



DIGITAL DESIGN AND COMPUTER ORGANIZATION

Finite State Machines - 4

Reetinder Sidhu

Department of Computer Science and Engineering

DIGITAL DESIGN AND COMPUTER ORGANIZATION

Finite State Machines - 4

Reetinder Sidhu

Department of Computer Science and
Engineering

- Digital Design
 - ▶ Combinational logic design
 - ▶ Sequential logic design
 - ★ **Finite State Machines - 4**
- Computer Organization
 - ▶ Architecture (microprocessor instruction set)
 - ▶ Microarchitecture (microprocessor operation)

Concepts covered

- Finite State Machine Design Example
 - ▶ One-Hot Encoding (OHE)

FINITE STATE MACHINES - 4

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 - 4

State Encoding



- **Binary encoding** Encode states as binary numbers 0 to $k - 1$
 - ▶ For k states $\lceil \log_2 k \rceil$ flip-flops required

- **Binary encoding** Encode states as binary numbers 0 to $k - 1$
 - ▶ For k states $\lceil \log_2 k \rceil$ flip-flops required
- **One-Hot encoding (OHE)** Encoding of each state has a single 1 bit (other bits being 0)
 - ▶ For k states k flip-flops required

FINITE STATE MACHINES - 4

Lift (Elevator) Control Logic



Source: *platformliftco*

FINITE STATE MACHINES - 4

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 - 4

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 - 4

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

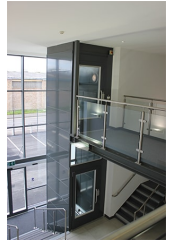


Source: *platformliftco*

FINITE STATE MACHINES - 4

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 - 4

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 - 4

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 - 4

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 - 4

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 - 4

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 - 4

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

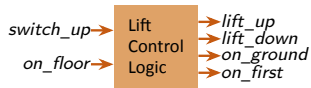
FINITE STATE MACHINES - 4

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 - 4

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 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 - 4

