



Eletrônica Digital I

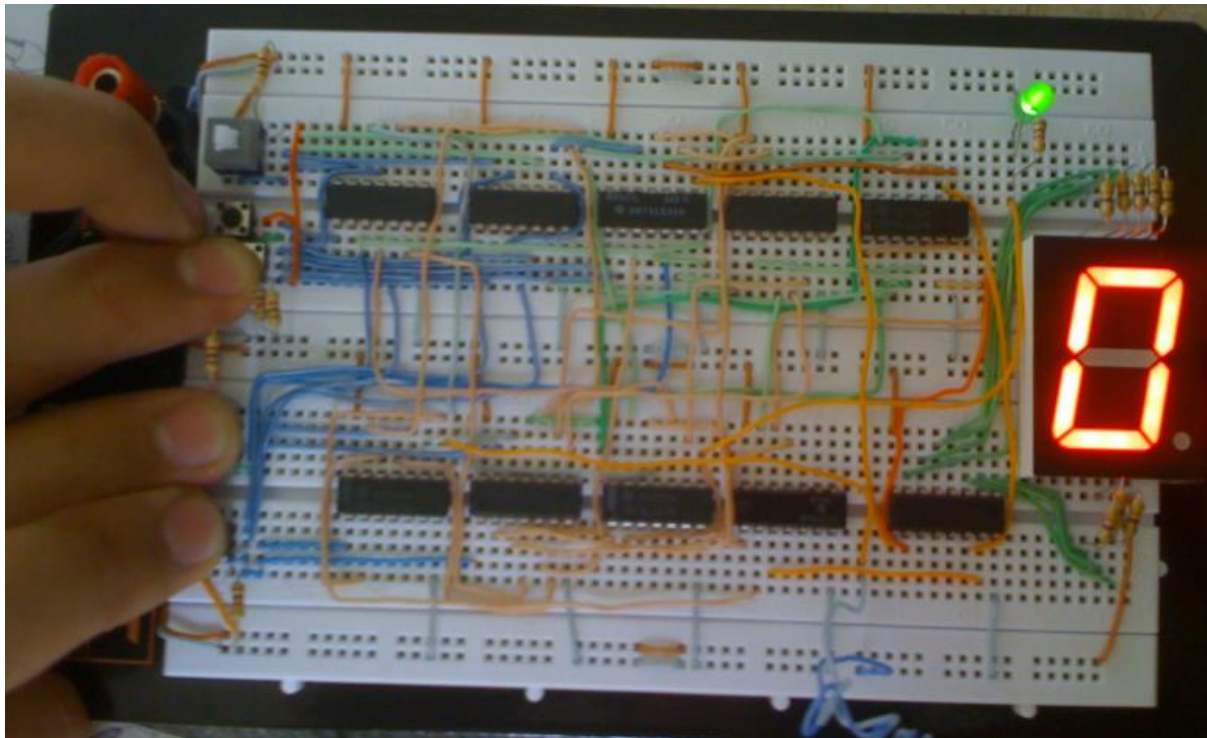
VHDL

Aula K e L – Software VHDL

Prof. MSc. Bruno de Oliveira Monteiro
Engenheiro de Telecomunicações

Inatel

A maior dificuldade nos métodos tradicionais de projeto é a conversão manual da descrição do projeto em um conjunto de equações Booleanas. Esta dificuldade é eliminada com o uso de linguagens de descrição de hardware – HDL (*Hardware Description Languages*). Pode-se, por exemplo, a partir de uma tabela verdade, ou descrição de uma máquina de estado, implementar um circuito usando-se HDL. Dentre as várias HDLs, as mais populares são VHDL, Verilog e Abel.



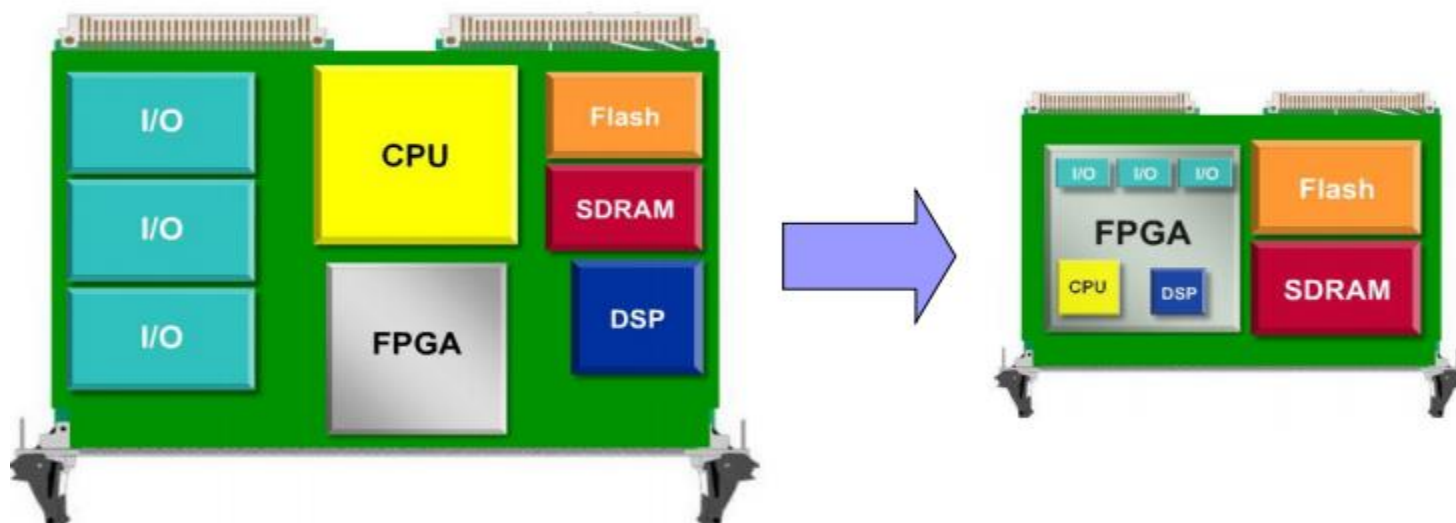
VHDL:

O VHDL foi criado visando simulação, modelagem e documentação, mas acabou recebendo mais tarde a possibilidade de síntese, com o objetivo de se automatizar o projeto de circuitos.

A letra V da palavra VHDL significa Very High Speed Integrated Circuit (Circuito Integrado de Alta Velocidade) e as demais letras, HDL significam Hardware Description Language . Esse nome foi criado pelo departamento de defesa dos Estados Unidos, que foi a primeira instituição a reconhecer a aplicabilidade do VHDL.

O VHDL ganhou popularidade fora do ambiente militar, graças ao IEEE (Institute of Electrical and Electronics Engineering) que estabeleceu padrões para tornar a linguagem universal. Com as ferramentas atuais, pode-se especificar um circuito a partir de seu comportamento ou de sua estrutura, em vários níveis.

