USP - ICMC - SSC
SSC 0113 (Lab ELD II) - 2o. Semestre 2012

Disciplina de
SSC0113 - Elementos de Lógica Digital II (Prática)

Prof. Fernando Osório
Email: fosorio [at] { icmc. usp. br , gmail. com }
Estagiário PAE: Diogo Ortiz Correa
Email: diogosoc [at] { icmc. usp. br }
Web: http://www.icmc.usp.br/~fosorio/
Wiki ICMC: http://wiki.icmc.usp.br/index.php/SSC-113-2012(fosorio)

Aula 04

Aula 04 – Máquinas de Estados
Finite State Machine

Agenda:

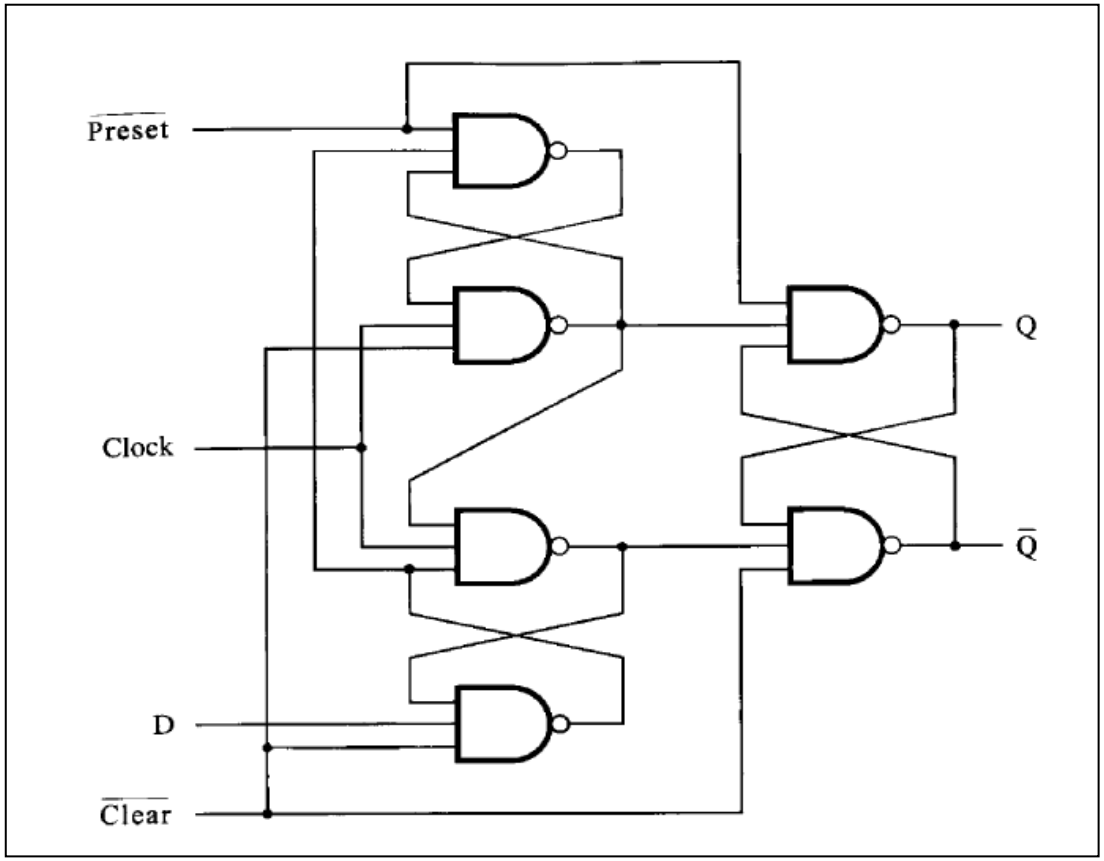
1. **Registradores (B&V Chapter 7)**
2. **Máquina de Estados: Circuitos Sequenciais (B&V Ch. 8)**

