

**CAD 2**

**Assignment# 1 #1**

Made by: Ahmed Mohamed Hussien

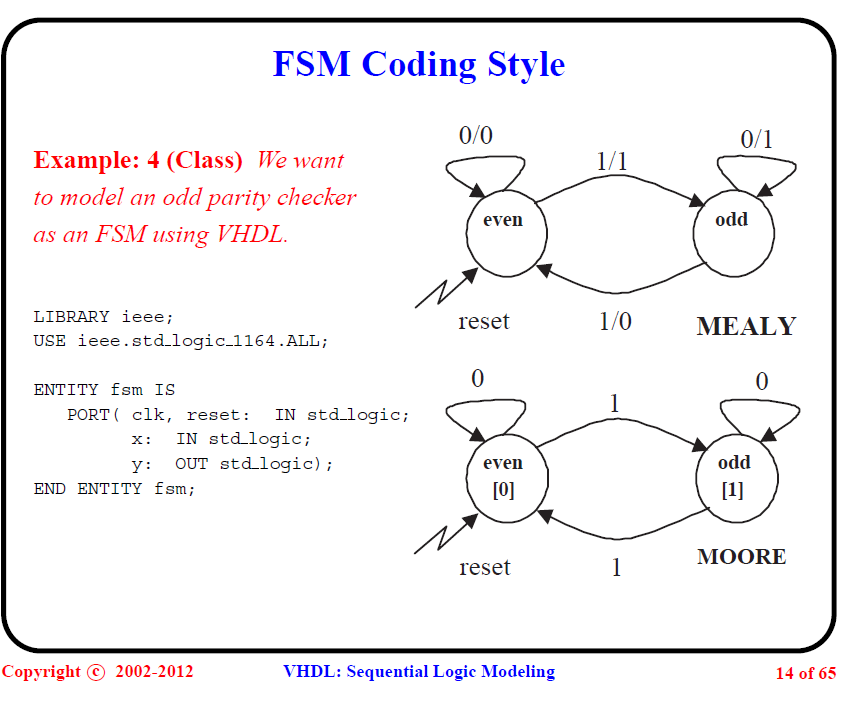
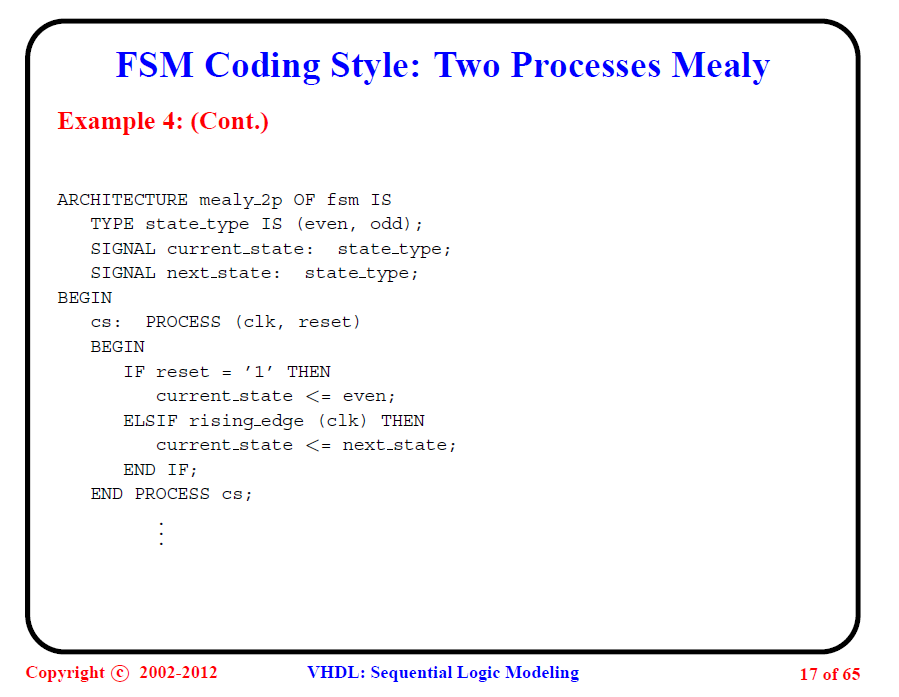
Department: Computer

