

**UNIDADE LÓGICA E ARITMÉTICA INTEIRA PARA PROCESSAMENTO DE 4 BITS****DESCRIPTION****FIELD OF INVENTION**

1 A unidade lógica e aritmética implementa as operações de XOR, AND, Soma, Subtração, Igual, Complemento a 2,  $>$ ,  $<$ .

Cada função é escolhida mediante uma entrada de seleção. A saída é representada por vetores numeros inteiros, bit de overflow e saída relacional, ver figura 1.

**BACKGROUND ART**

5 A seguinte invenção é projetada para ser utilizada em pequenos sistemas, e diferenciase das demais por ter também o complemento a 2 e dois displays de 7 bits como resultado.

A invenção de número de aplicação 7395294 e intitulada "Arithmetic logic unit" e publicada na data de 19/11/2013 é uma ULA capaz de realizar eficientemente operações 10 lógicas e aritméticas.

A invenção de numero de aplicação 1410084 e intitulada "ARITHMETIC LOGIC UNIT" e publicada na data de 15/10/1975 é uma ULA capaz de realizar separadamente operações de natureza lógica ou aritmética.

**GENERAL DESCRIPTION**

A ULA é composta por 8 blocos, a seguir:

**15 1. Somador**

O bloco Somador (ver figura 1) tem dois vetores de entrada A e B, ambos de 4 bits, e de saída, um vetor S de 4 bits e um bit overflow. A soma é feita por 4 blocos Adder (ver figura 2), que fazem soma bit a bit. Caso haja overflow na soma bit a bit, é passado para a soma adiante um bit de Cin, por meio do Cout do bloco Adder. A resposta da soma total será representada pelo vetor S, e caso a soma entre os dois vetores resulte num overflow, isto é, seja maior que 4 bits, o bit de overflow retorna 1.

Nas figuras 3, 4, 5 e 6, temos respectivamente o waveform do somador, a tabela-verdade para o bloco Adder e os Mapas de Karnaughs do bloco Adder.

**25 2. Subtrator**

O bloco Subtrator (ver figura 7) tem dois vetores de entrada A e B, ambos de 4 bits, e de saída, um vetor S de 4 bits e um bit overflow. A subtração é feita pela soma do vetor

1 A com o complemento a 2 (feito pelo bloco Complemento a 2, de figura 9) do vetor B.

Caso haja overflow na soma bit a bit, é passado para a soma adiante um bit de Cin, por meio do Cout do bloco Adder. A resposta da subtração será representada pelo vetor S, e caso a subtração entre os dois vetores resulte num overflow, isto é, seja maior que 4 bits, o bit de overflow retorna 1.

Na figura 8, temos o waveform do subtrator.

3. O bloco Complemento a 2 (ver figura 9) tem como entrada um vetor de 4 bits B, e um vetor de 4 bits S além de um bit de overflow como saída. O complemento a 2 é realizado ao somar o inverso (not) de cada bit do vetor B com um vetor [0001], e a soma é feita por meio de 4 blocos Adder, bit a bit.

Nas figuras 10 e 11, temos, respectivamente, o waveform do Complemento a 2 e a tabela verdade do bloco Not.

15 4. O bloco Igual (ver figura 12) tem dois vetores de entrada A e B de 4 bits, e de saída, um bit de resposta R. A operação de igualdade é feita inicialmente bit a bit, por meio de um bloco XOR conectado a um bloco NOT, e depois conectados a um AND de 4 input, onde seu output é R.

Nas figuras 13, 14 e 15, temos, respectivamente, o waveform da operação igual, a 20 tabela-verdade e o Mapa de Karnaugh das operações bit a bit.

5. O bloco maiorQue (ver figura 15) tem dois vetores de entrada A e B de 4 bits, e de saída, um bit de resposta R. A operação de maiorQue é feita, nos bits menos significativos, pelo bloco maiorQue\_Bitabitnão significativo (ver figura 17), que recebe os 25 bits correspondentes de cada vetor além de um bit de Cin como input, e retorna um Cout que é usado como Cin para as próximas operações. Ja a comparação dos bits significativos é feita pelo bloco maiorQue\_Bitabit significativo (ver figura 20), que retorna o bit de resposta R.

Em seguida as figuras 16, 22, 21, 19 e 18 representam, respectivamente, o waveform do bloco maiorQue, a tabela-verdade e o mapa de Karnaugh do maiorQue\_Bitabit significativo e a tabela-verdade e o mapa de Karnaugh do maiorQue\_Bitabit meno significativo.

6. O bloco menorQue (ver figura 23) tem dois vetores de entrada A e B de 4 bits, e de saída, um bit de resposta R. A operação de menorQue utiliza o bloco maiorQue (ver figura 15) ligado a um NOT, e o bloco Igual (ver figura 12), ambos são ligados a um bloco XOR, que retorna o bit de resposta R.

1 Segue abaixo, respectivamente, as figuras 24, 26, 25, 28 e 27 o waveform do bloco menorQue, a tabela-verdade e o mapa de Karnaugh da comparação bit a bit do menorQue para bits menos significativos e a tabela-verdade e o mapa de Karnaugh da comparação bit a bit do menorQue para bits mais significativos.

5

7. O bloco AND (ver figura 29) tem dois vetores de entrada A e B de 4 bits, e de saída, um vetor S de 4 bits. A operação AND retorna, em cada bit de S, um AND entre os bits correspondentes em A e B.

Segue abaixo, respectivamente, as figuras 30, 32 e 31 correspondentes ao waveform do bloco AND, a tabela-verdade e o mapa de Karnaugh do bloco AND para 1 bit.

8. O bloco XOR (ver figura 33) tem dois vetores de entrada A e B de 4 bits, e de saída, um vetor S de 4 bits. A operação XOR retorna, em cada bit de S, um AND para 1 bit

15 entre os bits correspondentes em A e B.

Segue abaixo, respectivamente, as figuras 34, 36 e 35 correspondentes ao waveform do bloco XOR, a tabela-verdade e o mapa de Karnaugh do bloco XOR para 1 bit.

Além destas operações, a ULA faz a seleção de qual operação será realizada por meio de um MUX de 8:1, que foi feito separadamente por meio de um MUX de 5:1 ( ver figura 37 ) e de 3:1 ( ver figura 40 ).

Segue, respectivamente, imagens das tabelas-verdade dos MUX, além do circuito de um dos blocos usados no MUX, de número 39, 41 e 38.

25 A ULA também possui dois displays representados por 7 bits, onde um deles representa o símbolo de negativo ( ver figura 43 ) e o outro representa os números de 0 até 8 ( ver figura 42 ).

Segue, respectivamente, imagens da tabela-verdade e dos Mapas de Karnaugh dos displays, de número 53, 45, 46, 47, 48, 49, 50, 51, 52.

30

## PREFERRED EMBODIMENTS

A Unidade Lógica e Aritmética é uma implementação que permite ser acoplada a um microcontrolador digital para processamento de dados, voltado para conversão de sinal Digital para Analógico.

