Chapter 8

Finite State Machine Optimization

8.1 Objectives

1. Analyse and design both Mealy and Moore sequential circuits with multiple inputs and multiple outputs
2. Convert between Mealy and Moore designs
3. Perform a state assignment using the guideline method
4. Reduce the number of states in a state table using row reduction and implication tables

8.2 Mealy vs Moore Finite State Machines

Definition 8.1 (Finite State Machines (FSM)). *[1, Sec 3.4] A FSM has M inputs, N outputs and k bits of unique states (2^k states).*

Definition 8.2 (Mealy FSM). *[1, Sec 3.4.3] Named after George M. Mealy, in Mealy FSM the output depends both on inputs and the states.*

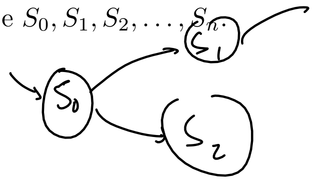
Definition 8.3 (Moore FSM). *[1, Sec 3.4.3] Named after Edward F. Moore, in Moore FSM the output depends only on the states.*

Example 8.1. *A sequential circuit has one input (X) and one output (Z). The circuit examines groups of four consecutive inputs and produces an output Z=1 if the input sequence 0010 or 0001 occurs. The sequences can overlap. Draw both Mealy and Moore timing diagrams. Find the Mealy and Moore state graph.*

Example 8.2. *A sequential circuit has one input (X) and one output (Z). The circuit examines groups of four consecutive inputs and produces an output Z=1 if the input sequence 0101 or 1001 occurs. The circuit resets after every four inputs. Draw both Mealy and Moore timing diagrams. Find the Mealy and Moore state graph.*

8.3 Full procedure for designing sequential logic circuit

1. Convert the word problem to a state transition diagram. Let the states be $S_0, S_1, S_2, \ldots, S_n$.



2. Draw state transition table with named states. For example,

Present State	Next State		Outputs	
	X = 0	X = 1	X=0	X=1
S_0	S_1	S_2	0	0
S_1	S_2	S_0	0	0
⋮	⋮	⋮	⋮	⋮

x = input

3. State reduction step: Reduce the number of required states to a minimum. Eliminate unnecessary or duplicate states.

4. State assignment step: Assign each state a binary representation. For example,

State name	State assignments ($Q_2Q_1Q_0$)
S_0	000
S_1	001
⋮	⋮

8 - possible state

cold

One-hot encoding

0000 0001 ← hot bit

0010 0000

5. Draw State assigned transition table. For example,

Inputs (X_1X_0)		Present State (Q_1Q_0)	Next State ($Q_1^+Q_0^+$)	Outputs (Z_1Z_0)	
0	0	00	01	0	0
0	0	01	10	0	0
⋮	⋮	⋮	⋮	⋮	⋮

(a) Use excitation tables to find truth tables for the combinational circuits. For example, the excitation table for J-K ff is

Q	Q^+	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

8.4 State assignment by guideline method [2, Section 8.2.5]

8.4.1 State Maps

Example 8.3. *Draw a state map for a sequential assignment of the states*

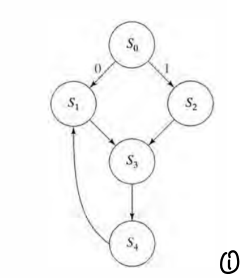


Figure 8.27 Five-state finite state machine.

