Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

USP - ICMC - SSC
SCE 0703 (PISE) - 2o. Semestre 2008

Disciplina de Projeto e Implementação de Sistemas Embarcados I

Prof. Fernando Santos Osório
Email: fosorio [at] { icmc. usp. br , gmail. com }
Web: http://www.icmc.usp.br/~fosorio/

Aula 05

Aula 05 – FSM: Máquina de Estados Finitos

Temas Abordados:

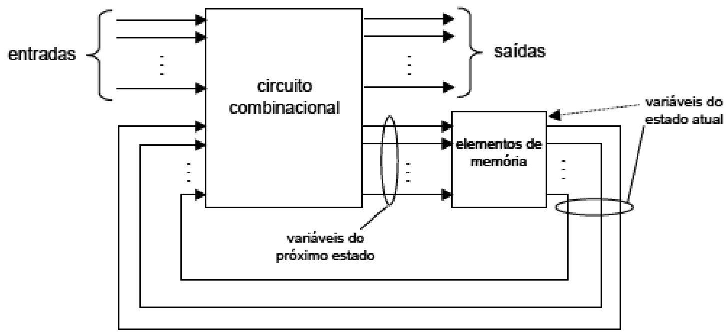
1. **Introdução: FSM - *Finite State Machines***
   > Conceitos
   Circuitos Seqüenciais
   Máquina de Estados: Assíncrona, Síncrona
   > Máquina de Estados
   Máquina de Moore
   Máquina de Mealy
   > Exemplos e Projetos
2. **FSM em VHDL**
   > Exemplos
3. **FSM no Quartus II**
   > FSM Wizard
   > FSM Editor

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

1. Circuitos Seqüenciais

Diagrama de Blocos de um Circuito Seqüencial



3
Agosto 2008

[Guntzer, J. L. & Nascimento, F.A. - http://minerva.ufpel.edu.br/~guntzel/isd/isd.html

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

1. Circuitos Seqüenciais

Circuito Seqüencial: <u>Síncrono</u> / Assíncrono

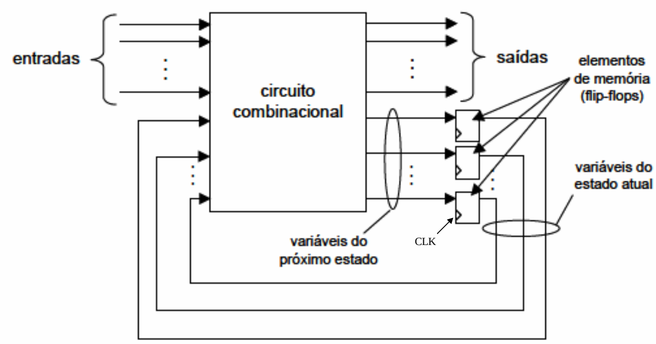


Diagrama de blocos de um circuito seqüencial síncrono.

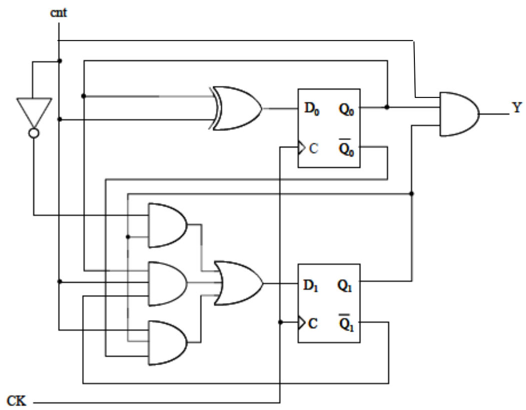
4
Agosto 2008

[Guntzer, J. L. & Nascimento, F.A. - http://minerva.ufpel.edu.br/~guntzel/isd/isd.html

FSM - Finite State Machine

1. Circuitos Seqüenciais

Circuito Seqüencial: Exemplo



Exemplo de circuito seqüencial.

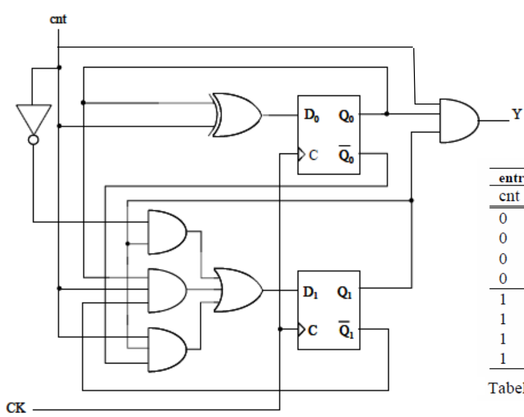
[Guntzer, J. L. & Nascimento, F.A.]

FSM - Finite State Machine

1. Circuitos Seqüenciais

Circuito Seqüencial: Exemplo

Entrada: cnt
Saída : Y
Síncrono: CK (clock)



entrada	estado atual		próximo estado	
cnt	$Q1_t$	$Q0_t$	$Q1_{t+1}$	$Q0_{t+1}$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Tabela de transição de estados para o circuito

Exemplo de circuito seqüencial.

[Guntzer, J. L. & Nascimento, F.A.]

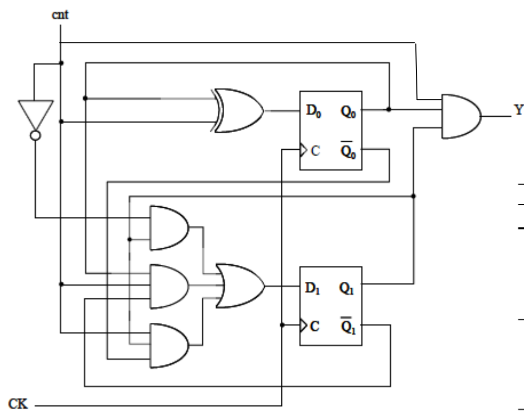
FSM - Finite State Machine

1. Circuitos Seqüenciais

Circuito Seqüencial: Exemplo

Entrada: cnt
Saída : Y
Síncrono: CK (clock)

Contador Síncrono de Módulo-4
(conta de 0 a 3, no 3 gera saida y=1)



entrada	estado atual		saída
cnt	$Q1_t$	$Q0_t$	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

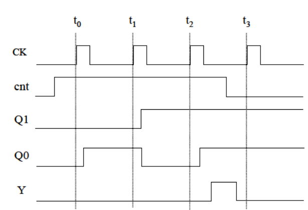
Exemplo de circuito seqüencial.

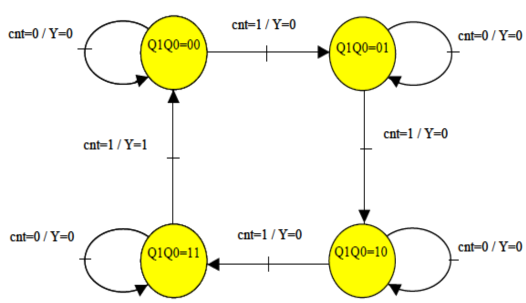
[Guntzer, J. L. & Nascimento, F.A.]