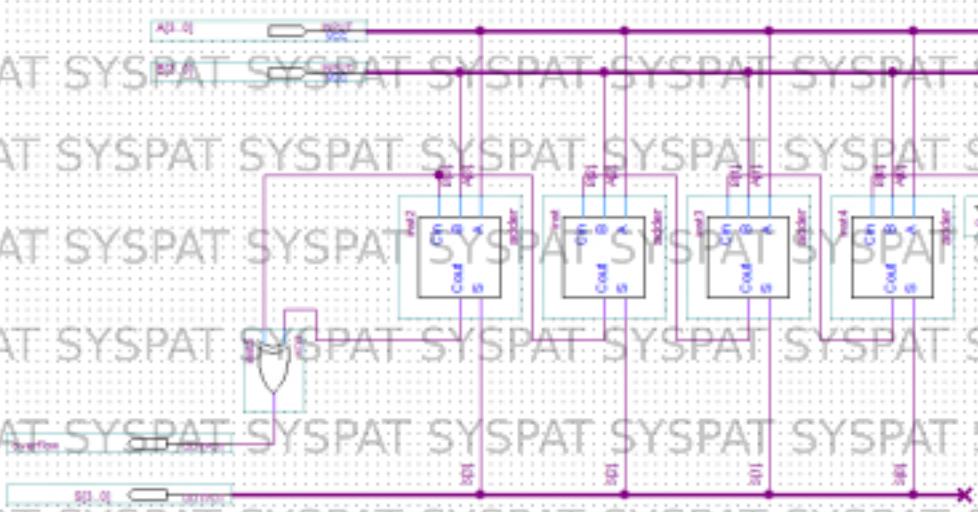
O sistema pode ser utilizada em pequenos sistemas digitais, como Raspberry PI e Arduinos para executar tarefas lógicas e aritméticas.

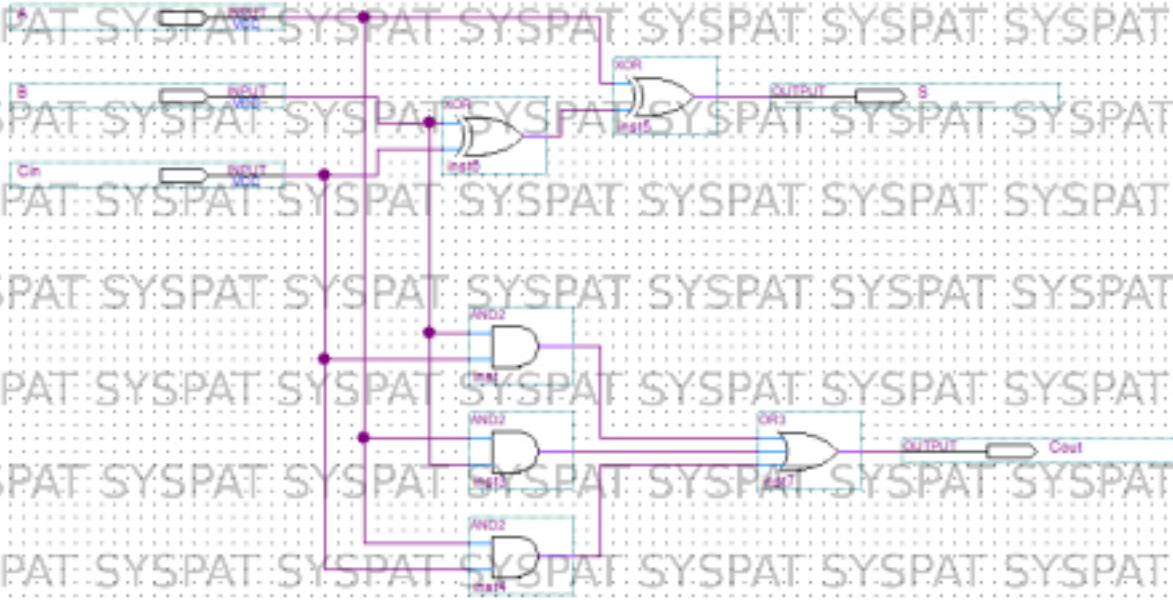
**CLAIMS**

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

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

O módulo de menorQue de 4 bits é feito a partir do bloco maiorQue de 4 bits.

**DRAWINGS****Bloco Somaador**

**Bloco Adder**



## Tabela-Verdade do bloco Adder

<b>Cin</b>	<b>A</b>	<b>B</b>	<b>Cout</b>	<b>Soma</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

Tabela Verdade do Bloco Adder

### Mapa de Karnaugh - Adder - Bit Cout

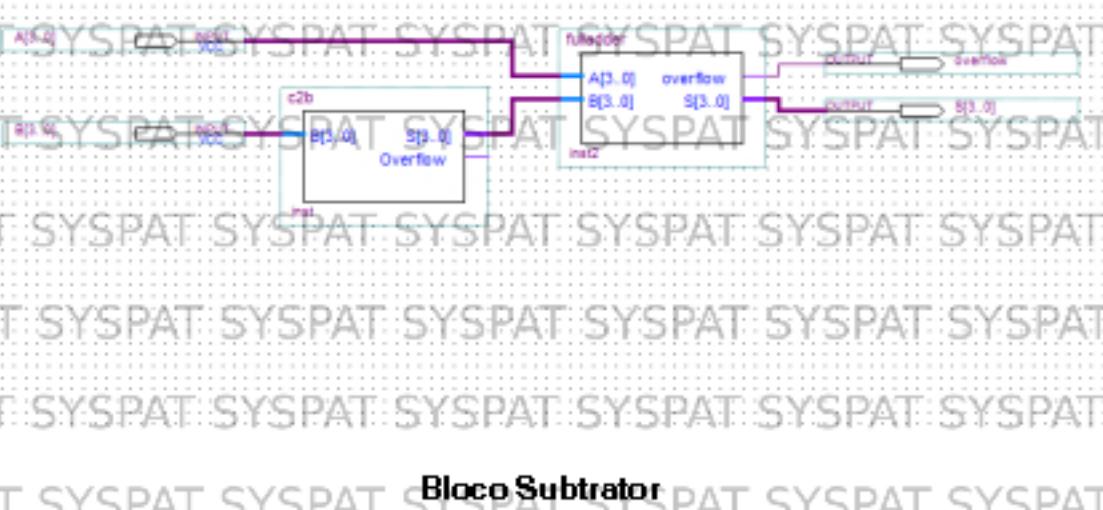
$\neg A \cdot \neg B$	$\neg A \cdot B$	$A \cdot B$	$A \cdot \neg B$	
$\neg Cin$	0	1	0	1
Cin	1	0	1	0

Mapa de Karnaugh do Bit Cout do Bloco Adder

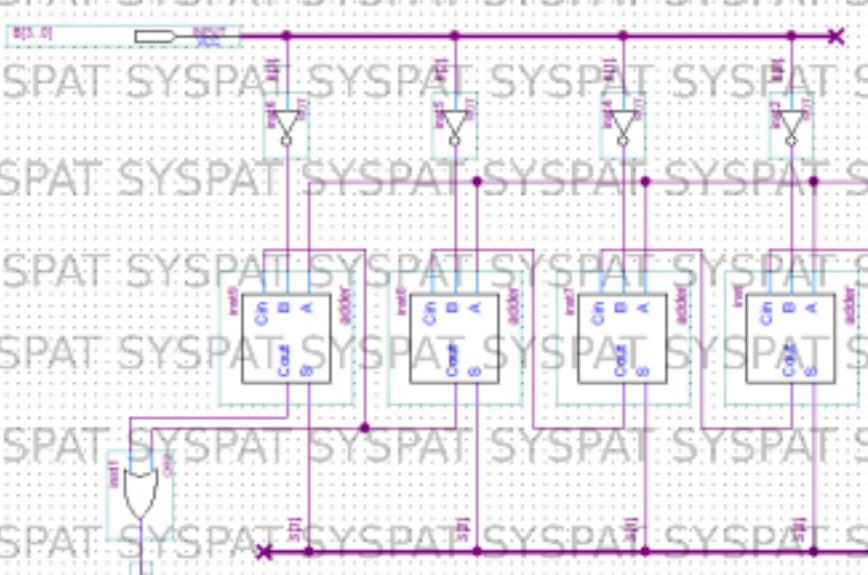
### Mapa de Karnaugh - Adder - Bit Soma

$\neg A \cdot \neg B$	$\neg A \cdot B$	$A \cdot B$	$A \cdot \neg B$	
$\neg Cin$	0	0	1	0
Cin	0	1	1	1

