

**UNIVERSIDADE FEDERAL DE PERNAMBUCO, CENTRO DE INFORMÁTICA  
(CIN)**

**PROJETO DE SISTEMAS DIGITAIS (ULA E SOMADOR BCD).**

**ALMERINDO PEREIRA DA SILVA NETO - APSN**

**HILDERLAN HERBERT FERREIRA DA SILVA - HHFS**

**JOÃO GUILHERME PONTES DE PAIVA - JGPP**

**JOÃO LUIS FERREIRA LACERDA - JLFL**

**MARIA BEATRIZ CAMPELO BEZERRA - MBCB**

**RECIFE**

**2019**

**RELATÓRIO DESCRIPTIVO**

**CAMPO DA INVENÇÃO**

- [001] Este projeto está vinculado à disciplina de Sistemas Digitais([cin.ufpe.br/~if675](http://cin.ufpe.br/~if675)), e visa à aplicação prática de conceitos teóricos vistos em sala de aula. Dentre as especificações do projeto, estão: desenvolvimento de uma Unidade Lógica e Aritmética (ULA) acoplado a um decodificador binário para display de sete-segmentos e o Somador BCD.  
[002] O projeto contou com a participação de cinco (05) membros, que juntos, trabalharam para concluir os objetivos do mesmo.

## ANTERIORIDADES

[003]

Sumário

[004] 1 Introdução;

[005] 2 Visão Geral da Unidade Lógica e Aritmética (ULA);

[006] 2.1 Somador / Subtrator;

[007] 2.2 Complementador a 2;

[008] 2.3 Função lógica AND;

[009] 2.4 Função lógica XOR;

[010] 2.5 Operações de Comparação: igualdade, maior que, menor que.

[011] 3 Somador BCD ;

[012] 4 Conclusão;

## DESCRICAÇÃO GERAL

[013]

1

Introdução

[014] Neste trabalho acadêmico, relativo à disciplina de sistemas digitais, estão contidas as etapas de elaboração de uma Unidade Lógica e Aritmética(ULA) e um somador BCD. Para o desenvolvimento dos sistemas de hardware citados acima, seguiu-se uma filosofia de modularização e dividir para conquistar. Ou seja, iniciou-se o projeto a partir do estudo de cada unidade básica que compõe uma ULA ou um somador BCD, posto que a união de cada uma dessas partes funcionando corretamente formará o sistema final. Devido a este processo de desenvolvimento, ao longo deste trabalho foram utilizados os seguintes métodos e técnicas na seguinte sequência: reconhecimento do problema a partir do uso de tabelas verdade, formulação de equações com o auxílio de mapas de Karnaugh, elaboração de circuitos lógicos a partir do uso de portas lógicas e, por último, a fase de testes. Este processo inicia-se em uma unidade básica e é repetido até o momento em que se chega ao sistema maior (ULA e somador).

BCD completo). Além dessas técnicas, foram utilizadas, também, ferramentas computacionais, como o Quartus, que possibilitaram sua aplicação prática. Livros e anotações de aula serviram para o embasamento teórico do projeto. Este trabalho teve o seu objetivo concluído quando foi possível chegar a um protótipo em hardware que não apresentasse falhas. No entanto, isto só foi possível devido ao uso das metodologias acima apresentadas.

[015]

2 Visão Geral da Unidade

#### Logica e Aritmética (ULA):

[016] Desde o surgimento dos primeiros computadores, a função de realizar cálculos está entre uma das principais atribuições desta invenção.

A importância é tanta que muitas das operações aritméticas realizadas pelos computadores atuais são impossíveis de serem calculadas em tempo hábil por um ser humano. É devido a estas importância e necessidade que os computadores possuem uma unidade responsável pelas operações lógicas e aritméticas que são requisitadas à máquina. Este componente do computador é chamado de Unidade de Lógica e Aritmética, mais facilmente conhecido por sua abreviatura ULA.

[017] O processo de resolução de cálculos em uma ULA é feita a partir do recebimento de informações em sua forma binária. Essa informação é manipulada em circuitos lógicos e aritméticos que são formados a partir da combinação de portas lógicas e flip-flops.

[018] Neste projeto, ao implementar-se a ULA, fez-se a construção das seguintes partes que podem compor esta unidade:

[019] -Somador/Substrator;

[020] -Complementador a 2;

[021] -Função lógica AND;

[022] -Função lógica XOR;

[023] -Operações de Comparação: igualdade, maior que, menor que;

[024]

[025]

**Somador/Subtrator**

[026]

[027] Somador 1 Bit:

[028]

[029]

[030] Na "Figura 1" poderá ser visto um circuito implementado de um somador de um bit.

[031] Na "Figura 1.1" poderá ser visto o mapa de Karnaugh implementado de um somador de um bit de saída.

[032] Na "Figura 1.2" poderá ser visto o mapa de Karnaugh implementado de um somador de um bit COUT.

[033] Na "Figura 1.3" poderá ser visto a tabela verdade de um somador de um bit.

[034] Este circuito realiza a soma de apenas 1 bit, levando em consideração o carry in que indica se houve um overflow na soma do bit anterior. A entrada desse somador são 3 bits, um bit do vetor A, um bit do vetor B e um bit carry in e tem como saída um bit resultado da soma e um bit carry out.

[035]

[036] Essa função é uma parte de uma função maior, o somador de 4 bits.

[037]

[038] Somador/Subtrator 4 bits:

[039] Na "Figura 2" poderá ser visto um circuito implementado de um somador de 4 bits.