Benefício da Lógica Programável



- ☐ Uso de poucos dispositivos
- ☐ Menor custo
- ☐ Economia de energia
- ☐ Flexibilidade
- ☐ Reprogramabilidade “in-system” (em alguns casos)
- ☐ Verificação funcional facilitada
- ☐ Segurança de projeto (previne engenharia reversa)
- ☐ Ferramentas de automatização facilitam o projeto

Exemplo do FPGA SPARTAN-3E do fabricante XILINX



ESTRUTURA DO VHDL:

1. Biblioteca

library (biblioteca)

2. Entidade

entity

3. Arquitetura

architecture

ESTRUTURA DO VHDL:

1. Biblioteca :

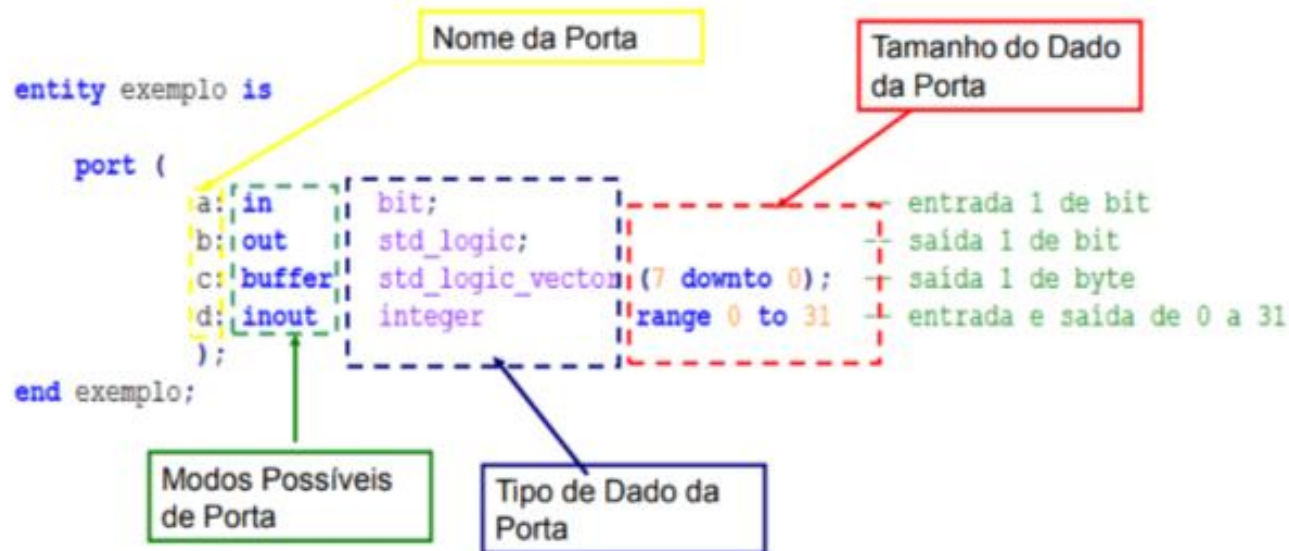
A biblioteca deve ser previamente definida, antes do início da entity (entidade). O uso do package é feito por meio de duas declarações: library e use.

```
LIBRARY IEEE;  
USE IEEE.std_logic_1164.ALL;  
ENTITY cmpl_sig IS  
    PORT (  
        a, b, sel : IN BIT;  
        x, y, z : OUT BIT  
    );  
END ENTITY cmpl_sig;  
ARCHITECTURE logic OF cmpl_sig IS  
BEGIN
```

ESTRUTURA DO VHDL:

2. Entidade :

A entity é a parte principal de qualquer projeto, pois descreve a interface do sistema. Tudo que é descrito na entity fica automaticamente visível a outras unidades associadas com a entity. O nome do sistema é o próprio nome da entity.



ESTRUTURA DO VHDL:

2. Entidade :

MODOS DE PORTA:

- ☐ IN e OUT: as portas operam exclusivamente como entrada e saída.
- ☐ BUFFER: a porta opera unicamente no modo saída, diferenciando do modo OUT porque o valor apresentado pode ser referenciado internamente pela arquitetura. Uma porta no modo OUT não pode por exemplo, controlar um sinal interno da entidade .
- ☐ INOUT: caracteriza uma porta bidirecional.

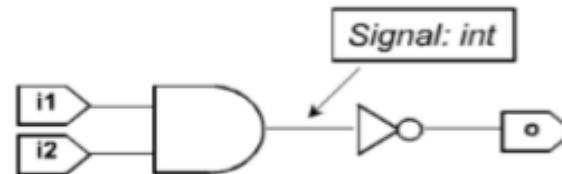
ESTRUTURA DO VHDL:

3. Arquitetura :

A arquitetura trata de toda a lógica programável do sistema, atribuindo funções lógicas as portas declaradas na entidade. Os valores entre a saída de uma porta e a entrada da próxima, são chamados de SIGNALS (sinais) e são declarados na arquitetura.

```
ENTITY simp IS
  PORT (
    i1, i2 : IN BIT;
    o : OUT BIT
  );
END ENTITY simp;

ARCHITECTURE logic OF simp IS
  SIGNAL int : BIT;
BEGIN
  int <= i1 AND i2;
  o <= NOT int;
END ARCHITECTURE logic;
```



Outra maneira

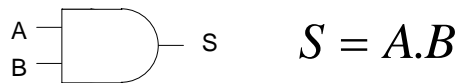
O<= i1 NAND i2

Ou

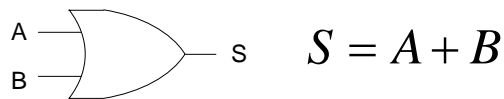
O<= NOT (i1 AND i2)

Blocos Lógicos Básicos

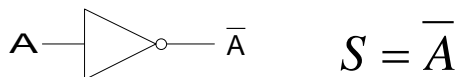
- Função Lógica E



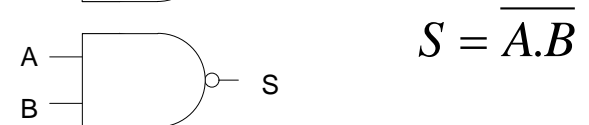
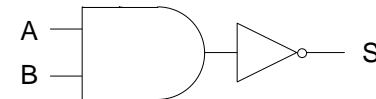
- Função Lógica OU



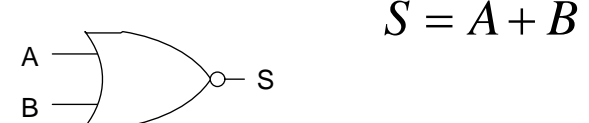
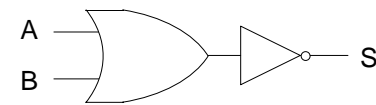
- Função Lógica Inversora (Não)



- Função Lógica Não E (NE)

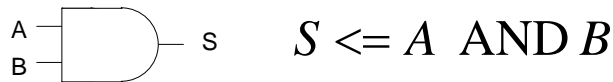


- Função Lógica Não OU (NOU)

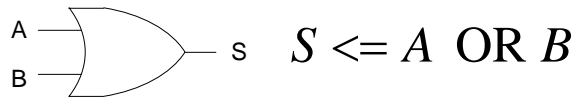


Blocos Lógicos Básicos em VHDL

- Função Lógica E



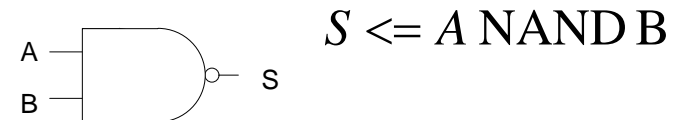
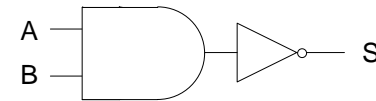
- Função Lógica OU



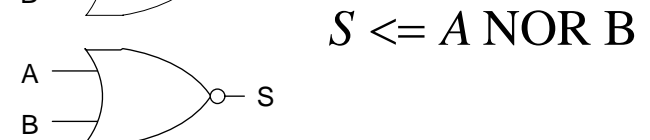
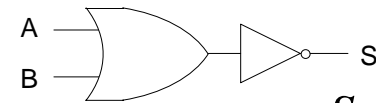
- Função Lógica Inversora (Não)



- Função Lógica Não E (NE)

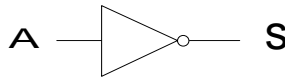


- Função Lógica Não OU (NOU)

