EECE 2322: Fundamentals of Digital Design and Computer Organization Lecture 11 1: Finite State Machine

Xiaolin Xu Department of ECE Northeastern University

Midterm2

- * March 31, usual class time 1:30-2:45PM
 - * Allowing 5mins for downloading and uploading
- * 4 Questions in total

Midterm2

* Topics:

- MIPS instruction basics
 - E.g., what does a line of assembly code do, and its results?
 - Pseudo-instruction to "real" MIPS instructions (non-pseudo)
- * Conversion in between C code and MIPS assembly code (slides)
 - * Fill in the missing parts with given register names

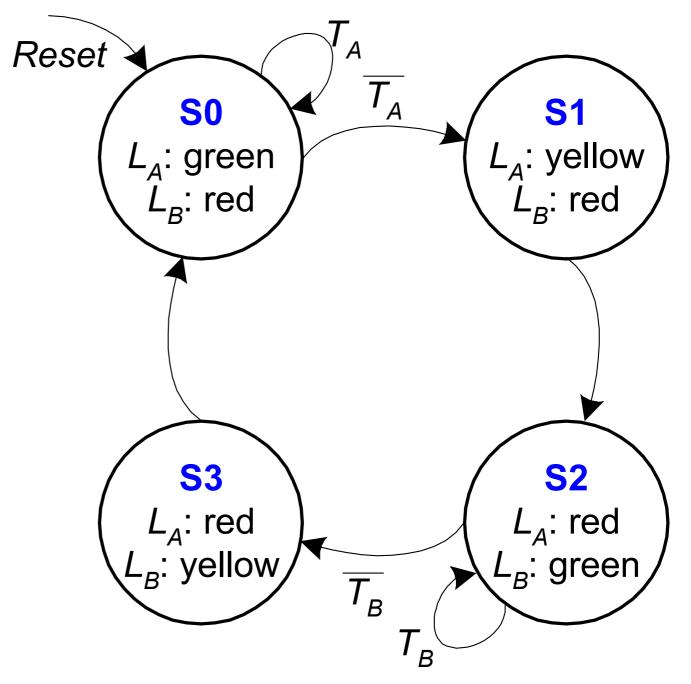
Midterm2

* Topics:

- * For a given code snippet
 - Straightforward conversion
 - Caller and Calle involved (stack!)
- * FSM
 - Design an FSM for a pre-defined function, including state transition table and circuit implementation

FSM State Transition Diagram

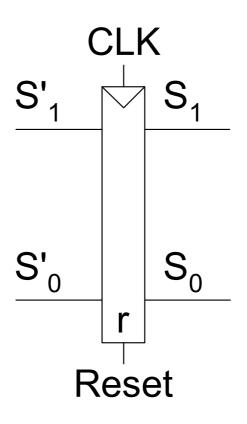
- Moore FSM: outputs labeled in each state
- States: Circles
- Transitions: Arcs



FSM Schematic: State Register

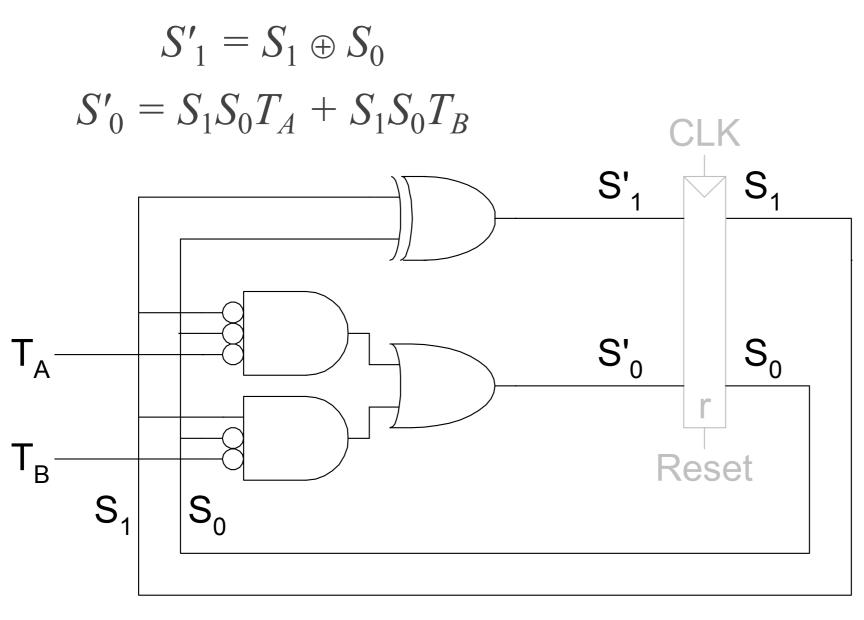
$$S'_1 = S_1 \oplus S_0$$

$$S'_0 = \overline{S_1} \overline{S_0} \overline{T_A} + S_1 \overline{S_0} \overline{T_B}$$



