

# PCS 2304

## Sistemas Digitais II

### Módulo 07 – Dispositivos de Lógica Programável

*versão: 1.04 Spina (outubro de 2017)*

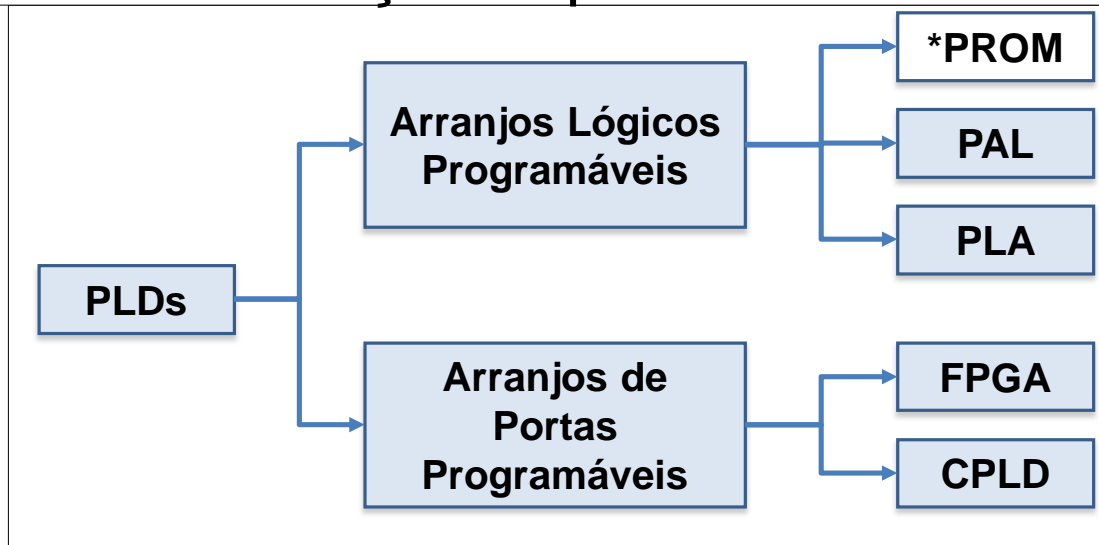
© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 1

## Conteúdo

- **Lógica Programável**
  - Introdução
  - Memórias como PLDs
  - Arranjos Lógicos Programáveis: PLAs e PALs
  - PLDs modernos: CPLDs e FPGAs
  - FPGAs: estrutura
  - Projeto Lógico com FPGAs

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 2

## Introdução: Tipos de PLDs



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 7

## Memórias como PLDs

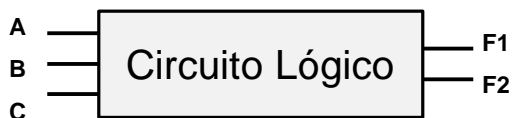
- As primeiras realizações de PLDs foram baseadas em memórias ROM (*Read Only Memory*).
- Posteriormente, outras memórias do tipo ROM (ex.: PROM, EPROM, EEPROM) passaram a ser usadas.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 8

## Memórias como PLDs

- Objetivo:** construir um circuito que implemente a seguinte lógica:

Entradas			Saídas	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1



$$F1 = \overline{A}BC + A\overline{B}C + ABC$$

$$F2 = \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C + ABC$$

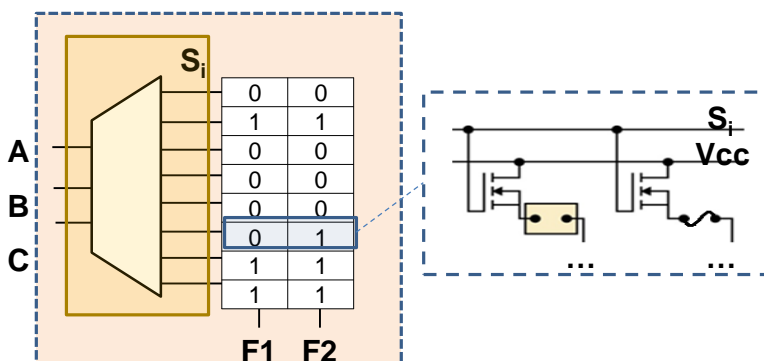
Minimização é fonte de erro !

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 9

## Memórias como PLDs

### Memória PROM 8 x 2

Entradas			Saídas	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

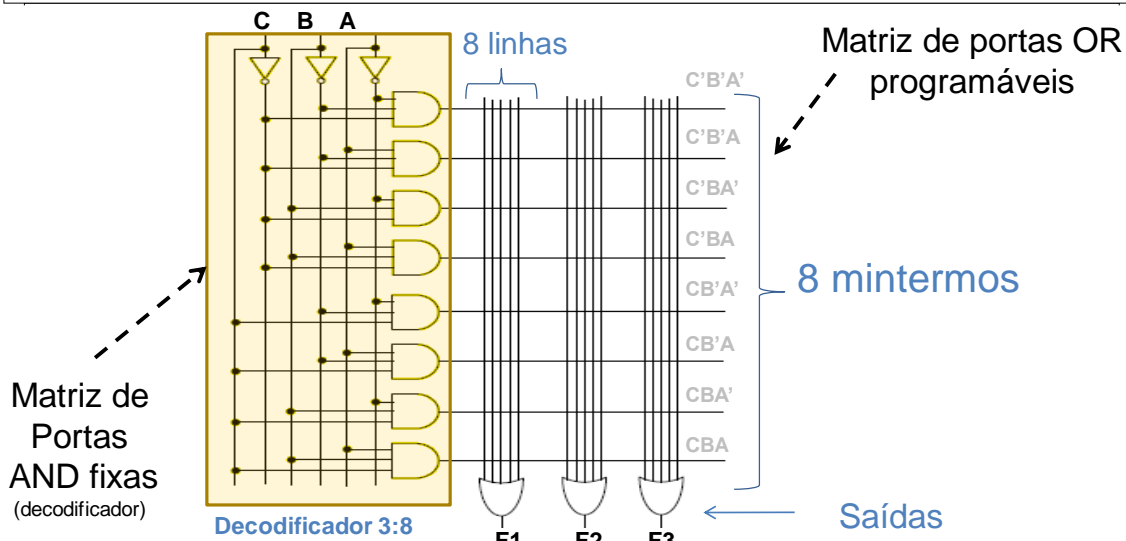


$$F1 = \overline{A}BC + \overline{A}B\overline{C} + ABC$$

$$F2 = \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C + ABC$$

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 10

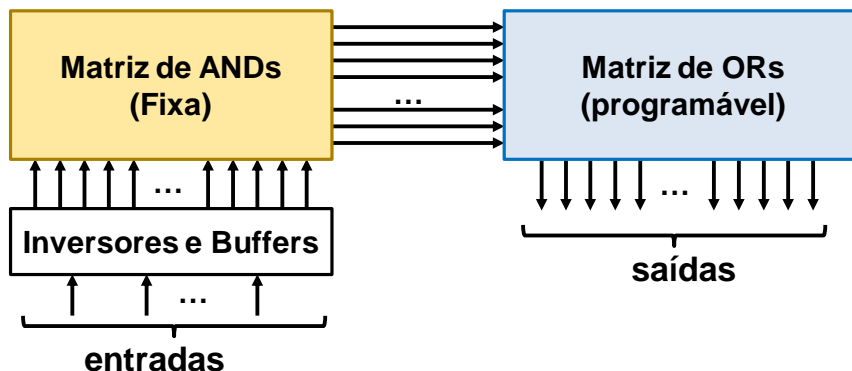
## Memórias como PLDs



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 11

## Memórias como PLDs

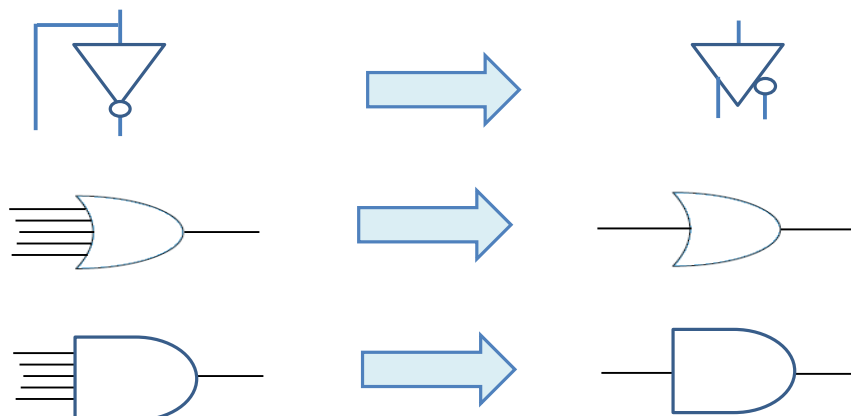
- Funcionalmente, \*PROMs são formadas por uma matriz de portas AND (DEC) fixas e uma matriz de portas OR programáveis.