[040] Este circuito tem como finalidade somar ou subtrair dois vetores de 4 bits cada. Ele faz isso utilizando-se de 4 caixas do circuito Somador 1 Bit. Além disso, foi implementado uma terceira entrada além dos vetores A e B, esta entrada tem apenas 1 bit e serve para selecionar se haverá uma soma ou uma subtração. A saída dessa função é um vetor de 4 bits e um bit indicando se houve overflow.

[041]

[042]

[043]

Complementador a 2

[044]

[045] Complemento a 2 (1 bit):

[046] O circuito de um complemento a 2 de 1 bit, pode ser visualizado na "Figura 3".

[047] O Mapa de Karnaugh de um complemento a 2 de 1 bit de saída, pode ser visualizado na "Figura 3.1".

[048] A tabela verdade de um complemento a 2 de 1 bit, pode ser visualizado na "Figura 3.2".

[049]

[050] Este circuito tem como entrada umbit do vetor B e um bit carry in. Gera duas saídas: um carry out e um bit resultado do complemento a 2.

[051]

[052] Complemento a 2 (4 bits):

[053]

[054] O circuito de um complemento a 2 de 4 bits, pode ser visualizado na "Figura 4".

[055] Este circuito utiliza 4 caixas docircuito complemento a 2 (1 bit) e é responsável por fazer o complemento a 2 do vetor B. Tem como saída um vetor resultado da função complemento a 2 e um bitindicando se houve overflow.

[056]

### 2.3 Função

[057]  
lógica AND;

[058] And 4 Bits:

[059]

[060] Um exemplo do circuito implementado pode ser visto em "Figura 5".

[061] Este circuito irá utilizar a função AND entre os bits dos vetores de entrada A e B e gerará como saída o resultado da mesma.

## 2.4 Função

[062]

[063]

lógica XOR;

[064]

[065] Xor 4 Bits:

[066] Um exemplo do circuito implementado pode ser visto em "Figura 6".

[067]

[068] Este circuito irá utilizar a função XOR entre os bits dos vetores de entrada A e B e gerará como saída o resultado da mesma.

[069]

## 2.5 Operações de Comparação: igualdade,

maior que, menor que.

[070]

[071]

[072] Igualdade 1 Bit:

[073] Este circuito tem a finalidade de comparar dois bits e indicar se eles são iguais. Tem como entrada um bit do vetor A, um bit do vetor B e um bit carry in que indica se há igualdade até o bit anterior. Assim, gerando como saída um bit carry out, além do bit resultadoda comparação.

[074] Na "Figura 7" é possível ver o circuito correspondente à Igualdade de 1 bit.

[075] Na "Figura 7.1" é possível ver o Mapa de Karnaugh correspondente a Igualdade de 1 bit de saída.

[076] Na "Figura 7.2" é possível ver a tabela verdade correspondente a Igualdade de 1 bit.

[077]

Igualdade 4 Bits:

[078] Este circuito tem a função de comparar se dois vetores são iguais. As entradas são apenas os vetores A e B e gera um bit indicando se há ou não uma igualdade.

[079] Na "Figura 8" podemos visualizar o circuito correspondente.

[080]

[081] Maior Que 1 Bit:

[082] Esta função irá indicar se um bit do vetor A é maior que um bit do vetor B. Além dos dois vetores, é preciso mais um bit de entrada, sendo ele o carry in, que indica se há uma relação desuperioridade até o bit anterior. A saída é apenas o bit indicando o resultado da comparação.

[083] Circuito contido na "Figura 9".

[084]

[085] Mapa de Karnaugh contido na "Figura 9.1".

[086] Tabela verdade contida na "Figura 9.2".

[087]

[088] Maior Que 4 Bit:

[089] Esta função irá comparar dois vetores e dirá se o primeiro vetor é numericamente maior que o segundo. Tem como entrada os vetores A e B e, como saída, um bit resultado da comparação em questão.

[090] Este circuito utiliza-se de 3 caixas do circuito Maior Que 1 Bit para realizar a comparação dos 3 primeiros bits e, para o último bit, há a necessidade de checar se os números representados pelos vetores são positivos ou negativos e compará-los a partir disso.

[091]

[092] Pode-se verificar o circuito de "Maior que 4 bits" na "Figura 10".

[093] Menor Que 1 Bit:

[094] Este circuito indica se um bit do vetor A é menor que um bit do vetor B. Além dos bits dos vetores em questão, também existe como entrada um bit carry in, indicando se há uma relação de inferioridade até o bit anterior.

[095] O circuito de "Menor que 1 bit", pode ser visualizado na "Figura 11".

[096] O Mapa de Karnaugh de "Menor que 1 bit", pode ser visualizado na "Figura 11.1".

[097] A tabela verdade de "Menor que 1 bit", pode ser visualizado na "Figura

11.2<sup>o</sup>

[098]

Menor Que 4 Bits:

[099] Este circuito irá comparar doisvetores e dirá se o primeiro vetor é numericamente menor que o segundo. Tem como entrada os vetores A e B e, como saída, um bit resultado da comparação em questão.

[100] Este circuito utiliza-se de 3caixas do circuito Menor Que 1 Bit para realizar a comparação dos 3 primeirosbits e, para o último bit, há a necessidade de checar se os númerosrepresentados pelos vetores são positivos ou negativos e compará-los a partir disso.

[101] Pode-se verificar o circuito na "Figura 12".

[102] 3 Visão Geral do

Somador BCD

[103] O Somador BCD é uma dos métodos utilizados para somar númerosbinários na forma BCD (Binary Coded Decimal), em que as entradas variam de 00 a 99. Como estas são muito grandes para serem resolvidas de uma únicavez, é preciso "quebrar" esse problema em outros menores (modularização). Em tal módulo, as entradas variam de 0 a 9. Portanto, para explicar ecaracterizar o Somador BCD com entradas variando de 00 a 99 é precisoexplicar primeiro o funcionamento dos seus módulos menores.

