



Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Curso de Graduação em Ciências da Computação



Lógica Programável

INE 5348

Aula 7

Máquinas Seqüências Síncronas: Síntese de circuitos seqüenciais. Exemplos. Mapeamento e alternativas de implementação de máquinas de estado: "hardwired", PLA, ROM e PLD.

Prof. José Luís Güntzel
guntzel@inf.ufsc.br

www.inf.ufsc.br/~guntzel/ine5348/ine5348.html

Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Registrador de Estados:

- **Tipos de registradores:**

- Feito com FFDs ou com FFJKs ou com FFTs
- Pode ser um registrador-deslocador
- Pode ser um registrador-contador

- **Quanto à forma de fabricação:**

- Registradores podem estar prontos, integrados em chips com 4 ou 8 bits, cascadeáveis (componentes MSI CMOS ou TTL)
- Registradores podem fazer uso de flip-flops pre-existentes dentro de um componente programável tipo SPLD, CPLD ou FPGA.
- Registradores podem ser especificados para serem fabricados do zero (opção de fabricação com máscaras ou *masked*)

Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Lógica de Próximo Estado e Lógica de Saída:

- Tipos de implementações:

hardwired

- Implementando as equações por meio de um circuito combinacional (“lógica aleatória”)
- Implementando as equações pela configuração de planos “E” e “OU” (PALs e PLAs)

Micropro-
gramação

- Gravando a tabela-verdade em bloco de memória (ROM, EPROM, EEPROM ou RAM)

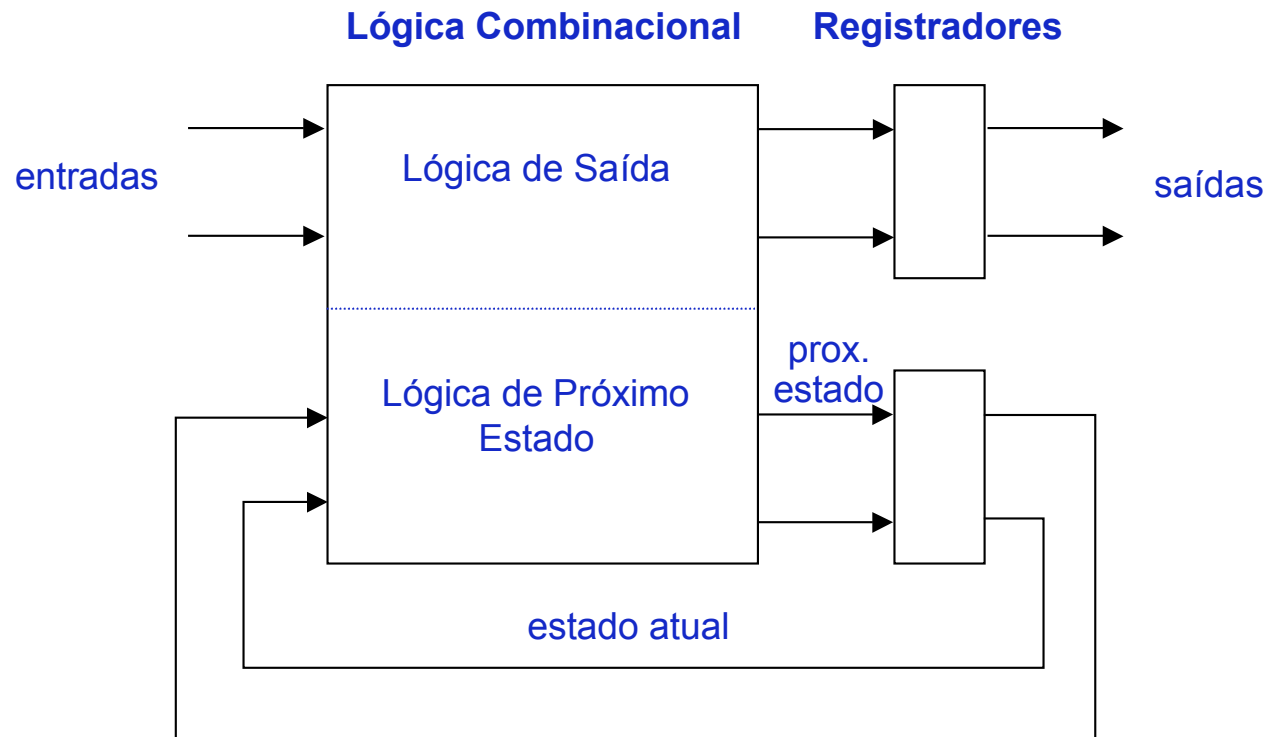
- Quanto à forma de fabricação:

- Usando chips de memória ROM, EPROM ou EEPROM
- Usando chips programáveis SPLDs: PLAs ou PALs
- Usando chips programáveis CPLDs ou FPGAs
- Mandando fabricar um chip do zero (*masked*)

Máquinas Seqüenciais Síncronas

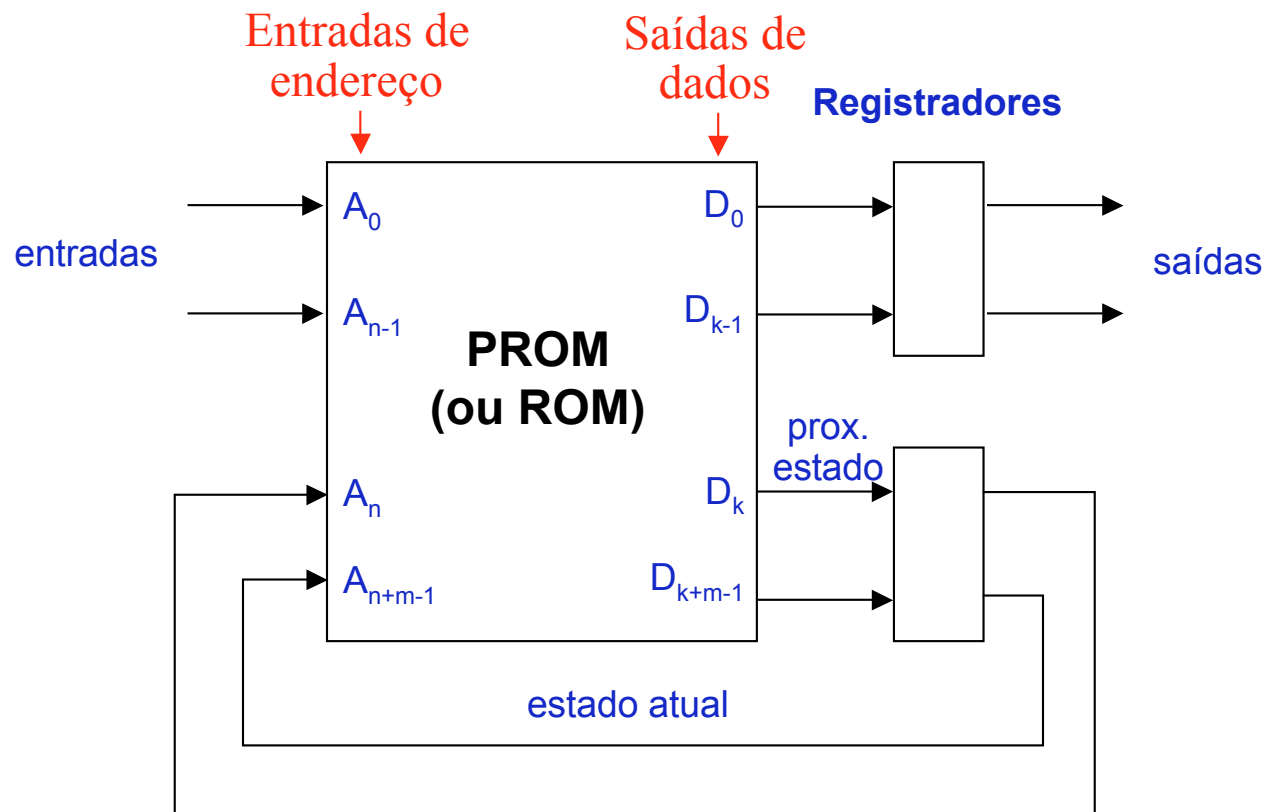
► Alternativas de Implementação de FSMs

Lógica de Próximo Estado e Lógica de Saída



Máquinas Seqüenciais Síncronas

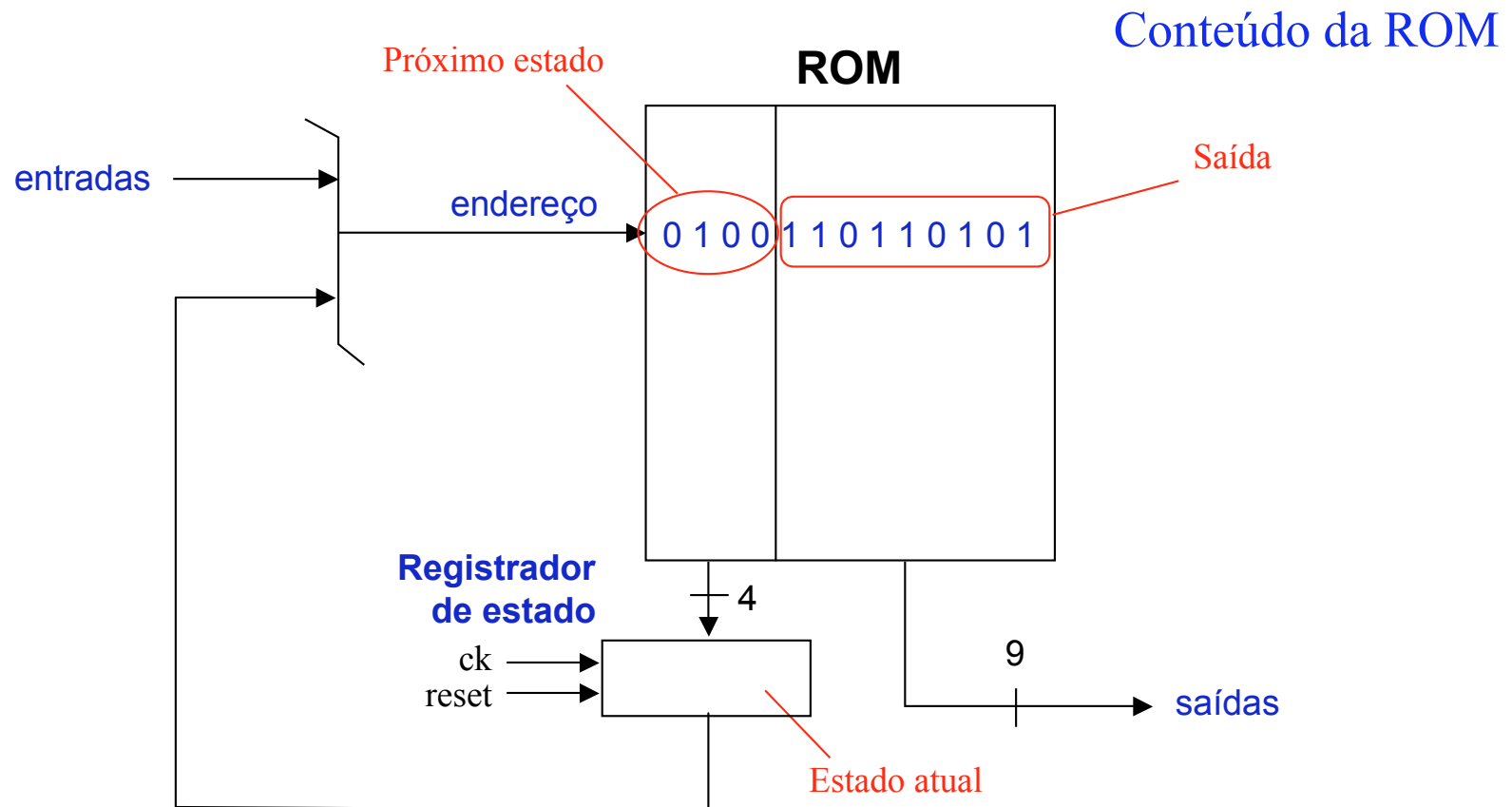
► Alternativas de Implementação de FSMs Lógica de Próximo Estado e Lógica de Saída como Bloco ROM



Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Lógica de Próximo Estado e Lógica de Saída como Bloco ROM

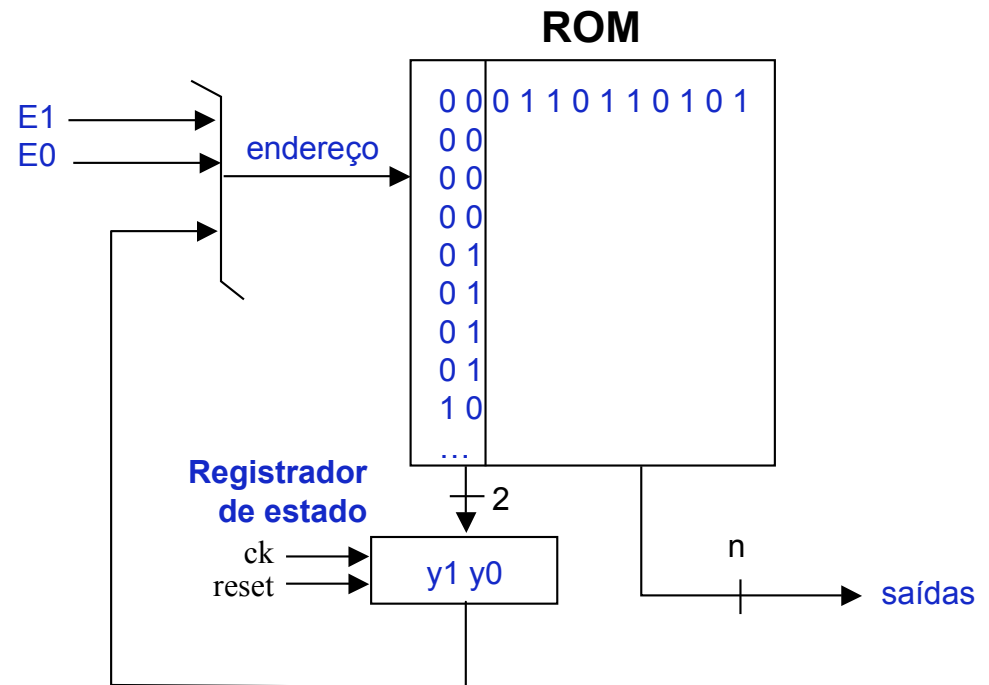


Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Para Cada Linha da Tabela de Estados uma Linha da ROM

Estado atual y1y0	E1	E0	Próximo estado Y1Y0	saídas
00	0	0	00	
00	0	1	00	
00	1	0	01	
00	1	1	01	
01	0	0	01	
01	0	1	10	
01	1	0	01	
01	1	1	10	
10	0	0	11	
10	0	1	11	
10	1	0	11	
10	1	1	11	
11	0	0	00	
11	0	1	00	
11	1	0	00	
11	1	1	00	



- O número de linhas da ROM é limitado
- Pode haver diversas linhas da tabela de transição que resultem nos mesmo valores para as saídas

Máquinas Seqüenciais Síncronas

- **Alternativas de Implementação de FSMs**
A Solução é Agrupar Casos Equivalentes na Tabela de Transição

Estado atual y1y0	E1	E0	Próximo estado Y1Y0
00	0	0	00
00	0	1	00
00	1	0	01
00	1	1	01
01	0	0	01
01	0	1	10
01	1	0	01
01	1	1	10
10	X	X	11
11	X	X	00

Necessita de **10** linhas da ROM



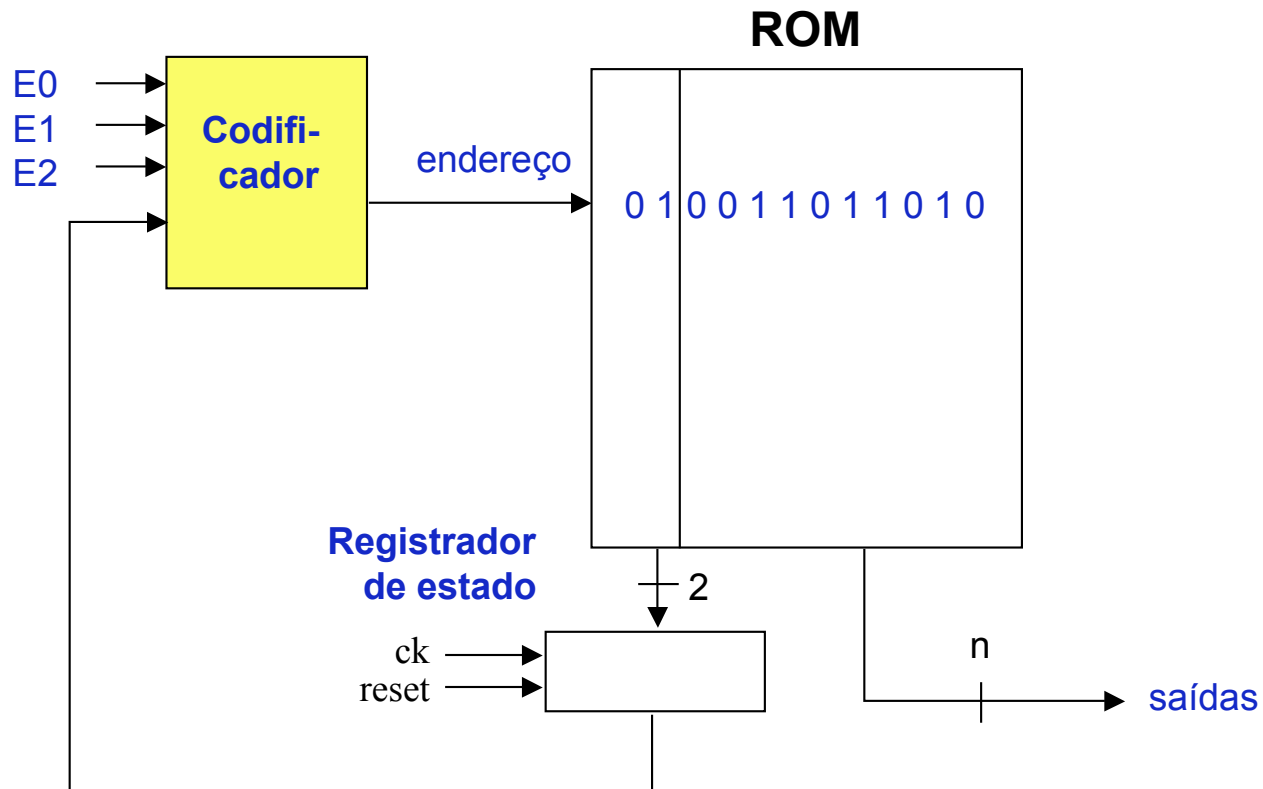
Agrupando-se
os *don't cares*

Estado atual y1y0	E1	E0	Próximo estado Y1Y0
00	0	X	00
00	1	X	01
01	X	0	01
01	X	1	10
10	X	X	11
11	X	X	00

Necessita de **6** linhas da ROM

Máquinas Seqüenciais Síncronas

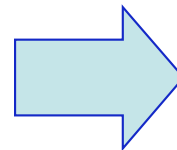
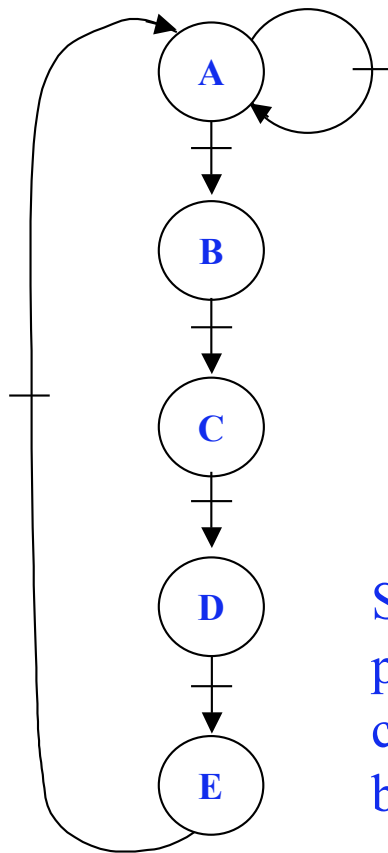
- ▶ **Alternativas de Implementação de FSMs**
O Agrupamento Pode Ser Implementado Por um Codificador