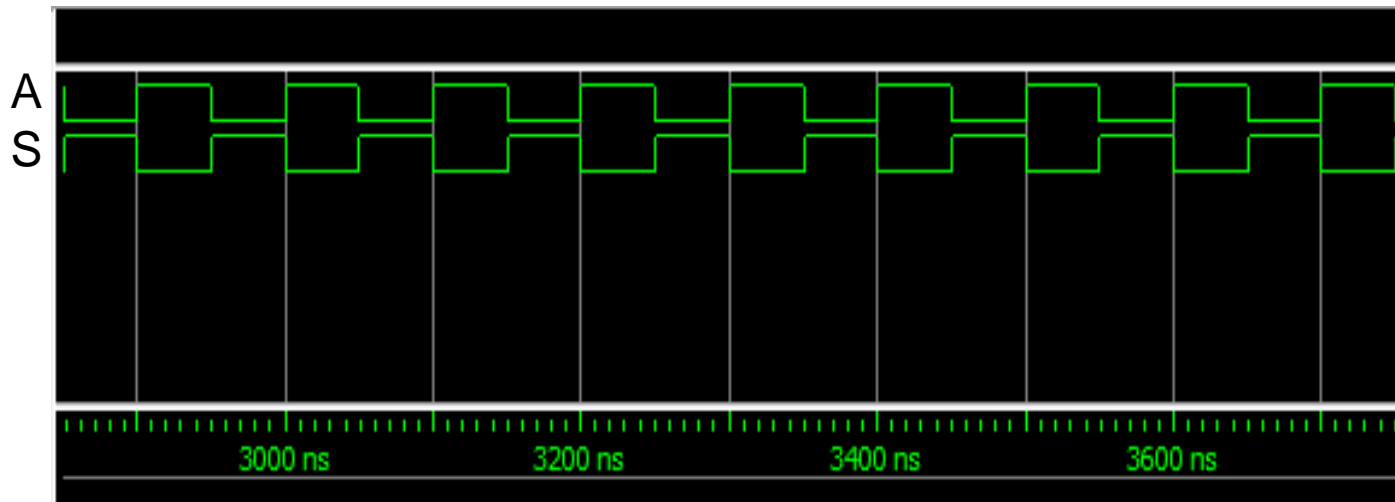


Forma de onda utilizando Software ModelSim:

- Função Lógica Inversora (Não)

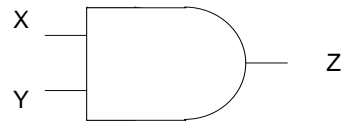


$S \leq \text{NOT } A$

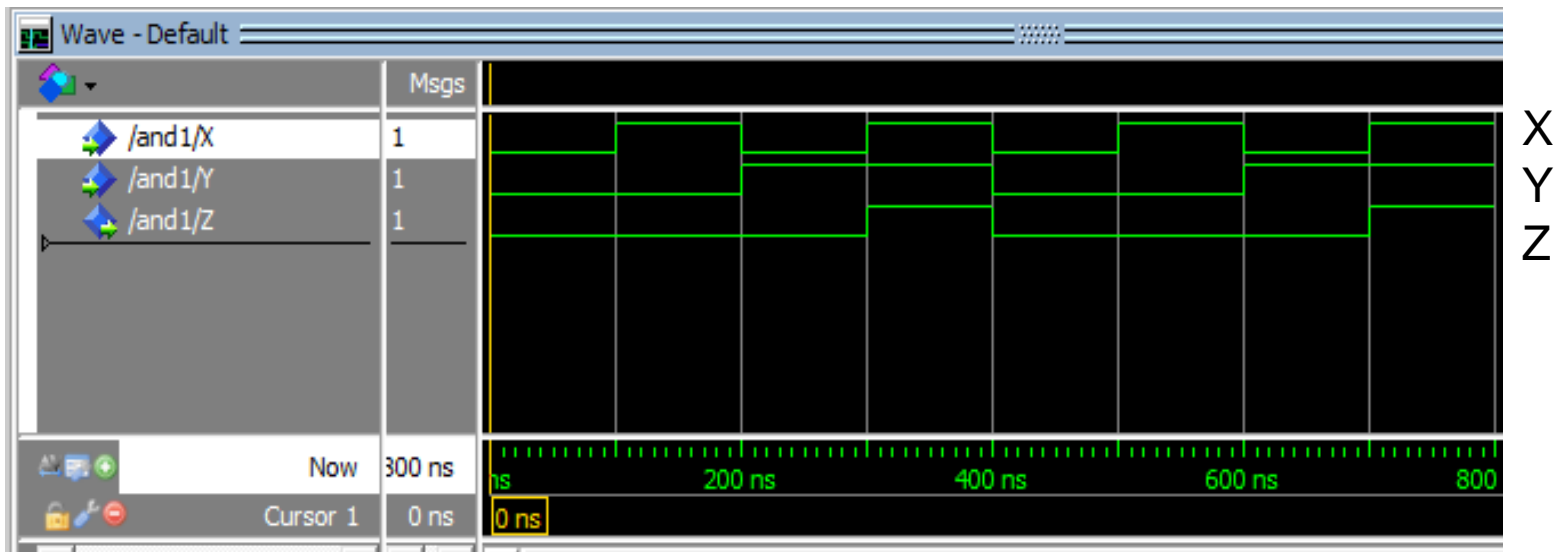


Forma de onda utilizando Software ModelSim:

- Função Lógica E

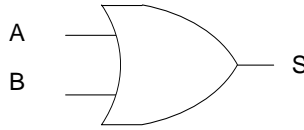


$Z \leq X \text{ AND } Y$

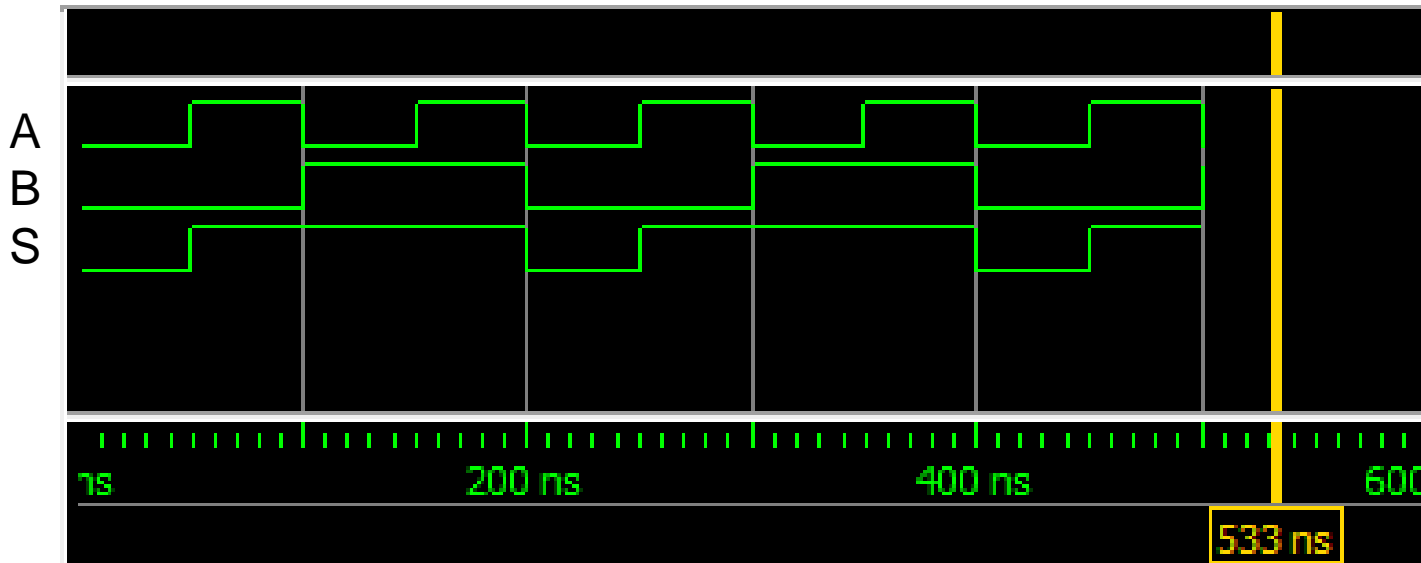


Forma de onda utilizando Software ModelSim:

- Função Lógica OU

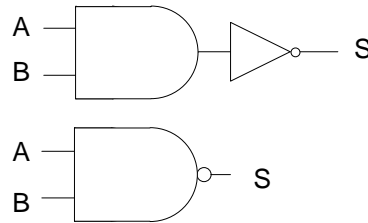


$S \leq A \text{ OR } B$

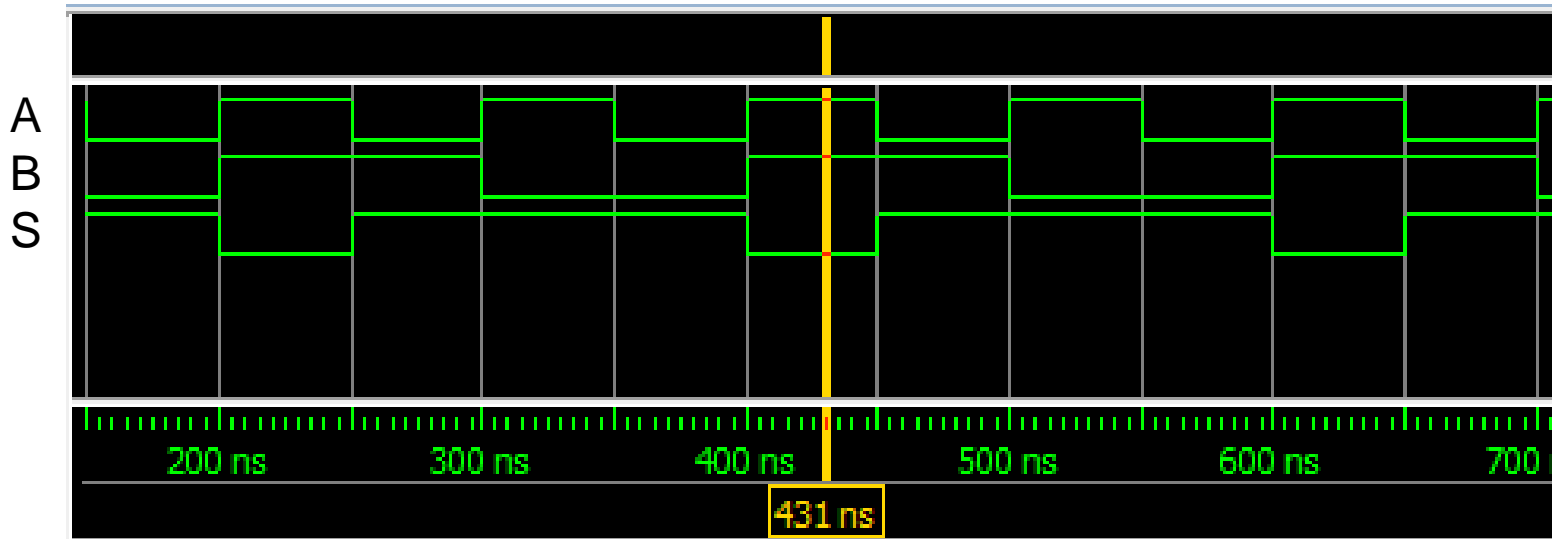


Forma de onda utilizando Software ModelSim:

- Função Lógica Não E (NE)

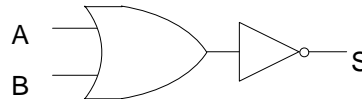


$S \leq A \text{ NAND } B$

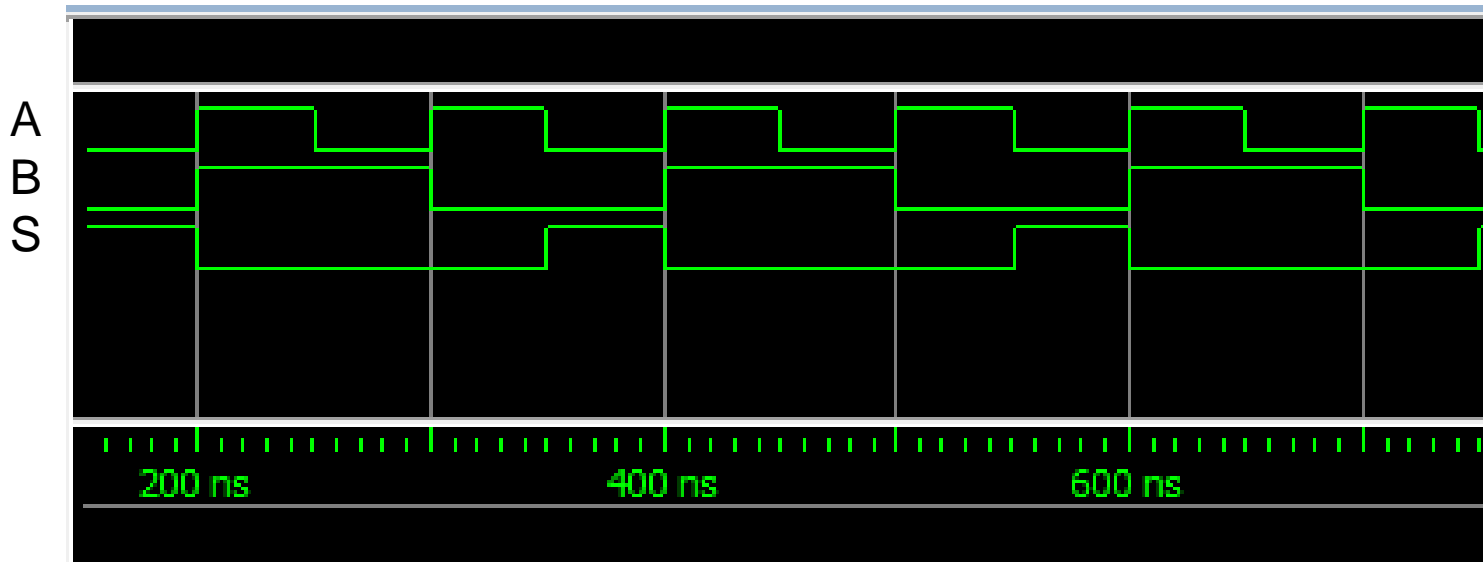
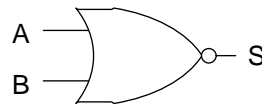


Forma de onda utilizando Software ModelSim:

- Função Lógica Não OU (NOR)



$$S \leq A \text{ NOR } B$$



Funções e Portas Lógicas *Inatel*

- Função e Porta Lógica “OU Exclusivo” ou “XOR”: A saída será igual a “1” quando somente uma das entradas forem “1”.

“OU EXCLUSIVO” ou “EXCLUSIVE OR (XOR)”

Tabela Verdade

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

Expressão Booleana

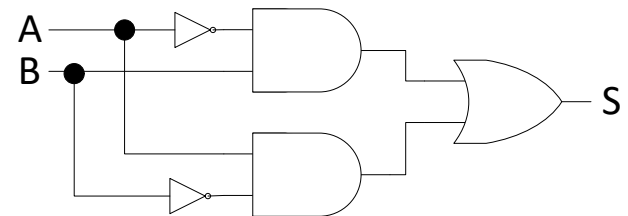
$$S = \bar{A} \cdot B + A \cdot \bar{B}$$

$$S = A \oplus B$$

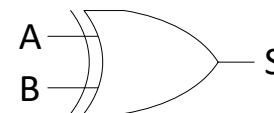
Bloco lógico em VHDL:

$$S <= A \text{ XOR } B$$

Circuito Lógico

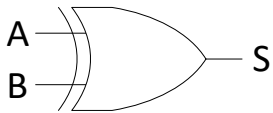


Bloco Lógico

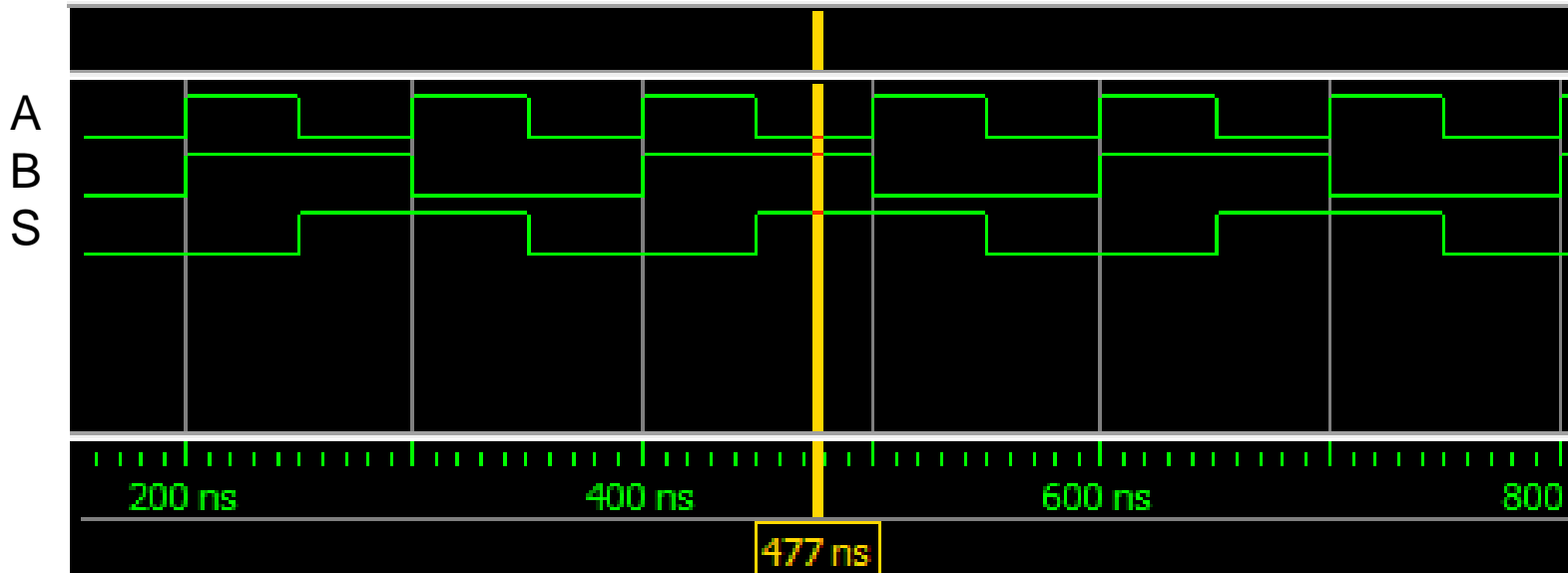


Forma de onda utilizando Software ModelSim:

- Função e Porta Lógica “OU Exclusivo” ou “XOR”:



$$S \leq A \text{ XOR } B$$



Funções e Portas Lógicas

- Função e Porta Lógica “Coincidência” ou “XNOR”: A saída será igual a “1” quando todas as entradas forem iguais.

“NOU EXCLUSIVO” ou “EXCLUSIVE NOR (XNOR)”

Tabela Verdade

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

Expressão Booleana

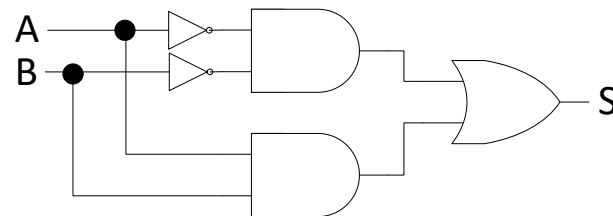
$$S = \bar{A} \cdot \bar{B} + A \cdot B$$

$$S = A \odot B$$

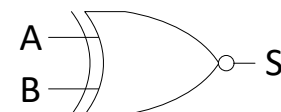
Bloco lógico em VHDL:

$$S <= A \text{ XNOR } B$$

Circuito Lógico

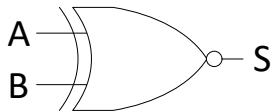


Bloco Lógico

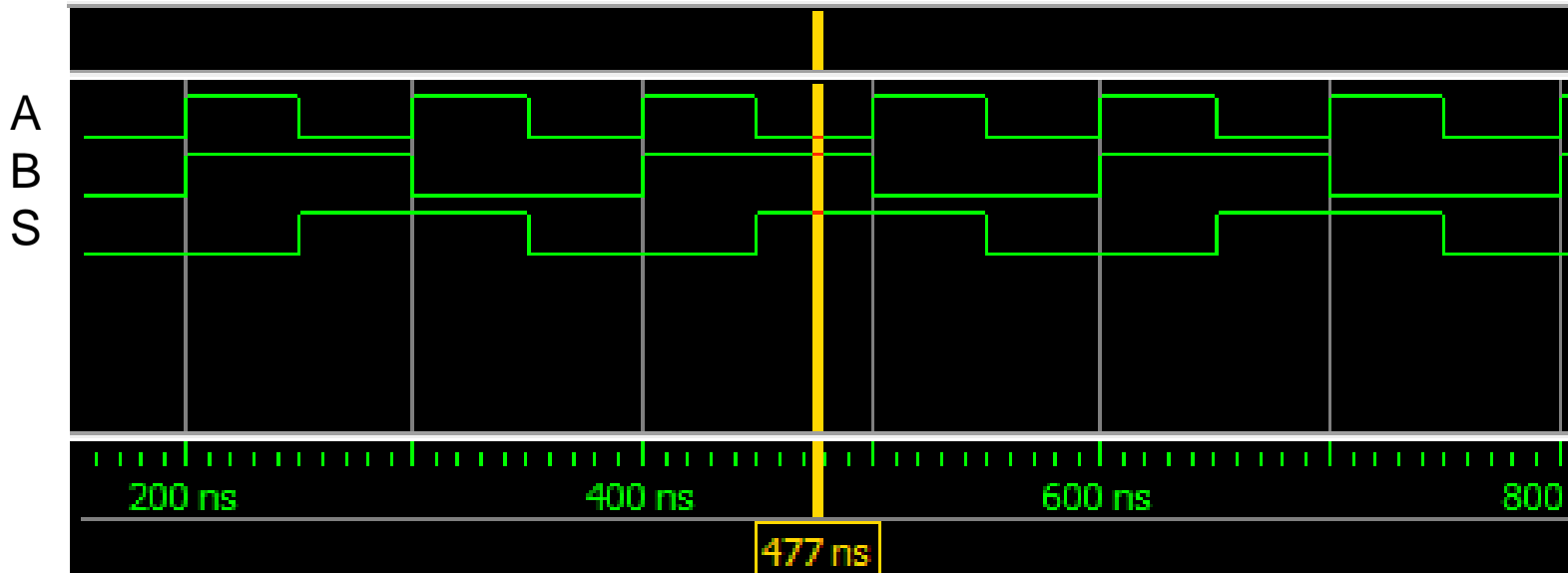


Forma de onda utilizando Software ModelSim:

- Função e Porta Lógica “Coincidência” ou “XNOR”:



$$S \leq A \text{ XNOR } B$$

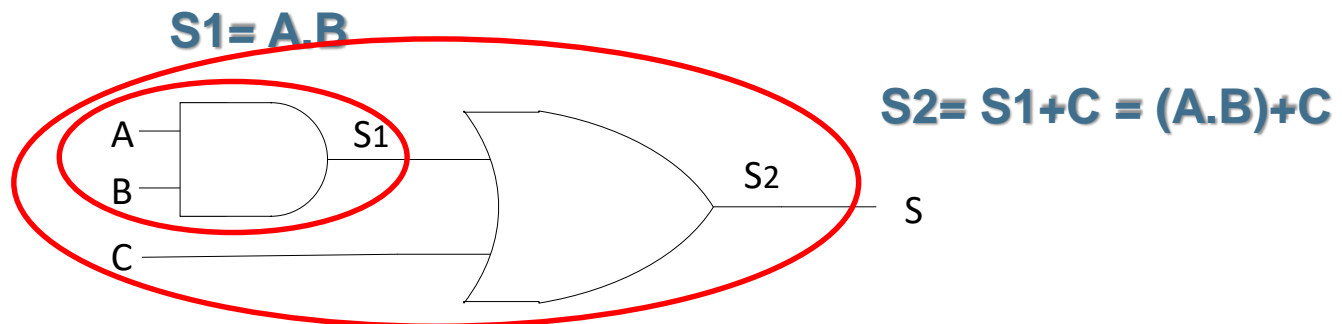


Expressões Booleanas e Circuitos Lógicos

Todo Circuito Lógico é formado a partir da interconexão das portas lógicas básicas.