state register

FSM Schematic: Next State Logic



$$L_{A1} = S_1$$

$$L_{A0} = \overline{S}_1 S_0$$

$$L_{B1} = \overline{S}_1$$

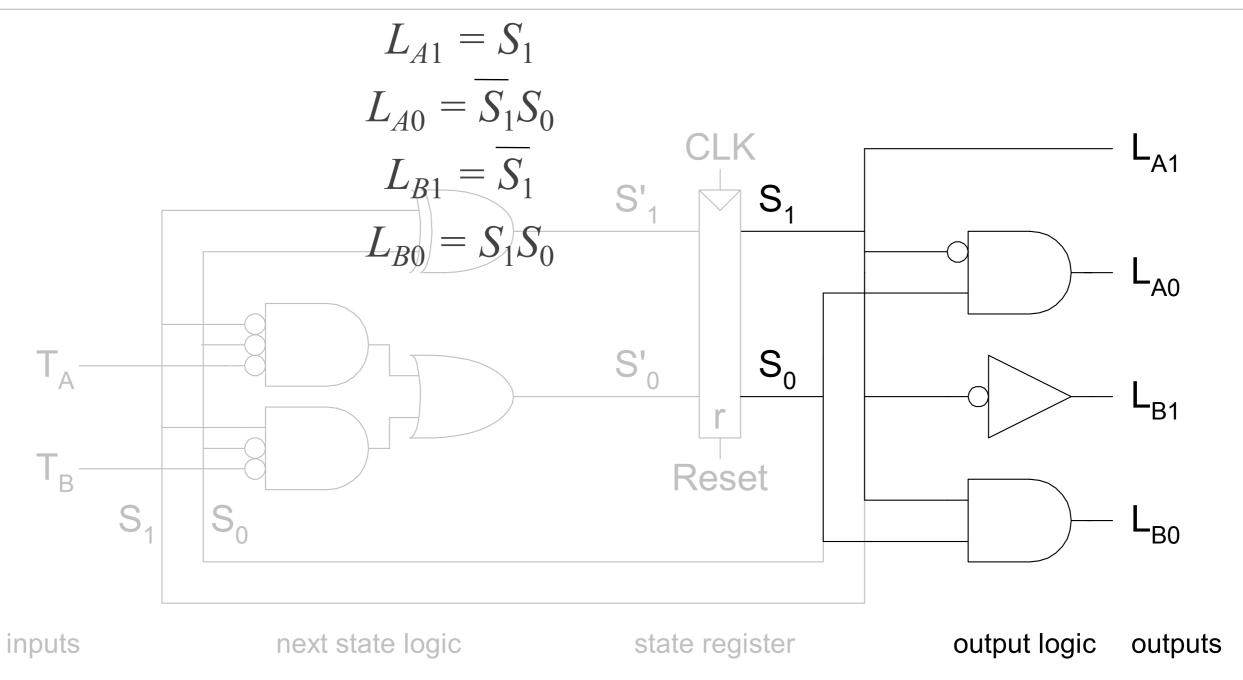
$$L_{B0} = S_1 S_0$$

inputs

next state logic

state register

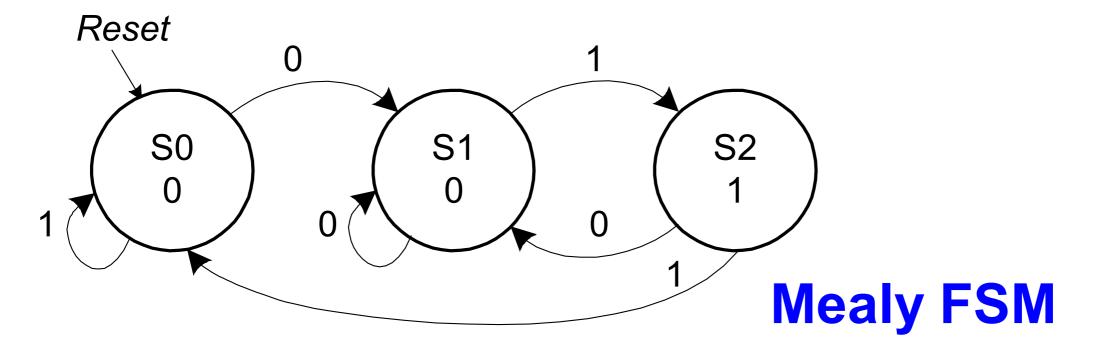
FSM Schematic: Output Logic

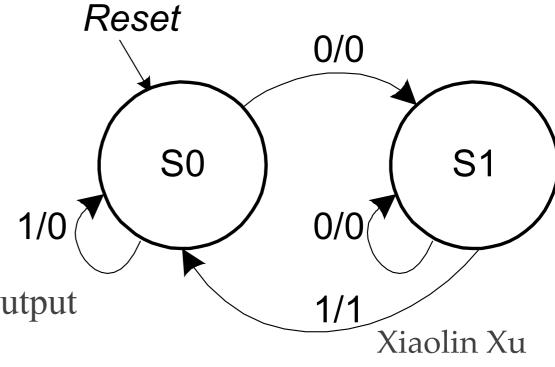


EECE2322 Xiaolin Xu

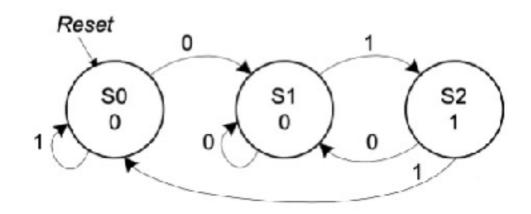
State Transition Diagrams

Moore FSM



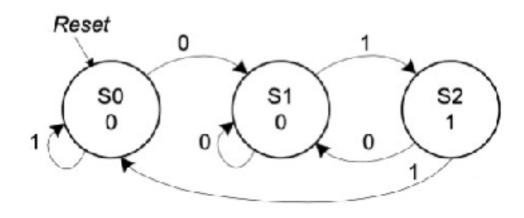


Mealy FSM: arcs indicate input/output



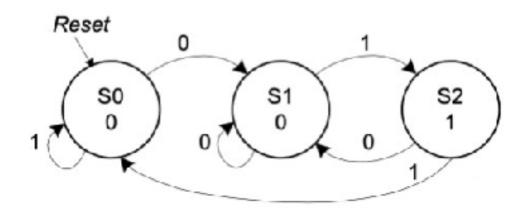
Current		Inputs	Next State	
S_1	S_0	A	<i>S</i> ′ ₁	S'_0
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		

State	Encoding
S0	00
S 1	01
S2	10



Cur	rent	Inputs	Next	State
S_1	S_0	A	<i>S</i> ′ ₁	S'_0
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0

