Digital Design and Computer Organization

Finite State Machines - 2

Reetinder Sidhu

Department of Computer Science and Engineering

Digital Design and Computer Organization

Finite State Machines - 2

Reetinder Sidhu

Department of Computer Science and Engineering

Finite State Machines - 2

Course Outline

- Digital Design
  - Combinational logic design
  - Sequential logic design
    - **Finite State Machines - 2**
- Computer Organization
  - Architecture (microprocessor instruction set)
  - Microarchitecure (microprocessor operation)

Concepts covered

- Finite State Machine Design Example

Finite State Machines - 2

How to Design Synchronous Sequential Logic Circuits? (Moore FSM)

- Determine inputs and outputs
- State transition diagram
- Encoding tables
  - State
  - Output
- State transition table
- Output table
- Logic minimization
  - State transition table yields Boolean formulas for next state logic
  - Output table yields Boolean formulas for output logic
- Logic circuit construction

Finite State Machines - 2

Lift (Elevator) Control Logic

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up

Source: *platformliftco*

Finite State Machines - 2
Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors
- Control logic should output signals to control the motor that takes the lift up and down

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors
- Control logic should output signals to control the motor that takes the lift up and down
  - Signal *lift_up* is 1 when the lift should move up and 0 otherwise
  - Signal *lift_down* is 1 when the lift should move down and 0 otherwise

Source: *platformliftco*

Finite State Machines - 2
Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors
- Control logic should output signals to control the motor that takes the lift up and down
  - Signal *lift_up* is 1 when the lift should move up and 0 otherwise
  - Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- Control logic should also output signals indicating which floor the lift is on

Source: *platformliftco*

Finite State Machines - 2
Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors
- Control logic should output signals to control the motor that takes the lift up and down
  - Signal *lift_up* is 1 when the lift should move up and 0 otherwise
  - Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- Control logic should also output signals indicating which floor the lift is on
  - *on_ground* is 1 when the lift is on ground floor and 0 otherwise
  - *on_first* is 1 when lift is on first floor and 0 otherwise

Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors
- Control logic should output signals to control the motor that takes the lift up and down
  - Signal *lift_up* is 1 when the lift should move up and 0 otherwise
  - Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- Control logic should also output signals indicating which floor the lift is on
  - *on_ground* is 1 when the lift is on ground floor and 0 otherwise
  - *on_first* is 1 when lift is on first floor and 0 otherwise
- Elevator initially on ground floor

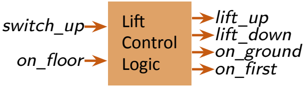
Source: *platformliftco*

Finite State Machines - 2

Lift (Elevator) Control Logic

- Problem is to design the control logic for a lift in a building of two floors: ground and first
- The lift has a switch with two positions, down and up, which respectively indicate that the lift should go down or up
  - The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The lift system has a sensor which indicates when the lift is stationary at one of the floors or is moving between them
  - The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors
- Control logic should output signals to control the motor that takes the lift up and down
  - Signal *lift_up* is 1 when the lift should move up and 0 otherwise
  - Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- Control logic should also output signals indicating which floor the lift is on
  - *on_ground* is 1 when the lift is on ground floor and 0 otherwise
  - *on_first* is 1 when lift is on first floor and 0 otherwise
- Elevator initially on ground floor

Source: *platformliftco*



switch_up

on_floor

Lift Control Logic

lift_up

lift_down

on_ground

on_first

FINITE STATE MACHINES - 2
State Transition Diagram

State Transition Diagram

- A visual representation of an FSM
  - States represent by circles (called **nodes/vertices**)
  - Transitions between states represented by directed line segments (called **arcs/edges/arrows**
  - Each edge is labeled with labeled with a Boolean formula of inputs corresponding to the transition
- In each clock cycle, only one state is **active**
- Active state transitions occur at the rising edge of the clock signal
- If a state is active in the current clock cycle, and it has an outgoing transition labeled with a Boolean formula which is 1 in that clock cycle, then the destination state of the transition will be the active state in the next clock cycle