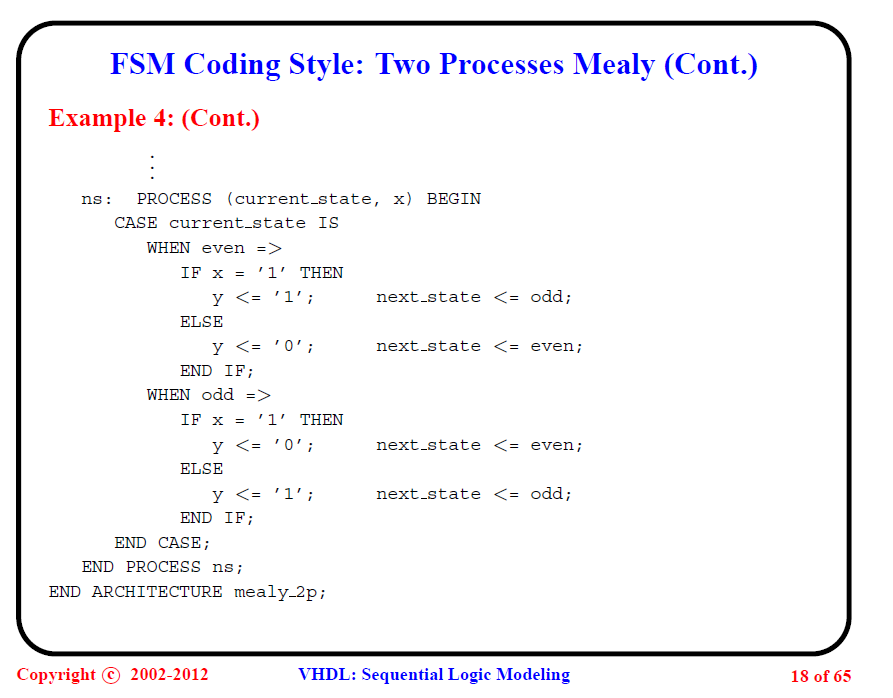
Masters ID: 10049

Submitted to: Dr. Mohamed Dessouky

# Example 65 (fsm\_mealy\_2p)

## Fsm\_mealy\_2p





## 2- Test strategy

Table will show all the test cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tested features | Inputs | | | | Delay after | Expected output | |
|  | Clk | reset | IN | CURRENT STATE |  | NEXT STATE | OUT |
|  | 0 | 1 | - | EVEN | 20 ns | - | - |
|  | 1 | 0 | - | EVEN | 20 ns | - | - |
| no change when input is 0 and state is even | 0,1 | 0 | 0 | EVEN | 40 ns | EVEN | 0 |
| Changing the state will change the output | 0 | 0 | 1 | EVEN | 20 ns | ODD | 1 |
| 1 | 0 | 1 | ODD | 20 ns | EVEN | 0 |
| no change when input is 0 and state is ODD | 0,1 | 0 | 0 | ODD | 40 ns | ODD | 1 |
| Changing the state will change the output | 0 | 0 | 1 | ODD | 20 ns | EVEN | 0 |
| 1 | 0 | 1 | EVEN | 20 ns | ODD | 1 |

0,1 mean that its 0 for half the period and 1 for the other half it will not change so it’s done like this

At the beginning set **reset** to **1** and to make sure you are in the **EVEN STATE** then set the **reset** to **0** then apply **IN** with the values shown in the above table in order asserting the **NEXT STATE** and **OUT.**

The mealy state machine output depend on the state and the input and the state only change at the positive clock edge (Clk change from 0 to 1 )

## 3-Testbench code

**LIBRARY** ieee**;**

**USE** ieee**.**std\_logic\_1164**.ALL;**

**entity** FSM\_tb **is** -- test bench entity

**end** **entity** FSM\_tb**;**

**architecture** test\_bench **of** FSM\_tb **is** --architecture of the test bench

**component** fsm **is** -- defining the component FSM\_mealy\_2p

**port(** clk**,** reset**:** **IN** std\_logic**;**

x**:** **IN** std\_logic**;**

y**:** **OUT** std\_logic **);**

**end** **component** fsm**;**

**for** DUT**:** fsm **use** **entity** work**.**fsm **(**mealy\_2p**);** -- getting the component definition into our design

**signal** reset**,**x**,**y **:** std\_logic**;** -- wires that will be connected to our component

**signal** clk **:** std\_logic**:=** '0'**;**

**constant** clk\_period **:** time **:=** 40 ns**;** -- signals we can look at in the simulation

**begin**

DUT**:** fsm **port** **map** **(**clk**,**reset**,**x**,**y**);**

clk\_l**:** **process** **is** --producing clk with 40 ns period

**begin**

**wait** **for** clk\_period**/**2 **;**

clk **<=** not clk **;**

**end** **process** clk\_l**;**

PD**:** **process** **is**

**begin**

reset **<=** '1'**;** **wait** **for** 20 ns**;** --state is even

reset **<=** '0'**;** **wait** **for** 20 ns**;**

x **<=** '0'**;** **wait** **for** 40 ns**;** --state dosent change

**assert** y **=** '0'

**report** "problem the output change "

**severity** error**;**

x **<=** '1'**;** **wait** **for** 40 ns**;** --state will be odd at the clk edge

**assert** y **=** '1' -- this is not always true this is true for the even state only

**report** "problem the output change "

**severity** error**;**

x **<=** '0'**;** **wait** **for** 40 ns**;** --state dosent change

**assert** y **=** '1'

**report** "problem the output change "

**severity** error**;**

x **<=** '1'**;** **wait** **for** 40 ns**;** --state will be even at the clk edge

**assert** y **=** '0' -- this is true only for the odd state changing to even will give an error

**report** "problem the output change "

**severity** error**;**

-- total time needed for full simulation is 4\*40+20+10 ns= 190 ns

**wait;**

**end** **process** PD**;**

**end** **architecture** test\_bench**;**

## 4- Simulation Result



The circuit works as expected when input is 0 the state didn’t change when the input is one the state change at the clock edge. On the other hand the output change with the input change on the transition and not at the clock edge (mealy state machine).

## 5- Used tools

ModelSim is used for testbench simulation