

## Lecture 3 REVIEW - Sequential Logic Design

---

---

---

---

---

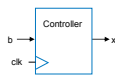
---

---

---

### Sequential Circuit Design (Controllers)

- Sequential Circuits
  - Output based on inputs as well as previous inputs (state)
  - Previously learned how to store data which can store "state" of circuit (Registers)
- Two categories of sequential circuits
  - Synchronous – clock used to control operation of circuit
  - Asynchronous – no clock
- We'll focus on *synchronous sequential circuits*



A sequential circuit that controls Boolean outputs based on Boolean inputs and a specific time-ordered behavior is often called a **controller**.

---

---

---

---

---

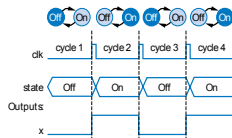
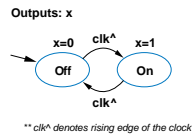
---

---

---

### Describing Behavior of Sequential Circuit: FSM

- State Diagram or Finite-State Machine (FSM)
  - A way to describe desired behavior of sequential circuit
  - List states, and transitions among states
- Example: Make  $x$  toggle (0 to 1, or 1 to 0) every clock cycle
  - Two states
    - Off ( $x=0$ )
    - On ( $x=1$ )
  - Transition from Off to On, or On to Off, on rising clock edge
  - Arrow with no starting state points to initial state (when circuit first starts)




---

---

---

---

---

---

---

---





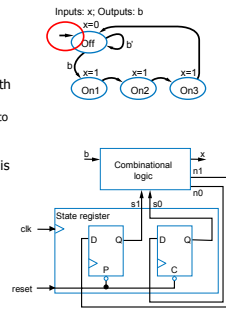






## Initial State of a Controller

- All our FSMs had initial state
  - But our sequential circuit designs did not
- Can accomplish using flip-flops with preset/clear inputs
  - Shown circuit initializes flip-flops to 01
- Designer must ensure reset input is 1 during power up of circuit
  - By electronic circuit design




---

---

---

---

---

---

---

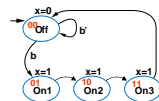
---

---

---

## Representations of Finite State Machines

- Different ways to represent same functionality
  - Graphical
  - State Table
  - Formal
- Able to convert among different representations



Inputs		Outputs			
$s_1$	$s_0$	$b$	$n_1$	$n_0$	$x$
Off	0	0	0	0	0
Off	0	0	1	0	1
On1	0	1	0	1	0
On1	0	1	1	1	0
On2	1	0	0	1	1
On2	1	0	1	1	1
On3	1	1	0	0	0
On3	1	1	1	0	0

---

---

---

---

---

---

---

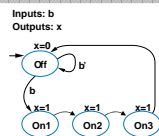
---

---

---

## FSM Formal Definition

- FSM defined by quintuple
  - $M = (X, Y, S, \delta, \lambda, s_0)$
  - $X$  is the input alphabet
  - $Y$  is the output alphabet
  - $S$  is a finite set of states
  - $\delta$  is the transition function,  $\delta: X \times S \rightarrow S$ 
    - Given input and state, what is the next state
  - $\lambda$  is the output function,  $\lambda: S \rightarrow Y$ 
    - Mealy FSM,  $\lambda: X \times S \rightarrow Y$
  - $s_0$  is the initial state




---

---

---

---

---

---

---

---

---

---

## FSM Formal Definition – Example 1

- Formally specify the Laser Timer FSM
  - $M = (X, Y, S, \delta, \lambda, s_0)$

LaserTimer =  $(X, Y, S, \delta, \lambda, q_0)$ , where

$X = \{0, 1\}$

$Y = \{0, 1\}$

$S = \{Off, On1, On2, On3\}$

$\delta(Off, 0) = Off, \delta(Off, 1) = On1$

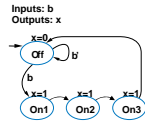
$\delta(On1, 0) = On2, \delta(On1, 1) = On2$