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 - 4

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

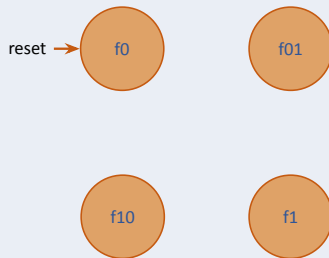


FINITE STATE MACHINES - 4

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

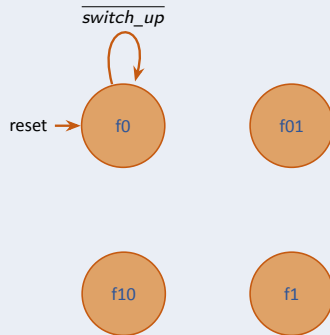


FINITE STATE MACHINES - 4

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

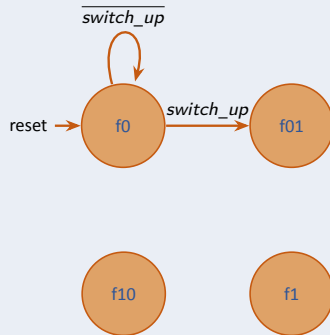


FINITE STATE MACHINES - 4

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

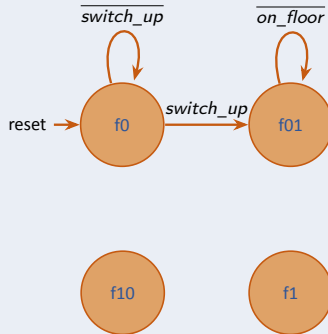


FINITE STATE MACHINES - 4

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

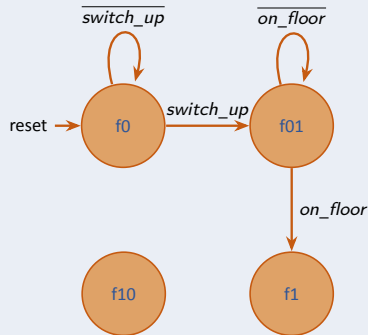


FINITE STATE MACHINES - 4

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 - 4

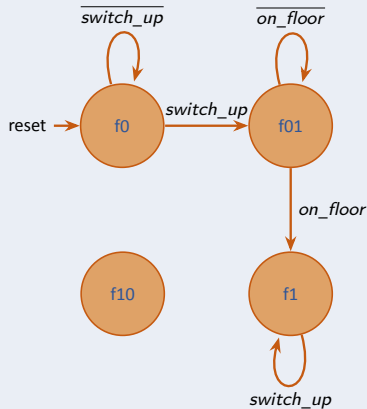
State Transition Diagram

- 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



PES
UNIVERSITY
ONLINE

Elevator Example State Transition Diagram

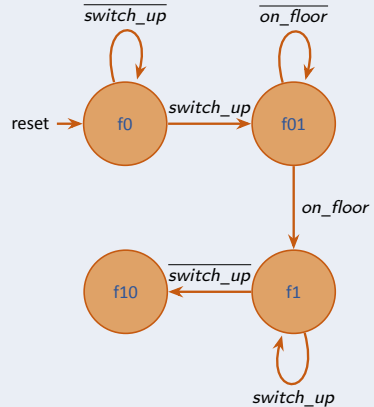


FINITE STATE MACHINES - 4

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 - 4

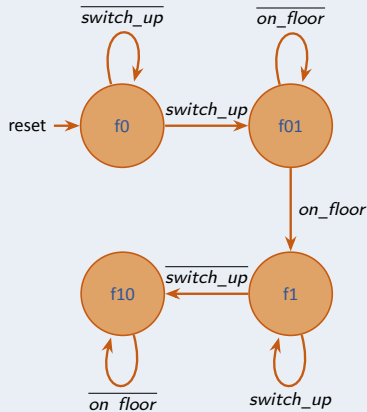
State Transition Diagram

- 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



PES
UNIVERSITY
ONLINE

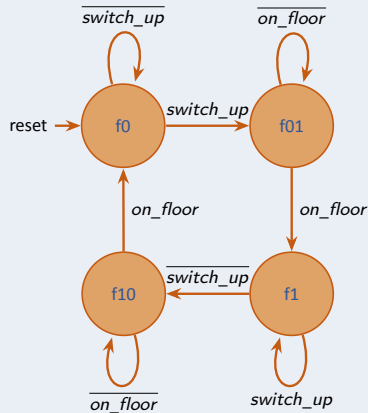
Elevator Example State Transition Diagram



