



FEDERAL UNIVERSITY
OF SANTA CATARINA

EEL5105 – Circuitos e Técnicas Digitais

Aula 11

Prof. Héctor Pettenghi

hector@eel.ufsc.br

<http://hectorpettenghi.paginas.ufsc.br>

Material desenvolvido com apoio de arquivos de apresentação do livro de Frank Vahid

Memorias e aplicação à FSMs

- Memórias RAM e ROM
- FSM Controlado por ROM.
- FSM Controlado por ROM com endereçamento explícito.

Nesta aula veremos sobre memórias a como elas podem ser aplicadas à FSMs.

Memórias

- Memórias
 - Frequentemente, é necessário armazenar um conjunto muito grande de palavras em simultâneo

Soluções:

- Banco de Registos → limitado a poucas dezenas de palavras...
- Circuitos de memória:
 - RAM
 - ROM
- Outros tipos de memória:
 - Discos magnéticos, CD, DVD, Blu-Ray, etc..
 - Normalmente ligados a outros dispositivos (ex: PCs)

Frequentemente é necessário armazenar um conjunto muito grande de palavras (informações) em simultâneo. Para permitir isso foram desenvolvidos tipos diferentes de memórias, como bancos de registros, circuitos de memória, discos magnéticos, CD, DVD e entre tantos outros.

Nesta aula veremos com mais cuidado as RAMs e as ROMs.

Tipos de Memória

- Tipos de memórias

- O nome **RAM** vem de “*Random Access Memory*”: o tempo de acesso à informação na RAM é sempre igual, independentemente da posição (endereço) “aleatória” que se pretende.
- Antes do aparecimento deste tipo de dispositivos, existiam apenas memórias com acesso série (ex: fitas magnéticas ou outras semelhantes a registos de deslocamento), em que o tempo de acesso à informação dependia da distância a que ela estava do início da fita ou da saída série do circuito de deslocamento.



O nome RAM vem de "Random Access Memory", isso porque o seu tempo de acesso à informação é sempre igual, independentemente da posição acessada. Antes do aparecimento deste tipo de dispositivos, existiam apenas memórias de acesso em série, nas quais o tempo de acesso dependia da posição da informação na fita.

Tipos de Memória

- Tipos de memórias
 - **RAM** (*Random Access Memory*) – é possível **ler** e **escrever** dados, na sequência normal de funcionamento de um dado sistema digital.

Exemplos:

- **Estáticas (SRAM)**
- **Dinâmicas (DRAM)**



Nas RAMs, é possível tanto ler como escrever dados. Alguns exemplos são a SRAM (Static RAM) e a DRAM (Dynamic RAM).

Tipos de Memória

- Tipos de memórias
 - **ROM** (*Read-Only Memory*) – podem ser programadas uma ou relativamente poucas vezes e, no funcionamento normal do sistema, são **apenas lidas**.

Exemplos:

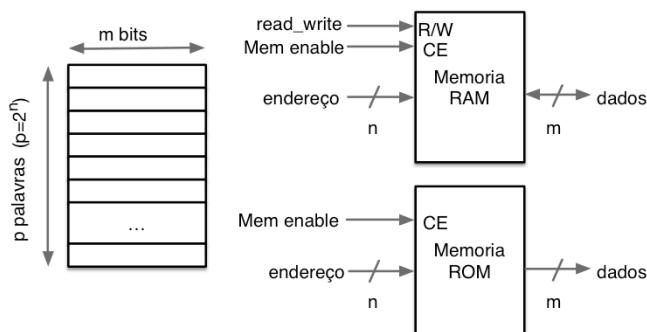
- ROM
- PROM
- EPROM
- EEPROM



Já as ROMs só podem ser lidas no funcionamento normal do sistema. Alguns exemplos de ROMs são a própria ROM, a PROM (Programmable ROM), a EPROM (Erasable Programmable ROM) e a EEPROM (Electrically-Erasable Programmable ROM).