$\delta(On2, 0) = On3, \delta(On2, 1) = On3$

$\delta(On3, 0) = Off, \delta(On3, 1) = Off$

$\lambda(Off) = 0, \lambda(On1) = 1, \lambda(On2) = 1, \lambda(On3) = 1$

$s_0 = Off$



X is the input alphabet

Y is the output alphabet

S is a finite set of states

$\delta$  is the transition function,  $\delta: X \times S \rightarrow S$   
Given and input and state, what is the next state

$\lambda$  is the output function,  $\lambda: S \rightarrow Y$

$s_0$  is the initial state

---

---

---

---

---

---

---

---

---

---

## FSM Formal Definition - Example 2

- Given formal specification of ThreeOnes FSM convert to graphical representation

ThreeOnes =  $(X, Y, S, \delta, \lambda, q_0)$ , where

$X = \{0, 1\}$

$Y = \{0, 1\}$

$S = \{Zero, One, Two, Three\}$

$\delta(Zero, 0) = Zero, \delta(Zero, 1) = One$

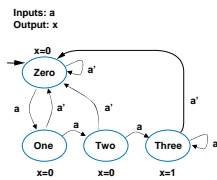
$\delta(One, 0) = Zero, \delta(One, 1) = Two$

$\delta(Two, 0) = Zero, \delta(Two, 1) = Three$

$\delta(Three, 0) = Zero, \delta(Three, 1) = Three$

$\lambda(Zero) = 0, \lambda(One) = 0, \lambda(Two) = 0, \lambda(Three) = 1$

$q_0 = Zero$




---

---

---

---

---

---

---

---

---

---

## Mealy vs. Moore FSM

- Previously associated output with current state
  - Moore FSM
- Another type associates output with both the state and the FSM input (transitions)
  - Mealy FSM

- Example: Soda dispenser

- Input

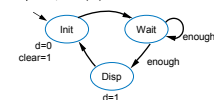
- enough* - indicates when sufficient money deposited

- Output

- d'* - releases a soda
- clear* - zeros device counting money deposited

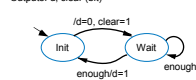
### Moore FSM

Inputs: enough (bit)  
Outputs: d, clear (bit)



### Mealy FSM

Inputs: enough (bit)  
Outputs: d, clear (bit)



Outputs not explicitly assigned on a transition are implicitly assigned a 0. Assignment to 0 still listed if it is key to the FSM's behavior

---

---

---

---

---

---

---

---

---

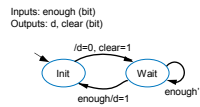
---





## Implementing a Mealy FSM

- Implementing a Mealy FSM
  - Same steps as a Moore FSM
    - Capture the FSM
    - Create the architecture
    - Encode the states
    - Create the state table
    - Derive equations for each output, implement the combinational logic
  - Key difference from Moore
    - FSM output are based on state and FSM inputs
    - May have different value in same state, depending on input values




---

---

---

---

---

---

---

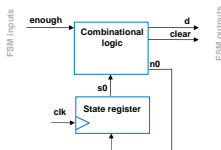
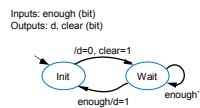
---

---

---

## Implementing a Mealy FSM Example 1

- Capture the FSM
  - Done!
- Create the architecture
  - Inputs – enough
  - Outputs – d, clear
  - State register – 2 states, 1-bit




---

---

---

---

---

---

---

---

---

---

## Implementing a Mealy FSM Example 1 (cont')

- Encode the states
  - Use minimum-width binary encoding
- Create the state table
  - This is where it differs from a Moore implementation
    - FSM output are based on state and FSM inputs
    - May have different value in same state, depending on input values



	Inputs		Outputs		
	s0	enough	n0	d	clear
Init	0	0	1	0	1
	0	1	1	0	1
Wait	1	0	1	0	0
	1	1	0	1	0

---

---

---

---

---

---

---

---

---

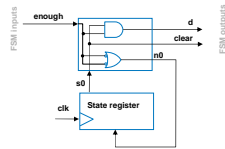
---

## Implementing a Mealy FSM Example 1 (cont')

- Implement the combinational logic

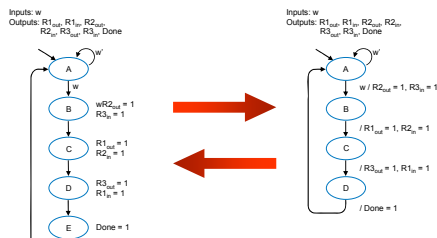
- $n0 = s0' + \text{enough}'$
- $d = s0 \cdot \text{enough}$
- $\text{clear} = s0$

