

UNIDADE LÓGICA E ARITMÉTICA INTEIRA PARA PROCESSAMENTO DE 4 BITS.

RELATÓRIO DESCRIPTIVO

CAMPO DA INVENÇÃO

[001] A unidade lógica e aritmética implementa 8 funções: XOR, AND, adição, subtração, complemento a 2, lógica relacional $>$, $<$, $=$, considerando números inteiros de 4 bits.

[002] Cada função é escolhida mediante um entrada de seleção. A saída é representada por vetores de números inteiros, bit do overflow e saída relacional.

ANTERIORIDADES

[003] Uma ULA pode ser projetada de forma que receba diferentes operações. Essas operações podem ser simples (operações aritméticas com números inteiros, operações lógicas bit a bit ou operações de descolamento de bits), ou extremamente complexas, como é o caso de uma ULA que calcule a raiz quadrada de um número.

A patente GB 1171266, publicada com título "Arithmetic and Logic Circuits, e.g., for use in Computing", em 19 de novembro de 1969, exemplifica uma ULA que realiza operações de soma, além de conter outras operações lógicas, como XOR, AND, NAND e NOT.

Apesar de ser um circuito digital aparentemente simples, uma ULA é fundamental para diversos sistemas bastante complexos.

A patente CN 108108151, publicada com título "Arithmetic logic unit operation method and system for superconduction single magnetic flux quantum processor", em 01 de junho de 2018, refere-se a uma tecnologia complexa de supercondutão adotada com o objetivo de superar o problema de baixa velocidade e alto consumo de energia de um sistema tradicional. A ULA

presente no sistema adota um somador de uma estrutura paralela serial de 16 bits para processamento de operação.

DESCRÍÇÃO GERAL

[004] Esta ULA foi projetada com 8 operações sobre vetores de 4 bits, comparando bit a bit e podendo gerar saídas de 1 bit ou outro vetor de 4 bits. Ela realiza algumas operações simples, consegue lidar com Carry in, Carry out e overflow trabalhando somente com portas lógicas e selecionadas através de um display de 3 bits.

São suportadas as seguintes operações:

As operações de soma e subtração foram feitas em um mesmo circuito, porém com algumas modificações para a tomada de decisão entre qual operação será feita.

1 - Operação de soma - Display 0 - 0 - 0:

A operação de soma é realizada com o input de 2 vetores de 4 bits e que passam por uma porta lógica XOR. Como a entrada é 0 - 0 - 0, ele faz uma comparação com os vetores de B porém não os modifica. Após essa porta XOR, ele as leva para uma BOX chamada fullAdder, que realiza essa soma. A entrada desta porta é composta por um Carry In, e pelos vetores A e B. A caixa fullAdder é composta por portas lógicas XOR, OR e AND, sendo duas portas XOR utilizadas para a operação que resulta na soma dos bits de entrada, e as OR e AND utilizadas para checar se há um Carry Out. O sistema completo realiza a operação bit a bit e para isso são necessárias 4 caixas fullAdder com o Carry out de cada bit ligado ao Carry in da outra fullAdder e resultando como output um vetor S.

2 - Operação de subtração - Display 0 - 0 - 1:

A operação de subtração funciona de forma semelhante a da soma, sendo a grande diferença o 1 do display, que passa pelo XOR invertendo o vetor B e entrando como 1 no Carry In do primeiro fullAdder, fazendo com que seja a soma de A + B invertido + 1.

3 - Complemento a 2 - Display 0 -1 - 0:

O complemento a 2 de um número pode ser feito através do complemento a um (inversão dos bits) + 1. Na ULA, essas operações são feitas com portas lógicas XOR e 4 caixas fullAdder. O input é um vetor de 4 bits a ser retornado seu complemento e possuem duas constantes VCC(1) e GND(0), sendo o VCC utilizado como o vetor B bit a bit, passando por uma porta XOR e o resultado entrando no "B" do fullAdder, além de entrar no Carry In do fullAdder. Já o GND entra no "A" do fullAdder. Possui saída para overflow e o primeiro bit representa o sinal, dizendo se o número é positivo ou negativo.

4 - Igual - Display 0 - 1 - 1:

É utilizado para verificar se dois vetores são iguais, seu circuito é composto por 4 "EqualsBox" e uma porta "And" com 4 entradas. O EqualsBox recebe como entrada 2 bits, sendo um de A e outro de B para ser feita a comparação, passando por uma porta "And" o valor normal dos bits e em outra "And" o seu valor negado, e suas saídas passam por uma porta "OR", finalizando a Box.

Após cada bit passar pela EqualsBox, entra num "And" de 4 entradas e gera um output de um bit, sendo 1 para iguais e 0 para diferentes.

O "maior que" e "menor que" foram implementados no mesmo circuito, tendo somente como diferença o bit de entrada para operação.

5 - Maiorque - Display 1 - 0 - 0:

O circuito não é tão simples quanto as outras operações, realizando algumas operações lógicas a mais. O circuito recebe como input dois vetores e, bit a bit, compara os dois com um XOR, e sua saída vai direto para um AND, que recebe além deste bit a constante VCC (1), somente no primeiro bit do vetor, sua saída passa para o próximo AND, com o outro bit do XOR, e assim sucessivamente. O circuito também nega o bit do vetor A, que passa por um AND junto com o bit do B, e sua saída, junto com o VCC, passa por um AND e todas se unem em um OR de 4 entradas. O resultado do OR é negado e entra em um AND, junto com a saída final das sucessivas passagens do VCC e dos bits que passaram pelo XOR, no início. A saída desse AND passa por outro AND, junto com uma entrada de valor 1, caso a operação seja "maior que", ou junto com uma entrada de valor 0, caso a operação seja "menor que". A saída do OR de 4 entradas passa por um AND, junto com o valor referente a operação que está sendo realizada, só que negado. O resultado desse AND passa por um OR, junto com o resultado do AND passado (que recebeu a saída do OR de 4 entradas e o valor referente a operação que está sendo realizada), retornando 1 para verdadeiro ou 0 para falso, finalizando o circuito.

6 - Menor que - Display 1 - 0 - 1:

Funciona de modo igual ao maior que, só recebendo o bit 1 ao invés de 0 no terceiro input.

7 - Função AND - Display 1-1-0

A função AND é uma função simples, que recebe como input dois vetores e bit a bit aplica a porta lógica AND sobre eles, retornando o vetor S.

8 - Função XOR - Display 1 - 1 - 1

A função XOR funciona de modo semelhante a função AND, mas ao invés de aplicar a função lógica AND, aplica a XOR também bit a bit e retornando o vetor S.

Todas essas funções foram juntadas em um multiplexador (MUX), que possui

um circuito com todas as operações e que escolherá, através de um outro input de 3 bits (Codificador da operação), o que será feito. Após passar pelo MUX e a operação ser realizada, ele irá para a box do display, transformando o vetor de 4 bits em um vetor de 7 que fica responsável por controlar segmentos de LED de um display para mostrar visualmente o número que antes era representado por apenas 4 bits.

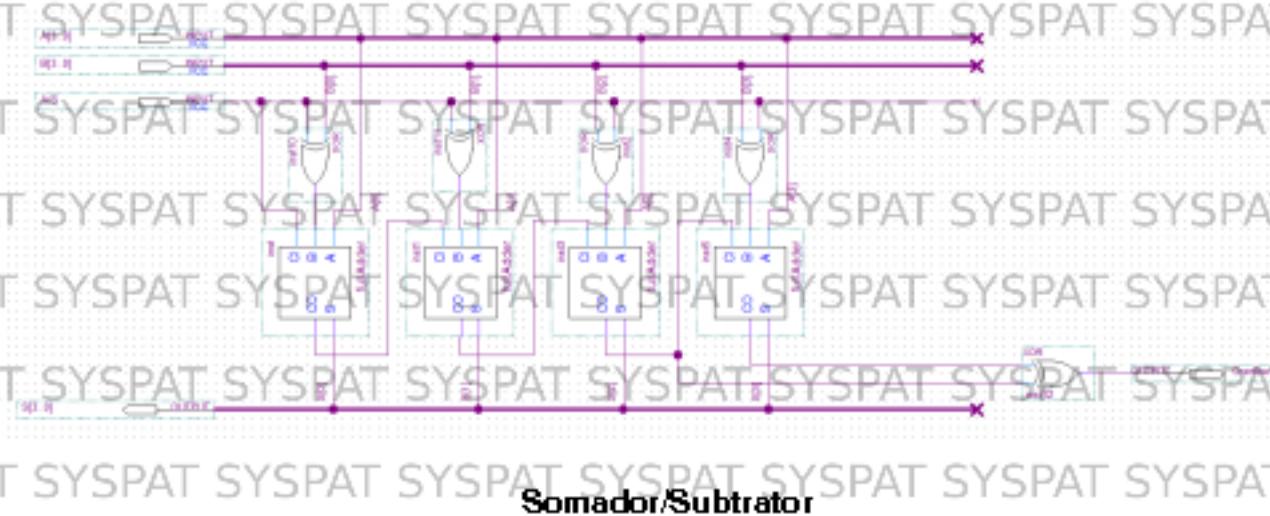
MODALIDADES PREFERIDAS

[005] A primeira modalidade preferida trata da implementação dessa ULA (unidade lógica aritmética) na área da computação, nesta área esse tipo de produto, de rápido processamento, possui em excelente desempenho, por executar os processos de maneira mais eficaz que outras unidades similares no mercado, mostrando o resultado dos processos em um display, mais precisamente, para ser utilizada em microprocessadores, por ser capaz de realizar funções com números de até 4 bits, permitindo então um processamento rápido de uma quantidade massiva de dados em pouco tempo.

[006] A unidade Lógica e Aritmética de 4 bits é uma implementação que permite ser acoplada a um microcontrolador digital para processamento de dados, voltados para conversão de sinal Digital para Analógico.

REMINDICAÇÕES

1. Unidade lógica e aritmética utilizada para microprocessadores, caracterizada por conter 8 funções diferentes (que são selecionadas a partir do input), baseadas em números de 4 bits e conectadas a um display capaz de mostrar o resultado de determinado input em LED de forma extremamente eficaz.