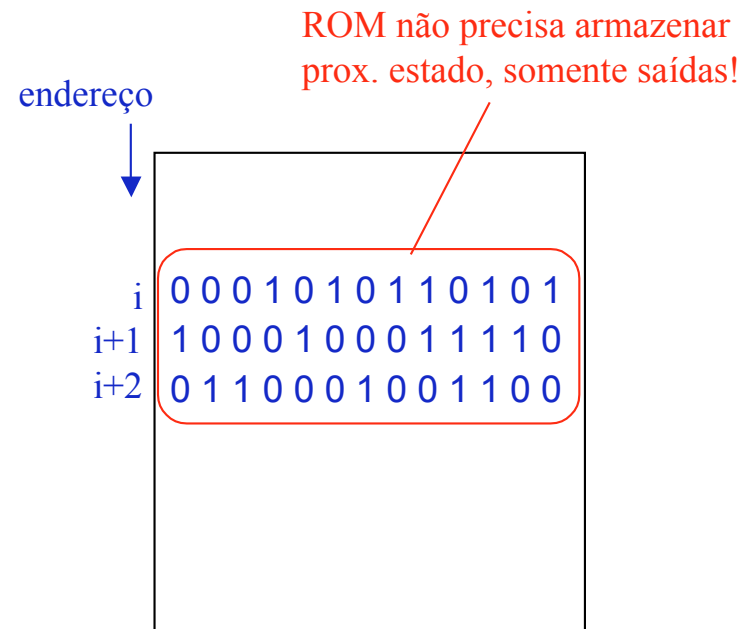
Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Explorando Características da FSM

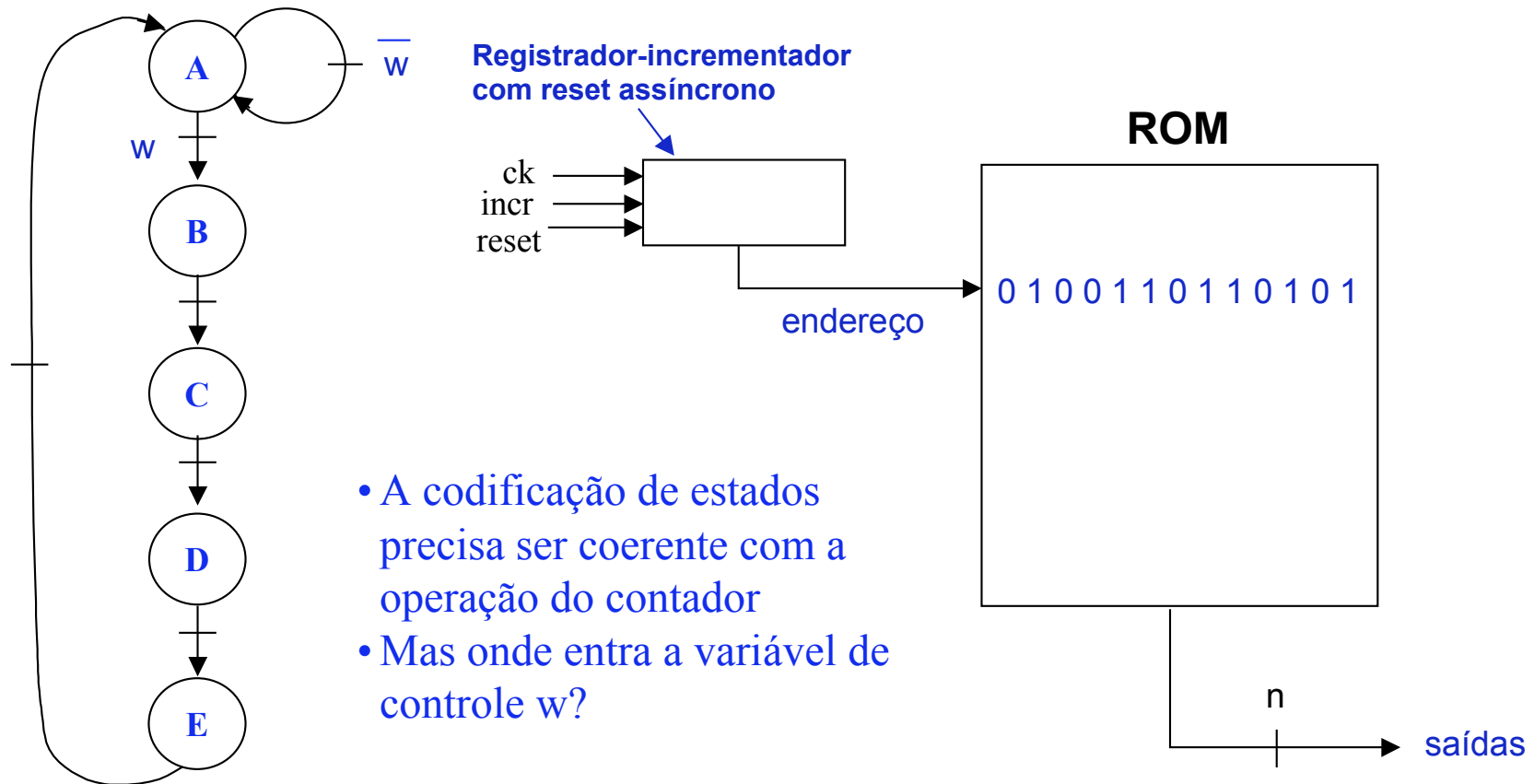


Seqüências de estados
podem ser implementados
como contagem em
binário...



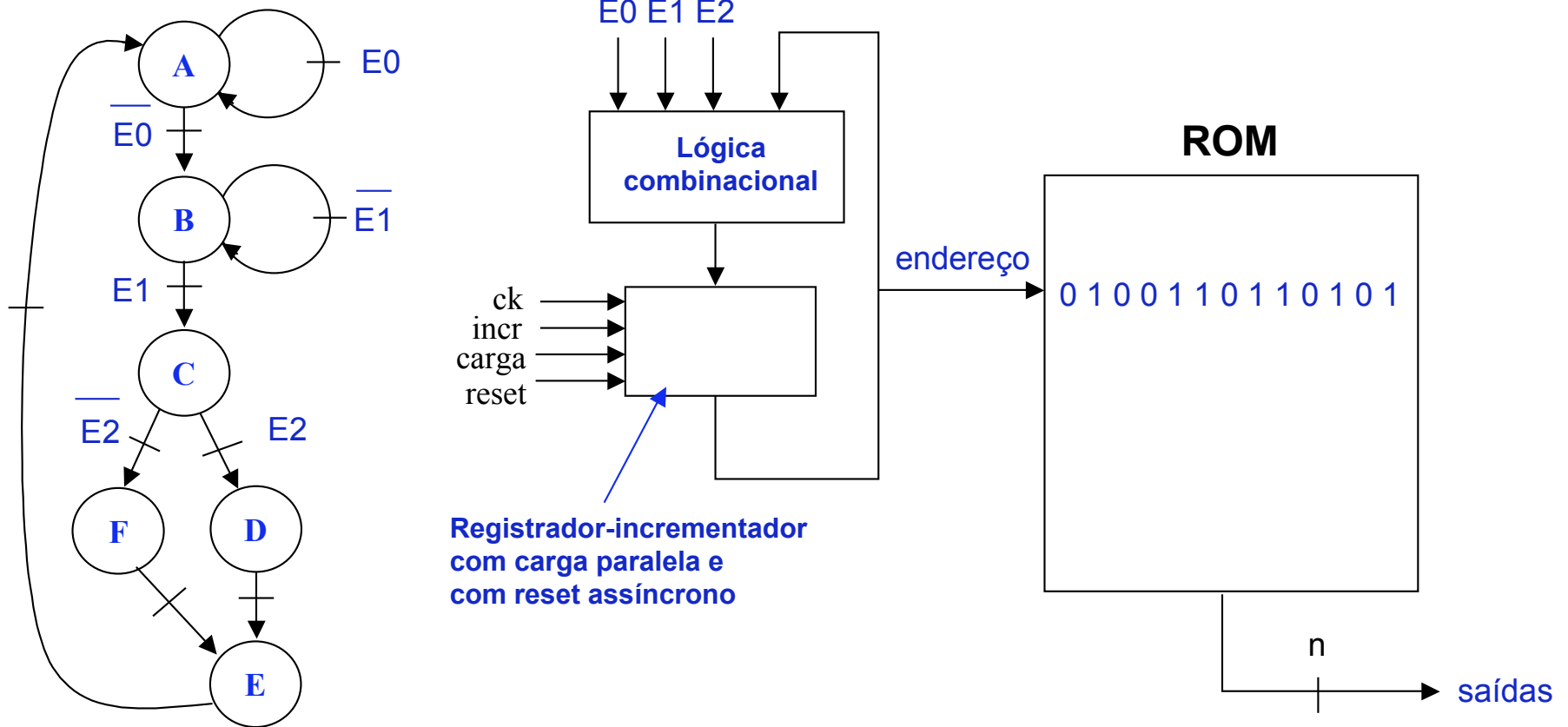
Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs Bloco ROM + Registrador Contador (Incrementador)



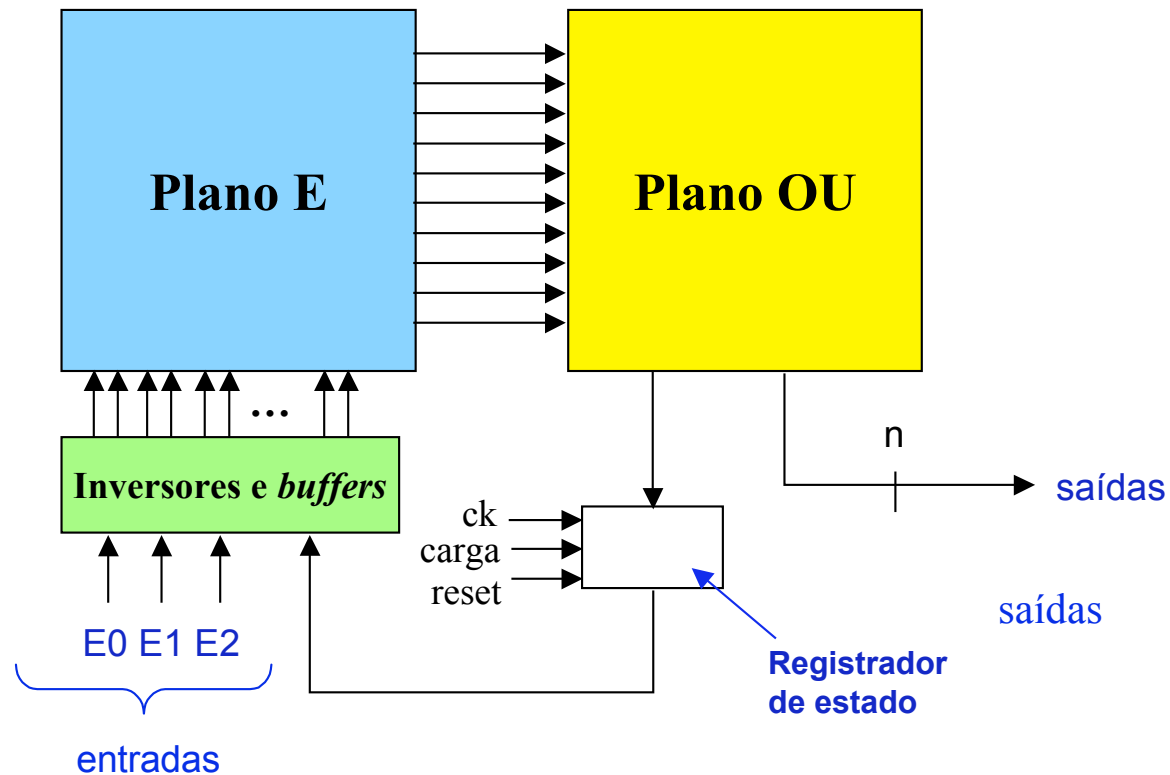
Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs Bloco ROM + Registrador Contador (Incrementador)



Máquinas Seqüenciais Síncronas

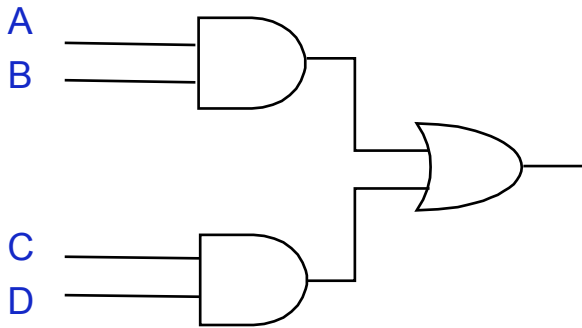
► Alternativas de Implementação de FSMs Lógica de Próximo Estado e Lógica de Saída como PLA



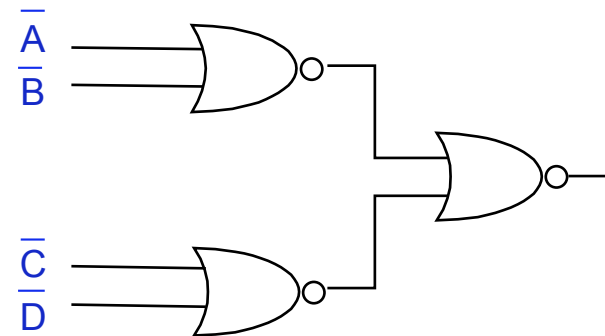
Máquinas Seqüenciais Síncronas

▶ Alternativas de Implementação de FSMs PLA em Tecnologia CMOS

Soma de Produtos (SdP)



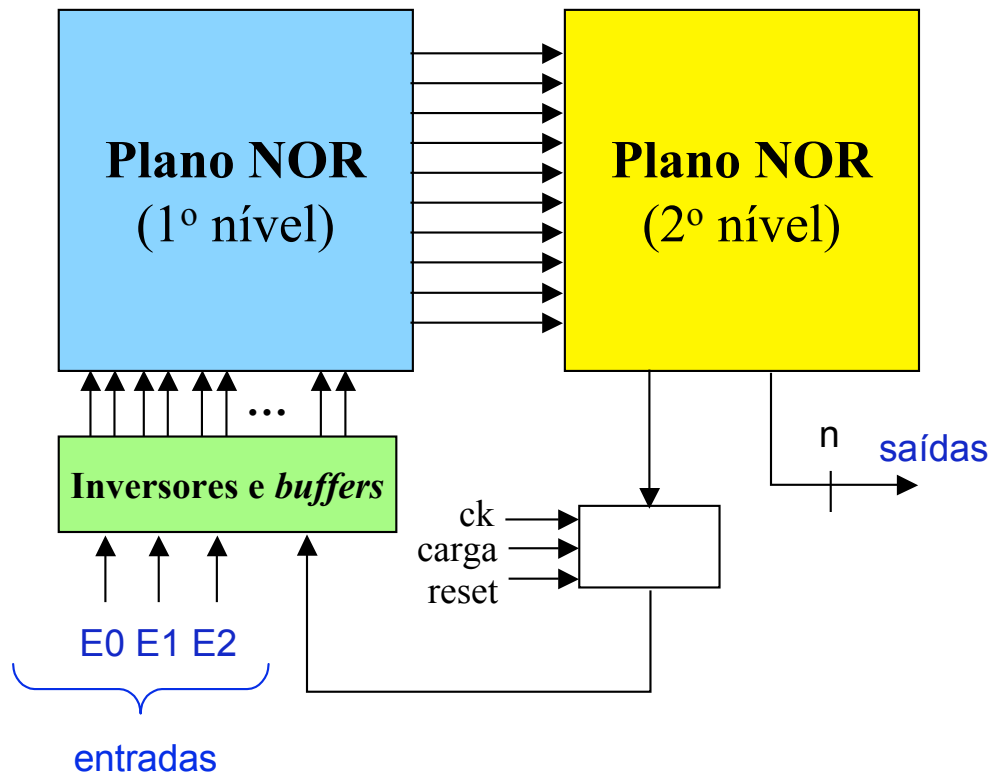
“NOR de NORs”



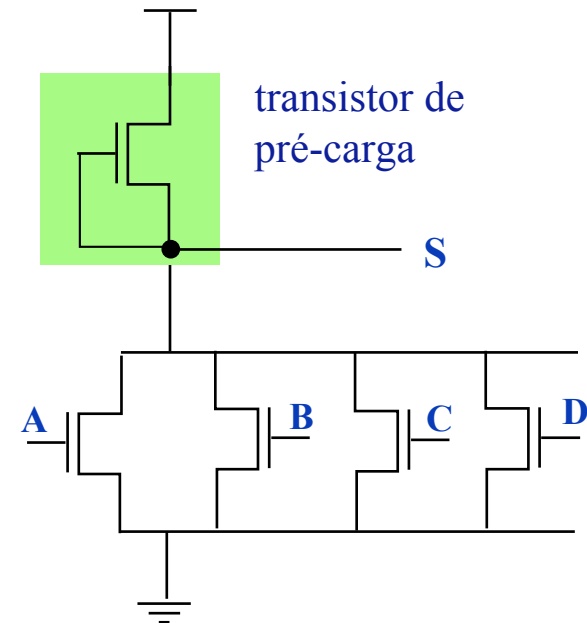
Negando as entradas e aplicando De Morgan...

Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs Lógica de Próximo Estado e Lógica de Saída como PLA

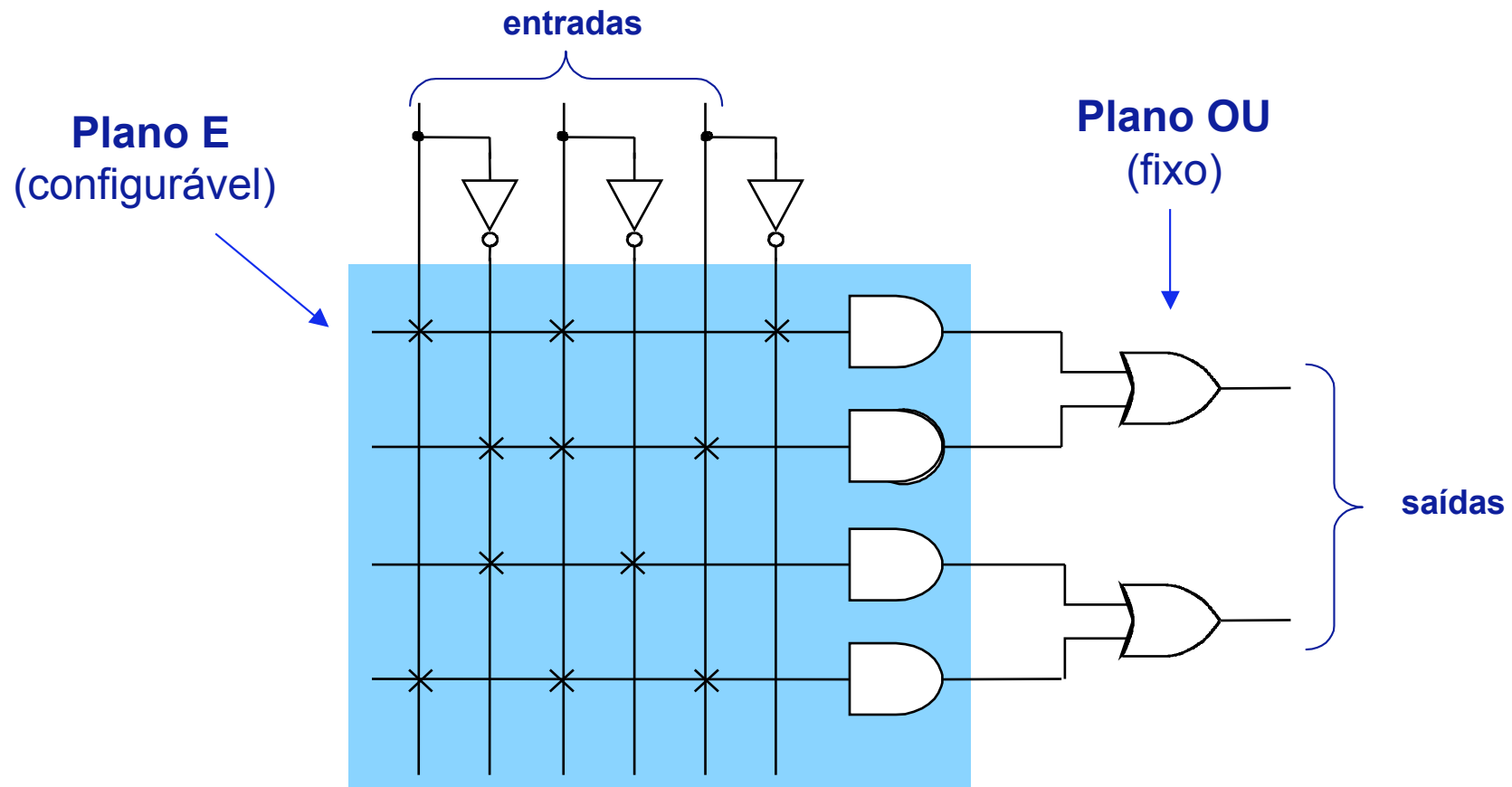


NOR CMOS com muitas entradas



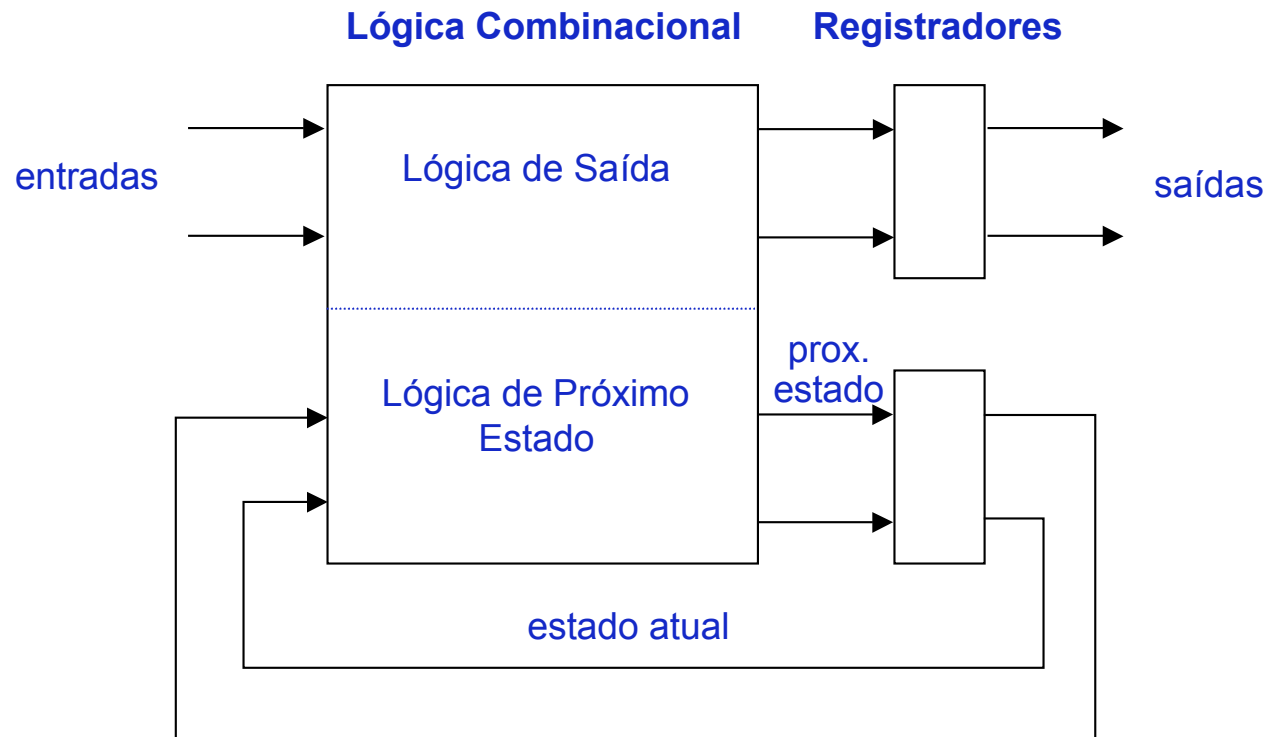
Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs Lógica de Próximo Estado e Lógica de Saída como PAL



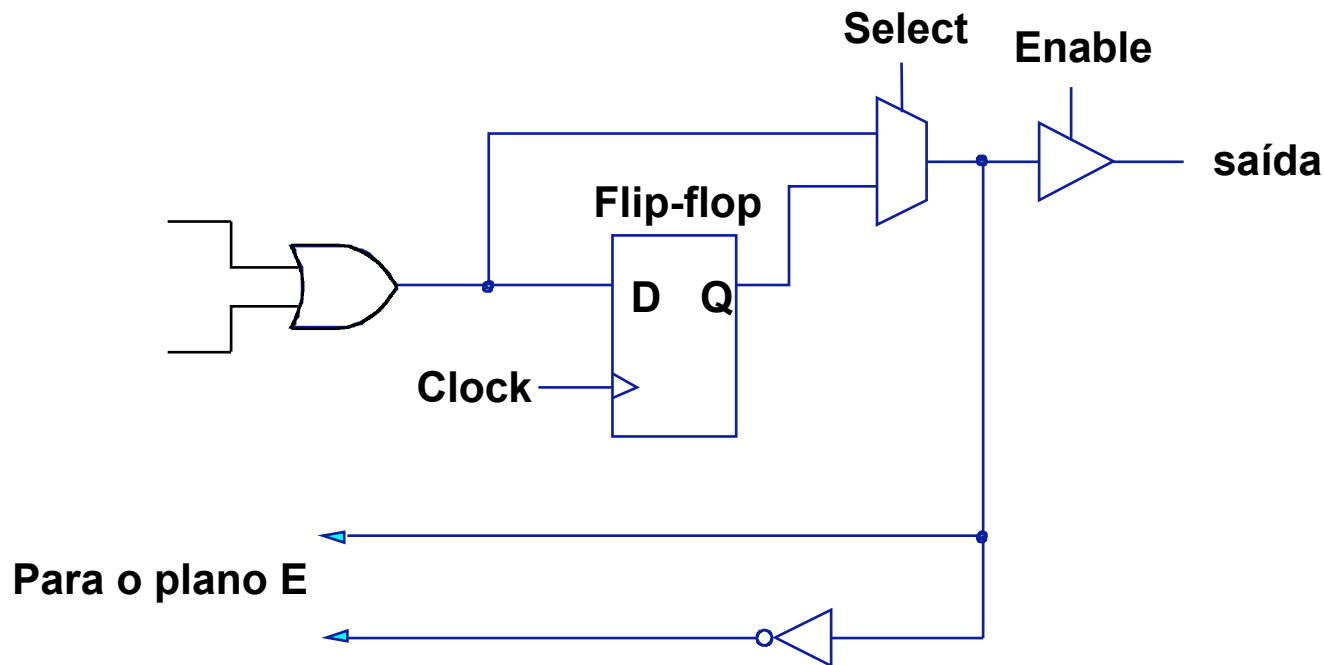
Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs Implementando uma FSM Completa com PAL



Máquinas Seqüenciais Síncronas

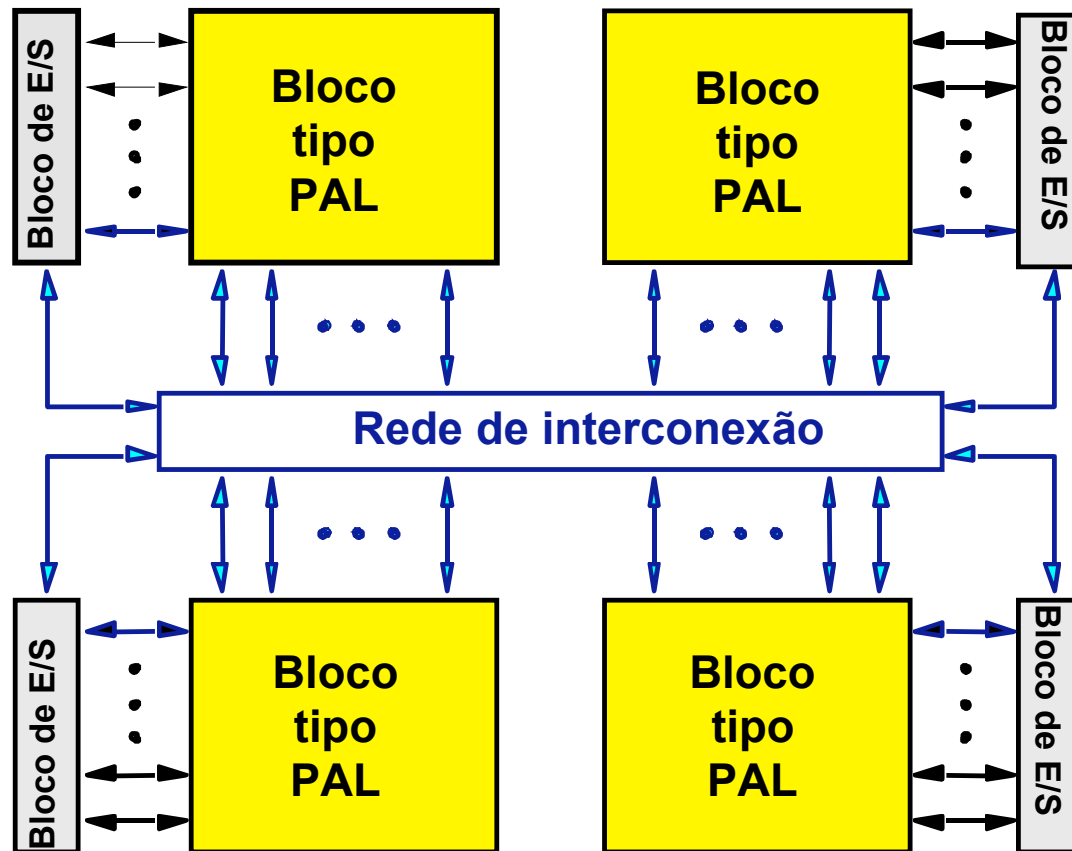
► Alternativas de Implementação de FSMs Implementando uma FSM Completa com PAL



Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Implementando uma FSM Completa em um CPLD



Estrutura de um CPLD

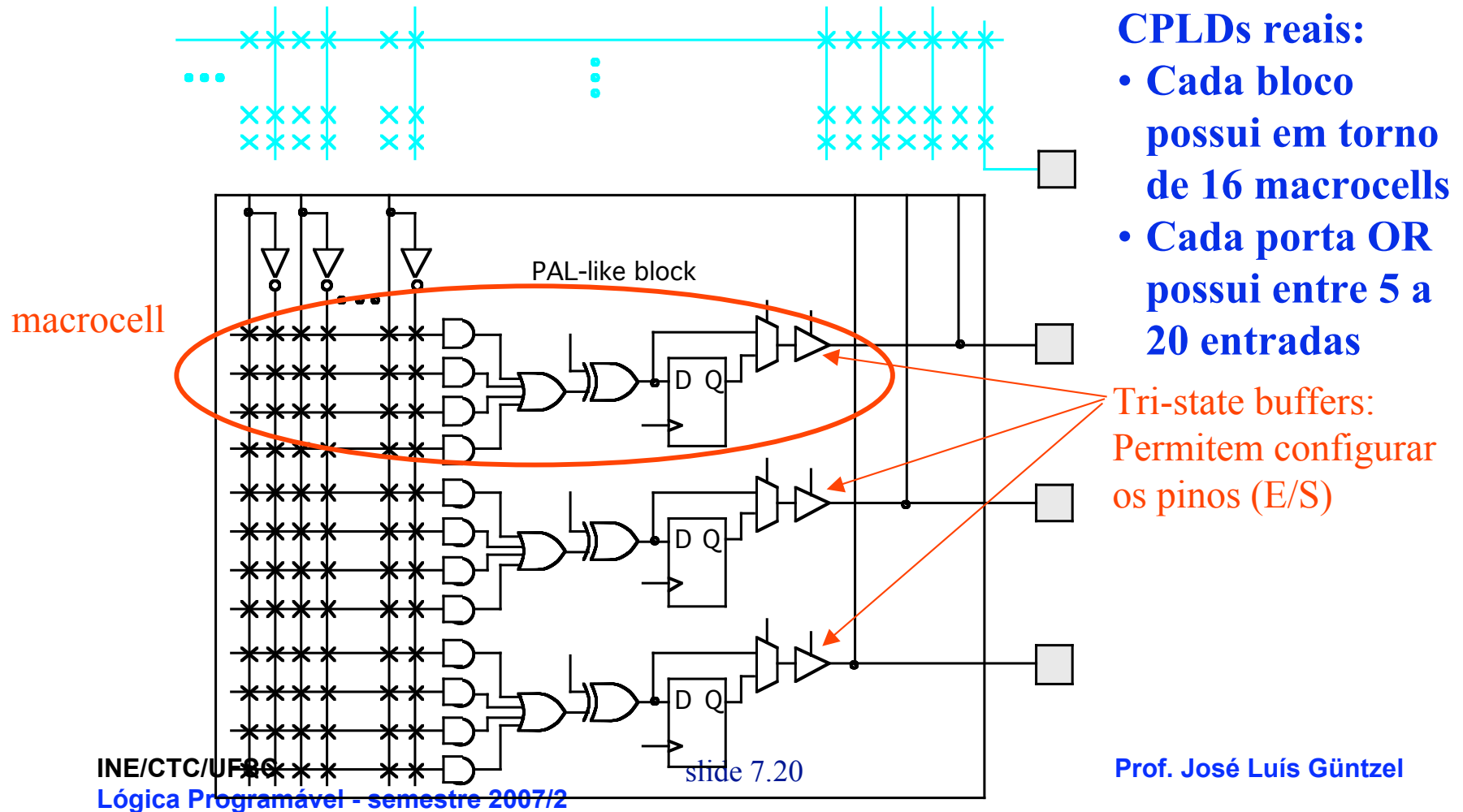
CPLDs existentes no mercado:

- Possuem entre 2 a 100 blocos tipo PAL
- Variedade de encapsulamentos, com até 200 pinos (QFP)

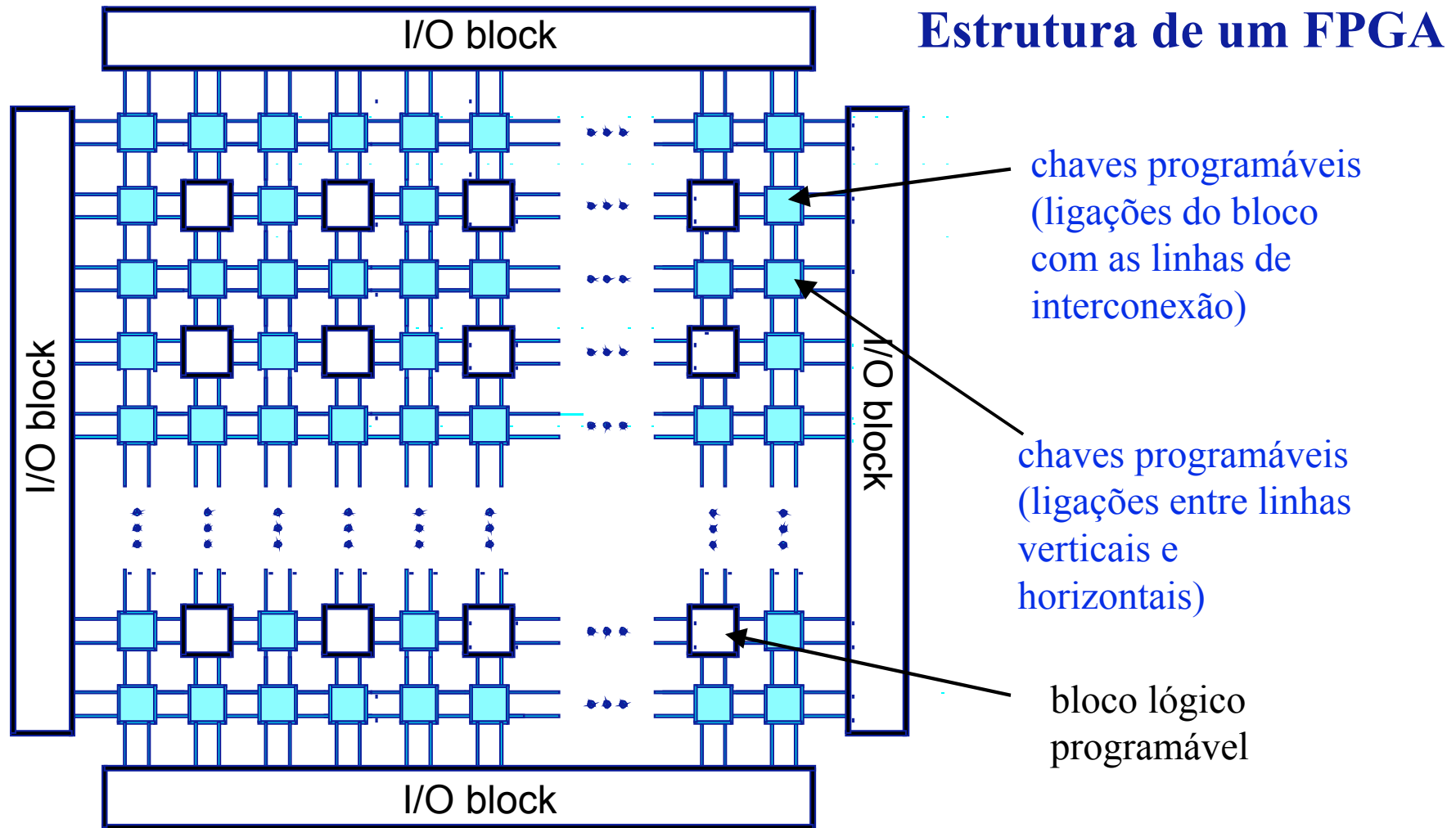
Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs

Implementando uma FSM Completa em um CPLD

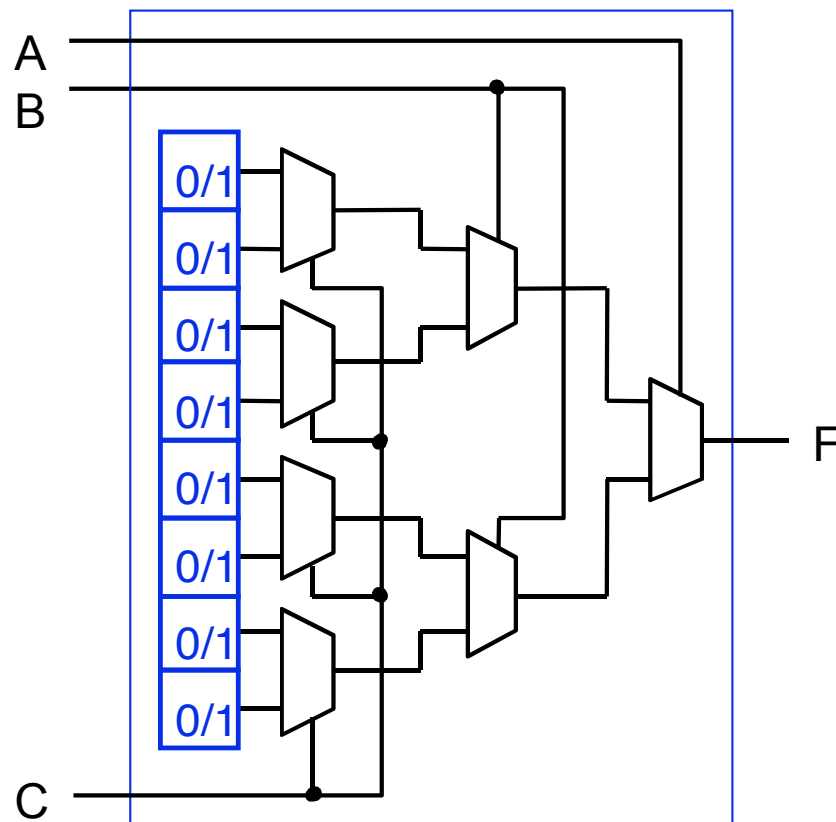


Máquinas Seqüenciais Síncronas



Máquinas Seqüenciais Síncronas

► FPGAs: LUTs (*Lookup Tables*)



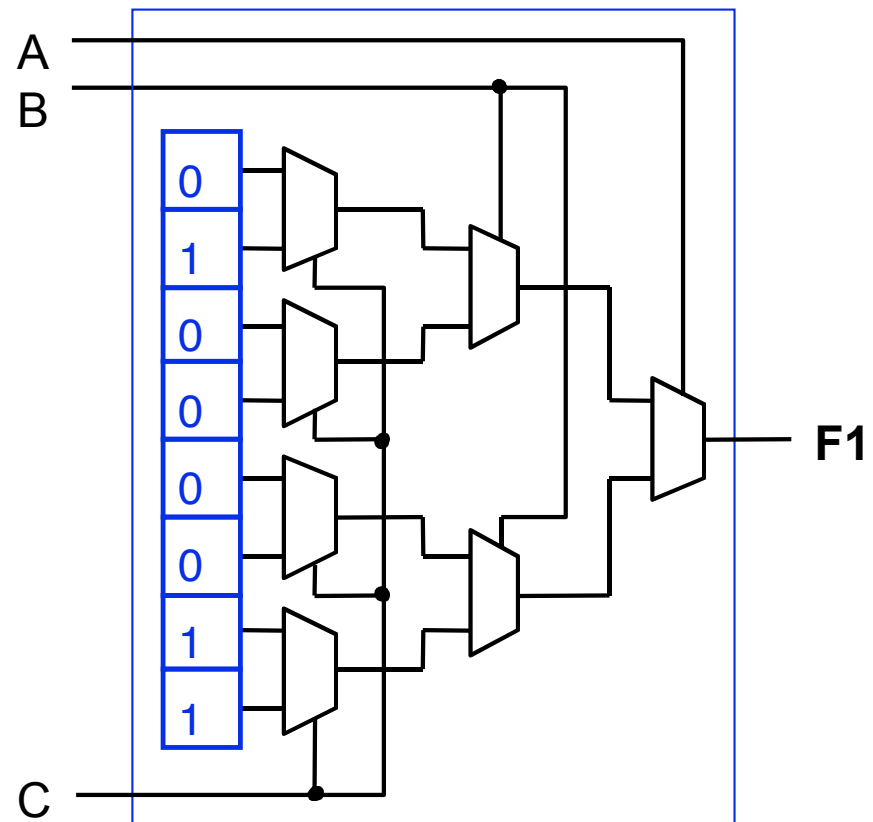
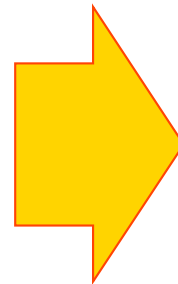
- Implementadas com muxes 2:1 e bits de memória, SRAM (reprogramabilidade...)
- Normalmente, possuem 4 ou 5 entradas
- Implementam qualquer função lógica. Para 4 entradas, existem $2^{2^4} = 65.536$ diferentes funções!!!

Máquinas Seqüenciais Síncronas

► FPGAs: LUTs (*Lookup Tables*)

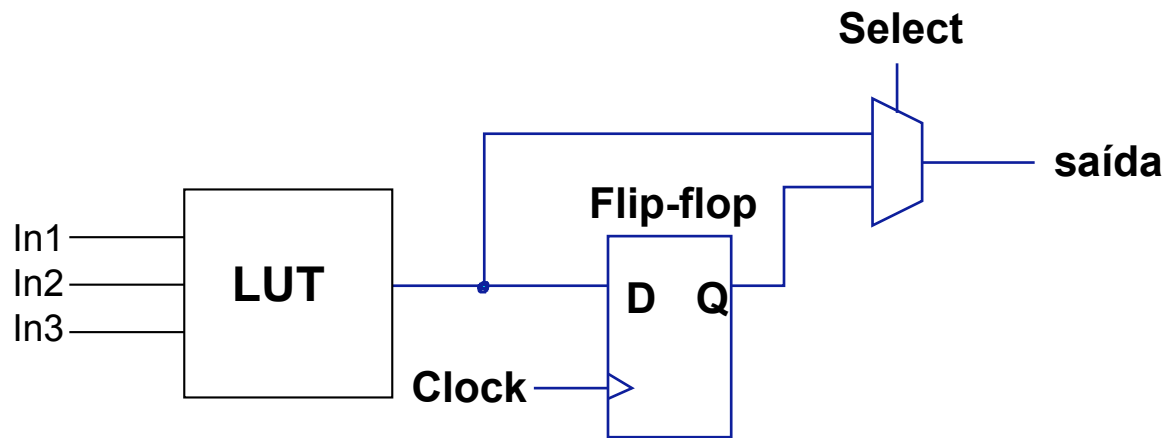
Programando LUTs

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



Máquinas Seqüenciais Síncronas

► Alternativas de Implementação de FSMs Implementando uma FSM Completa em um FPGA



Máquinas Seqüenciais Síncronas

Arquitetura do CLB do dispositivo VIRTEX-II

- Fast Carry Logic Path
- Provides fast arithmetic add and sub

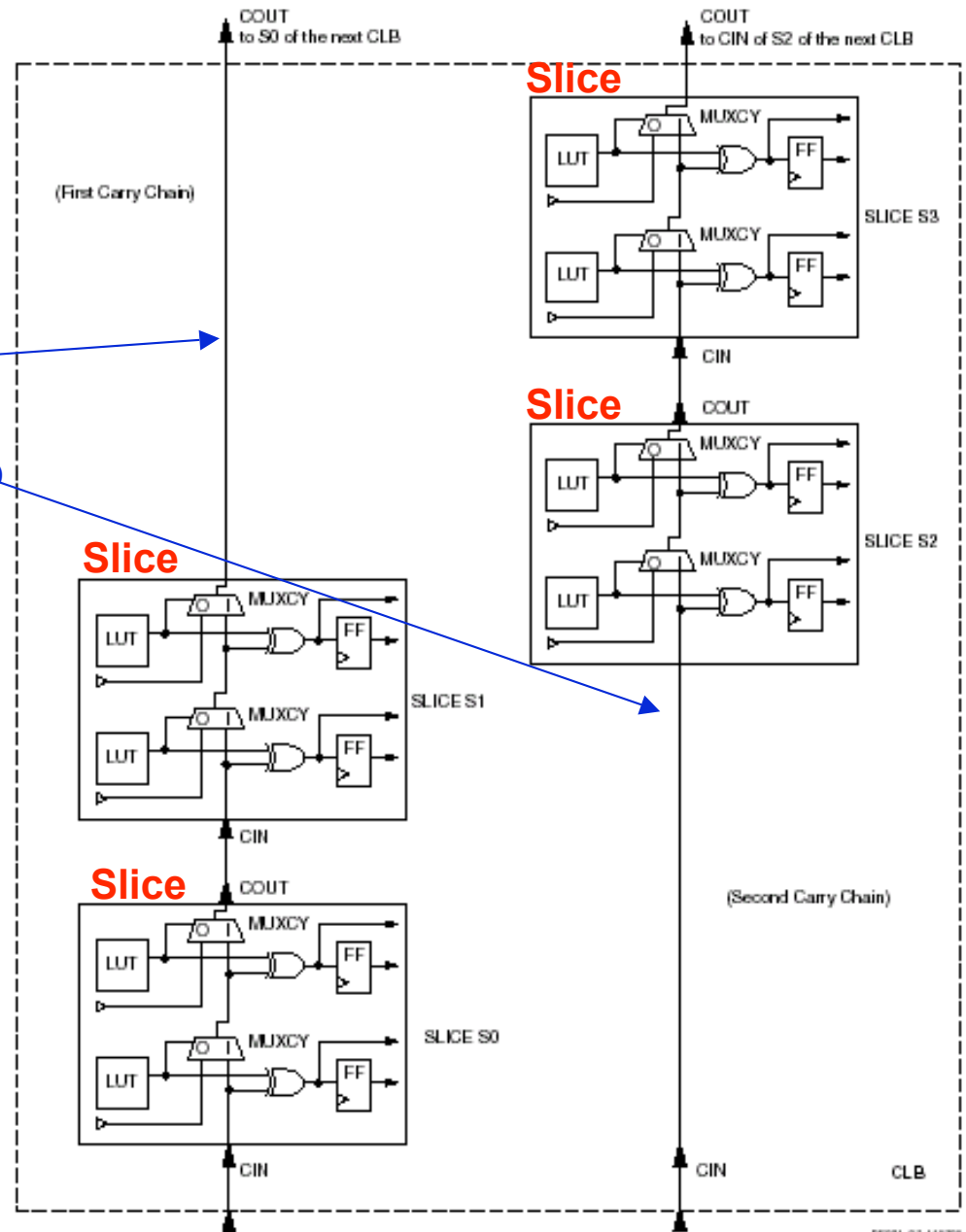
RESUMINDO O CLB

- 4 Slices
- 8 LUTS / 8 Flip-Flops
- 2 cadeias de vai-um
- 64 bits para memória
- 64 bits para shift-register

Transparência de F. Moraes (PUCRS)

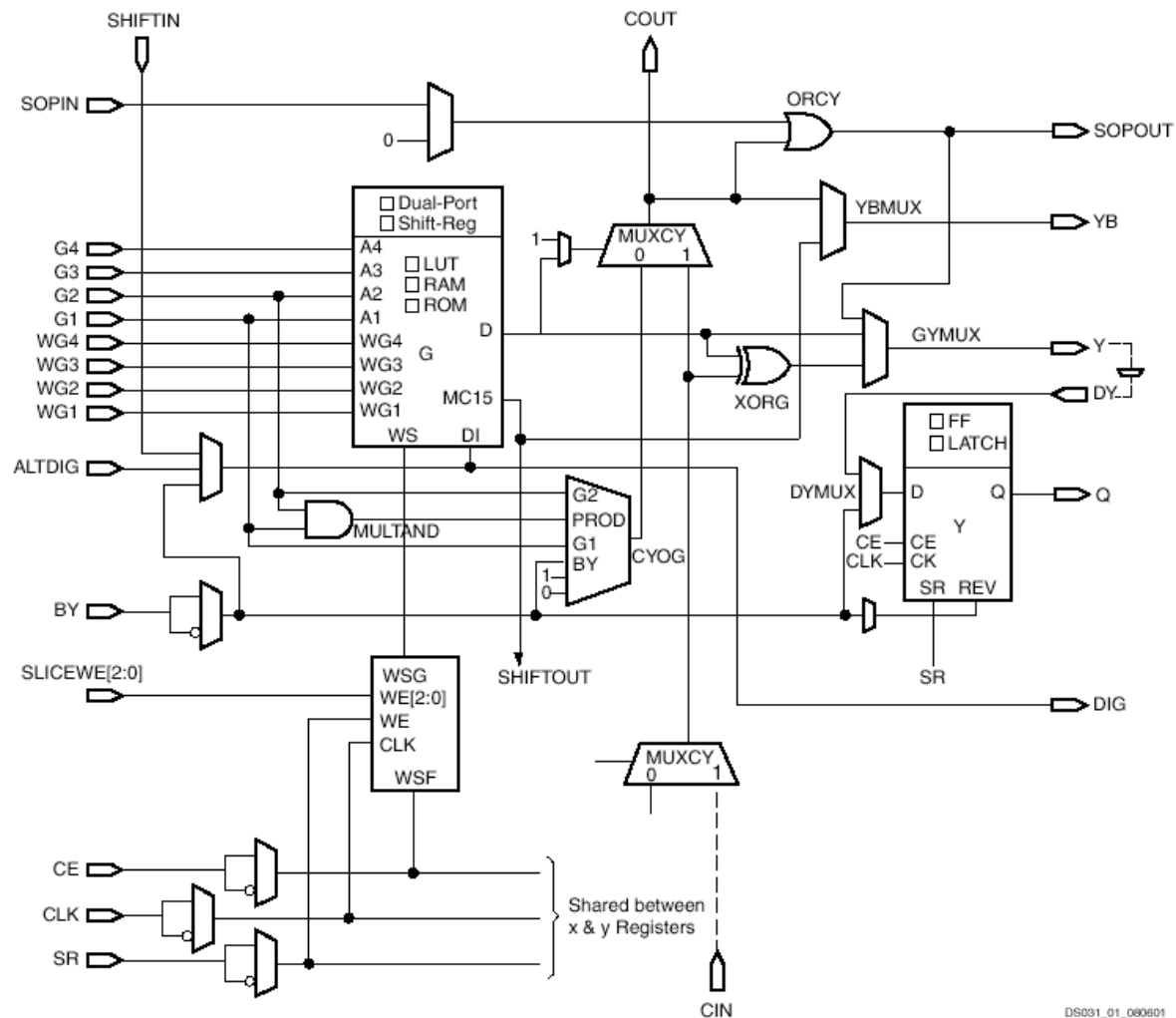
INE/CTC/UFSC

Lógica Programável - semestre 2007/2



Máquinas Seqüenciais Síncronas

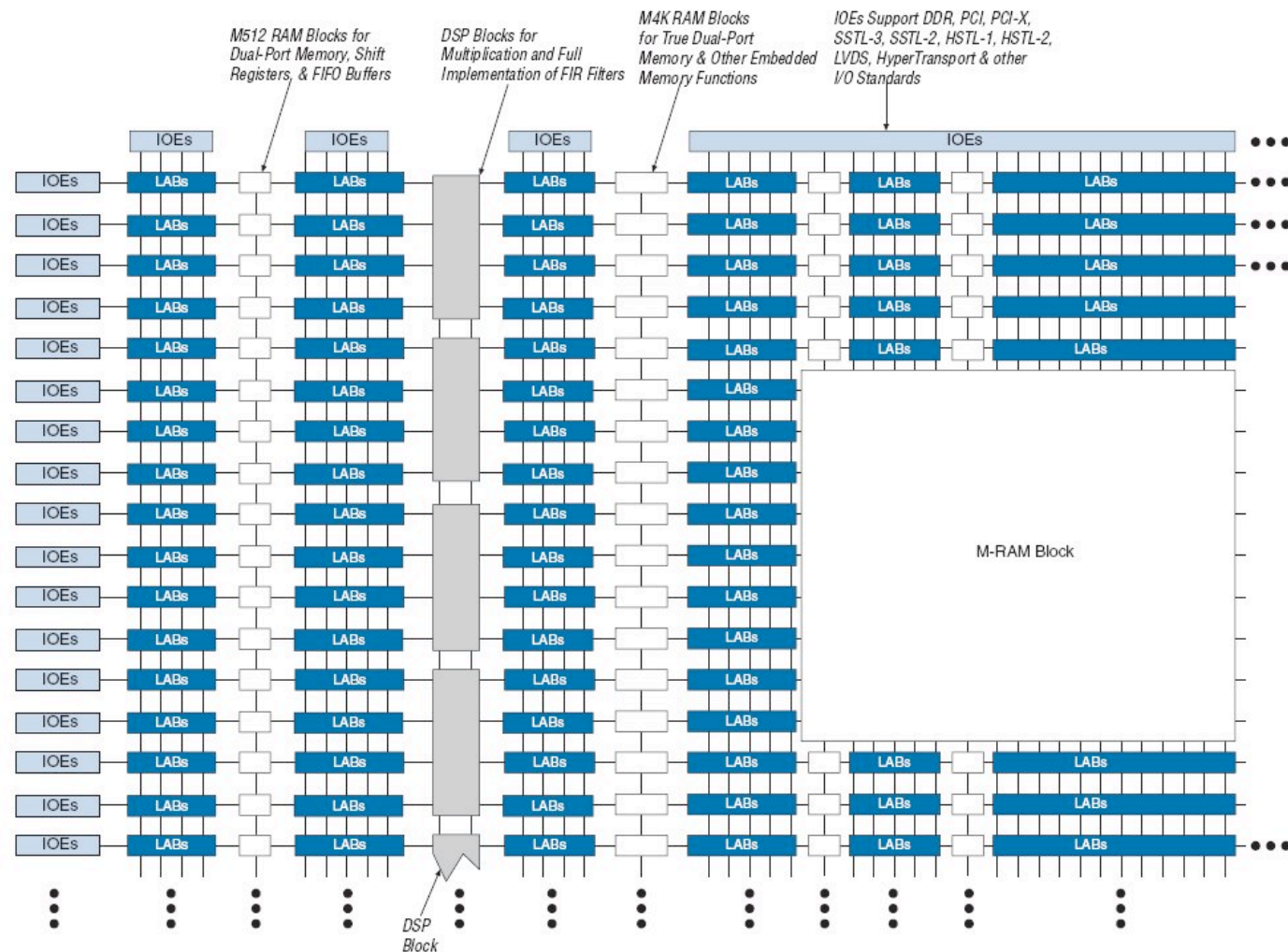
Arquitetura (metade) do Slice



Máquinas Seqüenciais Síncronas

► FPGAs Altera: Stratix II

Estrutura Básica da Matriz do Stratix II



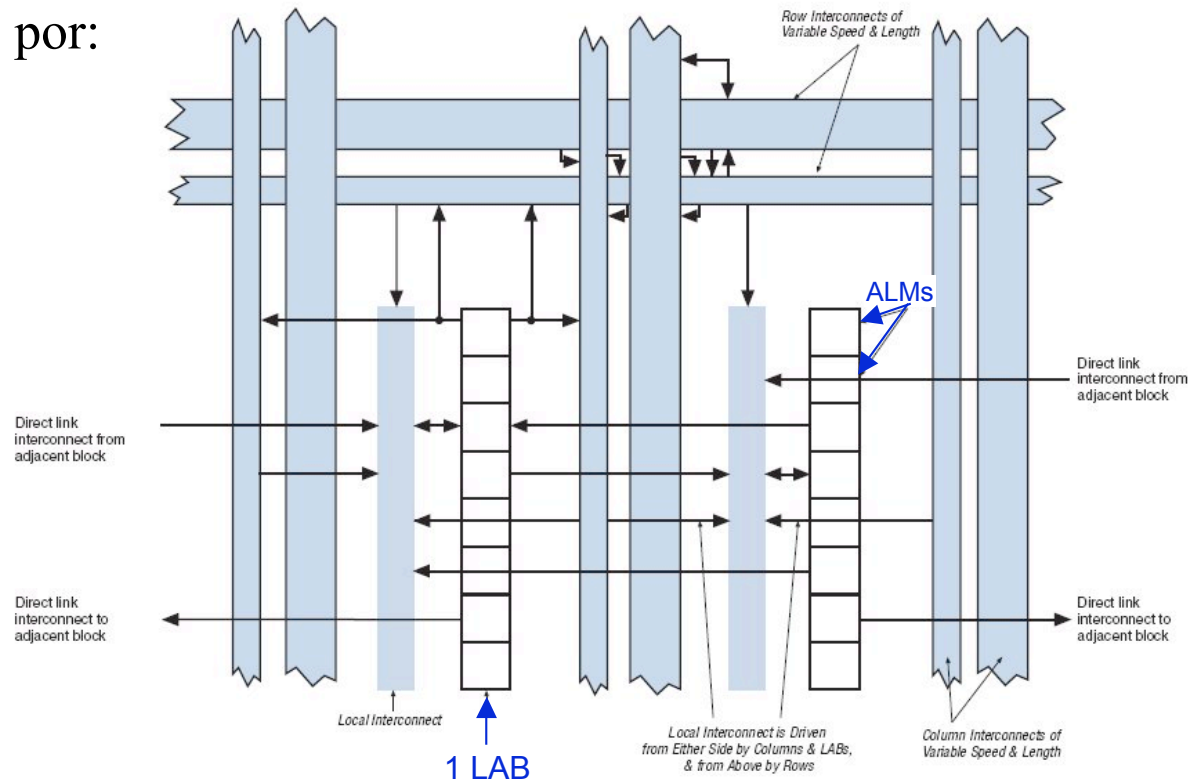
Máquinas Seqüenciais Síncronas

► FPGAs Altera: Stratix II

Estrutura de um LAB (*Logic Array Block*)

