**Universidade Federal de Santa Catarina**
**Centro Tecnológico**
Departamento de Informática e Estatística
**Curso de Graduação em Ciências da Computação**


# Sistemas Digitais

**INE 5406**

## Aula 5-P

### Descrição em VHDL, síntese e simulação de circuitos seqüenciais.

**Prof. José Luís Güntzel**
**guntzel@inf.ufsc.br**

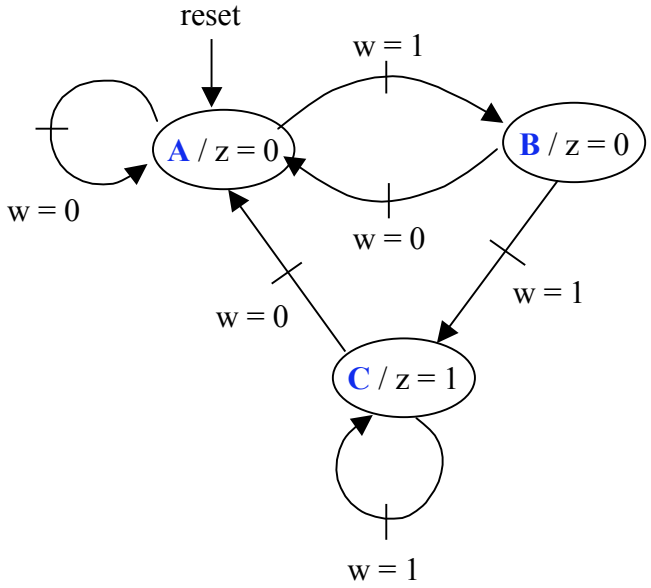
**www.inf.ufsc.br/~guntzel/ine5406/ine5406.html**

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

- VHDL não define padrão para a descrição de máquinas de estados finitos
- Existe mais de uma maneira de se descrever uma dada FSM
- Considere a FSM com o seguinte diagrama de estados

- Esta FSM segue o **modelo de Moore**



# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

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

3   ENTITY simple IS
4         PORT (   Clock, Resetn, w      : IN     STD_LOGIC ;
                   z                     : OUT    STD_LOGIC ) ;
5   END simple ;

7   ARCHITECTURE Behavior OF simple IS
8         TYPE State_type IS (A, B, C) ;
9         SIGNAL y : State_type ;
10  BEGIN
11        PROCESS ( Resetn, Clock )
12        BEGIN
13              IF Resetn = '0' THEN
14                    y <= A ;
15              ELSIF (Clock'EVENT AND Clock = '1') THEN
16                    CASE y IS
17                          WHEN A =>
18                                IF w = '0' THEN
19                                      y <= A ;
20                                ELSE
21                                      y <= B ;
22                                END IF ;
```

**FSM descrita segundo o Modelo de Moore, Versão 1 (somente 1 processo)**

"TYPE" permite criar um tipo de sinal definido pelo usuário.
Neste caso, se está definindo um dado chamado State_type que pode assumir um entre 3 valores: A, B. C

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

```
15              ELSIF (Clock'EVENT AND Clock = '1') THEN
16                      CASE y IS
17                              WHEN A =>
18                                      IF w = '0' THEN
19                                              y <= A ;
20                                      ELSE
21                                              y <= B ;
22                                      END IF ;
23                              WHEN B =>
24                                      IF w = '0' THEN
25                                              y <= A ;
26                                      ELSE
27                                              y <= C ;
28                                      END IF ;
29                              WHEN C =>
30                                      IF w = '0' THEN
31                                              y <= A ;
32                                      ELSE
33                                              y <= C ;
34                                      END IF ;
35                      END CASE ;
36              END IF ;
37      END PROCESS ;
38      z <= '1' WHEN y = C ELSE '0' ;
39  END Behavior ;
```

**FSM descrita segundo o Modelo de Moore, Versão 1 (continuação)**

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