[104] Pode-se visualizar o circuito de um Somador BCD na "Figura 13"

[105]

4 Conclusão

Na implementação deste projeto, utilizou-se a ferramenta de design dedispositivos lógicos programáveis Quartus II. Tal software teve extremaiimportância na modelagem dos circuitos utilizados para construção dossistemas pedidos. A construção de uma ULA, Unidade Lógica e Aritmética, é um objetivodemasiado grande para ser solucionado sem que haja uma divisão do sistemaem partes menores. Tal divisão é chamada de modularização e se trata daresolução do problema em módulos menores.

seguida da junção dessas soluções parciais, de modo a solucionar o problema como um todo. A construção do Somador BCD (Binary Coded Decimal), além de ter mostrado extremamente válida, instigante e recompensadora do ponto de vista do aprendizado do assunto da disciplina de Sistemas Digitais, se mostrou também uma ótima oportunidade de conhecer mais sobre essa forma de representação e codificação de números. Esse projeto é de grande importância no entendimento de sistemas mais elaborados, tais como a CPU, Central Processing Unit, visto que introduz unidades mais simples, como comparadores, somadores/substratores e multiplexadores. Tais módulos servem de base para a confecção de circuitos mais complexos. Todo conhecimento adquirido no projeto será certamente utilizado, não só na disciplina de Sistemas Digitais, como em todo o decorrer do curso de Ciência da Computação.

[106]

[107]

#### MODALIDADES PREFERIDAS

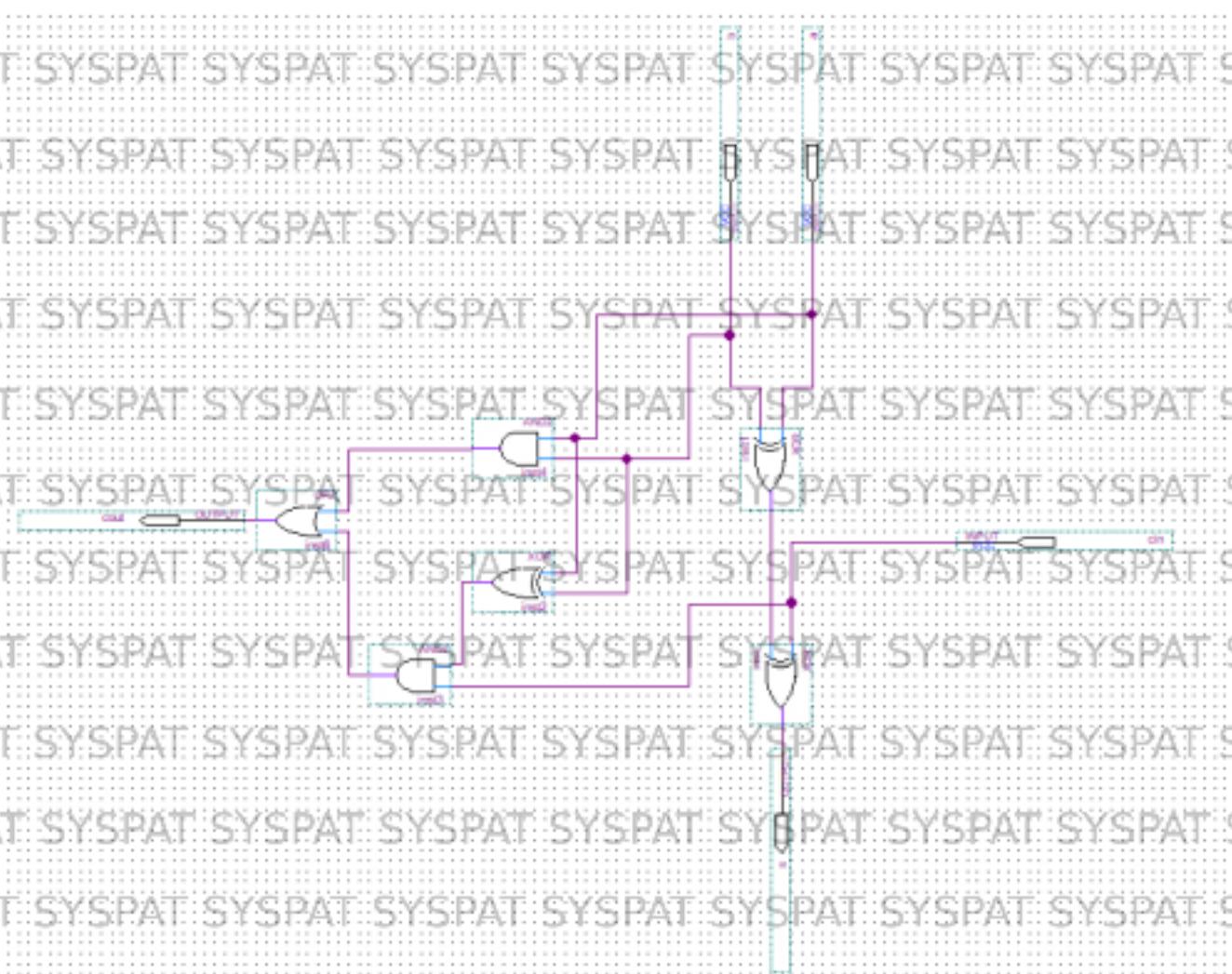
[108] A Unidade Lógica e Aritmética ( ULA ) serve para executar as principais operações lógicas e aritméticas do computador. Ela pode somar, subtrair, determinar se o número é positivo ou não.

[109] O somador BCD é capaz de representar números decimais acima de 9, apenas com número de 0 a 9 ou 0000 a 1001, caso seja em binário.