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 12

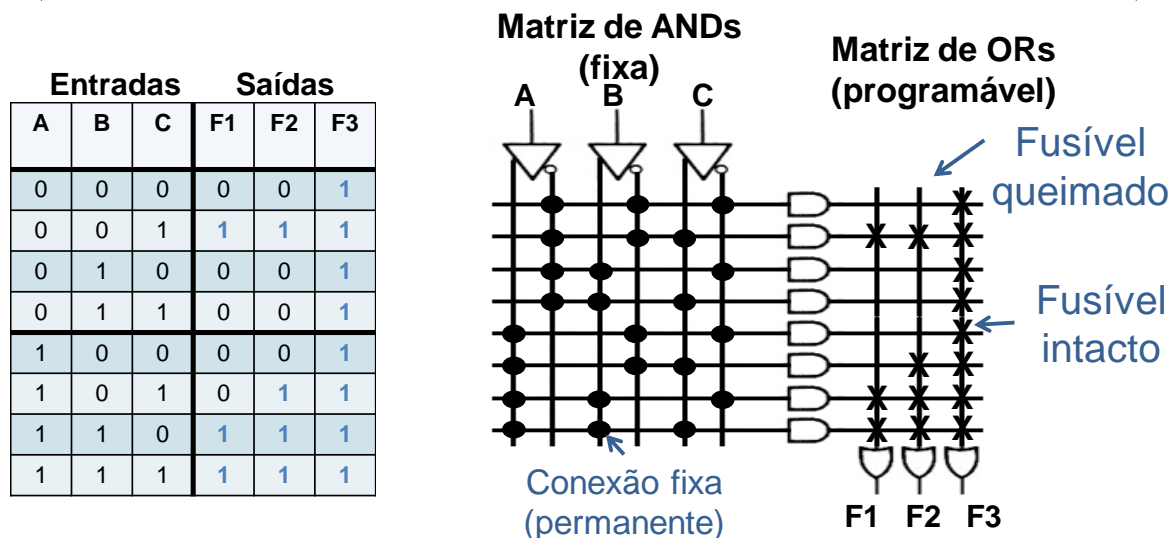
## Memórias como PLDs

Notação:



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 13

## Memórias como PLDs



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 14

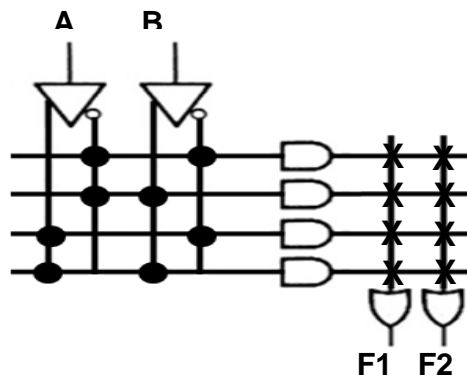
## Memórias como PLDs: Exercício

- Use uma \***PROM** para programar as funções F1 e F2 abaixo

Entradas		Saídas	
A	B	F1	F2
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$$F1 = \bar{A}B + AB = B$$

$$F2 = \bar{A}B + A\bar{B}$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 15

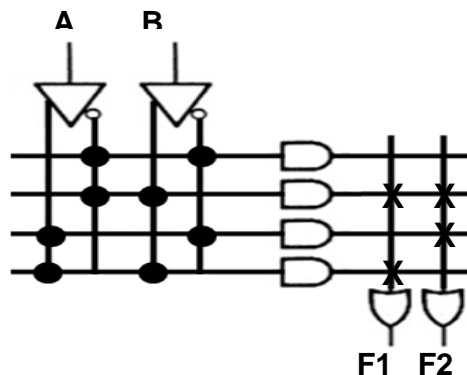
## Memórias como PLDs: Exercício

- Use uma \***PROM** para programar as funções F1 e F2 abaixo
- Resposta

Entradas		Saídas	
A	B	F1	F2
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$$F1 = \bar{A}B + AB = B$$

$$F2 = \bar{A}B + A\bar{B}$$



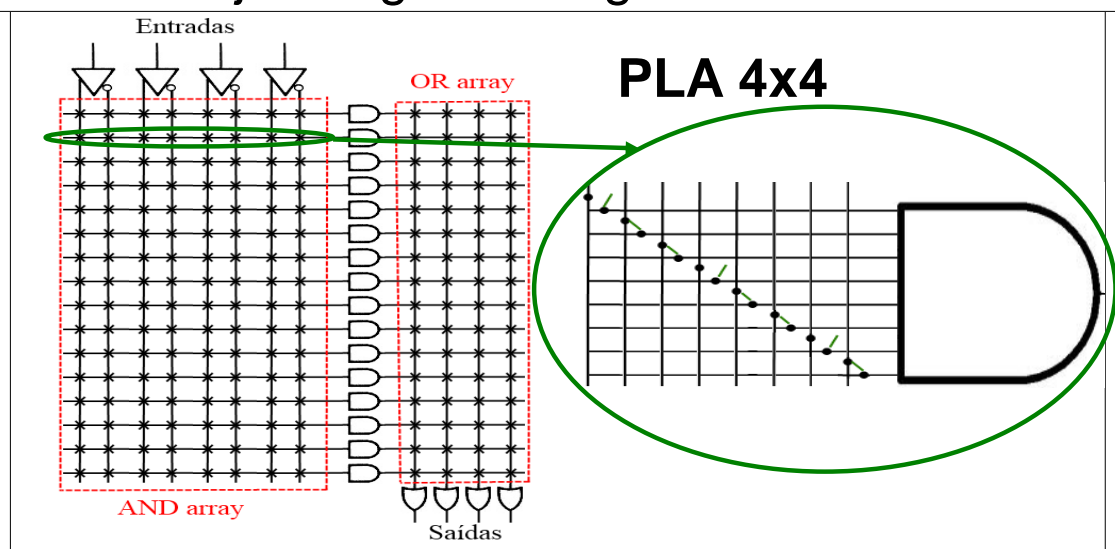
© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 16

## Arranjos Lógicos Programáveis: PLAs

- **\*PROMs** como PLDs: Implementam **tabela verdade completa**, sem minimização. Paralelamente ao uso de memórias, dispositivos de lógica programável de fato foram criados.
- As **PLAs** (*Programmable Logic Arrays*) são matrizes de lógica programável nas quais tanto a **matriz de ANDs** (primeiro nível) como a **matriz de ORs** (segundo nível) são **programáveis**.
- PLAs sintetizam tb funções na forma de **soma de produtos**: Produtos compartilhados entre diversas saídas.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 17