FINITE STATE MACHINES - 4

State Transition Diagram

Elevator Example State Transition Diagram

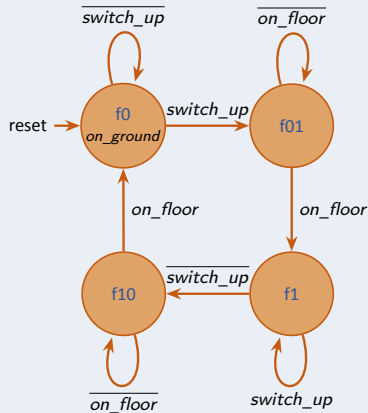


FINITE STATE MACHINES - 4

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

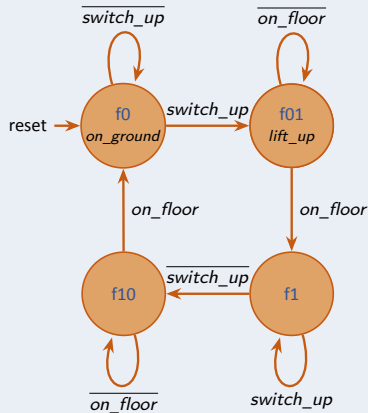


FINITE STATE MACHINES - 4

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

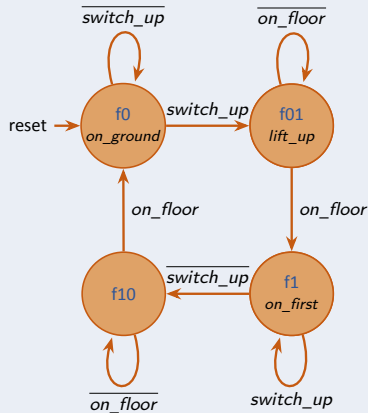


FINITE STATE MACHINES - 4

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

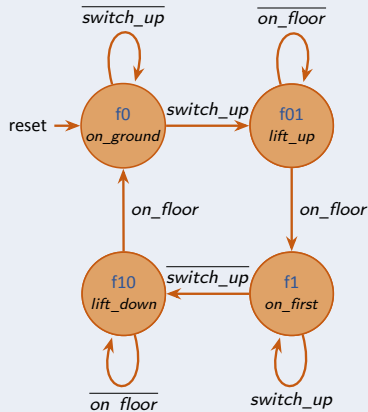


FINITE STATE MACHINES - 4

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 - 4

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 - 4

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 ($s_3 s_2 s_1 s_0$)
f0	0001
f01	0010
f1	0100
f10	1000

FINITE STATE MACHINES - 4

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 ($s_3 s_2 s_1 s_0$)
f0	0001
f01	0010
f1	0100
f10	1000

Elevator Output Encoding Tables

FINITE STATE MACHINES - 4

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 ($s_3 s_2 s_1 s_0$)
f0	0001
f01	0010
f1	0100
f10	1000

Elevator Output Encoding Tables

- *on_ground*

Meaning	Encoding
Lift on ground floor	1
Lift anywhere else	0

FINITE STATE MACHINES - 4

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 ($s_3 s_2 s_1 s_0$)
f0	0001
f01	0010
f1	0100
f10	1000

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 - 4

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 ($s_3 s_2 s_1 s_0$)
f0	0001
f01	0010
f1	0100
f10	1000

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 - 4

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 ($s_3 s_2 s_1 s_0$)
f0	0001
f01	0010
f1	0100
f10	1000

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*

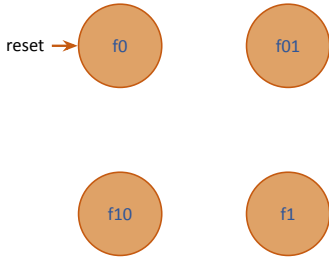
Meaning	Encoding
Lift going from ground to first floor	1
Lift anywhere else	0

- *lift_down*

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

FINITE STATE MACHINES - 4

State Transition Table

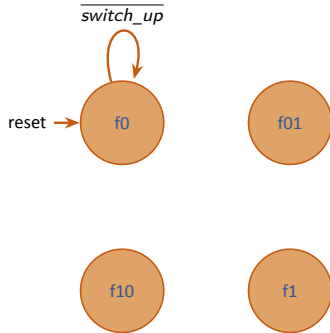


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0				
0	0	0	1	0	1				
0	0	0	1	1	0				
0	0	0	1	1	1				
0	0	1	0	0	0				
0	0	1	0	0	1				
0	0	1	0	1	0				
0	0	1	0	1	1				
0	1	0	0	0	0				
0	1	0	0	0	1				
0	1	0	0	1	0				
0	1	0	0	1	1				
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

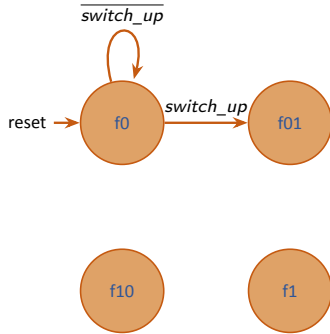


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0				
0	0	0	1	1	1				
0	0	1	0	0	0				
0	0	1	0	0	1				
0	0	1	0	1	0				
0	0	1	0	1	1				
0	1	0	0	0	0				
0	1	0	0	0	1				
0	1	0	0	1	0				
0	1	0	0	1	1				
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

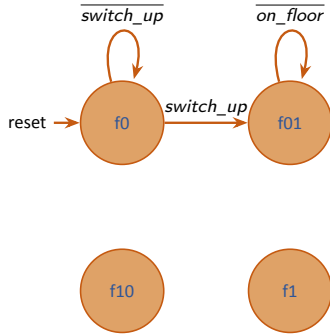


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0				
0	0	1	0	0	1				
0	0	1	0	1	0				
0	0	1	0	1	1				
0	1	0	0	0	0				
0	1	0	0	0	1				
0	1	0	0	1	0				
0	1	0	0	1	1				
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