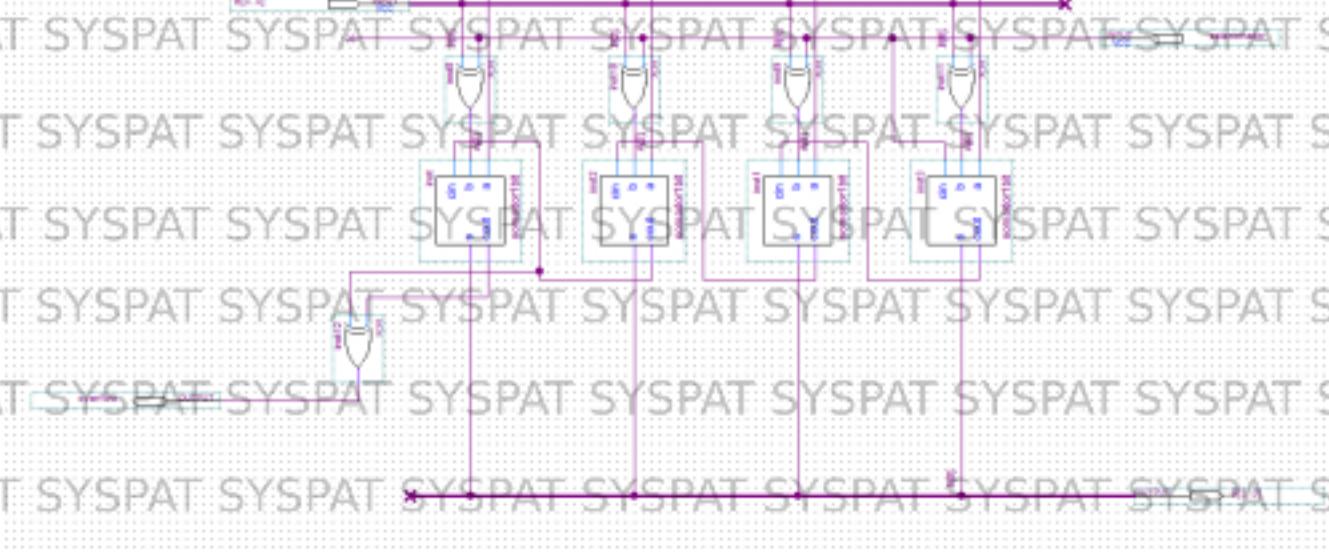
**REMINDICAÇÕES**

1. Unidade lógica e aritmética de 4 bits caracterizada por implementar 8 funções em um único componente.

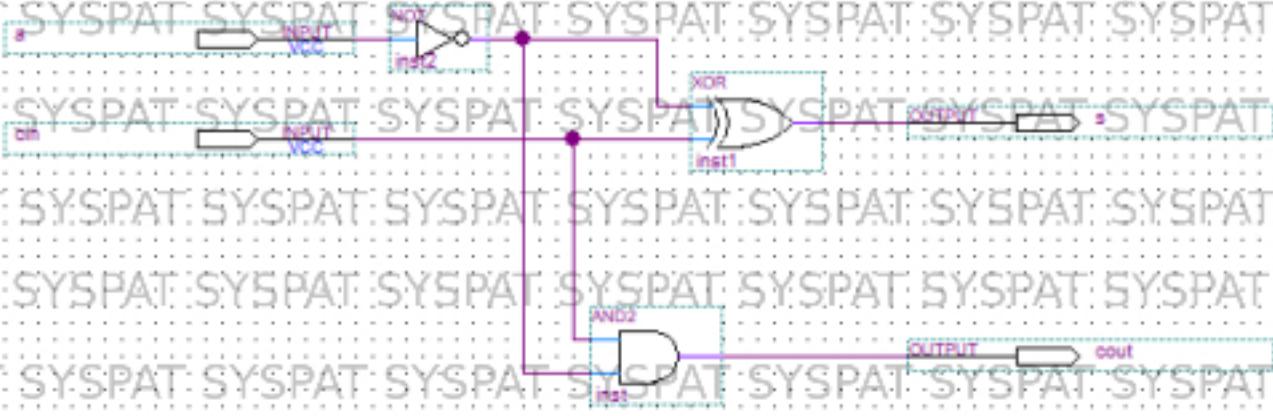
2. O módulo de adição de 4 bits, implementado na ULA do item 1 é caracterizado por gerar além da soma, o sinal overflow.