É possível representar um Circuito Lógico de forma algébrica através das expressões Booleanas

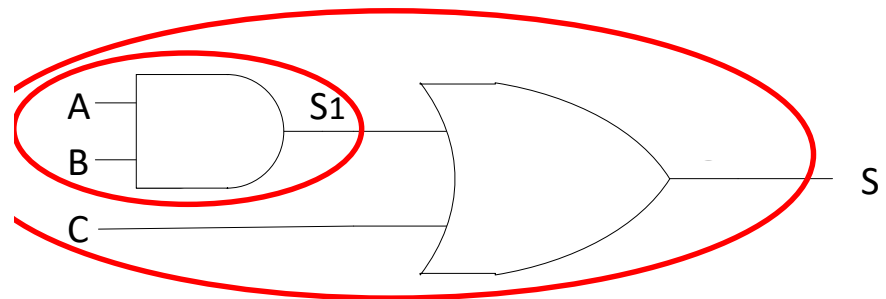
Exemplo:



Em VHDL:

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL
3
4  ENTITY exemplo1 IS
5  PORT (
6      A,B,C : IN BIT;
7      S : OUT BIT;
8  );
9  END ENTITY exemplo1;
10
11
12
13  ARCHITECTURE logic OF exemplo1 IS
14      SIGNAL S1 : BIT;
15  BEGIN
16      S1 <= A AND B;
17      S <= S1 OR C;
18  END ARCHITECTURE logic;
```

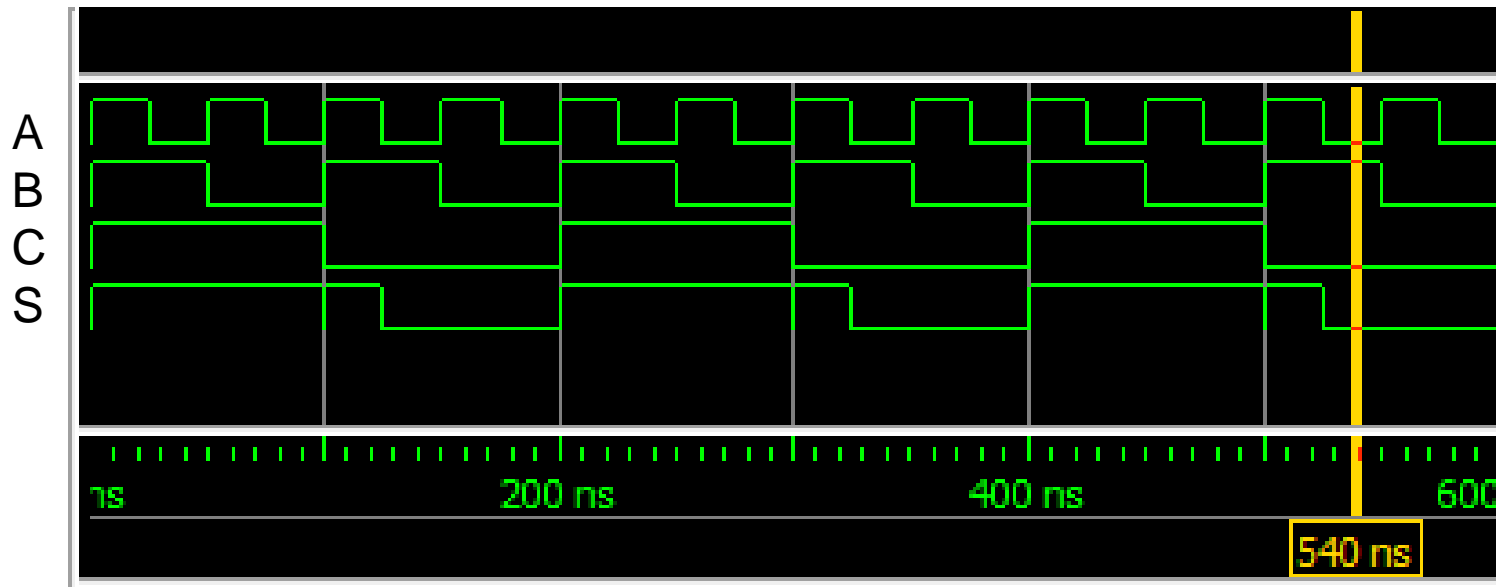
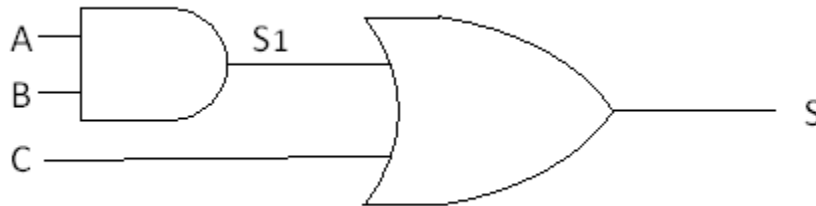
Exemplo1:



Outra maneira:

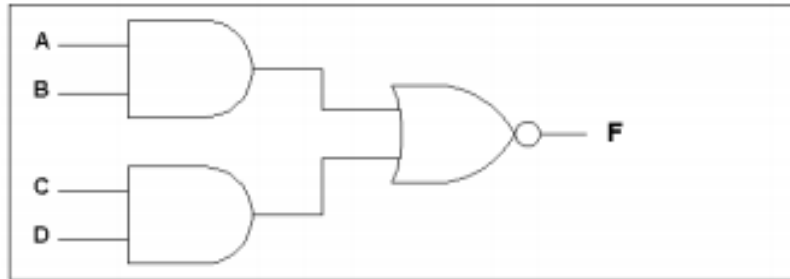
$S \leq (A \text{ AND } B) \text{ OR } C$

Forma de onda utilizando Software ModelSim:



Exercício:

- Descreva em VHDL o circuito da figura abaixo de forma comportamental:



Sintaxe:

```
ENTITY exemplo IS
```

```
    PORT (
        -- declaração das portas
    );
```

```
END exemplo;
```

```
architecture nome_arquitetura of e;
```

```
    -- declarações opcionais (sinais e componentes)    Signal S1,S2: bit;
```