Mapa de Karnaugh do Bit Soma do Bloco Adder





**Bloco Complemento a 2**

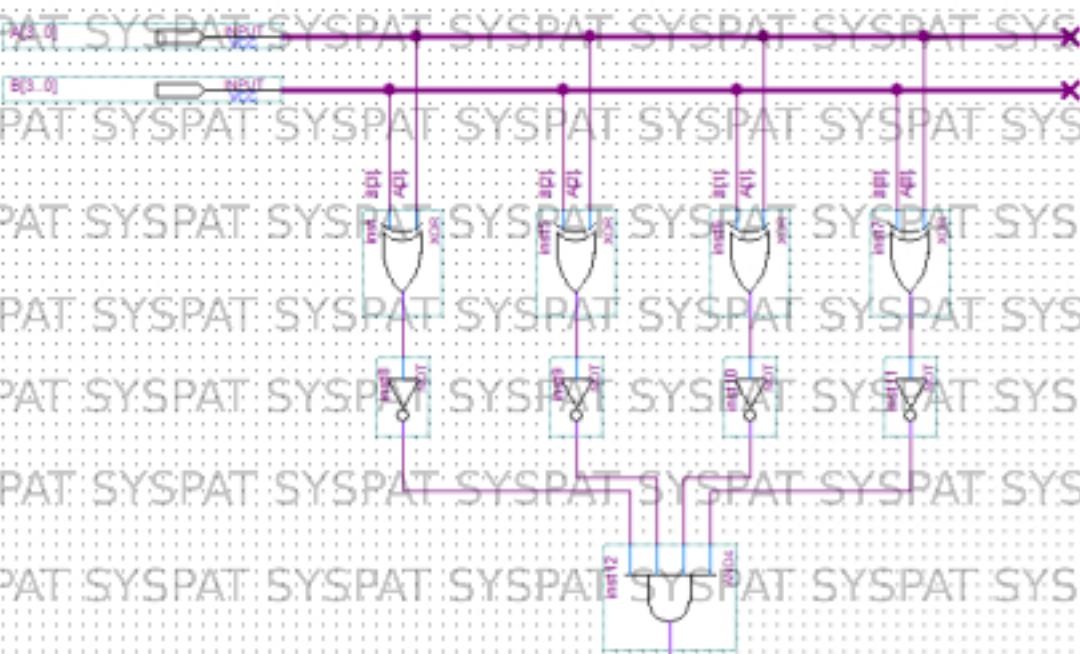


**Waveform do Bloco Complemento a 2**

## Tabela-Verdade do bloco NOT

B	Cout
0	1
1	0

Tabela Verdade do Bloco Not



**Bloco Igual**

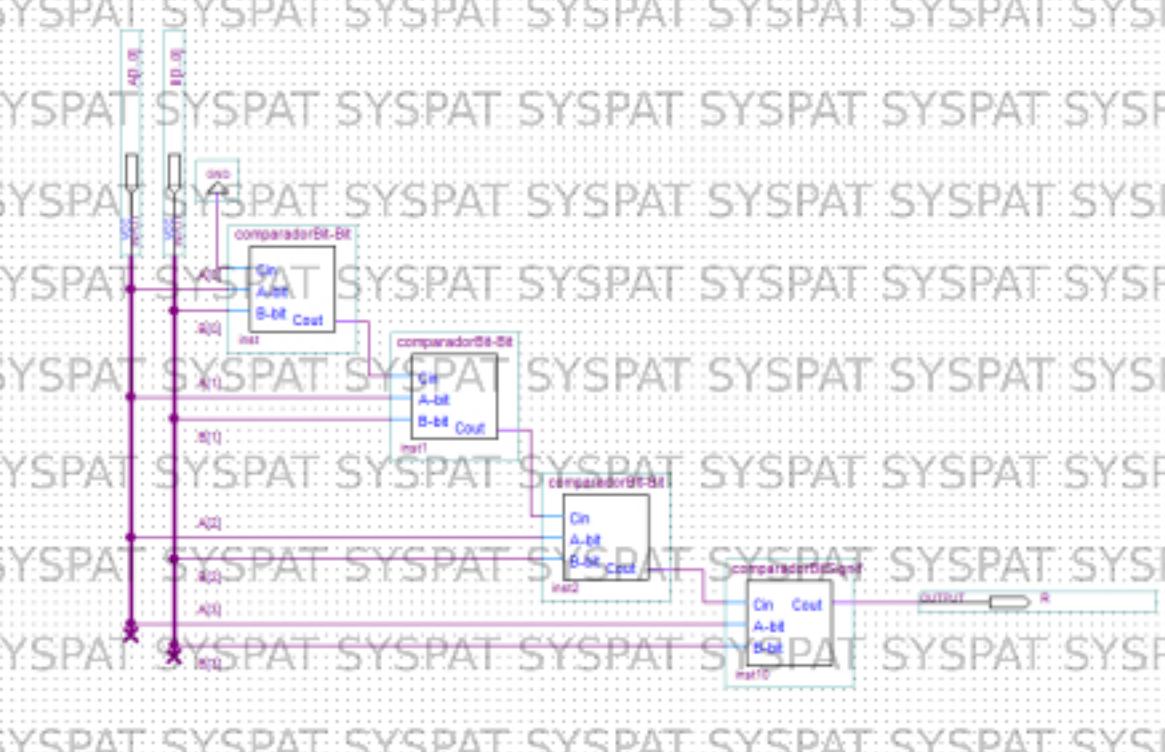


Waveform do Bloco Igual

## Mapa de Karnaugh - Igual

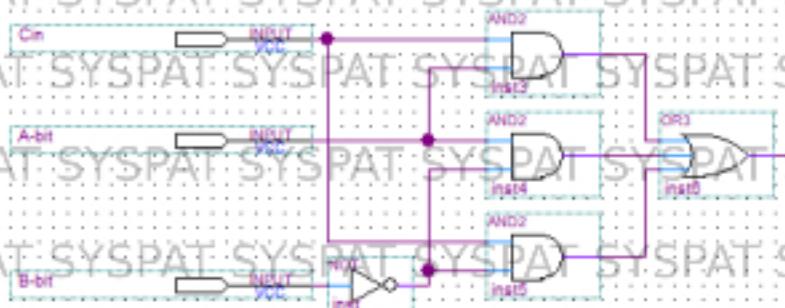
	A	A
$\neg A$	1	0
$\neg B$	0	1
B	0	1

Mapa de Karnaugh do Bloco Igual





**Waveform do Bloco Maior Que**



**Bloco Maior Que para Bits Menos Significativos**

**Mapa de Karnaugh - Maior Que - Bit Menos Significante.**

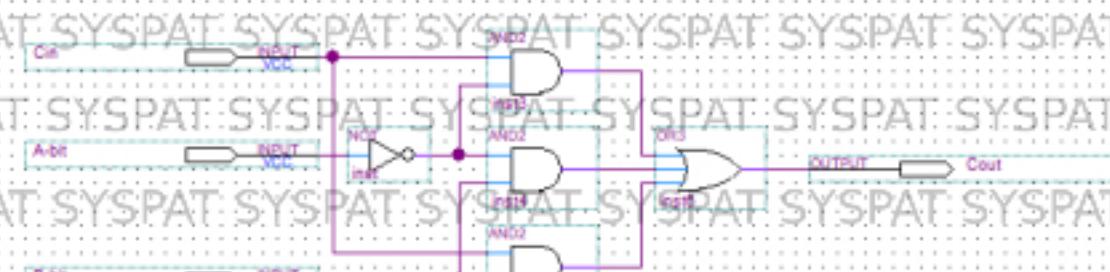
	$\neg A \cdot \neg B$	$\neg A \cdot B$	$A \cdot B$	$A \cdot \neg B$
$\neg Cin$	0	0	0	1
$Cin$	1	0	1	1