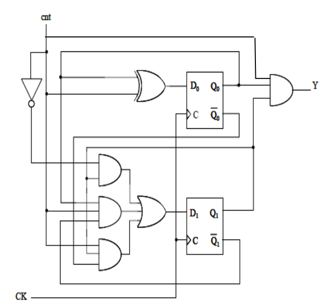
FSM - Finite State Machine

1. Circuitos Seqüenciais

Circuito Seqüencial: Exemplo







Exemplo de circuito seqüencial.

[Guntzer, J. L. & Nascimento, F.A.]

FSM - Finite State Machine

1. Circuitos Seqüenciais

Circuito Seqüencial: Síncrono

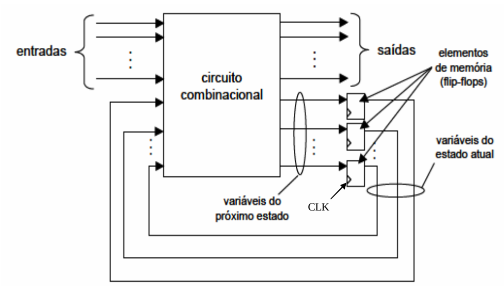
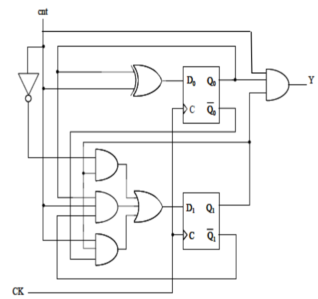
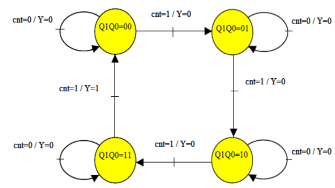


Diagrama de blocos de um circuito seqüencial síncrono.



Exemplo de circuito seqüencial.



[Guntzer, J. L. & Nascimento, F.A.]

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

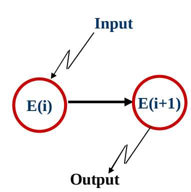
2. Máquina de Estados Finitos

Conceito:

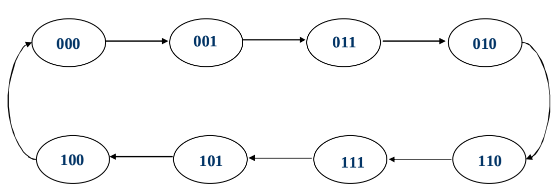
- Estado: Contexto / Estado(i) - Estado atual

- Transição: Entrada + Estado(i) => Estado(i+1)

- Seqüência de Estados / Diagrama de Estados



Exemplo: Diagrama de Estados - Contador "gray code" de 3 bits



Exemplos: Semáforo de Trânsito, Elevador, ...

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

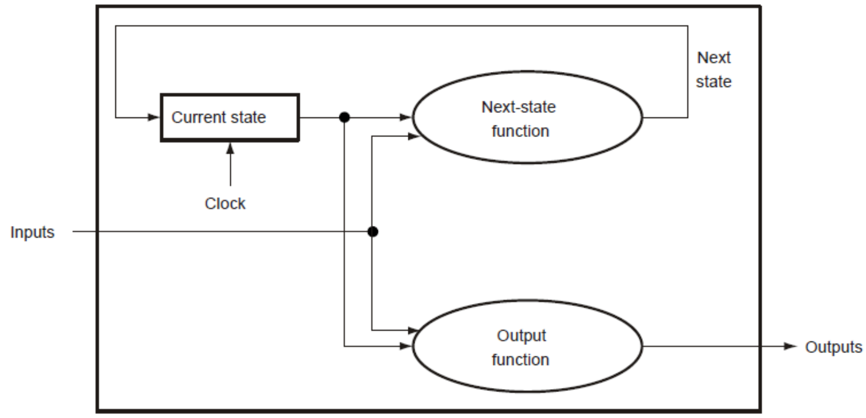
FSM - Finite State Machine

2. Máquina de Estados Finitos

Próximo Estado = F(Estado Atual, Entradas)
ou seja, **E(i+1)** = F(x, y) = F(E(i), Inputs)

Saídas = G(x) ou **Saídas** = G(x,y)
ou seja, **Output** = G(Estado Atual [,Inputs])

Diagrama Esquemático - FSM:



USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Máquina de Moore

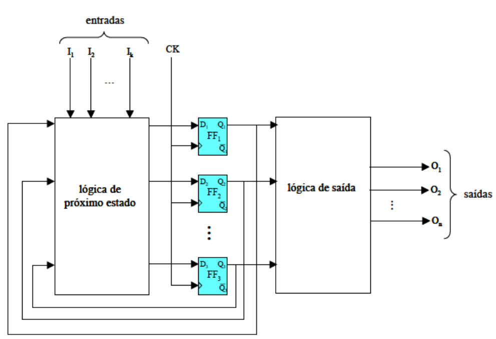


Diagrama de blocos para o modelo de Moore.

Máquina de Mealy

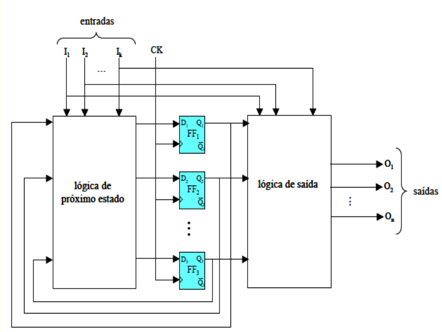


Diagrama de blocos para o modelo de Mealy.

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: *Traffic Lights*

>> Especificação / Dados gerais:

- * Semáforo de um cruzamento, apenas com duas luzes: verde e vermelho.
- * Relógio de 0,033 Hz (1 ciclo cada 30 s)
- * Duas estradas: norte/sul (NS) e este/oeste (EW)

>> Sinais de saída:

- * NSlite: ativo ! luz NS verde (senão vermelha)
- * EWlite: ativo ! luz EW verde (senão vermelha)

>> Sinais de entrada:

- * NScar: ativo ! carro presente na estrada NS
- * EWcar: ativa ! carro presente na estrada EW

>> O semáforo deve mudar apenas quando existe um carro presente na outra estrada; caso contrário a via onde passou o último carro deve permanecer a verde

