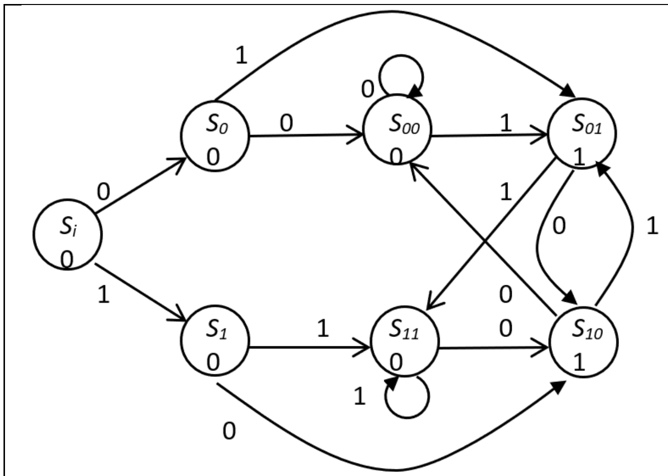
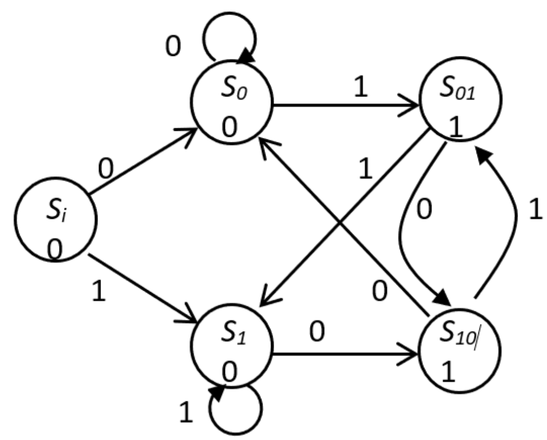
Lecture Notes | 10.14 | ECS 154A
2018-08-21 | Moore model FSM minimization | Summer Session II 2018

1. From the last lecture
   a. Was working on the Moore model edge detector FSM
   b. Used Partition Minimization Procedure to minimize the original state diagram
      i. $P_3 = (S_i)(S_1, S_{11})(S_0, S_{00})(S_{01})(S_{10})$
   c. Now use this to create minimized state diagram
      i. K-successors are the same for the multiple-state blocks, use that to combine them together



Original state diagram



Minimized state diagram

2. Implementing the minimized FSM
   a. From our minimized, equivalent FSM we get the following state table

Present State	Next State $x = 0$	Next State $x = 1$	Output z
i	0	1	0
0	00	01	0
1	10	11	0
00	00	01	0
01	10	01	1
10	00	01	1
11	10	11	0

Original State Table

Present State	Next State $x = 0$	Next State $x = 1$	Output z
i	0	1	0
0	0	01	0
1	10	1	0
01	10	1	1
10	0	01	1

Minimized State Table

   b. Next, assign binary codes
      i. Will need 3 flip flops to represent 5 states, call these A, B, and C

Present State	Binary Code	Present State *A*	*B*	*C*	Input *x*	Next State *A'*	*B'*	*C'*	Output *z*
i	000	0	0	0	0	0	0	1	0
i	000	0	0	0	1	0	1	0	0
0	001	0	0	1	0	0	0	1	0
0	001	0	0	1	1	0	1	1	0
1	010	0	1	0	0	1	0	0	0
1	010	0	1	0	1	0	1	0	0
01	011	0	1	1	0	1	0	0	1
01	011	0	1	1	1	0	1	0	1
10	100	1	0	0	0	0	0	1	1
10	100	1	0	0	1	0	1	1	1

c. Create K-maps for each flip flop based on input and present state
   i. States that weren't assigned form don't cares

A'

Cx* \ *AB	**00**	**01**	**11**	**10**
00	0	1	d	0
01	0	0	d	0
11	0	0	d	d
10	0	1	d	d

***A'* = $B\overline{x}$**

B'

Cx* \ *AB	**00**	**01**	**11**	**10**
00	0	0	d	0
01	1	1	d	1
11	1	1	d	d
10	0	0	d	d

B'* = *x

C'

Cx* \ *AB	**00**	**01**	**11**	**10**
00	1	0	d	1
01	0	0	d	1
11	1	0	d	d
10	1	0	d	d

C'= $\boldsymbol{A + \overline{Bx} + \overline{B}C}$

d. Use derivations from these K-maps to design initial combinational circuit
e. Create a K-Map based on flip-flops to determine the output combinational circuit
   i. Assign don't cares for the same reason as above

z

C \ *AB*	**00**	**01**	**11**	**10**
0	0	0	d	1
1	0	1	d	d

z = *A* + BC