

ECE212/212s Lecture Notes

Lecture: Chapter 9

Dr. Clay Gloster, Jr.

State Assignment Guidelines

HP: State with the same next state for a given input transition should be given adjacent assignments.

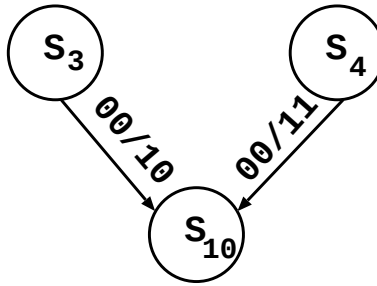


Figure 1: Partial state diagram: S_3, S_4 adjacent.

MP: Next states of the same state should be given adjacent assignments.

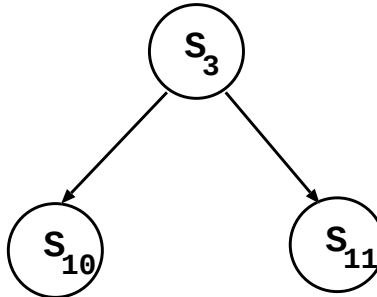
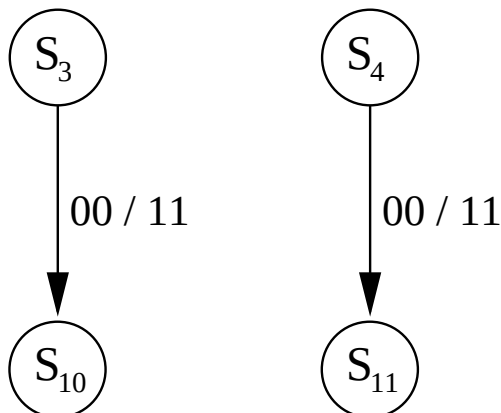
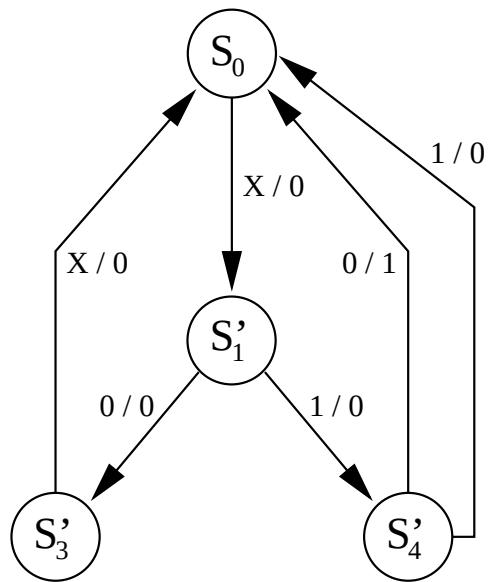


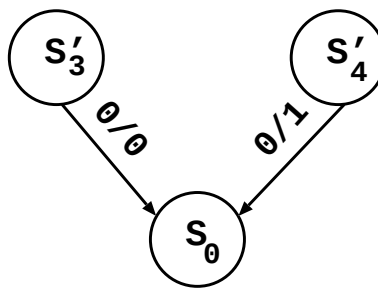
Figure 2: Partial state diagram: S_{10}, S_{11} adjacent.

LP: States with the same output for a given input should be given adjacent assignments.

