUGH PARA A SAÍDA COUT DO SOMADOR E SUBTRATOR COMPLETO -----	12
FIGURA 7 CIRCUITO SOMADOR E SUBTRATOR COMPLETO COM BIT DE SELEÇÃO -----	12
FIGURA 8 SOMADOR E SUBTRATOR COMPLETOS SEM O BIT DE SELEÇÃO -----	13
FIGURA 9 ULA COMPLETA -----	14

Sumário

Lista de Tabelas	2
Lista de Figuras	2
Introdução	4
ULA	4
Decodificador	4
Bloco Lógico	7
Bloco Aritmético	8
Somador Completo	8
Subtrator Completo	10
Somador e Subtrator Completo	11
Integração de todos os componentes.....	12
Conclusão	14
Referências Bibliográficas	14

Introdução

Segundo (GUNTZEL e NASCIMENTO, 2001) um circuito combinacional é constituído por um conjunto de portas lógicas as quais determinam os valores das saídas diretamente a partir dos valores atuais das entradas. Estes circuitos constituem uma classe básica, porém importante de circuitos digitais, e que assumem certa importância quando funcionam de base para classes mais complexas de circuitos.

Este trabalho possui o objetivo de estudar e analisar os circuitos combinacionais através da construção de uma unidade lógica e aritmética (ULA).

ULA

O matemático John von Neumann propôs o conceito de ULA em 1945, segundo ele a ULA era uma necessidade para o computador porque ela garantiria que o computador calcularia operações matemáticas básicas, incluindo adição, subtração, multiplicação e divisão. Ele então achava razoável que um computador contivesse um órgão especializado para essas operações (STALLINGS, 2009).

Segundo (VAHID, 2008) O funcionamento de uma ULA pode ser descrito como um componente de bloco operacional capaz de executar diversas operações aritméticas e lógicas com duas entradas de dados, com N bits de larguras, gerando uma saída de dados de N bits.

A ULA cuja será apresentada neste trabalho é constituída de duas operações aritméticas e sete operações lógicas, a sua construção e detalhamento se encontram no decorrer do texto.

Decodificador

De acordo com (FLOYD, 2007), um decodificador é um circuito digital que detecta a presença de uma combinação específica de bits (código) em suas entradas indicando a presença desse código através de um nível de saída especificado. Em sua forma geral, um decodificador tem n linhas de entrada para manipular n bits e de uma a 2n linhas de saída para indicar a presença de uma ou mais combinações de n bits.

O decodificador a ser implementado neste projeto, devido a necessidade de 10 saídas (foram usadas 10 ao invés de 9 saídas pois foram adicionadas duas portas NOT sendo uma para cada entrada), é um decodificador 4:16 sendo que este possui 6 condições irrelevantes. Cada função da ULA só será executada caso a opção selecionada através do decodificador estiver ativa (nível lógico um), enquanto que as outras saídas estarão desativas (nível lógico zero). Segue abaixo a tabela que indica as saídas do decodificador e as respectivas funções que este deverá ativar, seguida da tabela verdade para o decodificador desenvolvido.

Saída	Operação
S0	Adição
S1	Subtração
S2	NOT (Y)
S3	AND
S4	NAND
S5	OR
S6	NOR
S7	XOR
S8	XNOR
S9	NOT (X)

Tabela 1 Saídas do decodificador e respectivas funções

A	B	C	D	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	X	X	X	X	X	X	X	X	X	X
1	0	1	0	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X
1	1	1	1	0	0	0	0	0	0	0	0	0	1

Tabela 2 Tabela verdade do decodificador 4:16

Para que se possa montar as expressões booleanas de cada saída será desenvolvido o mapa de Karnaugh para as respectivas saídas de S0 a S9.

