Lecture 17

- General finite state machine (FSM) design
- Moore/Mealy machines

Finite state machines

- FSM: A system that visits a finite number of logically distinct states
- Counters are simple FSMs
  - Outputs and states are identical
  - Visit states in a fixed sequence without inputs

More than counters

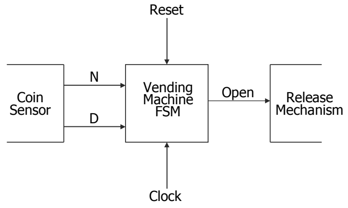
- FSMs are typically more complex than counters
  - Outputs can depend on current state and on inputs
  - State sequencing depends on current state and on inputs

FSM design

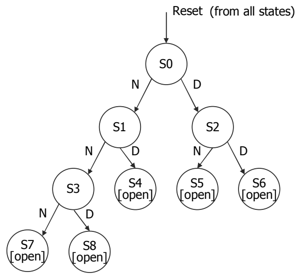
- Counter design procedure
  1. State diagram
  2. State-transition table
  3. Next-state logic minimization
  4. Implement the design
- FSM design procedure
  1. State diagram
  2. State-transition table
  3. State minimization
  4. State encoding
  5. Next-state logic minimization
  6. Implement the design

Example: Vending machine

- 15 cents for a cup of coffee
- Doesn't take pennies or quarters
- Doesn't provide any change



1. State diagram



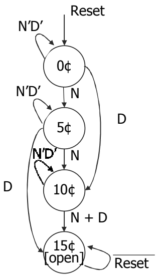
Draw self-loops for N' D' for S0 to S3

Also draw self-loops for 1 for S4 to S8

2. State transition table

present state	inputs D	inputs N	next state	output open
S0	0	0	S0	0
	0	1	S1	0
	1	0	S2	0
	1	1	X	X
S1	0	0	S1	0
	0	1	S3	0
	1	0	S4	0
	1	1	X	X
S2	0	0	S2	0
	0	1	S5	0
	1	0	S6	0
	1	1	X	X
S3	0	0	S3	0
	0	1	S7	0
	1	0	S8	0
	1	1	X	X
S4	X	X	S4	1
S5	X	X	S5	1
S6	X	X	S6	1
S7	X	X	S7	1
S8	X	X	S8	1

3. State minimization



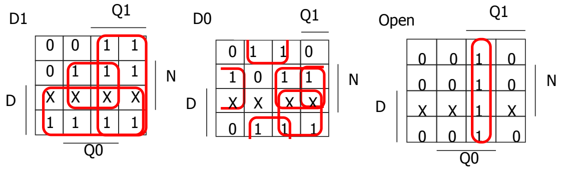
present state	inputs D	inputs N	next state	output open
0¢	0	0	0¢	0
	0	1	5¢	0
	1	0	10¢	0
	1	1	–	–
5¢	0	0	5¢	0
	0	1	10¢	0
	1	0	15¢	0
	1	1	–	–
10¢	0	0	10¢	
	0	1	15¢	0
	1	0	15¢	0
	1	1	–	–
15¢	–	–	15¢	1

symbolic state table

4. State encoding

present state Q1	present state Q0	inputs D	inputs N	next state D1	next state D0	output open
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	–	–	–
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	–	–	–
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	–	–	–
1	1	–	–	1	1	1