Finite State Machines - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

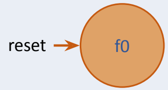
Elevator Example State Transition Diagram

Finite State Machines - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

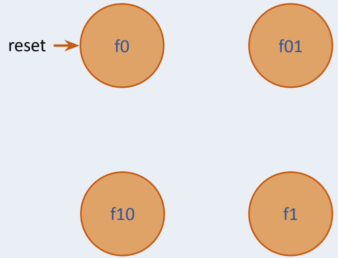
Elevator Example State Transition Diagram



State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

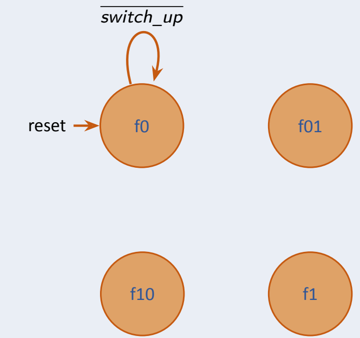
Elevator Example State Transition Diagram



Finite State Machines - 2
State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Elevator Example State Transition Diagram

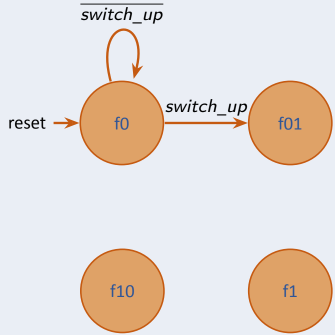


Finite State Machines - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Elevator Example State Transition Diagram

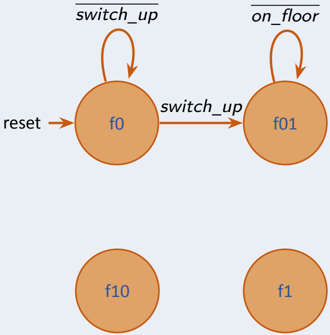


Finite State Machines - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Elevator Example State Transition Diagram

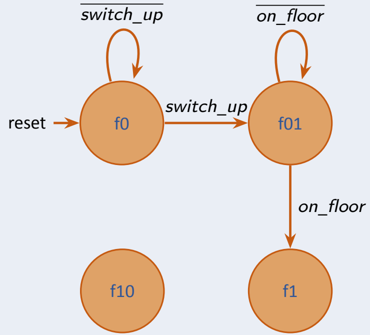


Finite State Machines - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Elevator Example State Transition Diagram

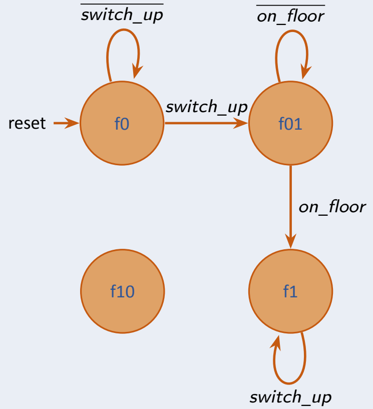


FINITE STATE MACHINES - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Elevator Example State Transition Diagram

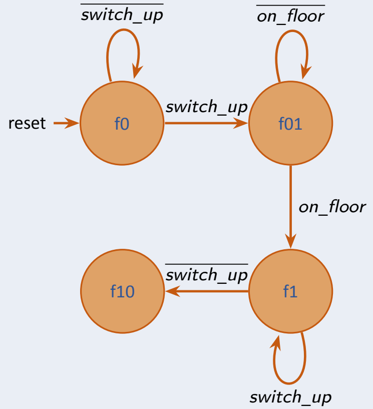


FINITE STATE MACHINES - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

Elevator Example State Transition Diagram

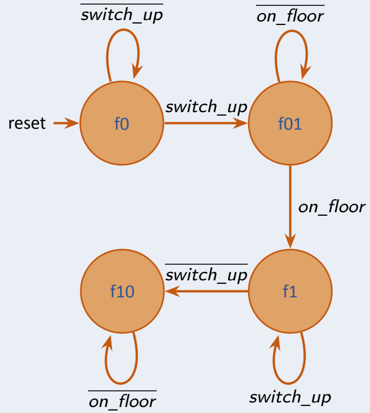


Finite State Machines - 2

State Transition Diagram

- The switch signal *switch_up* is 0 when the switch indicates the lift should go down and 1 when switch indicates it should go up
- The sensor signal *on_floor* is 1 when the lift is at ground or first floor, and 0 when the lift is moving between floors

