

EEL7020 – Sistemas Digitais
Aula 15:
Mapeamento e alternativas de implementação de máquinas de estado: "hardwired", PLA, ROM e PLD.
 Prof. Djones Vinicius Lettnin
 lettin@eel.ufsc.br
<http://dlettin.paginas.ufsc.br/>

Disclaimer: slides adapted for EEL7020 by D. Lettnin from the original slides made available by the author J. Guentzel.

Exemplo SAD

Projetando um Sistema Digital

Exemplo 3: cálculo da SAD (*Sum of Absolute Differences*)

- A SAD é uma operação realizada sobre duas matrizes de pixels (A e B), gerando um valor único:

$$\text{SAD} = \sum_{i=0}^{i=7} \sum_{j=0}^{j=7} \text{ABS}(\text{pixel}_A(i,j) - \text{pixel}_B(i,j))$$

© J. Guentzel – Adapted by D. Lettnin

Plano de Aula

- Mapeamento e alternativas de implementação de máquinas de estado: "hardwired", PLA, ROM e PLD.**

3

Implementação de FSM

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, cascatabéis (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*, ASIC)

© J. Guentzel – Adapted by D. Lettnin

4

Implementação de FSM

Alternativas de Implementação de FSMs

Lógica de Próximo Estado e Lógica de Saída:

- Tipos de implementações:**
 - Implementando as equações por meio de um circuito combinacional
 - Implementando as equações pela configuração de planos "E" e "OU" (PALs e PLAs)
 - 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*)

5

Implementação de FSM

Alternativas de Implementação de FSMs

Moore

Mealy

Os blocos lógica de próximo estado e lógica de saída podem ser realizados por:

- Memória (PROM, EPROM, EEPROM)
- Arranjos regulares (PLA, PAL)
- Circuitos-padrão (TTLs ...)

© J. Guentzel – Adapted by D. Lettnin

6

Implementação de FSM

Alternativas de Implementação de FSMs
Mealy Usando Memória ROM

© J. Guentzel – Adapted by D. Lettmann

7

Implementação de FSM

Alternativas de Implementação de FSMs
Mealy Usando Memória ROM (outra forma de desenhar...)

© J. Guentzel – Adapted by D. Lettmann

8

Implementação de FSM

Alternativas de Implementação de FSMs
Exemplo 7

Propor uma implementação do circuito do exemplo 3, versão Moore, usando o esquema mostrado no slide anterior (Usar somente uma memória ROM).

Mostrar o conteúdo a ser gravado na memória ROM.

Estado atual y1y0	w	Próximo estado Y1Y0	
A 00	0	00	A
A 00	1	01	B
B 01	0	00	A
B 01	1	10	C
C 10	0	00	A
C 10	1	10	C
- 11	0	XX	-
- 11	1	XX	-

© J. Guentzel – Adapted by D. Lettmann

9

Implementação de FSM

Alternativas de Implementação de FSMs
Exemplo 7

Juntando a tabela de transição com a tabela de saída (pois só será usada uma ROM).

Estado atual y1y0	w	Próximo estado Y1Y0	z
A 00	0	00	A 0
A 00	1	01	B 0
B 01	0	00	A 0
B 01	1	10	C 0
C 10	0	00	A 1
C 10	1	10	C 1
- 11	0	XX	- -
- 11	1	XX	- -

© J. Guentzel – Adapted by D. Lettmann

10

Implementação de FSM

Alternativas de Implementação de FSMs
Exemplo 7

Arrumando a nova tabela.

Estado atual y1y0	w	Próximo estado Y1Y0	z
00	0	00	0
00	1	01	0
01	0	00	0
01	1	10	0
10	0	00	1
10	1	10	1
11	0	XX	-
11	1	XX	-

© J. Guentzel – Adapted by D. Lettmann

11

Implementação de FSM

Alternativas de Implementação de FSMs
Exemplo 7

Diagrama de blocos final.

Endereço da ROM	000	00	0
001	01	0	
010	00	0	
011	10	0	
100	00	1	
101	10	1	
110	XX	X	
111	XX	X	

© J. Guentzel – Adapted by D. Lettmann

12

Implementação de FSM

Alternativas de Implementação de FSMs

Exemplo 8

Estado atual y1y0	E1	E0	Próximo estado Y1Y0	saidas
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	

Propor uma implementação do circuito cujo comportamento está na tabela ao lado, usando o esquema mostrado no slide anterior. Mostrar o conteúdo a ser gravado na memória ROM.