DESENHOS**Somador/Subtrator**

A	B	Ci	S	Co
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

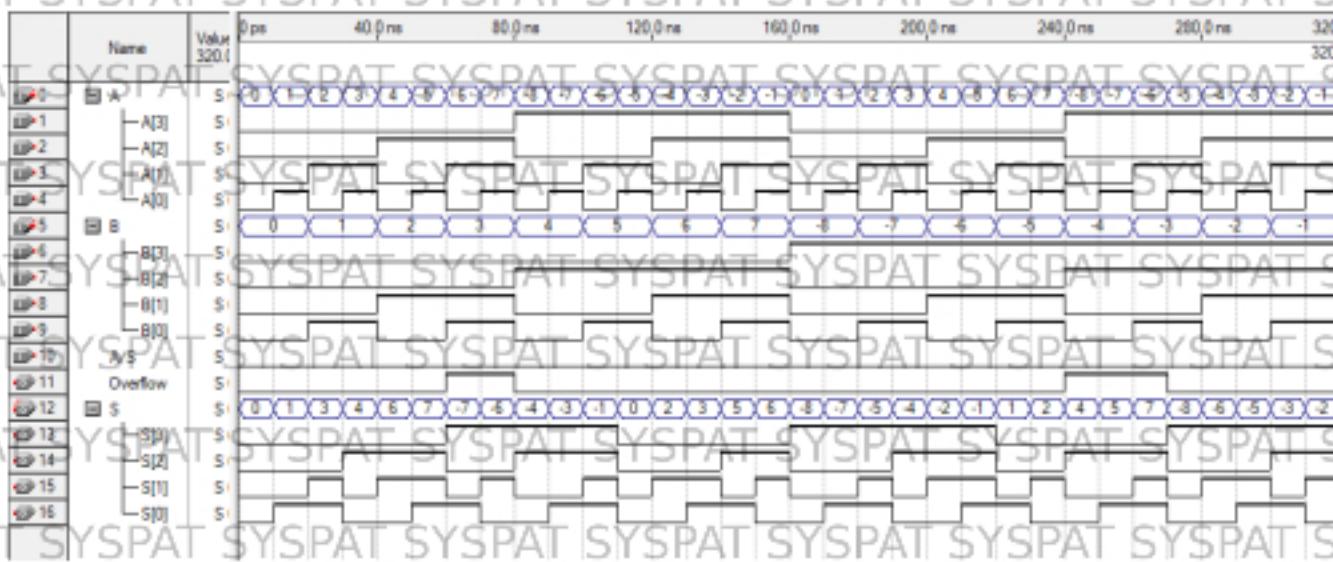
Somador/Subtrator Tabela Verdade

	$A \backslash B'_{Cin}$	00	01	11	10
B'_{Cin}	A	0	1	0	1
	0	0	1	0	1
	1	1	0	1	0

$$S = AB'Cin' + A'B'Cin + ABCin + ABCin'$$

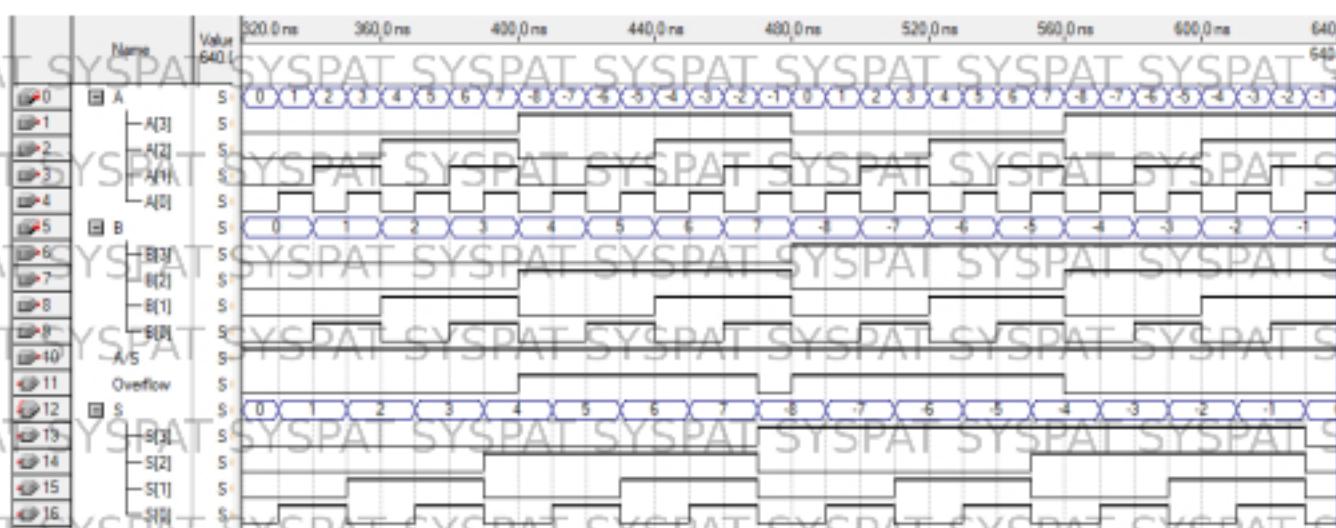
Somador Karnaugh S

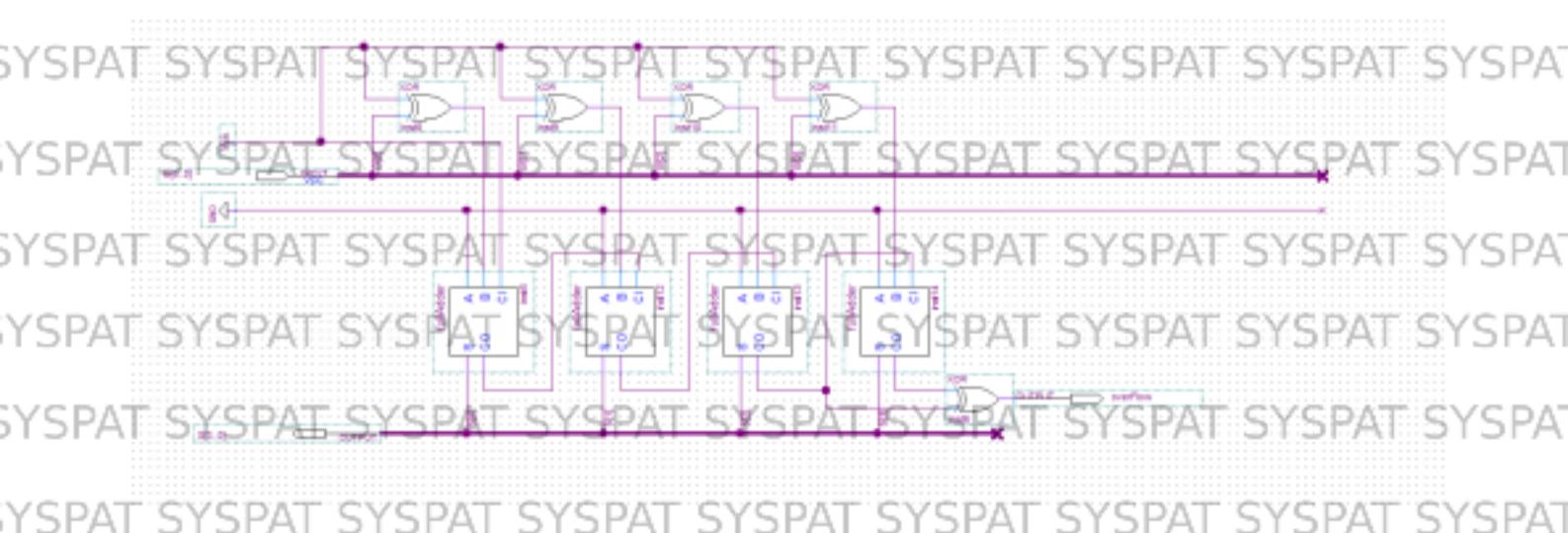
4/47

