

# Sequential logic design

Vikas Dhiman for ECE275

October 30, 2023

## 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
5. Partition a system into multiple state machines

## 2 Mealy vs Moore Finite State Machines

**Definition 1** (Finite State Machines (FSM)). [1, Sec 3.4]

FSM are sequential circuits.

1. Inputs and input alphabet: binary, multiple binary inputs
  2. Set of all states
  3. Initial state (state entered no reset)
- Definition 2** (Mealy FSM). [1, Sec 3.4.3]

4. State transition table
5. Output table  
(input, states) -> output

Output: (input, states)->output

Output only depends on both the states and inputs.

Adv: Input is reflected immediately in the output

**Definition 3** (Moore FSM). [1, Sec 3.4.3]

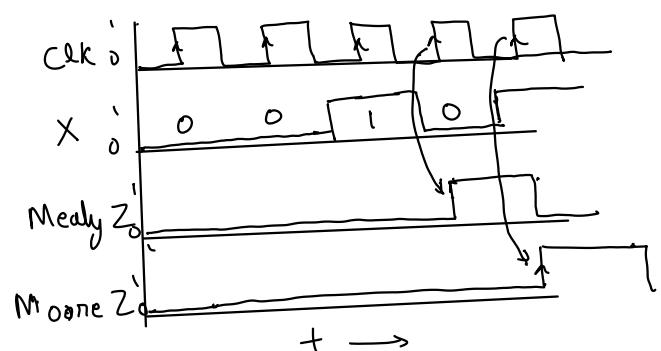
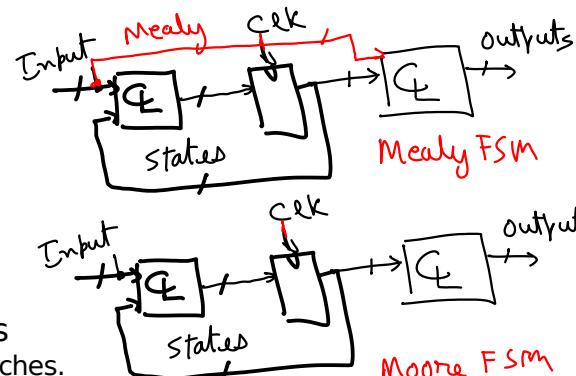
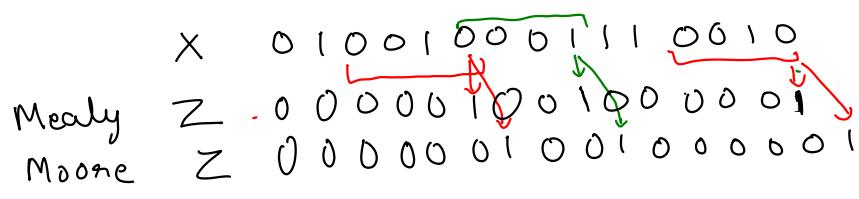
Output: (states)->output

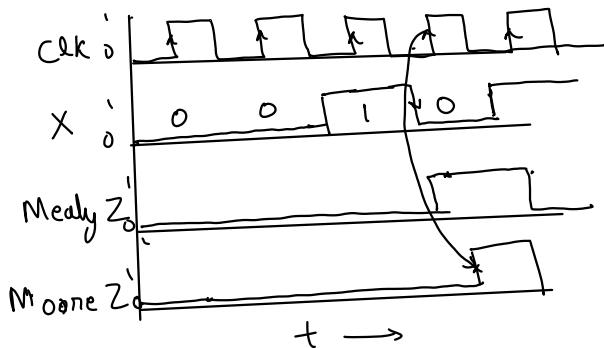
Output only depends on the states, not on the inputs

Adv: Output is synchronized with the clock. No need to worry about hazards/glitches.

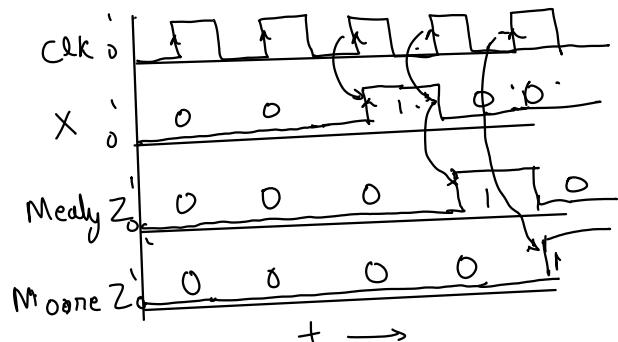
**Example 1.** A sequential circuit has one input ( $X$ ) and one output ( $Z$ ). The circuit examines groups of four consecutive inputs and produces an output  $Z=1$  if the input sequence 0010 or 0001 occurs. The sequences can overlap. Draw both Mealy and Moore timing diagrams. Find the Mealy and Moore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

