Cronologia

- Ler o endereço [i] da memória;
- Armazenar o valor do endereço [i] numa variável [Data];
- Ler do [8:0] Counter uma combinação X de 6 elementos que nos dará o próximo endereço (esse endereço não pode ultrapassar o valor
- Ler o valor [j] do endereço obtido do [8 : 0] Counter
- Colocar o valor de [j] em [i]
- de [j] - Acrescentar ao endereço de [i] + 1 e

- Colocar o valor de [x] no endereço

repetir os passos anteriores até o ultimo endereço de memória

Atribuições Default

Saídas

Prefixos das variáveis

Variável de Controle: pode ser um fio (wire) ou um registrador (reg), mantém o valor que estava no estado anterior. Pode receber usado como entrada pelos estados.

vc valores nos estados. Pode ser Variável de Saída (Output): pode ser um fio (wire) ou um registrador (reg). Pode receber valores nos vo

> como entrada nos estados. Entradas (Input), pode ser ativada a qualquer momento. Saída (Output), é um fio (wire) de

estados. Não pode ser usada

saída da máquina de estados (binário). Pode receber valores nos estados e não depende de clock.

Glossário

Entradas

Saída de dados da memória i_MemData (4 bits)

Saída da máquina de estados indicando se o Shuffler deve embaralhar ou não a i_ActShuffler memória (1 bit)

Próximo endereço da memória, obtido i_Addr_J através de um módulo a parte. (6 bits)

> Saídas Indica se o baralho está embaralhado ou

> > (1 bit)

Indica se a memória deve ser escrita ou não. Em 1 a escrita é habilitada e em 0

Registrador que armazena **a carta**, localizada no endereço **vc_Addr_I** da

Registrador que armazena a carta,

localizada no endereço **i_Addr_J** da

memória.

não para a máquina de estados global

o_Shuffled

o_Write

vc_Data_I

vc_Data_J

vo_Addr_I

Saída de dados para escrita na memória o_Data (4 bits)

Indica o endereço que deve ser acessado o_Address na memória (6 bits)

Clock da memória o_MemClk (1 bit)

> apenas leitura. (1 bit) Variáveis

> > memória, a ser trocada com a próxima posição i_Addr_J (4 bits)

(4 bits) Registrador que armazena **o endereço de memória** da carta a ser trocada

$o_Shuffled = 0$

o_Data = XXXX = XXXXXX o_Address

o_MemClk = 0 o_Write Variáveis de saída

Variáveis de controle

 $vo_Addr_I = XXXXXX$

 $vc_Data_I = XXXX$ vc_Data_J = XXXX X = Don't care IF_State i_ActShuffler = 1 vo_Addr_I >= 51 i_ActShuffler = 1 vo_Addr_I < 51 i_ActShuffler = 0 i_ActShuffler = 1 Shuffled Increase_Address o_Shuffled = 1 vo_Addr_I = vo_Addr_I + 1 $i_ActShuffler = 0$ i_ActShuffler = 1 i_ActShuffler = 0 Initialize_Output o_Address = vo_Addr_I i_ActShuffler = 0 i_ActShuffler = 1 Read_Mem_Output_I o_Mem_Clk = 1 o_Adress = vo_Addr_I i_ActShuffler = 0 i_ActShuffler = 1 $Store_Mem_Output_I$ o_Adress = vo_Addr_I vc_Data_I = i_MemData i_ActShuffler = 0 i_ActShuffler = 1 Get_Next_Address $o_Adress = i_Addr_J$ i_ActShuffler = 0 i_ActShuffler = 1 Read_Mem_Output_J o_Mem_Clk = 1 **o_Adress** = i_Addr_J

i_ActShuffler = 0

i_ActShuffler = 0

i_ActShuffler = 1

o_Data

i_ActShuffler = 1

o_Data

o_Mem_Clk = 1 o_Write

Write_Mem_Addr_J

 $o_Adress = i_Addr_J$

i_ActShuffler = 1 i_ActShuffler = 0

= vc_Data_J

i_ActShuffler = 0

Begin

 $vo_Addr_I = 000000$

i_ActShuffler = 0

i_ActShuffler = 1

Change_Address

o_Adress = vo_Addr_I

o_Data

 $o_Mem_Clk = 0$ o_Write

i_ActShuffler = 1

 $Write_Mem_Addr_I$

o_Adress = vc_Addr_I

= vc_Data_J

i_ActShuffler = 0

i_ActShuffler = 1

o_Data

o_Mem_Clk = 1 o_Write

= vc_Data_I

 $o_MemClk = 0$ o_Write

o_Adress = i_Addr_J $vc_Data_J = i_MemData$

 $Store_Mem_Output_J$

= vc_Data_I