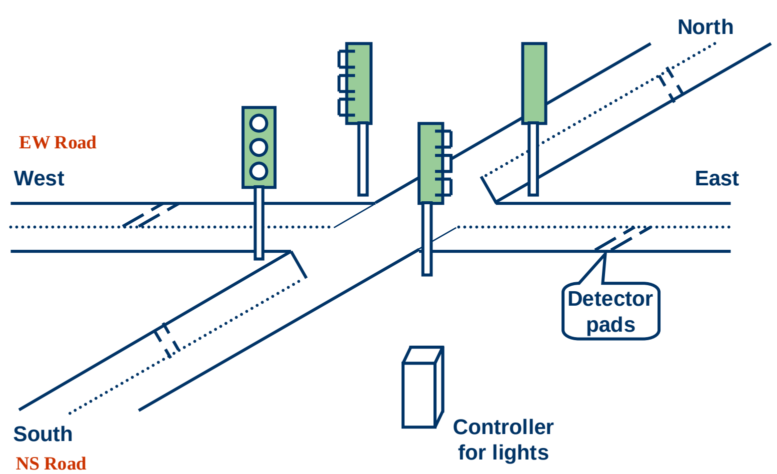
13
Agosto 2008

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: *Traffic Lights*

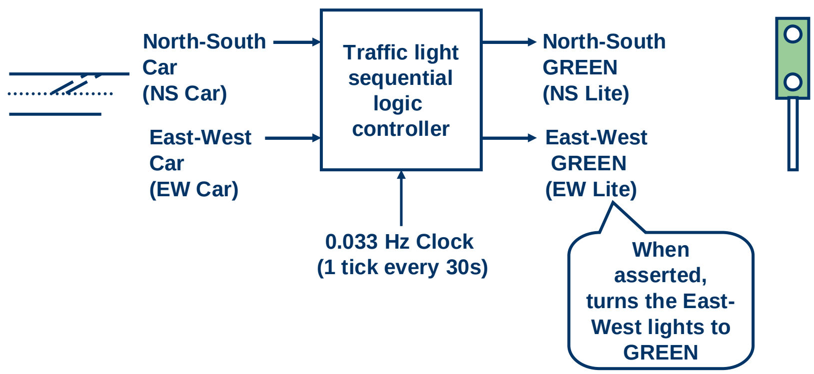


14
Agosto 2008

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: *Traffic Lights*



USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: *Traffic Lights*

E. actual	NScar	EWcar	E. seguinte
NSgreen	0	0	NSgreen
NSgreen	0	1	EWgreen
NSgreen	1	0	NSgreen
NSgreen	1	1	EWgreen
EWgreen	0	0	EWgreen
EWgreen	0	1	EWgreen
EWgreen	1	0	NSgreen
EWgreen	1	1	NSgreen

Dois estados:

- *NSgreen*: luz verde na direcção NS.
- *EWgreen*: luz verde na direcção EW.

E. actual	NSlite	EWLite
NSGreen	1	0
EWGreen	0	1

☞ Codificação de estados (1 variável de estado):

- NSgreen: 0
- EWgreen: 1

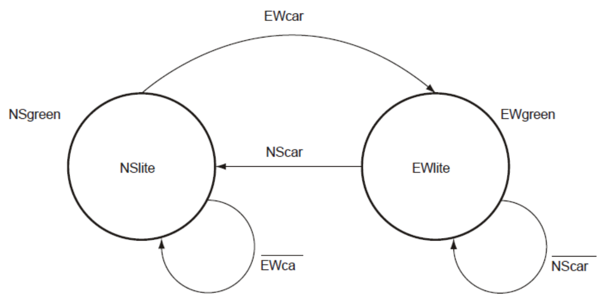
Current state	North-South GREEN	East-West GREEN
0	ON	OFF
1	OFF	ON

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: ***Traffic Lights***



**** Funciona bem?***

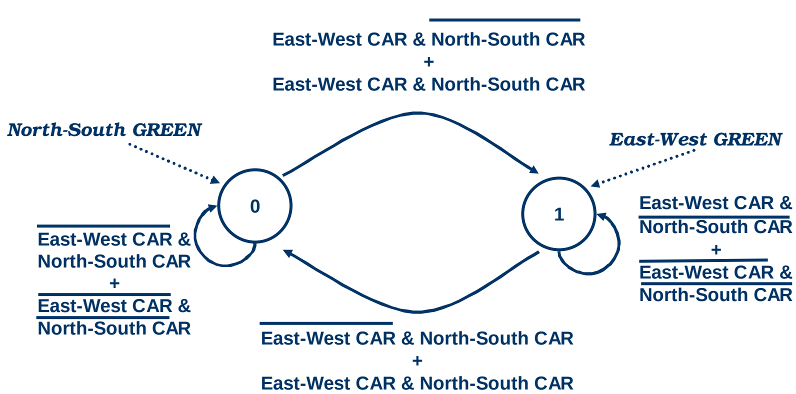
17
Agosto 2008

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: ***Traffic Lights***



**** Funciona bem?***

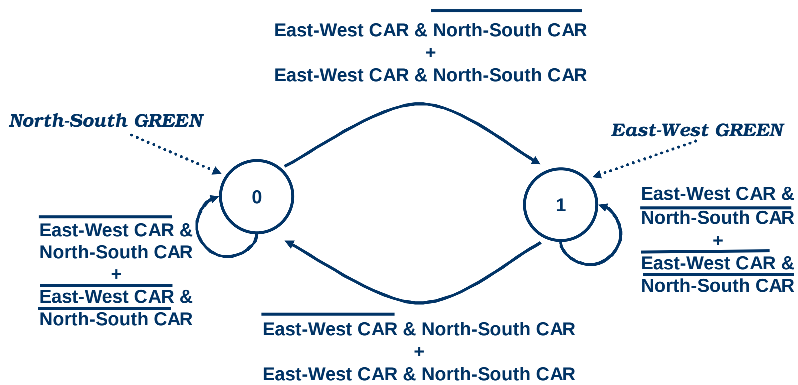
18
Agosto 2008

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

2. Máquina de Estados Finitos: FSM

Exemplo FSM: ***Traffic Lights***



** Funciona bem? Extensões... Prioridade Highway, Pedestre,*
Controle Remoto, Dependência de Horários, etc

19
Agosto 2008

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

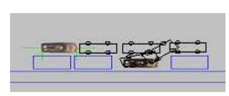
FSM - Finite State Machine

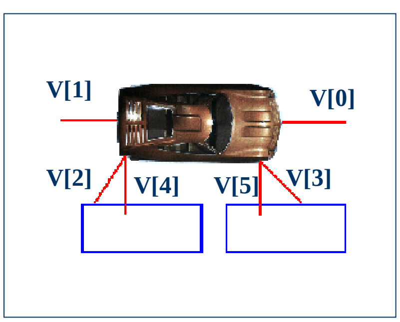
2. Máquina de Estados Finitos: FSM

Exemplo FSM:

Sistema de Estacionamento de Veículos Autônomo (SEVA)

** Exemplo mais complexo...*





Entradas:

-Sensores de distância V[0] a V[5]

-Sensores posicionados de forma estratégica, especificamente para estacionamento em vagas paralelas.