**Mapa de Karnaugh para o Maior Que dos Bits Menos Significativos**

**Tabela-Verdade do bloco Maior Que dos Bits Menos Significativos**

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

**Tabela Verdade do Bloco Maior Que para Bits Menos Significativos**



### **Bloco Maior Que para o Bit Mais Significativo**

## **Mapa de Karnaugh - Maior Que - Bit + Signif.**

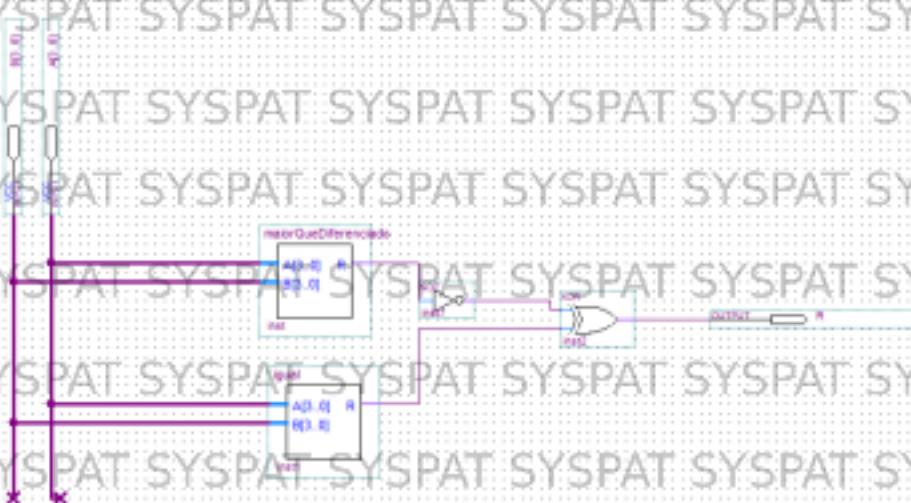
$\neg A \cdot \neg B$	$\neg A \cdot B$	$A \cdot B$	$A \cdot \neg B$
$\neg Cin$	0	1	0
Cin	1	1	1

**Mapa de Karnaugh para o Maior Que do Bit Mais Significativo**

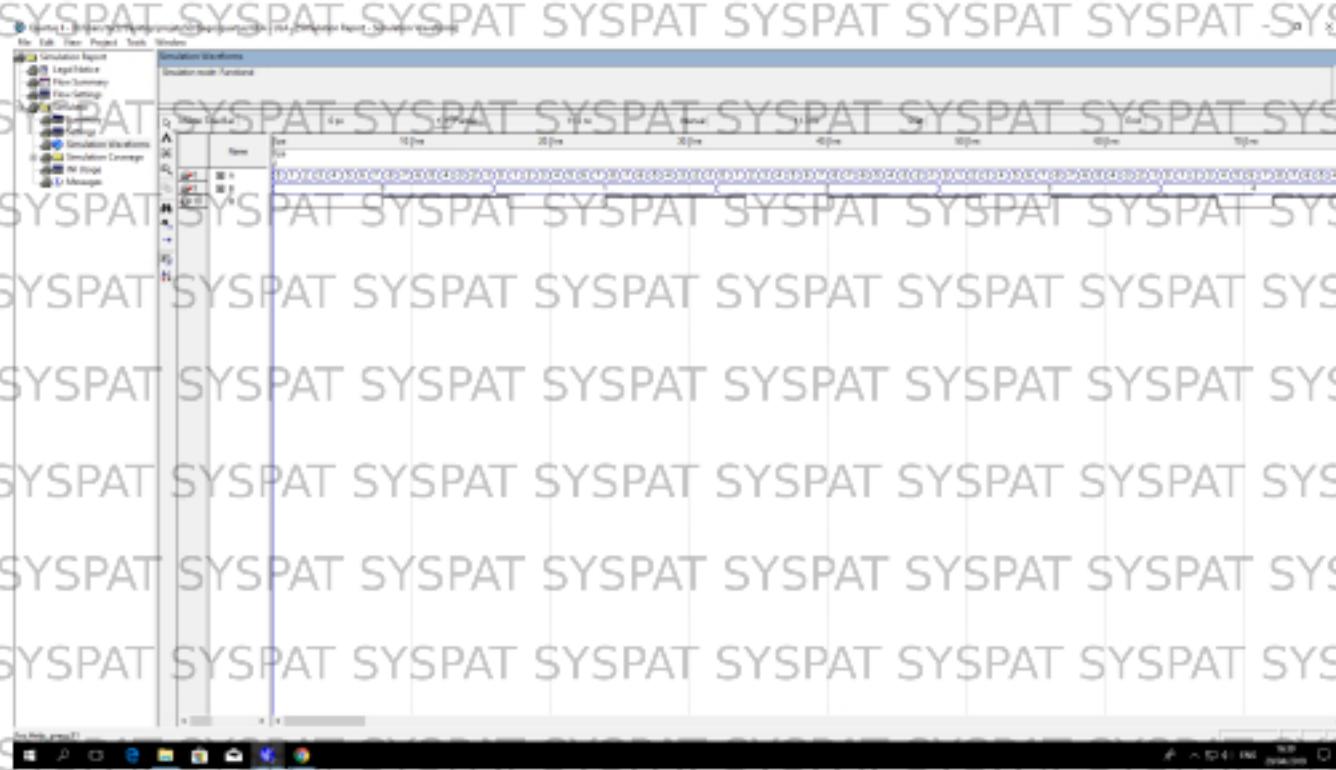
**Tabela-Verdade do bloco Maior Que do Bit Mais Significativo**

Cin	A	B	Cout
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

**Tabela Verdade do Bloco Maior Que para Bit Mais Significativo**



**Bloco Menor Que**



Waveform do Bloco Menor Que

### **Mapa de Karnaugh - Menor Que - Bit Menos Significante.**

	$\neg A \cdot \neg B$	$\neg A \cdot B$	$A \cdot B$	$A \cdot \neg B$
$\neg Cin$	0	1	0	0
$Cin$	1	1	1	0