State	Encoding
S0	00
S 1	01
S2	10



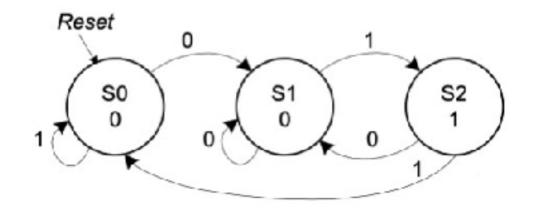
$$S_1' = S_0 A$$
$$S_0' = \overline{A}$$

Cur	rent	Inputs	Next	State
S_1	S_0	A	<i>S</i> ′ ₁	S'_0
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0

State	Encoding
S0	00
S 1	01
S2	10

Moore FSM Output Table

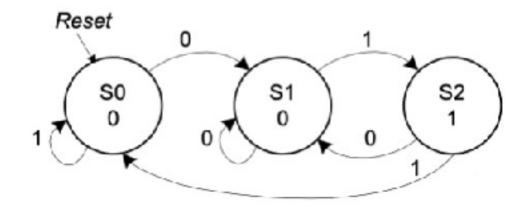
Current State		Output
S_1	S_0	Y
0	0	0
0	1	0
1	0	1



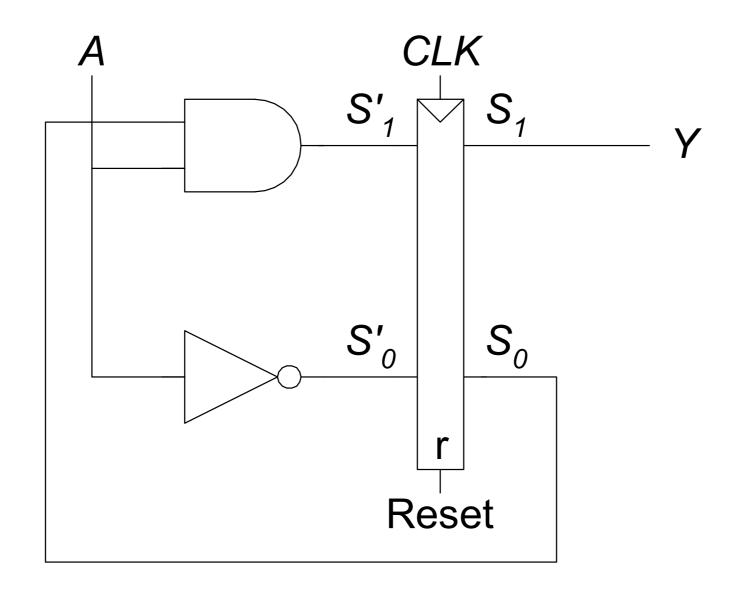
Moore FSM Output Table

Current State		Output
S_1	S_0	Y
0	0	0
0	1	0
1	0	1

$$Y = S_1$$



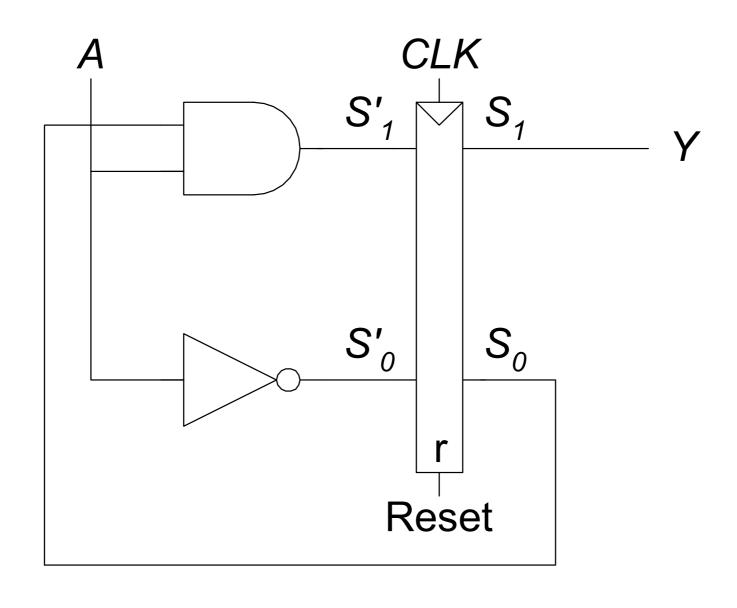
Moore FSM Schematic



$$S_1' = S_0 A$$

 $S_0' = \overline{A}$

Moore FSM Schematic



$$S_1' = S_0 A$$

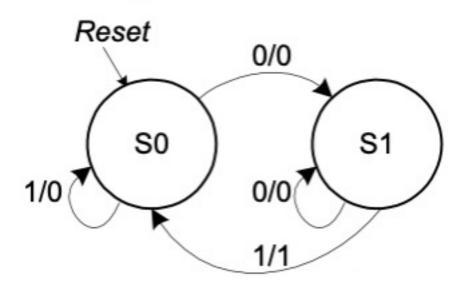
 $S_0' = \overline{A}$
 $Y = S_1$

Mealy FSM State Transition & Output Table

Current	Input	Next	Output