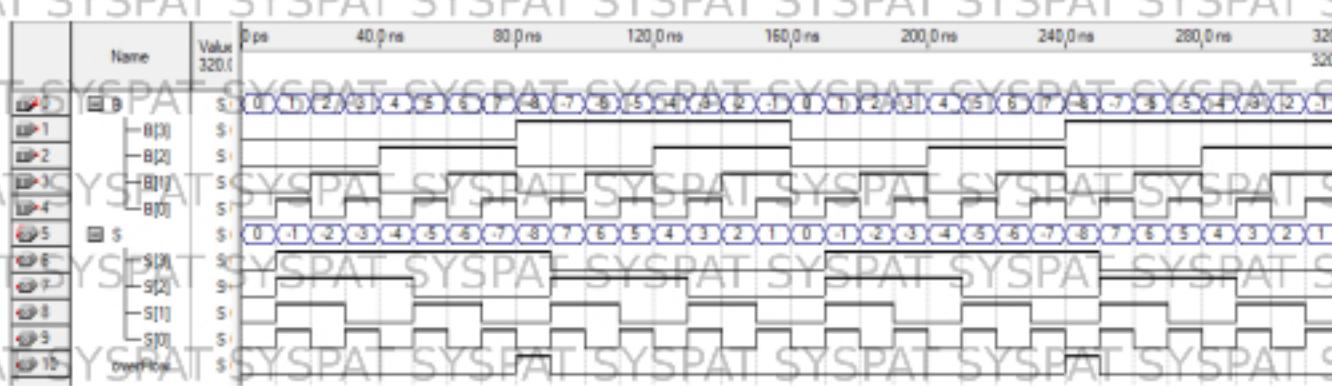


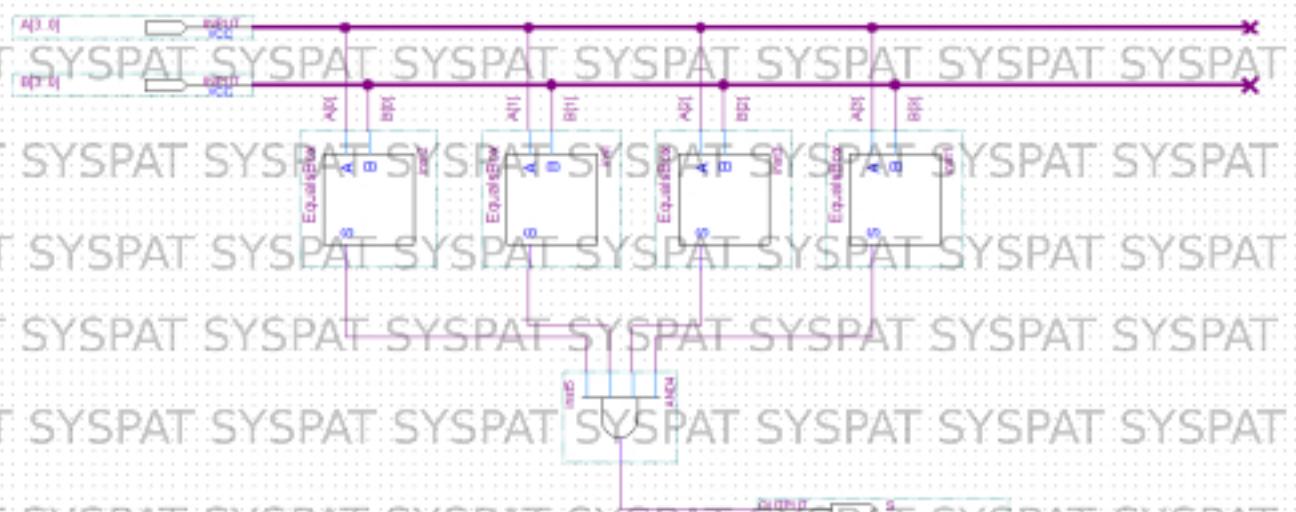
Somador Waveform



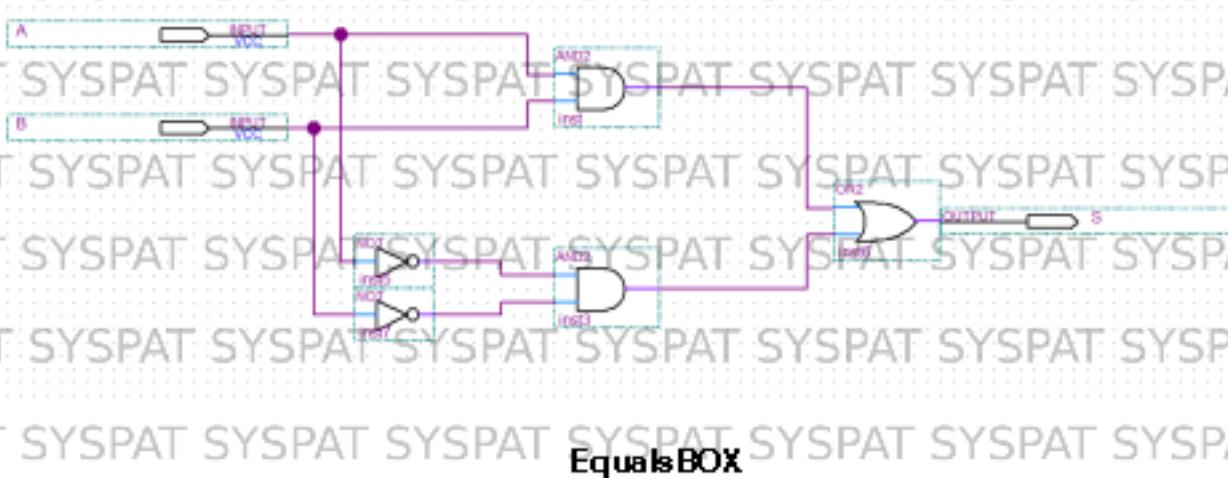


**Complemento a 2**





Equals



12/47

A

B

S

0

0

1

0

1

0

1

0

0

1

1

1

Equals tabela verdade

13/47

A

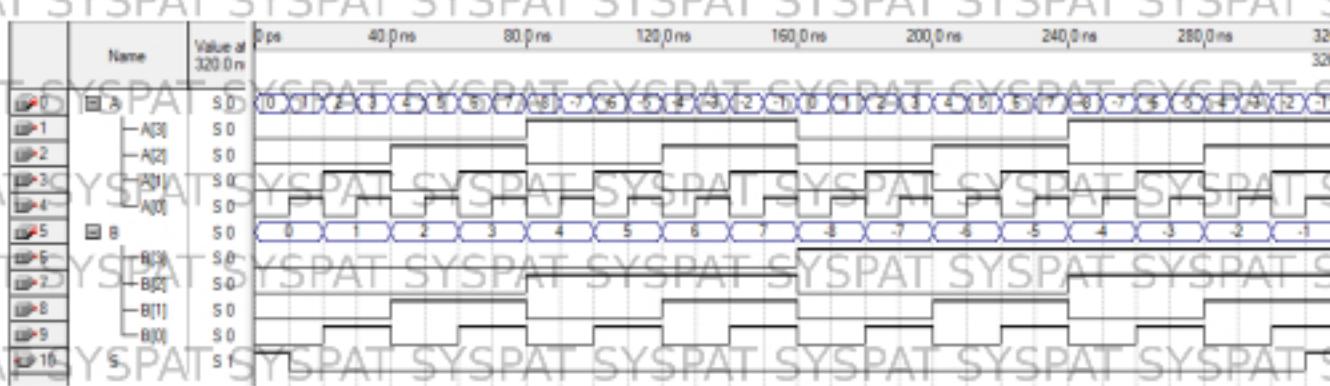
B

0

1

$$S = A'B' + AB$$

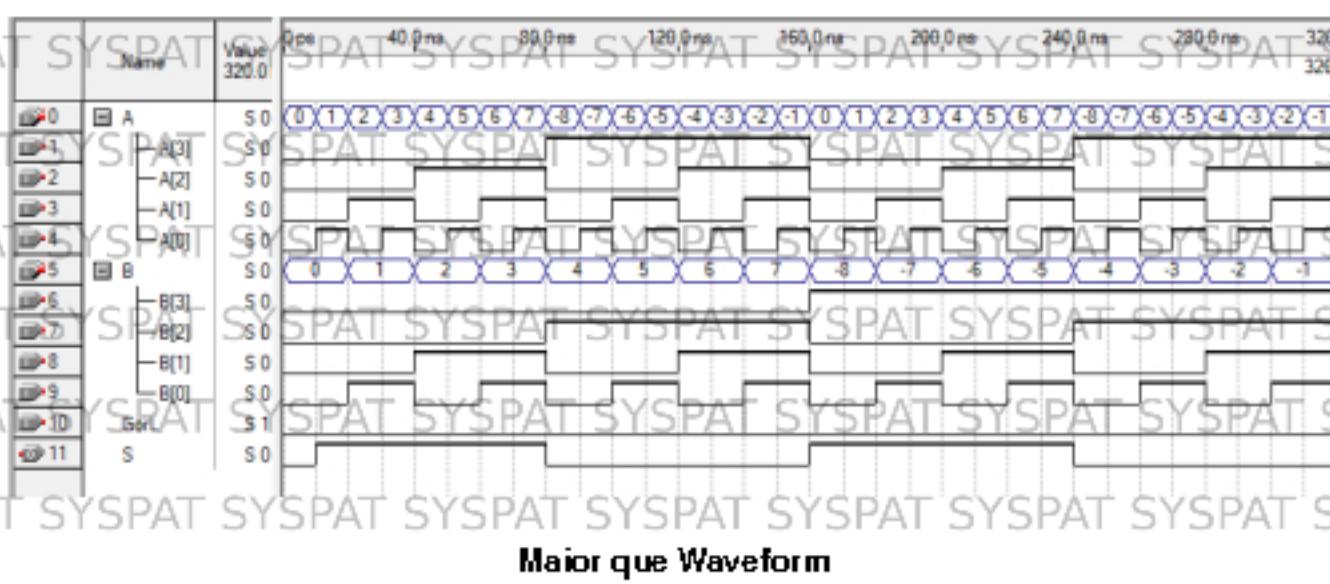
Equals Karnaugh

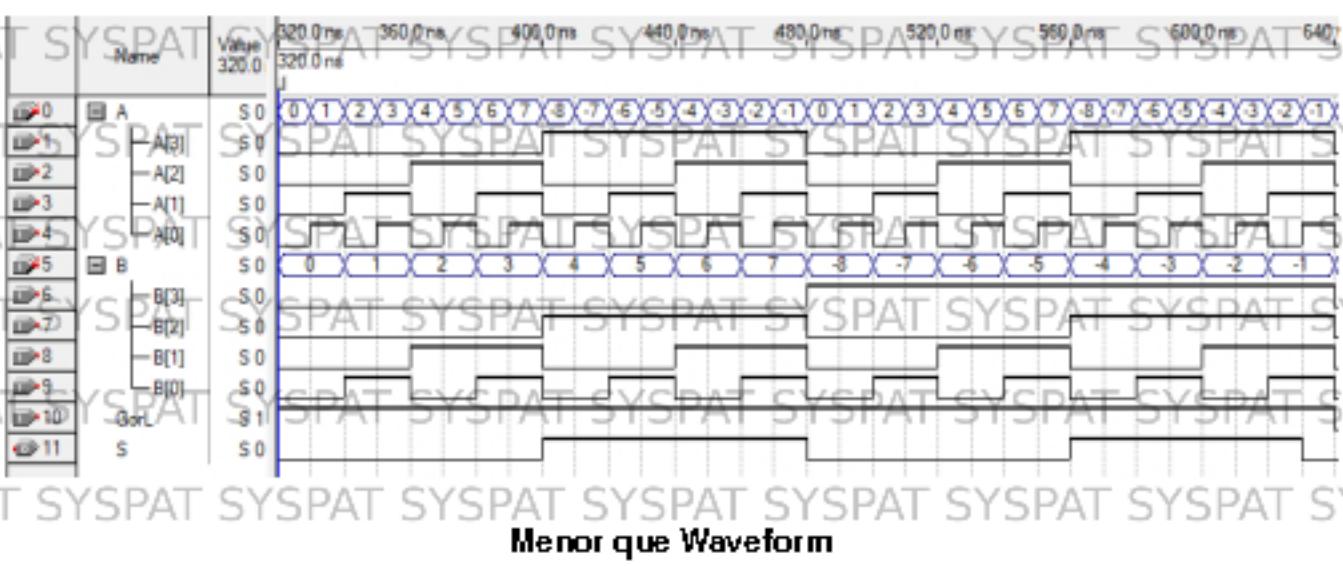