Saídas:

- Atuadores: Direção / Acelerador (steering wheel / speed +/-)

[Refs.: Ver site http://www.icmc.usp.br/~fosorio/ (Publicações)]
[Procure por > Osorio Heinen SEVA < no Google]

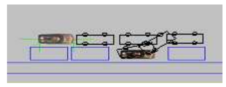
20
Agosto 2008

FSM - Finite State Machine

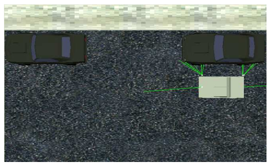
2. Máquina de Estados Finitos: FSM

Exemplo FSM: ***Sistema de Estacionamento de Veículos Autônomo***

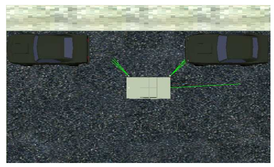
**** Exemplo mais complexo...***



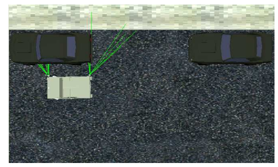
SEVA 2D
SEVA 3D



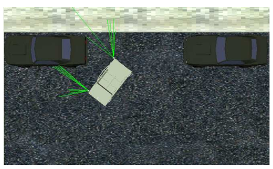
Searching Parking Space



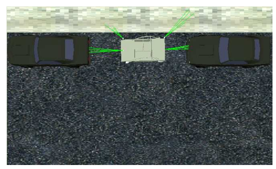
Positioning Outside



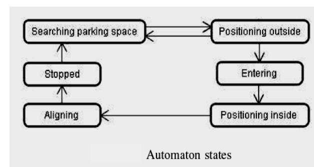
Entering



Positioning Inside



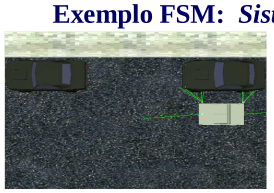
Aligning

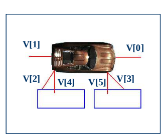


FSM - Finite State Machine

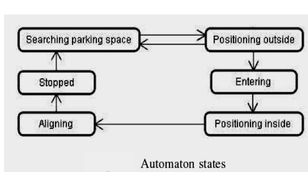
2. Máquina de Estados Finitos: FSM

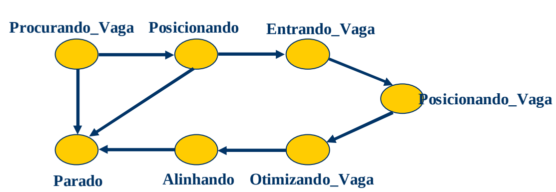
Exemplo FSM: ***Sistema de Estacionamento de Veículos Autônomo***






```
Se Estado_Atual(Procurando_Vaga) e Próximo_ao_Obstáculo(V[4]) e
   Próximo_ao_Obstáculo(V[5])
Então Speed = Avanço_Rápido e RotVel =  Direção_Reta;

Se Estado_Atual(Procurando_Vaga) e Longe__do_Obstáculo(V[2])e
   Longe__do_Obstáculo(V[3]) e Longe__do_Obstáculo(V[4]) e
   Longe__do_Obstáculo(V[5])
Então Troca_Estado(Posicionando) e Inicializa(Odômetro);

Se Estado_Atual(Posicionando)
Então Speed = Avanço_Rápido e Rotvel = Direção_Reta;

Se Estado_Atual(Posicionando) e Longe_do_Obstáculo(V[4]) e
   Deslocamento_Suficiente(Odômetro)
Então Estado_atual(Entrando_Vaga) e Inicializa(Odômetro);

Se Estado_Atual(Entrando_Vaga)
Então Speed = Ré_Rápida e RotVel = Giro_Esquerda_Max;
```



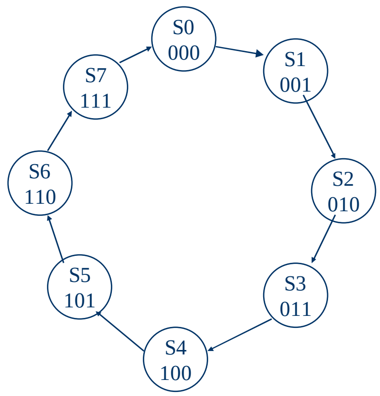


USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL: Introdução - Conceitos básicos



Exemplo simples:

- Estados e Transições
- Estados: S0 a S7
- Transições: Contador binário
  (Sem entradas)

Como ficaria o código VHDL
para implementar este contador binário...

23
Agosto 2008

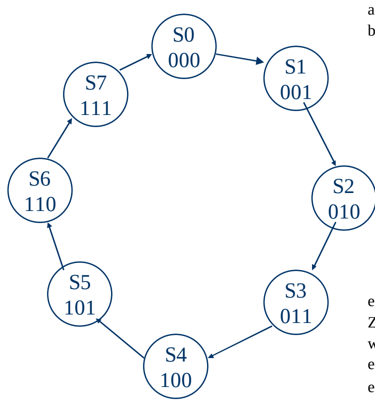
[Ref.: www.ece.northwestern.edu/~haizhou/Lec14.ppt]

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL: Introdução - Conceitos básicos




```
architecture ALGORITHM of BIN_COUNTER is
begin
  process
  variable PRESENT_STATE: BIT_VECTOR(2 downto 0) := B”111”;
  begin
    case PRESENT_STATE is
    when B”000”  => PRESENT_STATE := B”001”;
    when B”001”  => PRESENT_STATE := B”010”;
    when B”010”  => PRESENT_STATE := B”011”;
    when B”011”  => PRESENT_STATE := B”100”;
    when B”100”  => PRESENT_STATE := B”101”;
    when B”101”  => PRESENT_STATE := B”110”;
    when B”110”  => PRESENT_STATE := B”111”;
    when B”111”  => PRESENT_STATE := B”000”;
end case;
Z <= PRESENT_STATE after 10 nsec;
wait until (CLK = ‘1’;
end process;
end ALGORITHM;
```

24
Agosto 2008

[Ref.: www.ece.northwestern.edu/~haizhou/Lec14.ppt]

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

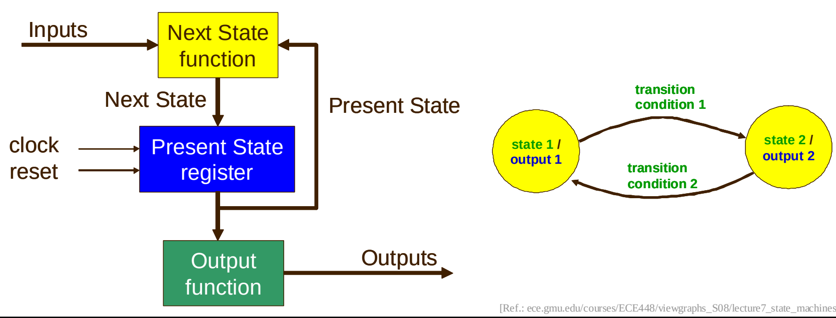
3. FSM em VHDL

Criando um FSM em VHDL: Moore Machine

Define a VHDL model of a FSM

using two processes

- One for the combinational logic for the next state and output functions
- One for the sequential elements