   ***Finite State Machine* (FSM) / Finite State Automaton (FSA)**

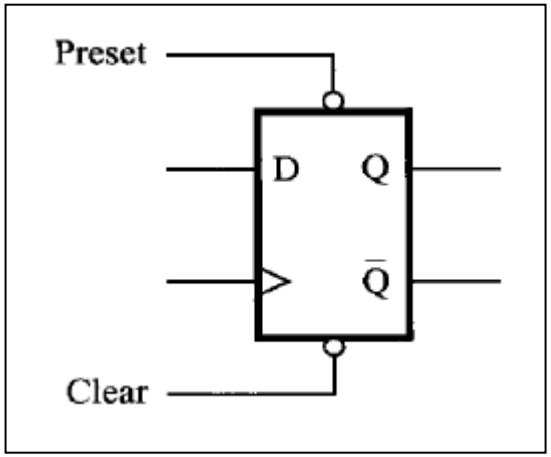
   >> State Machine <<
3. **Máquina de Moore**
4. **Máquina de Mealy**

1. Registradores em VHDL

Flip-Flops e Registradores



Positive-edge-triggered D flip-flop with *Clear* and *Preset*.



Synchronous

1. Registradores em VHDL

VHDL

D Flip-Flop with Asynchronous Reset

```
-- D Flip-Flop with Asynchronous Reset

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY flipflop IS
    PORT (   D, Resetn, Clock : IN STD_LOGIC ;
             Q : OUT STD_LOGIC ) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
       IF Resetn = '0' THEN
            Q <= '0' ;
       ELSIF Clock'EVENT AND Clock = '1' THEN
            Q <= D ;
       END IF ;
    END PROCESS ;
END Behavior ;
```

1. Registradores em VHDL

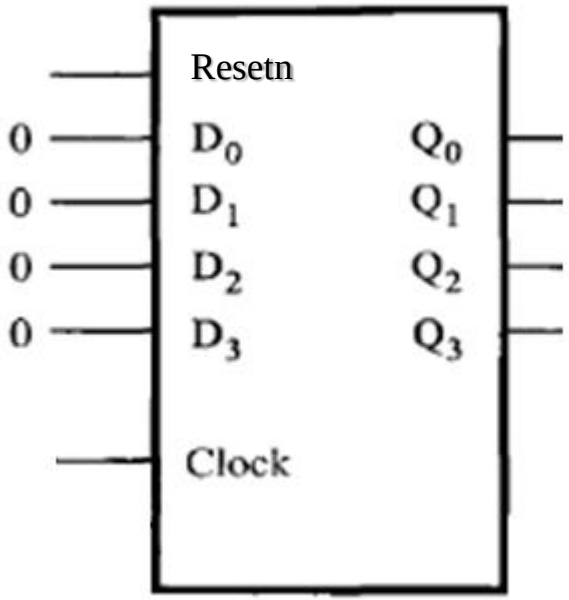
VHDL

D Flip-Flop with Synchronous Reset

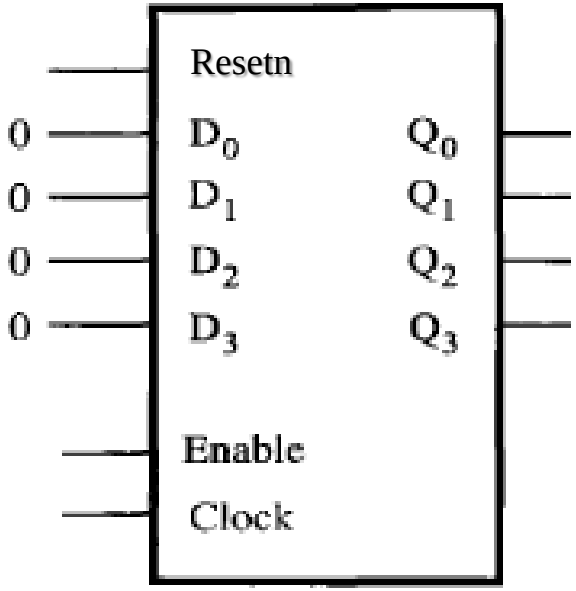
```
-- D Flip-Flop with Synchronous Reset

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY flipflop IS
    PORT (   D, Resetn, Clock : IN STD_LOGIC ;
             Q : OUT STD_LOGIC ) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS
    BEGIN
       WAIT UNTIL Clock = '1' ;
       IF Resetn = '0' THEN
            Q <= '0' ;
       ELSE
            Q <= D ;
       END IF ;
    END PROCESS ;
END Behavior ;
```