**Mapa de Karnaugh para o Menor Que dos Bits Menos Significativos**

**Tabela-Verdade do bloco Menor Que dos Bits Menos Significativos**

Cin	A	B	Cout
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

**Tabela Verdade do Bloco Menor Que para Bits Menos Significativos**

## **Mapa de Karnaugh - Menor Que - Bit + Signif.**

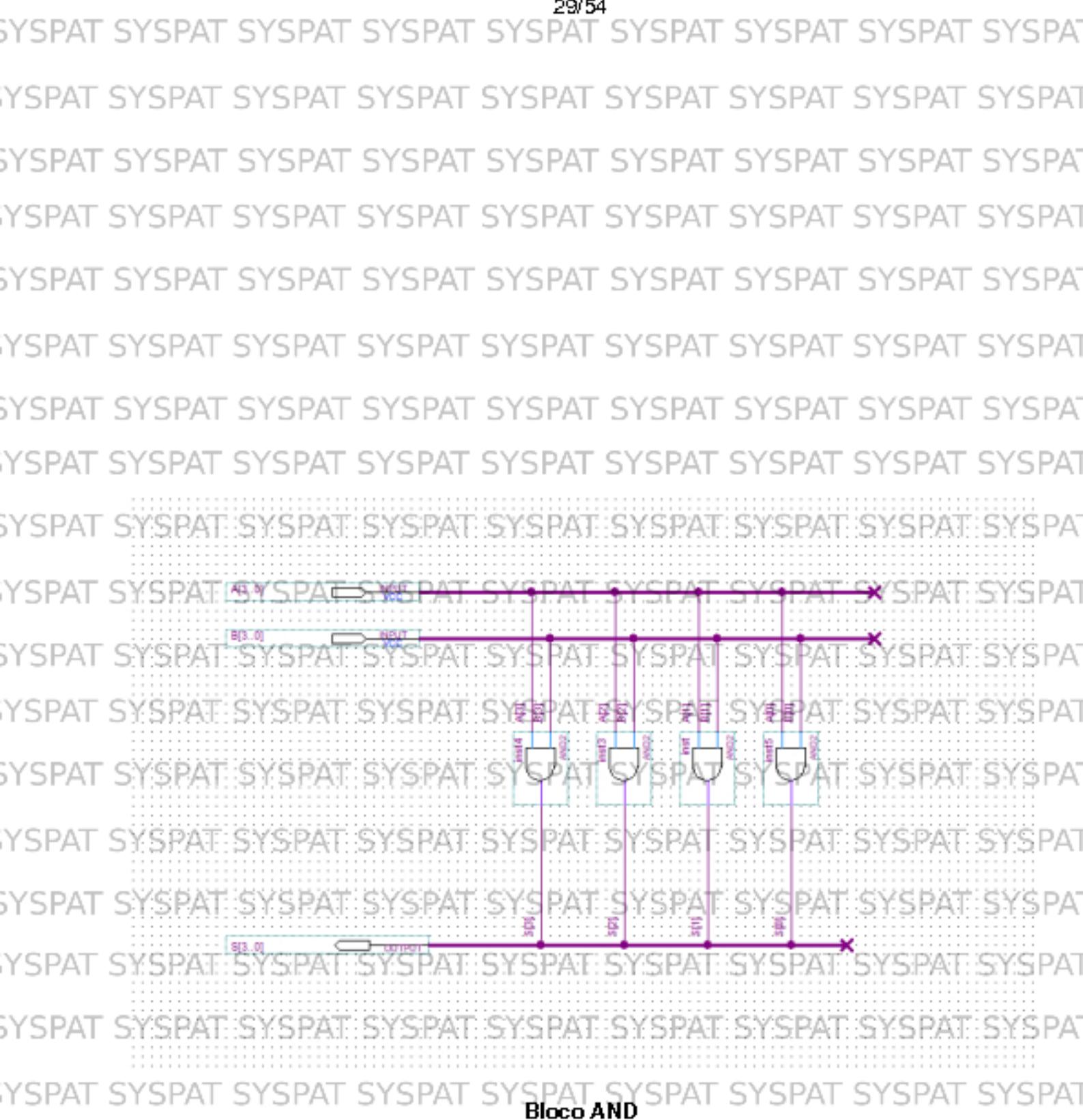
$\neg A \cdot \neg B$	$\neg A \cdot B$	$A \cdot \neg B$	$A \cdot B$	
$\neg Cin$	0	0	0	1
Cin	1	0	1	1

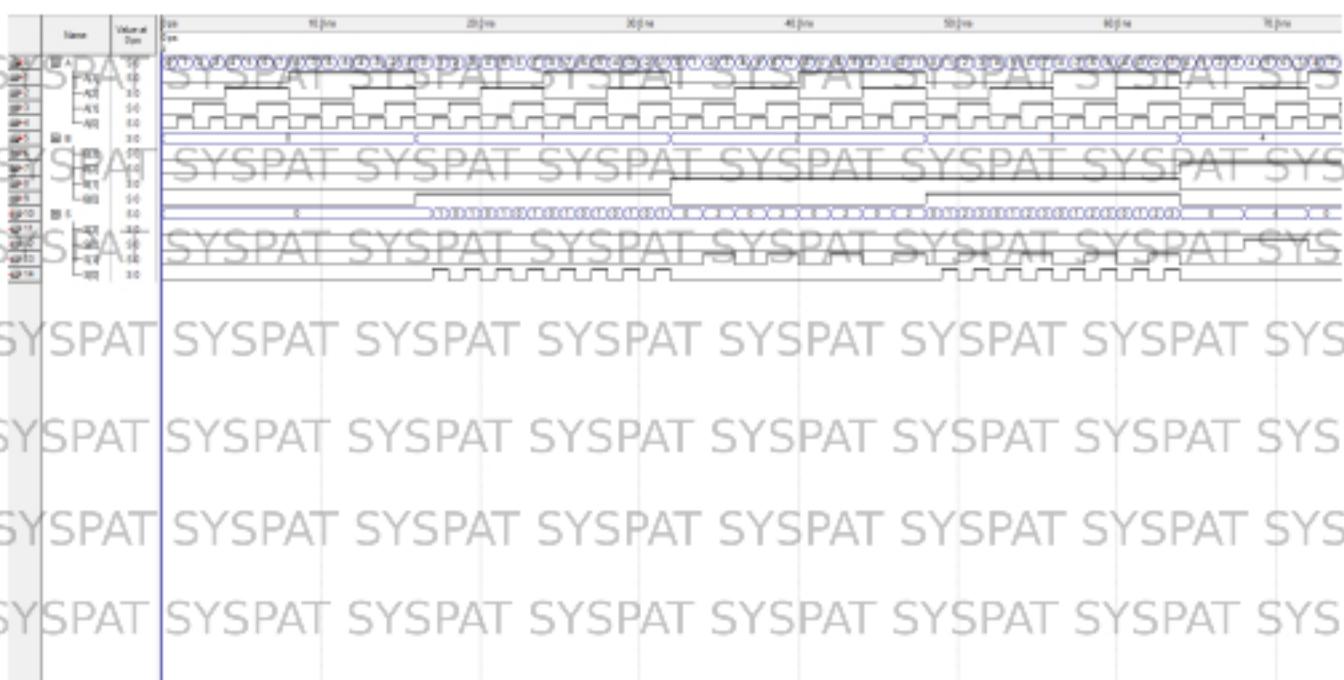
**Mapa de Karnaugh para o Menor Que do Bit Mais Significativo**