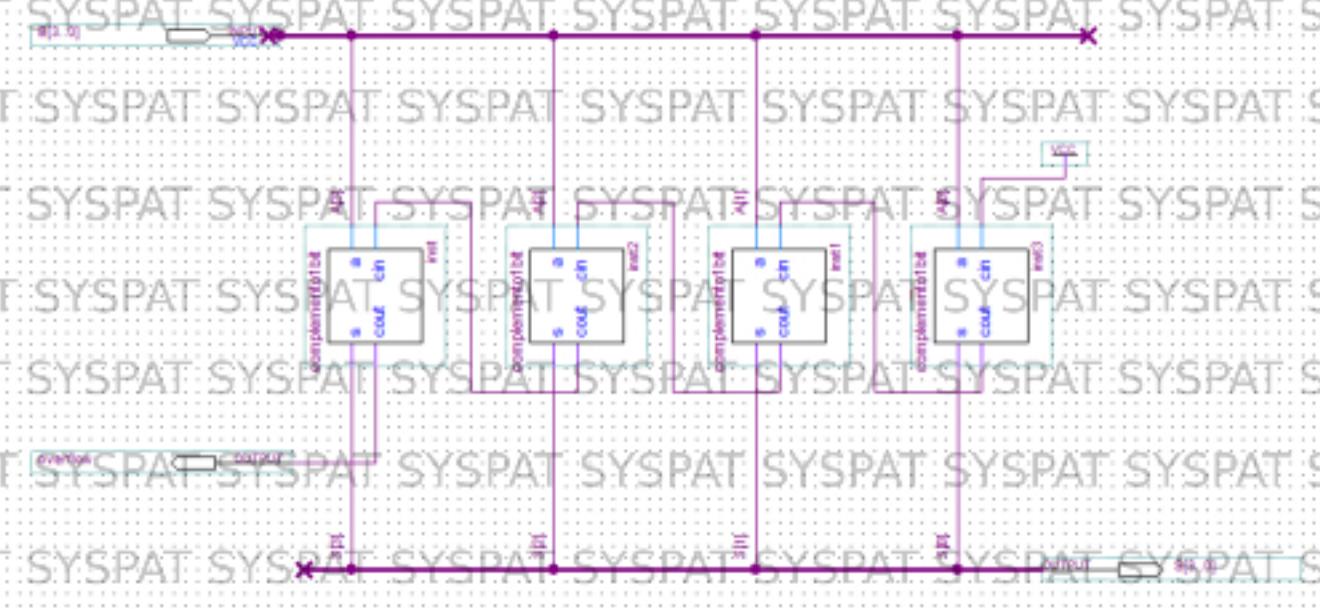
**Figura 1 ( somador 1 bit )**



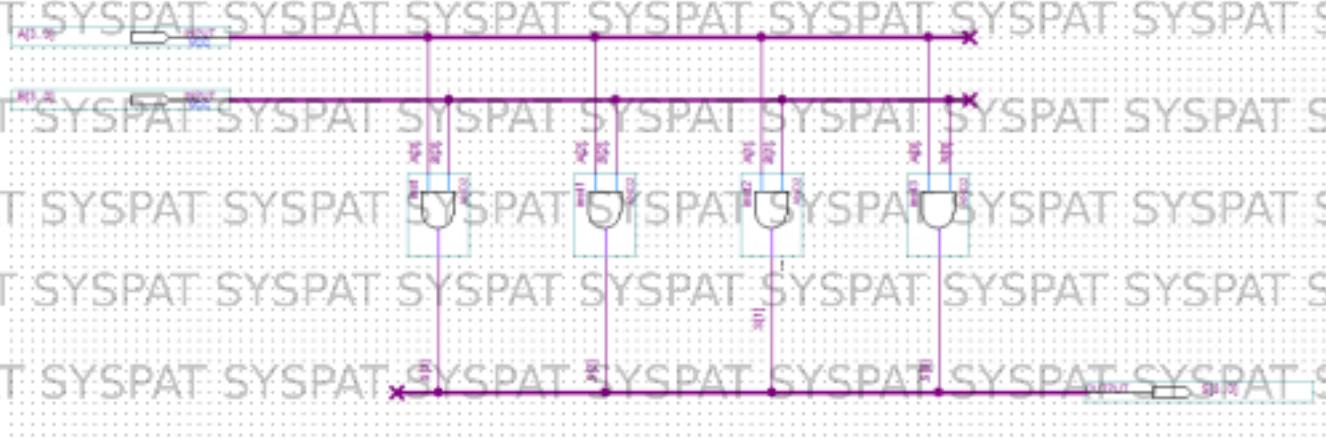
**Figura 2 (Somador/Substrator 4 bits)**



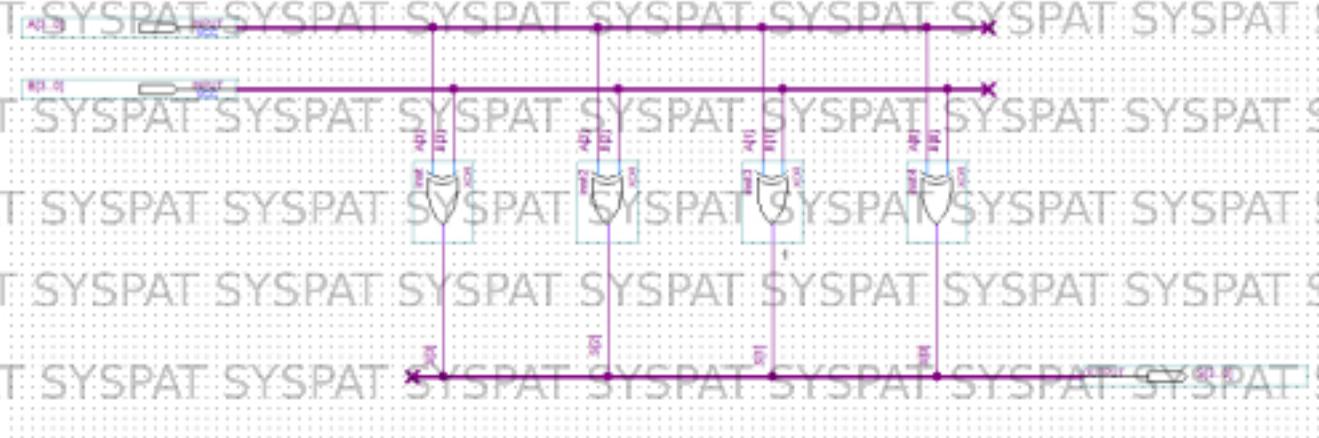
**Figura 3 (Complemento a 2 de 1 bit)**



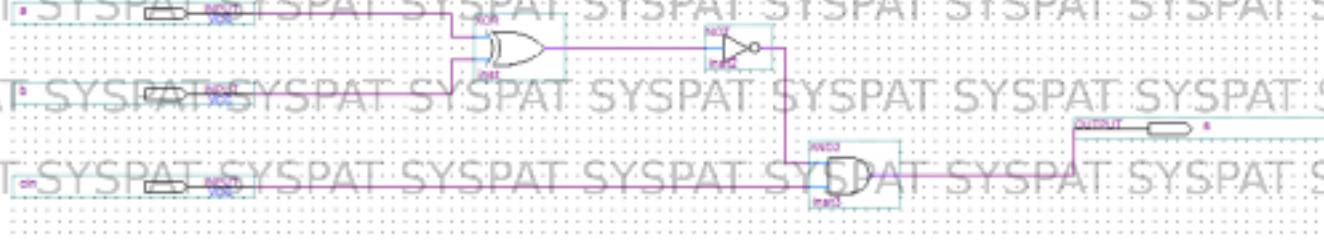
**Figura 4 ( Complemento a 2 de 4 bit)**