1. Registradores em VHDL

Registradores



4-bit register with asynchronous Reset



4-bit register with
async Reset and sync Enable

1. Registradores em VHDL

VHDL

4 Bit Register with Asynchronous Reset

```
-- 4-bit register with asynchronous Reset

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY reg4 IS
    PORT (   D : IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
             Resetn, Clock : IN STD_LOGIC ;
             Q : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) ) ;
END reg4 ;

ARCHITECTURE Behavior OF reg4 IS
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
       IF Resetn = '0' THEN
            Q <= "0000" ;
       ELSIF Clock'EVENT AND Clock = '1' THEN
            Q <= D ;
       END IF ;
    END PROCESS ;
END Behavior ;
```

1. Registradores em VHDL

VHDL

N Bit Register with Asynchronous Reset

```
-- N-bit register with asynchronous Reset
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY reg_n IS
    GENERIC ( N: INTEGER := 16 );
    PORT (   D : IN STD_LOGIC_VECTOR(N-1 DOWNTO 0) ;
             Resetn, Clock : IN STD_LOGIC ;
             Q : OUT STD_LOGIC_VECTOR(N-1 DOWNTO 0) ) ;
END reg_n ;

ARCHITECTURE Behavior OF reg_n IS
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
       IF Resetn = '0' THEN
            Q <= (OTHERS => '0') ;
       ELSIF Clock'EVENT AND Clock = '1' THEN
            Q <= D ;
       END IF ;
    END PROCESS ;
END Behavior ;
```

1. Registradores em VHDL

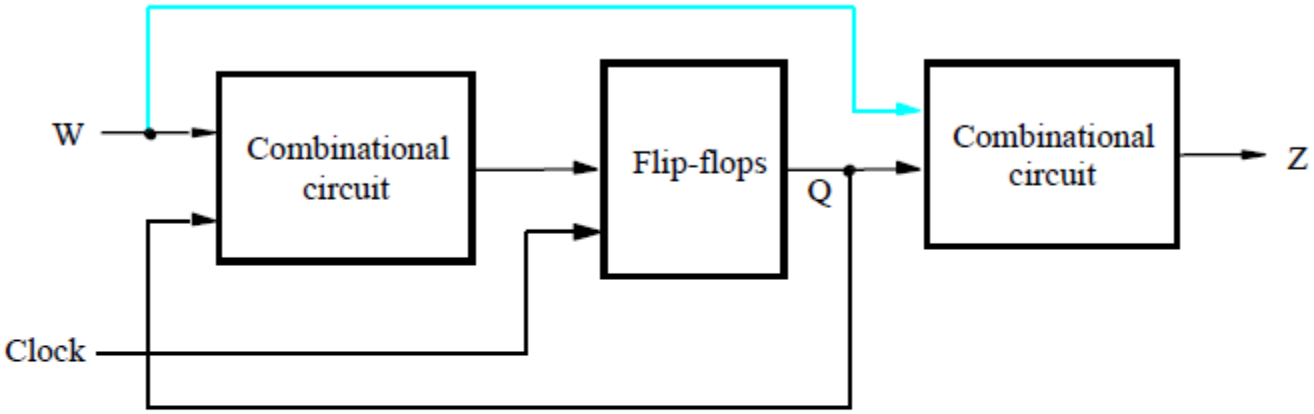
VHDL

N Bit
Register
with
Asynchronous
Reset
Synchronous
Enable

```
-- N-bit register with asynchronous Reset
--        and syncronous Enable

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY reg_nsync IS
    GENERIC ( N: INTEGER := 16 );
    PORT (   D : IN STD_LOGIC_VECTOR(N-1 DOWNTO 0) ;
             Resetn, Enable, Clock : IN STD_LOGIC ;
             Q : OUT STD_LOGIC_VECTOR(N-1 DOWNTO 0) ) ;
END reg_nsync ;

ARCHITECTURE Behavior OF reg_nsync IS
BEGIN
    PROCESS ( Resetn, Enable, Clock )
    BEGIN

--  JUST DO IT !      :^)

    END PROCESS ;
END Behavior ;
```