Cada LAB é constituído por:

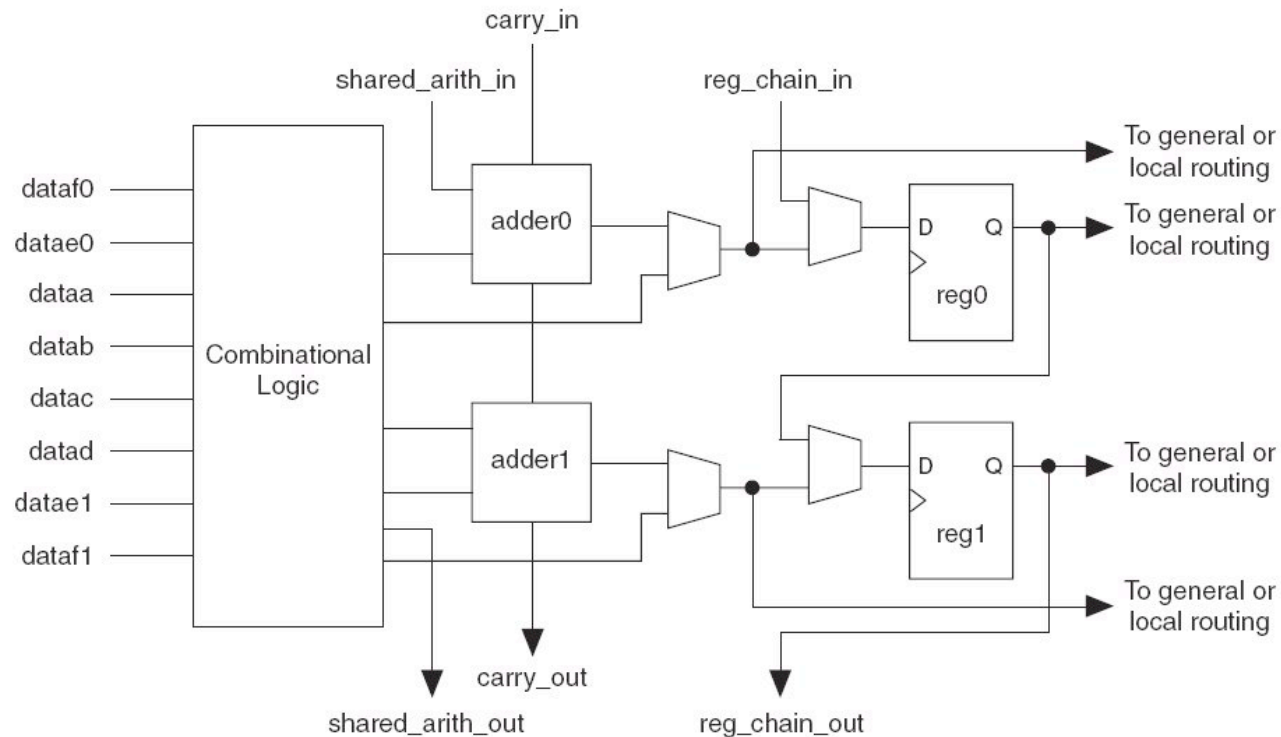
- 8 ALMs (*Adaptative Logic Modules*)
- Cadeia de *carry*
- Cadeia aritmética compartilhada
- Sinais de controle do LAB
- Conexões locais
- Cadeia de registradores



Máquinas Seqüenciais Síncronas

► FPGAs Altera: Stratix II

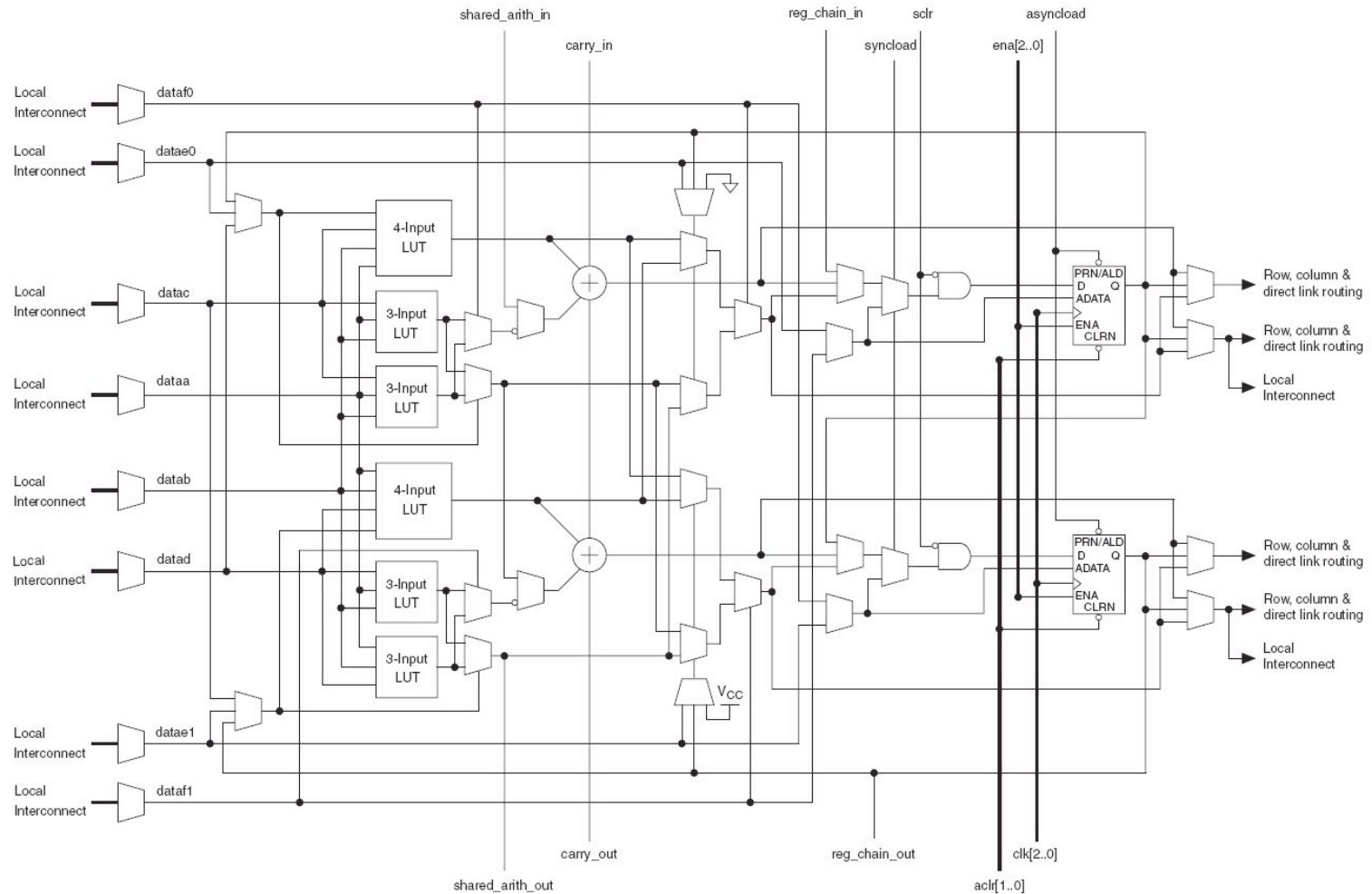
Diagrama de blocos de um ALM



Máquinas Seqüenciais Síncronas

▶ FPGAs Altera: Stratix II

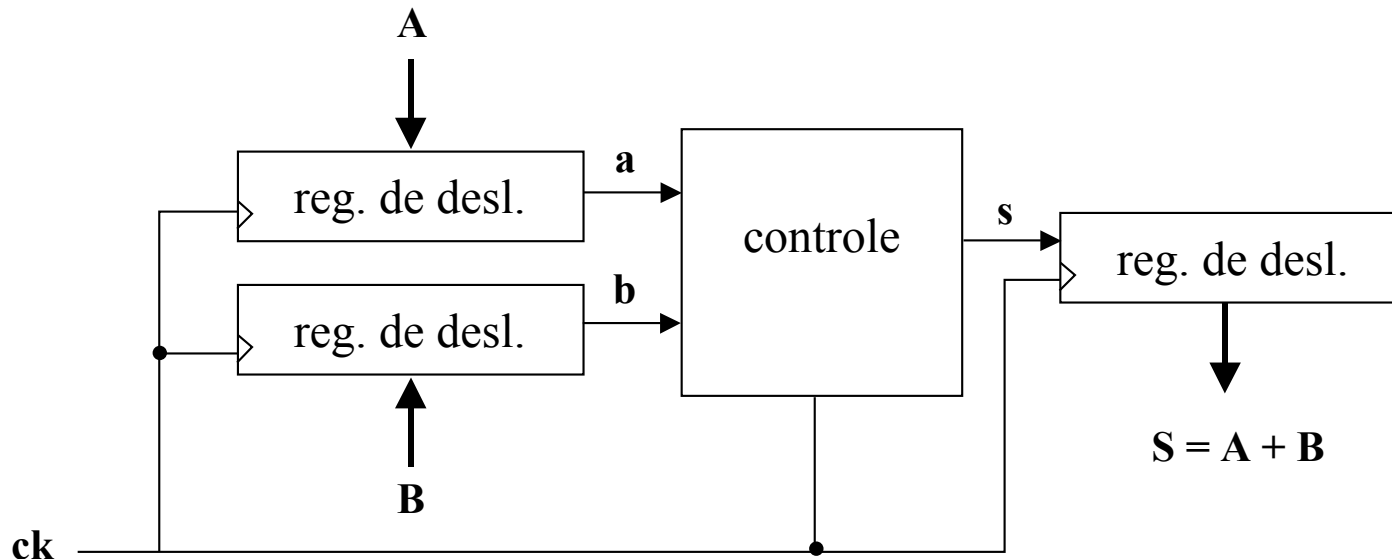
Detalhes de um ALM



Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

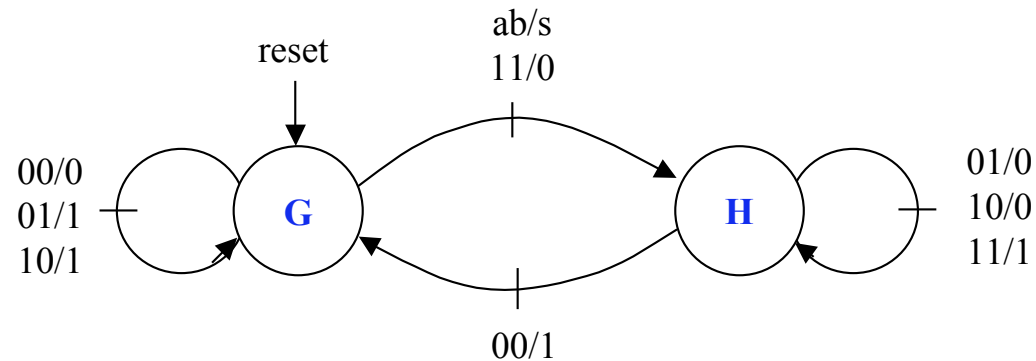
Exemplo 7: Somador (Bit) Serial



Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Mealy



Estado G: carry-in = 0

Estado H: carry-in = 1

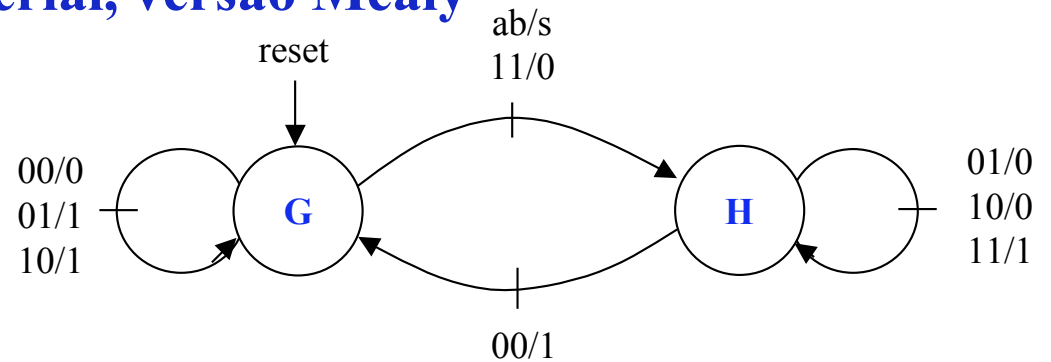
Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Mealy

Tabela de (transição de) estados

Estado atual	a	b	Próximo estado
G	0	0	G
G	0	1	G
G	1	0	G
G	1	1	H
H	0	0	G
H	0	1	H
H	1	0	H
H	1	1	H



Estado G: carry-in = 0

Estado H: carry-in = 1

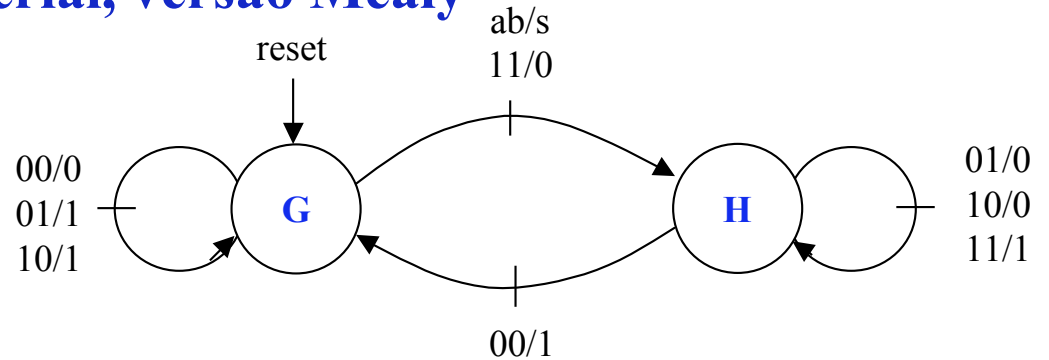
Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Mealy

Tabela de (transição de) estados

y	a	b	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Y	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$	$a \cdot b$
\bar{y}	0	0	1	0	
y	0	1	1	1	$a \cdot y$

$b \cdot y$

$$Y = a \cdot b + a \cdot y + b \cdot y$$

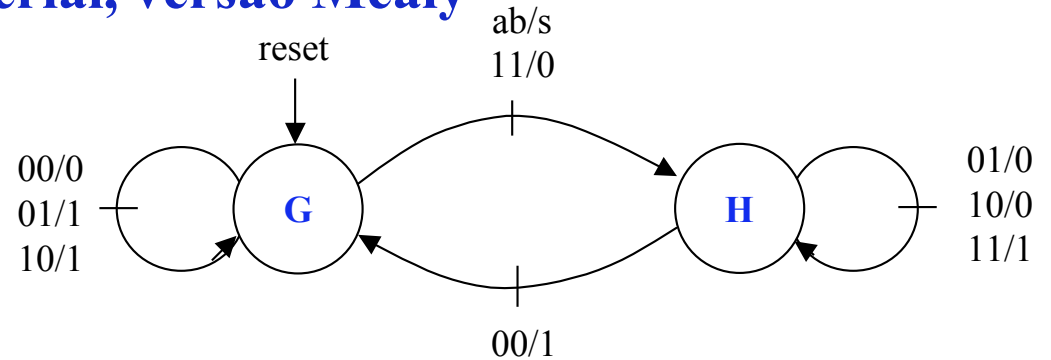
Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Mealy

Tabela de saída

y	a	b	s
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



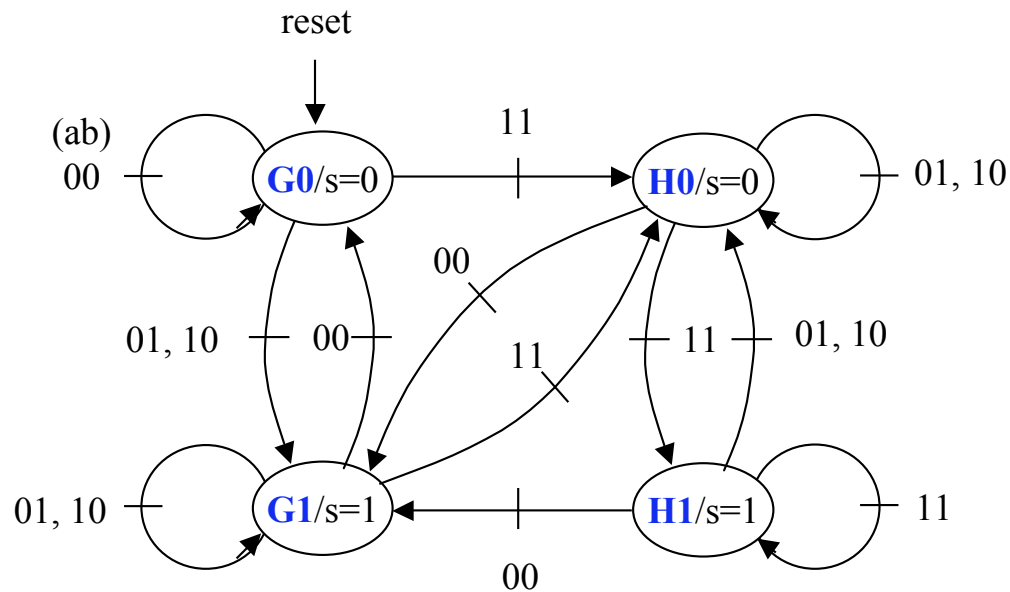
Y	$\bar{a}\bar{b}$	$\bar{a}b$	ab	$a\bar{b}$
\bar{y}	0	1	0	1
y	1	0	1	0

$$s = a \oplus b \oplus y$$

Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Moore



Estados G0 e G1: carry-out = 0

Estados H0 e H1: carry-out = 1

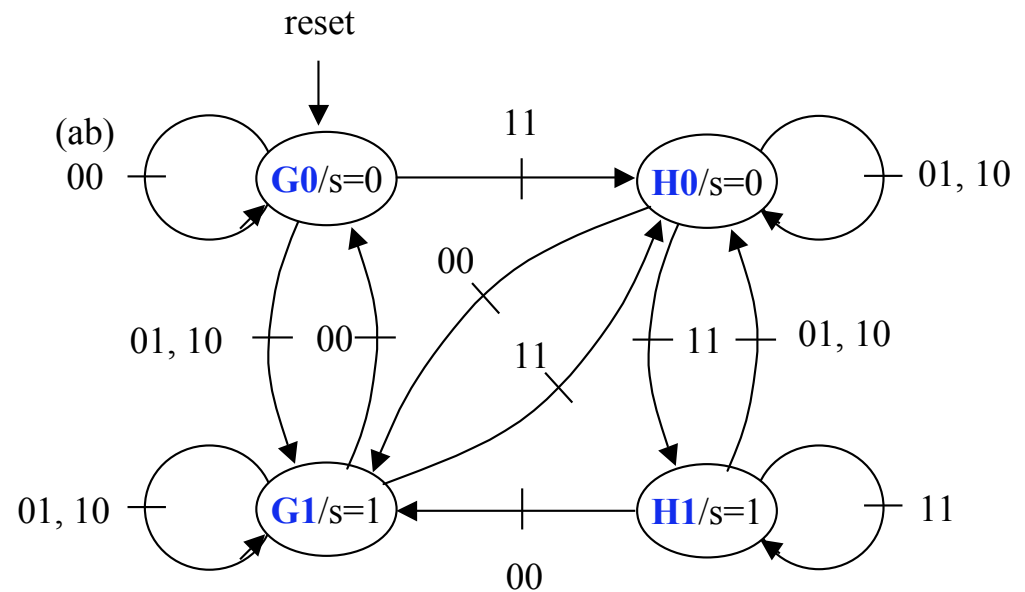
Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Moore