Example





Asynchronous X



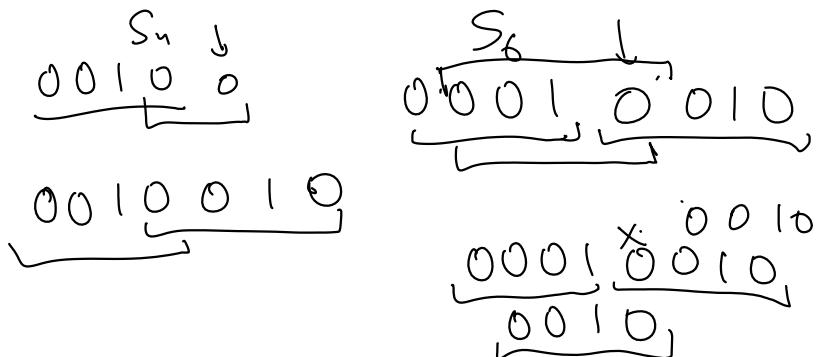
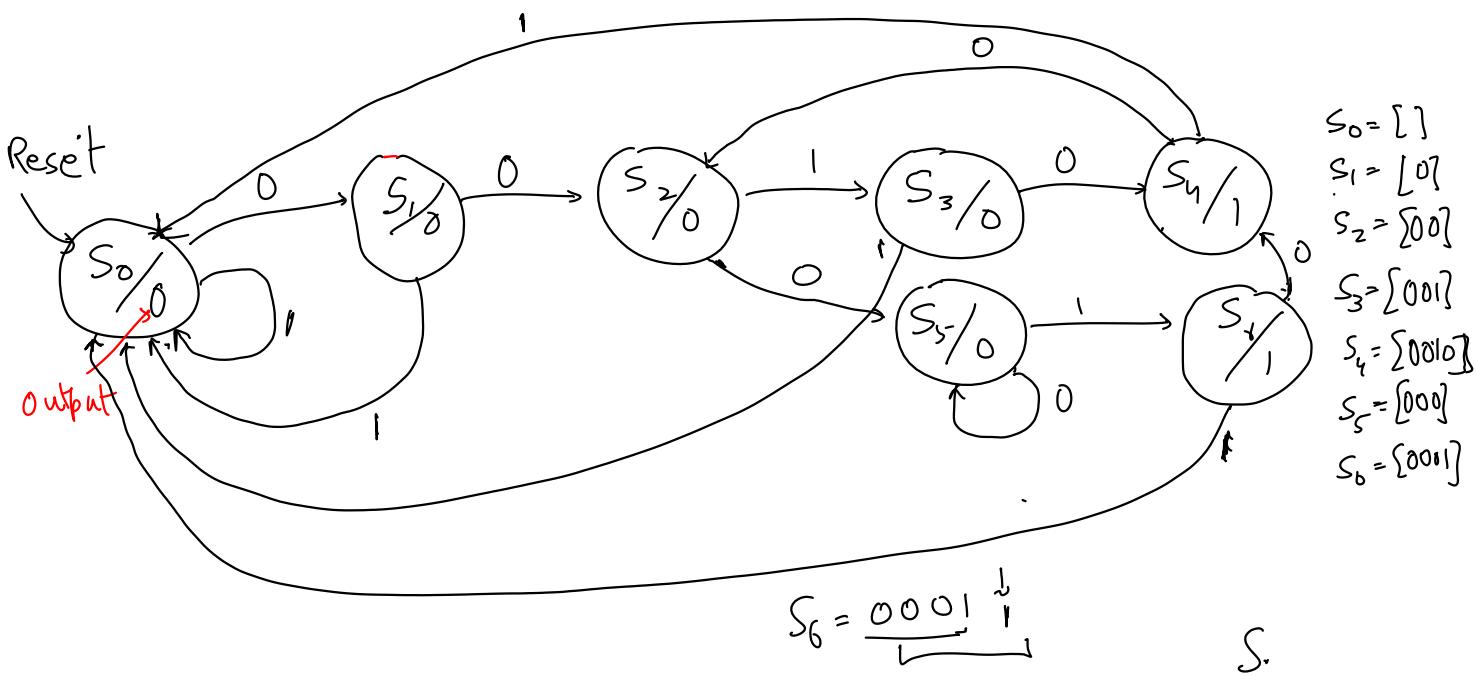
Synchronous X