© J. Guentzel – Adapted by D. Lettmin

Implementação de FSM

Alternativas de Implementação de FSMs

Para Cada Linha da Tabela de Estados uma Linha da ROM

Estado atual y1y0	E1	E0	Próximo estado Y1Y0	saidas
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	

ROM

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

© J. Guentzel – Adapted by D. Lettmin

Implementação de FSM

Alternativas de Implementação de FSMs

A Solução é Agrupar Casos Equivalentes na Tabela de Transição

Estado atual y1y0	E1	E0	Próximo estado Y1Y0	saidas
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	

Agrupando-se os don't cares

Estado atual y1y0	E1	E0	Próximo estado Y1Y0	saidas
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 10 linhas da ROM

Necessita de 6 linhas da ROM

© J. Guentzel – Adapted by D. Lettmin

Implementação de FSM

Alternativas de Implementação de FSMs

Restrição do Agrupamento de Casos Equivalentes

No caso de FSM de Moore:

- Não há restrição para o agrupamento pois os valores das saídas dependem apenas do estado atual!

No caso de FSM de Mealy:

- Só podem ser agrupados os casos em que as saídas têm os mesmos valores!!
Vide exemplo ao lado...

Estado y1y0	E1	E0	Próximo estado Y1Y0	saidas
00	0	X	00	saida1
00	1	X	01	saida2
01	X	0	01	saida3
01	X	1	10	saida4
10	X	X	11	saida5
11	X	X	00	saida6

© J. Guentzel – Adapted by D. Lettmin

Implementação de FSM

Alternativas de Implementação de FSMs

Montagem dos Bits a Serem Usados para Endereçar a ROM

Estado y1y0	E1	E0	Próximo estado Y1Y0	saidas
00	0	X	00	saida1
00	1	X	01	saida2
01	X	0	01	saida3
01	X	1	10	saida4
10	X	X	11	saida5
11	X	X	00	saida6

© J. Guentzel – Adapted by D. Lettmin

Implementação de FSM

Alternativas de Implementação de FSMs

Montagem dos Bits a Serem Usados para Endereçar a ROM

Estado y1y0	E1	E0	Próximo estado Y1Y0	saidas
00	0	0	00	saida1
00	0	1	01	saida2
00	1	0	01	saida3
01	X	0	01	saida4
01	X	1	10	saida4
10	X	X	11	saida5
10	X	X	11	saida5
11	0	0	00	saida6
11	0	0	00	saida6

© J. Guentzel – Adapted by D. Lettmin

Implementação de FSM

Alternativas de Implementação de FSMs
Caso Geral...