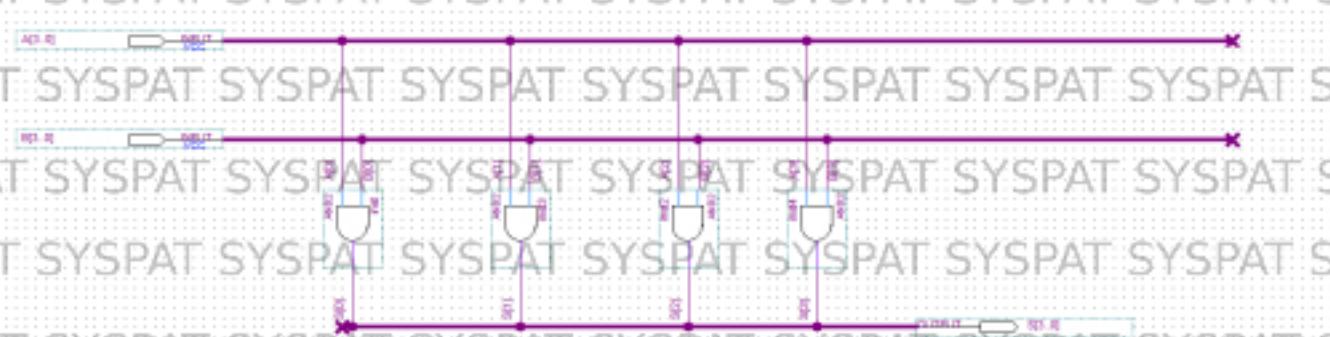
Equals Waveform



Major/Menor que







And

A

B

S

0

0

0

0

1

0

1

0

0

1

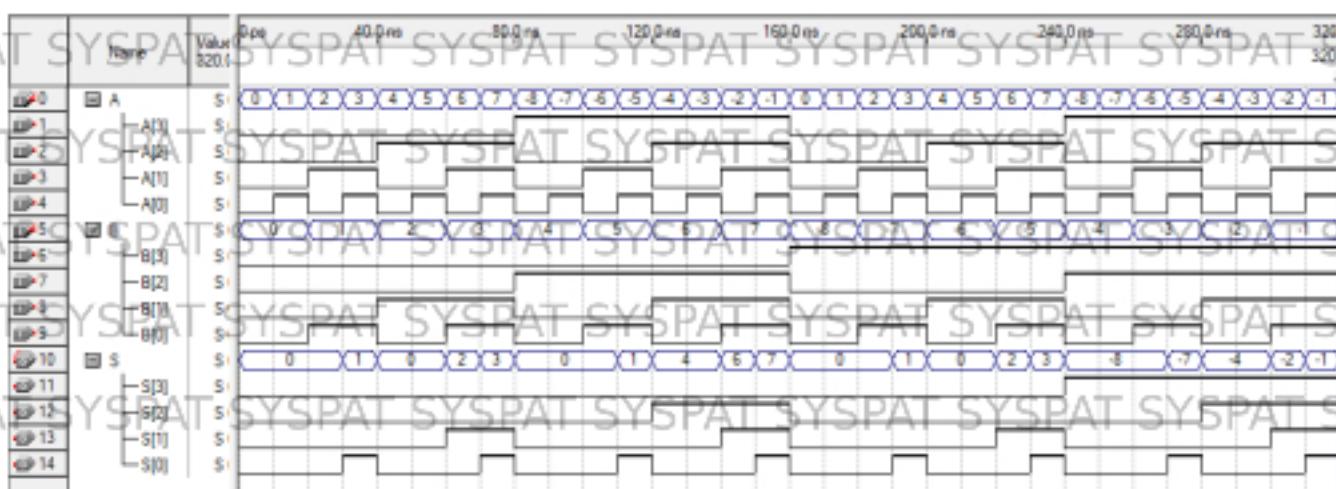
1

1

And Tabela Verdade

	B	0	1
A	0	0	0
1	0	1	1

And Karnaugh



And Waveform

~~AD 1~~ ~~DATA~~

~~AD 2~~ ~~DATA~~

~~AD 3~~ ~~DATA~~

~~AD 4~~ ~~DATA~~

~~AD 5~~ ~~DATA~~

~~AD 6~~ ~~DATA~~

XOR

A**B****S**

0

0

0

0

1

1