S0(A, B, C, D)

s0	\overline{C}		C		\overline{B}
	1	0	0	0	
A	0	0	0	0	B
	x	x	0	x	
	0	x	x	x	\overline{B}
	\overline{D}	D		\overline{D}	

S1(A, B, C, D)

s1	\overline{C}		C		\overline{B}
	0	1	0	0	
A	0	0	0	0	B
	x	x	0	x	
	0	x	x	x	\overline{B}
	\overline{D}	D		\overline{D}	

S2(A, B, C, D)

s2	\overline{C}		C		
\overline{A}	0	0	0	1	\overline{B}
	0	0	0	0	
A	X	X	0	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

S3(A, B, C, D)

s3	\overline{C}		C		
\overline{A}	0	0	1	0	\overline{B}
	0	0	0	0	
A	X	X	0	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

S4(A, B, C, D)

s4	\overline{C}		C		
\overline{A}	0	0	0	0	\overline{B}
	1	0	0	0	
A	X	X	0	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

S5(A, B, C, D)

s5	\overline{C}		C		
\overline{A}	0	0	0	0	\overline{B}
	0	1	0	0	
A	X	X	0	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

S6(A, B, C, D)

s6	\overline{C}		C		
\overline{A}	0	0	0	0	\overline{B}
	0	0	0	1	
A	X	X	0	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

S7(A, B, C, D)

s7	\overline{C}		C		
\overline{A}	0	0	0	0	\overline{B}
	0	0	1	0	
A	X	X	0	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

S8(A, B, C, D)

s8	\overline{C}		C		
\overline{A}	0	0	0	0	\overline{B}
	0	0	0	0	
A	X	X	0	X	B
	1	X	X	X	
	\overline{D}	D		\overline{D}	

S9(A, B, C, D)

s9	\overline{C}		C		
\overline{A}	0	0	0	0	\overline{B}
	0	0	0	0	
A	X	X	1	X	B
	0	X	X	X	
	\overline{D}	D		\overline{D}	

A simplificação utilizando mapa de Karnaugh, resulta nas seguintes expressões booleanas.

$$S0 = \overline{A}.\overline{B}.\overline{C}.\overline{D}$$

$$S1 = \overline{B}.\overline{C}.D$$

$$S2 = \overline{B}.C.\overline{D}$$

$$S3 = \overline{B}.C.D$$

$$S4 = B.\overline{C}.\overline{D}$$

$$S5 = B.\overline{C}.D$$

$$S6 = B.C.\overline{D}$$

$$S7 = \overline{A}.B.C.D$$

$$S8 = A.\overline{B}.\overline{C}.\overline{D}$$

$$S9 = A.B.C.D$$

O circuito lógico resultante é mostrado abaixo:

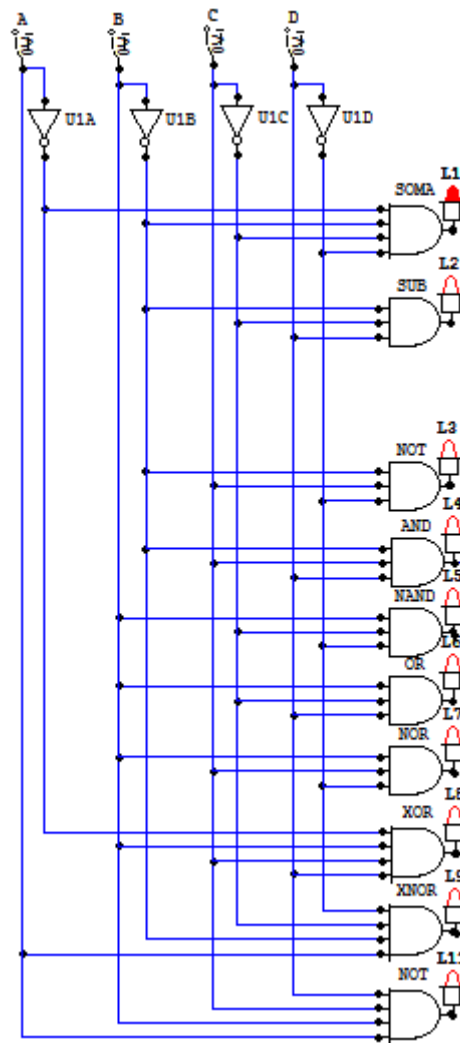


Figura 1 Circuito do decodificador

Bloco Lógico

O bloco lógico desenvolvido neste trabalho é responsável pelas sete operações lógicas já enunciadas. A sua construção foi realizada com duas entradas denominadas X e Y e uma saída S (cuja só será visualizada na finalização do projeto). Para implementar as funções contidas neste bloco foram usadas portas lógicas das respectivas funções.