## Arranjos Lógicos Programáveis: PLAs



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 18

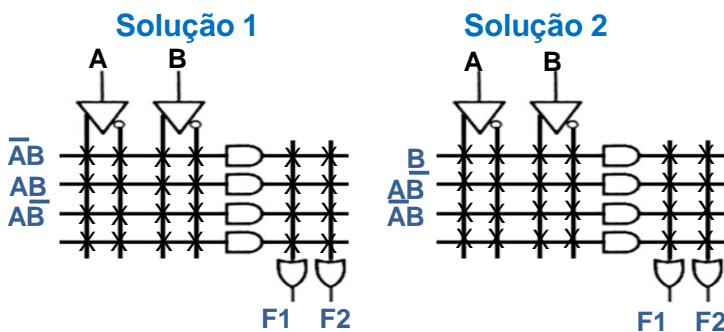
## Arranjos Lógicos Programáveis: Exercício 1

- Use uma **PLA** para programar as funções F1 e F2

Entradas		Saídas	
A	B	F1	F2
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$$F1 = \bar{A}B + \underline{AB} = B$$

$$F2 = \underline{AB} + \bar{A}B$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 19

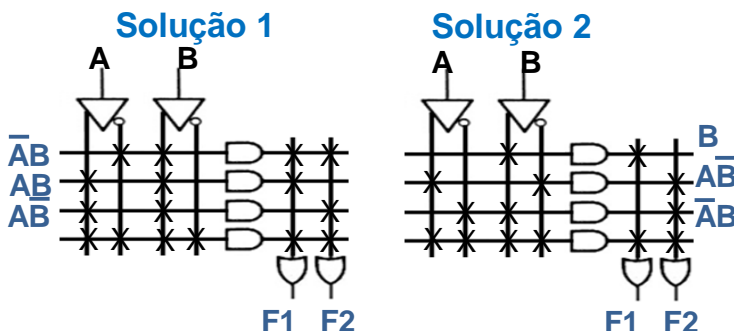
## Arranjos Lógicos Programáveis: Exercício 1

- Use uma **PLA** para programar as funções F1 e F2
- Resposta:**

Entradas		Saídas	
A	B	F1	F2
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$$F1 = \bar{A}B + \underline{AB} = B$$

$$F2 = \underline{AB} + \bar{A}B$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 20

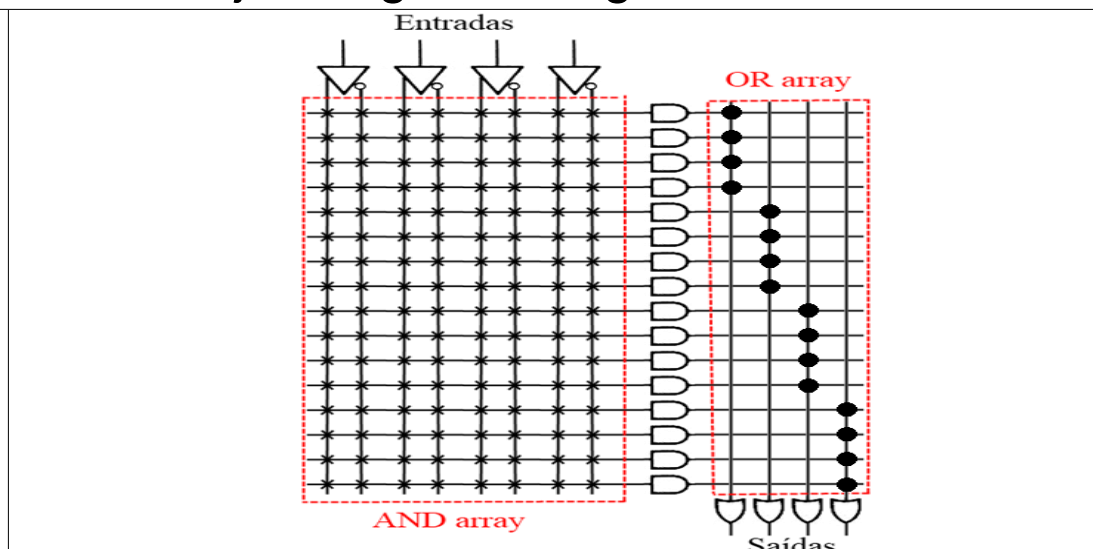


## Arranjos Lógicos Programáveis: PALs

- PLAs eram de difícil utilização e caras.
- Como alternativa foram propostas as **PALs**, onde o **nível de ANDs é programável**, mas o **nível de ORs é fixo**.
  - Menor flexibilidade, porém custo também menor.
- Nesse caso, um número fixo de entradas para cada porta OR é usado.
  - Deve ser usada uma PAL adequada ao nível de complexidade desejado (número de termos).
- Tanto PALs como PLAs são não voláteis, ou seja, podem ser desligadas sem perda de informação.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 21

## Arranjos Lógicos Programáveis: PALs



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 22

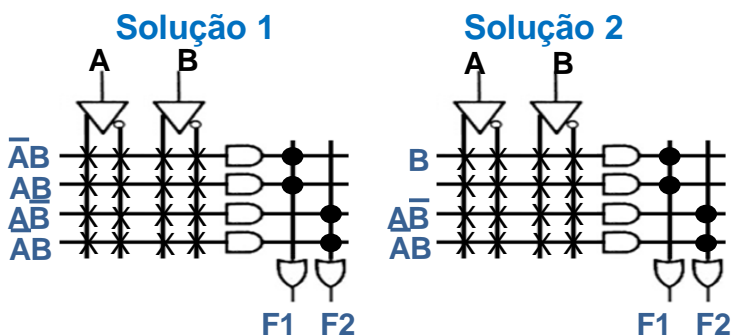
## Arranjos Lógicos Programáveis: Exercício 2

- Use uma **PAL** para programar as funções F1 e F2 abaixo

Entradas		Saídas	
A	B	F1	F2
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$$F1 = \bar{A}B + AB = B$$

$$F2 = AB + A\bar{B}$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 23

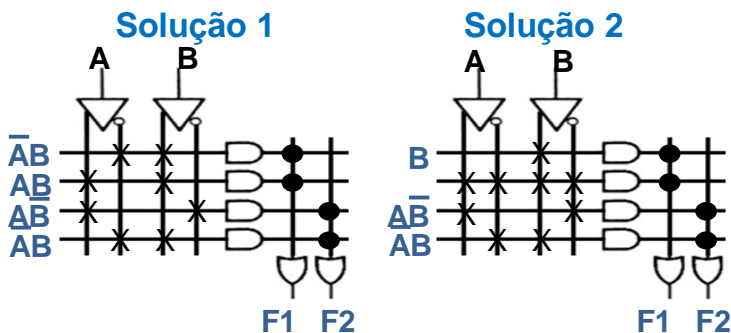
## Arranjos Lógicos Programáveis: Exercício 2

- Use uma **PAL** para programar as funções F1 e F2 abaixo
- Resposta:**

Entradas		Saídas	
A	B	F1	F2
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

$$F1 = \bar{A}B + AB = B$$

$$F2 = AB + A\bar{B}$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 24

## Arranjos Lógicos Programáveis: Exemplo

- Sintetizar as funções F1 e F2:

Entradas			Saídas	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 26

## Arranjos Lógicos Programáveis: Exemplo

- Passo 1: Determinar os produtos

Entradas			Saídas	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

$$F1 = A'B'C + ABC' + ABC$$

$$F2 = A'B'C + AB'C + ABC' + ABC$$

•\*PROM: todos os possíveis mintermos e escolher

•PLA ou PAL: minimiza a implementação !

