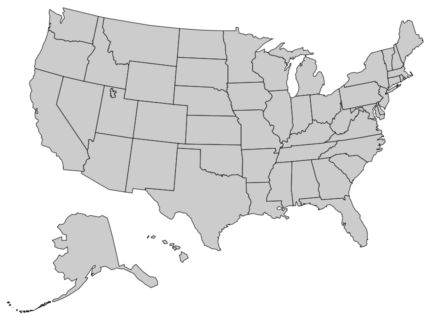
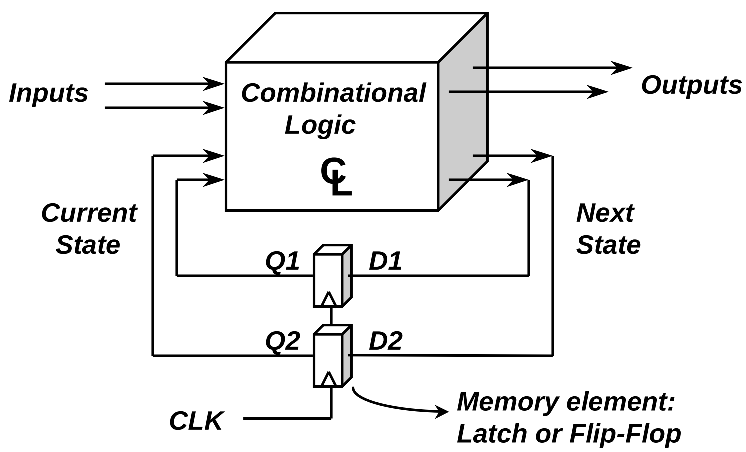
6004 Spring 1998: L12

Finite State Machines



1

Synchronous Sequential Circuit



Issues: Specification, design, clocking and timing

2

Abstraction: Finite State Machine

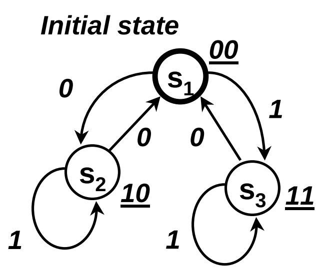
- **A Finite State Machine (FSM) has:**
  - **K states, $S = \{s_1, s_2, \ldots, s_K\}$, initial state s_1**
  - **N inputs, $I = \{i_1, i_2, \ldots, i_N\}$**
  - **M outputs, $O = \{o_1, o_2, \ldots, o_M\}$**
  - **Transition function T(S, I) mapping each current state and input to a next state**
  - **Output function O(S) mapping each current state to an output**

- **Given a sequence of inputs the FSM produces a sequence of outputs which is dependent on s_1, T(S, I) and O(S)**

3

FSM Representations

State Transition Graph



	t	*t+1*	*t+2*
Inputs:	*0*	*1*	*0*
Outputs:	*00*	___	

State Transition Table

T(S, I)

0	s_1	s_2
1	s_1	s_3
0	s_2	s_1
1	s_2	s_2
0	s_3	s_1
1	s_3	s_3

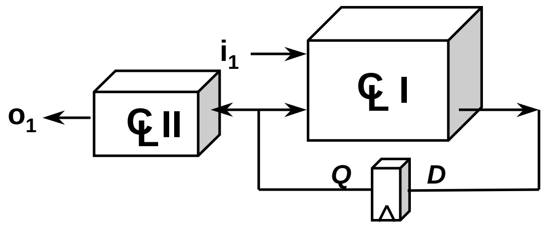
O(S)

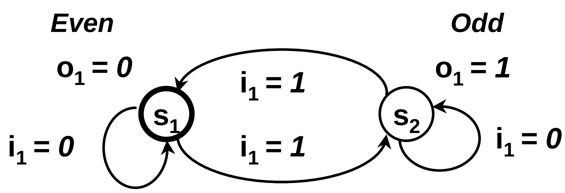
s_1	*00*
s_2	*10*
s_3	*11*

4

Simple Design Example

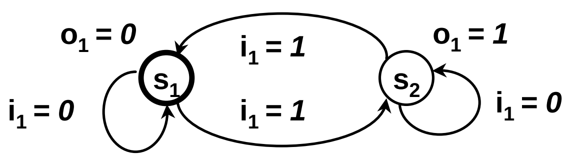
- Design a FSM that outputs a 1 if and only if the number of 1's in the input sequence is odd





5

State Encoding



- State Encoding: Choose a <u>unique</u> binary code for each s_i so the combinational logic can be specified
  - Choose $s_1 = 0$ and $s_2 = 1$
  - Choose $s_1 = 1$ and $s_2 = 0$