S_0	A	S'_0	Y
0	0		
0	1		
1	0		
1	1		

State	Encoding
S0	00
S 1	01

Mealy FSM

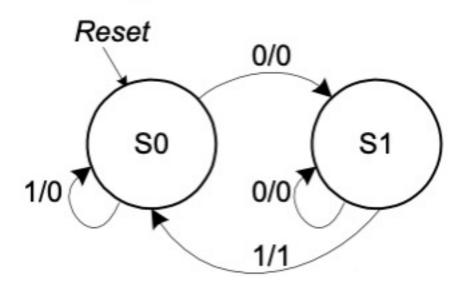


Mealy FSM State Transition & Output Table

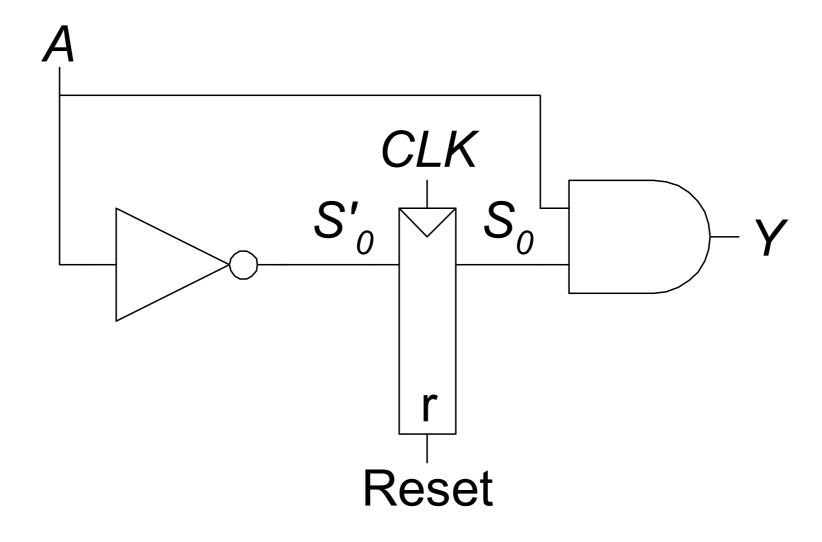
Current	Input	Next	Output
S_0	A	S'_0	Y
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	1

State	Encoding
S0	0
S 1	1

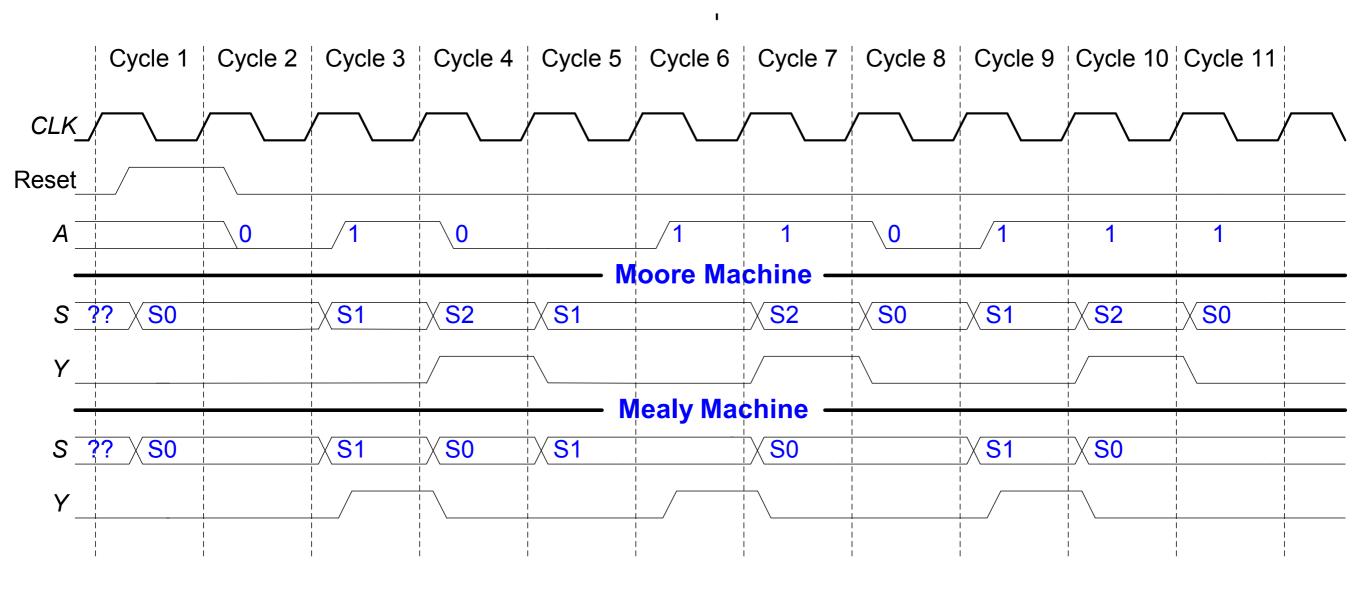
Mealy FSM



Mealy FSM Schematic



Moore & Mealy Timing Diagram

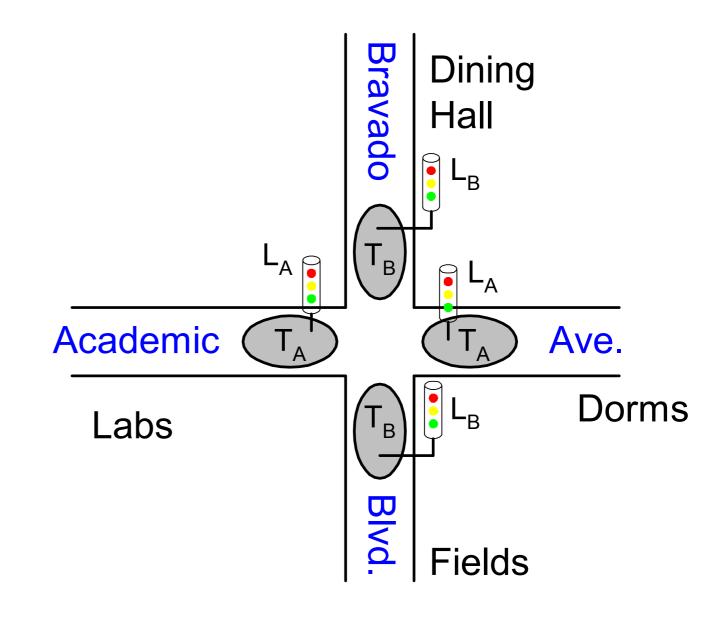


Practical Finite State Machine Design

- * A large design space, i.e., too many states
- * Factoring: Break complex FSMs into smaller interacting FSMs
 - Into two or more simpler ones
 - Greatly simplify the design of a state machine by separating orthogonal aspects
 - Separate FSMs can be handled independently