Círculo de Memória

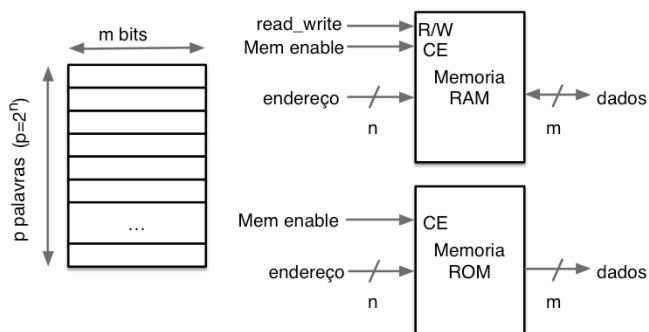
- Círculo de Memória
 - Círculo capaz de armazenar um conjunto p de palavras, cada uma com m bits, através do **barramento de dados**;
 - A palavra pretendida é indicada por um endereço, colocado no **barramento de endereços**. Habitualmente, p e m são potências inteiras de 2.



Um círculo de memória é um círculo capaz de armazenar um conjunto p de palavras, cada uma com m bits, através do barramento de dados. Cada palavra é indicada por um endereço, colocado no barramento de endereços. Habitualmente, p e m são potências inteiras de 2.

Círculo de Memória

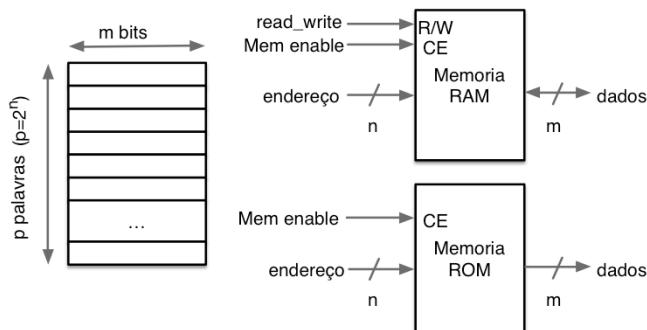
- Círculo de Memória
 - A indicação das operações na **memória RAM** de leitura ou escrita é dada:
 - Por duas linhas independentes (ex: **READ** e **WRITE**)
 - Por uma única linha (ex: **READ/WRITE** ou **R/W**)



A indicação das operações na memória RAM de leitura ou escrita é dada ou por duas linhas independentes ou por uma única linha.

Círculo de Memória

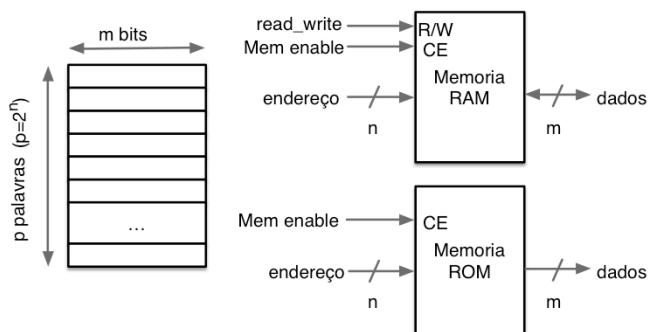
- Círculo de Memória
 - A ativação ou desativação do dispositivo de memória pode ser feita através do sinal de enable: **CS** (Chip Select), **CE** (Chip Enable), ou **Mem Enable**.
 - Quando inativo, este sinal coloca o **barramento de dados** em alta impedância.



Já ativação ou desativação do dispositivo de memória pode ser feita através do sinal de enable. Quando inativo, este sinal coloca o barramento de dados em alta impedância.

Círculo de Memória

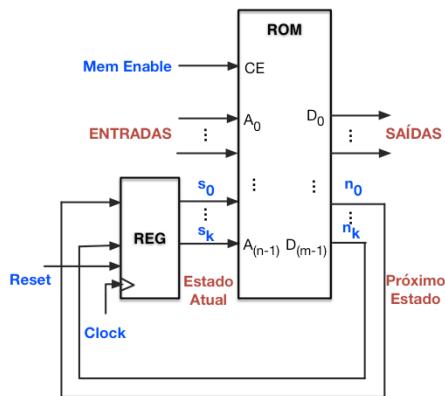
- Círculo de Memória
 - Em geral, nas memórias **RAM** o **barramento de dados** é bidireccional, i.e., é utilizado para escrever (input) e ler (output) informação;
 - Nas memórias **ROM** o **barramento de dados** é unidireccional, i.e., é utilizado apenas para ler (output) informação.