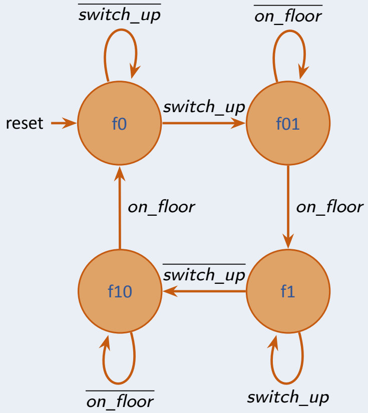
Elevator Example State Transition Diagram



Finite State Machines - 2

State Transition Diagram

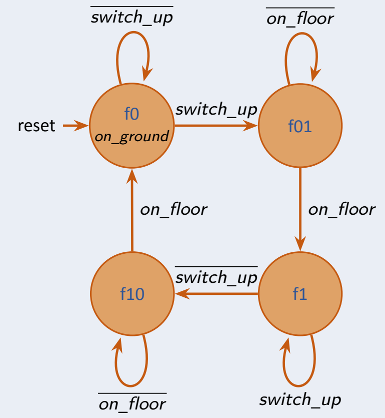
Elevator Example State Transition Diagram



Finite State Machines - 2
State Transition Diagram

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

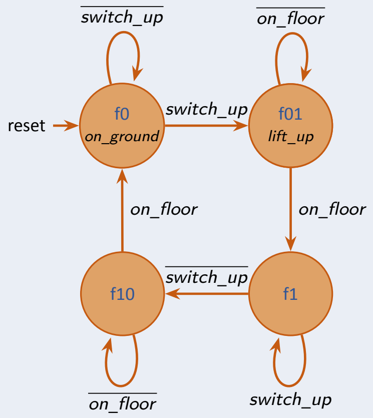
Elevator Example State Transition Diagram



FINITE STATE MACHINES - 2
State Transition Diagram

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

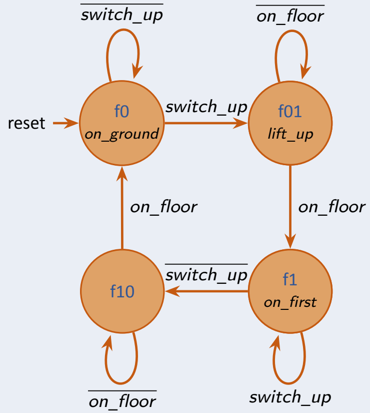
Elevator Example State Transition Diagram



Finite State Machines - 2
State Transition Diagram

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator Example State Transition Diagram

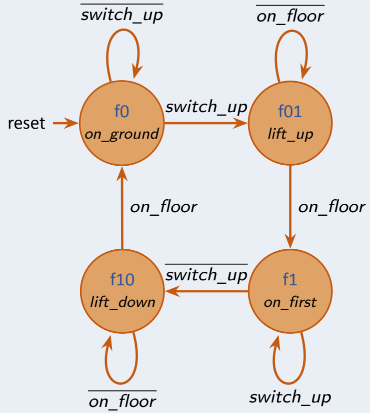


Finite State Machines - 2

State Transition Diagram

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator Example State Transition Diagram



Finite State Machines - 2

State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

FINITE STATE MACHINES - 2

State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator State Encoding Table

State	Encoding (s_1 s_0)
f0	00
f01	01
f1	11
f10	10

Finite State Machines - 2

State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator State Encoding Table

State	Encoding (s_1 s_0)
f0	00
f01	01
f1	11
f10	10

Elevator Output Encoding Tables

Finite State Machines - 2
State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator State Encoding Table