```
ARCHITECTURE Behavior OF simple IS
    TYPE State_type IS (A, B, C) ;
    SIGNAL y_present, y_next : State_type ;
BEGIN
    PROCESS ( w, y_present )
    BEGIN
        CASE y_present IS
            WHEN A =>
                IF w = '0' THEN
                    y_next <= A ;
                ELSE
                    y_next <= B ;
                END IF ;
            WHEN B =>
                IF w = '0' THEN
                    y_next <= A ;
                ELSE
                    y_next <= C ;
                END IF ;
            WHEN C =>
                IF w = '0' THEN
                    y_next <= A ;
                ELSE
                    y_next <= C ;
            END IF ;
        END CASE ;
    END PROCESS ;
```

**FSM descrita segundo o Modelo de Moore, Versão 2 (2 processos)**

Um processo para o bloco de próxima estado …

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

**FSM descrita segundo o Modelo de Moore, Versão 2 (2 processos)**

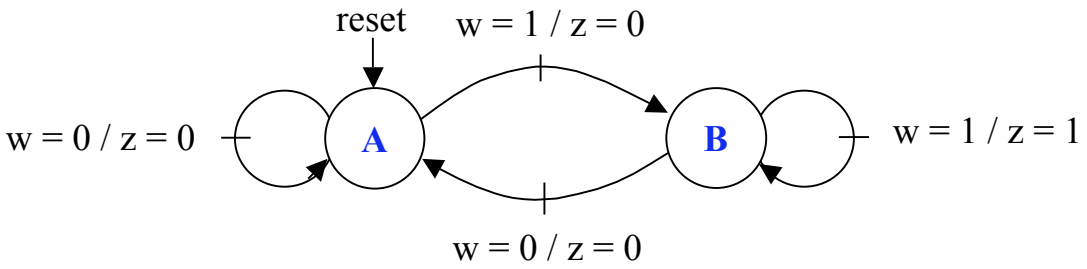
```
    PROCESS (Clock, Resetn)
    BEGIN
        IF Resetn = '0' THEN
            y_present <= A ;
        ELSIF (Clock'EVENT AND Clock = '1') THEN
            y_present <= y_next ;
        END IF ;
    END PROCESS ;

    z <= '1' WHEN y_present = C ELSE '0' ;
END Behavior ;
```

… e outro processo para o registrador de estado

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

- **Considere a FSM descrita pelo diagrama de estados que segue, a qual segue o modelo de Mealy**



# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY mealy IS
      PORT ( Clock, Resetn, w       : IN      STD_LOGIC ;
                   z                : OUT     STD_LOGIC ) ;
END mealy ;

ARCHITECTURE Behavior OF mealy IS
      TYPE State_type IS (A, B) ;
      SIGNAL y : State_type ;
BEGIN
      PROCESS ( Resetn, Clock )
      BEGIN
            IF Resetn = '0' THEN
                  y <= A ;
            ELSIF (Clock'EVENT AND Clock = '1') THEN
                  CASE y IS
                        WHEN A =>
                              IF w = '0' THEN y <= A ;
                              ELSE y <= B ;
                              END IF ;
                         WHEN B =>
                              IF w = '0' THEN y <= A ;
                              ELSE y <= B ;
                              END IF ;
                  END CASE ;
            END IF ;
      END PROCESS ;
```

**FSM descrita segundo o Modelo de Mealy (2 processos)**

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados

```
    PROCESS ( y, w )
    BEGIN
        CASE y IS
            WHEN A =>
                z <= '0' ;
            WHEN B =>
                z <= w ;
        END CASE ;
    END PROCESS ;
END Behavior ;
```

**FSM descrita segundo o Modelo de Mealy (2 processos)**
Continuação

# Máquinas de Estados

## Outra FSM

```
Use ieee.std_logic_1164.all;

ENTITY Cont IS
  PORT(clk, reset, flag1 :  IN STD_LOGIC;
           S1, S2, S3 : OUT STD_LOGIC);
END Cont;

ARCHITECTURE comportamento OF Cont IS
SIGNAL state: STD_LOGIC_VECTOR(1 DOWNTO 0);
BEGIN
PROCESS (clk, reset)
BEGIN
  IF (reset='0') THEN
    state <= "00";
```

Definição das variáveis de estado

# Circuitos Seqüenciais em VHDL

## Máquinas de Estados (continuação)

### Outra FSM

```
ELSIF (clk'EVENT AND clk = '1') THEN
   CASE state IS
     WHEN "00" =>  state <= "01";   S1 <= '0';   S2 <= '1';   S3 <= '1';
     WHEN "01" =>  IF flag1 = '1' THEN   state <= "10";
                   ELSE state <= "11";  END IF;
                   S1 <= '1';   S2 <= '1';   S3 <= '0';
     WHEN "10" =>  state <= "11";  S1 <= '1';   S2 <= '1';   S3 <= '1';
     WHEN "11" =>  state <= "00";  S1 <= '1';   S2 <= '0';   S3 <= '1';
     WHEN OTHERS => state <= "00";
   END CASE;
  END IF;
END PROCESS;
END comportamento;
```