Nas memórias RAM o barramento de dados é bidireccional, ou seja, é utilizado para leitura e escrita. Já nas memória ROM ele é unidireccional, apenas utilizado para leitura.

FSM Controlado por ROM

- Estrutura básica de controle por ROM

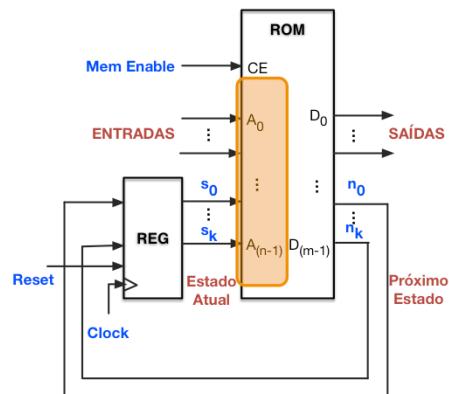


- A memória **ROM** substitui a lógica combinatória para gerar:
 - Estado seguinte
 - Saída do circuito

É possível fazer o controle de uma FSM a partir de uma memória ROM, substituindo a lógica combinatória para gerar o estado seguinte e a saída do circuito. Neste slide temos a estrutura básica do bloco de controle com a memória ROM.

FSM Controlado por ROM

- Estrutura básica de controle por ROM

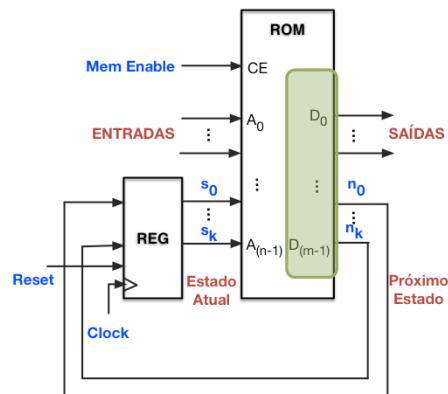


- Entradas da **ROM** (**barramento de endereços**):
 - Entradas externas da máquina de estados
 - Estado atual

Nas entradas da ROM, ou seja, no barramento de endereços, temos as entradas externas da máquina de estados e o estado atual.

FSM Controlado por ROM

- Estrutura básica de controle por ROM

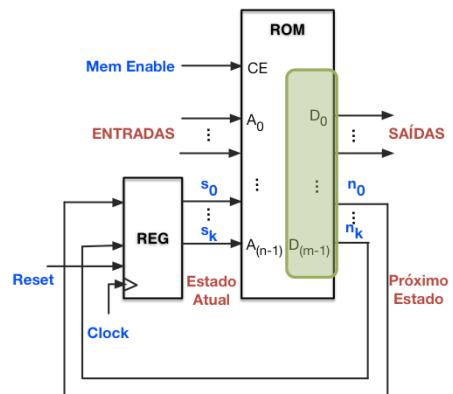


- Saídas da ROM (**barramento de dados**):
 - Saídas para o exterior da máquina de estados
 - Saídas (comandos) internas + **estado seguinte**

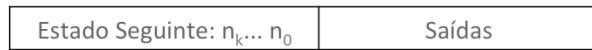
Já no barramento de dados (saídas da ROM) temos as saídas para o exterior da máquina de estados, ou seja, os comandos a serem enviados ao datapath e o estado seguinte.

FSM Controlado por ROM

- Estrutura básica de controle por ROM



- O conteúdo de cada endereço na ROM é constituído por dois campos:



Assim, o conteúdo de cada endereço da ROM é constituído por dois campos, um com o estado seguinte e outro com as saídas.