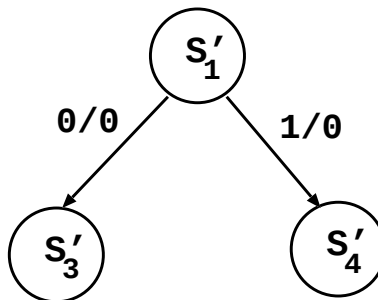




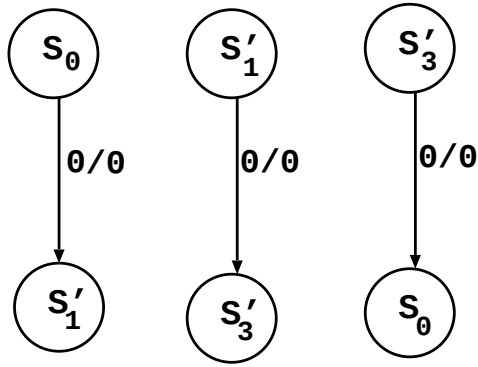
HP $\Rightarrow (S'_3, S'_4)$



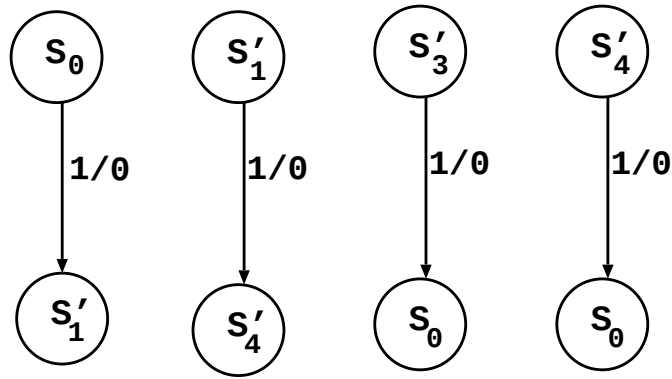
MP $\Rightarrow (S'_3, S'_4)$



LP $\Rightarrow (S_0, S'_1, S'_3)$

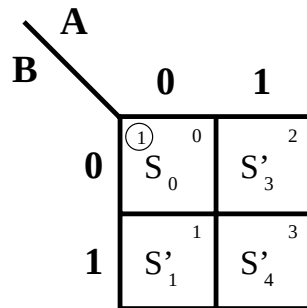


LP $\Rightarrow (S_0, S'_1, S'_3, S'_4)$



Possible Assignment:

	B	A	
S_0	0	0	(Reset)
S'_1	1	0	
S'_3	0	1	
S'_4	1	1	

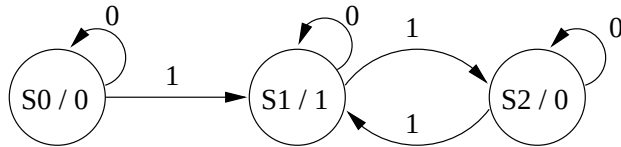


1. $S_0 = 00$ (Reset)
2. S'_3 adj to S'_4 HP,MP
3. S'_1 left over

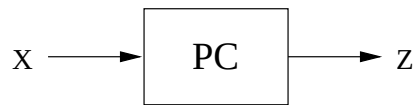
State Minimization

Exercise:

Given state diagram for parity checker below. Find a state diagram with smaller number of states that is functionally equivalent.

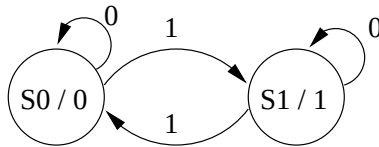


Parity Checker



Output $Z = 1$ when an odd number of 1's has been seen on X (0 otherwise).