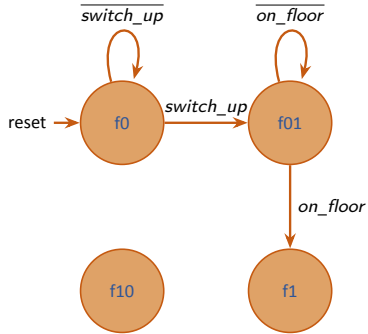


Elevator Example State Tansition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1				
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1				
0	1	0	0	0	0				
0	1	0	0	0	1				
0	1	0	0	1	0				
0	1	0	0	1	1				
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

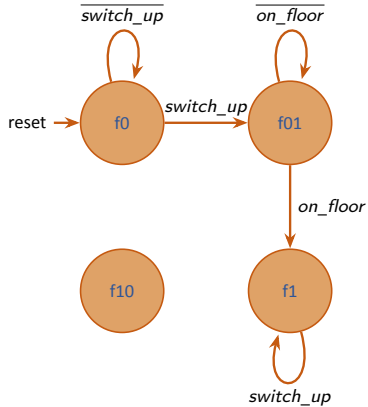


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0				
0	1	0	0	0	1				
0	1	0	0	1	0				
0	1	0	0	1	1				
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

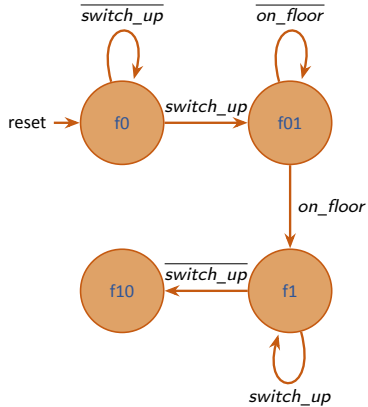


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0				
0	1	0	0	0	1				
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

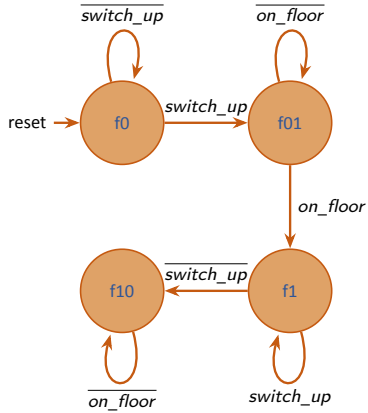


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0				
1	0	0	0	0	1				
1	0	0	0	1	0				
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

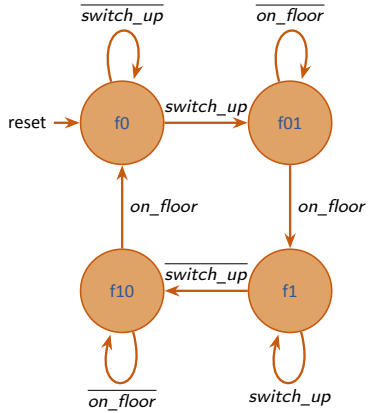


Elevator Example State Transition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1				
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1				

FINITE STATE MACHINES - 4

State Transition Table

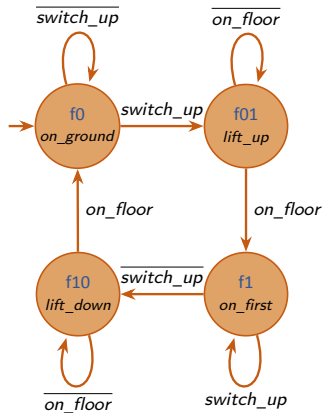


Elevator Example State Tansition Table

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

FINITE STATE MACHINES - 4

Output Table

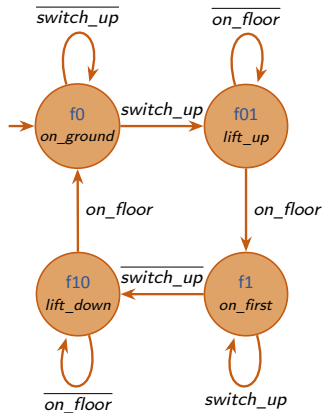


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1				
0	0	1	0				
0	1	0	0				
1	0	0	0				