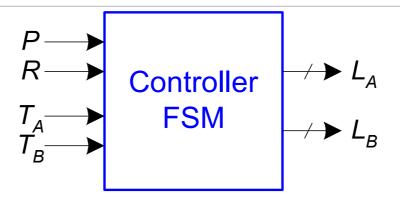
Factoring State Machines

- * Example: Modify traffic light controller to have Parade Mode.
 - * Two more inputs: *P*, *R*
 - When P = 1, enter
 Parade Mode &
 Bravado Blvd light
 stays green
 - When *R* = 1, leaveParade Mode

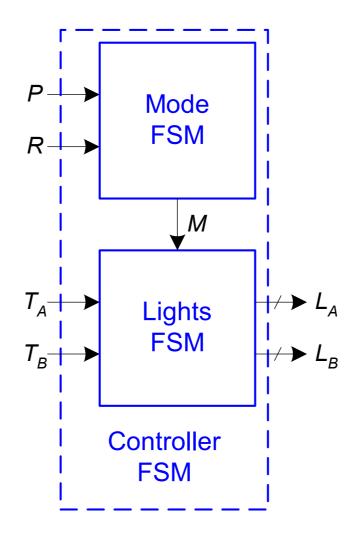


Parade FSM

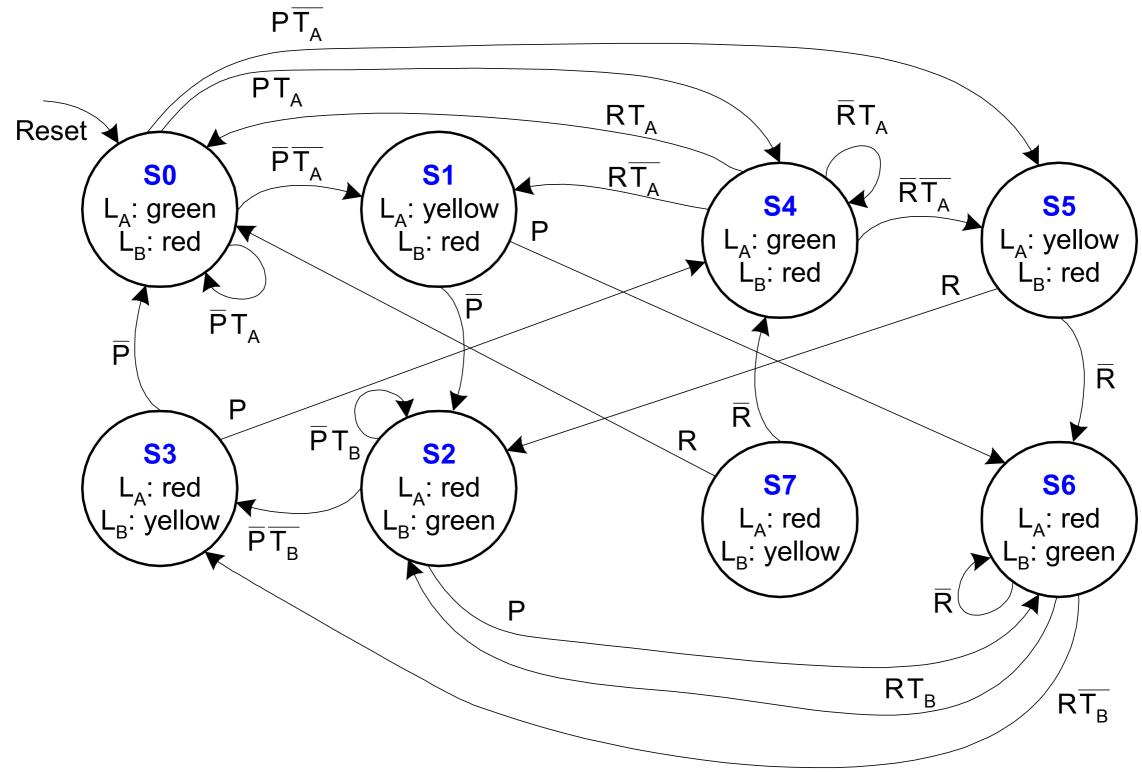
Unfactored FSM



Factored FSM



Unfactored FSM

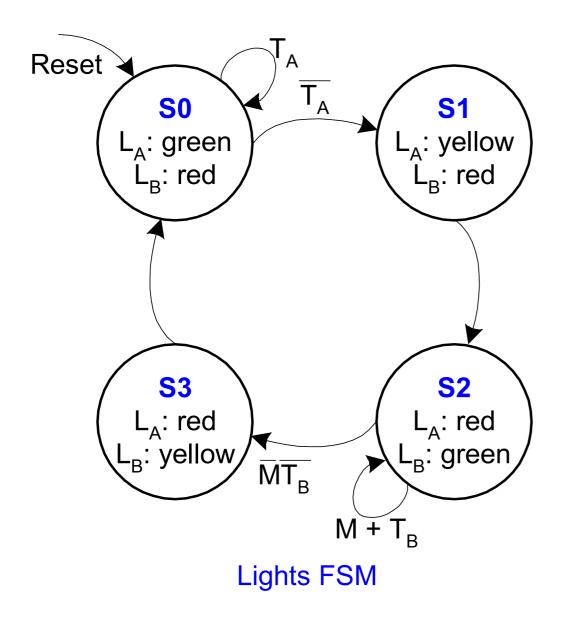


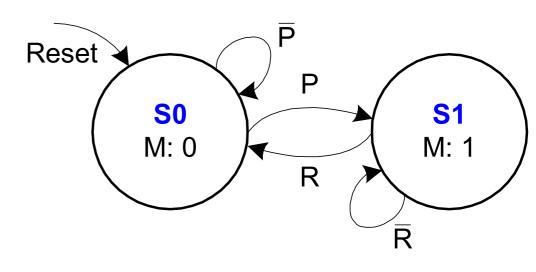
21

EECE2322

Xiaolin Xu

Factored FSM

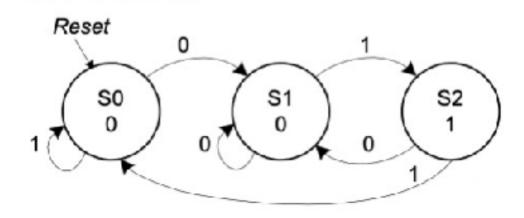




Mode FSM

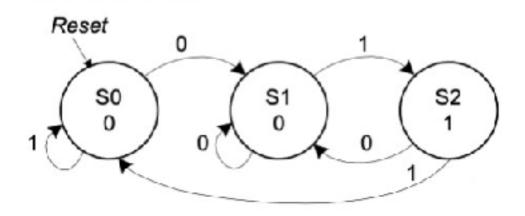
FSM Design Procedure

- Identify inputs and outputs
- * Sketch state transition diagram
- * Write state transition table
- * Select state encodings
- * For Moore machine:
 - Rewrite state transition table with state encodings
 - Write output table



Cur	rent	Inputs	Next State			
S_1	S_0	A	<i>S</i> ′ ₁	S'_0		
0	0	0	0	1		
0	0	1	0	0		
0	1	0	0	1		
0	1	1	1	0		
1	0	0	0	1		
1	0	1	0	0		

State	Encoding
S0	00
S 1	01
S2	10



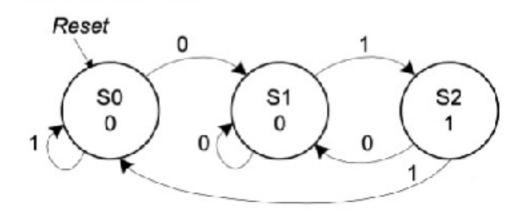
$$S_1' = S_0 A$$