FSM Controlado por ROM

- **Exemplo**

- 4 estados:
 - E0 ($s_1s_0=00$)
 - E1 ($s_1s_0=01$)
 - E2 ($s_1s_0=10$)
 - E3 ($s_1s_0=11$)
- 2 entradas: A e B
- 1 saída: F

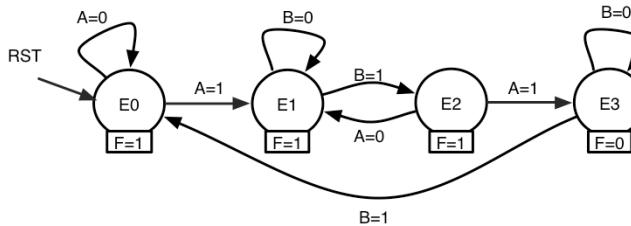


Tabela transição estados

s_1	s_0	A	B	n_1	n_0	F
0	0	0	-	0	0	1
0	0	1	-	0	1	1
0	1	-	0	0	1	1
0	1	-	1	1	0	1
1	0	0	-	0	1	1
1	0	1	-	1	1	1
1	1	-	0	1	1	0
1	1	-	1	0	0	0

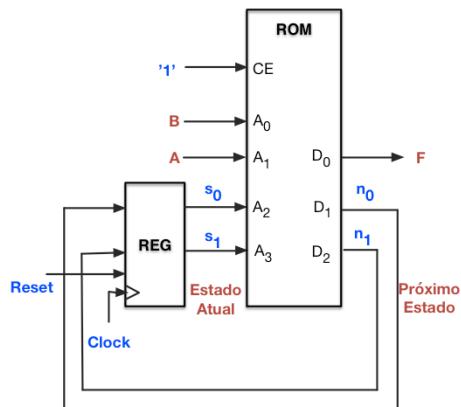
15

Consideremos este exemplo.

FSM Controlado por ROM

- **Exemplo**

- Diagrama de blocos de um controlador implementado com ROM:



- Formato de cada palavra da ROM $\rightarrow \{n_1n_0F\}$

16

O diagrama de blocos do controlador implementado com ROM para este exemplo é o seguinte. Veja que o formato de cada palavra da ROM é $\{n_1n_0F\}$, seguindo a mesma lógica das entradas.

FSM Controlado por ROM

- Exemplo

Tabela de transição de estados

s_1	s_0	A	B	n_1	n_0	F
0	0	0	-	0	0	1
0	0	1	-	0	1	1
0	1	-	0	0	1	1
0	1	-	1	1	0	1
1	0	0	-	0	1	1
1	0	1	-	1	1	1
1	1	-	0	1	1	0
1	1	-	1	0	0	0

Conteúdo da ROM

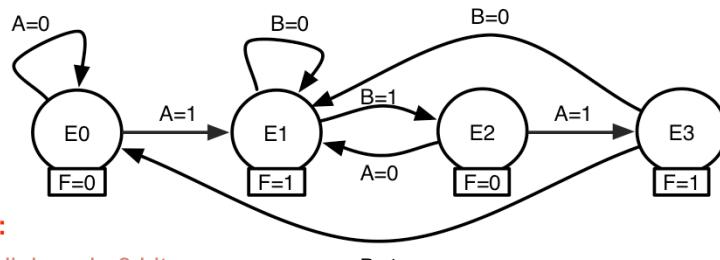
Endereço	Dados
$A_3A_2A_1A_0$	$D_2D_1D_0$
0000	001
0001	001
0010	011
0011	011
0100	011
0101	101
0110	011
0111	101
1000	011
1001	011
1010	111
1011	111
1100	110
1101	000
1110	110
1111	000

Assim, este será o conteúdo da ROM.

PROBLEMAS

Problema 11.1. Projete a FSM usando memoria ROM para gerar os próximos estados e saída F. Obtenha as dimensões da memoria ROM a partir do diagrama de estados usando:

- a) Codificação Binária.
- b) Codificação *One-Hot*.