5-states
3-bits

State map

$Q_2 Q_1 Q_0$

Q_2

	S_0 0	S_2 2	6	S_4 4
Q_0	S_1 1	S_3 3	7	5

Q_1

① Allows us to place neighboring states next to each other
② Resulting in a state assignment

$S_0 = 000$
$S_1 = 001$
$S_2 = 010$
$S_3 = 011$
$S_4 = 111$

8.4.2 Guideline method

Guideline method states that the following states should be adjacent in the state map according the following priorities:

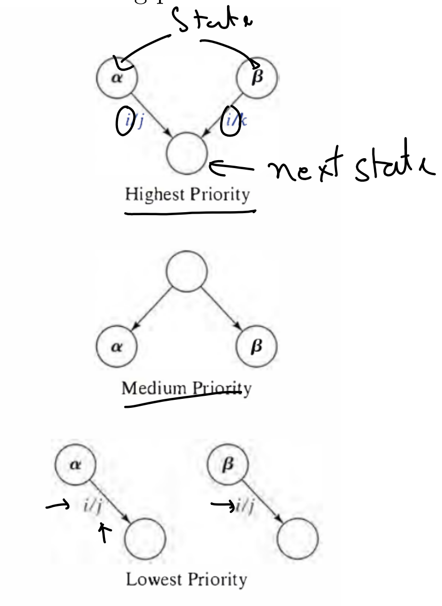


Figure 8.29 Adjacent assignment priorities.

(S_1, S_2) (S_4, S_0)

Two states which result in the same output for the same input

Example 8.4. *A state transition table is given. Find optimal state assignment by using the guideline method.*

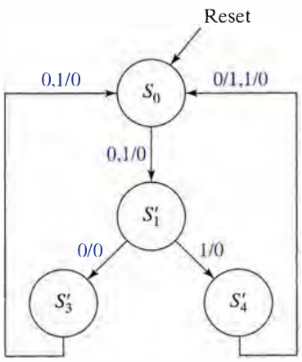
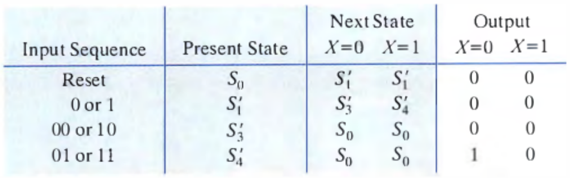
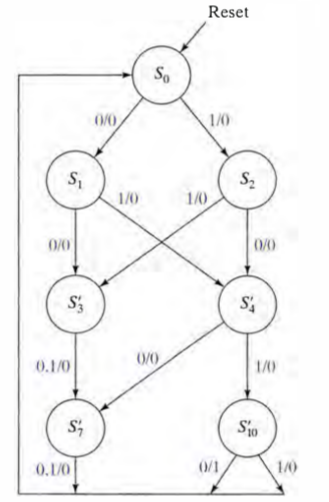


Figure 8.30 **Reduced state diagram for 3-bit sequence detector.**



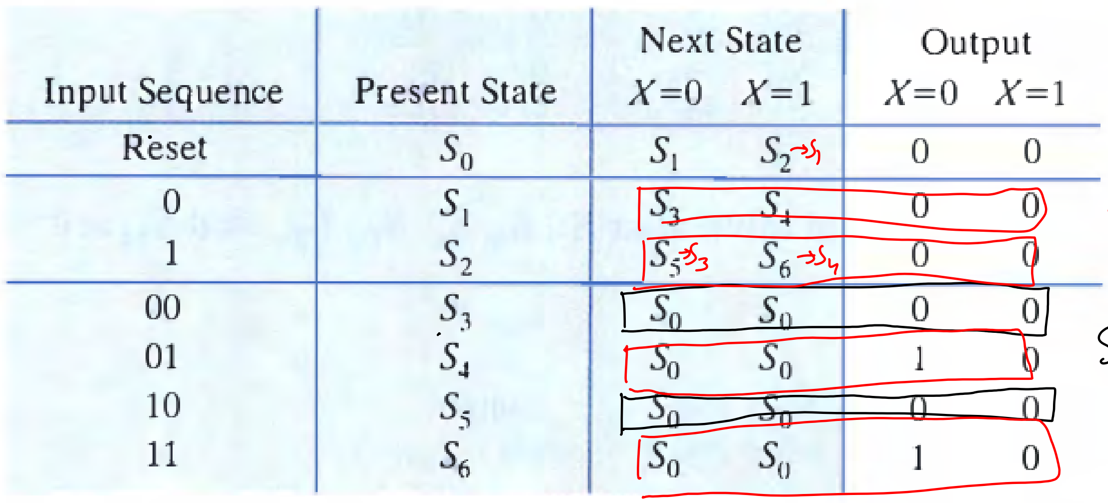
Input Sequence	Present State	Next State		Output	
		$X=0$	$X=1$	$X=0$	$X=1$
Reset	S_0	S'_1	S'_1	0	0
0 or 1	S'_1	S'_3	S'_4	0	0
00 or 10	S'_3	S_0	S_0	0	0
01 or 11	S'_4	S_0	S_0	1	0