$$S_0' = \overline{A}$$

Cur	rent	Inputs	Next State			
S_1	S_0	A	<i>S</i> ′ ₁	S'_0		
0	0	0	0	1		
0	0	1	0	0		
0	1	0	0	1		
0	1	1	1	0		
1	0	0	0	1		
1	0	1	0	0		

State	Encoding
S0	00
S1	01
S2	10

Moore FSM Output Table

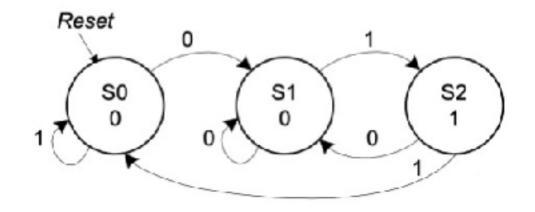
Curren	Output	
S_1	S_0	Y
0	0	0
0	1	0
1	0	1



Moore FSM Output Table

Curren	Output	
S_1	S_0	Y
0	0	0
0	1	0
1	0	1

$$Y = S_1$$



FSM Design Procedure

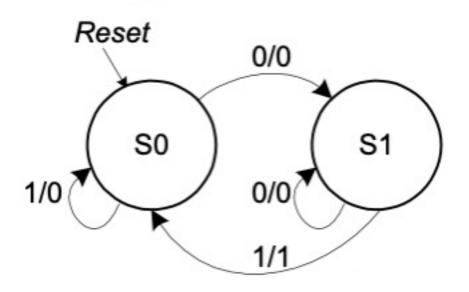
- Identify inputs and outputs
- Sketch state transition diagram
- * Write state transition table
- Select state encodings
- * For a Mealy machine:
 - Rewrite combined state transition and output table with state encodings

Mealy FSM State Transition & Output Table

Current	Input	Next	Output
S_0	A	S'_0	Y
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	1

State	Encoding
S0	0
S 1	1

Mealy FSM



Last Step

- Write Boolean equations for next state and output logic
- * Sketch the circuit schematic

A Design Example

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

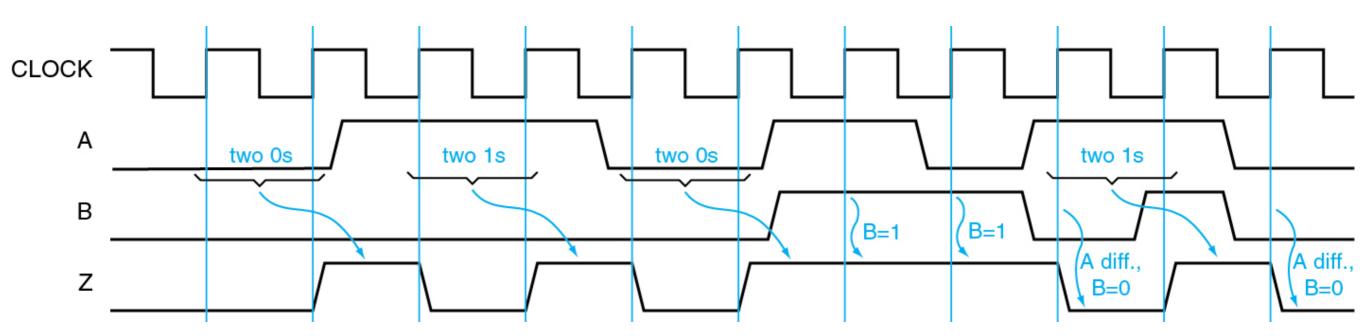
Otherwise, the output should be 0.