25
Agosto 2008

[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

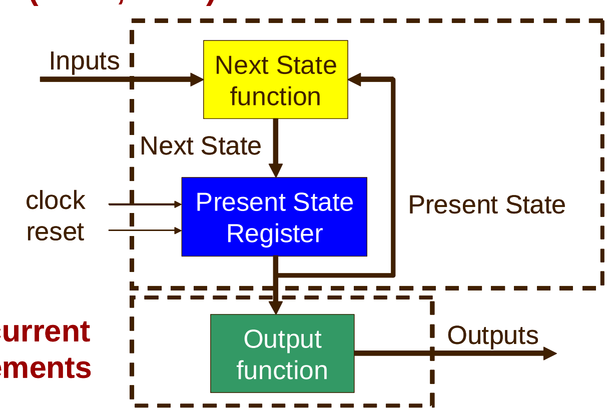
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL: Moore Machine

process(clock, reset)



concurrent statements

26
Agosto 2008

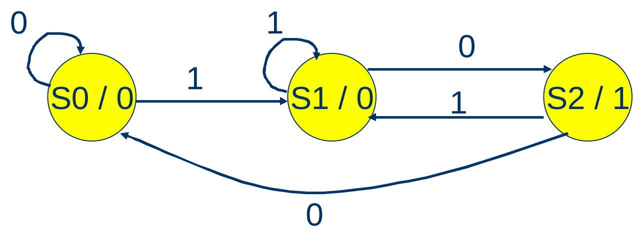
[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL: Moore Machine

Máquina de Moore que reconhece uma seqüência de entrada: "10"

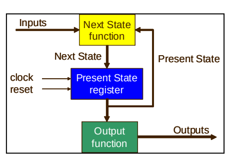


[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

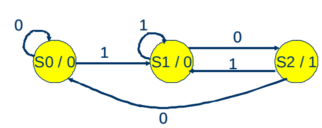
FSM - Finite State Machine

3. FSM em VHDL

Máquina de Moore que reconhece uma seqüência de entrada: "10"




```
TYPE state IS (S0, S1, S2);
SIGNAL Moore_state: state;

U_Moore: PROCESS (clock, reset)
BEGIN
    IF(reset = '1') THEN
      Moore_state <= S0;
    ELSIF (clock = '1' AND clock'event) THEN
      CASE Moore_state IS
        WHEN S0 =>
          IF input = '1' THEN
            Moore_state <= S1;
          ELSE
            Moore_state <= S0;
          END IF;
        WHEN S1 =>
          IF input = '0' THEN
            Moore_state <= S2;
          ELSE
            Moore_state <= S1;
          END IF;
        WHEN S2 =>
          IF input = '0' THEN
            Moore_state <= S0;
          ELSE
            Moore_state <= S1;
          END IF;
      END CASE;
    END IF;
END PROCESS;

Output <= '1' WHEN Moore_state = S2 ELSE '0';
```



[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

FSM - Finite State Machine

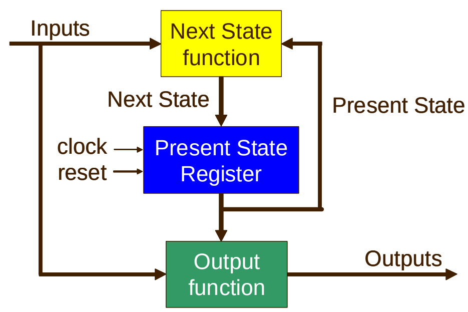
3. FSM em VHDL

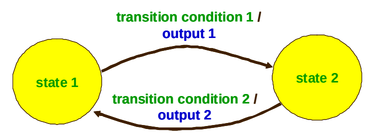
Criando um FSM em VHDL: Mealy Machine

Define a VHDL model of a FSM

using two processes

- One for the combinational logic for the next state and output functions
- One for the sequential elements





[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

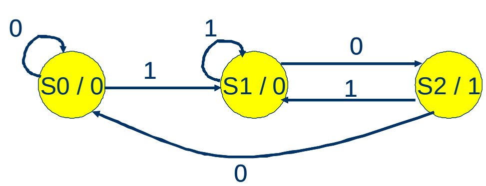
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

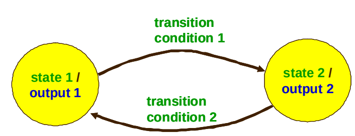
FSM - Finite State Machine

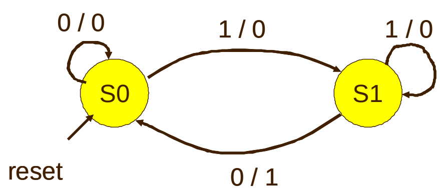
3. FSM em VHDL

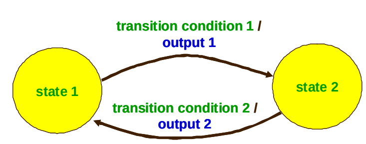
Criando um FSM em VHDL: Moore Machine x Mealy Machine

Máquina de Estados que reconhece uma seqüência de entrada: "10"









[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

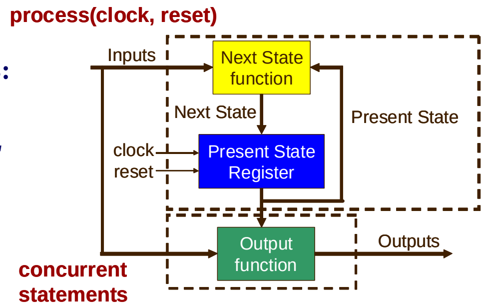
FSM - Finite State Machine

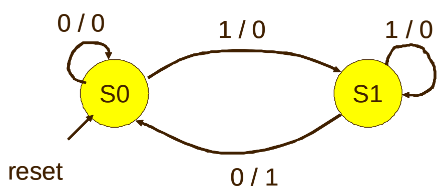
3. FSM em VHDL

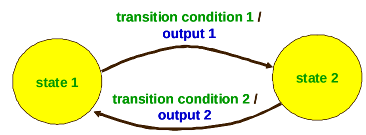
Criando um FSM em VHDL:

Máquina de Mealy

Reconhece uma seqüência: "10"







[Ref.: ece.gmu.edu/courses/ECE448/viewgraphs_S08/lecture7_state_machines.ppt]