**Tabela-Verdade do bloco Menor Que do Bit Mais Significativo**

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

**Tabela Verdade do Bloco Menor Que para Bit Mais Significativo**





Waveform do Bloco AND

## Mapa de Karnaugh - AND

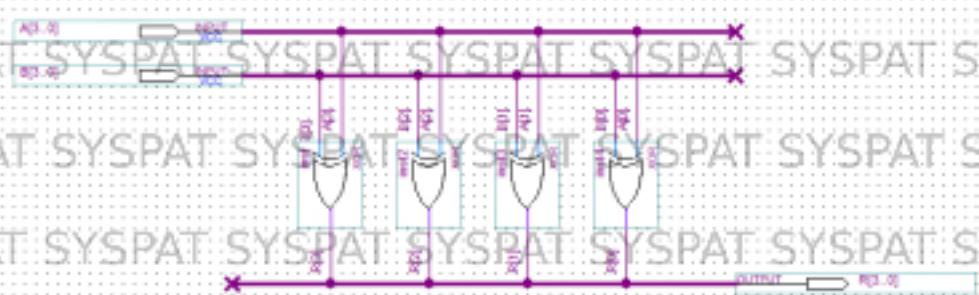
		A	A
	B	0	0
B		0	1

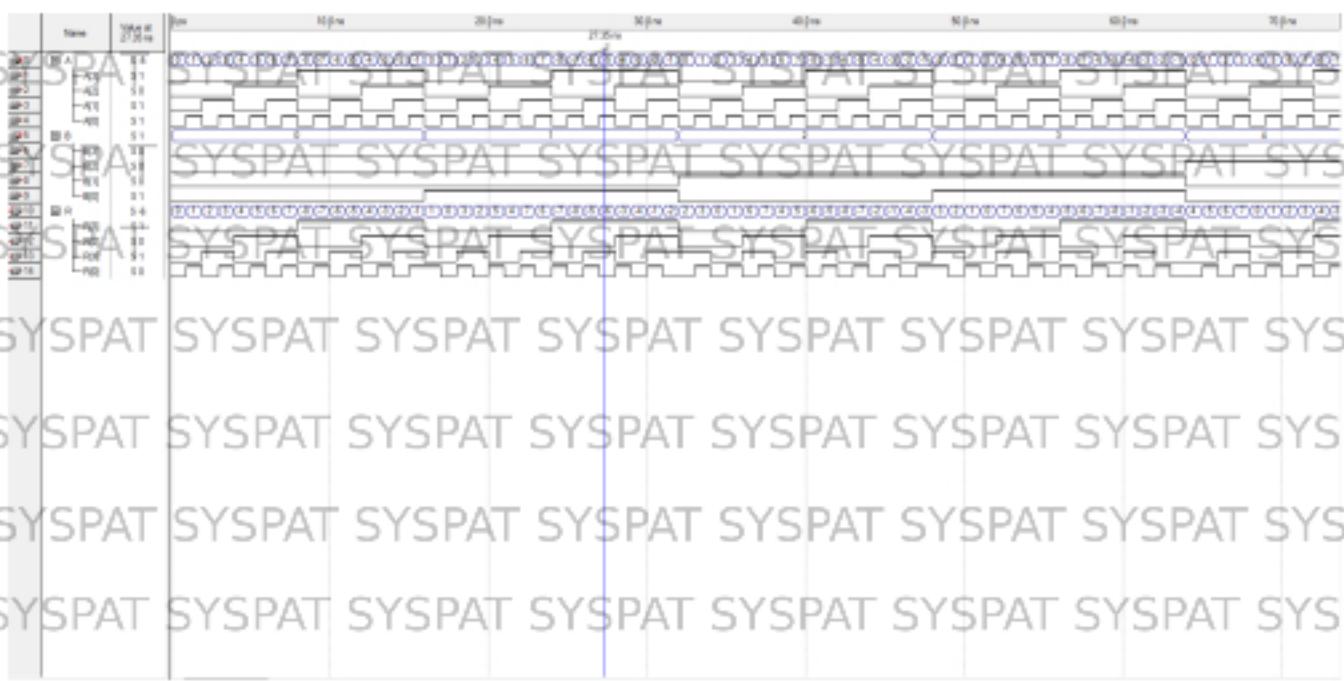
Mapa de Karnaugh do Bloco AND

## Tabela-Verdade do bloco AND

A	B	Cout
0	0	0
0	1	0
1	0	0
1	1	1

Tabela Verdade do Bloco AND

**Bloco XOR**



Waveform do Bloco XOR

## Mapa de Karnaugh - XOR

		$\neg A$	A
	$\neg B$	0	1
B		1	0

Mapa de Karnaugh do Bloco XOR

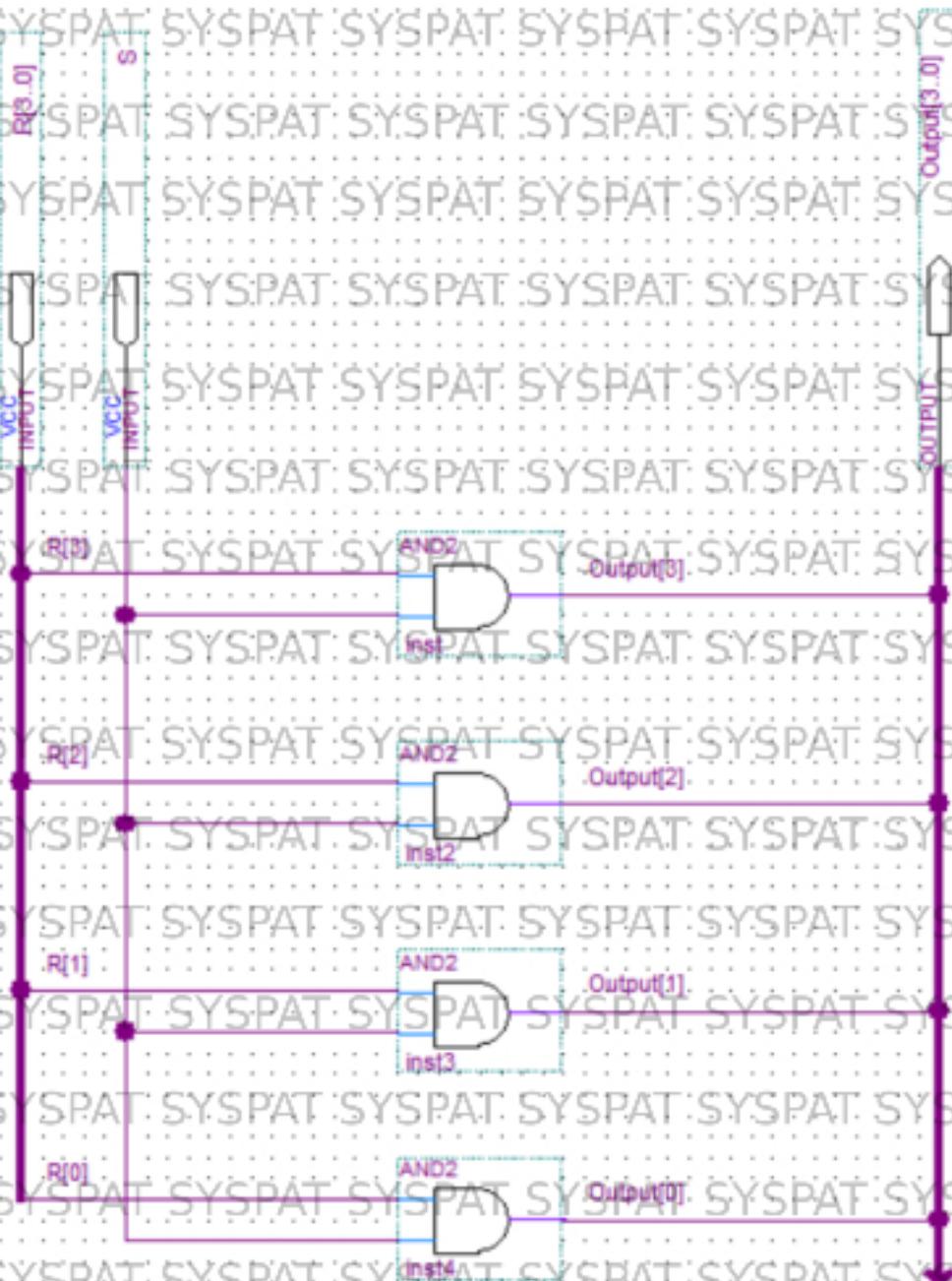