```
begin
```

```
    -- comportamento lógico do circuito
```

```
end;
```

A: IN bit;

B: IN bit;

C: IN bit;

D: IN bit;

F: OUT bit

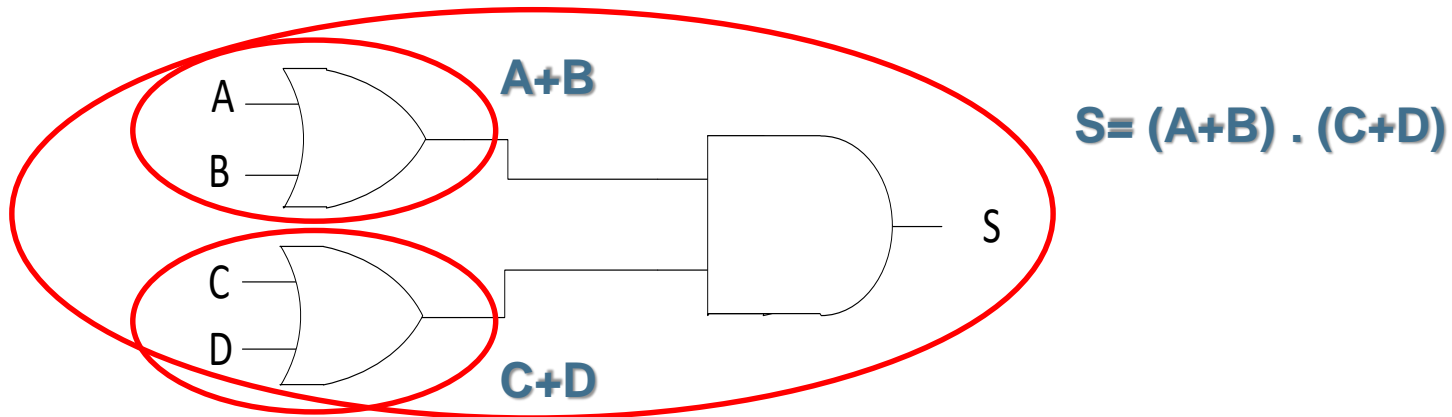
S1<= A and B;

S2<= C and D;

F<= S1 NOR S2;

Expressões Booleanas e Circuitos Lógicos

Exemplo:



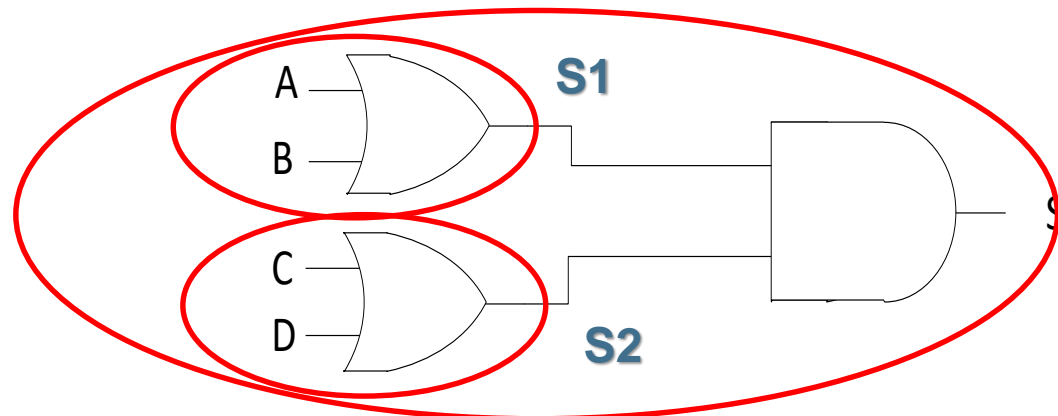
Em VHDL:

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3
4  ENTITY exemplo2 IS
5  PORT (
6      A,B,C,D : IN BIT;
7      S : OUT BIT;
8  );
9  END ENTITY exemplo2;
10
11 ARCHITECTURE logic OF exemplo2 IS
12     SIGNAL S1,S2 : BIT;
13 BEGIN
14     S1 <= A OR B;
15     S2 <= C OR D;
16     S <= S1 AND S2;
17 END ARCHITECTURE logic;
18
19
```

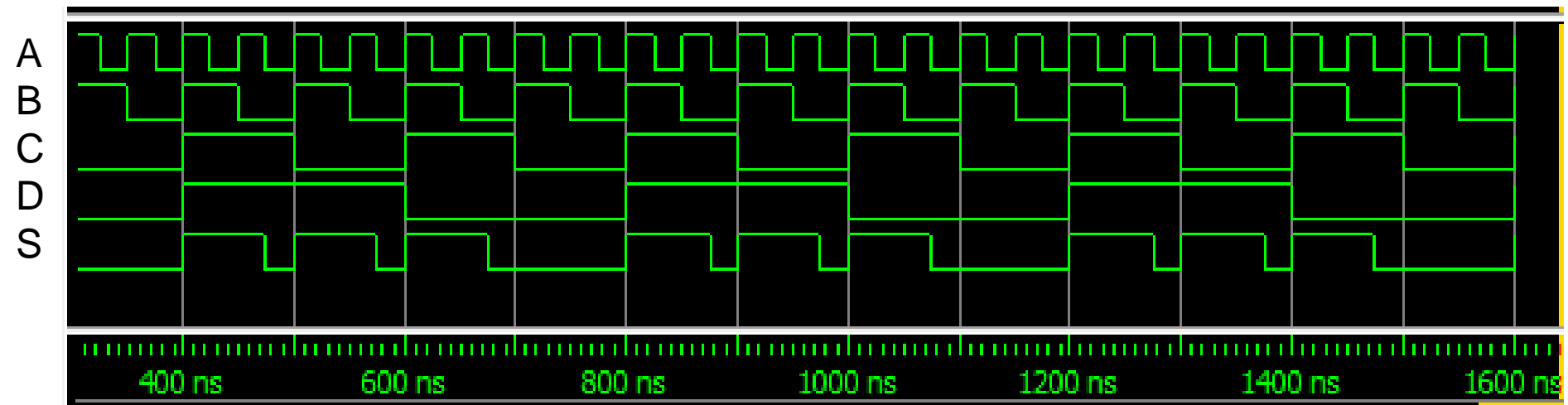
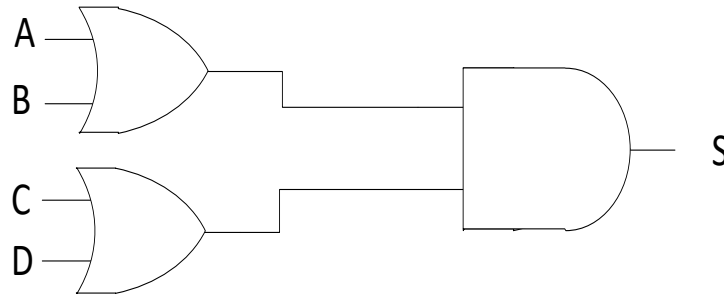
Outra Maneira:

$S \leq (A \text{ OR } B) \text{ AND } (C \text{ OR } D)$

Exemplo2:



Forma de onda utilizando Software ModelSim:

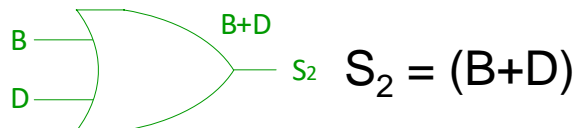
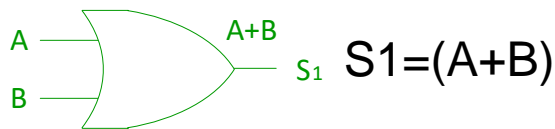


Expressões Booleanas e Circuitos Lógicos

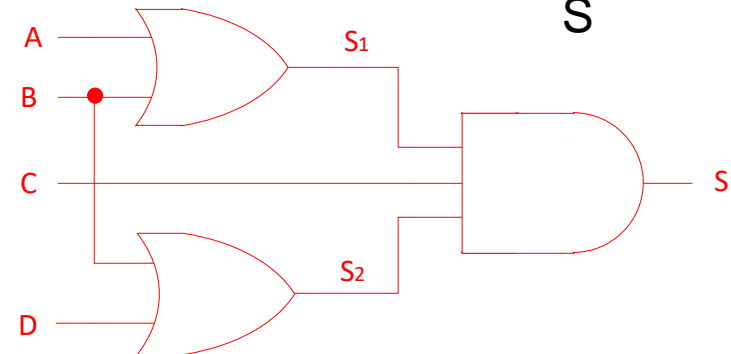
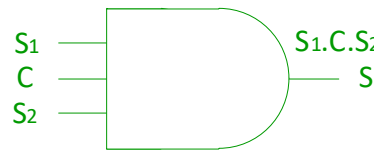
- Circuitos Lógicos obtidos de Expressões Booleanas :

De maneira análoga ao que utilizamos para obter a expressão booleana que um circuito lógico executa, podemos desenhar um circuito lógico que executa a expressão booleana.