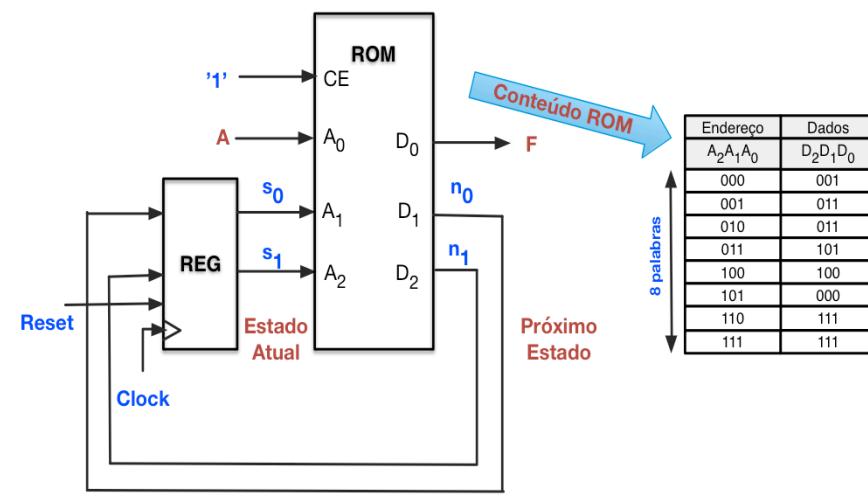


Solução:

- a) 16 linhas de 3 bits.
- b) 64 linhas de 5 bits.

PROBLEMAS

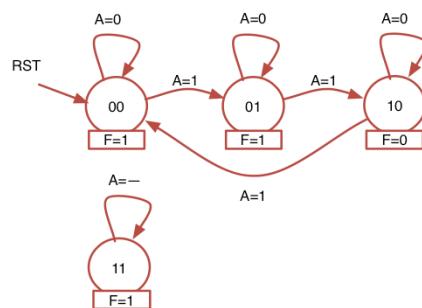
Problema 11.2. Obtenha o diagrama de estados da FSM a partir do conteúdo da memoria ROM da Figura.



PROBLEMAS

Problema 11.2. Obtenha o diagrama de estados da FSM a partir do conteúdo da memoria ROM da Figura.

Solução:

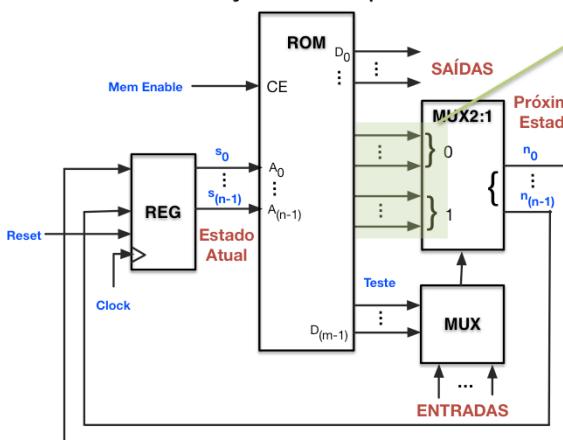


Controle por ROM com Endereçamento Explícito

- Controle por ROM com Endereçamento Explícito

Próximos estados explícitos

n_1 e n_0



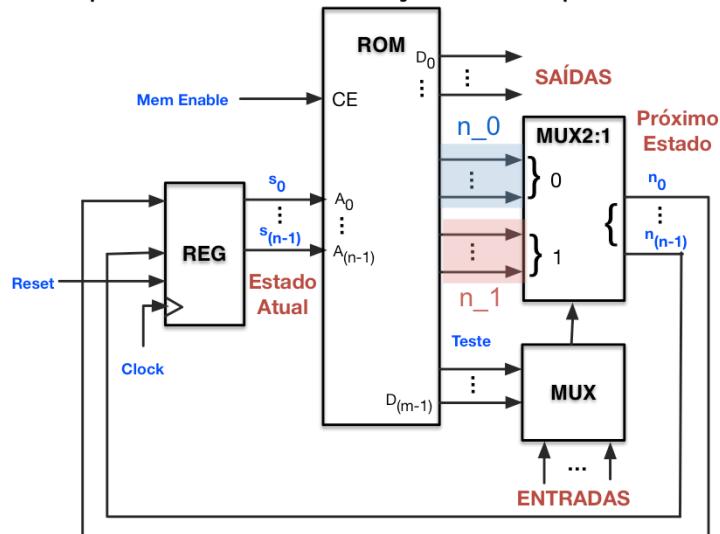
- Para reduzir o número de endereços, as entradas externas são retiradas do barramento de endereços;
- Cada estado atual só pode evoluir para um de dois estados.