Example 8.5. *Draw a Mealy FSM for detecting binary string 0110 or 1010. The machine returns to the reset state after each and every 4-bit sequence. Draw the state transition diagram on your own as practice problem. The state transition diagram is given here. Find optimal state assignment by using the guideline method.*



8.5 State reduction by implication chart

Example 8.6. *Design a Mealy FSM for detecting binary sequence 010 or 0110. The machine returns to reset state after each and every 3-bit sequence. For now the state transition table is given. Reduce the following state transition table*



Input Sequence	Present State	Next State $X=0$	Next State $X=1$	Output $X=0$	Output $X=1$
Reset	S_0	S_1	$S_2 \to S_1$	0	0
0	S_1	S_3	S_4	0	0
1	S_2	$S_5 \to S_3$	$S_6 \to S_4$	0	0
00	S_3	S_0	S_0	0	0
01	S_4	S_0	S_0	1	0
10	S_5	S_0	S_0	0	0
11	S_6	S_0	S_0	1	0

$S_1 \equiv S_2$

$S_3 \equiv S_5$

$S_4 \equiv S_6$

Row reduction

Implication table

If two states result in the same outputs and the same next states then those staets are equivalent and can be merged together.

8.5.1 Implication chart Summary

The algorithms for state reduction using the implication chart method consists of the following steps

1. Construct the implication chart, consisting of one square for each possible combination of states taken two at a time.

2. For each square labeled by states S_i and S_j, if the outputs of the states differ, mark the square with an X; the states are not equivalent. Otherwise, they may be equivalent. Within the square write implied pairs of equivalent next states for all input combinations.

3. Systematically advance through the squares of the implication chart. If the square labeled by states S_i, S_j contains an implied pair S_m, S_n and square S_m, S_n is marked with an X, then mark S_i, S_j with an X. Since S_m, S_n are not equivalent, neither are S_i, S_j.

4. Continue executing Step 3 until no new squares are marked with an X.

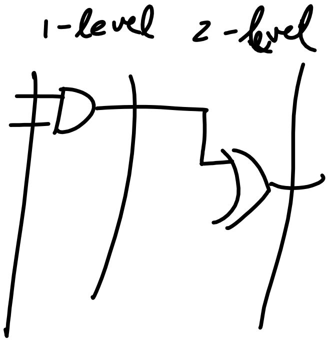
5. For each remaining unmarked square S_i, S_j, we can conclude that S_i, S_j are equivalent.

Review

8.6 Study guide

8.6.1 Midterm 1

- ☑ Binary numbers, Hexadecimal, Sign-magnitude, One's-complement and Two's complement. Conversions between them.
- ☑ Generate minterms, maxterms, SOP canonical form and POS canonical forms and convert between them
- ☑ Understand and use the laws and theorems of Boolean Algebra
- ☑ Perform algebraic simplification using Boolean algebra
- ☑ Simplification using K-maps (upto 5-input variables)
- ☑ Derive sum of product and product of sums expressions for a combinational circuit
- ☑ Convert combinational logic to NAND-NAND and NOR-NOR forms