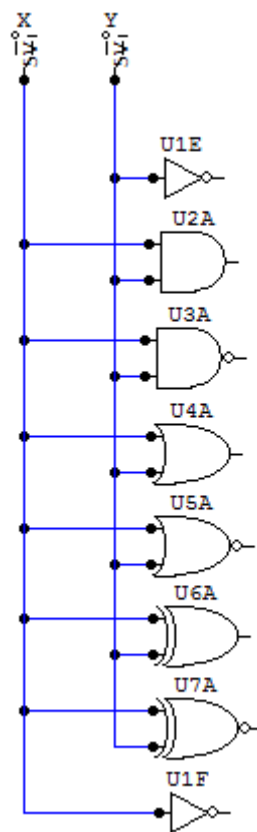


Figura 2 Circuito do bloco lógico

Bloco Aritmético

O bloco aritmético é responsável pela realização das operações aritméticas, no escopo deste trabalho é realizada a adição e subtração. Este é constituído por três entradas X, Y e Carry In, como saída resultante temos S, cuja armazena o resultado da operação e o Carry Out.

Somador Completo

Segundo (IDOETA e CAPUANO, 2008), para efetuarmos a adição no sistema binário, devemos agir numa adição convencional no sistema decimal, lembrando que, no sistema binário, temos apenas dois Algarismos, zero ou um. A operação de soma pode ser demonstrada da seguinte forma:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1 \ 0 \text{ (0 e "vai um")}$$

A operação “vai um” é denominada Carry onde C_{in} será a entrada do Carry e C_{out} será a saída do mesmo. A tabela verdade é exibida abaixo.

X	Y	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 3 Tabela verdade do somador completo

Com a tabela montada podemos utilizar o mapa de Karnaugh para simplificar as expressões.

S	\bar{Y}		Y	
\bar{X}	0	1	0	1
X	1	0	1	0
	\bar{Cin}	Cin		\bar{Cin}

Cout	\bar{Y}		Y	
\bar{X}	0	0	1	0
X	0	1	1	1
	\bar{Cin}	Cin		\bar{Cin}

Para a saída S temos a seguinte expressão simplificada:

$$S = X \oplus Y \oplus Cin$$

Para a saída Cout temos a seguinte expressão simplificada

$$Cout = X.Cin + Y.Cin + X.Y$$

Com essas informações podemos implementar o circuito lógico do somador completo.

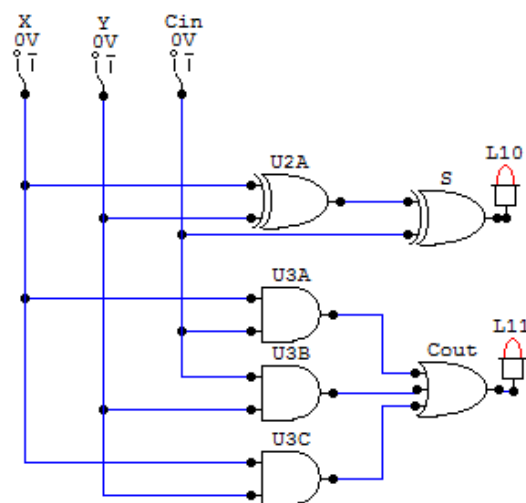


Figura 3 Circuito do somador completo

Subtrator Completo

Segundo (IDOETA e CAPUANO, 2008), o método de resolução é análogo a uma subtração no sistema decimal. Podemos demonstrá-la da seguinte forma:

$$0 - 0 = 0$$

$$0 - 1 = 1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

No caso 0 - 1, o resultado será igual a um, porém haverá um transporte, também chamado de “empresta um”, para a coluna seguinte que deve ser acumulado no subtraendo e, conseqüentemente, subtraído do minuendo. Esse transporte também é chamado Carry. A tabela verdade resultante desta operação é mostrada a seguir:

X	Y	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 4 Tabela verdade do subtrator completo