2. Máquinas de Estados

Finite State Machine

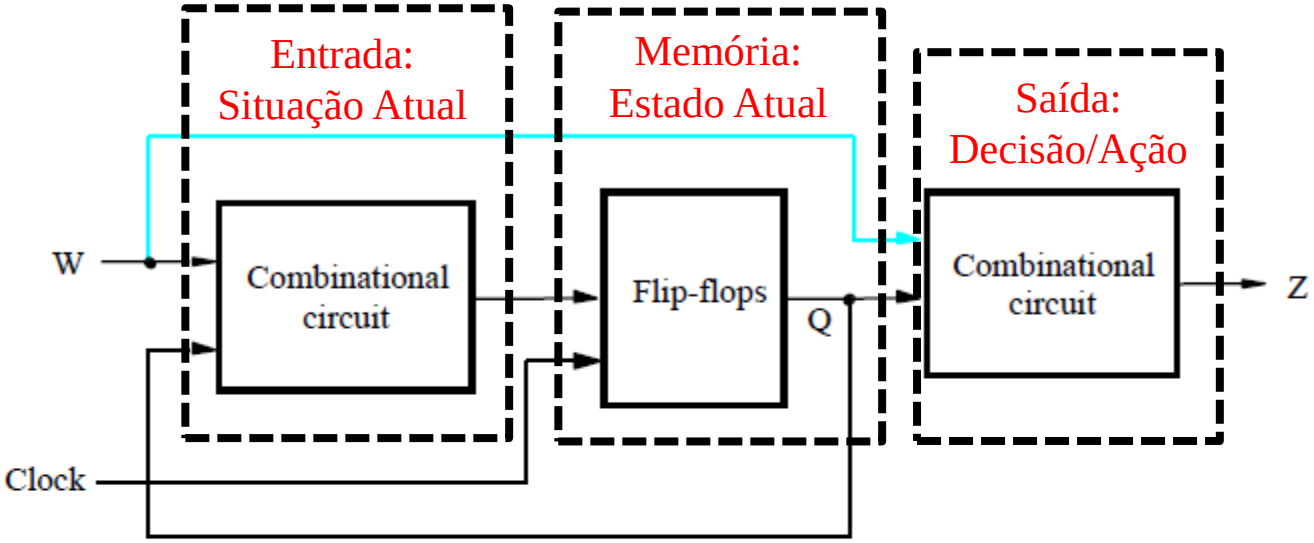


The general form of a sequential circuit.

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – Chapter 8)

2. Máquinas de Estados

Finite State Machine



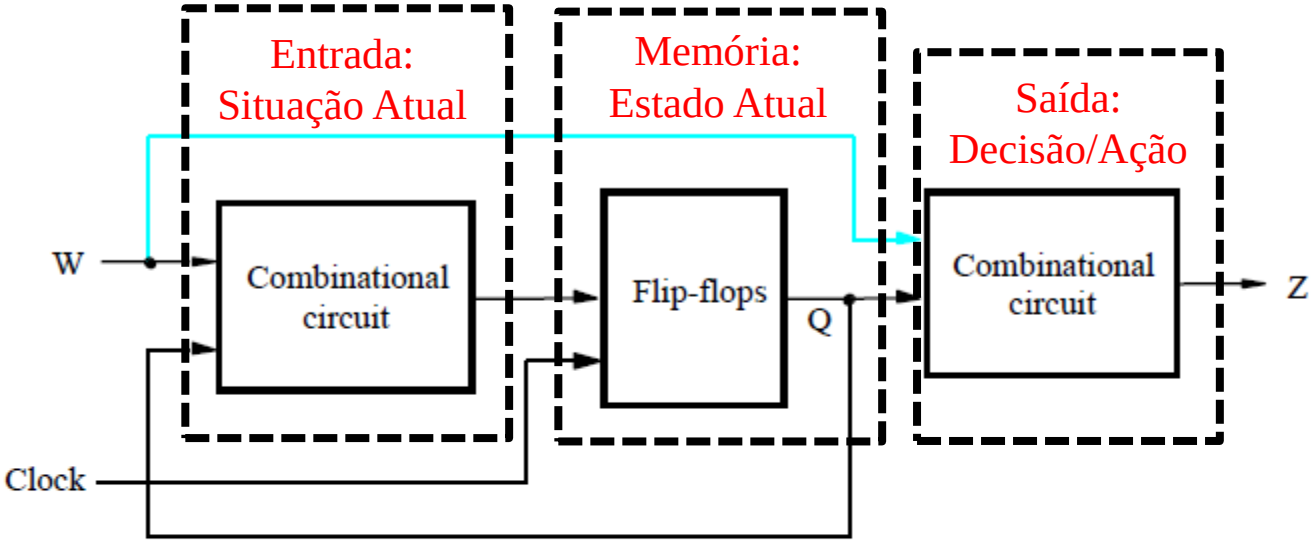
SEQUÊNCIA DADOS E DE ESTADOS (Evolui no tempo)
CIRCUITO SEQUENCIAL

The general form of a sequential circuit.