8.6.2 Midterm 2

timing diagrams analyze

- ☑ Design Hazard-free two level circuits.
- ☑ Different between and the limitations of level-triggered latches and edge-triggered flip-flops.
- ☑ Understand the difference between synchronous and asynchronous ~~inputs~~ circuits
- ☑ Derive a state graph or state table from a word description of the problem Mealy / Moor
- ☑ Implement a design using JK, ~~SR~~, D or T flip-flops
- x ☑ Analyze a sequential circuit and derive a state-table and a state-graph
- Convert between Mealy and Moore designs
- x Partition a system into multiple state machines

8.6.3 Final (includes previous topics)

☑ Reduce the number of states in a state table using row reduction and implication tables

☑ Perform a state assignment using the guideline method

☑ Analyse and design both Mealy and Moore sequential circuits with multiple inputs and multiple outputs

☑ Design combinational circuits using multiplexers and decoders

☑ Know fan-out and noise margin

☑ Know the differences and similarities between PAL, PLA, and ROMs ([☒] and can use each for logic design)

☒ Simplification using Quine-McCluskey method

☑ Design combinational circuits for positive and negative logic

☑ Design a CMOS complex gate

☑ Compute noise margin of one device

☑ Describe how tri-state and open-collector outputs are different from totem-pole outputs.

Chapter 9

Sample Midterm 2

Student Name: Student Email:

9.1 Instructions

- There are ~~five~~ 3 problems. All problems are required.
- Maximum number of marks is 50. This exam amounts 10% toward the final grade.
- Time allowed is 50 minutes.
- In order to minimize distraction to your fellow students, you may not leave during the last 10 minutes of the examination.
- The examination is closed-book. One 8 × 11 in two-sided cheatsheet is allowed.
- Non-programmable calculators are permitted.
- Please use a pen or heavy pencil to ensure legibility. Colored pens/pencils are recommended for K-map grouping.
- Please show your work; where appropriate, marks will be awarded for proper and well-reasoned explanations.

Problem 9.1. *Table 9.1 shows the truth table for a BCD to 7-segment display. The inputs corresponding to the missing rows in the truth table should be considered as don't care. Implement segment "a" using a 4:1 MUX (multiplexer) and one other gate. (10 marks)*

Row	w_3	w_2	w_1	w_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Table 9.1: Truth table for BCD to seven-segment display. The missing combinations of inputs should be considered as don't care.

Problem 9.2. *Draw Mealy state transition diagram which investigates an input sequence X and will produce an output of Z = 1 for any input sequence ending in 0011 or 110.*

last 3 bits

Example:

~~X = 1 1 0 0 1 0 0 1 0 1 0 0 1 0~~ 1

~~Z = 0 0 0 1 0 1 1 0 1 0 0 1 0 1~~ 0

Notice that the circuit does not reset to the start state when an output of Z = 1 occurs. A minimum solution requires six states. ~~*Assign 000 to the start state.*~~ *(20 marks).*