Podemos ainda fazer um bloco de controle com endereçamento explícito para reduzir o número de endereços. Nesta técnica as entradas externas são retiradas do barramento de endereços.

No nosso exemplo, cada estado atual só pode evoluir para um de dois estados, no máximo.

Controle por ROM com Endereçamento Explícito

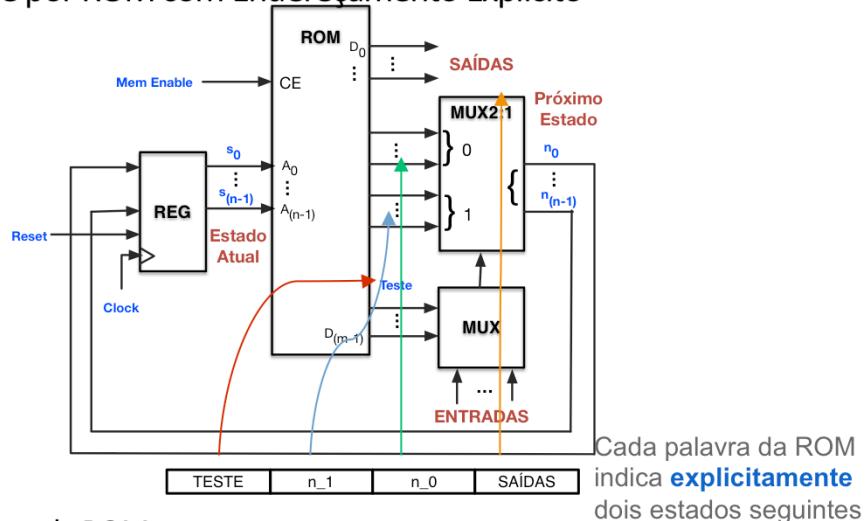
- Controle por ROM com Endereçamento Explícito



Note n_0 e n_1 , cada um com dois bits para cada estado possível.

Controle por ROM com Endereçamento Explícito

- Controle por ROM com Endereçamento Explícito

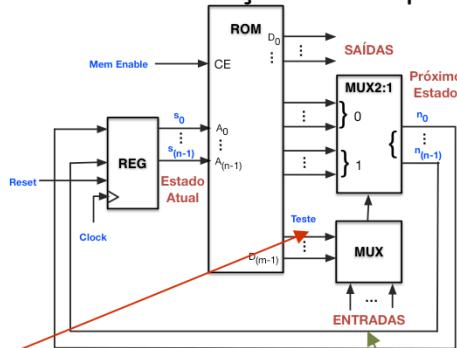


- Palavra da ROM:

Assim, as palavras da ROM terão o formato apresentado neste slide. Note que elas indicam explicitamente dois estados seguintes.

Controle por ROM com Endereçamento Explícito

- Controle por ROM com Endereçamento Explícito

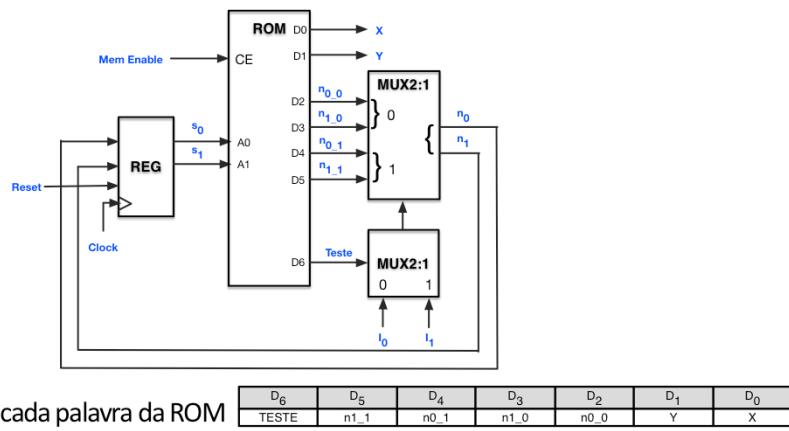


- O MUX1 tem as suas entradas ligadas às **entradas primárias** do circuito de controlo. O campo de **teste** da ROM permite, para cada estado atual, escolher a entrada ou combinações de entradas a testar.
 - Se a entrada selecionada tiver o valor 0, o estado seguinte escolhido é o que vier indicado no campo n_0 ;
 - No caso contrário, será o estado seguinte n_1 .