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – Chapter 8)

2. Máquinas de Estados

Finite State Machine



Exemplo: **"ao encontrar dois 1s seguidos (11) gera 1 na saída"**

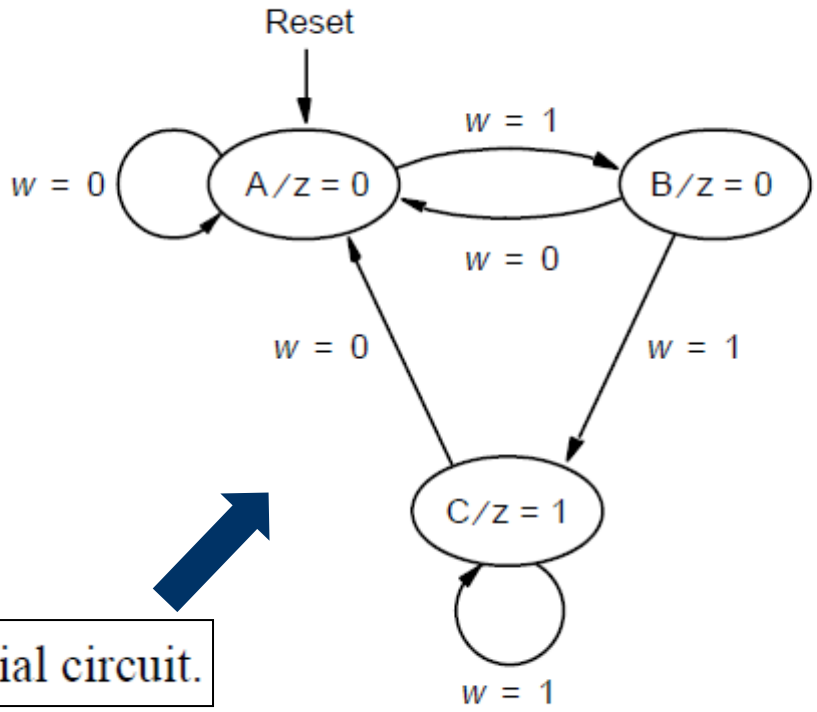
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – Chapter 8)

2. Máquinas de Estados

Finite State Machine

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	A	B	0
B	A	C	0
C	A	C	1



State diagram of a simple sequential circuit.

Exemplo: **"ao encontrar dois 1s seguidos (11) gera 1 na saída"**

Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – Chapter 8)

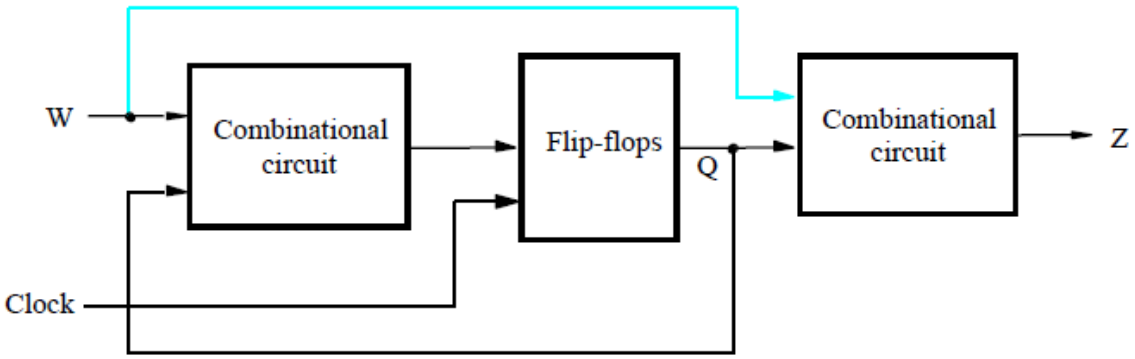
2. Máquinas de Estados: Moore e Mealy

Finite State Machine

Moore machine is a finite-state machine
whose output values are determined solely by its current state.
The Moore machine name is related to his creator Edward F. Moore.

This is in contrast to…

Mealy machine is a finite-state machine,
whose output values are determined both by
its current state and by the values of its inputs.
The Mealy machine name is related to his creator George H. Mealy.