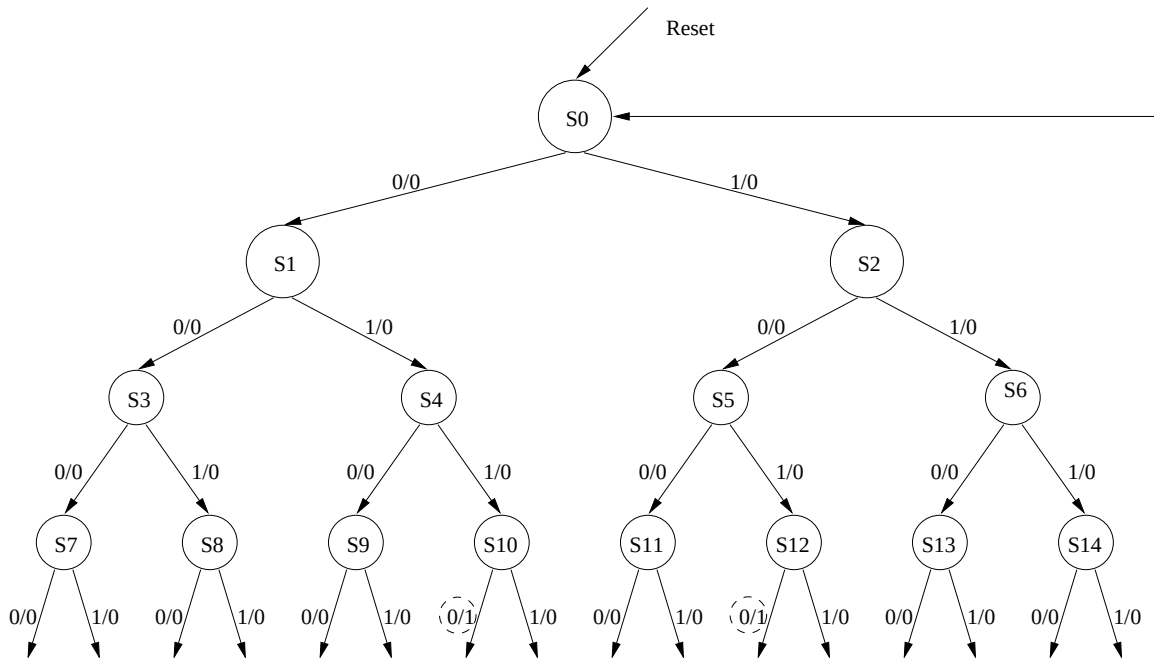
Solution



Four-bit Sequence Detector

The machine has input X and output Z. The output is asserted after each 4-bit sequence if and only if the sequence consists of the binary string 0110 or 1010.

Original State Diagram (Not minimized)



State Minimization

Example: 4-bit sequence Detector

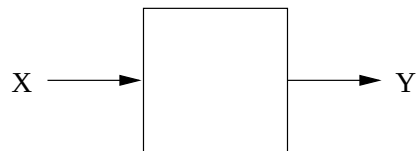


Figure 3: Recognizer block diagram: Recognizes 0110 or 1010

We presented the UNMINIMIZED STATE TABLE LAST LECTURE.

State Table (Symbolic) (New format)

Attempt to reduce the number of states, while maintaining I/O behavior of the FSM.

PS	X = 0	X = 1
S_0	$S_1,0$	$S_2,0$
S_1	$S_3,0$	$S_4,0$
S_2	$S_5,0$	$S_6,0$
S_3	$S_7,0$	$S_8,0$
S_4	$S_9,0$	$S_{10},0$
S_5	$S_{11},0$	$S_{12},0$
S_6	$S_{13},0$	$S_{14},0$
S_7	$S_0,0$	$S_0,0$
S_8	$S_0,0$	$S_0,0$
S_9	$S_0,0$	$S_0,0$
S_{10}	$S_0,1$	$S_0,0$
S_{11}	$S_0,0$	$S_0,0$
S_{12}	$S_0,1$	$S_0,0$
S_{13}	$S_0,0$	$S_0,0$
S_{14}	$S_0,0$	$S_0,0$

Use Row Matching

Find similar rows: Both NS + OUTPUTS MUST BE EQUAL.

S_{12} AND S_{10} have same rows

- $S_{12} \iff S_{10}$

New State S'_{10}

PS	NS, Z	
	X = 0	X = 1
S_0	$S_1,0$	$S_2,0$
S_1	$S_3,0$	$S_4,0$
S_2	$S_5,0$	$S_6,0$
S_3	$S_7,0$	$S_8,0$
S_4	$S_9,0$	$S'_{10},0$
S_5	$S_{11},0$	$S'_{10},0$
S_6	$S_{13},0$	$S_{14},0$
S_7	$S_0,0$	$S_0,0$
S_8	$S_0,0$	$S_0,0$
S_9	$S_0,0$	$S_0,0$
S'_{10}	$S_0,1$	$S_0,0$
S_{11}	$S_0,0$	$S_0,0$
S_{13}	$S_0,0$	$S_0,0$
S_{14}	$S_0,0$	$S_0,0$

$S_7, S_8, S_9, S_{11}, S_{13}, S_{14}$ are equivalent!

New states S'_7

PS	NS, Z	
	X = 0	X = 1
S_0	$S_{1,0}$	$S_{2,0}$
S_1	$S_{3,0}$	$S_{4,0}$
S_2	$S_{5,0}$	$S_{6,0}$
S_3	$S'_{7,0}$	$S'_{7,0}$
S_4	$S'_{7,0}$	$S'_{10,0}$
S_5	$S'_{7,0}$	$S'_{10,0}$
S_6	$S'_{7,0}$	$S'_{7,0}$
S'_7	$S_{0,0}$	$S_{0,0}$
S'_{10}	$S_{0,1}$	$S_{0,0}$