Analisando as tabelas verdade do somador e do subtrator nota-se que ambas possuem a saída S iguais, por isso, não há a necessidade de se construir novamente o mapa de Karnaugh para a mesma. Têm-se então o mapa para a saída Cout:

Cout	\overline{Y}		Y	
	0	1	1	1
\overline{X}	0	1	1	1
X	0	0	1	0
	\overline{Cin}	Cin		\overline{Cin}

Figura 4 Mapa de Karnaugh para saída Cout

Resultando na seguinte expressão simplificada:

$$Cout = Y.\overline{X} + \overline{X}.Cin + Y.Cin$$

E para S basta usarmos a mesma expressão resultante do somador.

$$S = X \oplus Y \oplus Cin$$

Com essas informações é possível implementar o circuito lógico do subtrator completo.

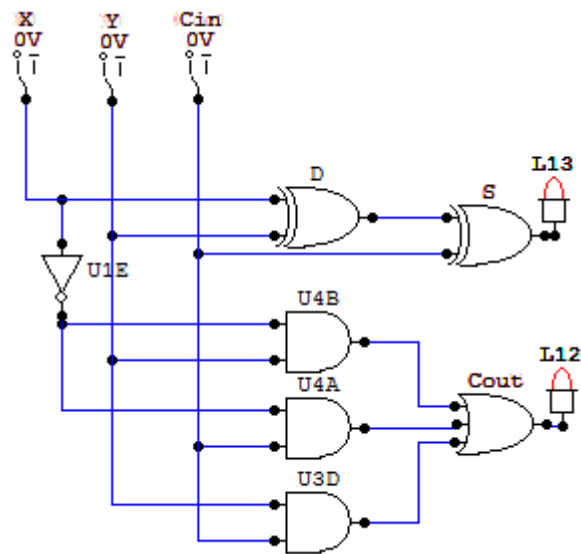


Figura 5 Circuito digital do subtrator completo

Somador e Subtrator Completo

Como vimos, o somador e o subtrator possuem a mesma saída S, por isso, é possível adicionar um bit de controle e então construir um circuito que execute as duas operações e exiba a saída Cout correta da operação correspondente. Podemos construir a tabela verdade da seguinte forma:

M	X	Y	Cin	S	Cout
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

Tabela 5 Tabela verdade do somador e subtrator completo

Neste caso não é necessário elaborar o mapa de Karnaugh para a saída S pois M será simplificado, basta então montarmos o mapa para a saída Cout.

Cout	\overline{Y}		Y		
\overline{M}	0	0	1	0	\overline{X}
	0	1	1	1	X
M	0	0	1	0	
	0	1	1	1	\overline{X}
	\overline{Cin}	Cin		\overline{Cin}	

Figura 6 Mapa de Karnaugh para a saída Cout do somador e subtrator completo

A expressão resultante da simplificação é dada por:

$$Cout = Y.Cin + M \oplus X.(Y + Cin)$$

O circuito lógico resultante do somador e subtrator completo é exibido abaixo:

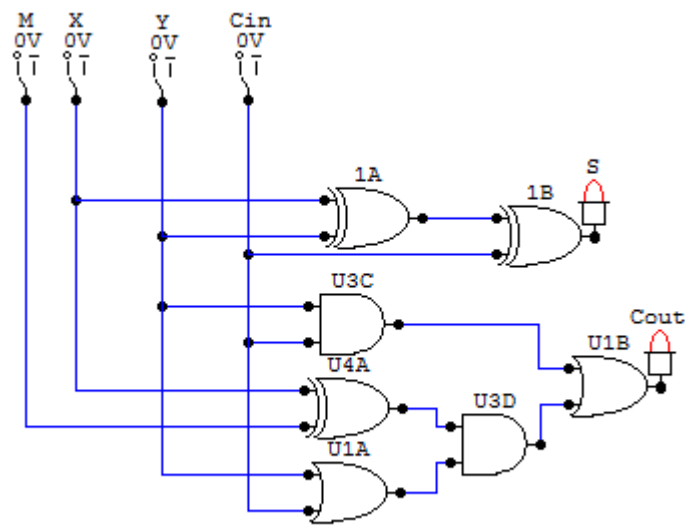


Figura 7 Circuito somador e subtrator completo com bit de seleção

Integração de todos os componentes

Para realizar a integração de todos os componentes pode-se pensar primeiramente na integração do bloco lógico e em seguida o bloco aritmético. Para o primeiro basta ligar através de uma porta AND cada saída do decodificador e com a função ao qual esta se refere. Para a integração do Bloco Aritmético com o decodificador foram realizadas

