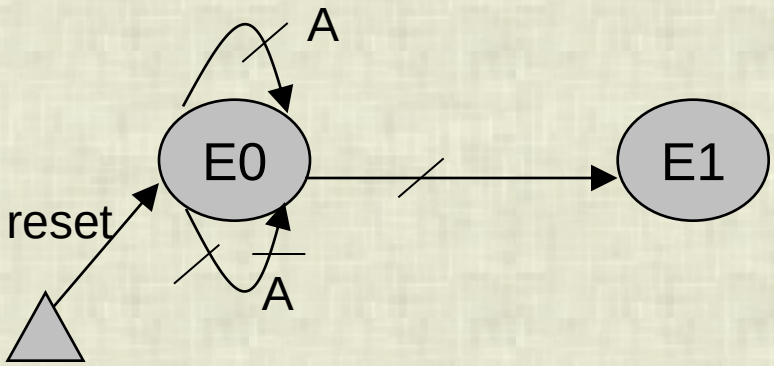
ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

Máquina de Estados Finito

prof. Dr. César Augusto M. Marcon
prof. Dr. Edson Ifarraguirre Moreno

MÁQUINA DE ESTADOS

- **Elementos do Modelo**
  - Estado
  - Transição
    - Relógio
  - Reset



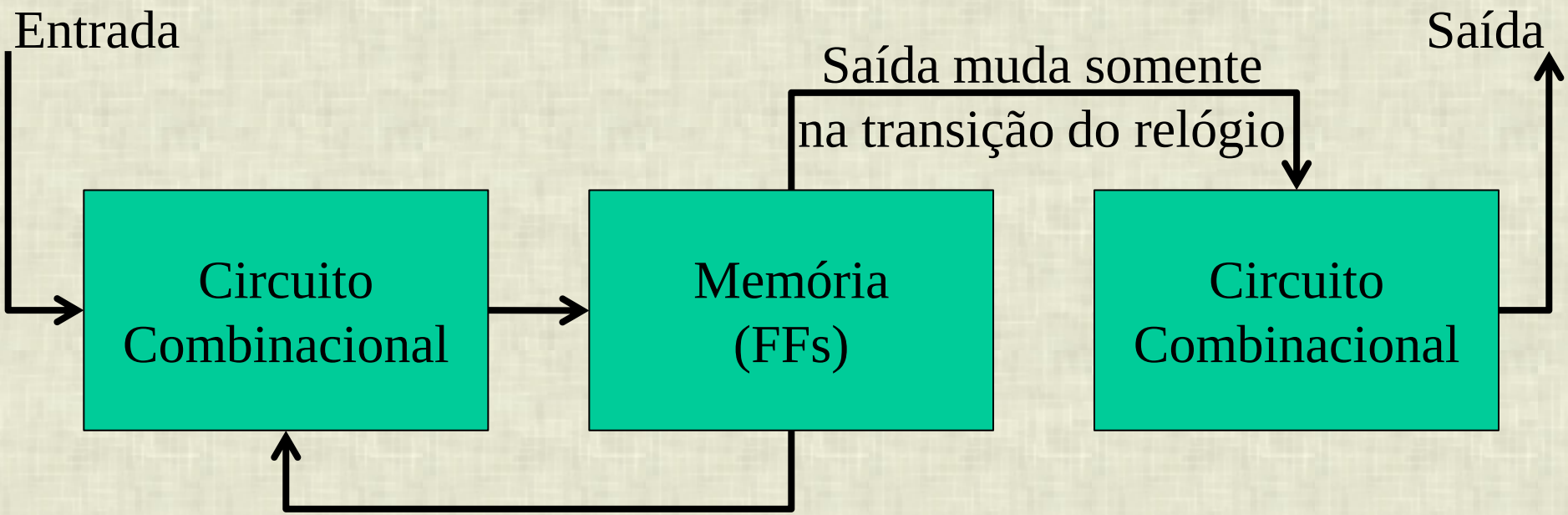
- **Determinismo**
  - Modelo determinístico
  - Modelo não determinístico

- **Tipos de Máquina**
  - Mealy versus Moore

- **Representação de Máquina de Estados Finito**
  - Um processo com um único sinal representando o estado
  - Dois processos, um representando a operação combinacional e outro representando o elemento de memorização para troca de estados