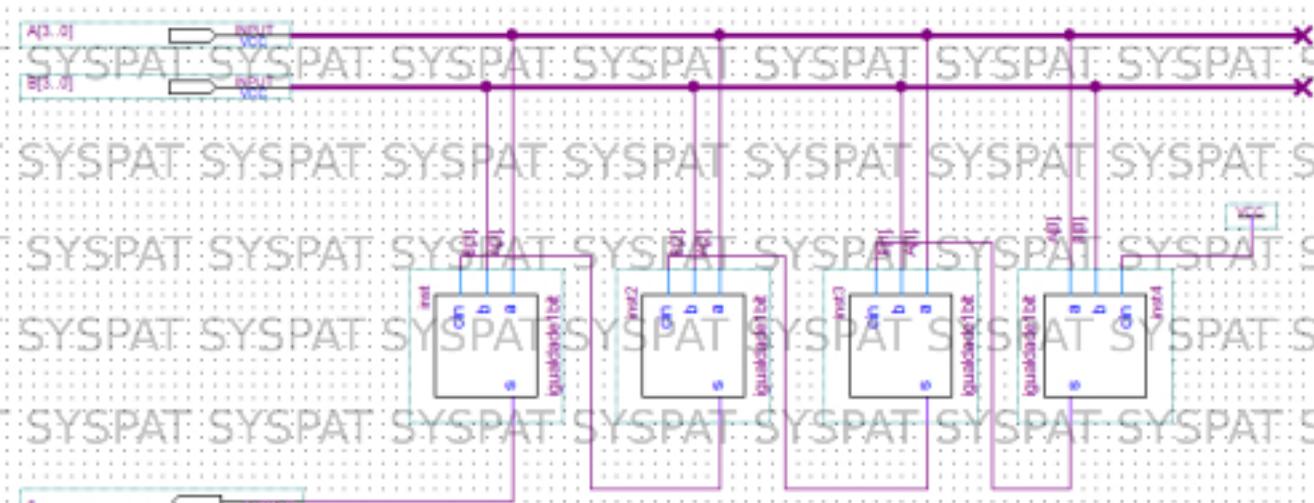
**Figura 5 ( Função AND )**



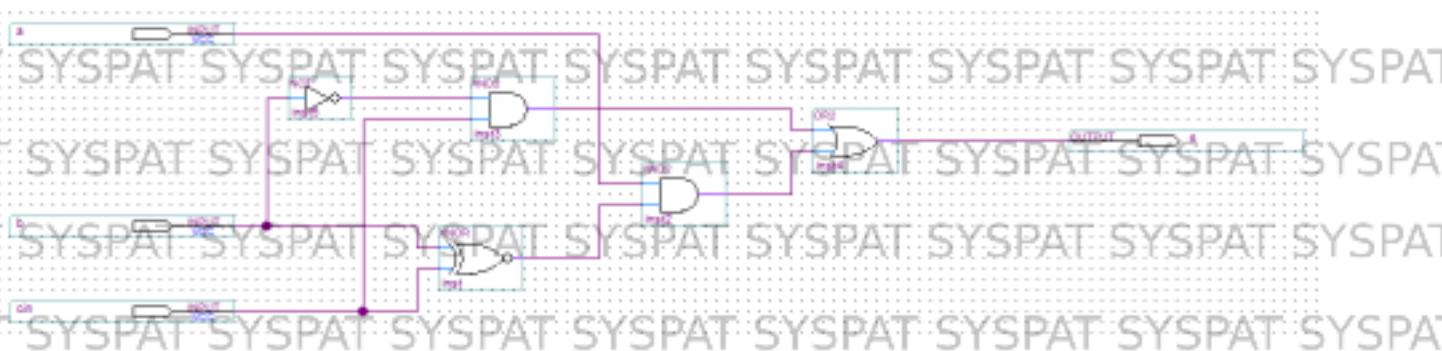
**Figura 6 ( Função XOR )**



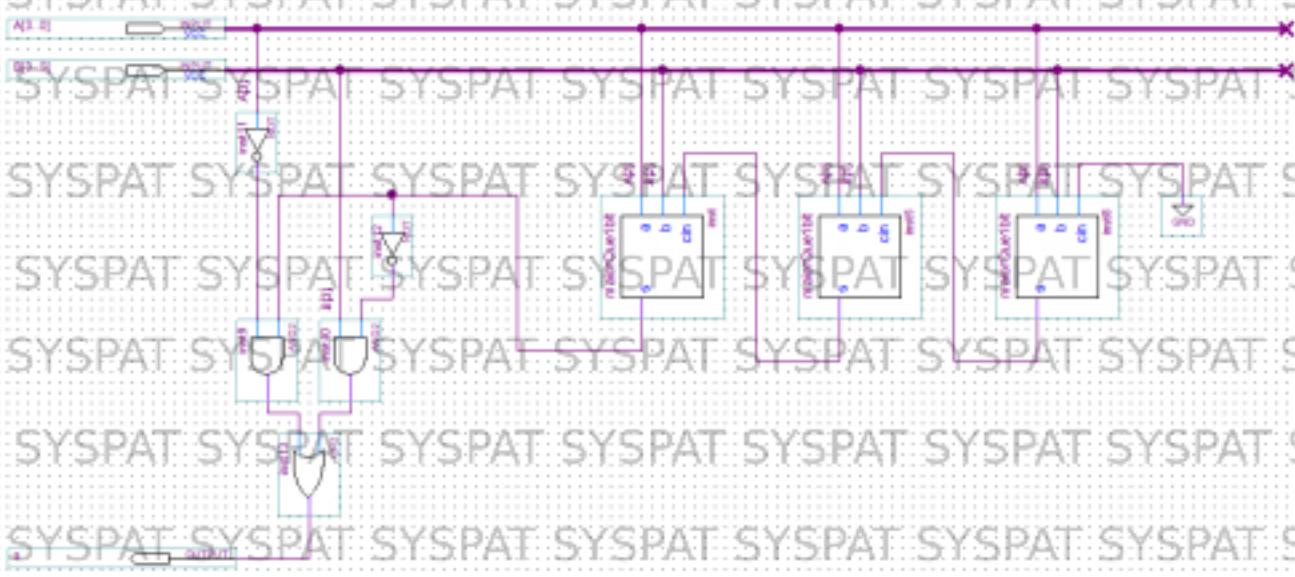
**Figura 7 (Igualdade de 1 bit)**



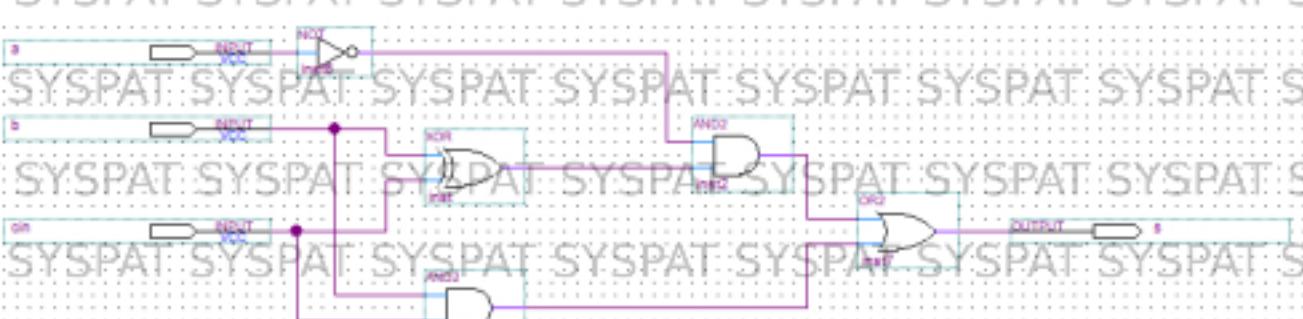
**Figura 8 ( Igualdade de 4 bits )**



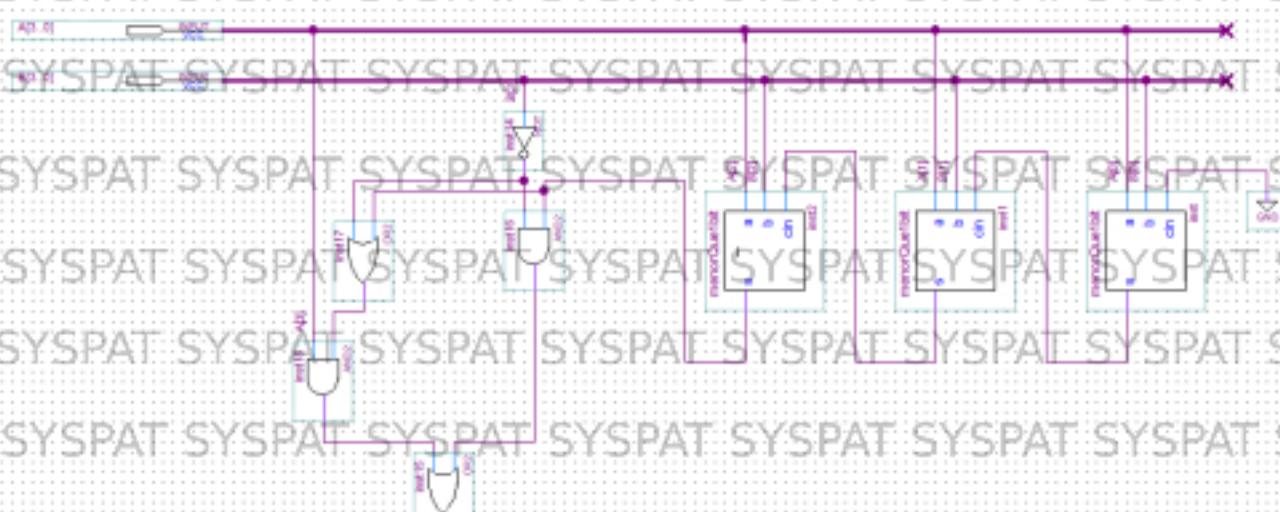
**Figura 9 ( Mayor que 1 bit )**



**Figura 10 ( Mayor que 4 bits )**



**Figura 11 ( Menor que 1 bit )**



**Figura 12 ( Memoria 4 bits )**

<b>cin/ab</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
0	0	1	0	1
1	1	0	1	0

**Figura 1.1 (KARNAUGH somador de 1 bit de saída )**

<b>cin/ab</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>

**Figura 1.2 ( KARNAUGH somador de 1 bit COUT )**

<b>cin/ab</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>

**Figura 3.1 ( KARNAUGH complemento a 2 de 1 bit de saída )**

<b>cin/ab</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
0	0	0	0	0
1	1	0	1	0

