

Projeto com Circuitos Reconfiguráveis

Projeto lógico combinacional Portas lógicas

Prof. Daniel M. Muñoz Arboleda

FGA - UnB

Sistemas combinacionais e sequenciais

Um *sistema digital combinacional* é qualquer sistema digital onde o comportamento de cada saída pode ser descrito como uma função que depende exclusivamente das **combinações** de valores instantâneos das entradas do sistema. Um sistema digital combinacional pode ser totalmente descrito por uma *tabela verdade*.

Um *sistema digital sequencial* é um sistema digital que, em geral, não pode ser descrito exclusivamente pela combinação das entradas. Portanto, é um sistema digital que sob as mesmas condições possui mais de um estado, isto é, depende dos valores passados das entradas (memória).

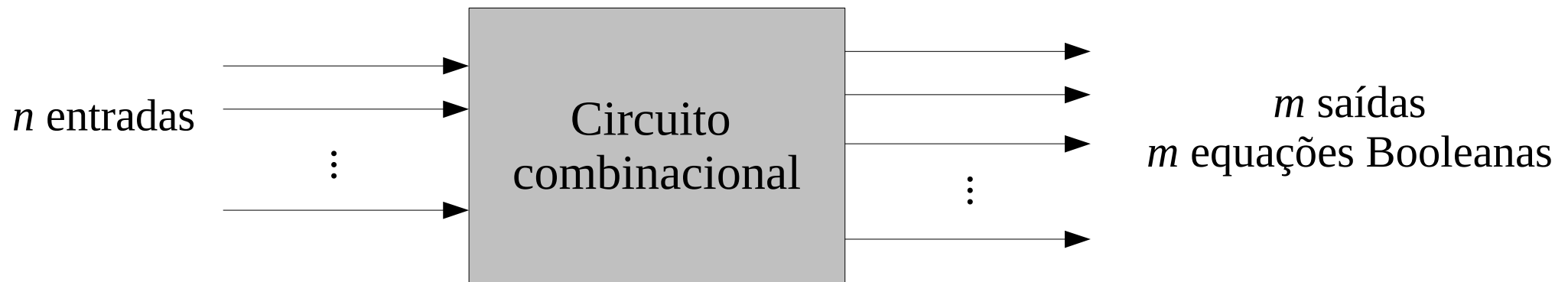
Um sistema combinacional é definido como um caso especial de um sistema sequencial.

Circuito combinacional

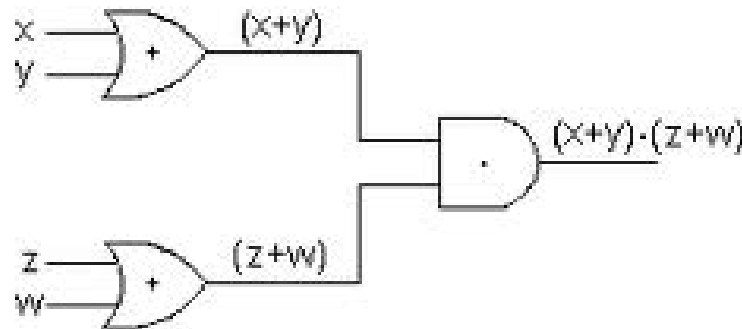
Nos circuitos lógicos combinacionais a qualquer instante o nível lógico da saída depende da combinação dos níveis lógicos presentes na entrada. Um circuito combinacional não possui características de memória e, portanto, a sua saída depende apenas do valor regular das entrada (não depende de valores passados de entradas ou saídas ou estados).

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.

Circuito combinacional



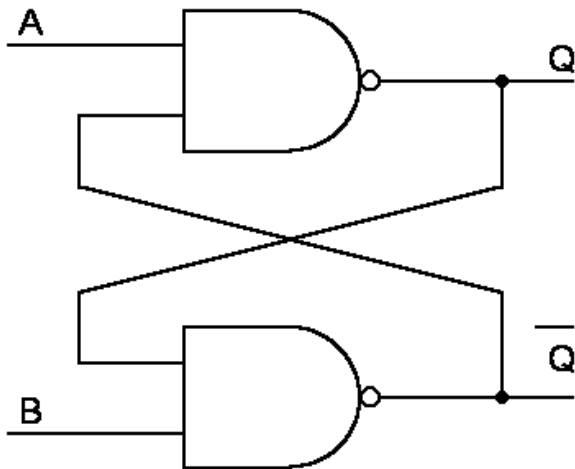
Pode-se dizer que um circuito combinacional realiza uma operação de processamento de informação a qual pode ser especificada por meio de um conjunto de equações Booleanas.