Máquina de Moore

Moore → saídas são calculadas dependendo apenas do ESTADO ATUAL



MÁQUINA DE ESTADOS - Moore

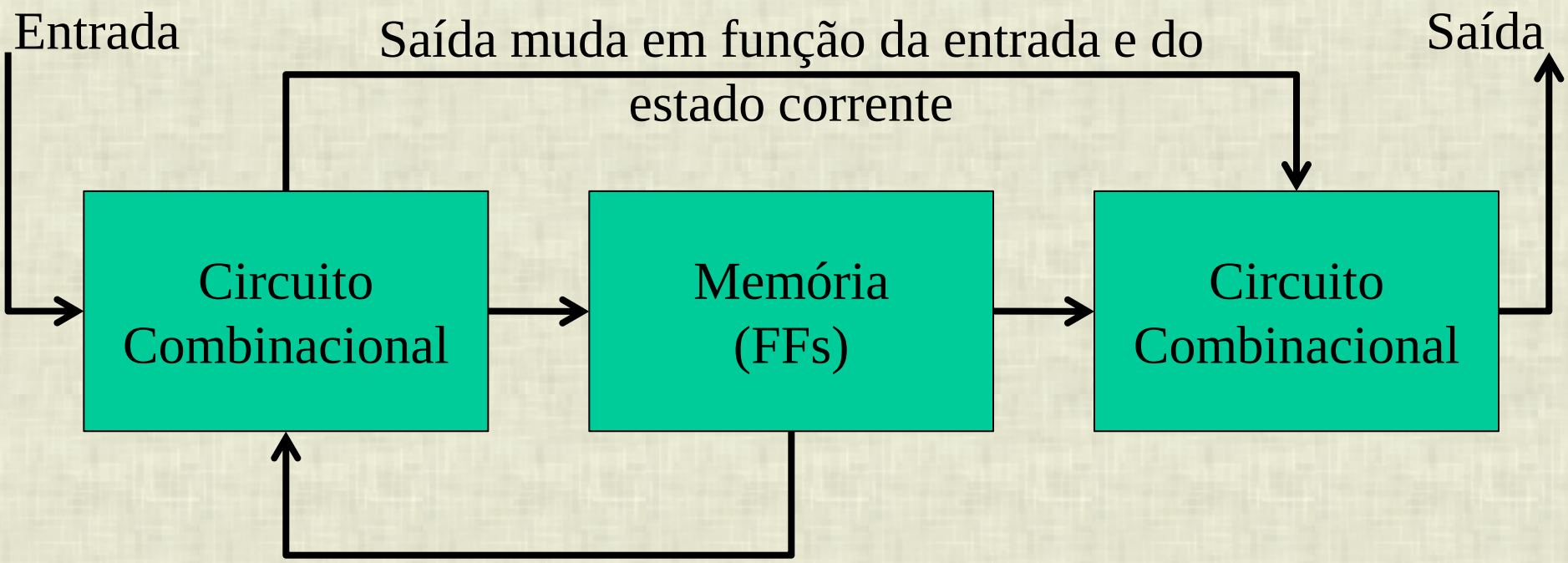
```
entity Moore is
    port
    (
        ck: in std_logic;
        entrada: in std_logic;
        saida: out std_logic
   );
end Moore;

architecture M of Moore is
    type STATE_TYPE is (S0, S1, S2, S3);
    signal estado: STATE_TYPE;
begin
    process(ck, reset)
    begin
        if reset = '1' then
            estado <= S0;
        elsif ck'event and ck = '1' then

                    << Máquina de estados >>

        end if;
    end process;
end A_Moore;
```

MÁQUINA DE ESTADOS - Moore

```
case estado is
     when S0 =>
          saida <= '0';
          if entrada = '1' then
               estado <= S2;
          end if;

     when S1 =>
          saida <= '1';
          if entrada = '0' then
               estado <= S0;
          else
               estado <= S2;
          end if;

     when S2 =>
          saida <= '1';
          if entrada = '1' then
               estado <= S3;
          end if;

     when S3 =>
          saida <= '0';
          if entrada = '1' then
               estado <= S1;
          end if;
end case;
```

Máquina de Mealy

Mealy → saídas são calculadas à partir do ESTADO ATUAL e ENTRADAS



MÁQUINA DE ESTADOS - Mealy

```
entity Mealy is
    port
    (
        ck: in std_logic;
        entrada: in std_logic;
        saida: out std_logic
   );
end Mealy;

architecture A_Mealy of Mealy is
    type STATE_TYPE is (S0, S1, S2, S3);
    signal estado: STATE_TYPE;
begin
    <PROCESSO DE CONTROLE DE ESTADO>
    <PROCESSO DE CONTROLE DE SAÍDA>
end A_Mealy;
```

MÁQUINA DE ESTADOS - Mealy

Processo de controle de estado

```
process(ck, reset)
begin
  if reset = '1' then
    estado <= S0;
  elsif ck'event and ck = '1' then
        << Máquina de estados >>
  end if;
end process;
```

```
case estado is
  when S0 =>
    if entrada = '1' then
      estado <= S2;
    end if;

  when S1 =>
    if entrada = '0' then
      estado <= S0;
    else
      estado <= S2;
    end if;

    when S2 =>
      if entrada = '1' then
        estado <= S3;
      end if;

    when S3 =>
      if entrada = '1' then
        estado <= S1;
      end if;
end case;
```

Processo de controle da saída

```
Saída <= '0' when entrada = '1' and
                 (estado  = S0 or estado = S3) else
         '1';
```

Exercícios

1. Descreva em VHDL o sistema SAgua através de uma máquina de estados.

Deve ser representado o comportamento da água frente a mudança de temperatura, sabendo que a água pode estar em um de três estados: ***sólido***, ***líquido*** ou ***gasoso***.

Considere que SAgua tem como entradas:

(i) ***temp*** - uma porta de um bit, que em 0 representa frio e em 1 representa calor;
(ii) ***estInic*** – porta com o estado da água codificado;
(iii) ***ck*** - sinal de controle para efetuar as transições em SAgua; e
(iv) ***start*** - porta de controle que, em 1, indica que o valor de estInic deve ser armazenado em SAgua – considere que start é síncrono com ck.

Como saídas SAgua tem:

(i) ***Estado*** que é um sinal codificado com o valor do estado da água.

Exercícios

- **Faça a descrição de uma máquina reconhecedora de padrões. Esta máquina deve reconhecer os seguintes padrões:**

A	0	x	x	0	0	1	1
B	1	x	1	x	x	0	1

- **Ao reconhecer um padrão válido, a saída reconheceu deve ir para 1, enquanto não tiver uma nova entrada e o contador cont deve ser incrementado.**
- **Este deve conter o número de vezes que um padrão foi reconhecido desde a máquina ter sido inicializada com o sinal de reset em 1**