State	Encoding (s_1 s_0)
f0	00
f01	01
f1	11
f10	10

Elevator Output Encoding Tables

- *on_ground*

Meaning	Encoding
Lift on ground floor	1
Lift anywhere else	0

FINITE STATE MACHINES - 2

State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator State Encoding Table

State	Encoding (s_1 s_0)
f0	00
f01	01
f1	11
f10	10

Elevator Output Encoding Tables

- *on_ground*

Meaning	Encoding
Lift on ground floor	1
Lift anywhere else	0

- *on_first*

Meaning	Encoding
Lift on first floor	1
Lift anywhere else	0

Finite State Machines - 2

State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator State Encoding Table

State	Encoding (s_1 s_0)
f0	00
f01	01
f1	11
f10	10

Elevator Output Encoding Tables

- *on_ground*

Meaning	Encoding
Lift on ground floor	1
Lift anywhere else	0

- *on_first*

Meaning	Encoding
Lift on first floor	1
Lift anywhere else	0

- *lift_up*

Meaning	Encoding
Lift going from ground to first floor	1
Lift anywhere else	0

Finite State Machines - 2

State and Output Encoding Tables

- Signal *lift_up* is 1 when the lift should move up and 0 otherwise
- Signal *lift_down* is 1 when the lift should move down and 0 otherwise
- *on_ground* is 1 when the lift is on ground floor and 0 otherwise
- *on_first* is 1 when lift is on first floor and 0 otherwise

Elevator State Encoding Table

State	Encoding (s_1 s_0)
f0	00
f01	01
f1	11
f10	10

Elevator Output Encoding Tables

- *on_ground*

Meaning	Encoding
Lift on ground floor	1
Lift anywhere else	0

- *on_first*

Meaning	Encoding
Lift on first floor	1
Lift anywhere else	0

- *lift_up*

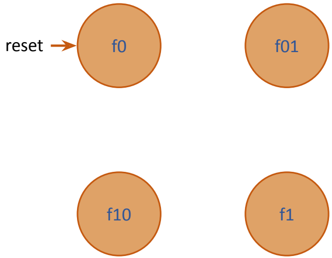
Meaning	Encoding
Lift going from ground to first floor	1
Lift anywhere else	0

- *lift_down*

Meaning	Encoding
Lift going from first to ground floor	1
Lift anywhere else	0

Finite State Machines - 2

State Transition Table

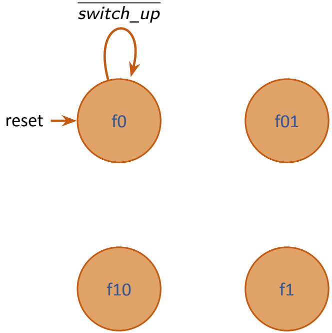


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s'_1	s'_0
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Finite State Machines - 2

State Transition Table

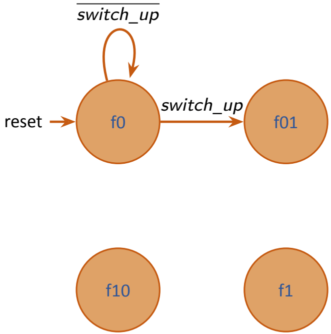


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

FINITE STATE MACHINES - 2

State Transition Table

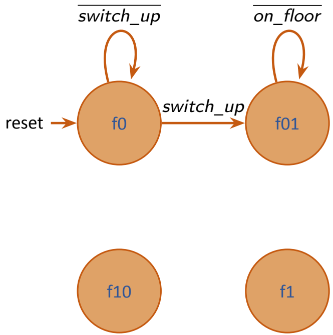


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	$switch\_up$	on_floor	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Finite State Machines - 2

State Transition Table

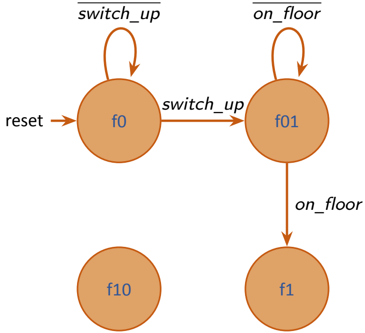


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1		
0	1	1	0	0	1
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Finite State Machines - 2