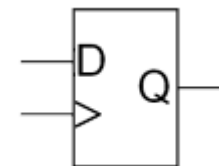
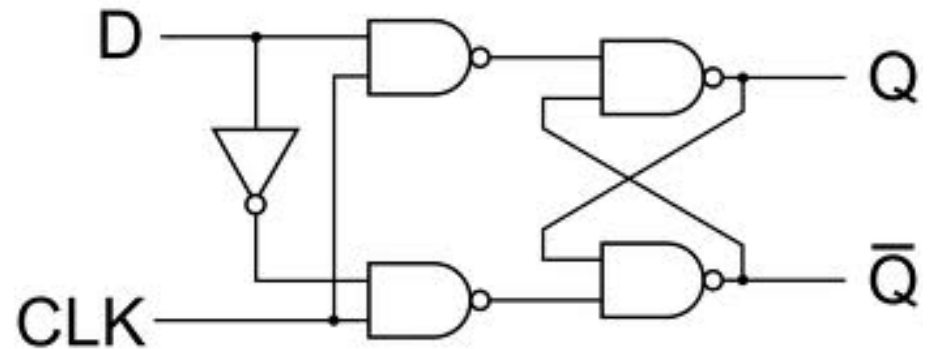
Circuito sequencial

Um circuito sequencial, por sua vez, emprega elementos de armazenamento denominados latches e flip-flops, além de portas lógicas. Os valores das saídas do circuito dependem dos valores das entradas e dos estados dos latches ou flip-flops utilizados.

Latch SR

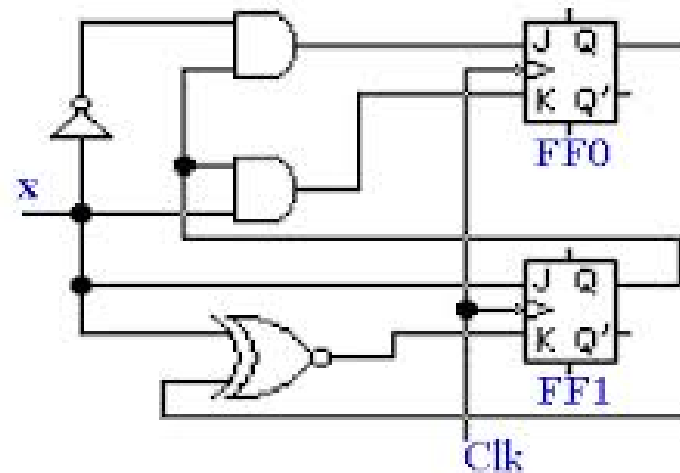
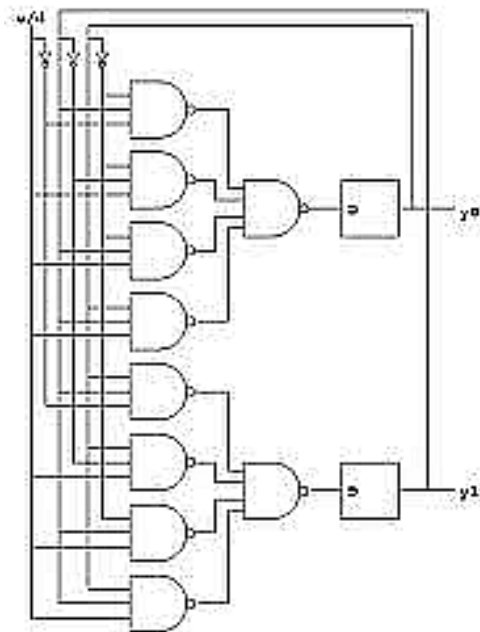


Flip-flop D



Circuito sequencial

Como os estados dos latches e flip-flops é função dos valores anteriores das entradas, diz-se que as saídas de um circuito sequencial dependem dos valores das entradas e do histórico do próprio circuito. Logo, o comportamento de um circuito sequencial é especificado pela sequência temporal das entradas e de seus estados internos.