$S_{12} = 0011$

Example

X = 1 1 0 0 0 1 1 0 0 1 0 1 1 0

Z = 0 0 1 0 0 0 1 1 0 0 0 0 0 1

0011

110

110

$S_3 \equiv S_7 = 000$ x

$S_8 = 001$ ✓

$S_9 = 110 \equiv S_1$

$S_{10} = 111 \equiv S_6$

$S_0 = * * *$

$S_1 = * * 0$

$S_2 = * * 1$

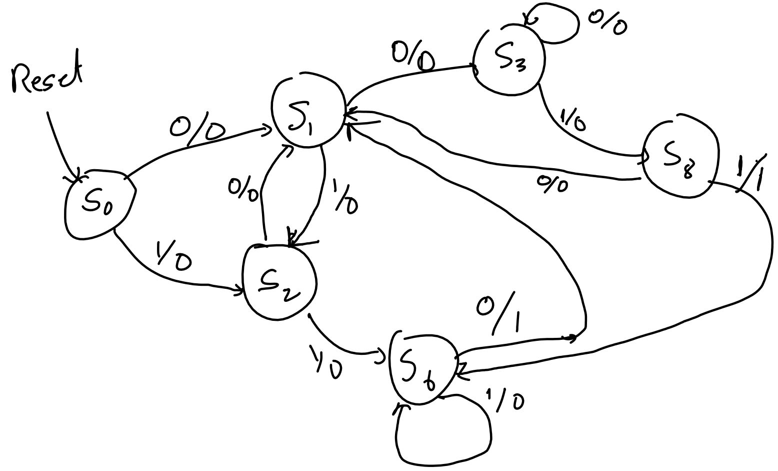
$S_3 = * 0 0$ ✓

$S_4 = * 0 1 \equiv S_2$ x

$S_5 = * 1 0 \equiv S_1$ x

$S_6 = * 1 1$ ✓

$S_{11} = 0010 \equiv S_1$



None of the interesting patterns start with either 01 or 10. So we do not need to remember the two bits we have observed so far. But the patterns we are interested in do start with 1 or 0. So we do want to remeber the 1 bit we have seen so far.

0011
110

Example

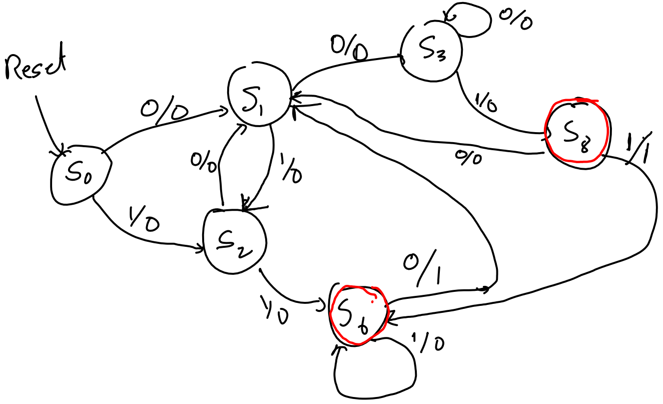
X = 1 1 0 0 0 1 1 0 0 1 0 1 1 0

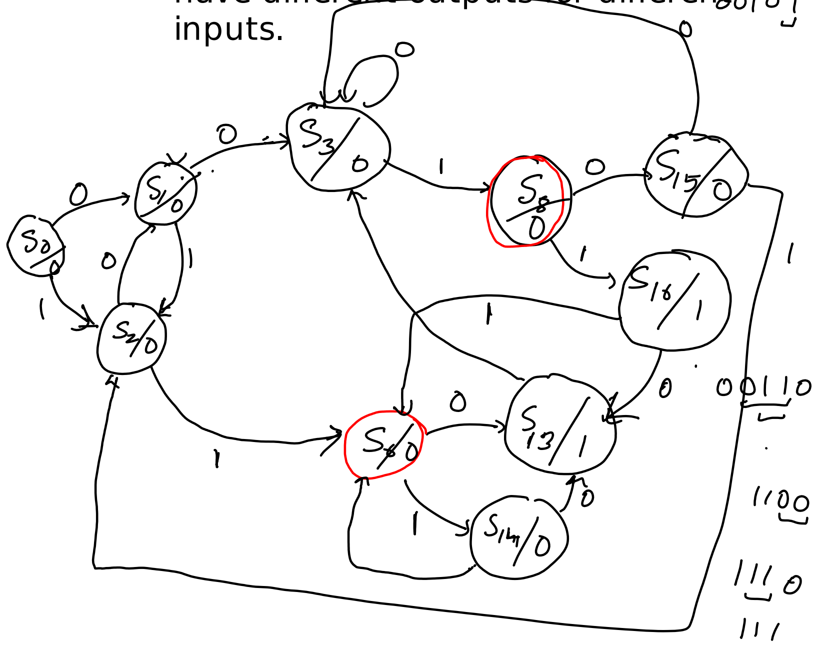
Z = 0 0 1 0 0 0 1 1 0 0 0 0 0 1

Identify the states which have different outputs for different inputs.