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 27

## Arranjos Lógicos Programáveis: Exemplo

- Passo 1 (PLA): Determinar os produtos  
Com PLA: maximizar produtos compartilhados

Entradas			Saídas	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

F1		A	
B	C	0	1
0	0	0	0
0	1	1	0
1	1	0	1
1	0	0	1

F2		A	
B	C	0	1
0	0	0	0
0	1	1	1
1	1	0	1
1	0	0	1

$A'B'C$   
 $AB$   
 $AC$

$F1 = A'B'C + AB$   
 $F2 = A'B'C + AB + AC$

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 28

## Arranjos Lógicos Programáveis: Exemplo

- Passo 2 (PLA): Matriz de Personalização

$F1 = A'B'C + AB$   
 $F2 = A'B'C + AB + AC$

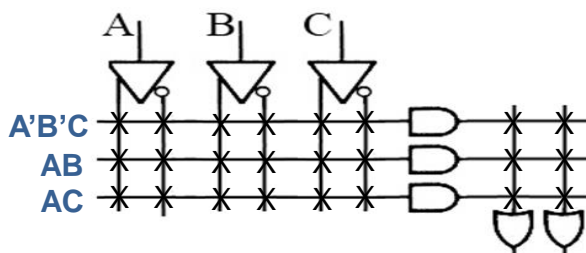
	Produto	Entrada			Saídas	
		A	B	C	F1	F2
Compartilhados	A'B'C	0	0	1	1	1
	AB	1	1	-	1	1
	AC	1	-	1	0	1

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 29

## Arranjos Lógicos Programáveis: Exemplo

- Passo 3 (PLA): Desenhar o circuito

Produto	Entrada			Saídas	
	A	B	C	F1	F2
A'B'C	0	0	1	1	1
AB	1	1	-	1	1
AC	1	-	1	0	1

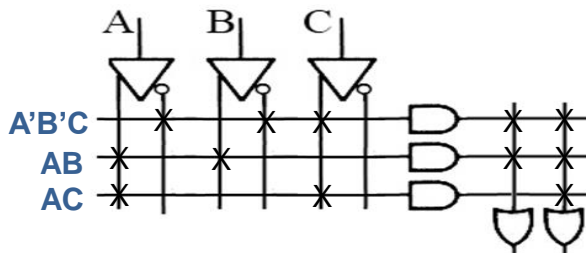


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 30

## Arranjos Lógicos Programáveis: Exemplo

- Passo 3 (PLA): Desenhar o circuito

Produto	Entrada			Saídas	
	A	B	C	F1	F2
A'B'C	0	0	1	1	1
AB	1	1	-	1	1
AC	1	-	1	0	1



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 31

## Arranjos Lógicos Programáveis: Exemplo

- Passo 1 (PAL): Determinar os produtos

Com PAL: minimizar número de produtos

Entradas			Saídas	
A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

F1		A			F2		A	
B	C	0	1		B	C	0	1
0	0	0	0		0	0	0	0
0	1	1	0	$A'B'C$	0	1	1	1
1	1	0	1	$AB$	1	1	0	1
1	0	0	1	$B'C$	1	0	0	1

$$F1 = A'B'C + AB$$

$$F2 = B'C + AB$$

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 32

## Arranjos Lógicos Programáveis: Exemplo

- Passo 2 (PAL): Matriz de Personalização

Obs.: não é essencial para a PAL, mas ajuda.

$$F1 = A'B'C + AB$$

$$F2 = B'C + AB$$

Produto	Entrada			Saídas	
	A	B	C	F1	F2
A'B'C	0	0	1	1	0
AB	1	1	-	1	0
B'C	-	0	1	0	1
AB	1	1	-	0	1

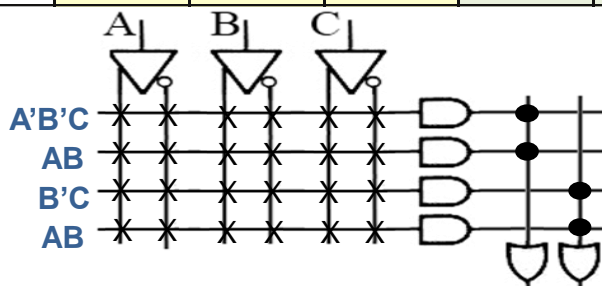
Repe-  
tidos

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 33

## Arranjos Lógicos Programáveis: Exemplo

- Passo 3 (PAL): Desenhar o circuito

Produto	Entrada			Saídas	
	A	B	C	F1	F2
A'B'C	0	0	1	1	0
AB	1	1	-	1	0
B'C	-	0	1	0	1
AB	1	1	-	0	1

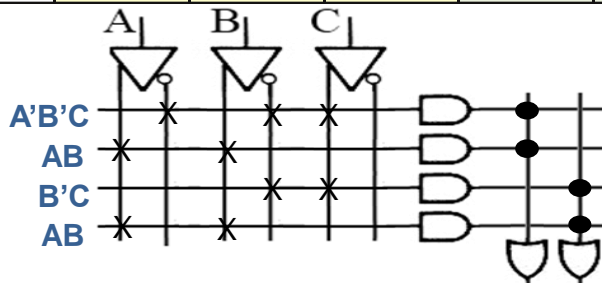


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 34

## Arranjos Lógicos Programáveis: Exemplo

- Passo 3 (PAL): Desenhar o circuito

Produto	Entrada			Saídas	
	A	B	C	F1	F2
A'B'C	0	0	1	1	0
AB	1	1	-	1	0
B'C	-	0	1	0	1
AB	1	1	-	0	1



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 35

## Arranjos Lógicos Programáveis: Exemplo

- Sintetizar:
  - $F0 = A + B'C'$
  - $F1 = AC' + AB$
  - $F2 = B'C' + AB$
  - $F3 = B'C + A$

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 36

## Arranjos Lógicos Programáveis: Exemplo

- Matriz das funções

Produto	Entrada			Saídas		
	A	B	C	F0	F1	F2
AB	1	1	-	0	1	1
B'C	-	0	1	0	0	0
AC'	1	-	0	0	1	0
B'C'	-	0	0	1	0	1
A	1	-	-	1	0	0

F3 ? 

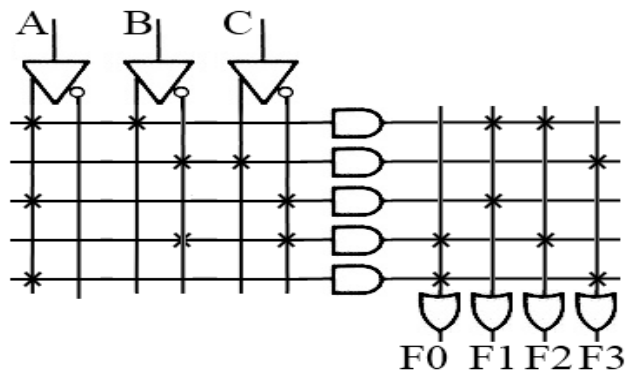
© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 37



## Arranjos Lógicos Programáveis: Exemplo

- Implementação com PLA

$$\begin{aligned} F0 &= A + B'C' \\ F1 &= AC' + AB \\ F2 &= B'C' + AB \\ F3 &= B'C + A \end{aligned}$$

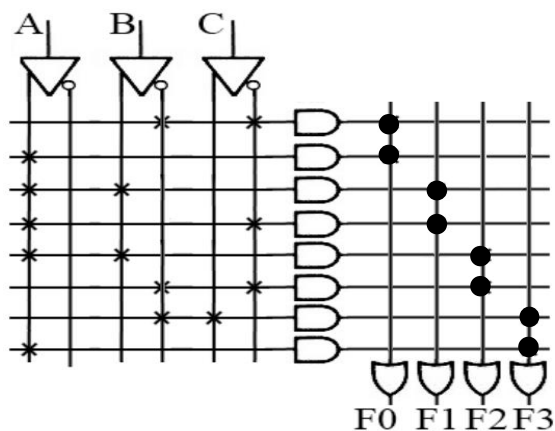


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 38

## Arranjos Lógicos Programáveis: Exemplo

- Implementação com PAL

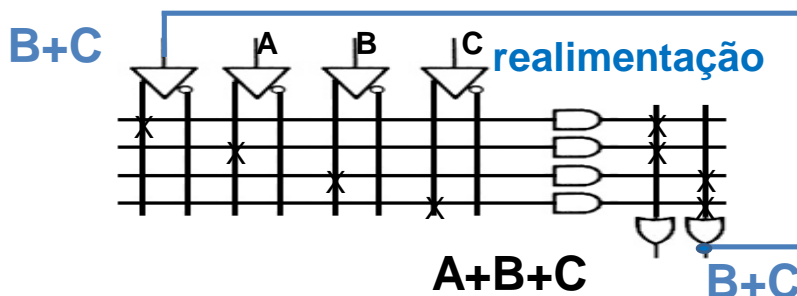
$$\begin{aligned} F0 &= A + B'C' \\ F1 &= AC' + AB \\ F2 &= B'C' + AB \\ F3 &= B'C + A \end{aligned}$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 39