1

0

1

1

1

0

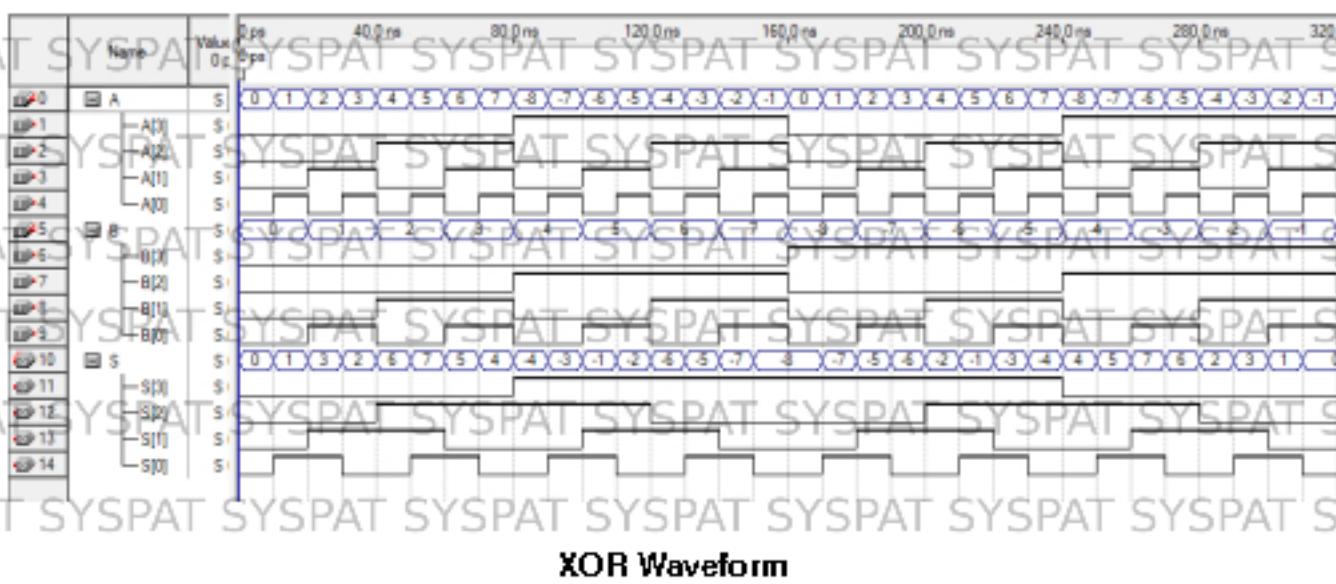
XOR Tabela verdade

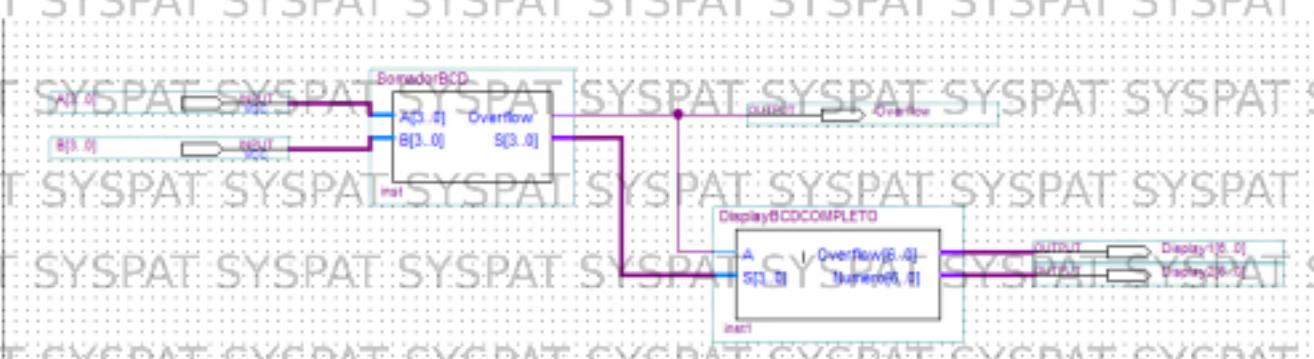
~~24/47~~

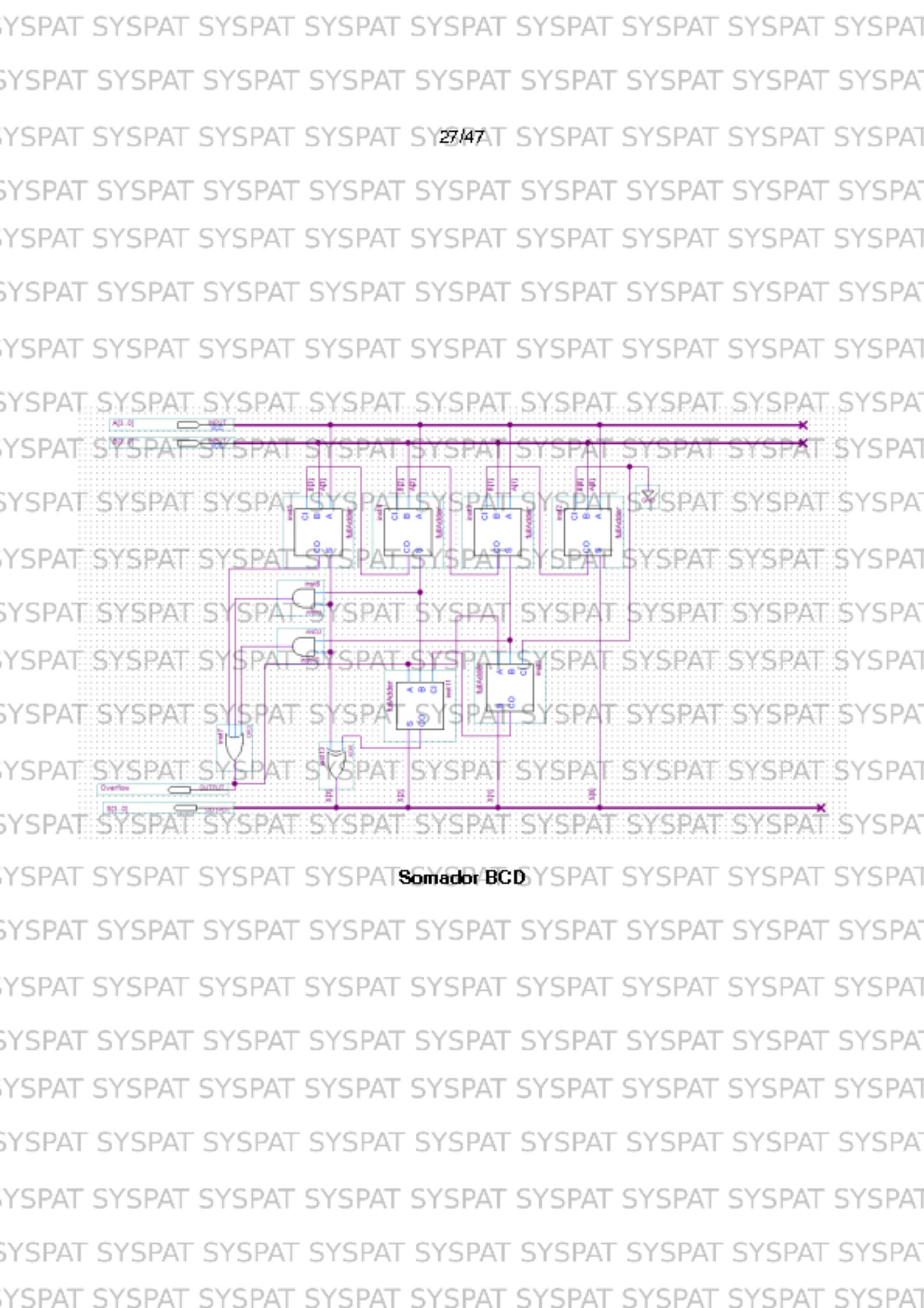
	B 0	1
A 0	0	1
1	1	0

$$S = AB' + A'B$$

XOR Karnaugh

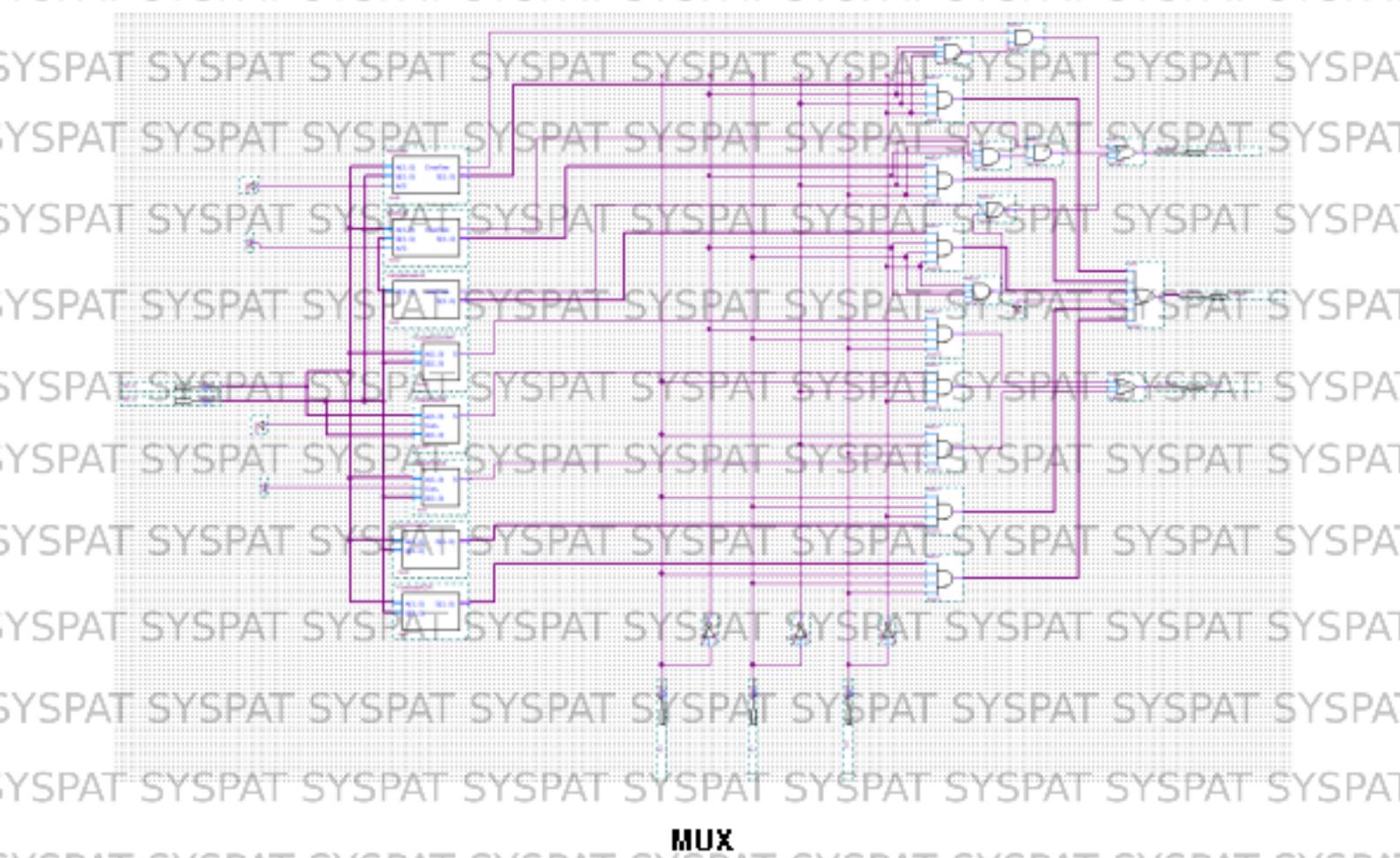


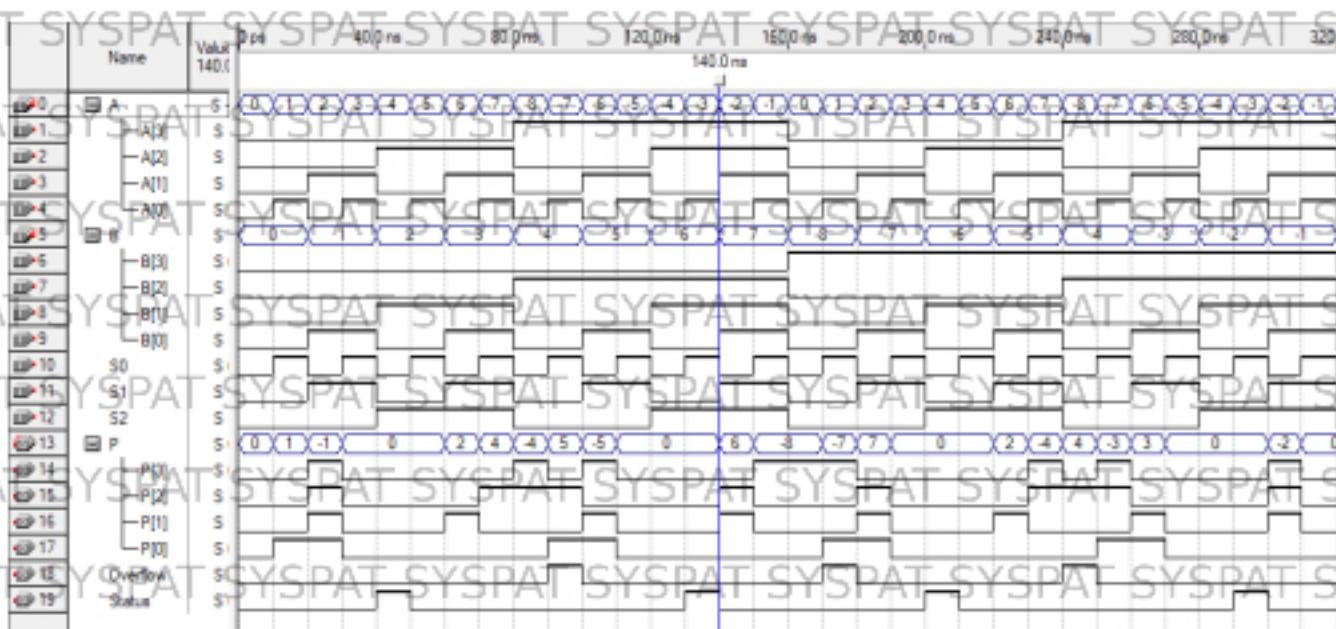
**BCD**



Somador BCD

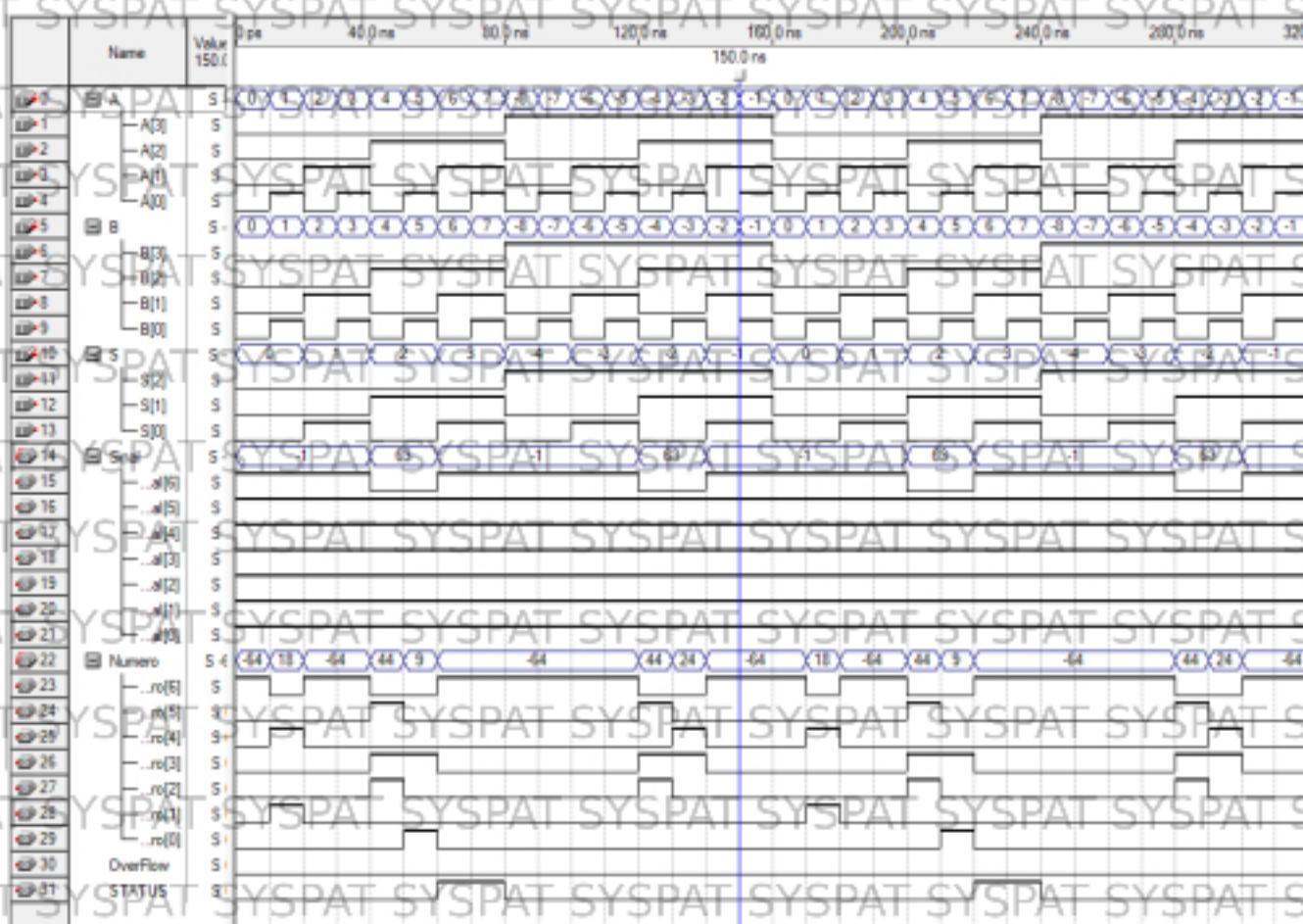


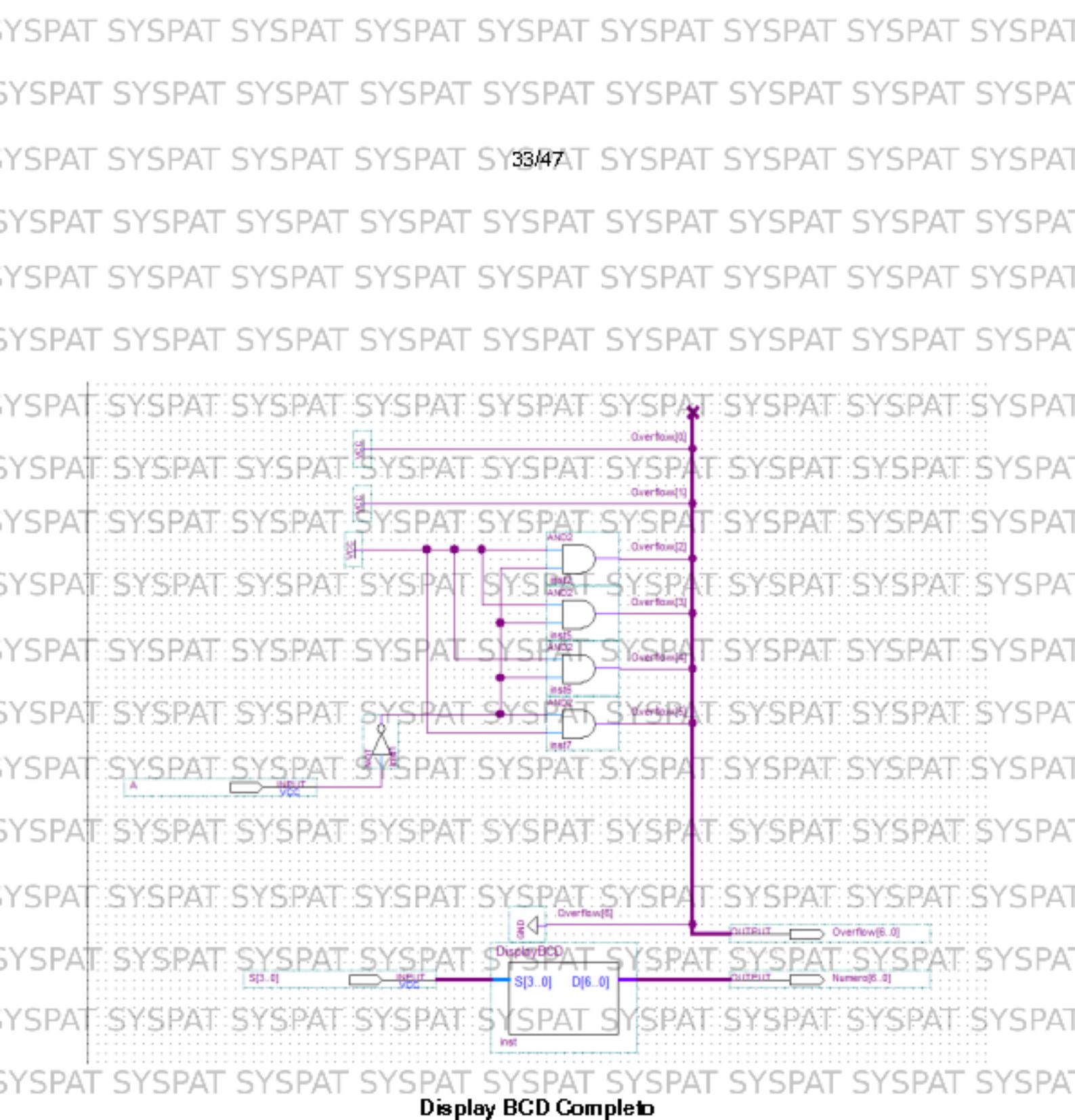