State graph  $\equiv$  State transition diagram

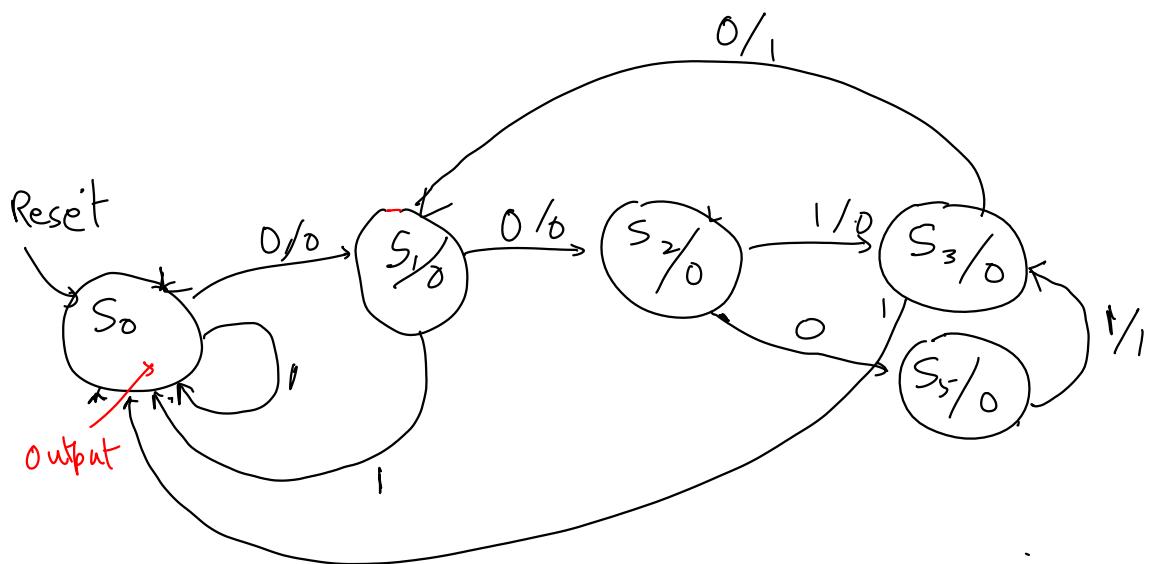
Moore state graph

Sequence detector

$\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix}$  or  $\begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix}$  } can overlap



# Mealy state graph



**Practice Problem 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 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.

### 3 Full procedure for designing sequential logic circuit

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

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 ( $Q_2 Q_1 Q_0$ )
$S_0$	000
$S_1$	001
$\vdots$	$\vdots$

5. Draw State assigned transition table. For example,

Inputs ( $X_1 X_0$ )	Present State ( $Q_1 Q_0$ )	Next State ( $Q_1^+ Q_0^+$ )	Outputs ( $Z_1 Z_0$ )
0 0	00	01	0 0
0 0	01	10	0 0
$\vdots$ $\vdots$	$\vdots$	$\vdots$	$\vdots$ $\vdots$

- (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

## 4 State assignment by guideline method [2, Section 8.2.5]

### 4.1 State Maps

**Example 2.** Draw a state map for a sequential assignment of the states

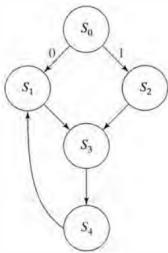


Figure 8.27 Five-state finite state machine.

### 4.2 Guideline method

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

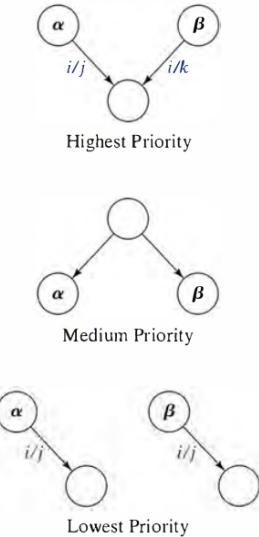


Figure 8.29 Adjacent assignment priorities.

**Example 3.** A state transition table is given. Find optimal state assignment by using the guideline

method.