State Transition Table

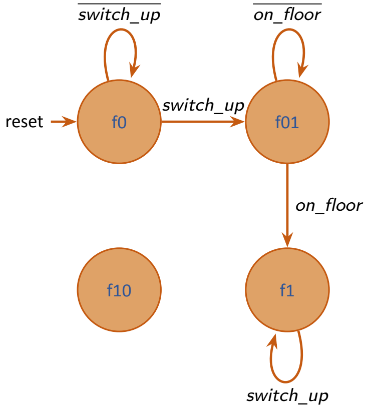


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s'_1	s'_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Finite State Machines - 2

State Transition Table

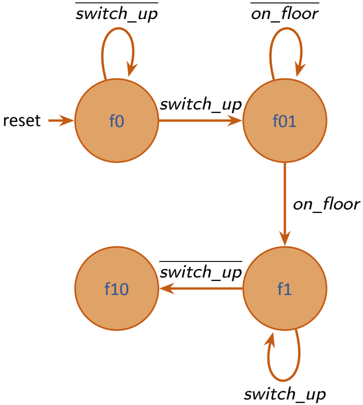


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0	1	1
1	1	1	1	1	1

Finite State Machines - 2

State Transition Table

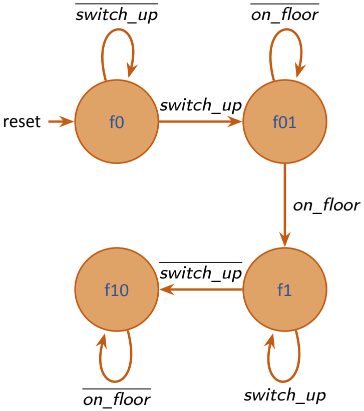


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Finite State Machines - 2

State Transition Table

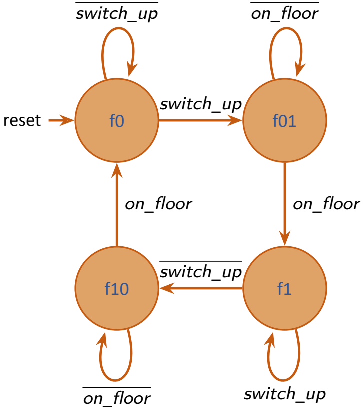


Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1		
1	0	1	0	1	0
1	0	1	1		
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Finite State Machines - 2

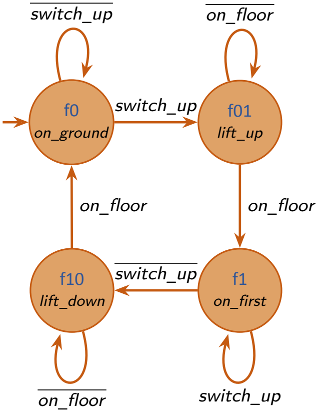
State Transition Table



Elevator Example State Tansition Table

Current State		Inputs		Next Sate	
s_1	s_0	$switch\_up$	on_floor	s'_1	s'_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Finite State Machines - 2
Output Table

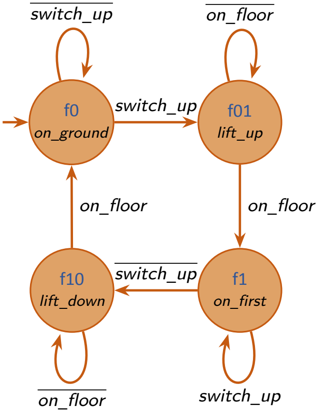


Elevator Example Output Table

State		Outputs			
s_1	s_0	on_ground	on_first	lift_up	lift_down
0	0				
0	1				
1	0				
1	1				