FINITE STATE MACHINES - 4

Output Table

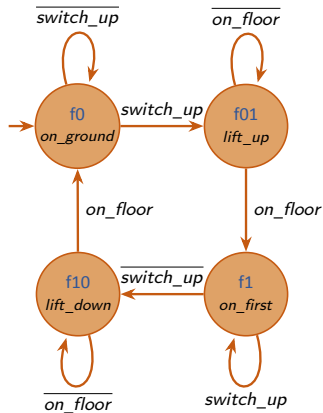


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1			
0	0	1	0				
0	1	0	0				
1	0	0	0				

FINITE STATE MACHINES - 4

Output Table

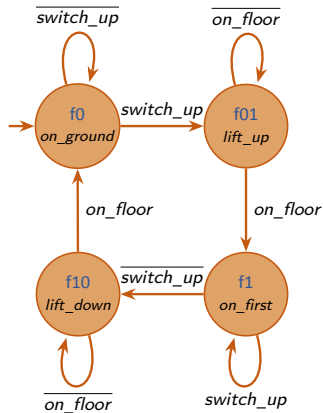


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1			
0	0	1	0	0			
0	1	0	0	0			
1	0	0	0	0			

FINITE STATE MACHINES - 4

Output Table

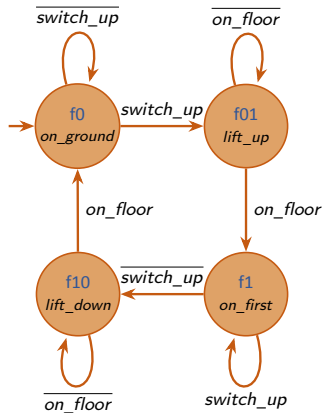


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1			
0	0	1	0	0		1	
0	1	0	0	0			
1	0	0	0	0			

FINITE STATE MACHINES - 4

Output Table

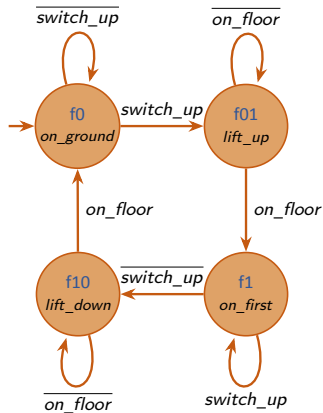


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1		0	
0	0	1	0	0		1	
0	1	0	0	0		0	
1	0	0	0	0		0	

FINITE STATE MACHINES - 4

Output Table

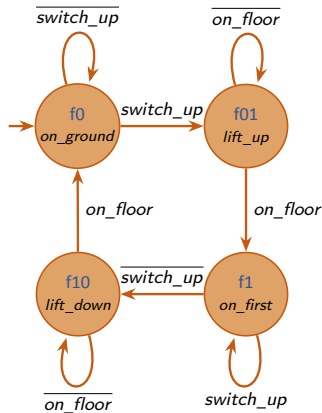


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1		0	
0	0	1	0	0		1	
0	1	0	0	0		0	
1	0	0	0	0	1	0	

FINITE STATE MACHINES - 4

Output Table

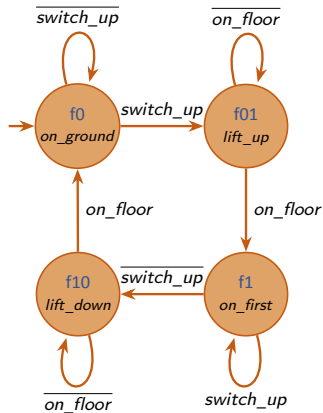


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1	0	0	
0	0	1	0	0	0	1	
0	1	0	0	0	0	0	
1	0	0	0	0	1	0	