VHDL

O que é VHDL? **V**ery High Speed Integrated Circuits **H**ardware **D**escription **L**anguage

VHDL é uma linguagem de descrição de *hardware*. Descreve o comportamento de um circuito eletrônico ou sistema, a partir do qual o circuito físico pode ser implementado.

História:

- VHSIC (Very High Speed Integrated Circuits) foi uma iniciativa do Departamento de Defesa dos Estados Unidos (DARPA), em meados da década de 1980, para documentar o comportamento de ASICs.
- Isto levou ao aparecimento da linguagem VHDL, desenvolvida para substituir os complexos manuais que descreviam o funcionamento dos ASICs.
- A linguagem VHDL foi posta em domínio público e foi padronizada pela IEEE no ano 1987.
- IEEE 1164 é um pacote VHDL com multi valores lógicos (`std_logic_1164`)
- Alterações e aprimoramento da linguagem. IEEE lança um padrão atualizado no ano 1993.
- Em 2008 foi aprovada a mais recente versão, IEEE 1076-2008, que considera extensões para sinais analógicas e mistas, VHDL-AMS (analog and mixed signal).

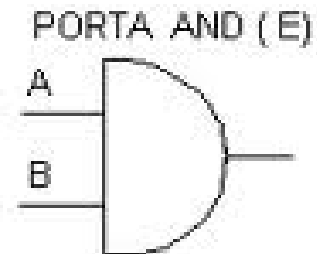
Estrutura do VHDL

Consultar livro Circuit Design with VHDL, Volnei Pedroni, 2004

Projeto lógico combinacional em VHDL

Projeto 1. Porta AND de duas entradas.
Simulação no ISim e Implementação na Spartan3.