## Tabela-Verdade do bloco XOR

A	B	Cout
0	0	0
0	1	1
1	0	1
1	1	0

Tabela Verdade do Bloco XOR



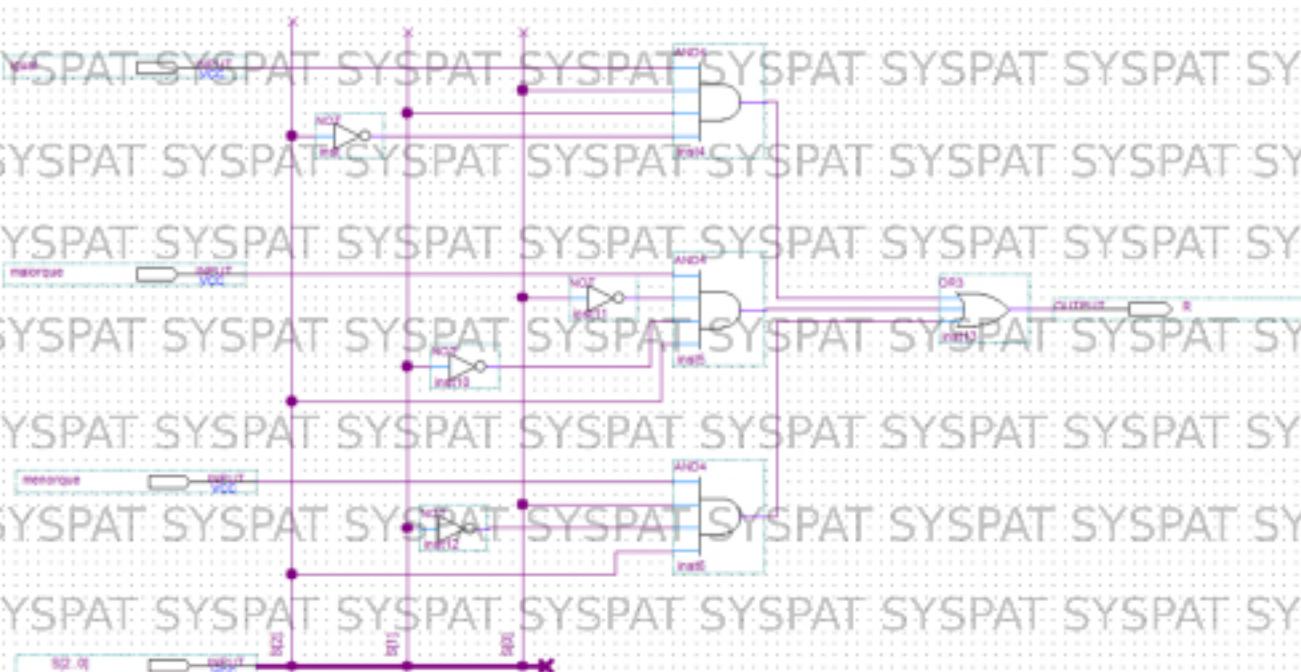
Bloco Mux 5:1

**Bloco AND-BitVector do Mux 5:1**

**Tabela-Verdade do bloco Mux 5**

<b>S2</b>	<b>S1</b>	<b>S0</b>	<b>A + B</b>	<b>A - B</b>	<b>Comp. 2 B</b>	<b>A AND B</b>	<b>A XOR B</b>
0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	0
1	1	0	0	0	0	1	0
1	1	1	0	0	0	0	1

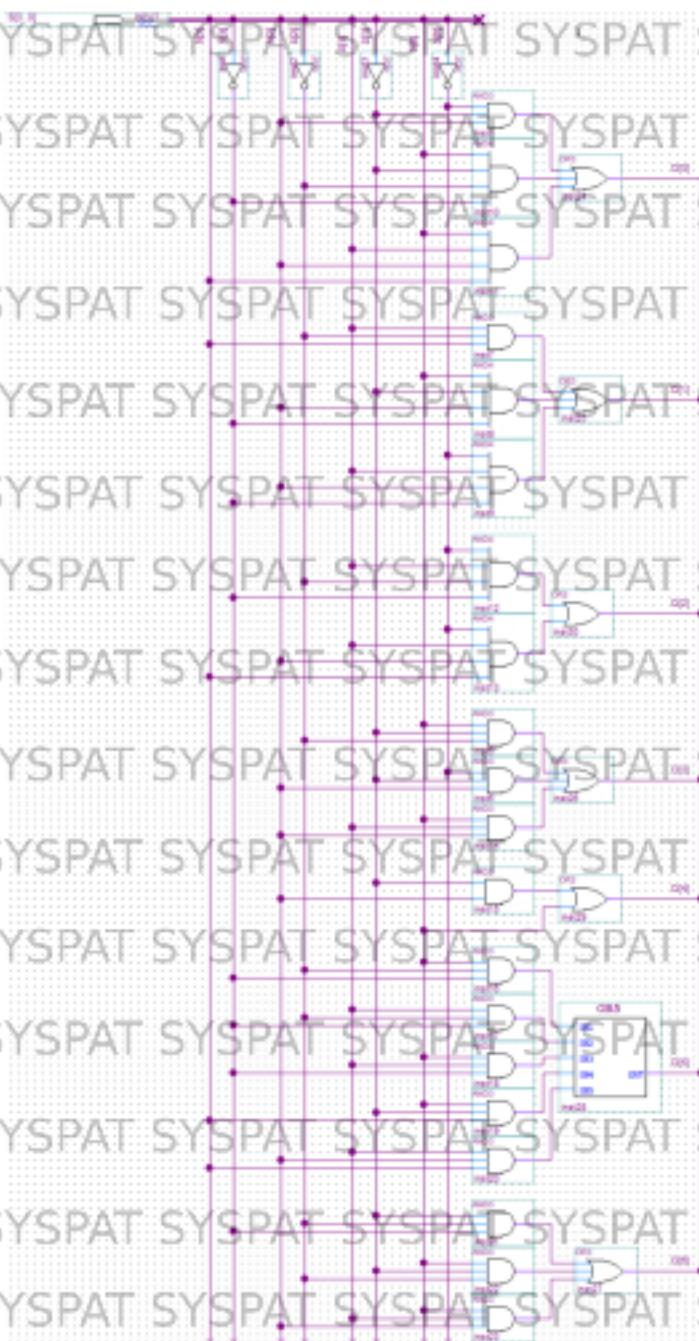
**Tabela Verdade do Bloco Mux 5:1**

**Bloco Mux 3:1**

### Tabela-Verdade do bloco Mux 3

<b>S2</b>	<b>S1</b>	<b>S0</b>	<b>A = B</b>	<b>A &gt; B</b>	<b>A &lt; B</b>
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1