Finite State Machines - 2

Output Table

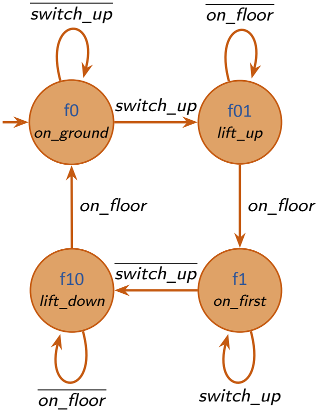


Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1			
0	1				
1	0				
1	1				

Finite State Machines - 2

Output Table

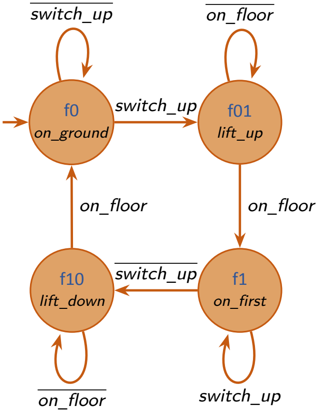


Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1			
0	1	0			
1	0	0			
1	1	0			

Finite State Machines - 2

Output Table

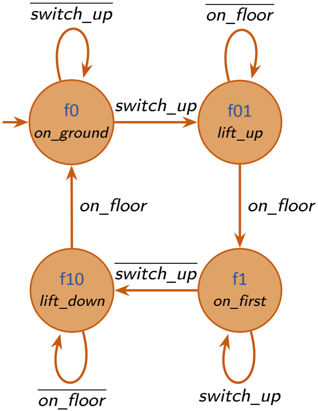


Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1			
0	1	0		1	
1	0	0			
1	1	0			

Finite State Machines - 2

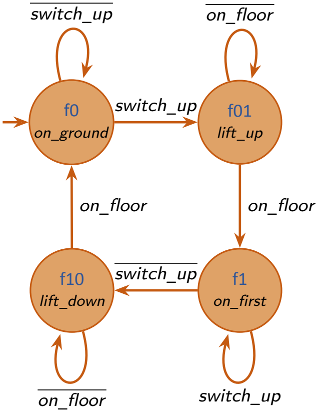
Output Table



Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1		0	
0	1	0		1	
1	0	0		0	
1	1	0		0	

FINITE STATE MACHINES - 2
Output Table

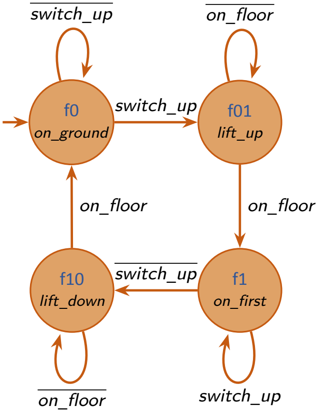


Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1		0	
0	1	0		1	
1	0	0		0	
1	1	0	1	0	

FINITE STATE MACHINES - 2

Output Table

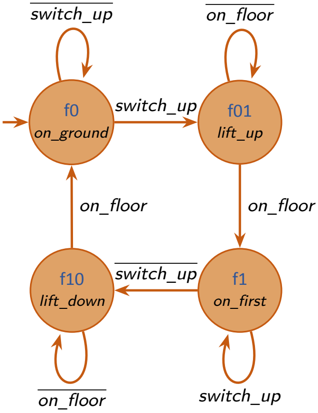


Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1	0	0	
0	1	0	0	1	
1	0	0	0	0	
1	1	0	1	0	

FINITE STATE MACHINES - 2

Output Table

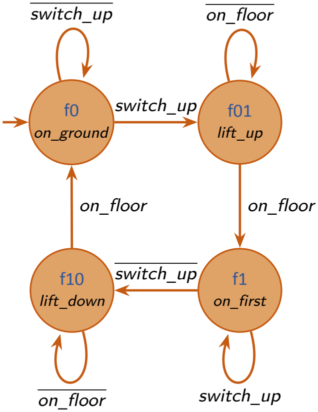


Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1	0	0	
0	1	0	0	1	
1	0	0	0	0	1
1	1	0	1	0	

Finite State Machines - 2

Output Table



Elevator Example Output Table

State		Outputs			
s_1	s_0	*on_ground*	*on_first*	*lift_up*	*lift_down*
0	0	1	0	0	0
0	1	0	0	1	0
1	0	0	0	0	1
1	1	0	1	0	0

Finite State Machines - 2

Logic Minimization (State)