© J. Guentzel – Adapted by D. Lettmann

19

Implementação de FSM

Alternativas de Implementação de FSMs
Explorando Características da FSM

© J. Guentzel – Adapted by D. Lettmann

20

Implementação de FSM

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

© J. Guentzel – Adapted by D. Lettmann

21

Implementação de FSM

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

© J. Guentzel – Adapted by D. Lettmann

22

Implementação de FSM

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

© J. Guentzel – Adapted by D. Lettmann

23

Implementação de FSM

Alternativas de Implementação de FSMs
PLA em Tecnologia CMOS

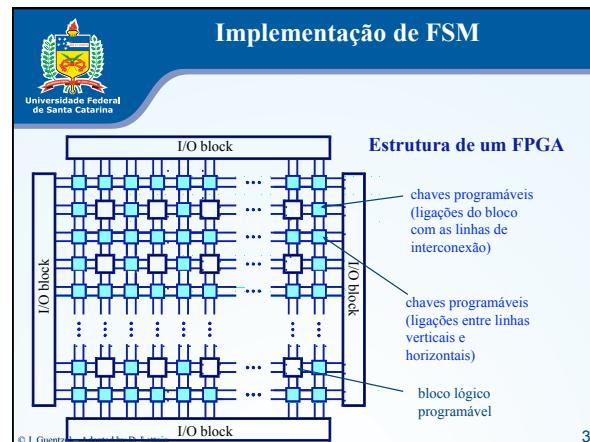
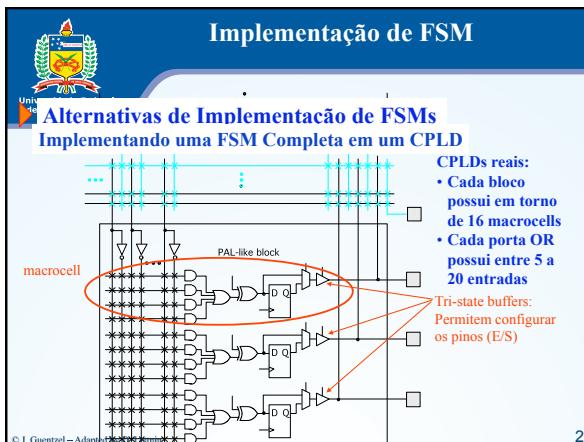
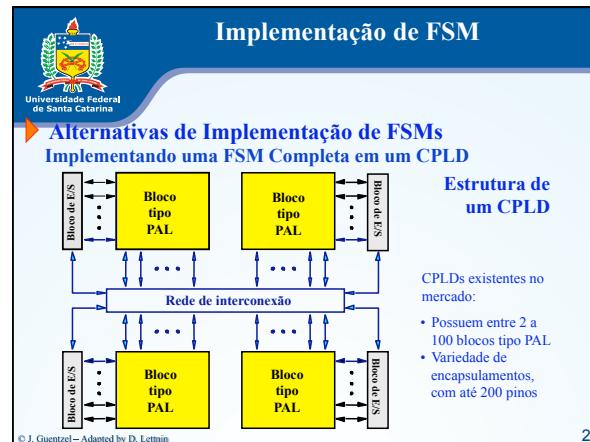
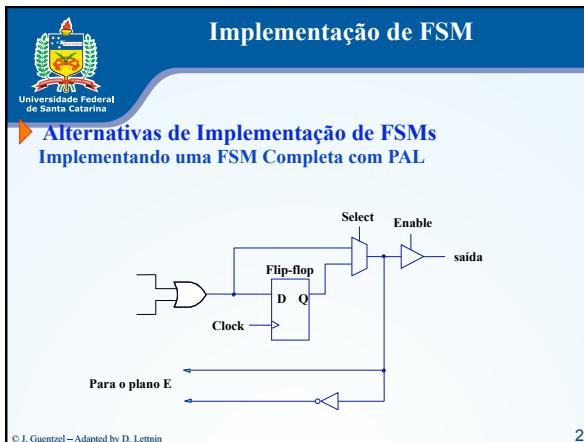
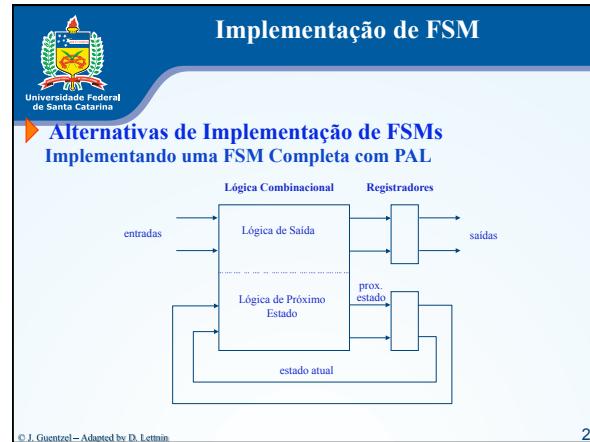
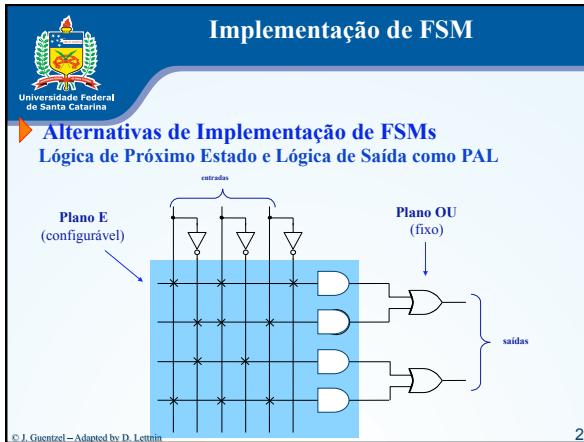
Soma de Produtos (SdP)

“NOR de NORs”

Negando as entradas e aplicando De Morgan...

© J. Guentzel – Adapted by D. Lettmann

24



Implementação de FSM

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!!!

© J. Guentzel – Adapted by D. Lettmann

Implementação de FSM

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

© J. Guentzel – Adapted by D. Lettmann

Implementação de FSM

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

© J. Guentzel – Adapted by D. Lettmann

Implementação de FSM

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 vail-um
- 64 bits para memória
- 64 bits para shift-register

Transcrição de F. Moreira (PUCRS)

© J. Guentzel – Adapted by D. Lettmann

Implementação de FSM

Arquitetura (metade) do Slice

© J. Guentzel – Adapted by D. Lettmann

Implementação de FSM

FPGAs Altera: Stratix II

Estrutura Básica da Matriz do Stratix II

© J. Guentzel – Adapted by D. Lettmann

Implementação de FSM

FPGAs Altera: Stratix II

Estrutura de um LAB (Logic Array Block)

Cada LAB é constituído por:

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

© J. Guentzel – Adapted by D. Lettnin

Implementação de FSM

FPGAs Altera: Stratix II

Diagrama de blocos de um ALM

© J. Guentzel – Adapted by D. Lettnin

Implementação de FSM

FPGAs Altera: Stratix II

Detalhes de um ALM

© J. Guentzel – Adapted by D. Lettnin

EEL7020 – Sistemas Digitais

Aula 15:

Mapeamento e alternativas de implementação de máquinas de estado: "hardwired", PLA, ROM e PLD.

Prof. Djones Vinicius Lettnin
lettnin@eel.ufsc.br
<http://dlettnin.paginas.ufsc.br/>

Disclaimer: slides adapted for EEL7020 by D. Lettnin from the original slides made available by the author J. Guentzel.