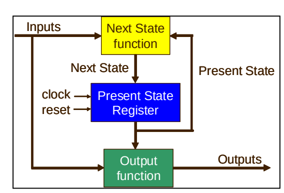
31
Agosto 2008

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

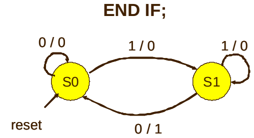
FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL: Mealy Machine




```
TYPE state IS (S0, S1);
SIGNAL Mealy_state: state;

U_Mealy: PROCESS(clock, reset)
BEGIN
    IF(reset = '1') THEN
      Mealy_state <= S0;
    ELSIF (clock = '1' AND clock'event) THEN
      CASE Mealy_state IS
          WHEN S0 =>
            IF input = '1' THEN
               Mealy_state <= S1;
            ELSE
               Mealy_state <= S0;
            END IF;
          WHEN S1 =>
            IF input = '0' THEN
               Mealy_state <= S0;
            ELSE
               Mealy_state <= S1;
            END IF;
      END CASE;
    END IF;
END PROCESS;

Output <= '1' WHEN (Mealy_state = S1 AND input = '0') ELSE '0';
```



32
Agosto 2008

FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL:

*** Podem ser criadas diretamente em VHDL, entretanto se for usado o software Quartus da Altera, alguns cuidados devem ser tomados**

"Synthesis tools can recognize and encode Verilog HDL and VHDL state machines during synthesis. This section presents guidelines to ensure the best results when you use state machines. "
[From: Recommended HDL Coding Styles, Quartus II 8.0 Handbook, Volume 1 - p.6-59]

Altera recommends that you observe the following guidelines, which apply to both Verilog HDL and VHDL:

- ■ Assign default values to outputs derived from the state machine so that synthesis does not generate unwanted latches.
- ■ Separate the state machine logic from all arithmetic functions and data paths, including assigning output values.
- ■ If your design contains an operation that is used by more than one state, define the operation outside the state machine and cause the output logic of the state machine to use this value.
- ■ Use a simple asynchronous or synchronous reset to ensure a defined power-up state.

FSM - Finite State Machine

3. FSM em VHDL

Criando um FSM em VHDL:

*** VHDL usando o software Quartus da Altera**
[From: Recommended HDL Coding Styles, Quartus II 8.0 Handbook, Volume 1 - p.6-65]

VHDL State Machines
To ensure proper recognition and inference of VHDL state machines, represent the states in a state machine with enumerated types and use the corresponding types to make state assignments.

VHDL State Machine Coding Example
The following entity, vhd1_fsm, is an example of a typical VHDL FSM implementation
- State machine: 5 states (state_0 to state_4).
- The asynchronous reset sets the variable state to state_0.
- The sum of in1 and in2 (inputs) is an output of the state machine in state_1 and state_2.
- The difference (in1 - in2) is also used in state_1 and state_2.
- The temporary variables tmp_out_0 and tmp_out_1 store the sum and the difference of in1 and in2.
- Using these temporary variables in the various states of the state machine ensures proper resource sharing between the mutually exclusive states.

FSM - Finite State Machine

3. FSM em VHDL

Example 6–46. VHDL State Machine

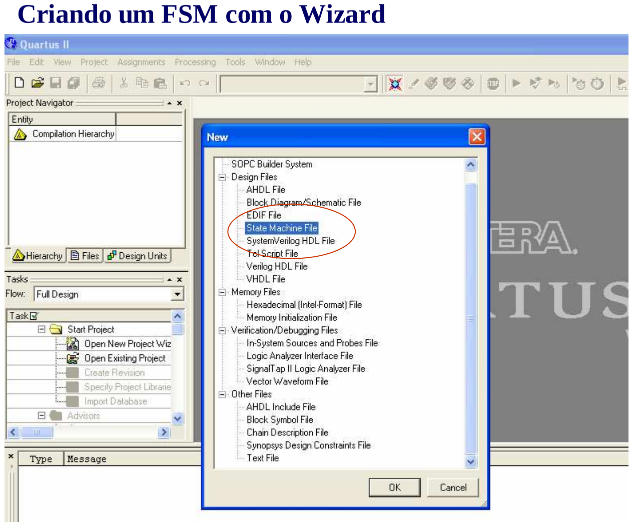
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.numeric_std.all;

ENTITY vhdl_fsm IS
   PORT(
      clk: IN STD_LOGIC;
      reset: IN STD_LOGIC;
      in1: IN UNSIGNED(4 downto 0);
      in2: IN UNSIGNED(4 downto 0);
      out_1: OUT UNSIGNED(4 downto 0)
      );
END vhdl_fsm;

ARCHITECTURE rtl OF vhdl_fsm IS
   TYPE Tstate IS (state_0, state_1, state_2, state_3, state_4);
   SIGNAL state: Tstate;
   SIGNAL next_state: Tstate;
BEGIN
   PROCESS(clk, reset)
   BEGIN
         IF reset = '1' THEN
            state <=state_0;
      ELSIF rising_edge(clk) THEN
            state <= next_state;
      END IF;
   END PROCESS;
   PROCESS (state, in1, in2)
      VARIABLE tmp_out_0: UNSIGNED (4 downto 0);
      VARIABLE tmp_out_1: UNSIGNED (4 downto 0);
   BEGIN ...
   END PROCESS;
END rtl;
```