5. Next-state logic minimization



D1

0	0	1	1
0	1	1	1
X	X	X	X
1	1	1	1

D0

0	1	1	0
1	0	1	1
X	X	X	X
0	1	1	1

Open

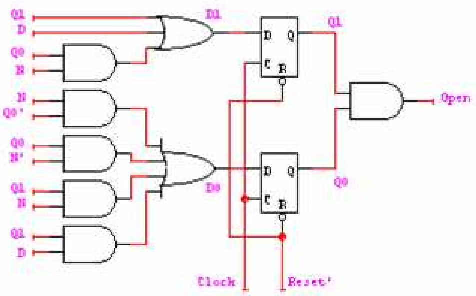
0	0	1	0
0	0	1	0
X	X	1	X
0	0	1	0

D1 = Q1 + D + Q0 N

D0 = Q0' N + Q0 N' + Q1 N + Q1 D

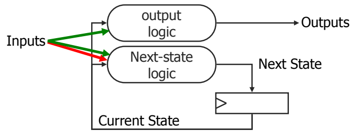
OPEN = Q1 Q0

6. Implement the design

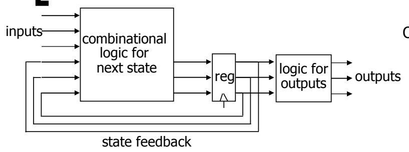


Generalized FSM model

- Combinational logic computes next state and outputs
  - Next state is a function of current state and inputs
  - Outputs are functions of
    - Current state (**Moore** machine)
    - Current state and inputs (**Mealy** machine)



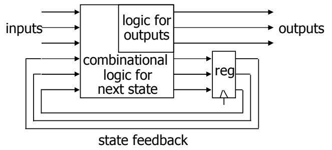
Moore vs. Mealy machines



Moore machine

Outputs are a function of current state

Outputs change synchronously with state changes



Mealy machine

Outputs depend on state and on inputs

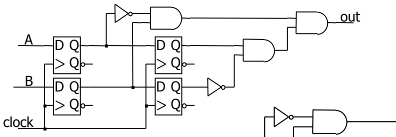
Input changes can cause immediate output changes **(asynchronous)**

State diagrams

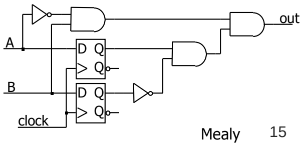
- Moore machine
  - Each *state* is labeled by a state-name/output pair.
- Mealy machine
  - Each *transition arc* is labeled by a input-condition/output pair.

Example: 10 → 01

- Circuits recognize AB=10 followed by AB=01
  - What kinds of machines are they?



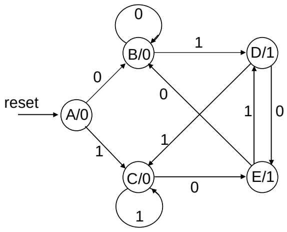
Moore



Mealy

Example: "01" or "10" detector

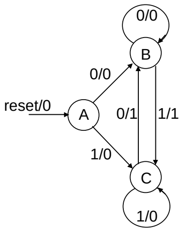
- Moore: Output is a function of state only
  - Specify output in the state bubble



reset	input	current state	next state	current output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	D	0
0	0	C	E	0
0	1	C	C	0
0	0	D	E	1
0	1	D	C	1
0	0	E	B	1
0	1	E	D	1

Example: "01" or "10" detector

- Mealy: Output is a function of state and inputs
  - Specify outputs on transition arcs



reset	input	current state	next state	current output
1	–	–	A	0
0	0	A	B	0
0	1	A	C	0
0	0	B	B	0
0	1	B	C	1
0	0	C	B	1
0	1	C	C	0

Moore vs. Mealy

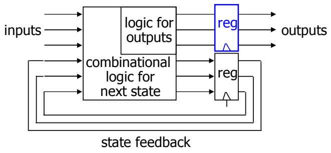
- Moore machines
  - + Safer to use because outputs change at clock edge
  - – May take additional logic to decode state into outputs
- Mealy machines
  - + Typically have fewer states
  - + React faster to inputs — don't wait for clock
  - – Asynchronous outputs can be dangerous

Synchronous Mealy machines

- We often design synchronous Mealy machines
  - Design a Mealy machine
  - Then register the outputs

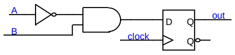
Synchronous Mealy machines

- Registered state and registered outputs
  - No glitches on outputs
  - No race conditions between communicating machines

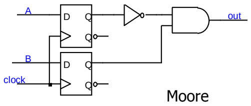


Example: "== 01?"

- Recognize AB = 01
  - Mealy or Moore?



Synchronous Mealy
(Moore)



Moore