A Design Example

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

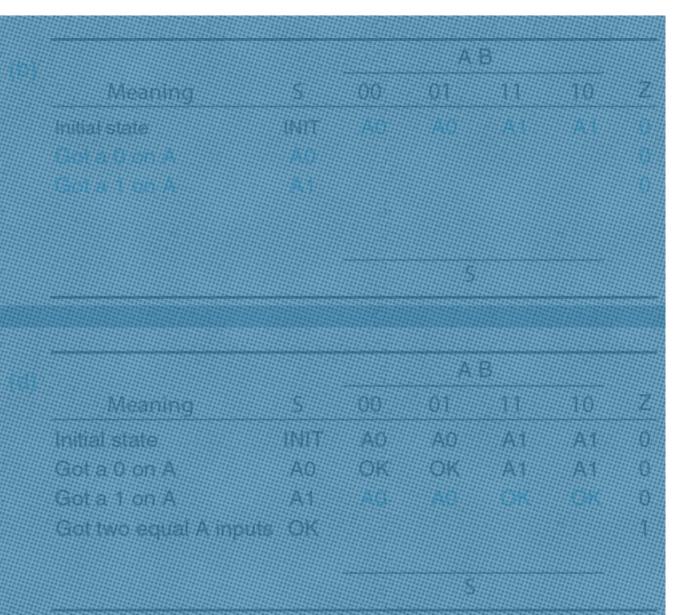
- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.



(a)				Α	В		
(4)	Meaning	S	00	01	11	10	Z
	Initial state	INIT					0
				S			

Meaning			
Initial state			
Got a 0 on A			
Got a 1 on A			



(a)				Α	В			(b)				Α	В		
(α)	Meaning	S	00	01	11	10	Z	(6)	Meaning	S	00	01	11	10	Z
	Initial state	INIT					0		Initial state	INIT	A0	A 0	A1	A1	0
									Got a 0 on A	AO					0
									Got a 1 on A	A1					0
				S								S			

Meaning			
Initial state	AO		
Got a 0 on A			
GotalonA			

Meaning		00			
Initial state			NAO!		
Got a 0 on A	AO	OK	l ok	A	
Got a 1 on A		AG			

(a)				Α	В		
(α)	Meaning	S	00	01	11	10	Z
	Initial state	INIT					0
				S			

(b)				Α	В		
(6)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A 0	A0	A1	A1	0
	Got a 0 on A	A0					0
	Got a 1 on A	A1					0
				S			

)			Α	В		
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	A1	A1	0
Got a 1 on A	A1					0
Got two equal A input	s OK					1
			S			

Meaning					
Initial state					
Got a 0 on A	A0	OK	A	A	
Got a 1 on A					
Got two equal A mouts	OK				

(b)

(a)				Α	В		
(ω)	Meaning	S	00	01	11	10	Z
	Initial state	INIT					0
				S			

			Α	В		
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0					0
Got a 1 on A	A1					0
			S			

c)			4	Α	В		
(0)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	Α0	A1	A1	0
	Got a 0 on A	A0	OK	OK	A1	A1	0
	Got a 1 on A	A1					0
	Got two equal A inpu	ts OK					1

(d)				Α	В		
(4)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	A0	A1	A1	0
	Got a 0 on A	A0	OK	OK	Α1	Α1	0
	Got a 1 on A	Α1	AO	A0	OK	OK	0
	Got two equal A inputs	OK					1
				S			

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

A had the same value at each of the two previous clock ticks, or

Meaning

B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

			Α	В		
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	Α1	Α1	0
Got a 1 on A	A1	Α0	A0	OK	OK	0
Got two equal A inp	uts OK	?	OK	OK	?	1
			S			

(a)

Meaning						
Initial state						
Got a C on A	AO	i okoli	OKO			0
Got a 1 on A			AQ	OK	OKI	0
Two equal, A=0 last	OKO					
Two equal, A=1 asi.						

hmalsate	INIT					
Got a Con A	AO				A	
Got a 1 on A		AQ.				
Meaning						
Initial state	NINIT	NAO!!!	AO II			
Got a 0 on A	AQ.	oke	OKO			
Gota 1 on A		AO		OKT	OKI	
Two equal. A=0 last	OKO	OKO	OKO	OKT	At	
Two equal, A=1 last	OKI					

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

A B						
Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK	OK	Α1	Α1	0
Got a 1 on A	Α1	A0	A0	OK	OK	0
Got two equal A inputs OK		?	OK	OK	?	1
			S			

(a)

Meaning						
Initial state			AO			
Got a 0 on A	AO	OKO	OKO	A		
Got a 1 on A		AG	ĄQ	OK	OKH	
Two equal, A=0 last	OKO					
Two equal, A=1 last						

Meaning			
Intial state			
GotalonA	NO.		
GotatonA		AO	

Initial state INIT

INVOLEDUATURE MIASURIO KA

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

(a)							
(u)	Meaning	K	00	01	11	10	Z
	Initial state	INIT	AQ	A0	Α1	A1	0
	Got a 0 on A	A0	OK/	OK	A1	Α1	0
	Got a 1 on A	A1	A0	AQ	OK	OK	0
	Got two equal A in	puts OK	? (OK	OK	?	1
				S			

Meaning					
Initial state					
Gora o on A		OKO	OKO		
Got a 1 on A		AO		OKI	OKT
Two equal, A=0 last	OKO				
Two equal, A=1 last					

Meaning		00					
initial state							
Got a 0 on A	ΑŌ						
Got a 1 on A		A0	AO				
Meaning							
Initial state		AÓ	AO				
Got a 0 on A		oke	OKO				
Got a 1 on A		AO		OKt	OKI		
Two equal, A=0 last	OKO	OKO	OKO	OKT	A		
Two equal, A=1 last							

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

			Α	В		
Meaning	K	00	01	11	10	Z
Initial state	INIT	AQ	A0	Α1	A1	0
Got a 0 on A	A0	QK,	OK	Α1	Α1	0
Got a 1 on A	A1	A0	AR	OK	OK	0
Got two equal A inpo	uts OK (?	OK	OK	?) 1
		\nearrow			1 ~	

(a)

Next state?