```
PROCESS (state, in1, in2)
   VARIABLE tmp_out_0: UNSIGNED (4 downto 0);
   VARIABLE tmp_out_1: UNSIGNED (4 downto 0);
BEGIN
   tmp_out_0 := in1 + in2;
   tmp_out_1 := in1 - in2;
   CASE state IS
      WHEN state_0 =>
         out_1 <= in1;
         next_state <= state_1;
      WHEN state_1 =>
         IF (in1 < in2) then
            next_state <= state_2;
            out_1 <= tmp_out_0;
         ELSE
            next_state <= state_3;
            out_1 <= tmp_out_1;
         END IF;
      WHEN state_2 =>
         IF (in1 < "0100") then
            out_1 <= tmp_out_0;
         ELSE
            out_1 <= tmp_out_1;
         END IF;
            next_state <= state_3;
      WHEN state_3 =>
            out_1 <= "11111";
            next_state <= state_4;
      WHEN state_4 =>
            out_1 <= in2;
            next_state <= state_0;
      WHEN OTHERS =>
            out_1 <= "00000";
            next_state <= state_0;
   END CASE;
END PROCESS;
END rtl;
```


FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard



Menu:
File
New
State Machine File

Início... Menu:
File
New Project Wizard

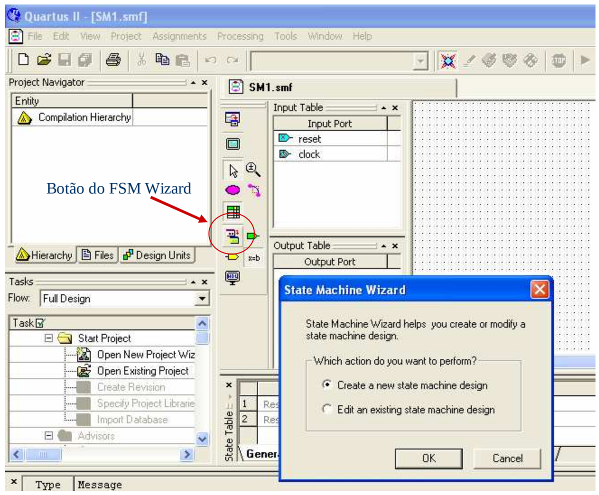
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: ***File – New - State Machine File***
State Machine Wizard



37
Agosto 2008

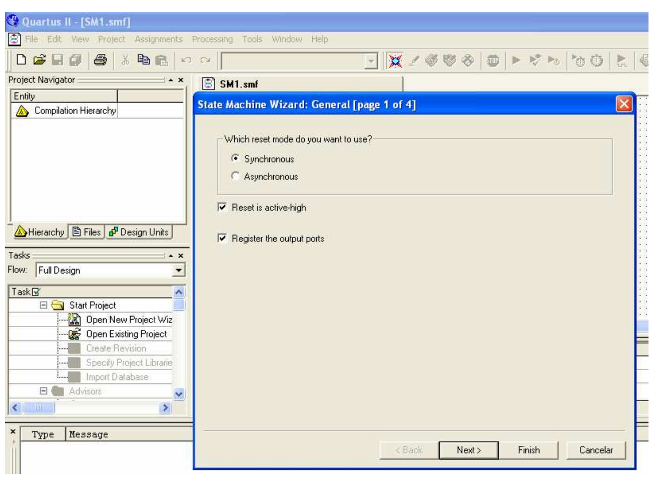
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: ***File – New - State Machine File***
State Machine Wizard



FSM
Synchronous

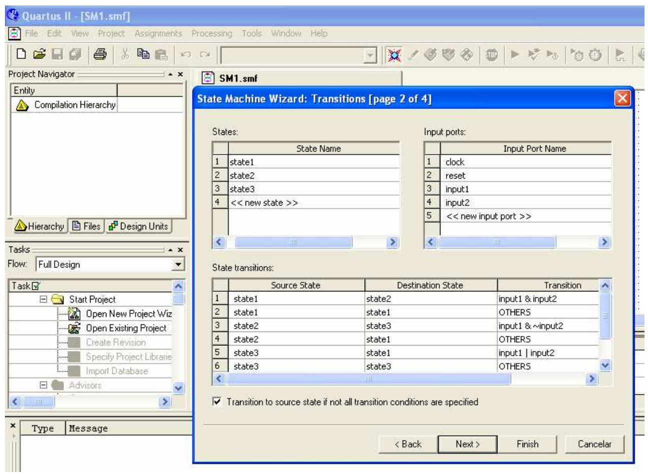
38
Agosto 2008

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: ***File – New - State Machine File***
State Machine Wizard



FSM
Synchronous

Define:
> States
> Input Ports
> State Transitions

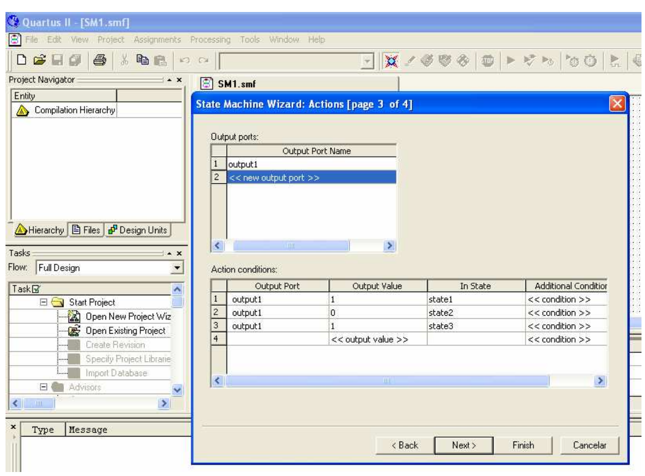
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: ***File – New - State Machine File***
State Machine Wizard



FSM
Synchronous

Define:
> States
> Input Ports
> State Transitions

> Output Ports
> Action Conditions

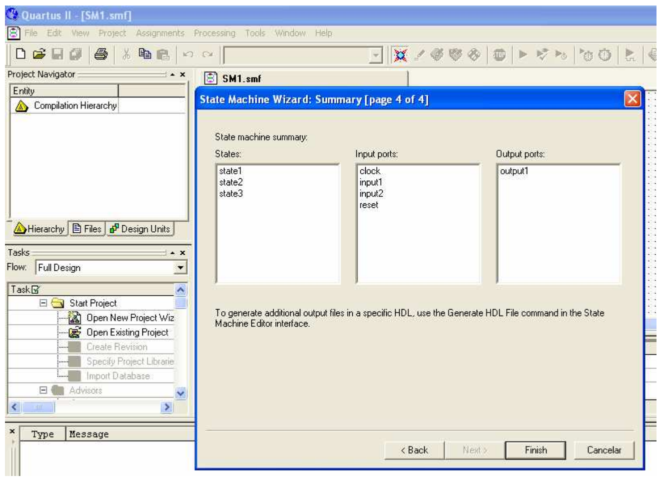
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: *File – New - State Machine File*
State Machine Wizard



FSM
Synchronous

Define:
> States
> Input Ports
> State Transitions

> Output Ports
> Action Conditions

FSM Summary

41
Agosto 2008

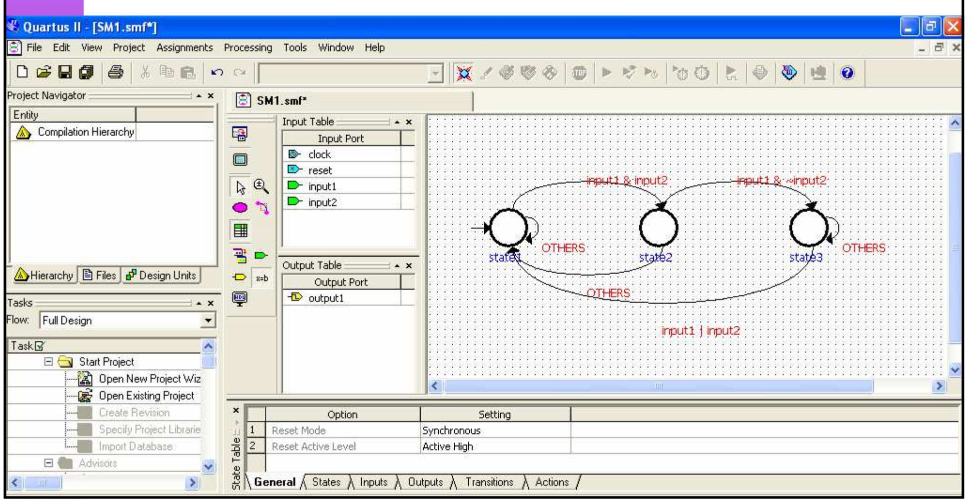
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: *File – New - State Machine File*
Edição (Wizard, Editor)
Menu: *File – Save as*

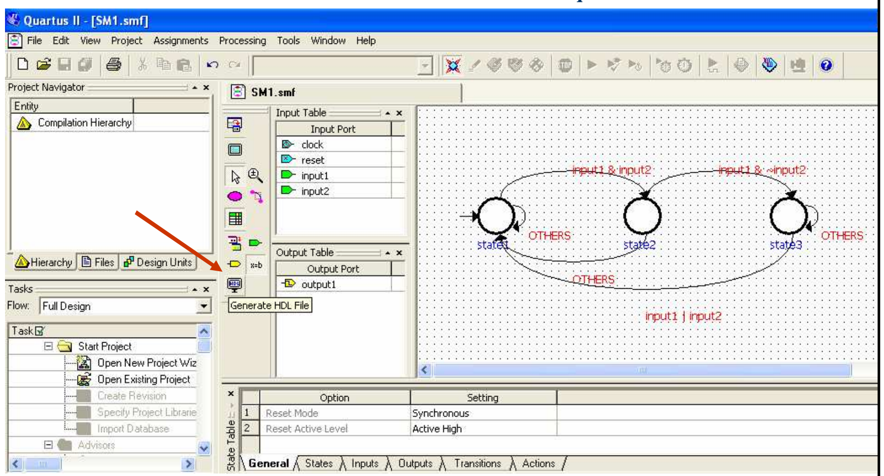


FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: ***File – New - State Machine File***