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity and2 is
5  port(
6      a, b: in std_logic;
7      s  : out std_logic);
8  end and2;
9
10 architecture comportamental of and2 is
11 begin
12     s <= a and b;
13
14 end comportamental;
```



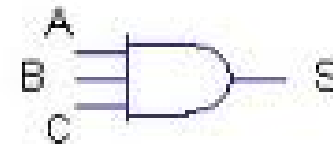
$X = A \cdot B$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Desenho lógico combinacional em VHDL

Projeto 2. Porta AND de três entradas
Simulação no ISim e Implementação na Spartan3.

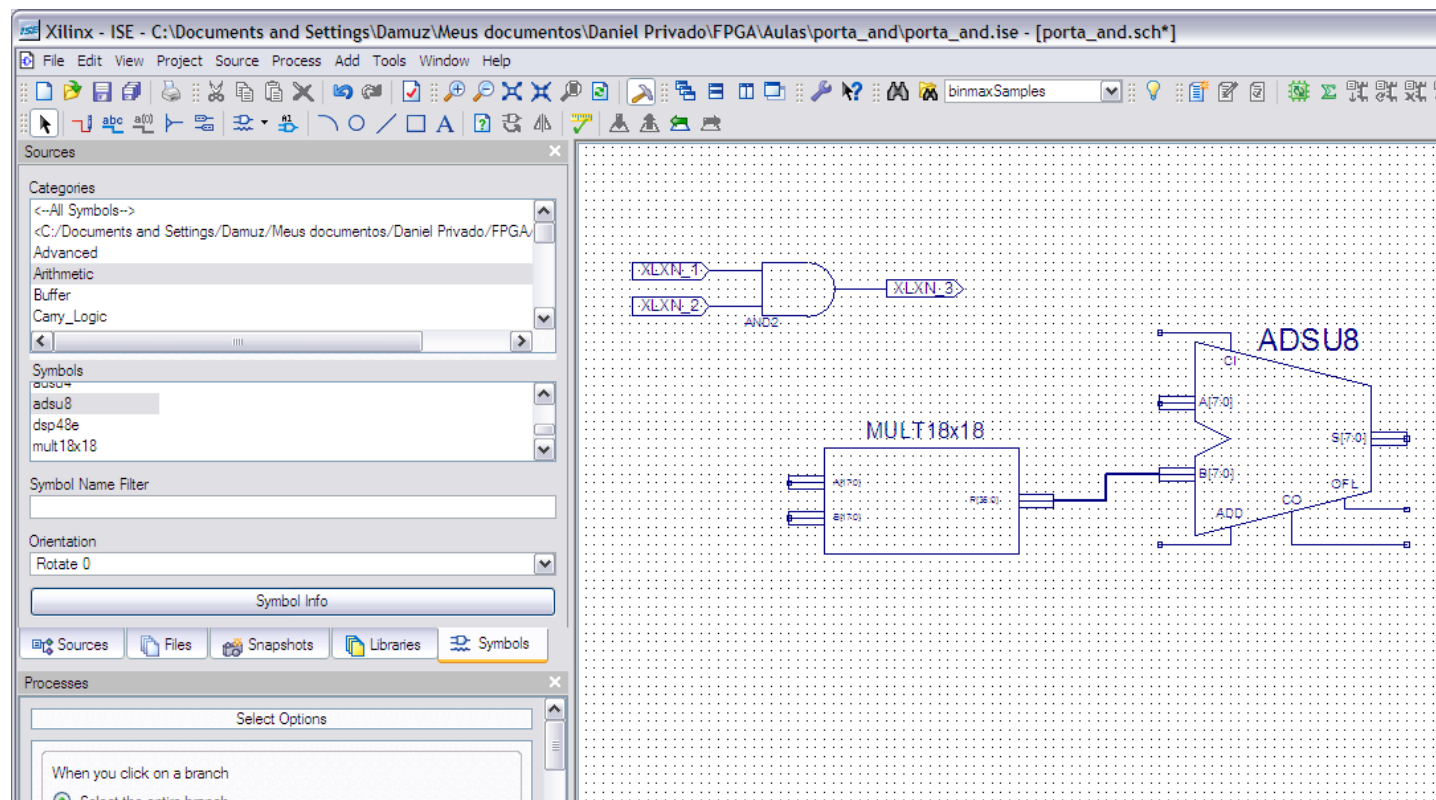
```
1  LIBRARY ieee;  
2  USE ieee.std_logic_1164.all;  
3  
4  entity and3 is port(  
5      a, b, c: in std_logic;  
6      s: out std_logic);  
7  end and3;  
8  
9  architecture comportamental of and3 is  
10 begin  
11     s <= (a and b) and c;  
12  
13 end comportamental;  
14
```



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

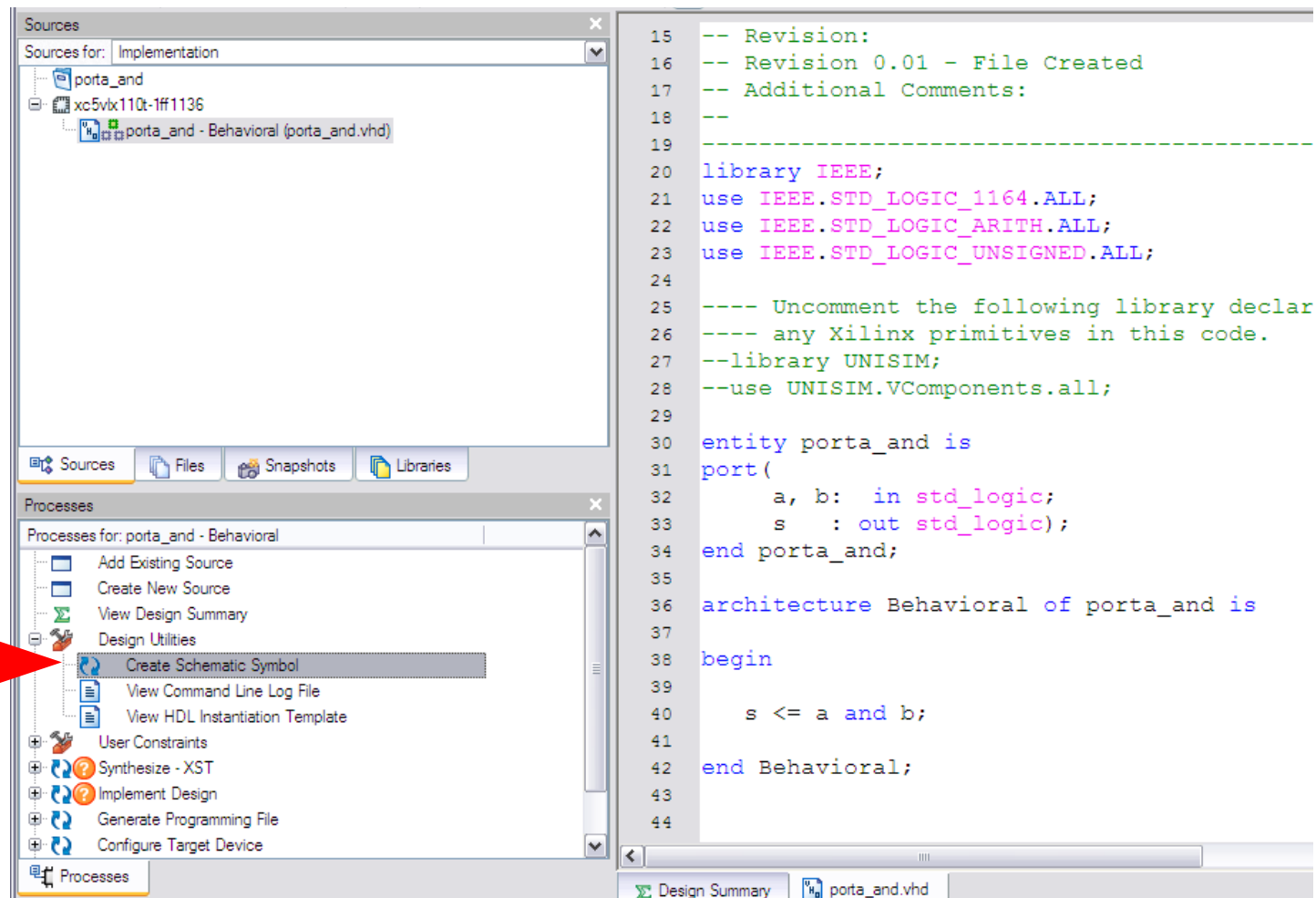
Desenho lógico combinacional em VHDL

Projeto 3. Editor de esquemáticos do ISE



Desenho lógico combinacional em VHDL

Projeto 3. Criação de símbolos esquemáticos



The screenshot displays the Xilinx ISE software interface. The **Sources** window on the left shows the project structure for 'Implementation', including a file named 'porta_and - Behavioral (porta_and.vhd)'. The **Processes** window on the left lists various tasks, with 'Create Schematic Symbol' highlighted by a red arrow. The right pane shows the VHDL code for the 'porta_and' entity, which implements a 2-input AND gate.