Exemplo: O circuito que representa a expressão booleana $S = (A+B) \cdot C \cdot (B+D)$



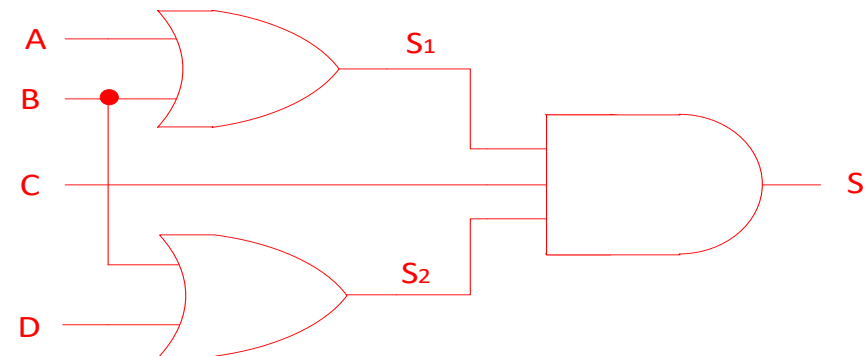
$$S = S_1 \cdot C \cdot S_2$$



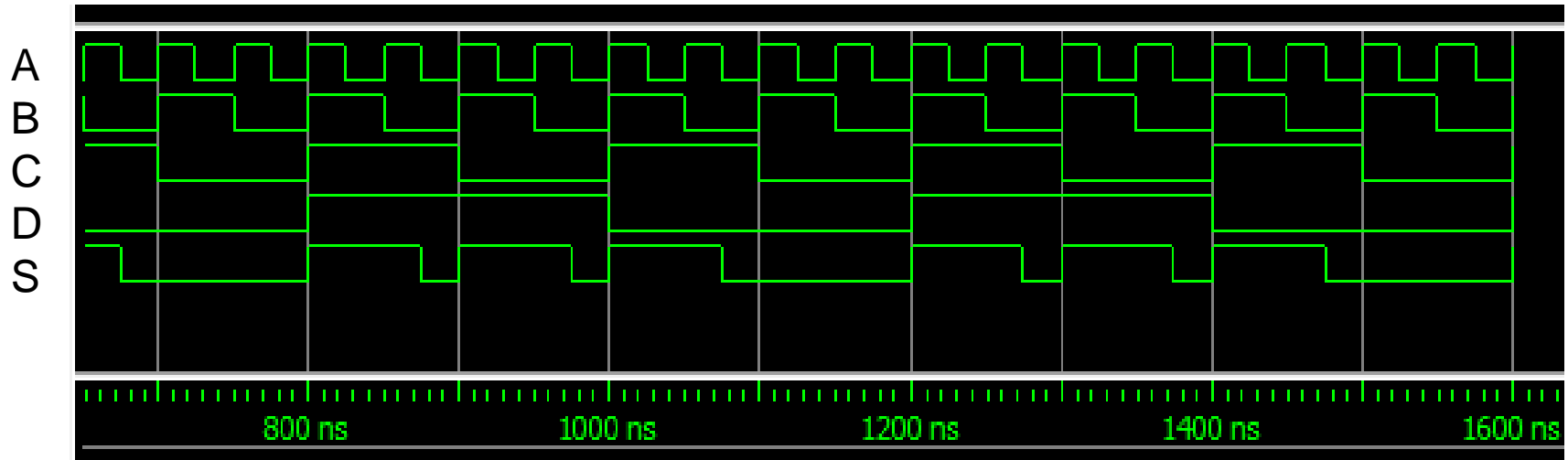
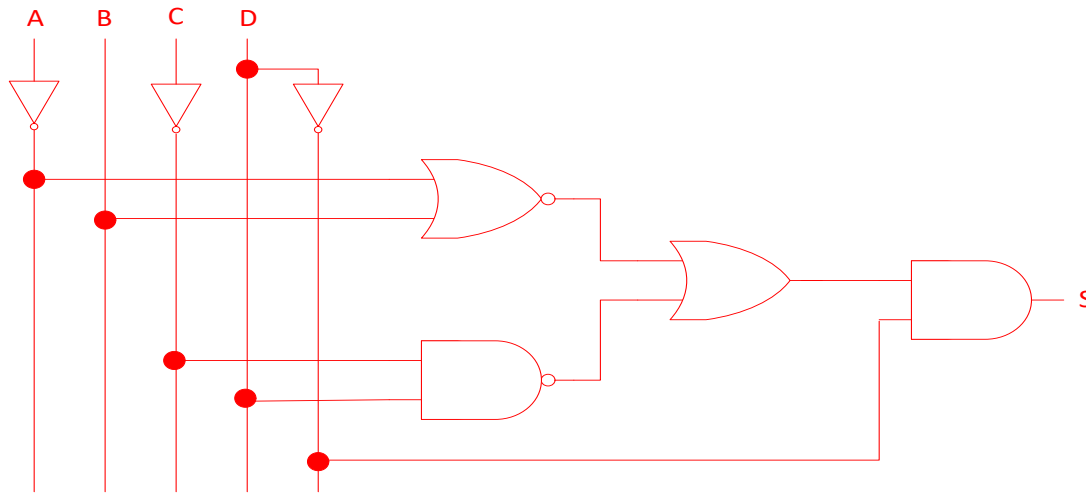
Em VHDL:

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL
3
4  ENTITY exemplo3 IS
5  PORT (
6      A,B,C,D : IN BIT;
7      S : OUT BIT;
8  );
9  END ENTITY exemplo3;
10
11
12  ARCHITECTURE logic OF exemplo3 IS
13      SIGNAL S1,S2 : BIT;
14  BEGIN
15      S1 <= A OR B;
16      S2 <= B OR D;
17      S <= S1 AND C AND S2;
18  END ARCHITECTURE logic;
```

Exemplo3:



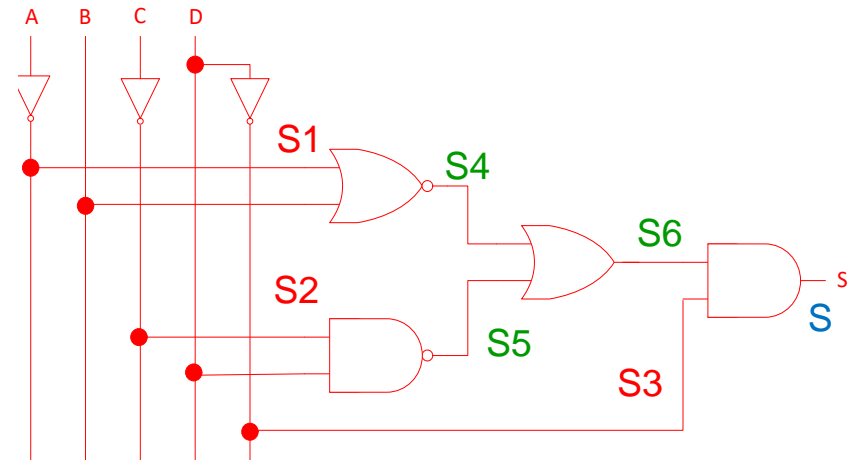
Forma de onda utilizando Software ModelSim:



Em VHDL:

Exemplo4:

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL
3
4  ENTITY exemplo4 IS
5  PORT (
6      A,B,C,D : IN BIT;
7      S : OUT BIT;
8  );
9  END ENTITY exemplo4;
10
11 ARCHITECTURE logic OF exemplo4 IS
12     SIGNAL S1,S2,S3,S4,S5,S6 : BIT;
13 BEGIN
14     S1 <= NOT A;
15     S2 <= NOT C;
16     S3 <= NOT D;
17     S4 <= S1 NOR B;
18     S5 <= S2 NAND D;
19     S6 <= S4 OR S5;
20     S <= S6 AND S3
21 END ARCHITECTURE logic;
```





Bons Estudos

Prof. MSc. Bruno de Oliveira Monteiro
Engenheiro de Telecomunicações

Inatel