Meaning						
Initial state	INIT		40			
Got a 0 on A		oko:	OKO			
Got a 1 on A		AO	NAO!	OKI	OK	
Two equal A=0 last	OKO					
Two equal, A=1 last						

Meaning						
Initial state	INIT		AO			0
Got a O on A	AO				AI	0
Gova Yor A			40			0
Meaning						
Initial state	INIT	A0	AO	A		0
Got a 0 on A	40	oke	OKO			0
Gora 1 on A			AG	OK1	OKI	0
Two equal, A=0 last	OKO	OKO	OKO	OK	A	
Two equal, A=1 last	OK					

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

(a)			A B				
(α)	Meaning	K	00	01	11	10	Z
	Initial state	INIT	AQ	A0	Α1	A1	0
	Got a 0 on A	A0	QK/	OK	Α1	Α1	0
	Got a 1 on A	A1	A0	AQ	OK	OK	0
	Got two equal A i	nputs OK (?	OK	OK	?) 1
			<i>/</i>			//	

(b)			A B				
(D)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	A0	A1	A1	0
	Got a 0 on A	A0	OK0	OK0	Α1	Α1	0
	Got a 1 on A	A1	A0	A0	OK1	OK1	0
	Two equal, A=0 last	OK0					1
	Two equal, A=1 last	OK1					1
				S			

7	r .			
	ext	\sim	-04	
	LX			

		AO				
AO	oko II	OKO				
	AO	ĄQ	OKI	OKI		
OKO						
oki.						
	INIT AO A1 OKO	INIT A0 A0 OKO A1 A0 OKO	INIT AC AC AC OKO OKO A1 AC AC OKO	INIT AO AO A1 AO OKO OKO A1 A1 AO AO OK1 OKO	A1 A0 A0 OK1 OK1 OK0	

Meaning						
initial state			AG			
Gola 0 on A		oko	OKO			
Got a 1 on A				OKI	OKI	
Two equal. A=0 last	OKO	OKO	OKO	OKT	At	
Two equal, A=1 last	joki.					

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

			Α	В		
Meaning	K	00	01	11	10	Z
Initial state	INIT	AQ	A0	Α1	A1	0
Got a 0 on A	A0	9K	OK	Α1	Α1	0
Got a 1 on A	A1	A0	AQ	OK	OK	0
Got two equal A inp	uts OK (?	OK	OK	?) 1
		\nearrow			1/	

(a)

(c)

(b)							
(D)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	A0	A1	A1	0
	Got a 0 on A	A0	OK0	OK ₀	Α1	Α1	0
	Got a 1 on A	A1	A0	A0	OK1	OK1	0
	Two equal, A=0 last	OK ₀					1
	Two equal, A=1 last	OK1					1
				S			

T T	•			
	ext	\mathbf{C}^{d}	- 🔿 -1	$\vdash \cap$
	$\mathbf{H} \mathbf{X} \mathbf{I}$	-		
T				

Meaning	S	00	01	11	10	Z
Initial state	INIT	A0	A0	A1	A1	0
Got a 0 on A	A0	OK0	OK0	Α1	Α1	0
Got a 1 on A	Α1	A0	A0	OK1	OK1	0
Two equal, A=0 last	OK0	OK ₀	OK ₀	OK1	A1	1
Two equal, A=1 last	OK1					1
			S			

Meaning							
Initial state		AO	ΑO				
Got a 0 on A	40	OKO					
Gora 1 on A				OKt	OKI		
Two equal. A=0 last	OKO	OKG	OKO	OKI	A		
Two equal, A=1 last	OKI						

Design a clocked synchronous state machine with two inputs, A and B, and a single output Z that is 1 if:

- A had the same value at each of the two previous clock ticks, or
- B has been 1 since the last time that the first condition was true.

Otherwise, the output should be 0.

(a)							
(α)	Meaning	K	00	01	11	10	Z
	Initial state	INIT	AQ	A0	Α1	A1	0
	Got a 0 on A	A0	QK/	OK	Α1	Α1	0
	Got a 1 on A	A1	A0	AQ	OK	OK	0
	Got two equal A in	puts OK (?	OK	OK	\?) 1
			<i>~</i>			1 /	

(b)			АВ				
(D)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	A0	A1	A1	0
	Got a 0 on A	A0	OK0	OK0	Α1	Α1	0
	Got a 1 on A	A1	Α0	A0	OK1	OK1	0
	Two equal, A=0 last	OK0					1
	Two equal, A=1 last	OK1					1
				S			

Next state?

(c)							
(0)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	A0	A1	A1	0
	Got a 0 on A	A0	OK0	OK0	Α1	Α1	0
	Got a 1 on A	Α1	A0	A0	OK1	OK1	0
	Two equal, A=0 last	OK0	OK0	OK ₀	OK1	A1	1
	Two equal, A=1 last	OK1					1
				S			

(d)			A B				
(u)	Meaning	S	00	01	11	10	Z
	Initial state	INIT	A0	A0	Α1	A1	0
	Got a 0 on A	A0	OK0	OK0	Α1	Α1	0
	Got a 1 on A	Α1	A0	A0	OK1	OK1	0
	Two equal, A=0 last	OK0	OK0	OK0	OK1	Α1	1
	Two equal, A=1 last	OK1	A0	OK0	OK1	OK1	1
				S			