FINITE STATE MACHINES - 4

Output Table

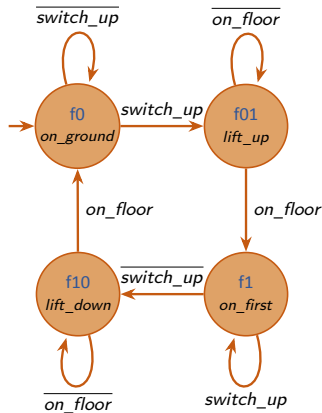


Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1	0	0	
0	0	1	0	0	0	1	
0	1	0	0	0	0	0	1
1	0	0	0	0	1	0	

FINITE STATE MACHINES - 4

Output Table



Elevator Example Output Table

State				Outputs			
s ₃	s ₂	s ₁	s ₀	on_ground	on_first	lift_up	lift_down
0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	0	1
1	0	0	0	0	1	0	0

FINITE STATE MACHINES - 4

Logic Minimization (State)

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

- K-map for s'_3 :

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	0	1	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

- K-map for s'_3 :
- SOP formula:

$$s'_3 = \overline{s_3} s_2 \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \overline{\text{on_floor}} + \overline{s_3} s_2 \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \text{on_floor} + s_3 \overline{s_2} \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \overline{\text{on_floor}} + s_3 \overline{s_2} \overline{s_1} \overline{s_0} \text{switch_up} \overline{\text{on_floor}}$$

$$s'_3 = \overline{s_3} s_2 \overline{s_1} \overline{s_0} \overline{\text{switch_up}} + s_3 \overline{s_2} \overline{s_1} \overline{s_0} \overline{\text{on_floor}}$$

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

- K-map for s'_3 :
- SOP formula:

$$s'_3 = \overline{s_3} s_2 \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \overline{\text{on_floor}} + \overline{s_3} s_2 \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \text{on_floor} + s_3 \overline{s_2} \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \overline{\text{on_floor}} + s_3 \overline{s_2} \overline{s_1} \overline{s_0} \overline{\text{switch_up}} \text{on_floor}$$

$$s'_3 = \overline{s_3} s_2 \overline{s_1} \overline{s_0} \overline{\text{switch_up}} + s_3 \overline{s_2} \overline{s_1} \overline{s_0} \overline{\text{on_floor}}$$
- Minimized formula:

$$s'_3 = s_2 \overline{\text{switch_up}} + s_3 \overline{\text{on_floor}}$$

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

- Minimized formula:

$$s'_2 = s_1 \text{ on_floor} + s_2 \text{ switch_up}$$

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

- Minimized formula:
$$s'_1 = s_0 \text{ switch_up} + s_1 \overline{\text{on_floor}}$$

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	switch_up	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

FINITE STATE MACHINES - 4

Logic Minimization (State)

- State transition table:

Current State				Inputs		Next State			
s_3	s_2	s_1	s_0	$switch_up$	on_floor	s'_3	s'_2	s'_1	s'_0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	1	0	0	0	1
0	0	0	1	1	0	0	0	1	0
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	1	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1

Minimized Boolean Formula

- Minimized formula:

$$s'_0 = s_0 \overline{switch_up} + s_3 \overline{on_floor}$$

FINITE STATE MACHINES - 4

Logic Minimization (Output)

- Output table:

State				Outputs			
s_3	s_2	s_1	s_0	<i>on_ground</i>	<i>on_first</i>	<i>lift_up</i>	<i>lift_down</i>
0	0	0	1				
0	0	1	0				
0	1	0	0				
1	0	0	0				

Boolean formulas

FINITE STATE MACHINES - 4

Logic Minimization (Output)

- Output table:

State				Outputs			
s_3	s_2	s_1	s_0	on_ground	on_first	$lift_up$	$lift_down$
0	0	0	1	1			
0	0	1	0	0			
0	1	0	0	0			
1	0	0	0	0			

Boolean formulas

- $on_ground = s_0$

FINITE STATE MACHINES - 4

Logic Minimization (Output)

- Output table:

State				Outputs			
s_3	s_2	s_1	s_0	<i>on_ground</i>	<i>on_first</i>	<i>lift_up</i>	<i>lift_down</i>
0	0	0	1	1	0		
0	0	1	0	0	0		
0	1	0	0	0	0		
1	0	0	0	0	1		