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

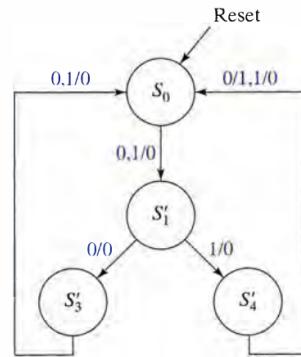
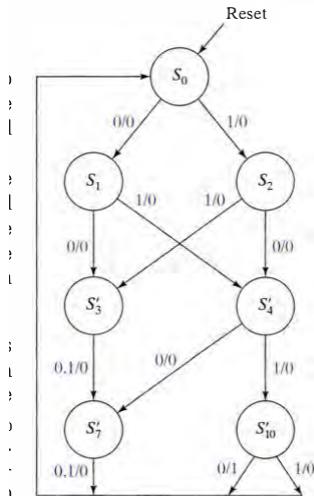


Figure 8.30 Reduced state diagram for 3-bit sequence detector.

**Example 4.** 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.



## 5 State reduction by implication chart

**Example 5.** 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		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	$S_3$	$S_0$	$S_0$	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

## 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.

## References

- [1] Sarah L Harris and David Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2022.
- [2] Randy Katz and Gaetano Barriello. *Contemporary Logic Design*. Prentice Hall, 2004.

Ask for reference PDF if you need it.

Solution to  
11/28 example State transition table (with assignments) ①

Present state	Next state		Output $\rightarrow \emptyset$
	$x=0$	$x=1$	
$Q_2\ Q_1\ Q_0$	$Q_2^+\ Q_1^+\ Q_0^+$	$Q_2^+\ Q_1^+\ Q_0^+$	$Z$
0 0 0	0 0 1	0 0 0	0
0 0 1	0 1 0	0 0 0	0
0 1 0	0 1 1	1 0 0	0
0 1 1	0 1 1	1 0 1	0
1 0 0	1 1 0	0 0 0	0
1 0 1	1 1 0	0 0 0	1
1 1 0	0 1 0	0 0 0	1
1 1 1	d d d	d d d	d

J-K ff excitation table

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

JK characteristic table

$J$	$K$	$Q$	$Q^+$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1

One way is draw Truth table for each  $J_2 K_2$ ,  $J, K$ , and  $J_0 K_0$

A shortcut is to draw K-maps for  $Q_2^+, Q_1^+, Q_0^+$

and then convert them to  $J_2 K_2$ ,  $J, K$ ,  $J_0 K_0$

~~with faults~~ K-maps

(2)

$Q_2^+$	$Q_2 Q_1$	$Q_0 X$	$Q_2$
0	0	0	12
0	1	0	13 0
0	1	d	15 11
0	0	d	14 10

replace  
 $Q_2 = 1$  with  
 $J_2 = d$

replace  $Q_2 = 0$  with  $K_2 = d$   
and flip  $Q_2^+$

$J_2$	$Q_2 Q_1$	$Q_2$
0	0	d d
0	1	d d
0	0	d d

$$J_2 = Q_1 \cdot X$$

$K_2$	$Q_2 Q_1$	$Q_2$
d	d	1 0
d	d	1 1
d	d	d
d	d	d

$$K_2 = X + Q_1$$

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

replace  
 $Q_1 = 1$   
with  $J_1 = d$

replace  $Q_1 = 0$  with  $K_1 = d$   
and flip rest

JK excitation			
$Q$	$Q^*$	$J$	$K$
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	d

$J_1 Q_2 Q_1$	$Q_2$	
$Q_0 X$		
0	d	d
0	d	d
0	d	d
1	d	d

$$J_1 = Q_0 \cdot \bar{X} + Q_1 \cdot \bar{X}$$

$K_1 Q_2 Q_1$	$Q_2$	
$Q_0 X$		
d	0	0
d	1	1
d	1	d
d	0	d

$$K_1 = X$$

(4)

$Q_0$	$Q_2 Q_1$	$\overline{Q_2}$	
$Q_0 X$		1	1
$Q_0$		0	0
		0	0
		0	1
		0	1

] X

replace  $Q_0 = 1$   
with  $J_0 = d$

replace  $Q_0 = 0$  with  $K_0 = d$   
and flip rest

$J_0$	$Q_2 Q_1$	$\overline{Q_2}$	
$Q_0 X$		1	1
$Q_0$		0	0
		0	0
		d	d
		d	d
		d	d

] X

$K_0$	$Q_2 Q_1$	$\overline{Q_2}$	
$Q_0 X$		d	d
$Q_0$		d	d
		1	0
		1	0

] X

$$J_0 = \bar{Q}_2 \cdot \bar{X}$$

$$K_0 = \bar{Q}_1$$

$Z$	$Q_2 Q_1$	$\overline{Q_2}$	
$Q_0$		0	1
$Q_0$		0	0
		0	1
		d	1

$$Z = Q_2 \cdot Q_1 + Q_2 \cdot Q_0$$

(5)