Um dos MUX tem as suas entradas ligadas às entradas primárias do circuito de controle. O campo "teste" da ROM permite, para cada estado atual, escolher a entrada ou combinações de entradas (portas lógicas) que terão influência na definição do próximo estado do circuito. No caso apresentado, se a entrada selecionada tiver o valor 0, o estado seguinte escolhido é o que vier indicado no campo n_0 , caso contrário, será escolhido n_1 .

Exemplo

- Exemplo (simples)
 - Diagrama de blocos de um controlador implementado com ROM com **endereçamento explícito**:



Assim, o formato de cada palavra da ROM será o indicado neste slide.

Exemplo

- Exemplo (simples)

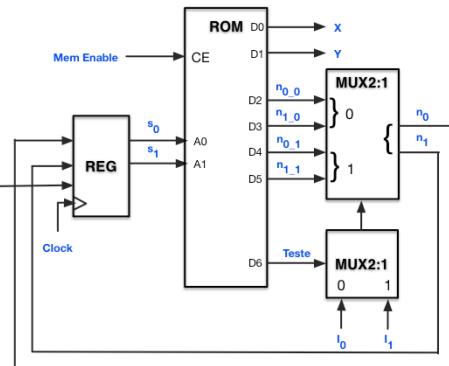
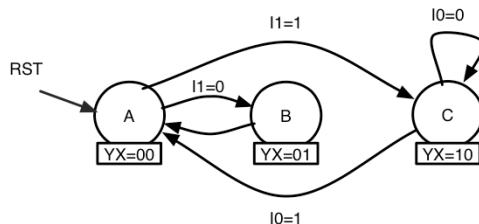


Tabela de transição de estados

s_1	s_0	Teste	n_{1_1}	n_{0_1}	n_{1_0}	n_{0_0}	Y	X
0	0	1	1	0	0	1	0	0
0	1	-	0	0	0	0	0	1
1	0	0	0	0	1	0	1	0
1	1	-	-	-	-	-	-	-

Neste slide temos um exemplo de como seriam possíveis palavras neste controle.

Exemplo

- Exemplo (simples)



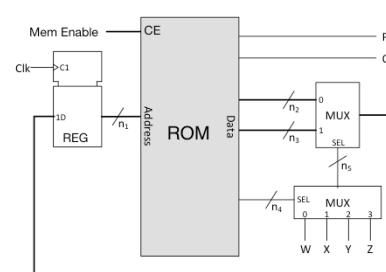
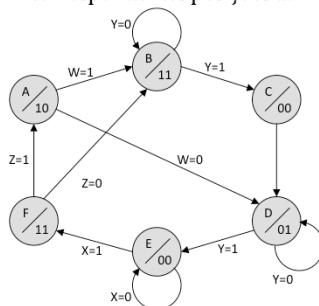
- Circuito controlador original com ROM:
 - 16 endereços de 4 bits (64 bits)
- Circuito controlador com endereçamento explícito da ROM:
 - 4 endereços de 7 bits (28 bits)

Veja a diferença: no circuito controlador original com ROM teríamos 16 endereços de 4 bits (um total de 64 bits a serem armazenados na memória), mas no novo são necessários apenas 4 endereços de 7 bits cada (um total de somente 28 bits).