	Inputs		Outputs	
	s0	enough	n0	d
Init	0	0	1	0
Wait	0	1	1	0
	1	0	1	0
	1	1	0	1



## Converting Between Mealy and Moore FSMs

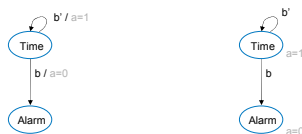
- Given a Mealy FSM we can convert to a Moore FSM
- Given a Moore FSM we can convert to a Mealy FSM



## Converting a Mealy to a Moore FSMs

### Template 1

- Converting a Mealy to a Moore FSM



#### State Time

What happens when  $b = 0$ ?

go to state time  
output  $a = 1$  (make this associated with state)

What happens when  $b = 1$ ?

go to state alarm  
output  $a = 0$  (make this associated with state)

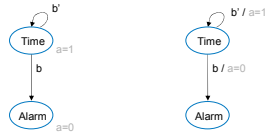




## Converting a Moore to a Mealy FSMs

Template 1

- Converting a Moore to a Mealy FSM



### State Time

What happens when  $b = 0$ ?

go to state time  
state time outputs  $a = 1$  (make this output associated with the transition)

What happens when  $b = 1$ ?

go to state alarm  
state alarm outputs  $a = 0$  (make this output associated with the transition)

Digital Design  
Copyright © 2006  
Frank Vahid

ECE 474a/575a  
Susan Lysdecky

43 of 48

---

---

---

---

---

---

---

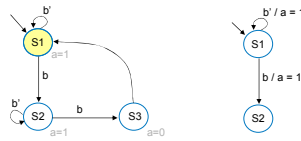
---

---

---

## Converting a Moore to a Mealy FSMs Example

- Converting the following Moore FSM to a Mealy FSM



### State S1

What happens when  $b$ ?

go to state S1  
output  $a = 1$

What happens when  $b$ ?

go to state S2  
output  $a = 1$

Digital Design  
Copyright © 2006  
Frank Vahid

ECE 474a/575a  
Susan Lysdecky

44 of 48

---

---

---

---

---

---

---

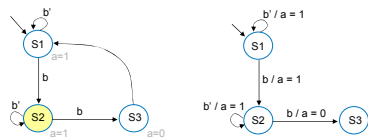
---

---

---

## Converting a Moore to a Mealy FSMs Example (cont')

- Converting the following Moore FSM to a Mealy FSM



### State S2

What happens when  $b$ ?

go to state S2  
output  $a = 1$

What happens when  $b$ ?

go to state S3  
output  $a = 0$

Digital Design  
Copyright © 2006  
Frank Vahid

ECE 474a/575a  
Susan Lysdecky

45 of 48

---

---

---

---

---

---

---

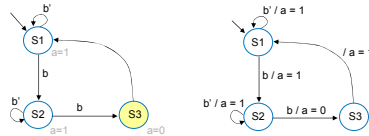
---

---

---

## Converting a Moore to a Mealy FSMs Example (cont')

- Converting the following Moore FSM to a Mealy FSM



### State S3

We don't care what input value b is equal to  
 go to state S1  
 output a = 1

---

---

---

---

---

---

---

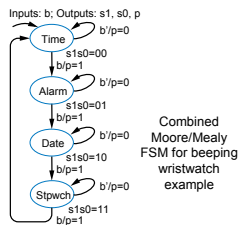
---

---

---

## Mealy and Moore FSMs Can be Combined

- Final note on Mealy/Moore
  - May be combined in same FSM




---

---

---

---

---

---

---

---

---

---

## Summary

- FSM Representations
  - Graphical – State Diagram (states & transitions)
  - Text - State Table
  - Formal Model,  $M = (X, Y, S, \delta, \lambda, s_0)$
- Controller design process
  - FSM specification to hardware
- Moore vs. Mealy FSM
  - Specification
  - Timing
  - Conversion between Moore and Mealy

---

---

---

---

---

---

---

---

---

---