## Arranjos Lógicos Programáveis: Extras

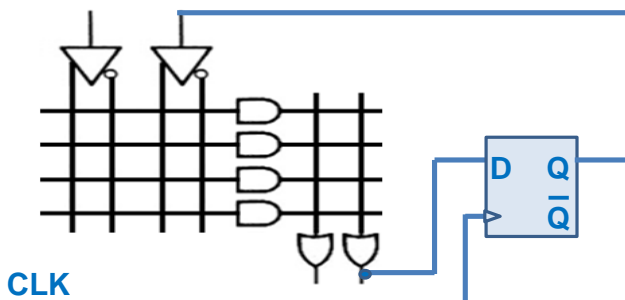
- Algumas PLAs e PALs comerciais apresentam funcionalidades adicionais, como:
  - Realimentação – Saída utilizada como entrada.



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 40

## Arranjos Lógicos Programáveis: Extras

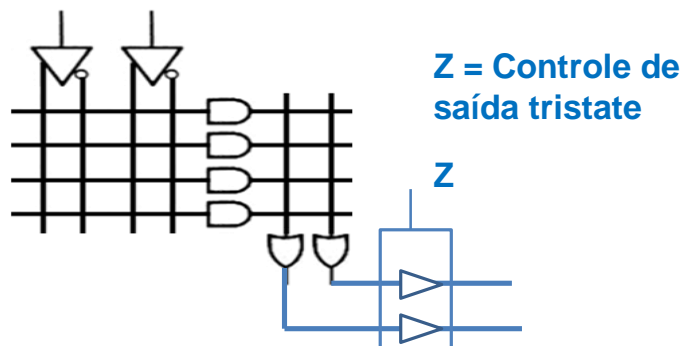
- Algumas PLAs e PALs comerciais apresentam funcionalidades adicionais, como:
  - Registradores na saída – Permitem sintetizar circuitos sequenciais.



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 41

## Arranjos Lógicos Programáveis: Extras

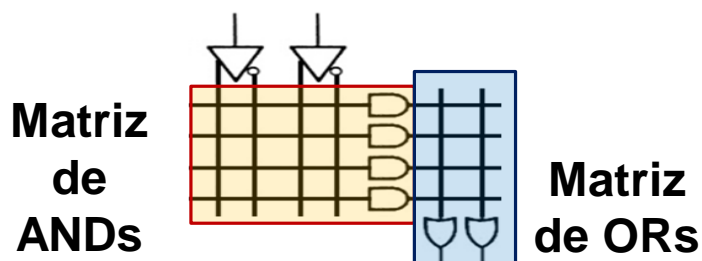
- Algumas PLAs e PALs comerciais apresentam funcionalidades adicionais, como:
  - Saídas *tri-state* – Permitem colocar a saída em estado de alta impedância.



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 42

## Arranjos Lógicos Programáveis: Resumo

	Matriz AND	Matriz OR
<b>*PROM</b>	fixa	programável
<b>PLA</b>	programável	programável
<b>PAL</b>	programável	fixa



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 43

## PLDs atuais: CPLDs e FPGAs

11/06/18

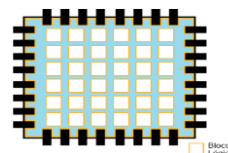
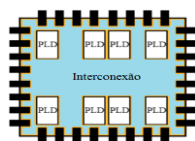
- Com o avanço da microeletrônica, circuitos programáveis mais complexos foram sendo elaborados. Circuitos atuais se enquadram em duas categorias:
  - **CPLD – Complex PLD;**
  - **FPGA – Field Programmable Gate Array.**



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 44

## PLDs modernos: Características

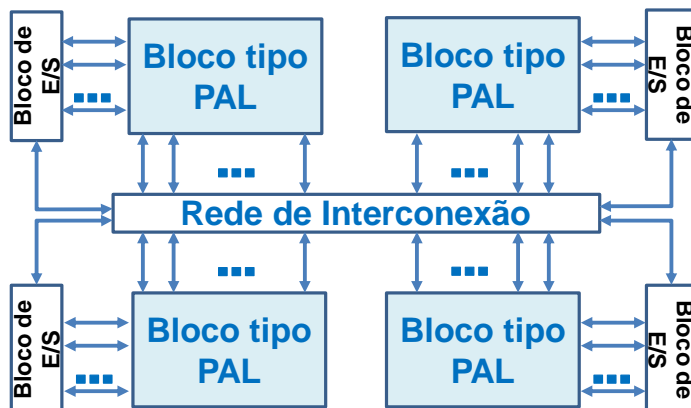
- Grande quantidade de lógica programável – Flip-flops pré-implementados.
- Interconexões programáveis entre lógica programável, Flip-flops e entradas/saídas do dispositivo – É possível controlar o roteamento dos sinais dentro do circuito.



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 45

## PLDs modernos: Estrutura

- **CPLD** – Estrutura contínua de conexão:
  - Em geral de 2 a 100 blocos tipo PAL.

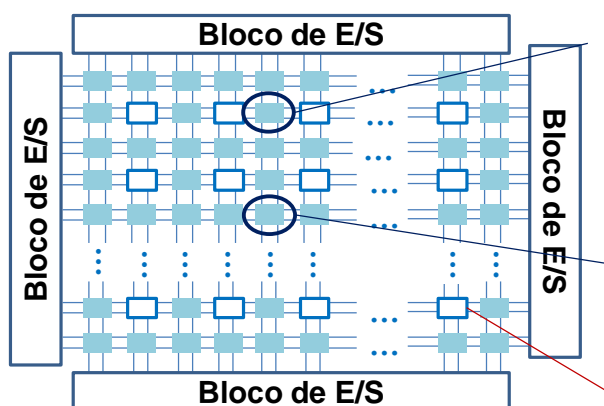


PAL muito grandes  
teriam muito mais  
problemas de  
desperdício de área  
e de desempenho

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 46

## PLDs modernos: Estrutura

- **FPGA**: estrutura segmentada de conexão



Chaves programáveis  
(ligações do bloco com  
linhas de interconexão)

Chaves programáveis  
(ligações entre linhas  
verticais e horizontais)

**Bloco lógico**  
bem menor que as plds !

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 47

## PLDs modernos: Comparação

- Complexidade:
  - PLAs e PALs são muito simples
    - Menos que 200 *gates* equivalentes;
  - CPLDs têm capacidade moderada
    - Até 100.000 *gates* equivalentes;
  - FPGAs atuais
    - Acima de 1M de *gates* equivalentes.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 48

## PLDs modernos: Comparação

- Volatilidade:
  - CPLDs são dispositivos não voláteis, i.e., podem ser desligados sem perda de informação (EEPROM);
  - FPGAs são voláteis, i.e., quando desligadas perdem a informação (*field programable!*);
    - Algumas FPGAs são associadas a ROMs para permitir que a informação seja recuperada ao ligar.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 49

## PLDs modernos: Programação

- Linguagens de programação próprias existentes – PALASM, CUPL, etc.
- Não são mais usadas devido às linguagens de descrição de Hardware de mais alto nível – VHDL, Verilog.
- As linguagens de alto nível são transcritas para fluxo de bits - usado para programar o dispositivo.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 50

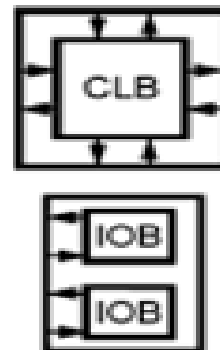
## FPGAs

- Atualmente FPGAs são os dispositivos mais utilizados.
- FPGAs modernas podem emular algumas centenas de milhares de portas lógicas.
- Algumas FPGAs possuem memória RAM e/ou processadores embutidos para facilitar o projeto.
- Existem versões em Flash e EEPROM – caras!