PROBLEMAS

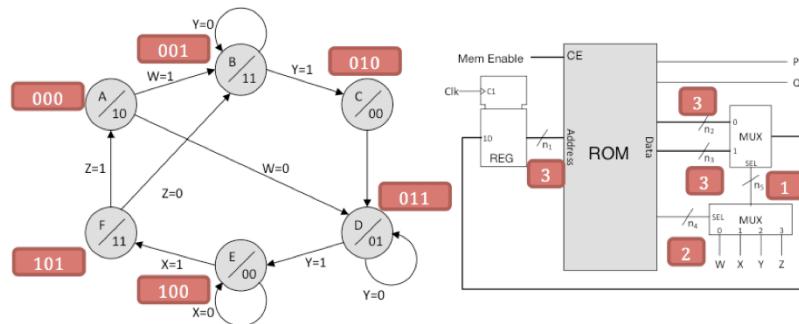
Problema 11.3. Considere o seguinte diagrama de estados de um circuito sequencial síncrono, caracterizado por 4 entradas (W,X,Y,Z) e 2 saídas (Q,P). Pretende-se implementar este circuito através de uma FSM micro-programada constituída por uma ROM e um registo como mostrado na Figura.

- Obtenha uma codificação possível para os diferentes estados do circuito.
- Identifique a largura (nº bits) dos sinais representados no diagrama: n_1 a n_5 .
- Determine o conteúdo da fração da ROM que permite implementar todas as transições do diagrama de estados que saem do estado A (indique o endereço e o valor das correspondentes posições da memória).



PROBLEMAS

Solução:

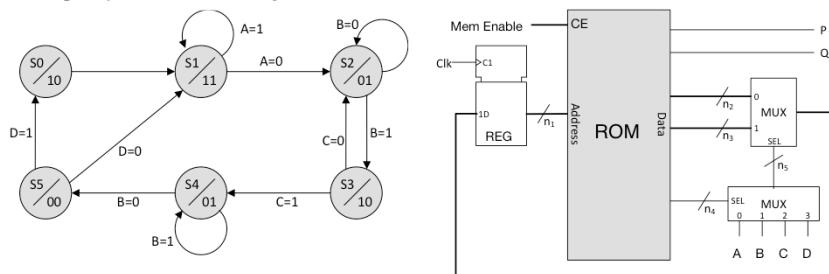


Endereço	Dados
n1	n4 n3 n2 Q P
A ₂ A ₁ A ₀	D ₉ D ₈ D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀
000	0000101110

PROBLEMAS

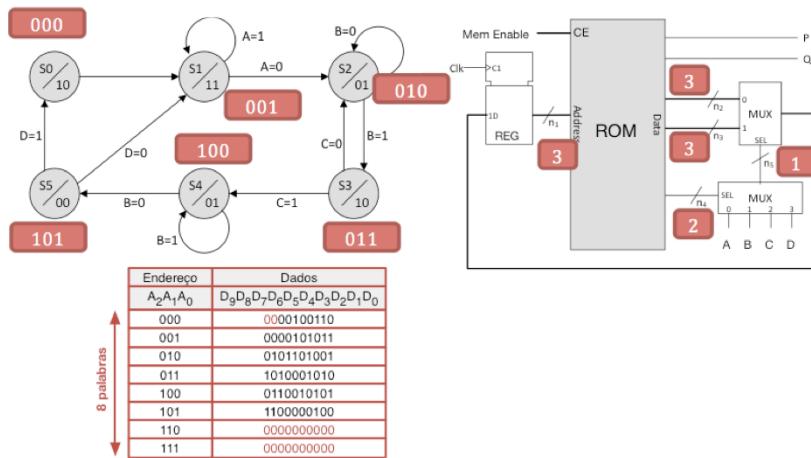
Problema 11.4. Considere o seguinte diagrama de estados de um circuito sequencial síncrono, caracterizado por 4 entradas (A,B,C,D) e 2 saídas (Q,P). Pretende-se implementar este circuito através de uma FSM micro-programada constituída por uma ROM e um registo como mostrado na Figura.

- Obtenha uma codificação possível para os diferentes estados do circuito.
- Identifique a largura (n^o bits) dos sinais representados no diagrama: n_1 a n_5 .
- Determine o conteúdo da ROM (indique o endereço e o valor das correspondentes posições da memória).



PROBLEMAS

Solução:





FEDERAL UNIVERSITY
OF SANTA CATARINA

EEL5105 – Circuitos e Técnicas Digitais

Aula 11

Prof. Héctor Pettenghi

hector@eel.ufsc.br

<http://hectorpettenghi.paginas.ufsc.br>

Material desenvolvido com apoio de arquivos de apresentação do livro de Frank Vahid