Edição (Wizard, Editor)
Gerar arquivo VHDL



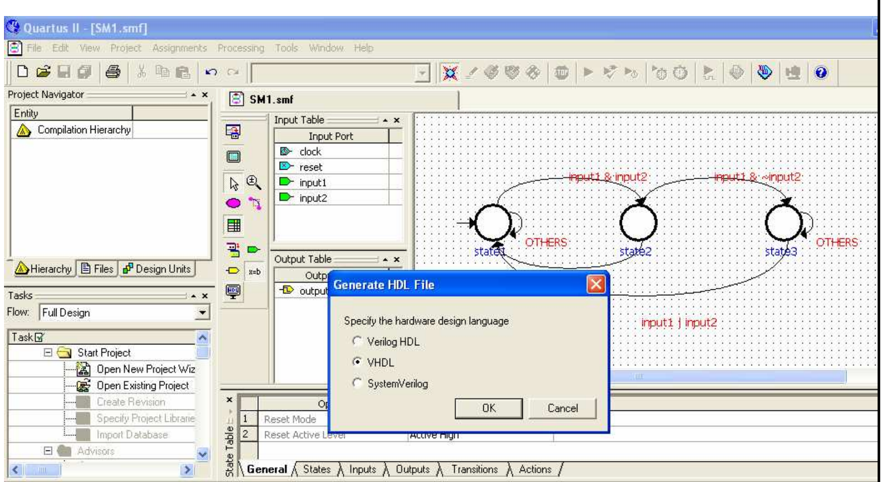
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

Menu: ***File – New - State Machine File***
Edição (Wizard, Editor)
Gerar arquivo VHDL



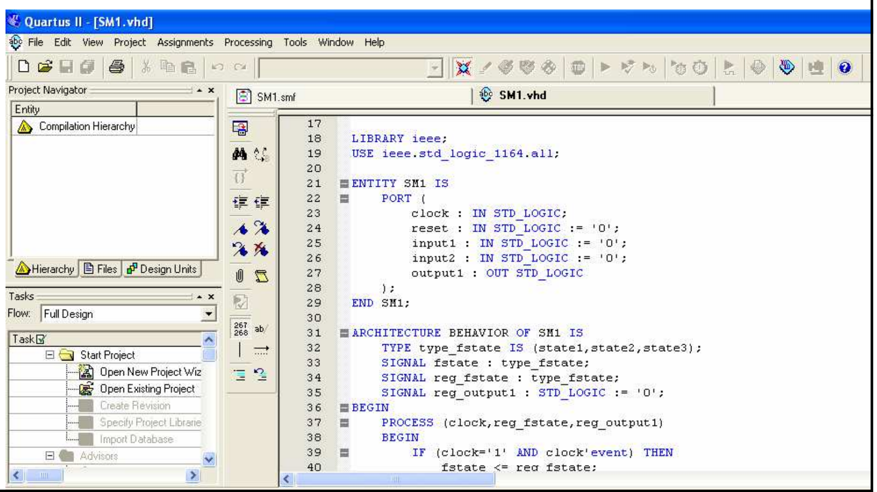
USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera)

Criando um FSM com o Wizard

FSM: Arquivo VHDL



USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

FSM - Finite State Machine

4. FSM no Quartus II (Altera) - VHDL FSM

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

ENTITY SM1 IS
  PORT (
    clock : IN STD_LOGIC;
    reset : IN STD_LOGIC := '0';
    input1 : IN STD_LOGIC := '0';
    input2 : IN STD_LOGIC := '0';
    output1 : OUT STD_LOGIC
  );
END SM1;

ARCHITECTURE BEHAVIOR OF SM1 IS
  TYPE type_fstate IS (state1,state2,state3);
  SIGNAL fstate : type_fstate;
  SIGNAL reg_fstate : type_fstate;
  SIGNAL reg_output1 : STD_LOGIC := '0';
BEGIN
  PROCESS (clock,reg_fstate,reg_output1)
  BEGIN
    IF (clock='1' AND clock'event) THEN
      fstate <= reg_fstate;
      output1 <= reg_output1;
    END IF;
  END PROCESS;
```

```
PROCESS (fstate,reset,input1,input2)
BEGIN
  IF (reset='1') THEN
    reg_fstate <= state1;
    reg_output1 <= '0';
  ELSE
    reg_output1 <= '0';
    CASE fstate IS
      WHEN state1 =>
        IF (((input1 = '1') AND (input2 = '1')))
        THEN  reg_fstate <= state2;
        ELSE  reg_fstate <= state1;    END IF;
        reg_output1 <= '1';
      WHEN state2 =>
        IF (((input1 = '1') AND NOT((input2 = '1'))))
        THEN  reg_fstate <= state3;
        ELSE  reg_fstate <= state1;    END IF;
        reg_output1 <= '0';
      WHEN state3 => . . .
      WHEN OTHERS =>
        reg_output1 <= 'X';
        report "Reach undefined state";
    END CASE;
  END IF;
END PROCESS;
END BEHAVIOR;
```

FSM - Finite State Machine

5. HANDS ON!

Crie uma Máquina de Estados (FSM)

Simule (Simulação Funcional)

Pratique... Especifique... Detalhe... Desenvolva... Implemente!

****

Trabalho Prático 01

****

Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO
Web institucional: http://www.icmc.usp.br/ssc/
Página pessoal: http://www.icmc.usp.br/~fosorio/
E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Proj. e Implementação de Sistemas Embarcados I
Ferramentas: Altera - Quartus, NIOS II, Cyclone Dev-Kit
Web disciplina: Http://www.icmc.usp.br/~fosorio/
> Programa, Material de Aulas, Critérios de Avaliação,
> Material de Apoio, Trabalhos Práticos