```
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  -----
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  ---- Uncomment the following library declar
26  ---- any Xilinx primitives in this code.
27  --library UNISIM;
28  --use UNISIM.VComponents.all;
29
30  entity porta_and is
31  port(
32      a, b:  in std_logic;
33      s   :  out std_logic);
34  end porta_and;
35
36  architecture Behavioral of porta_and is
37
38  begin
39
40      s <= a and b;
41
42  end Behavioral;
43
44
```

Descrição VHDL porta AND2

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity and2 is
5  port(
6      a, b: in std_logic;
7      s   : out std_logic);
8  end and2;
9
10 architecture comportamental of and2 is
11 begin
12     s <= a and b;
13
14 end comportamental;
```

Descrição VHDL porta AND3

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity and3 is port(
5      a, b, c: in std_logic;
6      s: out std_logic);
7  end and3;
8
9  architecture comportamental of and3 is
10 begin
11     s <= (a and b) and c;
12
13 end comportamental;
```

Descrição VHDL porta NOT

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity inv is port(
5      a: in std_logic;
6      s: out std_logic);
7  end inv;
8
9  architecture comportamental of inv is
10 begin
11     s <= not a;
12
13 end comportamental;
```

Descrição VHDL porta NAND2

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity nand2 is port(
5      a, b: in std_logic;
6      s: out std_logic);
7  end nand2;
8
9  architecture comportamental of nand2 is
10 begin
11     s <= a nand a;
12
13 end comportamental;
```


Descrição VHDL porta XNOR2

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  entity xnor2 is port(
5      a, b: in std_logic;
6      s: out std_logic);
7  end xnor2;
8
9  architecture comportamental of xnor2 is
10 begin
11     s <= a xnor b;
12
13 end comportamental;
```