**Figura 7.1 ( KARNAUGH igualde de 1 bit de saída )**

<b>cin/ab</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>

**Figura 9.1 ( KARNAUGH maior que 1 bit de saída )**

<b>cin/ab</b>	<b>00</b>	<b>01</b>	<b>11</b>	<b>10</b>
0	0	1	0	0
1	1	1	1	0

**Figura 11.1 ( KARNAUGH menor que 1 bit de saída )**

<b>a</b>	<b>b</b>	<b>cin</b>	<b>s</b>	<b>cout</b>
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

Figura 1.3 ( Tabela Verdade somador de 1 bit )

	<b>a</b>	<b>b</b>	<b>cin</b>	<b>s</b>
	0	0	0	0
	0	0	1	1
	0	1	0	0
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	1

**Figura 3.2 ( Tabela Verdade complemento a 2 de 1 bit )**

<b>a</b>	<b>b</b>	<b>cin</b>	<b>s</b>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

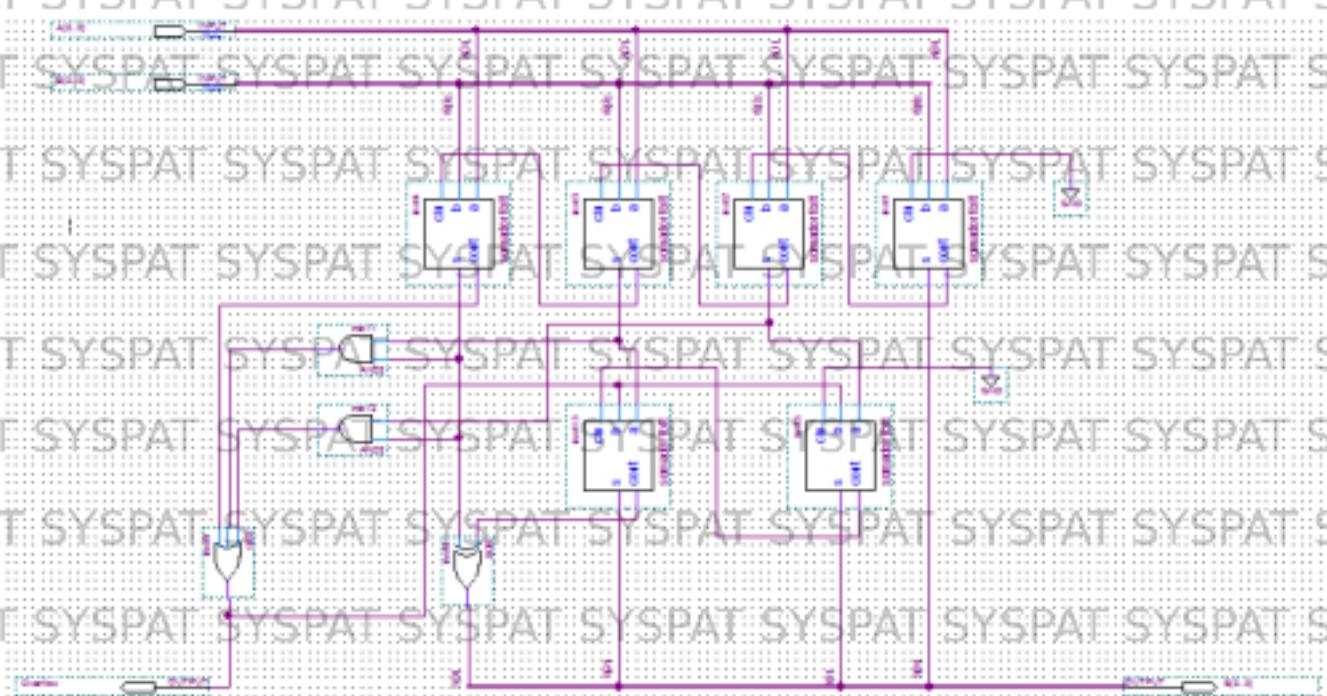
**Figura 7.2 ( Tabela Verdade Igualdade de 1 bit )**

<b>a</b>	<b>b</b>	<b>cin</b>	<b>s</b>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

**Figura 9.2 ( Tabela Verdade maior que 1 bit )**

<b>a</b>	<b>b</b>	<b>cin</b>	<b>s</b>
0	0	0	0
0	0	1	1
0	1	0	1
1	0	1	1
1	0	0	0
1	1	1	0
1	1	0	0
1	1	1	1

**Figura 11.2 ( Tabela Verdade menor que 1 bit )**



**Figura 13 ( Circuito de um Somador BCD )**

**UNIVERSIDADE FEDERAL DE PERNAMBUCO, CENTRO DE INFORMÁTICA  
(CIN)**

**PROJETO DE SISTEMAS DIGITAIS (ULA E SOMADOR BCD).**

**ALMERINDO PEREIRA DA SILVA NETO - APSN**

**HILDERLAN HERBERT FERREIRA DA SILVA - HHFS**

**JOÃO GUILHERME PONTES DE PAIVA - JGPP**

**JOÃO LUIS FERREIRA LACERDA - JLFL**

**MARIA BEATRIZ CAMPELO BEZERRA - MBCB**

**RECIFE**

**2019**

**RESUMO**

O projeto refere-se a implementação e explicação a cerca da Unidade de Lógica Aritmética ( ULA ) e do Somador BCD, visando o empenho e o aprendizado no assunto, dos membros do grupo. Após o projeto, todos puderam entender como funciona e como implementar a ULA e o Somador BCD.