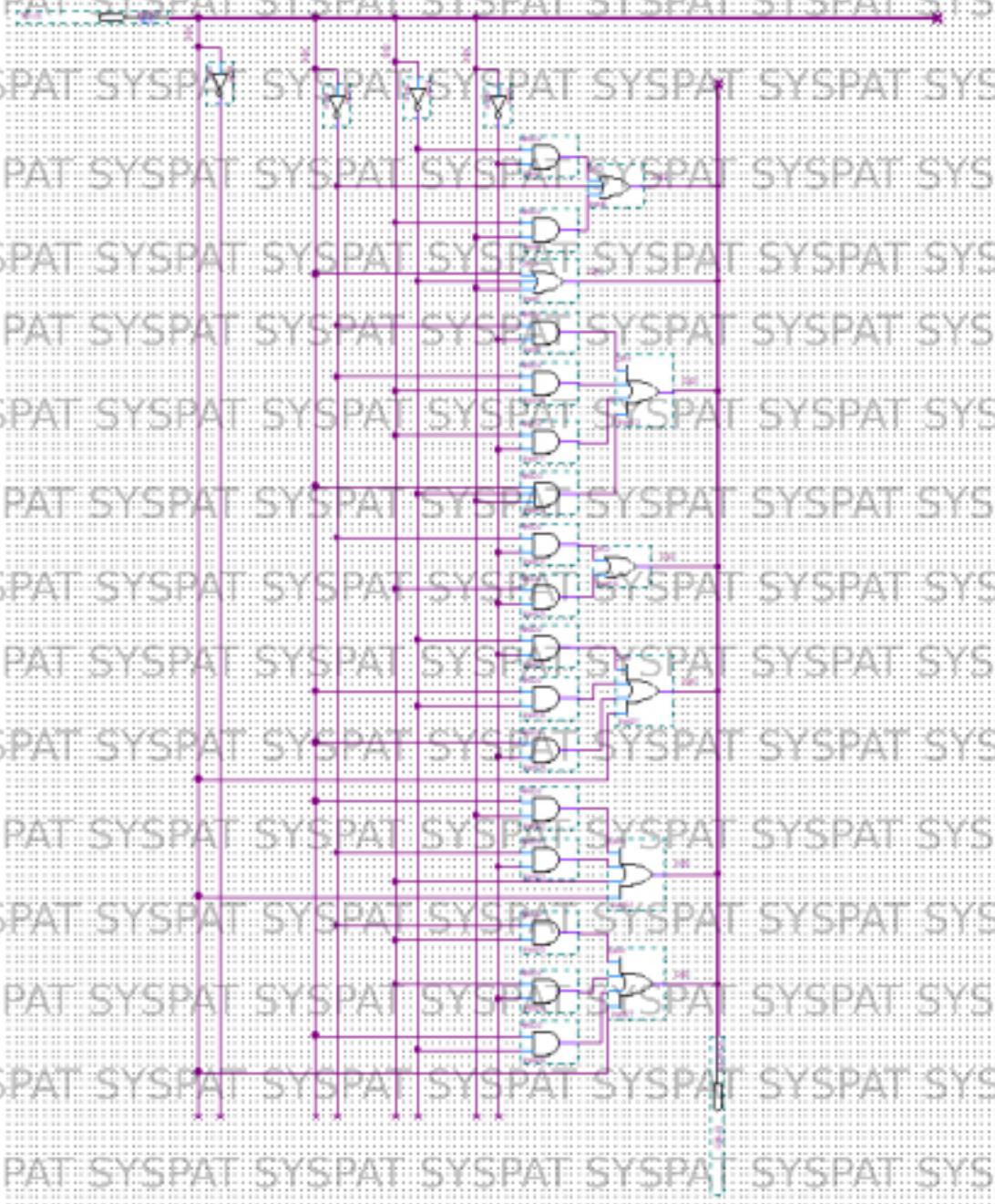


MUX Waveform

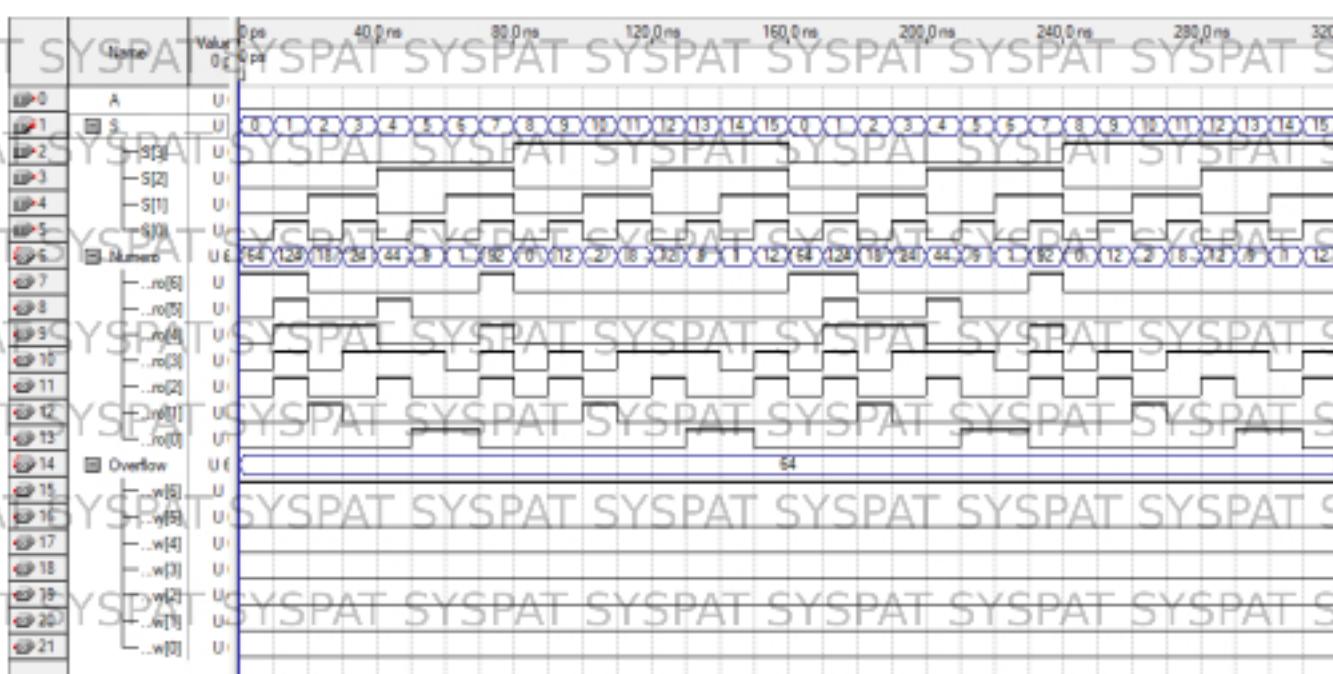




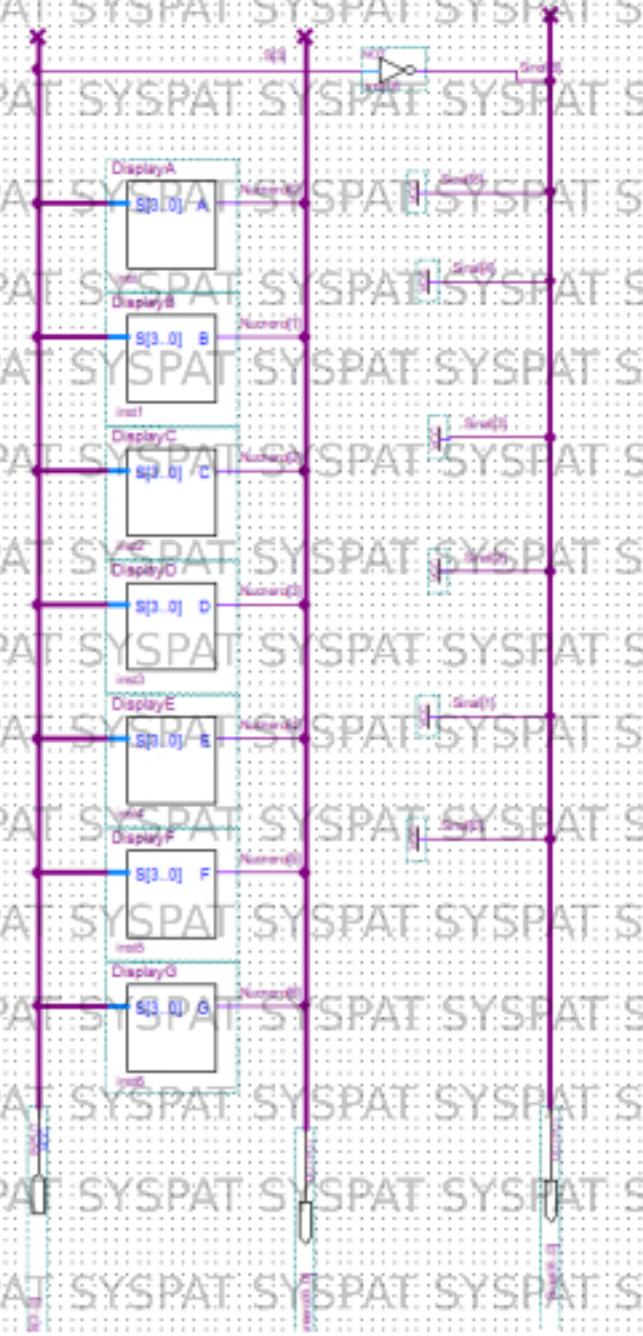




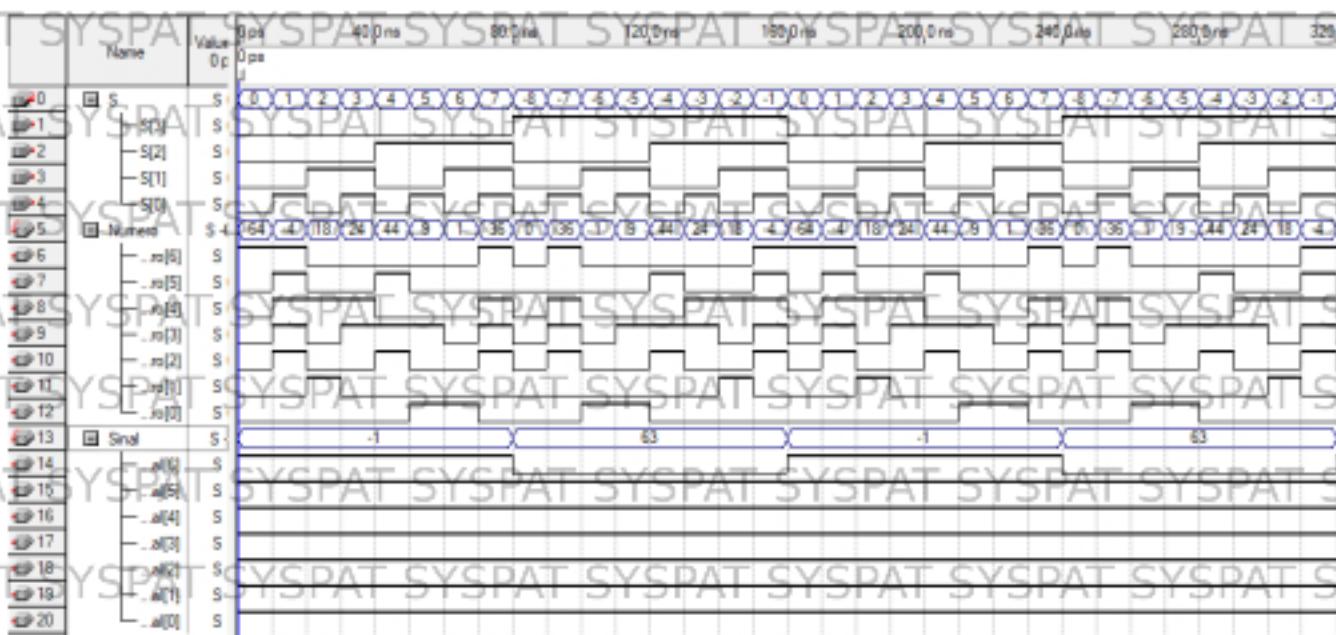
Display BCD

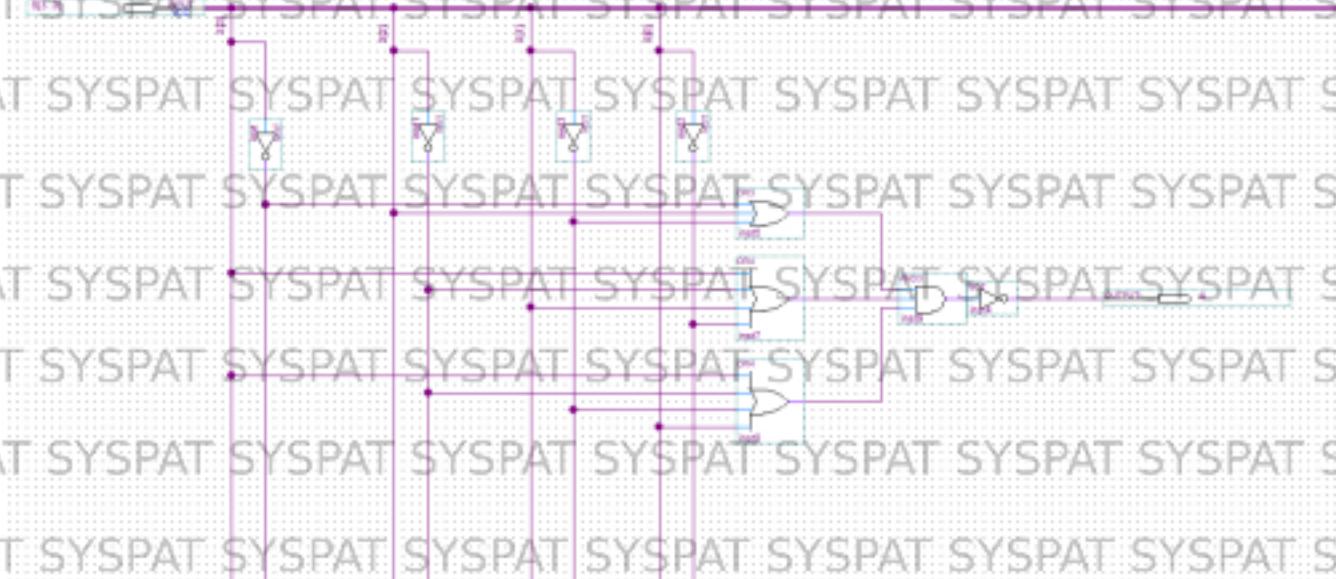


Display BCD Completo Waveform

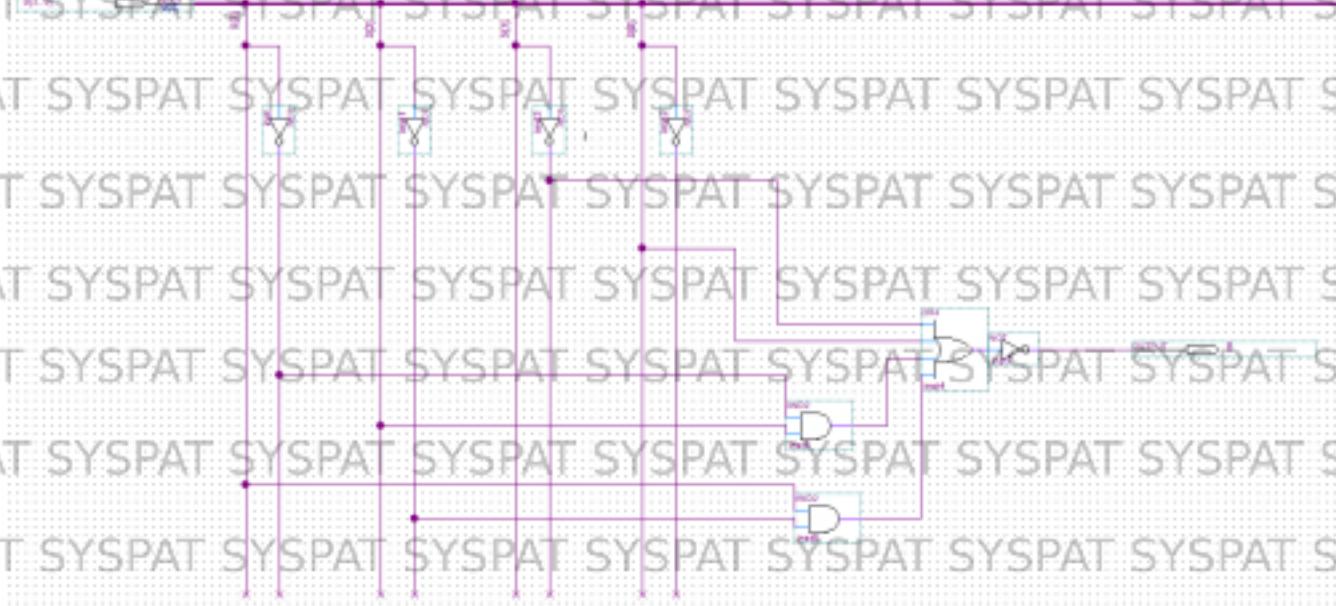


Display ULA

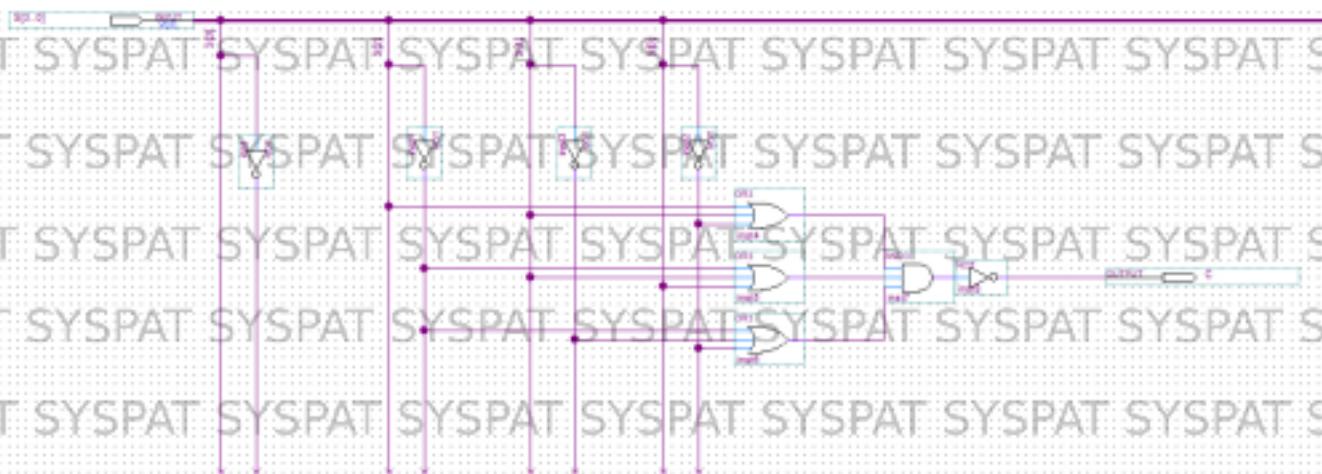




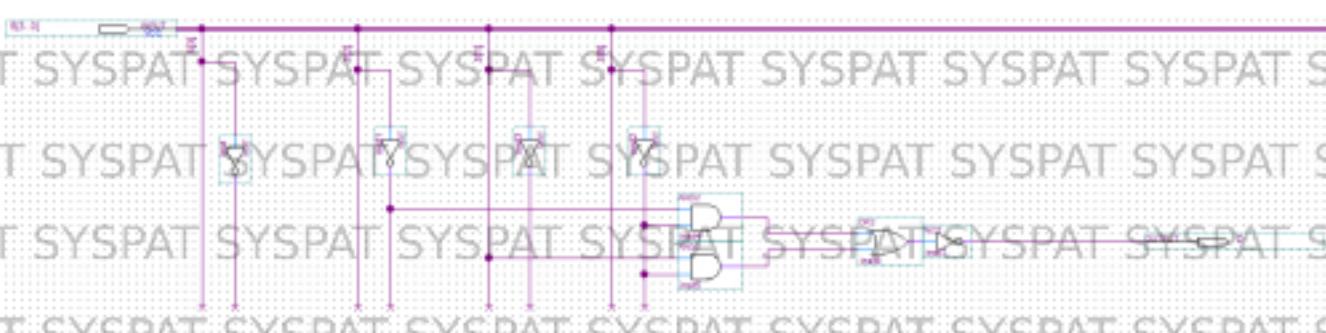