6

Logic Implementations

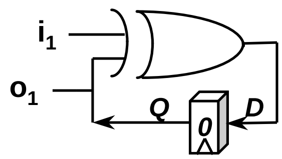
Choose s_1 ***= 0 and*** s_2 ***= 1***

CL I

i_1	Q	D
0	0	0
1	0	1
0	1	1
1	1	0

CL II

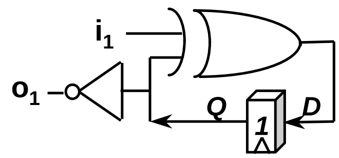
Q	o_1
0	0
1	1



Choose s_1 ***= 1 and*** s_2 ***= 0***

i_1	Q	D
0	1	1
1	1	0
0	0	0
1	0	1

Q	o_1
1	0
0	1



7

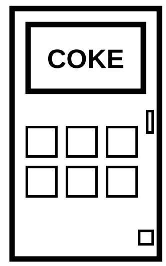
Observations

- **Number of bits required to encode K states is $\lceil \log_2 K \rceil$**
- **Encoding states results in combinational logic specifications for T(S, I) and O(S)**
- **Choice of encoding affects complexity of logic implementation**
  - **How does one find the optimum state encoding?**

8

Coke Machine Example

- Coke costs $.10
- Only nickels and dimes accepted
- FSM inputs:
  - 5: Nickel
  - 10: Dime
  - Coke: Give me a coke
  - Return: Give me my money back
- FSM outputs:
  - Drop: Drop a coke
  - Ret5: Return $.05
  - Ret10: Return $.10

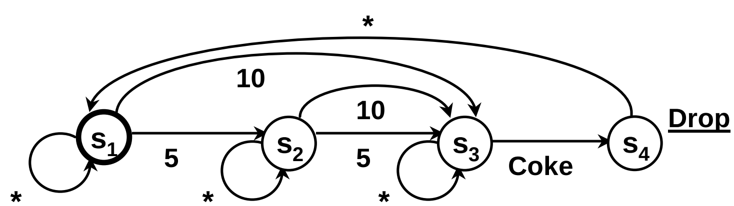


9

Coke Machine State Diagram

Assumption: At most one input among Coke, 5, 10, and Return is asserted

*** represents all unspecified transitions from state**

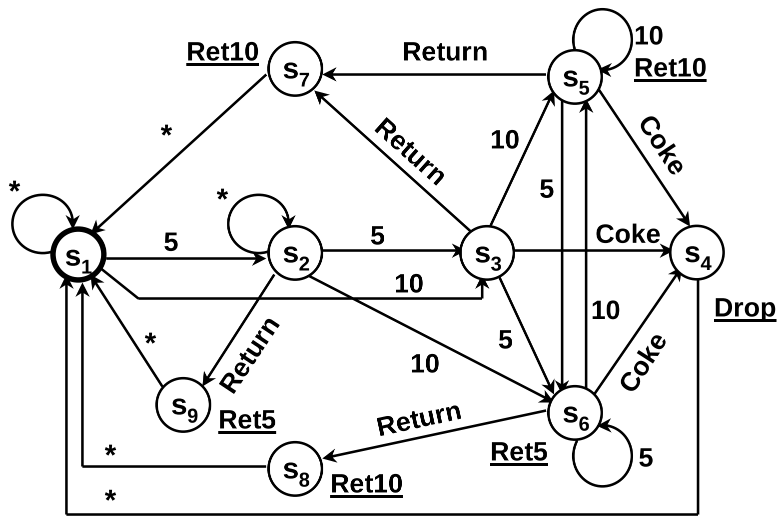


Does this work? ________________

10

6004 Spring 1998: L12

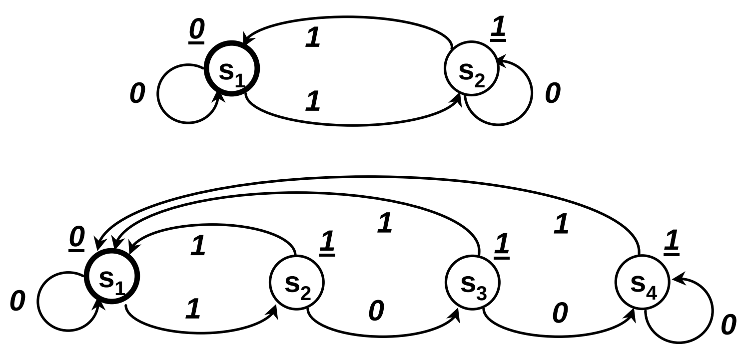
Coke Machine Diagram - II



After Return input, any input in the next cycle is ignored!

FSM Equivalence

Are these two FSMs equivalent?