Tabela de estados e de saída

Estado atual	Próximo estado				saída
	00	01	10	01	
G0	G0	G1	G1	H0	0
G1	G0	G1	G1	H0	1
H0	G1	H0	H0	H1	0
H1	G1	H0	H0	H1	1



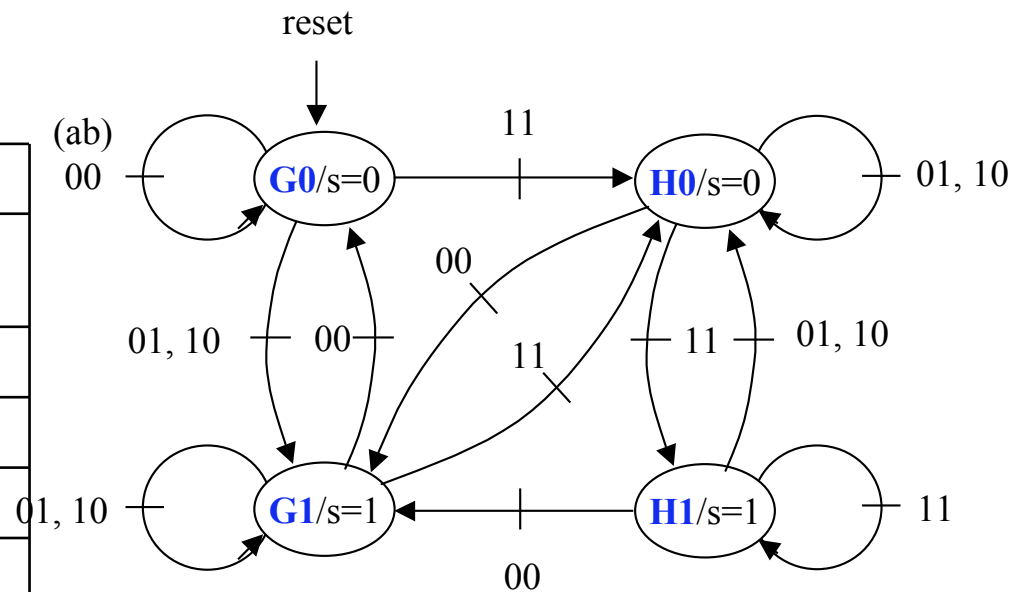
Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Moore

Tabela de estados e de saída
(Após a codificação dos estados)

Estado atual y_1y_0	Próximo estado (Y_1Y_0)				saída s
	00	01	10	01	
00	00	01	01	10	0
01	00	01	01	10	1
10	01	10	10	11	0
11	01	10	10	11	1



Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Moore

Estado atual y_1y_0	Próximo estado (Y_1Y_0)				saída s
	00	01	10	01	
00	00	01	01	10	0
01	00	01	01	10	1
10	01	10	10	11	0
11	01	10	10	11	1

$$Y_0 = a \oplus b \oplus y_1$$

$$Y_1 = a \cdot b + a \cdot y_1 + b \cdot y_1$$

$$s = y_0$$

Máquinas Seqüenciais Síncronas

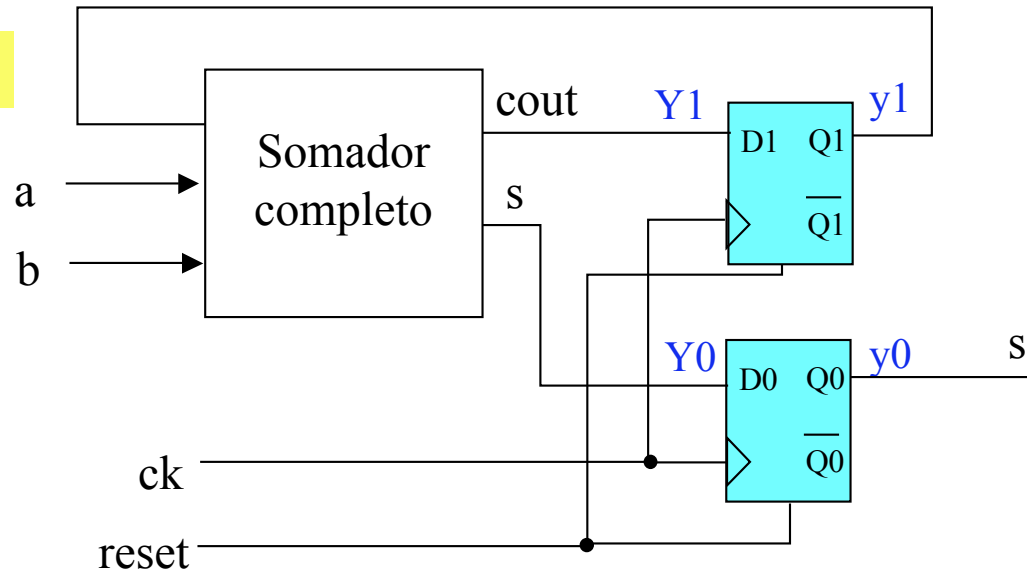
► Síntese de Circuitos Seqüenciais

Exemplo 7: Somador (Bit) Serial, versão Moore

$$Y0 = a \oplus b \oplus y1$$

$$Y1 = a \cdot b + a \cdot y1 + b \cdot y1$$

$$s = y0$$



Máquinas Seqüenciais Síncronas

VHDL para o Exemplo 7

Deslocador

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

-- registrador-deslocador para a direita, com carga paralela e sinal de enable
ENTITY shiftrne IS
    GENERIC ( N : INTEGER := 4 ) ;
    PORT ( R          : IN          STD_LOGIC_VECTOR(N-1 DOWNT0 0) ;
          L, E, w     : IN          STD_LOGIC ;
          Clock       : IN          STD_LOGIC ;
          Q           : BUFFER      STD_LOGIC_VECTOR(N-1 DOWNT0 0) ) ;
END shiftrne ;

ARCHITECTURE Behavior OF shiftrne IS
BEGIN
    PROCESS
    BEGIN
        WAIT UNTIL Clock'EVENT AND Clock = '1' ;
        IF E = '1' THEN
            IF L = '1' THEN
                Q <= R ;
            ELSE
                Genbits: FOR i IN 0 TO N-2 LOOP
                    Q(i) <= Q(i+1) ;
                END LOOP ;
                Q(N-1) <= w ;
            END IF ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

Retirado de BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with VHDL Design. McGraw-Hill, 2000.

Máquinas Seqüenciais Síncronas

VHDL para o Exemplo 7

Deslocador

```
1 LIBRARY ieee ;
2 USE ieee.std_logic_1164.all ;

3 ENTITY serial IS
4     GENERIC ( length : INTEGER := 8 ) ;
5     PORT ( Clock : IN          STD_LOGIC ;
6           Reset  : IN          STD_LOGIC ;
7           A, B   : IN          STD_LOGIC_VECTOR(length-1 DOWNTO 0) ;
8           Sum   : BUFFER      STD_LOGIC_VECTOR(length-1 DOWNTO 0) );
9 END serial ;

10 ARCHITECTURE Behavior OF serial IS
11     COMPONENT shiftreg
12         GENERIC ( N : INTEGER := 4 ) ;
13         PORT ( R      : IN          STD_LOGIC_VECTOR(N-1 DOWNTO 0) ;
14               L, E, w : IN          STD_LOGIC ;
15               Clock  : IN          STD_LOGIC ;
16               Q      : BUFFER      STD_LOGIC_VECTOR(N-1 DOWNTO 0) ) ;
17     END COMPONENT ;

18 SIGNAL QA, QB, Null_in : STD_LOGIC_VECTOR(length-1 DOWNTO 0) ;
19 SIGNAL s, Low, High, Run : STD_LOGIC ;
20 SIGNAL Count : INTEGER RANGE 0 TO length ;
21 TYPE State_type IS (G, H) ;
22 SIGNAL y : State_type ;
    continua no próximo slide
```

Retirado de BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with VHDL Design. McGraw-Hill, 2000.

Máquinas Seqüenciais Síncronas

VHDL para o Exemplo 7

Deslocador

```
23 BEGIN
24     Low <= '0' ; High <= '1' ;
25     ShiftA: shiftreg GENERIC MAP (N => length)
26         PORT MAP ( A, Reset, High, Low, Clock, QA ) ;
27     ShiftB: shiftreg GENERIC MAP (N => length)
28         PORT MAP ( B, Reset, High, Low, Clock, QB ) ;
29     AdderFSM: PROCESS ( Reset, Clock )
30     BEGIN
31         IF Reset = '1' THEN
32             y <= G ;
33         ELSIF Clock'EVENT AND Clock = '1' THEN
34             CASE y IS
35                 WHEN G =>
36                     IF QA(0) = '1' AND QB(0) = '1' THEN y <= H ;
37                     ELSE y <= G ;
38                     END IF ;
39                 WHEN H =>
40                     IF QA(0) = '0' AND QB(0) = '0' THEN y <= G ;
41                     ELSE y <= H ;
42                     END IF ;
43             END CASE ;
44         END IF ;
45     END PROCESS AdderFSM ;
```

Continua no próximo slide

Retirado de BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with VHDL Design. McGraw-Hill, 2000.

Máquinas Seqüenciais Síncronas

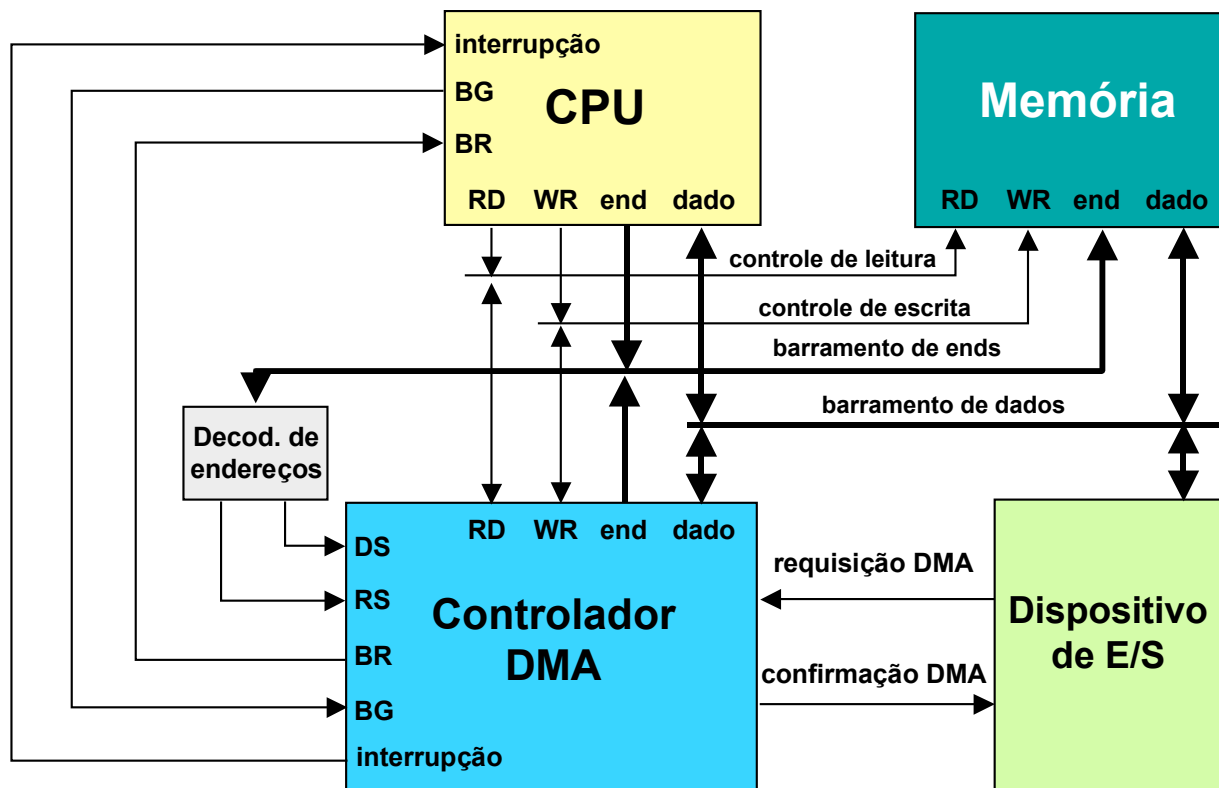
```
46 WITH y SELECT
47     s <= QA(0) XOR QB(0) WHEN G,
48         NOT ( QA(0) XOR QB(0) ) WHEN H ;
49 Null_in <= (OTHERS => '0') ;
50 ShiftSum: shiftrne GENERIC MAP ( N => length )
51         PORT MAP ( Null_in, Reset, Run, s, Clock, Sum ) ;
52 Stop: PROCESS
53 BEGIN
54     WAIT UNTIL (Clock'EVENT AND Clock = '1') ;
55     IF Reset = '1' THEN
56         Count <= length ;
57     ELSIF Run = '1' THEN
58         Count <= Count -1 ;
59     END IF ;
60 END PROCESS ;
61 Run <= '0' WHEN Count = 0 ELSE '1' ; -- stops counter and ShiftSum
62 END Behavior ;
```

Retirado de BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with VHDL Design. McGraw-Hill, 2000.

Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA



Sinais:

BR= Bus Request

BG= Bus Grant

RD= Read

WR= Write

DS= DMA select

RS= register select

Máquinas Seqüenciais Síncronas

▶ Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA

Passo 1: processador ativa o DMAC (controlador de DMA)

- Informando a identidade do dispositivo
- A operação a ser realizada pelo dispositivo
- O endereço de memória que é a fonte ou o destino do dado e
- O número de bytes a serem transferidos

Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA

Passo 2: O DMAC inicia a operação sobre o dispositivo:

- Arbitrando o barramento
- Quando os dados estiverem disponíveis ele os transfere
- Fornece os endereços de memória a serem lidos ou escritos
- Gera o próximo endereço de memória e inicia a nova transferência...

Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA

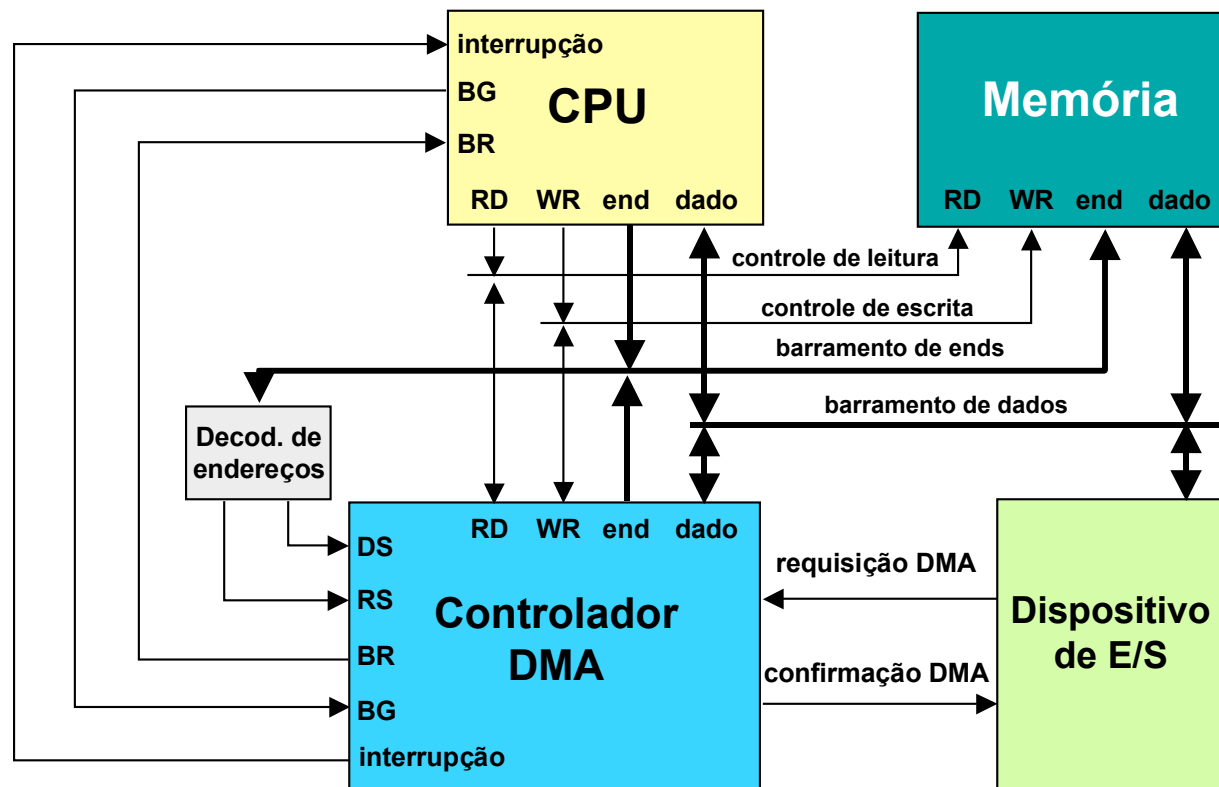
Passo 3: Uma vez completada a transferência sob o controle do DMAC:

- O DMAC interrompe o processador,
- O processador deve verificar se toda a operação de transferência foi realizada com sucesso.
- O processador retoma o controle do barramento

