

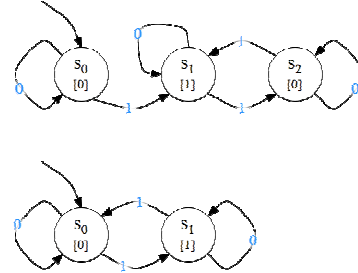
Lecture 20

- State minimization via row matching

1

FSM minimization

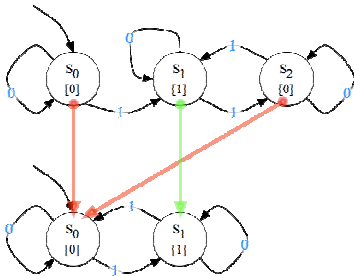
- Two simple FSMs for odd parity checking



2

Collapsing states

- Can collapse S_0 and S_2 into one state



3

FSM minimization

- Row matching
 - Easier to do by hand
 - Misses minimization opportunities
- Implication chart
 - Guaranteed to find the most reduced FSM
 - More complicated algorithm (but still relatively easy to write a program to do it)

4

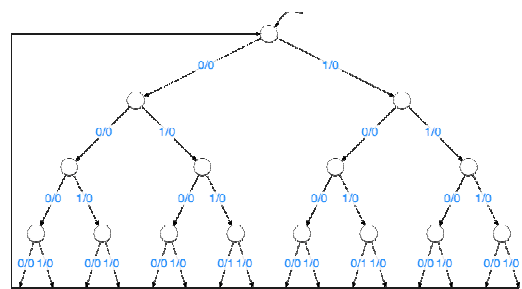
Simple problem

- Design a Mealy machine with a single bit input and a single bit output. The machine should output a 0, except once every four cycles, if the previous four inputs matched one of two patterns (0110, 1010)
- Example input/output trace:

in:	0010	0110	1100	1010	0011	...
out:	0000	0001	0000	0001	0000	...

5

... and a simple solution



6

Find matching rows

Input Sequence	Present State	Next State		Output	
		X=0	X=1	X=0	X=1
Reset	S ₀	S ₁	S ₂	0	0
0	S ₁	S ₃	S ₄	0	0
1	S ₂	S ₅	S ₆	0	0
00	S ₃	S ₇	S ₈	0	0
01	S ₄	S ₉	S ₁₀	0	0
10	S ₅	S ₁₁	S ₁₂	0	0
11	S ₆	S ₁₃	S ₁₄	0	0
000	S ₇	S ₀	S ₀	0	0
001	S ₈	S ₀	S ₀	0	0
010	S ₉	S ₀	S ₀	0	0
011	S ₁₀	S ₀	S ₀	1	0
100	S ₁₁	S ₀	S ₀	0	0
101	S ₁₂	S ₀	S ₀	1	0
110	S ₁₃	S ₀	S ₀	0	0
111	S ₁₄	S ₀	S ₀	0	0

Find matching rows

Input Sequence	Present State	Next State		Output	
		X=0	X=1	X=0	X=1
Reset	S ₀	S ₁	S ₂	0	0
0	S ₁	S ₃	S ₄	0	0
1	S ₂	S ₅	S ₆	0	0
00	S ₃	S ₇	S ₈	0	0
01	S ₄	S ₉	S ₁₀	0	0
10	S ₅	S ₁₁	S ₁₂	0	0
11	S ₆	S ₁₃	S ₁₄	0	0
000	S ₇	S ₀	S ₀	0	0
001	S ₈	S ₀	S ₀	0	0
010	S ₉	S ₀	S ₀	0	0
011	S ₁₀	S ₀	S ₀	1	0
100	S ₁₁	S ₀	S ₀	0	0
101	S ₁₂	S ₀	S ₀	1	0
110	S ₁₃	S ₀	S ₀	0	0
111	S ₁₄	S ₀	S ₀	0	0

Merge matching rows

Input Sequence	Present State	Next State		Output	
		X=0	X=1	X=0	X=1
Reset	S ₀	S ₁	S ₂	0	0
0	S ₁	S ₃	S ₄	0	0
1	S ₂	S ₅	S ₆	0	0
00	S ₃	S ₇	S ₈	0	0
01	S ₄	S ₉	S _{10'}	0	0
10	S ₅	S ₁₁	S _{10'}	0	0
11	S ₆	S ₁₃	S ₁₄	0	0
000	S ₇	S ₀	S ₀	0	0
001	S ₈	S ₀	S ₀	0	0
010	S ₉	S ₀	S ₀	0	0
011 or 101	S _{10'}	S ₀	S ₀	1	0
100	S ₁₁	S ₀	S ₀	0	0
110	S ₁₃	S ₀	S ₀	0	0
111	S ₁₄	S ₀	S ₀	0	0

Merge until no more matches

Input Sequence	Present State	Next State		Output	
		X=0	X=1	X=0	X=1
Reset	S ₀	S ₁	S ₂	0	0
0	S ₁	S ₃	S ₄	0	0
1	S ₂	S ₅	S ₆	0	0
00	S ₃	S ₇	S ₇	0	0
01	S ₄	S ₇	S _{10'}	0	0
10	S ₅	S ₇	S _{10'}	0	0
11	S ₆	S ₇	S ₇	0	0
Not (011 or 101)	S ₇	S ₀	S ₀	0	0
011 or 101	S _{10'}	S ₀	S ₀	1	0

Final state transition table

Input Sequence	Present State	Next State		Output	
		X=0	X=1	X=0	X=1
Reset	S ₀	S ₁	S ₂	0	0
0	S ₁	S ₃	S ₄	0	0
1	S ₂	S ₄	S ₃	0	0
00 or 11	S ₃	S ₇	S ₇	0	0
01 or 10	S ₄	S ₇	S _{10'}	0	0
Not (011 or 101)	S ₇	S ₀	S ₀	0	0
011 or 101	S _{10'}	S ₀	S ₀	1	0

Final FSM