Answer: ___________

Equivalence and Minimization

- Two FSMs are equivalent if and only if every input sequence yields identical output sequences
- Goal: Given an FSM, find the simplest equivalent FSM with a minimum number of states
  – Two states s_1 and s_2 in an FSM are equivalent if and only if each input sequence beginning from s_1 yields an output sequence identical to that obtained by starting from s_2

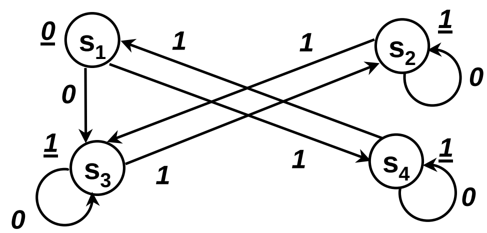
13

FSM Minimization

- Look at each pair of states (s_1, s_2) in the FSM
- If s_1 produces different outputs from s_2 , mark them non-equivalent
- For each state pair (s_1, s_2) not yet marked, for each input i , find state pair ($T(s_1, i)$, $T(s_2, i)$)
- If ($T(s_1, i)$, $T(s_2, i)$) are marked non-equivalent for any i, mark (s_1, s_2) non-equivalent
- Iterate until no more marking is possible
- Unmarked state pairs are equivalent, simplify FSM accordingly

14

Example Minimization

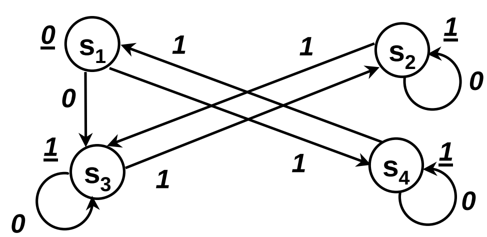


$(\cancel{s_1, s_2})$ $(\cancel{s_1, s_3})$ $(\cancel{s_1, s_4})$ (s_2, s_3) (s_2, s_4) (s_3, s_4)

(s_2, s_3) (s_2, s_4) (s_3, s_4)

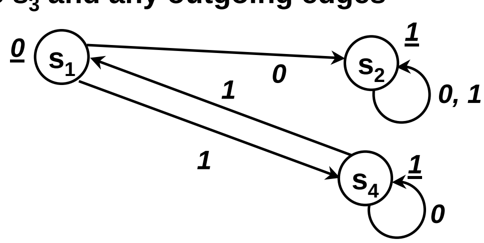
15

Simplified FSM



(s_2, s_3)
equivalent

Pick s_2 and move incident edges of s_3 to s_2
Delete s_3 and any outgoing edges

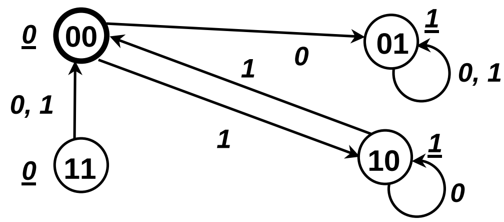


16

6004 Spring 1998: L12

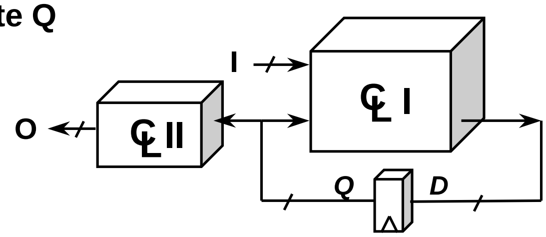
Illegal States

- Need 2 bits to encode 3 states
- Specify, for each illegal state, transitions to starting state on all inputs and "inactive" outputs

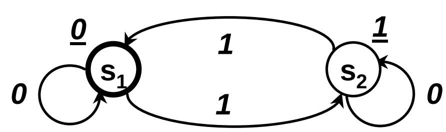


Moore Machines

- So far we considered Moore machines where the output O is a function of only the current state Q

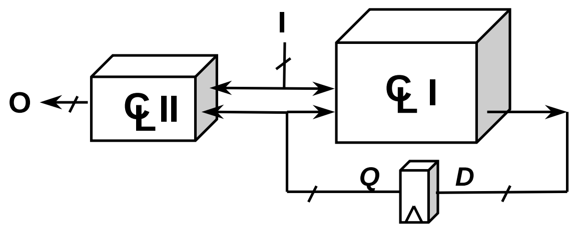


- Moore FSM State Transition Graph

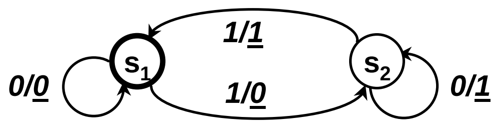


Mealy Machines

- **In Mealy machines the output O is a function of the current state Q and input I**



- **Mealy FSM State Transition Diagram**



19