$S_{15} = 00100$

00101





00110

1100

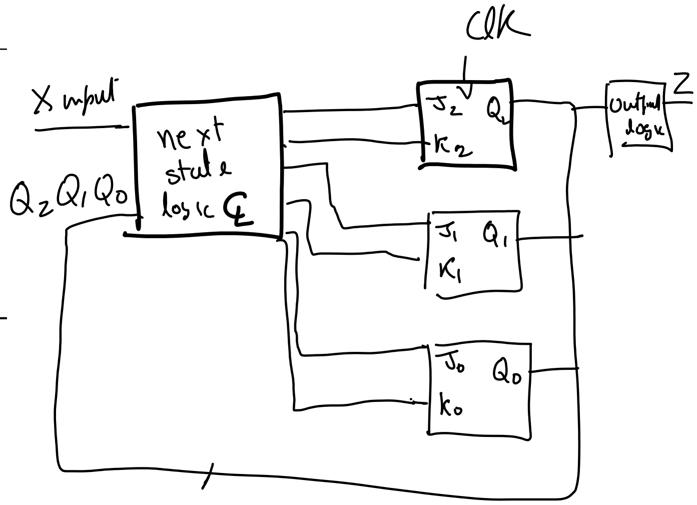
1110

111

Problem 9.3. *Find the expression for J_0 and K_0 assuming that J_0 and K_0 are inputs to the J-K flip flop that capture the state of the ~~second most significant~~ bit Q_0 of the following state encoded table. The state encoding table given with state encoding denoted as $Q_2Q_1Q_0$. (20 marks).*

Present State	*Next State*		*Output*	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
$Q_2Q_1Q_0$	$Q_2^+Q_1^+Q_0^+$	$Q_2^+Q_1^+Q_0^+$	Z	Z
000	*100*	*101*	*1*	*0*
001	*100*	*101*	*0*	*1*
010	*000*	*000*	*1*	*0*
011	*000*	*000*	*0*	*1*
100	*111*	*110*	*1*	*0*
101	*110*	*110*	*0*	*1*
110	*011*	*010*	*1*	*0*
111	*011*	*011*	*0*	*1*

X	Q_2	Q_1	Q_0	Q_0^+	J_0	K_0
0	0	0	0 →	0	0	d
0	0	0	1 →	0	d	1
0	0	1	0 →	0	0	d



Excitation table

Q_0	Q_0^+	J_0	K_0
0 →	0	0	d
0	1	1	d
1 →	0	d	1
1	1	d	0

Designing Dff using J-K ff

Mealy

table. The state encoding table given with state encoding done

Present State	Next State		Output	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
$Q_2Q_1Q_0$	$Q_2^+Q_1^+Q_0^+$	$Q_2^+Q_1^+Q_0^+$	Z	Z
000	100	101	1	0
001	100	101	0	1
010	000	000	1	0
011	000	000	0	1
100	111	110	1	0
101	110	110	0	1
110	011	010	1	0
111	011	011	0	1

Q

Q_0^+

				X	X	
		0	1	0	1	
		0	0	0	1	Q_0
Q_1		0	1	1	0	Q_0
Q_1		0	1	0	0	
			Q_2	Q_2		

Excitation table

Q_0	Q_0^+	J_0	K_0
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

J_0 K_0

J_0

			X	X	
	0	1	0	1	
	d	d	d	d	Q_0
Q_1	d	d	d	d	Q_0
Q_1	0	1	0	0	
		Q_2	Q_2		

K_0

			X	X	
	d	d	d	d	
	1	1	1	0	Q_0
Q	1	0	0	1	Q_0
Q	d	d	d	d	
		Q_2	Q_2		

when $Q_0 = 0$

$J_0 = Q_0^+$

$K_0 = d$

when $Q_0 = 1$

$K_0 = \overline{Q_0^+}$, $J_0 = d$

$$Z = f(X, Q_2, Q_1, Q_0)$$