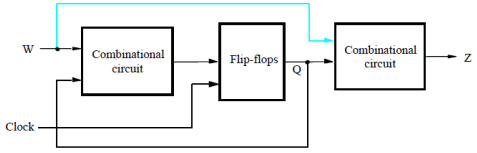


Extraído de Wikipedia: http://en.wikipedia.org/wiki/Finite-state_machine

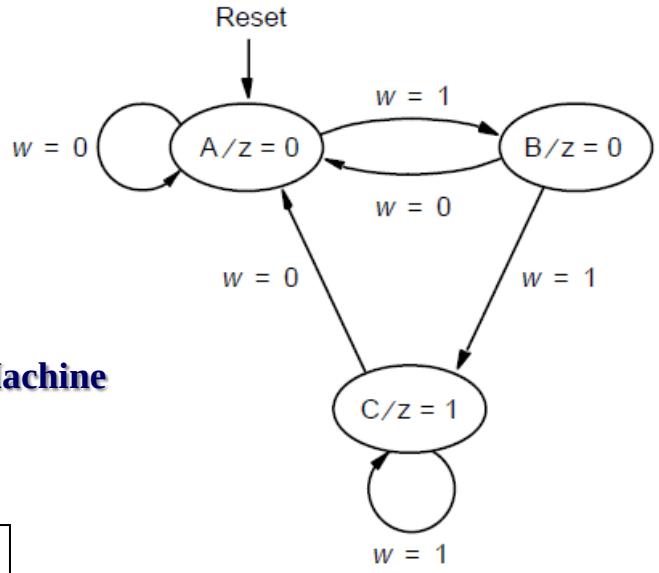
3. Máquina de Moore

Finite State Machine: Moore Machine

Moore machine is a finite-state machine
whose output values are determined solely by its current state.
The Moore machine name is related to his creator Edward F. Moore.



Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	A	B	0
B	A	C	0
C	A	C	1



Moore Machine

State diagram of a simple sequential circuit.

Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

Exemplo:
Ao encontrar dois 1s (11) gera 1 na saída, 0 no resto

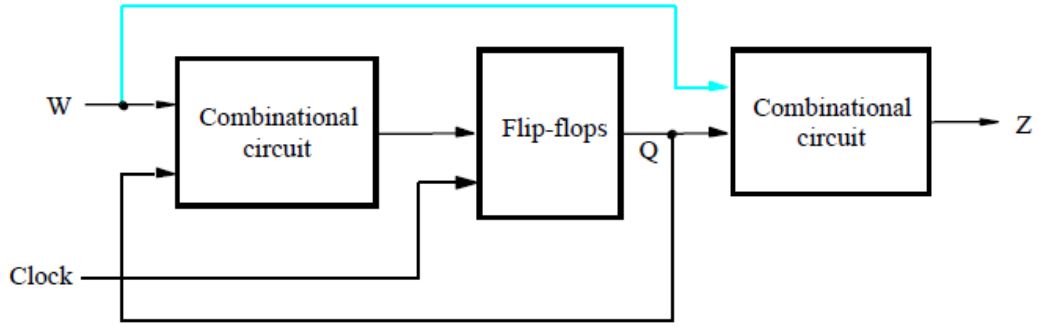
4. Máquina de Mealy

Finite State Machine: Mealy Machine

Mealy machine is a finite-state machine, whose output values are determined both by its current state and by the values of its inputs.
The Mealy machine name is related to his creator George H. Mealy.

Exemplo:
Ao encontrar dois 1s (11) gera 1 na saída imediatamente, gera 0 nos casos restantes

Mealy Machine



Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	1	0	0	1	1	0	0

Mealy Machine

Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

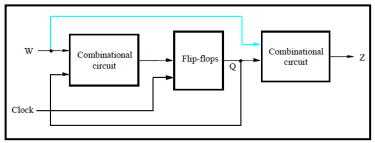
Moore Machine

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – Chapter 8)