Tabela Verdade do Bloco Mux 3:1



Bloco Display representando 0 a 8

S[3]

INPUT  
VCC

S[0]

S[1]

S[2]

S[3]

S[4]

S[5]

S[6]

D[0]

D[1]

D[2]

D[3]

D[4]

D[5]

D[6]

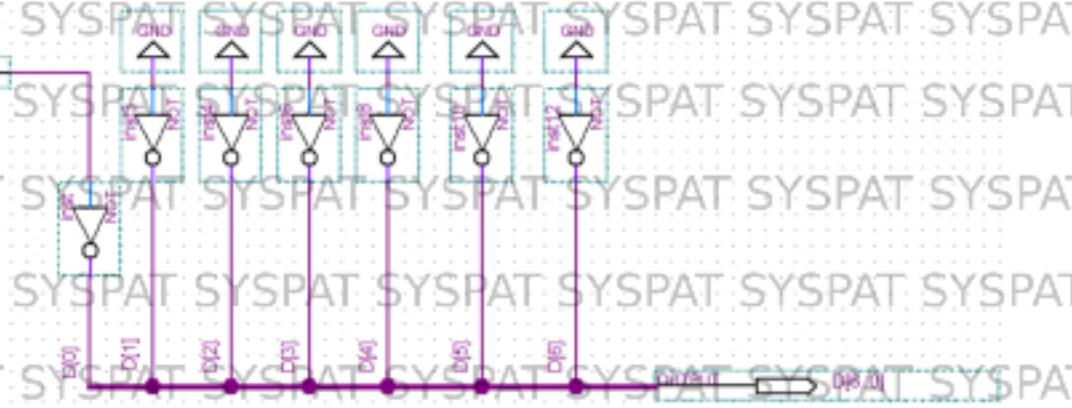
D[7]

D[8]

D[9]

**Bloco Display representando o Sinal Negativo**

S[3]

INPUT  
VCC**Bloco Display representando o Sinal Negativo**

## **Mapa de Karnaugh - Bit A do display ULA**

$\neg S_1 \cdot \neg S_0$	$\neg S_1 \cdot S_0$	$S_0 \cdot S_1$	$S_1 \cdot \neg S_0$	
$\neg S_3 \cdot \neg S_2$	0	1	0	0
$S_3 \cdot \neg S_2$	1	0	0	0
$S_3 \cdot S_2$	1	0	1	0
$\neg S_3 \cdot S_2$	0	0	0	0

**Mapa de Karnaugh Bit A do Display**

## Mapa de Karnaugh - Bit B do display ULA

	$\neg S_1.\neg S_0$	$\neg S_1.S_0$	$S_0.S_1$	$S_1.\neg S_0$
$\neg S_3.\neg S_2$	0	0	0	0
$S_3.\neg S_2$	0	1	0	1
$S_3.S_2$	0	0	0	0
$\neg S_3.S_2$	0	0	1	1

Mapa de Karnaugh Bit B do Display

## Mapa de Karnaugh - Bit C do display ULA

	$\neg S_1 \cdot \neg S_0$	$\neg S_1 \cdot S_0$	$S_0 \cdot S_1$	$S_1 \cdot \neg S_0$
$\neg S_3 \cdot \neg S_2$	0	0	0	1
$S_3 \cdot \neg S_2$	0	0	0	0
$S_3 \cdot S_2$	0	0	0	1
$\neg S_3 \cdot S_2$	0	0	0	0

Mapa de Karnaugh Bit C do Display

## Mapa de Karnaugh - Bit D do display ULA

	$\neg S_1 \cdot \neg S_0$	$\neg S_1 \cdot S_0$	$S_0 \cdot S_1$	$S_1 \cdot \neg S_0$
$\neg S_3 \cdot \neg S_2$	0	1	0	0
$S_3 \cdot \neg S_2$	1	0	1	0
$S_3 \cdot S_2$	1	0	1	0
$\neg S_3 \cdot S_2$	0	1	0	0

Mapa de Karnaugh Bit D do Display

## Mapa de Karnaugh - Bit E do display ULA

	$\neg S_1 \cdot \neg S_0$	$\neg S_1 \cdot S_0$	$S_0 \cdot S_1$	$S_1 \cdot \neg S_0$
$\neg S_3 \cdot \neg S_2$	0	1	1	0
$S_3 \cdot \neg S_2$	1	1	1	0
$S_3 \cdot S_2$	1	1	1	0
$\neg S_3 \cdot S_2$	0	1	1	0

**Mapa de Karnaugh Bit E do Display**

## Mapa de Karnaugh - Bit F do display ULA

	$\neg S_1.\neg S_0$	$\neg S_1.S_0$	$S_0.S_1$	$S_1.\neg S_0$
$\neg S_3.\neg S_2$	0	1	1	1
$S_3.\neg S_2$	0	0	1	0
$S_3.S_2$	0	1	1	1
$\neg S_3.S_2$	0	1	0	0

**Mapa de Karnaugh Bit F do Display**

## **Mapa de Karnaugh - Bit G do display ULA**

$\neg S_1.\neg S_0$	$\neg S_1.S_0$	$S_0.S_1$	$S_1.\neg S_0$
$\neg S_3.\neg S_2$	1	1	0
$S_3.\neg S_2$	0	0	1
$S_3.S_2$	0	0	1
$\neg S_3.S_2$	0	1	0

**Mapa de Karnaugh Bit G do Display**

## **Mapa de Karnaugh - Bit H do display ULA**

	$\neg S_1 \cdot \neg S_0$	$\neg S_1 \cdot S_0$	$S_0 \cdot S_1$	$S_1 \cdot \neg S_0$
$\neg S_3 \cdot \neg S_2$	1	1	1	1
$S_3 \cdot \neg S_2$	1	1	1	1
$S_3 \cdot S_2$	0	0	0	0
$\neg S_3 \cdot S_2$	0	0	0	0

**Mapa de Karnaugh Bit H do Display**

**Tabela-Verdade do display da ULA**

S3	S2	S1	S0	A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	1
0	0	1	1	0	0	0	0	1	1	0	1
0	1	0	0	1	0	0	1	1	0	0	1
0	1	0	1	0	1	0	0	1	0	0	1
0	1	1	0	0	1	0	0	0	0	0	1
0	1	1	1	0	0	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	1	1	1
1	0	1	0	0	1	0	0	0	0	0	1
1	0	1	1	0	1	0	0	1	0	0	1
1	1	0	0	1	0	0	1	1	0	0	1
1	1	0	1	0	0	0	0	1	1	0	1
1	1	1	0	0	0	1	0	0	1	0	1
1	1	1	1	1	0	0	1	1	1	1	1

**Tabela Verdade do Display**

## Tabela-Verdade do bloco Igual

A	B	Cout
0	0	1
0	1	0
1	0	0
1	1	1

Tabela Verdade do Bloco Igual

**UNIDADE LÓGICA E ARITMÉTICA INTEIRA PARA PROCESSAMENTO DE 4 BITS****ABSTRACT**

A Unidade Lógica e Aritmética é capaz de ser integrada a pequenos processadores e tem como funcionalidades a soma, subtração, dentre outras funcionalidades lógicas e aritméticas. Juntamente à ela, estão acoplados dois displays de 7 bits, que apresentam os valores retornados pelas operações aritméticas, além do bit de resposta para as operações lógicas e o bit de overflow para operações que geram overflow.