Máquinas Seqüenciais Síncronas

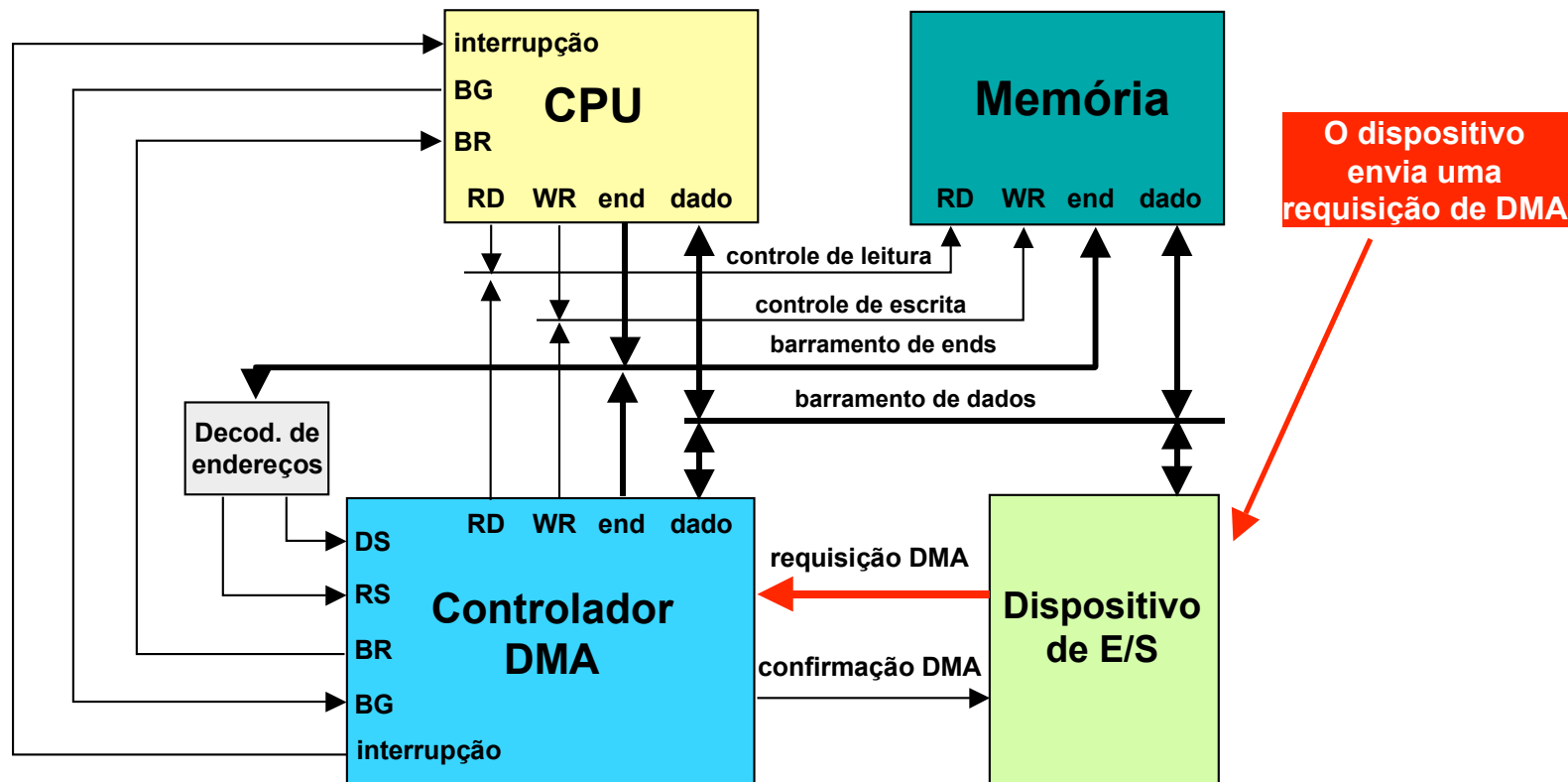
▶ Acesso Direto à Memória (DMA)