Display A



Display B



Display C

Display D

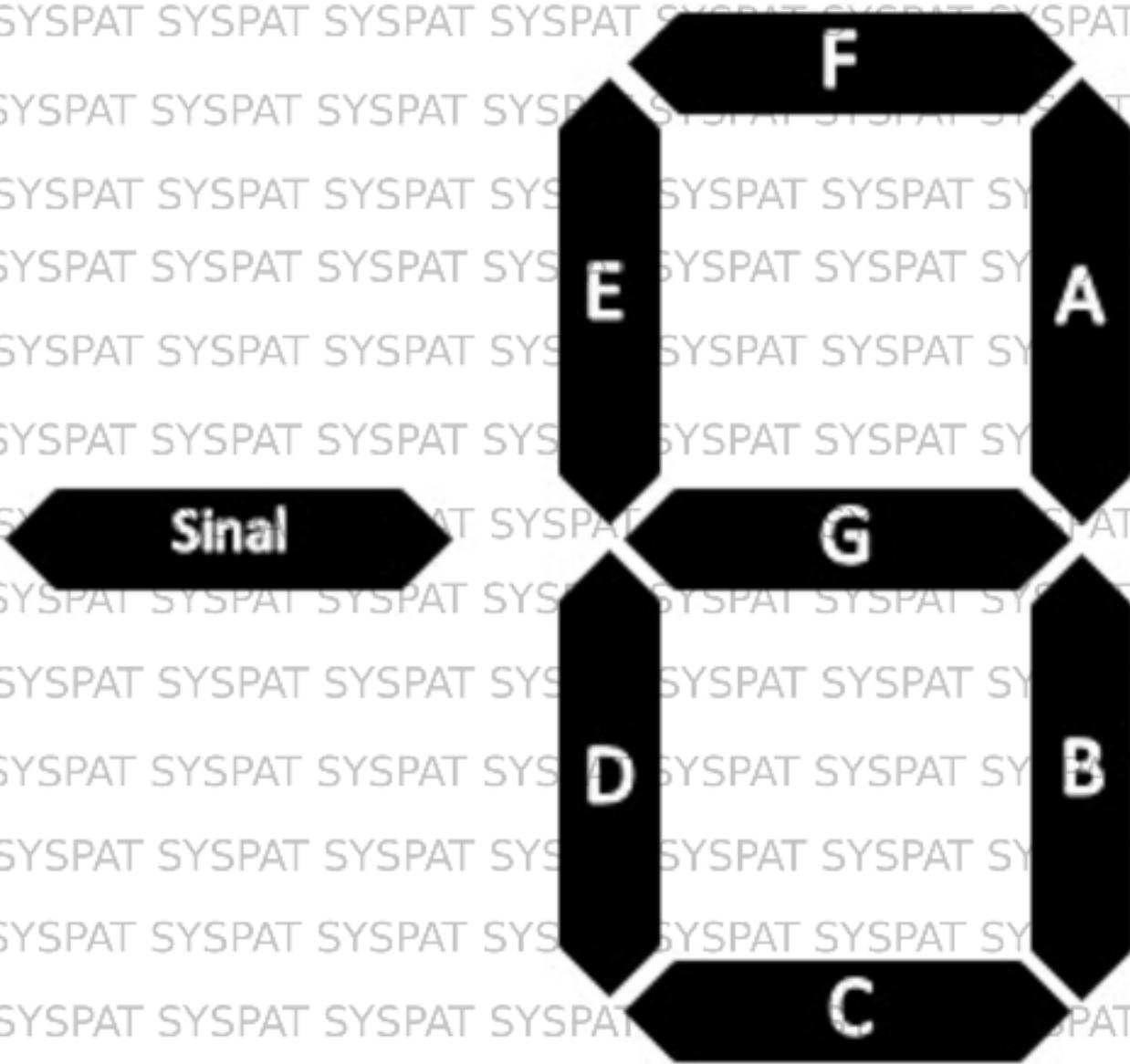
~~42/47~~

Display E

Display F

~~44/47~~

Display G



Esquema Display

A

B

S

0

0

0

0

1

0

1

0

1

1

1

0

Maior que Tabela verdade

UNIDADE LÓGICA E ARITMÉTICA INTEIRA PARA PROCESSAMENTO DE 4 BITS.

RESUMO

A Unidade lógica e aritmética utilizada para microprocessadores, se trata de um dispositivo composto por diferentes estruturas, todas essas capazes de realizar processos bit a bit em números de até 4 bits, utilizando apenas portas lógicas em toda a estrutura da unidade, que podem ser controlados pelo usuário, e retornar o resultado em um display, possibilitando assim à quem o utiliza, que independente da função sendo realizada nos números, se obtenha qualquer resultado de forma extremamente eficiente.