4. Máquina de Mealy

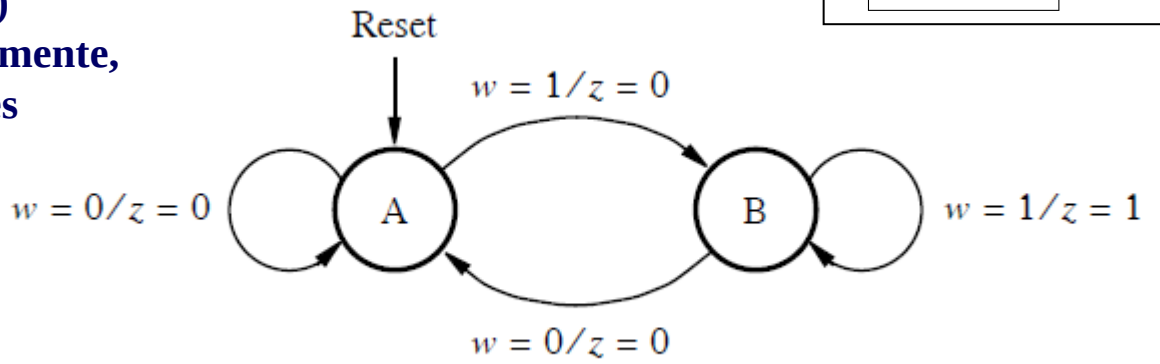
Finite State Machine: Mealy Machine

Mealy machine is a finite-state machine, whose output values are determined both by its current state and by the values of its inputs.
The Mealy machine name is related to his creator George H. Mealy.



Exemplo:
Ao encontrar dois 1s (11)
gera 1 na saída imediatamente,
gera 0 nos casos restantes

Mealy Machine



Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	1	0	0	1	1	0	0

Mealy Machine

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – Chapter 8)

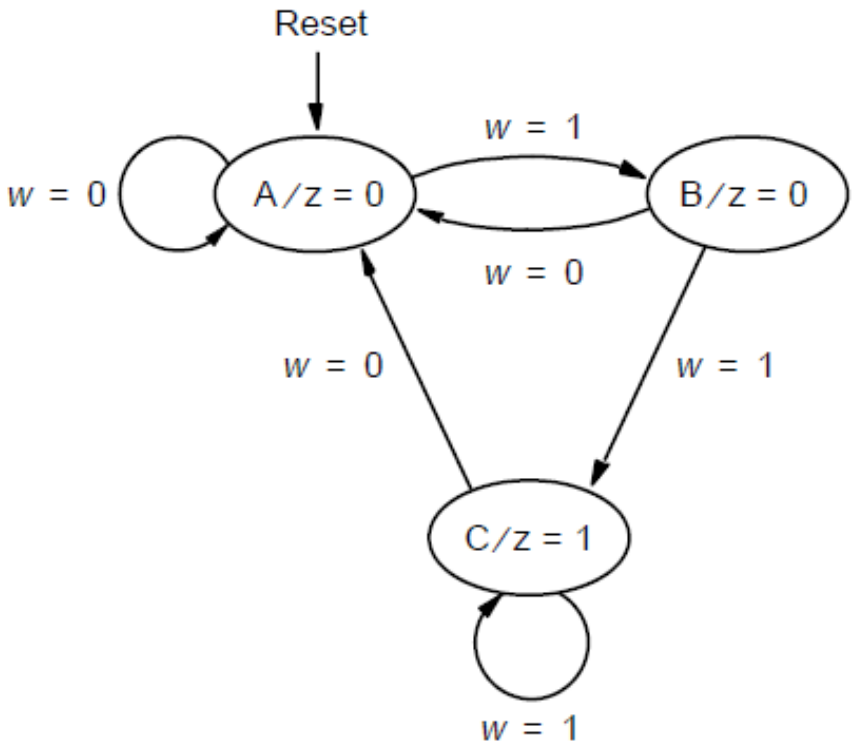
5. Finite State Machines - VHDL

FSM: Moore

VHDL

Moore

Machine



Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	A	B	0
B	A	C	0
C	A	C	1

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – VHDL Ref.)

5. Finite State Machines - VHDL

FSM:

VHDL

Moore

Machine

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

ENTITY moore IS
    PORT (     Clock : IN STD_LOGIC ;
                   w : IN STD_LOGIC ;
              Resetn : IN STD_LOGIC ;
                   z : OUT STD_LOGIC ) ;
END moore ;

ARCHITECTURE Behavior OF moore IS
    TYPE State_type IS (A, B, C) ;
    SIGNAL y : State_type ;
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
    (…)
    END PROCESS ;
    z <= '1' WHEN y = C ELSE '0' ;
END Behavior ;
```


Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – VHDL Ref.)

FSM:

VHDL

Moore

Machine

```
ARCHITECTURE Behavior OF moore IS
  TYPE State_type IS (A, B, C) ;
  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 <= C ;
                       END IF ;
                    WHEN C =>
                       IF w = '0'     THEN y <= A ;
                                      ELSE y <= C ;
                       END IF ;
            END CASE ;
    END IF ;
  END PROCESS ;