O controlador de DMA no sistema



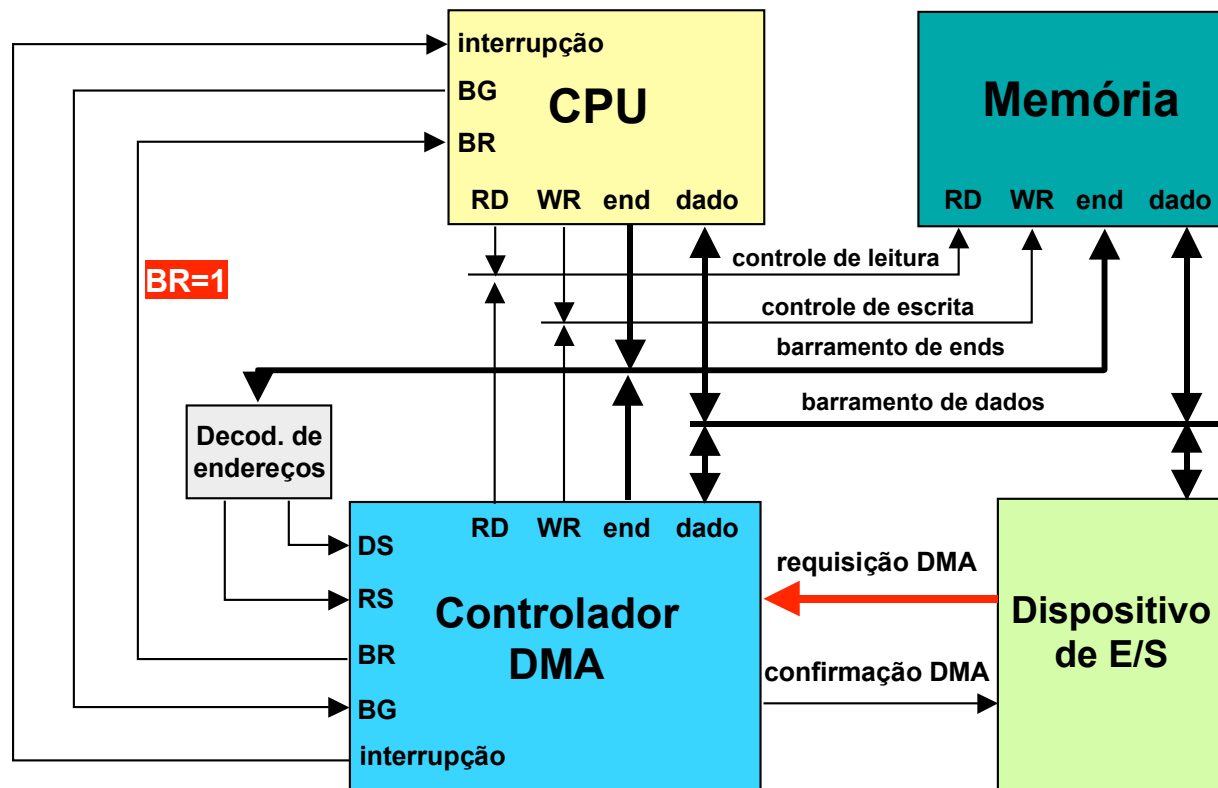
Máquinas Seqüenciais Síncronas

▶ Transferência DMA



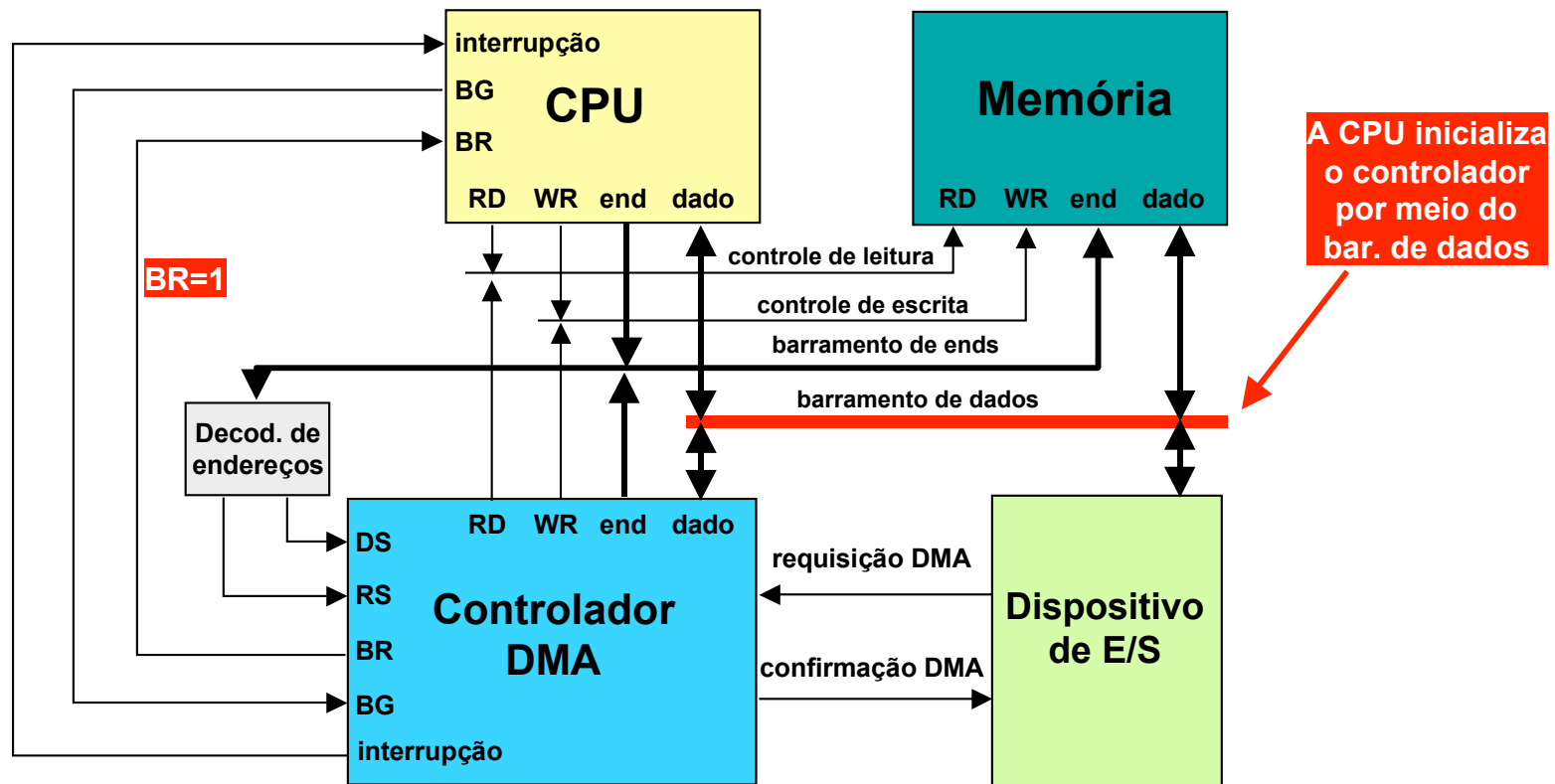
Máquinas Seqüenciais Síncronas

► Transferência DMA



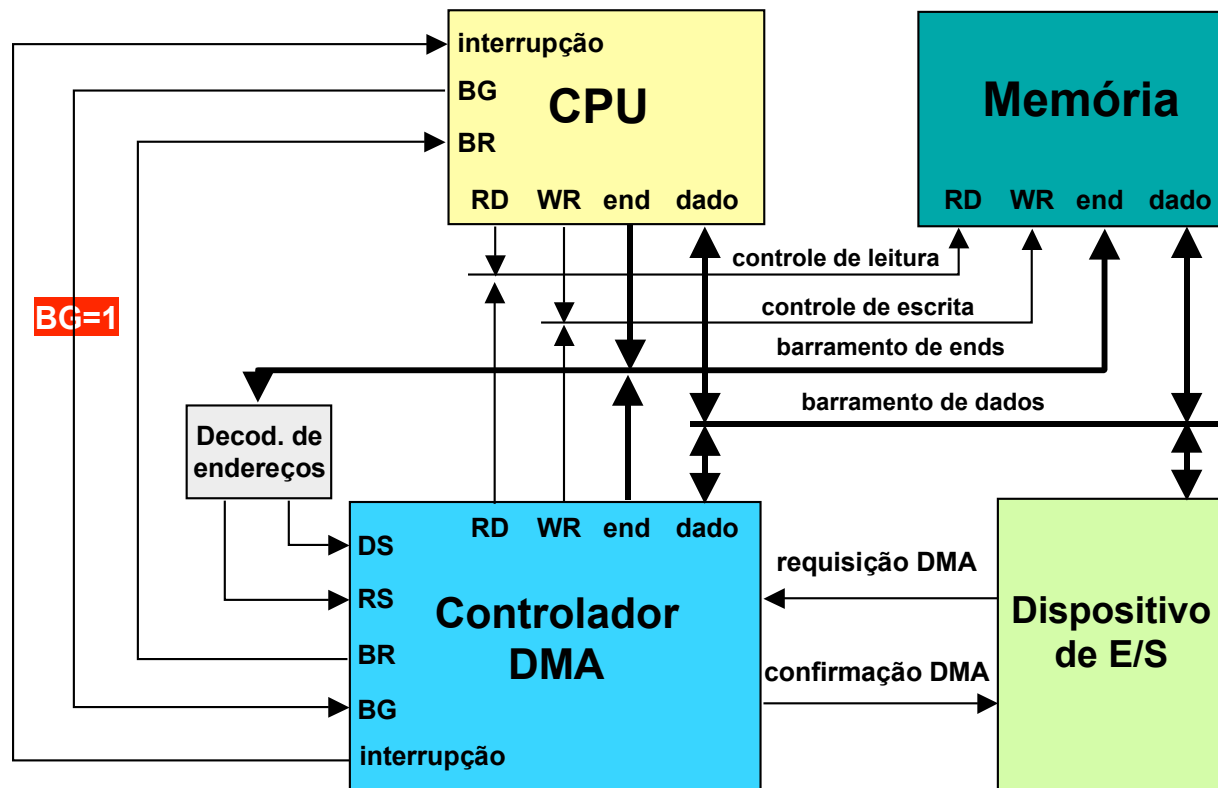
Máquinas Seqüenciais Síncronas

▶ Transferência DMA



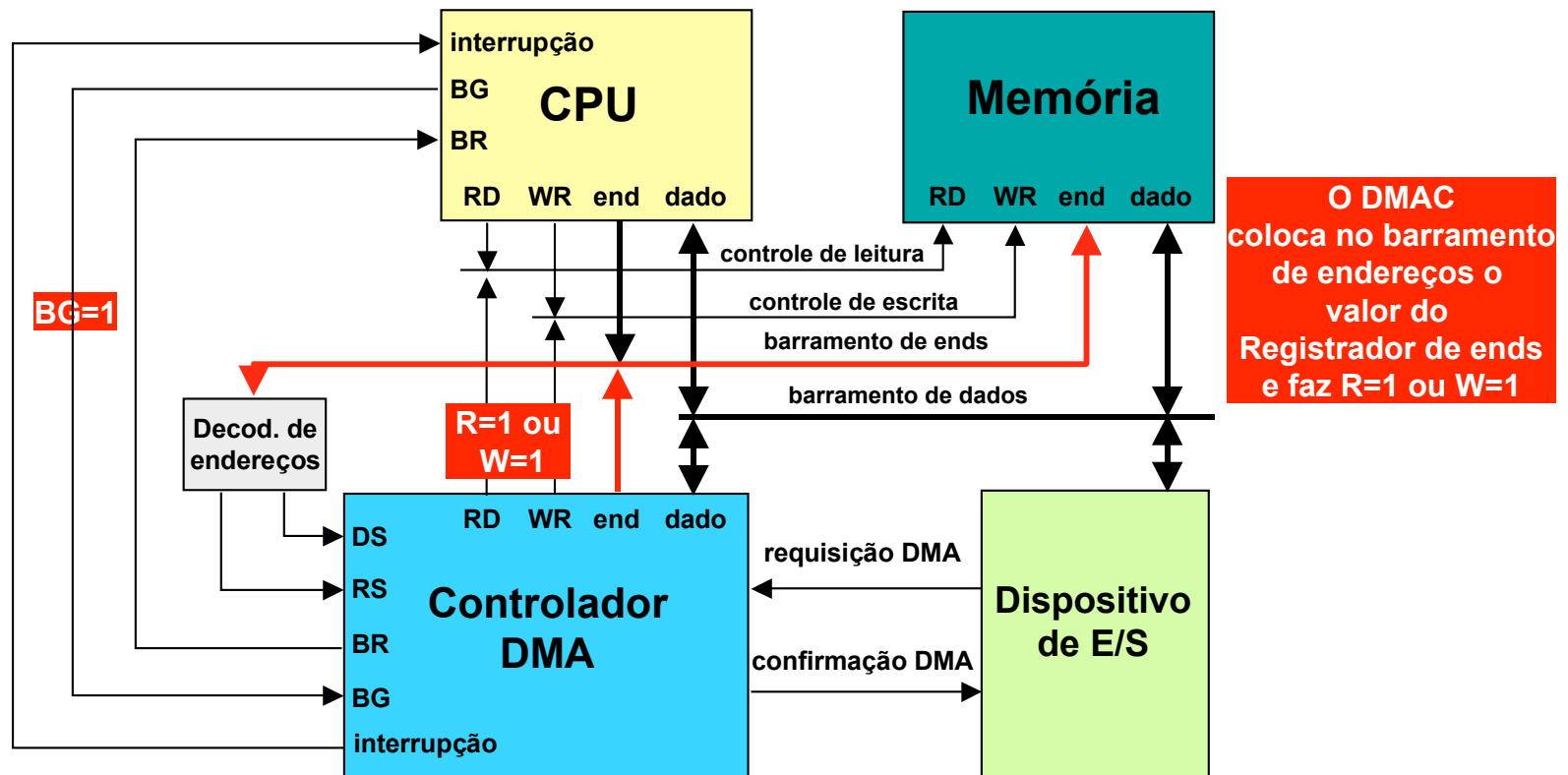
Máquinas Seqüenciais Síncronas

► Transferência DMA



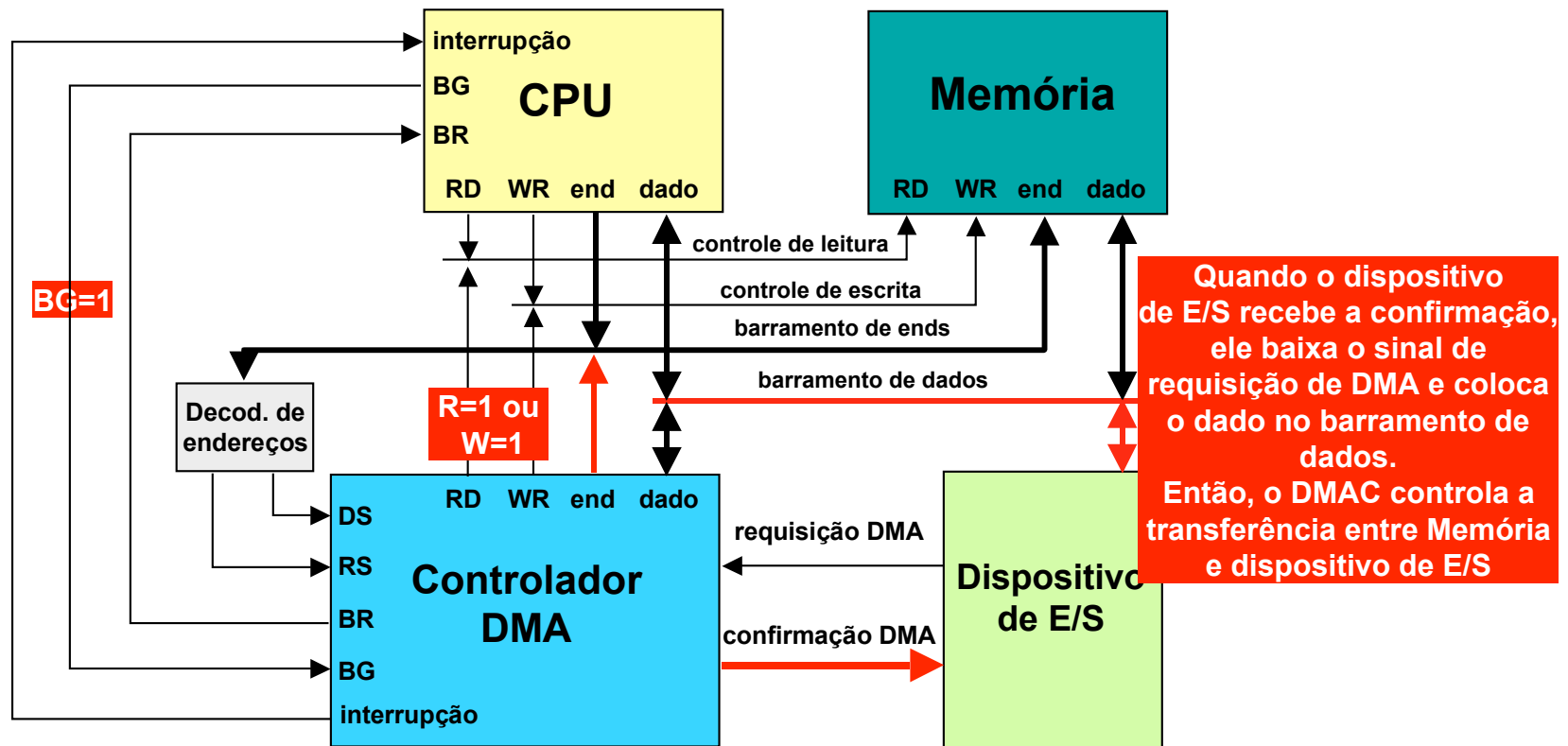
Máquinas Seqüenciais Síncronas

► Transferência DMA



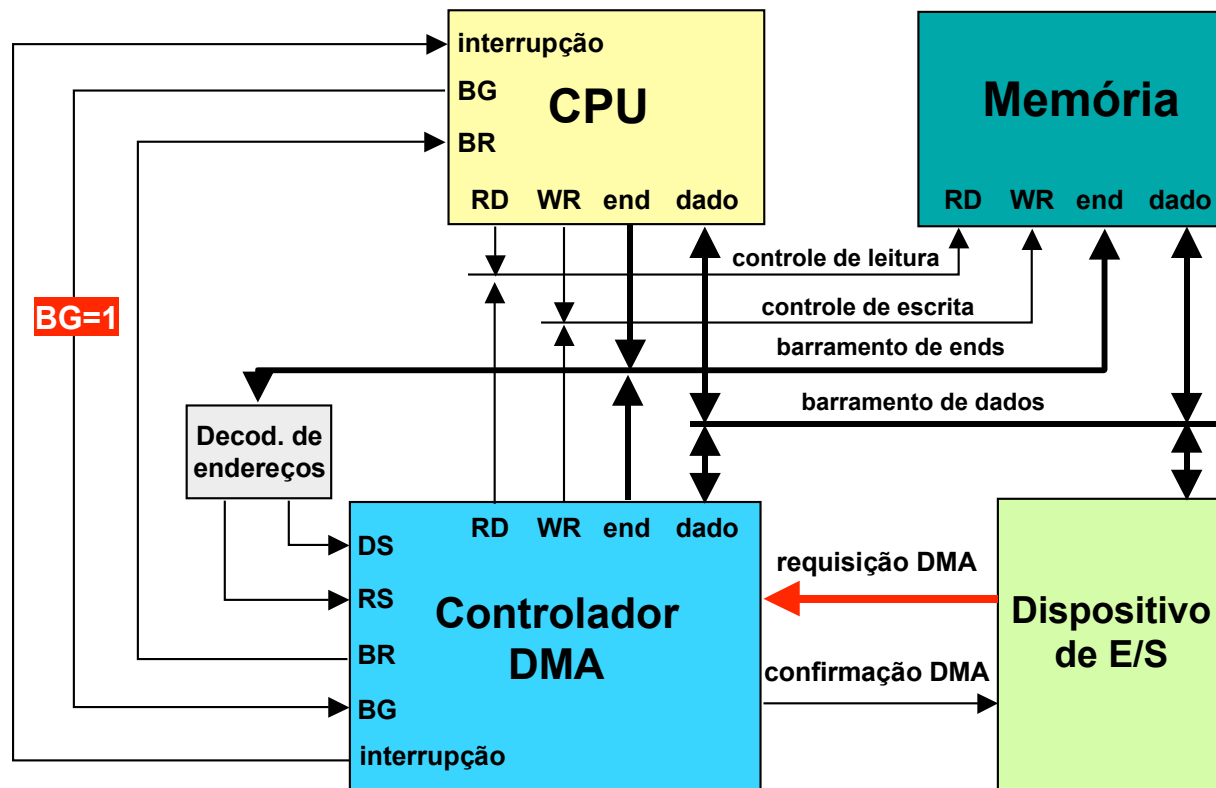
Máquinas Seqüenciais Síncronas

▶ Transferência DMA



Máquinas Seqüenciais Síncronas

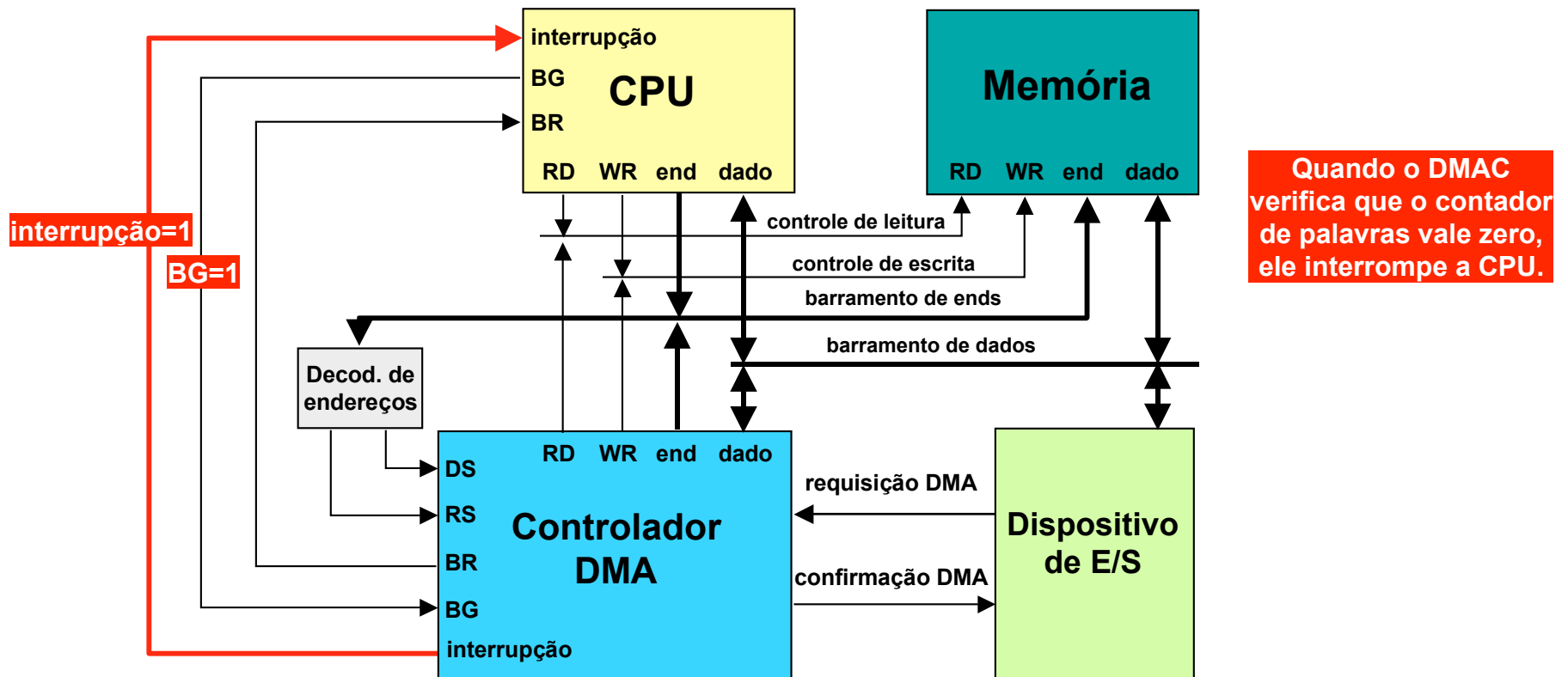
► Transferência DMA



O DMAC testa se o contador de palavras vale zero. Caso negativo, ele fica a espera do dispositivo de E/S sinalizar que está pronto para nova transferência.

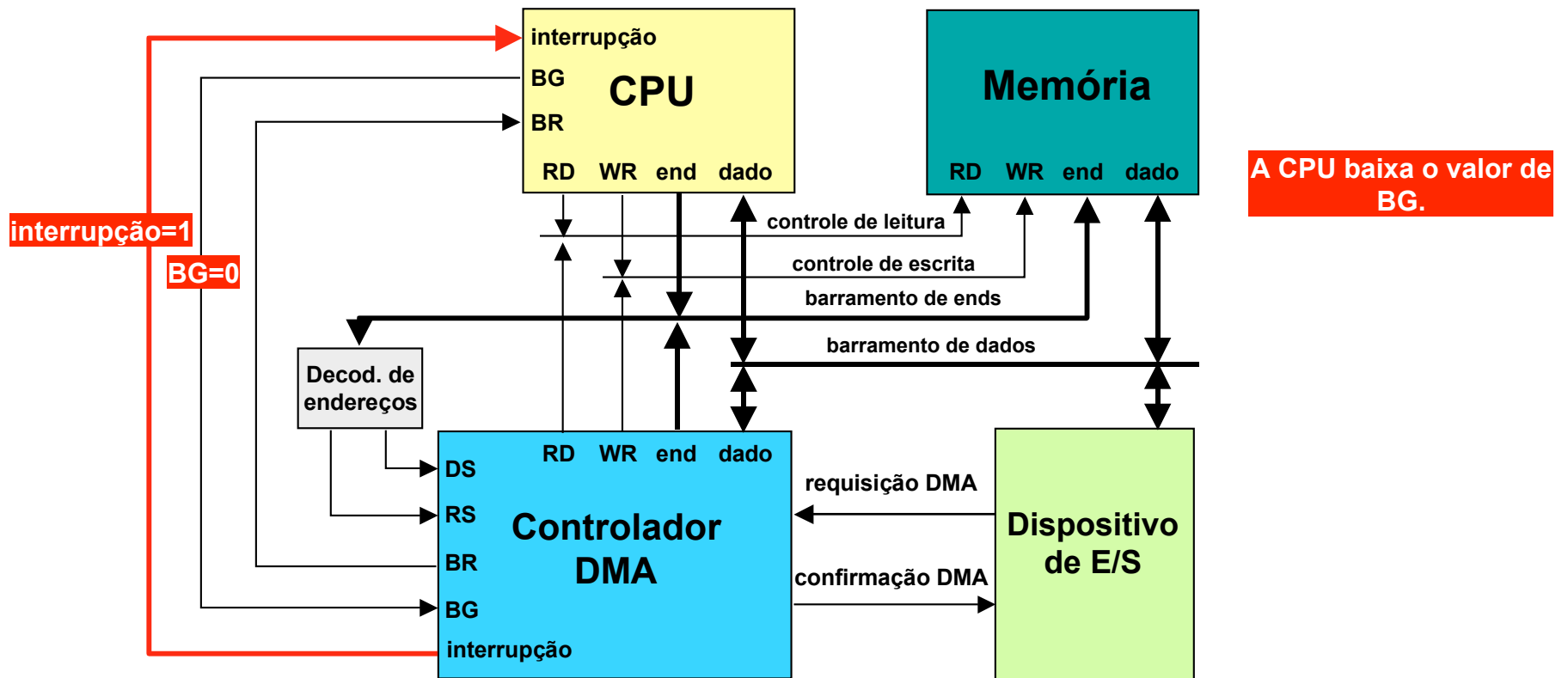
Máquinas Seqüenciais Síncronas

▶ Transferência DMA



Máquinas Seqüenciais Síncronas

▶ Transferência DMA

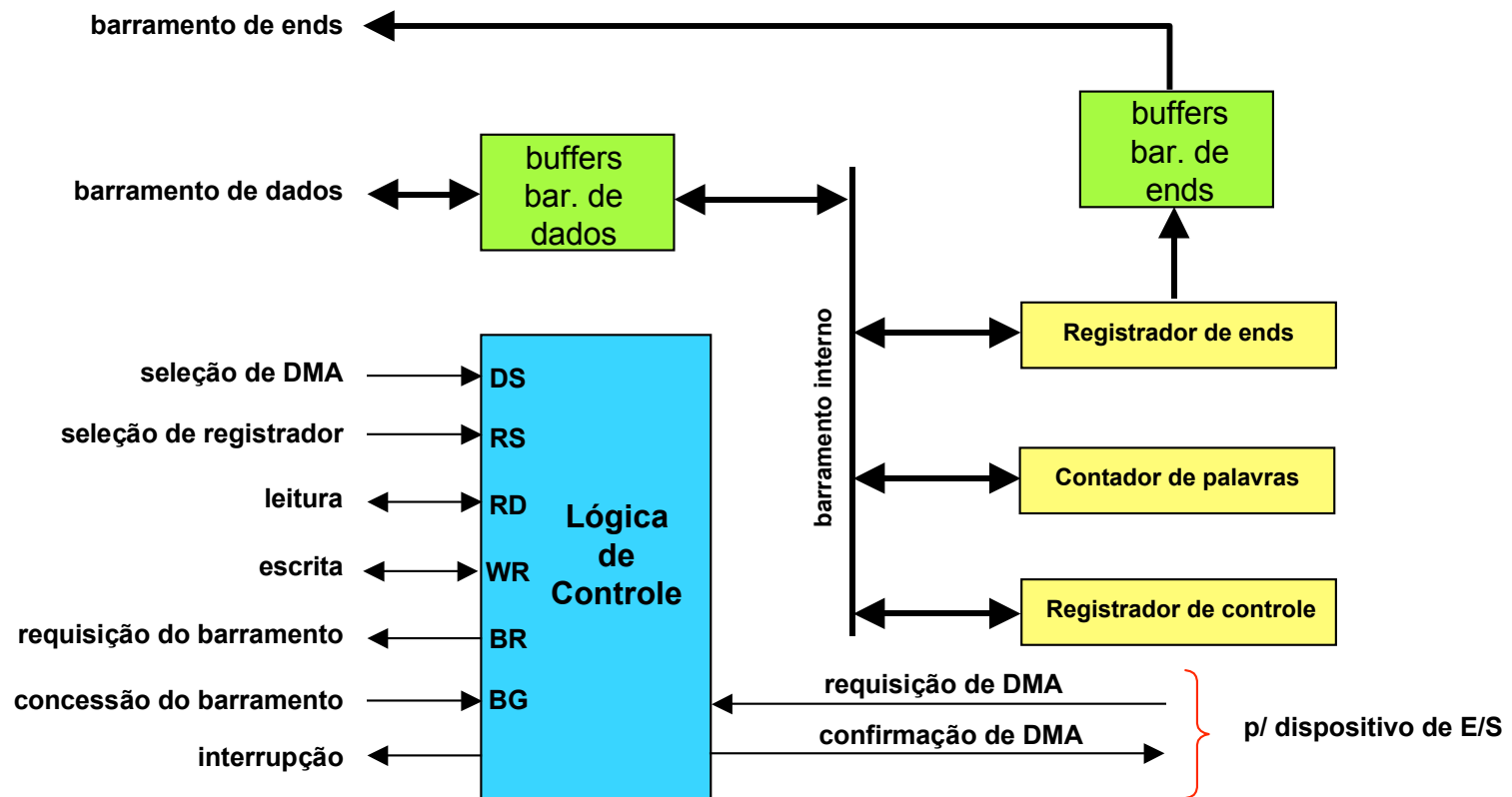


Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA

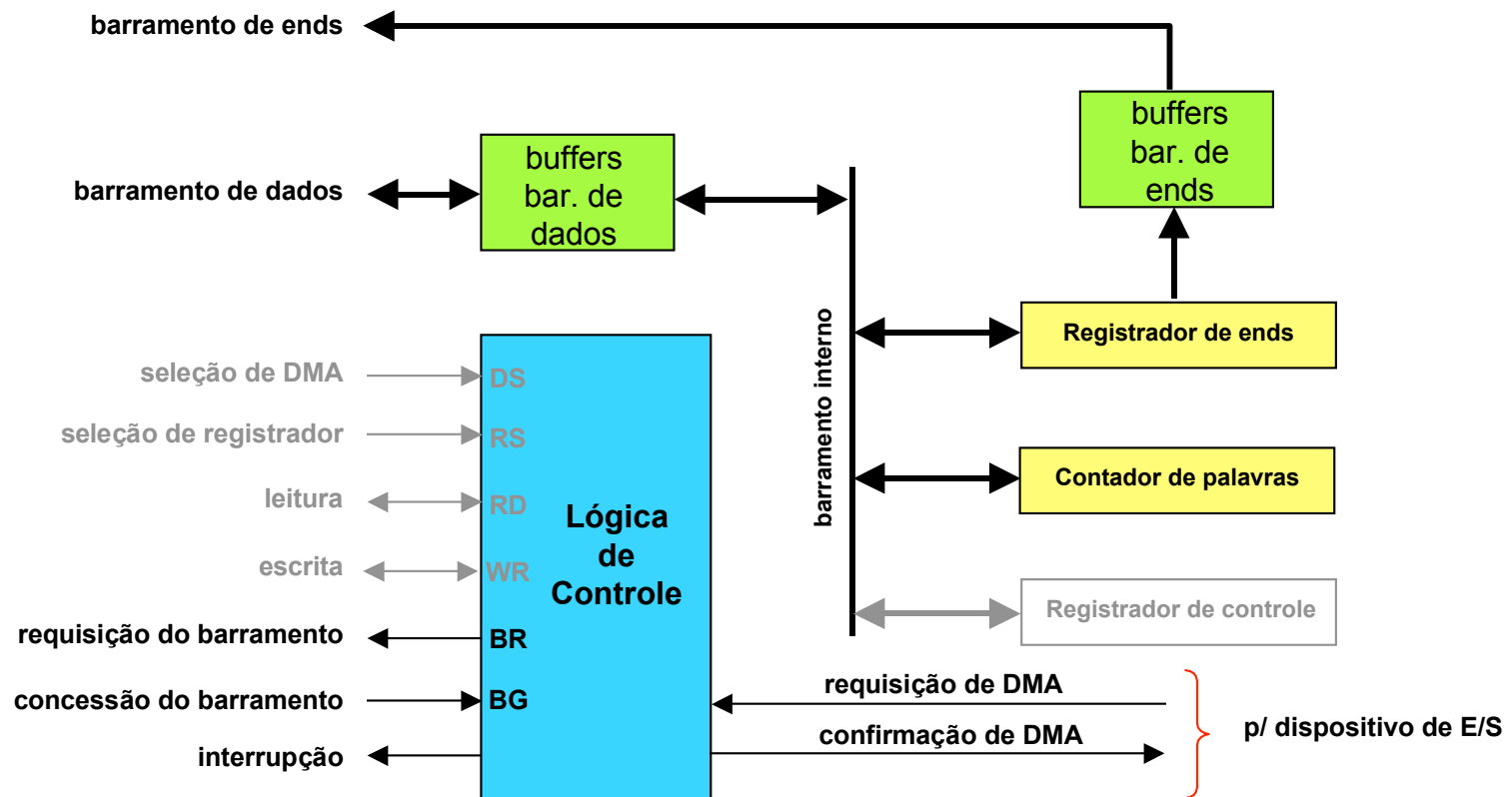
Estrutura Interna



Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA. Projetando o Bloco de Controle



Máquinas Seqüenciais Síncronas

► Síntese de Circuitos Seqüenciais

Exemplo 8: Controlador de DMA. Projetando o Bloco de Controle