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 51

## FPGAs: Componentes

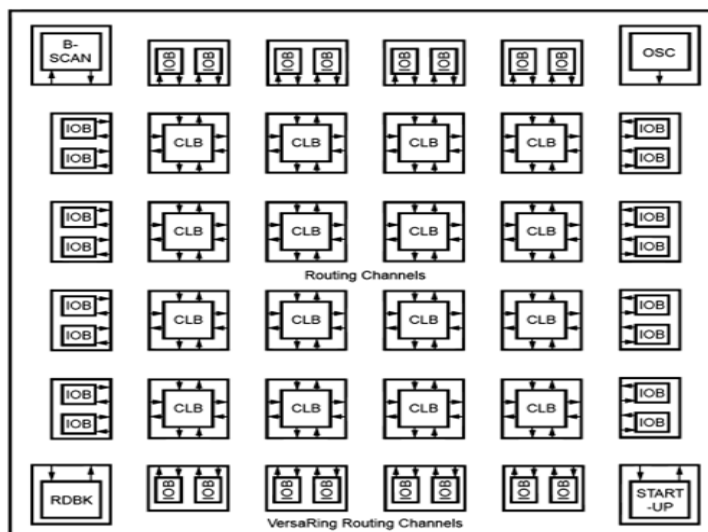
- Compostas de:
  - CLBs (*Configurable Logic Blocks*)
    - Blocos lógicos que executam funções
  - IOBs (*Input/Output Buffers*)
    - Interface com mundo externo;
  - Interconexões programáveis
    - Conecta CLBs e IOBs.



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 52

## FPGAs: Componentes

- Diagrama de blocos do Xilinx Spartan



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 53

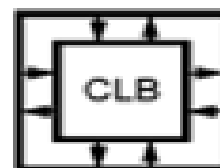


## FPGAs: CLBs

Configurable Logic Blocks

- CLBs são compostos de:
  - LUTs (*lookup tables*)
    - Implementam lógica combinatória
  - Flip-flops
    - Implementam funções sequenciais
  - Multiplexadores
    - Conectam LUTs e Flip-flops.

equivalente a ~25 gates  
e 2 FF

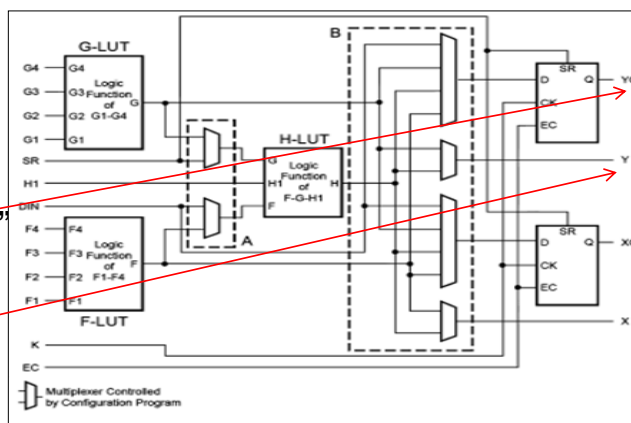


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 54

## FPGAs: CLBs

Configurable Logic Blocks

- 3 LUTs:
  - LUT-F (4 x 1-bit)
  - LUT-G (4 x 1-bit)
  - LUT-H (3 x 1-bit)
- 2 saídas “registradas”
  - XQ e YQ
- 2 saídas combinatórias
  - X e Y



Xilinx Spartan

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 55

## FPGAs: LUTs

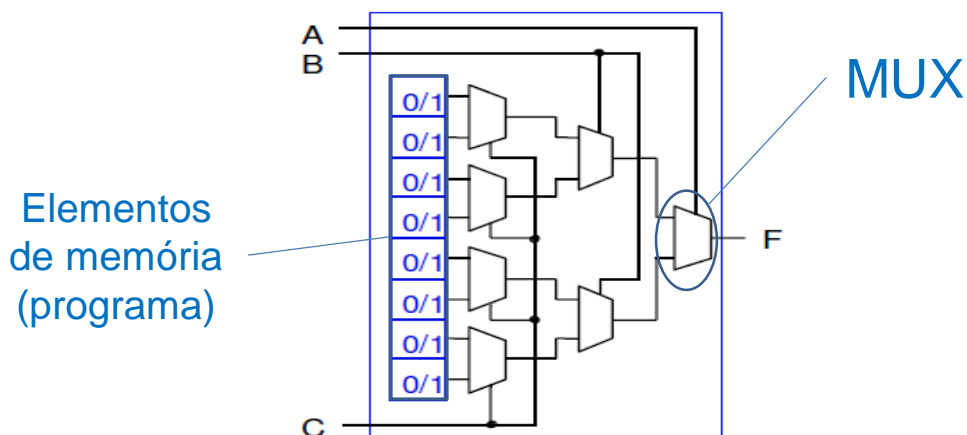
*lookup table*

- LUT-N – Uma **tabela-verdade** de N entradas e  $2^N$  posições de 1 bit:
  - Programar uma LUT significa preencher os valores da tabela-verdade,
  - Uma LUT com 4 entradas pode emular  $2^{16}$  (~64 K) funções booleanas (todas as possibilidades com 16 possíveis produtos =  $2^{16}$ );
  - FPGAs comerciais podem conter LUTs de até 6 entradas (mais de 16 bilhões de funções emuláveis).
- Implementadas usando MUXes
  - Mais compactos do que arranjos AND/OR
  - Necessitam de elementos de memória, normalmente do tipo SRAM.
  - Fornecem maior velocidade ao circuito
  - Dados alimentados na SRAM a partir de memória não volátil (ex.: EEPROM) que contém o programa

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 56

## FPGAs: LUTs

- Exemplo 1:

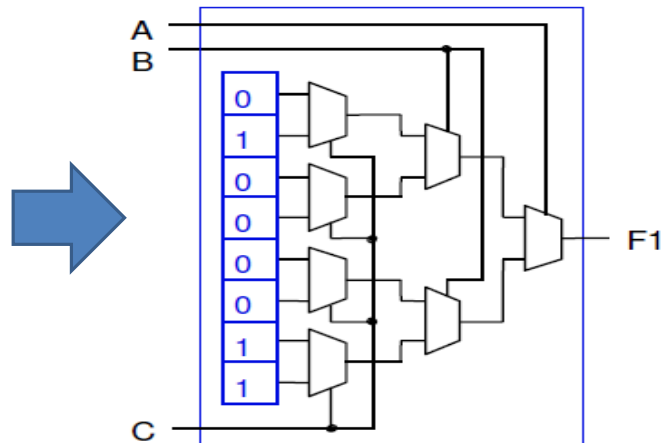


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 57

## FPGAs: LUTs

- Exemplo 1: Programação

A	B	C	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

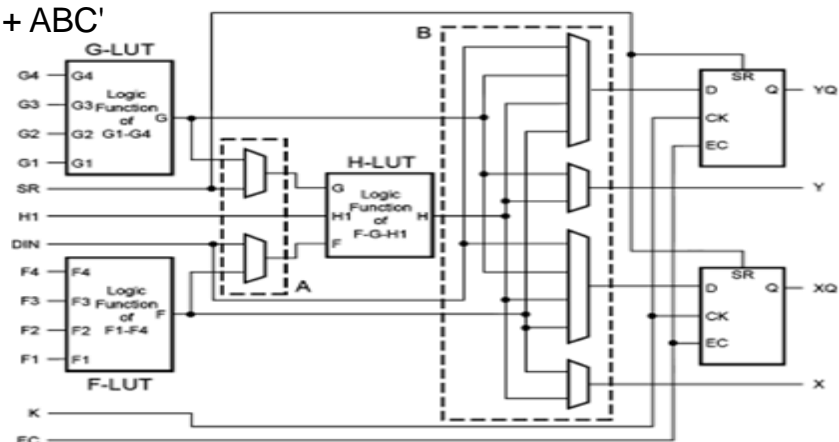


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 58

## FPGAs: LUTs

- Exemplo 2: programação das funções X e Y na Xilinx Spartan:

- $X = A'B'C + ABC'$
- $Y = AB'$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 59

## FPGAs: LUTs

- Exemplo 2: programação das funções X e Y na Xilinx Spartan:

$$- X = A'B'C + ABC'$$

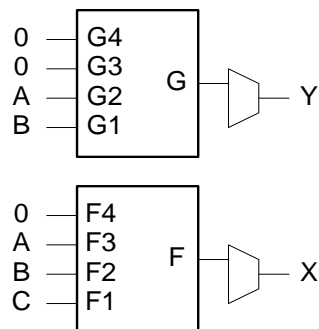
$$- Y = AB'$$

F4	(A) F3	(B) F2	(C) F1	(X) F
X	0	0	0	0
X	0	0	1	1
X	0	1	0	0
X	0	1	1	0
X	1	0	0	0
X	1	0	1	0
X	1	1	0	1
x	1	1	1	0

G4	G3	(A) G2	(B) G1	(Y) G
X	X	0	0	0
X	X	0	1	0
X	X	1	0	1
X	X	1	1	0

Programa na LUT-G

Programa na LUT-F

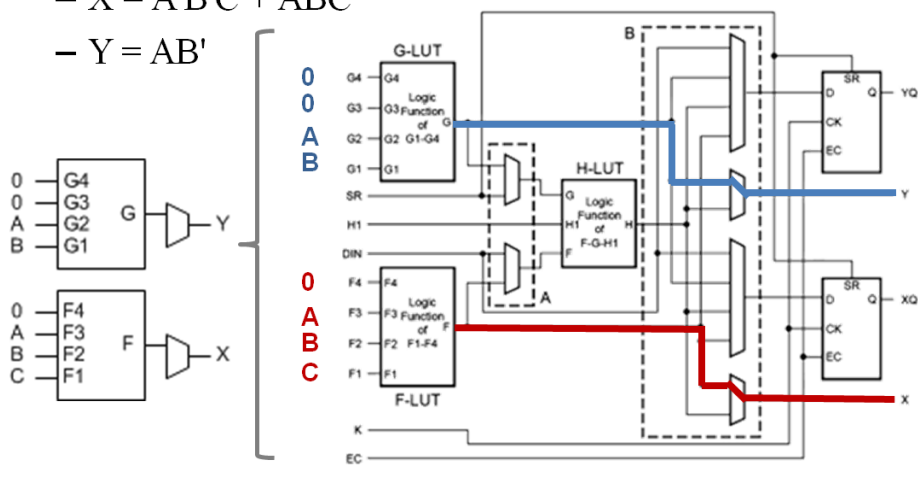


© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 60

## FPGAs: LUTs

$$- X = A'B'C + ABC'$$

$$- Y = AB'$$



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 61

## FPGAs: LUTs

- Usando LUT-4: é possível calcular funções com maior número de variáveis associando 2 ou mais LUTs.
- Por exemplo, vamos calcular a função:
  - $X = ABCDEFGH$
- Para isso precisamos inicialmente dividir a função em subfunções:
  - $X_1 = ABCD$
  - $X_2 = EFGH$
  - $X = X_1X_2$

**3 LUTs**

LUT-4 = 4 entradas, 16 posições

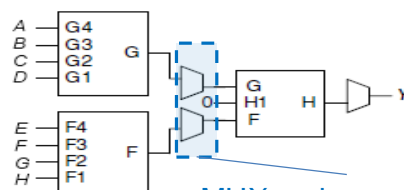
© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 62

## FPGAs: LUTs

- Vamos configurar:
  - LUT-F para calcular  $X_1 = ABCD$
  - LUT-G para calcular  $X_2 = EFGH$
  - LUT-H para calcular  $X = X_1X_2$

F4	F3	F2	F1	F	G4	G3	G2	G1	G
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	0	0	0	1	1	0
0	1	0	0	0	0	1	0	0	0
0	1	0	1	0	0	1	0	1	0
0	1	1	0	0	0	1	1	0	0
0	1	1	1	0	0	1	1	1	0
1	0	0	0	0	1	0	0	0	0
1	0	0	1	0	1	0	0	1	0
1	0	1	0	0	1	0	1	0	0
1	0	1	1	0	1	0	1	1	0
1	1	0	0	0	1	1	0	0	0
1	1	0	1	0	1	1	0	1	0
1	1	1	0	0	1	1	1	0	0
1	1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1

H1	F	G	H
x	0	0	0
x	0	1	0
x	1	0	0
x	1	1	1

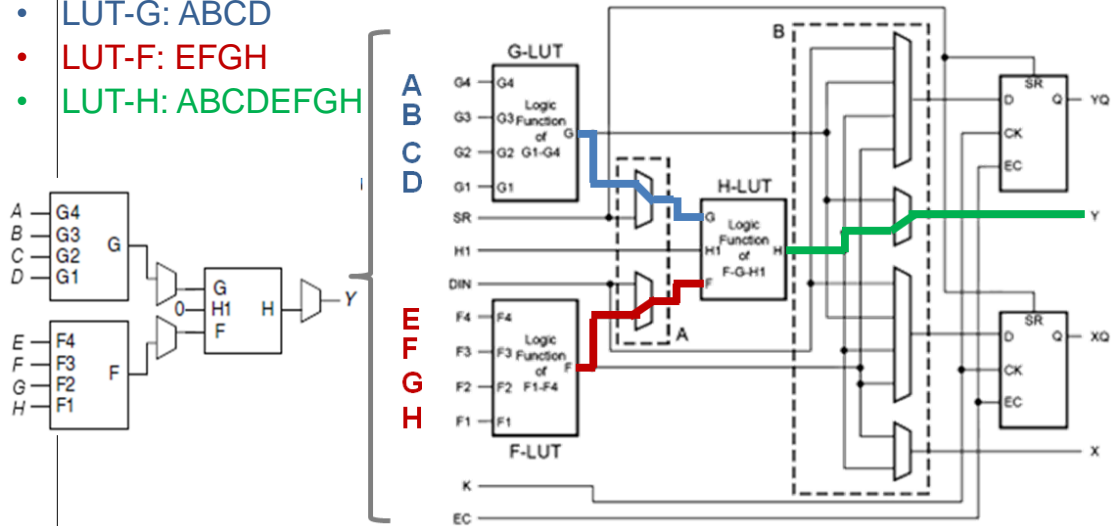


MUXes de passagem devem ser configurados.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 63

## FPGAs: LUTs

- LUT-G: ABCD
- LUT-F: EFGH
- LUT-H: ABCDEFGH



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 64

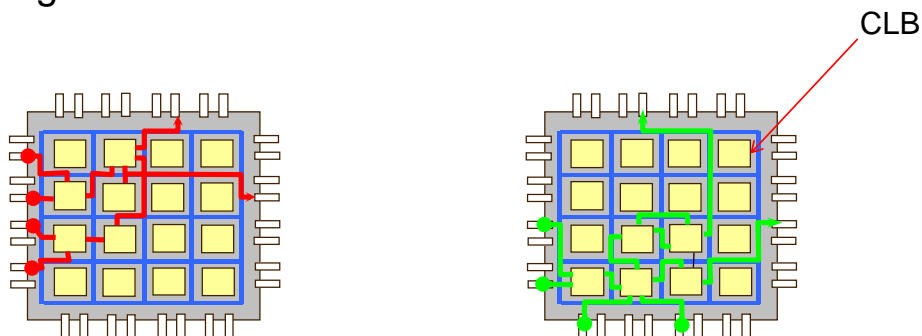
## FPGAs: LUTs

- Usando associações, cada CLB Spartan pode calcular funções de até 9 variáveis.
- É possível calcular funções de mais variáveis associando mais de um CLB, por meio das saídas combinatórias.
- Ao se associar diversos CLBs, existe uma redução na frequência máxima, pois o sinal deverá atravessar um número maior de LUTs até chegar a um Flip-Flop.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 65