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

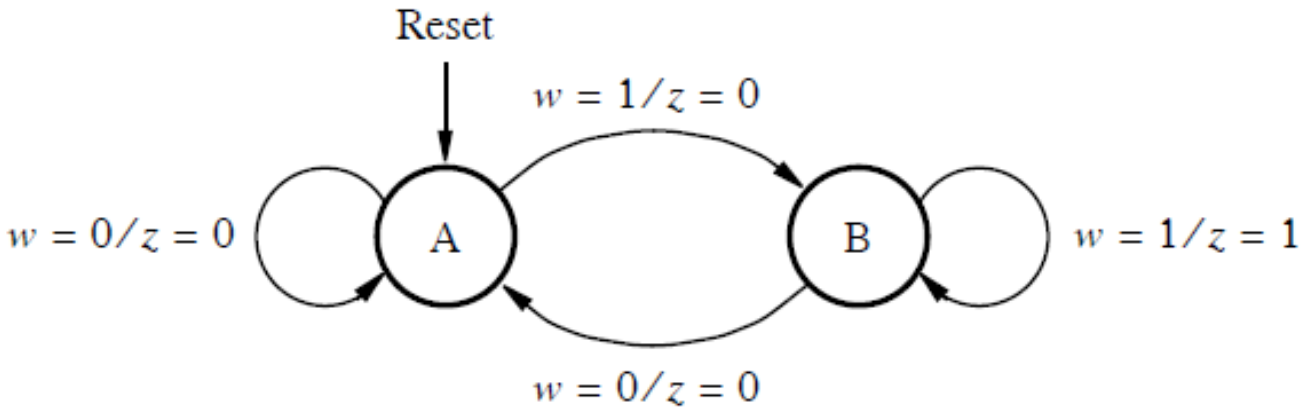
5. Finite State Machines - VHDL

FSM: Mealy

VHDL

Mealy

Machine



Present state	Next state		Output z	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	A	B	0	0
B	A	B	0	1

Figure 8.24 State table for the FSM in Figure 8.23.

	Present state	Next state		Output	
		$w = 0$	$w = 1$	$w = 0$	$w = 1$
	y	Y	Y	z	z
A	0	0	1	0	0
B	1	0	1	0	1

Figure 8.25 State-assigned table for the FSM in Figure 8.24.

Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – VHDL Ref.)

5. Finite State Machines - VHDL

FSM:

VHDL

Mealy

Machine

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

ENTITY mealy IS
    PORT ( Clock, Resetn : IN STD_LOGIC ;
           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
    (...)
    END PROCESS ;

    PROCESS ( y, w )
    BEGIN
    (...)
    END PROCESS ;
END Behavior ;
```


Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – VHDL Ref.)

FSM:

VHDL

Mealy

Machine

```
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 ;

    PROCESS ( y, w )
    BEGIN
    (...)
    END PROCESS ;
END Behavior ;
```

5. Finite State Machines - VHDL

FSM:

VHDL

Mealy

Machine

```
ARCHITECTURE Behavior OF mealy IS
    TYPE State_type IS (A, B) ;
    SIGNAL y : State_type ;
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
    (…)
    END PROCESS ;

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


Extraído de: "Fundamentals of Digital Logic with VHDL Design" – Stephen Brown and Zvonko Vranesic (B&V Book – VHDL Ref.)

AULA 04

VHDL: FSM - REFERENCIAS

SITES:

VHDL Reference => http://osorio.wait4.org/SSC0113/VHDL/ (Livro B&V)

AULAS Teóricas => SSC0113 (Bonato, Simões)

Finite State Machines: Wikipedia
http://en.wikipedia.org/wiki/Finite-state machine

EXEMPLOS FSM... Ver material complementar Aula 04

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

LRM – Laboratório de Robótica Móvel
Web LRM: Http://lrm.icmc.usp.br/
Página pessoal: Http://www.icmc.usp.br/ ~fosorio/
E-mail: fosorio [at] { icmc. usp. br , gmail. com } – F.Osório
E-mail: diogosoc [at] { icmc. usp. br } - Diogo Correa (PAE)

Disciplina de Laboratório de Elementos de Lógica Digital II [LELD2]
Web Disciplinas: Http://www.icmc.usp.br/~fosorio/
Web Wiki: http://wiki.icmc.usp.br/index.php/SSC-113-2012(fosorio)
> Programa, Material de Aulas, Critérios de Avaliação,
> Material de Apoio, Trabalhos Práticos