$S_3 \iff S_6$ New State S'_3

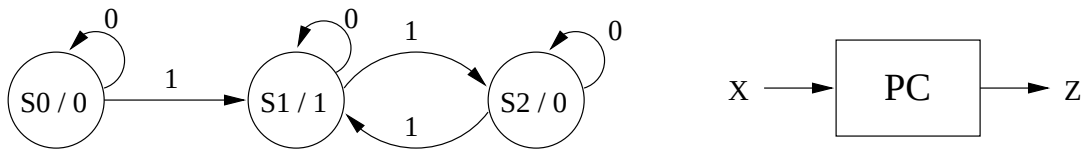
PS	NS, Z	
	X = 0	X = 1
S_0	$S_{1,0}$	$S_{2,0}$
S_1	$S'_{3,0}$	$S_{4,0}$
S_2	$S_{5,0}$	$S'_{3,0}$
S'_3	$S'_{7,0}$	$S'_{7,0}$
S_4	$S'_{7,0}$	$S'_{10,0}$
S_5	$S'_{7,0}$	$S'_{10,0}$
S'_7	$S_{0,0}$	$S_{0,0}$
S'_{10}	$S_{0,1}$	$S_{0,0}$

$S_4 \iff S_5$ New State S'_4

PS	NS, Z	
	X = 0	X = 1
S_0	$S_{1,0}$	$S_{2,0}$
S_1	$S'_{3,0}$	$S'_{4,0}$
S_2	$S'_{4,0}$	$S'_{3,0}$
S'_3	$S'_{7,0}$	$S'_{7,0}$
S'_4	$S'_{7,0}$	$S'_{10,0}$
S'_7	$S_{0,0}$	$S_{0,0}$
S'_{10}	$S_{0,1}$	$S_{0,0}$

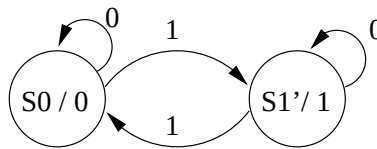
Problem with ROW Matching

Return to ODD Parity Checker.



PS	NS		OUTPUT
	X = 0	X = 1	
S_0	S_0	S_1	0
S_1	S_1	S_2	1
S_2	S_2	S_1	0

No 2 rows match, but we know that there is a solution with 2 states.



We need a better minimization technique.

We use the Implication Chart Method. Table which enumerates all pairs of states. If S_i and S_j cannot be equivalent, an X is placed in the table.

3-bit Sequence Detector Example

Circuit produces a 1 on output if 010 or 110 has occurred.

State Transition Table

PS	NS, Z	
	X = 0	X = 1
S_0	$S_1, 0$	$S_2, 0$
S_1	$S_3, 0$	$S_4, 0$
S_2	$S_5, 0$	$S_6, 0$
S_3	$S_0, 0$	$S_0, 0$
S_4	$S_0, 1$	$S_0, 0$
S_5	$S_0, 0$	$S_0, 0$
S_6	$S_0, 1$	$S_0, 0$

To combine S_i , S_j implies that their next states are also equivalent. Their outputs must also be equivalent.

1. Check outputs and fill in X's.
2. Fill in implied equivalent pairs.

3. If S_i, S_j contains an X, then cross out boxes containing $S_i - S_j$.
4. Repeat until there are no more boxes to cross out.

S_1	$S_1 - S_3$ $S_2 - S_4$				
S_2	$S_1 - S_5$ $S_2 - S_6$	$S_3 - S_5$ $S_4 - S_6$			
S_3	$S_1 - S_0$ $S_2 - S_0$	$S_3 - S_0$ $S_4 - S_0$	$S_5 - S_0$ $S_6 - S_0$		
S_4					
S_5	$S_1 - S_0$ $S_2 - S_0$	$S_3 - S_0$ $S_4 - S_0$	$S_5 - S_0$ $S_6 - S_0$	$S_0 - S_0$ $S_0 - S_0$	
S_6				$S_0 - S_0$ $S_0 - S_0$	
	S_0	S_1	S_2	S_3	S_4

S_1	$S_1 - S_3$ $S_2 - S_4$				
S_2	$S_1 - S_5$ $S_2 - S_6$	$S_3 - S_5$ $S_4 - S_6$			
S_3	$S_1 - S_0$ $S_2 - S_0$	$S_3 - S_0$ $S_4 - S_0$	$S_5 - S_0$ $S_6 - S_0$		
S_4					
S_5	$S_1 - S_0$ $S_2 - S_0$	$S_3 - S_0$ $S_4 - S_0$	$S_5 - S_0$ $S_6 - S_0$	$S_0 - S_0$ $S_0 - S_0$	
S_6				$S_0 - S_0$ $S_0 - S_0$	
	S_0	S_1	S_2	S_3	S_4

$S_1 \iff S_2$ New state S'_1
 $S_3 \iff S_5$ New state S'_3
 $S_4 \iff S_6$ New state S'_4