## FPGAs: Interconexões Programáveis

- Permitem conexão entre os CLBs
- Configuráveis de acordo com necessidade



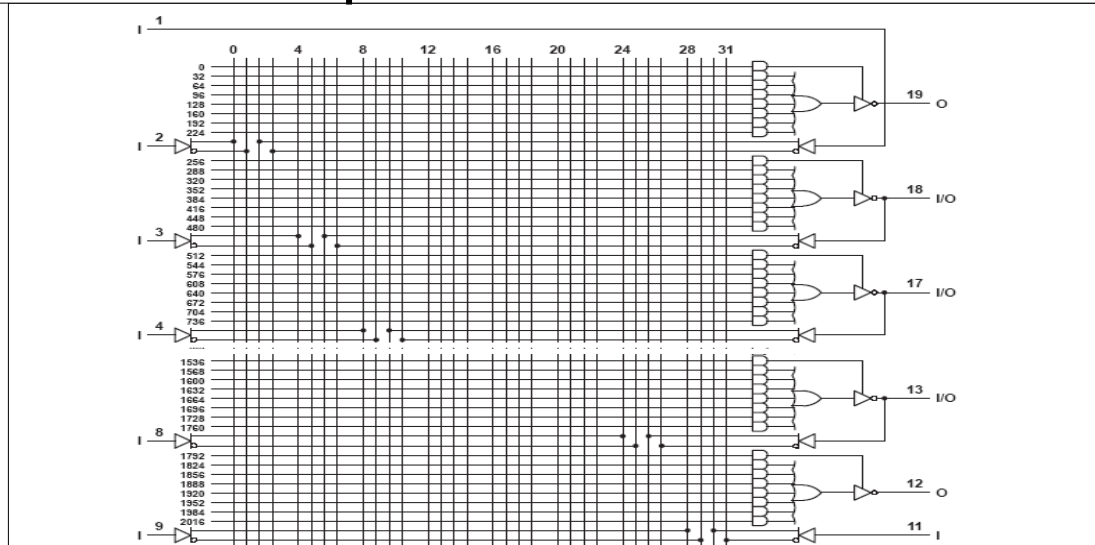
© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 66

## Misc: Fabricantes

- Actel(<http://www.actel.com/>)
- Altera (<http://www.altera.com/>)
- Atmel(<http://www.atmel.com/products/>)
- Cypress(<http://www.cypress.com/>)
- Lattice(<http://www.lattice.com/>)
- Quicklogic(<http://www.quicklogic.com/>)
- Xilinx(<http://www.xilinx.com/>)

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 84

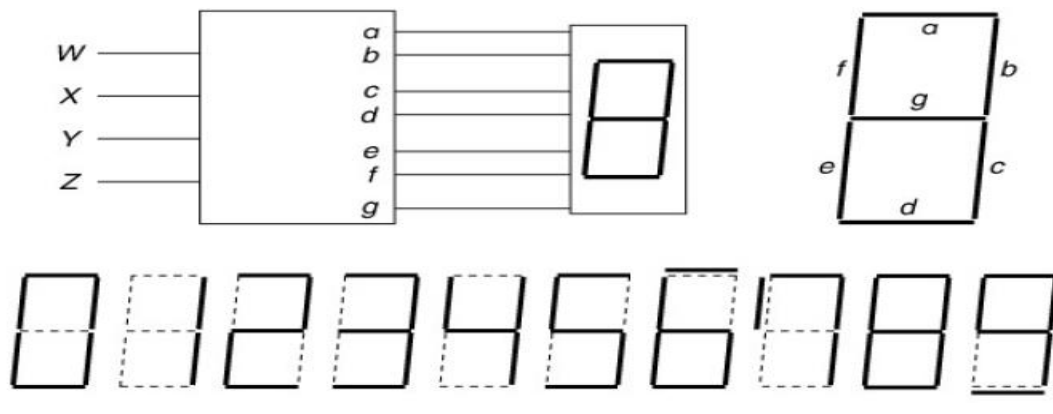
## Componente: PAL 16L8



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 85

## Misc: Exemplo/Exercício

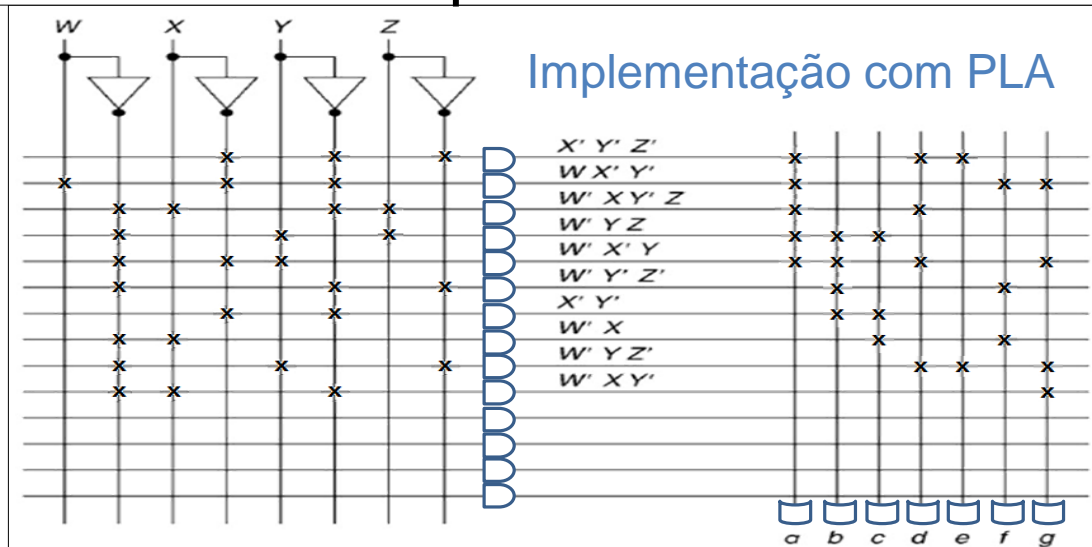
- Implementação de um display de 7 segmentos usando PAL, PLA e FPGA



© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 86



## Exemplo/Exercício



## Livro Texto

- Wakerly, J.F.; *Digital Design – Principles & Practices*; Fourth Edition, ISBN: 0-13-186389-4, Pearson & Prentice-Hall, Upper Saddle, River, New Jersey, 07458, 2006.

## Lição de Casa

- **Leitura Obrigatória:**
  - Capítulo 9 do Livro Texto, itens 9.5 e 9.6.
- **Exercícios Obrigatórios:**
  - Exercícios 9.28 a 9.36 do Capítulo 9 do Livro Texto.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 89

## Bibliografia Adicional Deste Assunto

- “Digital Design and Computer Architecture”. Harris, D.M. & Harris, S.L. Morgan Kaufmann, 2007.
- “Digital Systems Design Using VHDL”. Charles H. Roth Jr and Lizy Kurian John. 2nd Edition, 2008. ISBN-10: 0534384625 ISBN-13: 9780534384623
- “The Ten Commandments of Excellent Design—VHDL Code Examples”. Peter Chambers. VLSI Technology.
- Chapter 3 - The Art of VHDL Synthesis. LeonardoSpectrum HDL Synthesis Manual.

© Andrade, Midorikawa, Saraiva, Símplicio e Spina 2.012 <Circ.Seq.: Bloc. Bás.I> PCS 2304 Sistemas Digitais II 90