algumas adaptações para que C_{out} só seja exibido quando for selecionado as funções do bloco aritmético. Para realizar essa adaptação foi interligada em uma porta AND a saída de C_{out} e a saída de uma porta OR tendo como entrada a saída do decodificador correspondente às funções do somador e do subtrator, isso garante que C_{out} será exibido se, e somente se, a função de soma ou subtração for selecionada. Uma outra adaptação realizada se refere ao bit de controle M, cujo seleciona o subtrator ou somador, nesse caso o decodificador só tem a saída 1 para a função que estiver selecionada através do código de entrada, recebida de A, B, C e D, então inicialmente não será possível implementar esse bit de controle, porém, para resolver este problema foram usadas duas portas ANDs, a primeira possuindo como entrada a saída do decodificador referente a soma e X, já na segunda porta é ligada a saída do decodificador referente a subtração e X invertido. As saídas destas portas ANDs são ligadas então a porta XOR.

As modificações no somador e subtrator são mostradas abaixo:

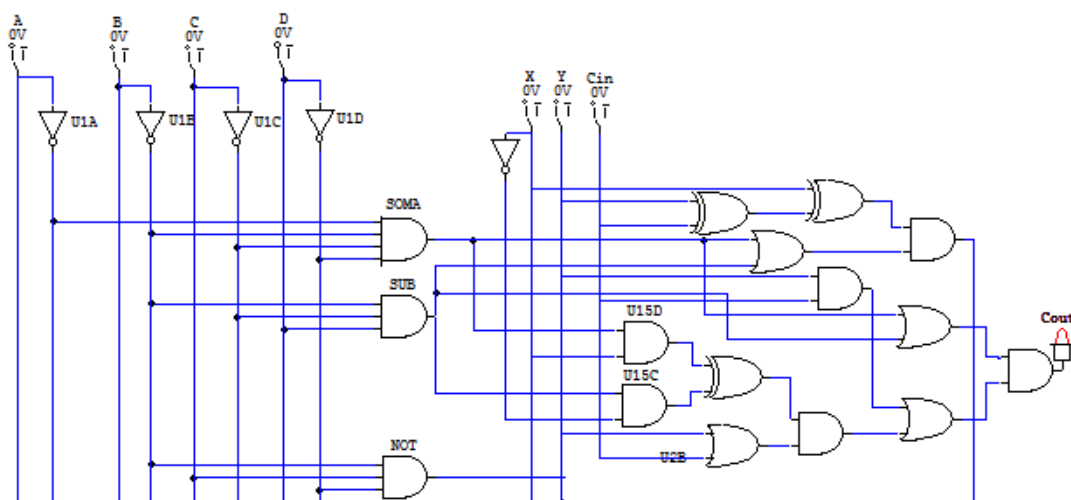


Figura 8 Somador e Subtrator completos sem o bit de seleção

Feitas estas modificações, temos a ULA completa implementada da seguinte forma:

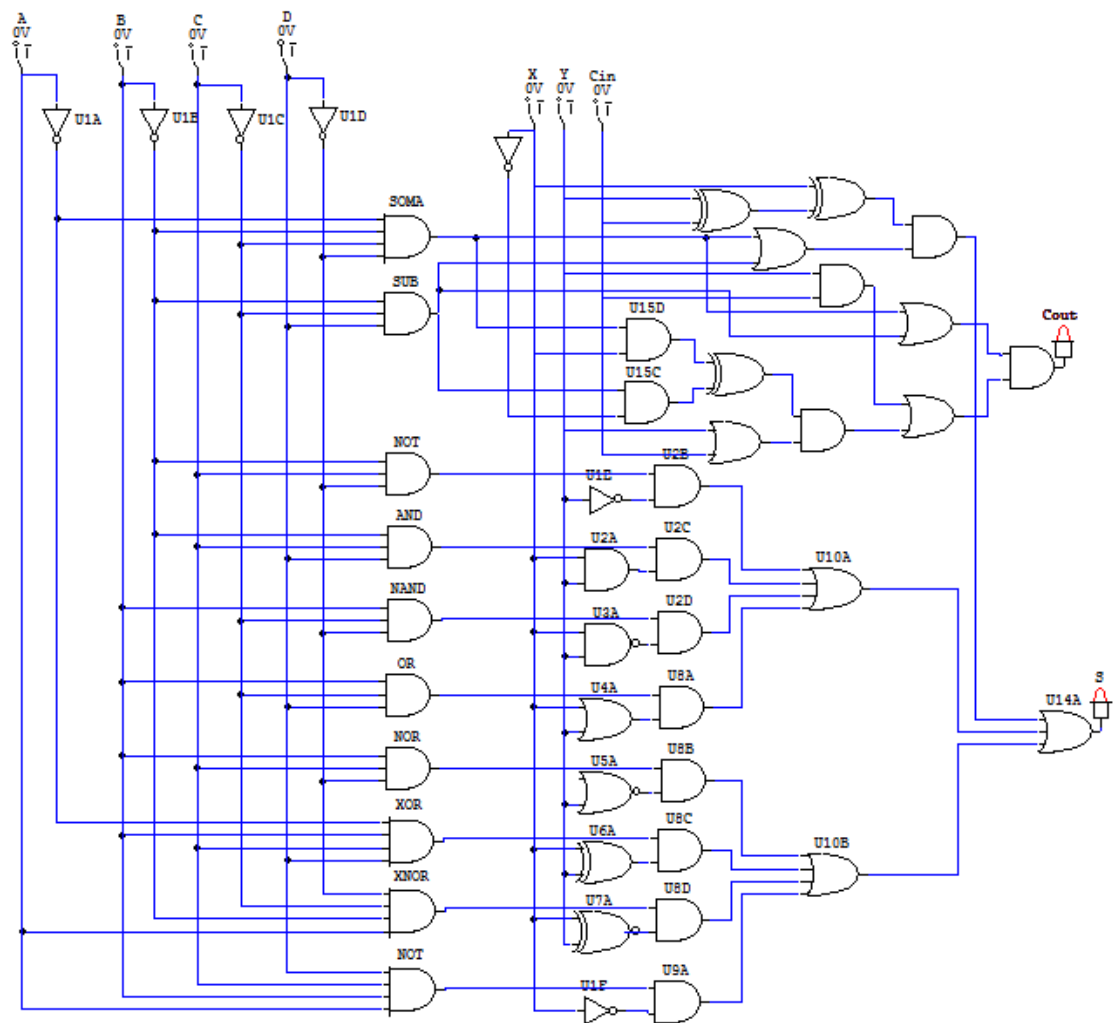


Figura 9 ULA completa

O funcionamento da ULA consiste basicamente em escolher uma entrada no decodificador e caso esta seja do bloco lógico inserir as entradas X e Y ou caso esta seja do bloco aritmético inserir também a entrada Cin.

Conclusão

O projeto cumpriu seu objetivo de analisar a fundo os circuitos combinacionais e também o funcionamento de uma Unidade Lógica e Aritmética, apresentando resultado satisfatório em sua execução.

Referências Bibliográficas

FLOYD, Thomas.; Sistemas Digitais: Fundamentos e Aplicações. 9º Ed, 2009, Bookman Companhia Editora.

GUNTZEL, José Luís.; Nascimento, Francisco Assis.; Introdução aos Sistemas Digitais. Volume 1, 2001.

IDOETA, I. V.; CAPUANO, F. F.; Elementos de Eletrônica Digital. 2006, Editora Érica Ltda.

STALLINGS, William.; Arquitetura e Organização de Computadores. 8º Ed, 2009, Editora Pearson.

VAHID, Frank.; Sistemas Digitais: Projeto, Otimização e HDLS. 1ªEd, 2008, Bookman Companhia Editora.