Boolean formulas

- $on_ground = s_0$
- $on_first = s_3$

FINITE STATE MACHINES - 4

Logic Minimization (Output)

- Output table:

State				Outputs			
s_3	s_2	s_1	s_0	<i>on_ground</i>	<i>on_first</i>	<i>lift_up</i>	<i>lift_down</i>
0	0	0	1	1	0	0	
0	0	1	0	0	0	1	
0	1	0	0	0	0	0	
1	0	0	0	0	1	0	

Boolean formulas

- $on_ground = s_0$
- $on_first = s_3$
- $lift_up = s_1$

FINITE STATE MACHINES - 4

Logic Minimization (Output)

- Output table:

State				Outputs			
s_3	s_2	s_1	s_0	<i>on_ground</i>	<i>on_first</i>	<i>lift_up</i>	<i>lift_down</i>
0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	0	1
1	0	0	0	0	1	0	0

Boolean formulas

- $on_ground = s_0$
- $on_first = s_3$
- $lift_up = s_1$
- $lift_down = s_2$

FINITE STATE MACHINES - 4

Logic Diagram

- Next state formulas:

- ▶ $s'_3 = s_2 \overline{\text{switch_up}} + s_3 \overline{\text{on_floor}}$
- ▶ $s'_2 = s_1 \overline{\text{on_floor}} + s_2 \overline{\text{switch_up}}$
- ▶ $s'_1 = s_0 \overline{\text{switch_up}} + s_1 \overline{\text{on_floor}}$
- ▶ $s'_0 = s_0 \overline{\text{switch_up}} + s_3 \overline{\text{on_floor}}$

- Output formulas:

FINITE STATE MACHINES - 4

Logic Diagram

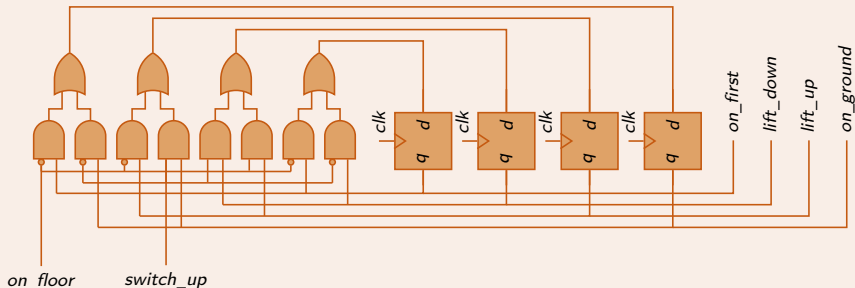
- Next state formulas:

- ▶ $s'_3 = s_2 \text{ switch_up} + s_3 \text{ on_floor}$
- ▶ $s'_2 = s_1 \text{ on_floor} + s_2 \text{ switch_up}$
- ▶ $s'_1 = s_0 \text{ switch_up} + s_1 \text{ on_floor}$
- ▶ $s'_0 = s_0 \text{ switch_up} + s_3 \text{ on_floor}$

- Output formulas:

- ▶ $\text{on_ground} = s_0$

Elevator Example Logic Diagram



FINITE STATE MACHINES - 4

Logic Diagram

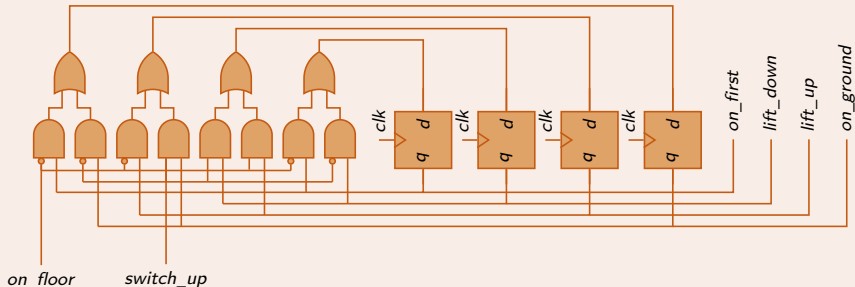
- Next state formulas:

- ▶ $s'_3 = s_2 \text{ switch_up} + s_3 \text{ on_floor}$
- ▶ $s'_2 = s_1 \text{ on_floor} + s_2 \text{ switch_up}$
- ▶ $s'_1 = s_0 \text{ switch_up} + s_1 \text{ on_floor}$
- ▶ $s'_0 = s_0 \text{ switch_up} + s_3 \text{ on_floor}$

- Output formulas:

- ▶ $\text{on_ground} = s_0$
- ▶ $\text{on_first} = s_3$

Elevator Example Logic Diagram



FINITE STATE MACHINES - 4

Logic Diagram

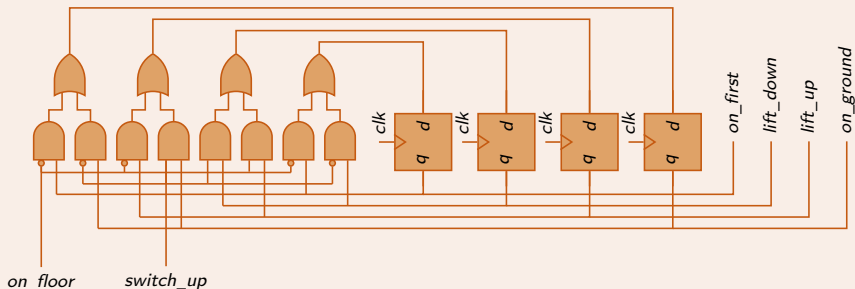
- Next state formulas:

- ▶ $s'_3 = s_2 \text{ switch_up} + s_3 \text{ on_floor}$
- ▶ $s'_2 = s_1 \text{ on_floor} + s_2 \text{ switch_up}$
- ▶ $s'_1 = s_0 \text{ switch_up} + s_1 \text{ on_floor}$
- ▶ $s'_0 = s_0 \text{ switch_up} + s_3 \text{ on_floor}$

- Output formulas:

- ▶ $\text{on_ground} = s_0$
- ▶ $\text{on_first} = s_3$
- ▶ $\text{lift_up} = s_1$

Elevator Example Logic Diagram



FINITE STATE MACHINES - 4

Logic Diagram

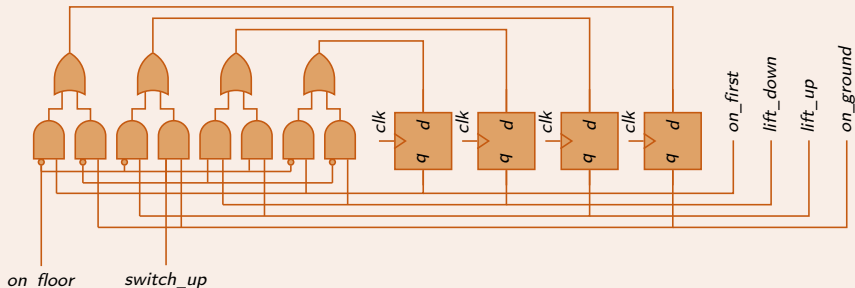
- Next state formulas:

- ▶ $s'_3 = s_2 \text{ switch_up} + s_3 \text{ on_floor}$
- ▶ $s'_2 = s_1 \text{ on_floor} + s_2 \text{ switch_up}$
- ▶ $s'_1 = s_0 \text{ switch_up} + s_1 \text{ on_floor}$
- ▶ $s'_0 = s_0 \text{ switch_up} + s_3 \text{ on_floor}$

- Output formulas:

- ▶ $\text{on_ground} = s_0$
- ▶ $\text{on_first} = s_3$
- ▶ $\text{lift_up} = s_1$
- ▶ $\text{lift_down} = s_2$

Elevator Example Logic Diagram



FINITE STATE MACHINES - 4

Sequential Logic Design Variations

- FSM type:
 - ▶ Mealy
 - ▶ Moore
- Encoding type:
 - ▶ Binary
 - ▶ One-Hot Encoding
- Elevator example:
 - ▶ Moore type with binary encoding
 - ▶ Mealy type with binary encoding
 - ▶ Moore type with one-hot encoding

FINITE STATE MACHINES - 4

Think About It



For the elevator example, try Mealy type with one-hot encoding