Finite State Machines - 2

Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Finite State Machines - 2

Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Minimized Boolean Formula

- K-map for s_1':

	switch_up on_floor			
$s_1 s_0$	00	01	11	10
00				
01				
11				
10				

Finite State Machines - 2

Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Minimized Boolean Formula

- K-map for s_1':

$s_1\,s_0$ \ *switch_up on_floor*	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	1	1
10	1	0	0	1

Finite State Machines - 2

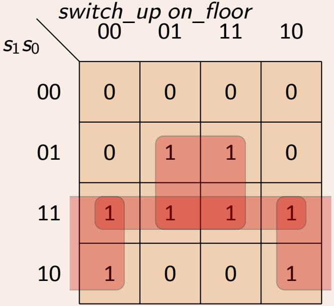
Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Minimized Boolean Formula

- K-map for s_1':



$s_1 s_0$ \ *switch_up on_floor*	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	1	1
10	1	0	0	1

- Minimized formula:

$$s_1' = s_1\ s_0 + s_1\ \overline{on\_floor} + s_0\ on\_floor$$

Finite State Machines - 2
Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Minimized Boolean Formula

- K-map for s_0':

ab cd	00	01	11	10
00				
01				
11				
10				

Finite State Machines - 2

Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Minimized Boolean Formula

- K-map for s_0':

cd ab	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	1	1
10	0	0	0	0

Finite State Machines - 2

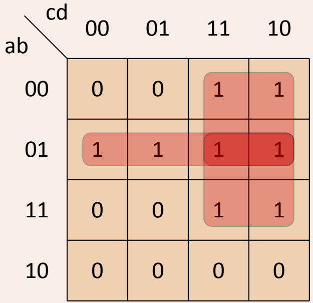
Logic Minimization (State)

- State transition table:

Current State		Inputs		Next Sate	
s_1	s_0	*switch_up*	*on_floor*	s_1'	s_0'
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

Minimized Boolean Formula

- K-map for s_0':



ab \ cd	00	01	11	10
00	0	0	1	1
01	1	1	1	1
11	0	0	1	1
10	0	0	0	0

- Minimized formula:

$$s_0' = \overline{s_1}\, s_0 + \overline{s_1}\, switch\_up + s_0\, switch\_up$$

Finite State Machines - 2

Logic Diagram

- Next state formulas:
  - $s_1' = s_1\ s_0 + s_1\ \overline{on\_floor} + s_0\ on\_floor$
  - $s_0' = \overline{s_1}\ s_0 + \overline{s_1}\ switch\_up + s_0\ switch\_up$

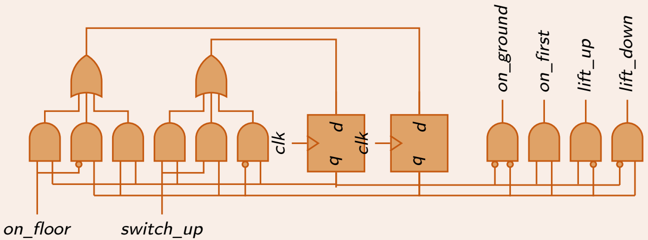
- Output formulas:
  - $on\_ground = \overline{s_1}\ \overline{s_0}$
  - $on\_first = s_1 s_0$
  - $lift\_up = \overline{s_1} s_0$
  - $lift\_down = s_1 \overline{s_0}$

Finite State Machines - 2

Logic Diagram

- Next state formulas:
  - $s_1' = s_1 \, s_0 + s_1 \, \overline{on\_floor} + s_0 \, on\_floor$
  - $s_0' = \overline{s_1} \, s_0 + \overline{s_1} \, switch\_up + s_0 \, switch\_up$
- Output formulas:
  - $on\_ground = \overline{s_1} \, \overline{s_0}$
  - $on\_first = s_1 s_0$
  - $lift\_up = \overline{s_1} s_0$
  - $lift\_down = s_1 \overline{s_0}$

Elevator Example Logic Diagram



Finite State Machines - 2

Think About It

Design of a Moore FSM for traffic light controller (section 3.4.1 of your textbook)