Digital Logic Design
ENEE 244-010x

Lecture 23

Announcements

- Homework 9 up on course webpage, due on Monday, 12/7
  - Final homework assignment
- Please fill out Course Evaluations online.
  - Class time on Monday, 12/7
  - Make sure to bring in laptop, phone, etc.

Agenda

- Last Time:
  - State Table Reduction (7.4)
  - The State Assignment Problem (7.5)

- This Time:
  - Finish The State Assignment Problem (7.5)
  - Completing the Design of Clocked Synchronous Sequential Networks (7.6)

Modeling clocked synchronous sequential network behavior

- Approach for the synthesis of clocked synchronous sequential networks:
  - State table/state diagram is constructed from word specifications.
  - State reduction technique to obtain a state table with minimum number of states.
  - Transition table is formed by coding the states of the state table.
  - Excitation table is constructed based on the flip-flop types to be used.
  - From the excitation table, the excitation and output expressions for the network are determined.
  - Finally, the logic diagram is drawn.

Last time we left off. . .

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignment based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	**1**	**0**	**1**
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(*a*)

Next Step:

Constructing Transition Table from State Table

- A binary code representation for the states of the state table is selected.
- This is referred to as the state-assignment problem.
- Different state assignments result in realizations of different costs.
  - We want to find a state assignment that minimizes the cost of the network realization.

State Assignment

- If there are s states to be coded, the minimum number of binary digits p required is the smallest integer greater than or equal to the base-2 logarithm of s.
- This guarantees minimal number of flip-flops but not necessarily minimum cost realization.
- Even using the minimum binary digits the state assignment problem is not necessarily simple.
  – There are $2^p!/(2^p - s)!$ Ways of assignming a unique binary code of p digits to the s states.
  – For a six-row state table in which 3 binary digits are used to code each state, there are 20,160 different state assignemnts.

Simplest Approach

- Use the first s binary integers as the binary-code representation of the s states.

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignment based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(*a*)

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*$A \rightarrow 000$	000	001	0	0
$B \rightarrow 001$	001	010	0	0
$C \rightarrow 010$	011	100	0	0
$D \rightarrow 011$	101	110	1	0
$E \rightarrow 100$	010	001	0	1
$F \rightarrow 101$	011	111	1	0
$G \rightarrow 110$	001	010	0	1
$H \rightarrow 111$	101	110	0	0

(*b*)

Guidelines for Obtaining State Assignments

- Define two states as being adjacent if their binary codes differ in exactly one bit.
- Two input combinations are adjacent if they differ in exactly one bit.

Guidelines for Obtaining State Assignments

- Rule I: Two or more present states that have the same next state for a given input combination should be made adjacent.
- Rule II: For any present state and two adjacent input combinations, the two next states should be made adjacent.
- Rule III: Two or more present states that produce the same output symbol, for a given input combination should be made adjacent (only needs to be done for one of the two output symbols).

Rationale for Guidelines

- n input variables p state variables.
- Consider $(n + p)$-variable K-maps for each bit of next state and output.
- Rule I: provide for large subcubes on K-map by causing identical entries to appear in adjacent cells.
- Rule II: Cells in K-map will be the same for $p - 1$ of the maps corresponding to bits of the state.
- Rule III: Does to the output maps what Rule I does to the next-state maps.

Example

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignment based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(*a*)

- State B is the next state for both present states B, G when $x = 0$. Rule 1: B, G should be adjacent.
- States C, F should be coded as adjacent states since their next states are both state D.
- States D, H should be coded as adjacent states since their next states are both F.

Rule I: $(B, G)(2\times), (C, F), (D, H)(2\times), (A, E)$
$(2\times)$ indicates that the recommended adjacency conditions appear twice and should be given higher priority than those that appear only once.

Example

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignment based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	**1**	**0**	**1**
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(*a*)

Next consider Rule II:

- Since $x = 0, x = 1$ are adjacent, the next-state pair for each present state should be made adjacent according to Rule II.

Rule II: $(A, B), (B, C)(3\times), (D, E), (F, G)(2\times), (D, H)$

Example

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignment based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	**1**	**0**	**1**
*A	A	B	0	0
B	B	C	0	0
C	D	E	0	0
D	F	G	1	0
E	C	B	0	1
F	D	H	1	0
G	B	C	0	1
H	F	G	0	0

(*a*)

Next consider Rule III:
Look at present states that produce output symbol 1 on the same input.

Rule III: $(D, F), (E, G)$

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A			

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B		

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B		
	G		

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B	C	
	G		

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B	C	
E	G		

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B	C	
E	G		D

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B	C	
E	G	F	D

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

I.S.

A	B	C	H
E	G	F	D

- (B,G) 2x
- (C,F)
- (D,H) 2x
- (A,E)

- (A,B)
- (B,C) 3x
- (D,E)
- (F,G) 2x
- (D,H)

- (D,F)
- (E,G)

State Assignment Map

- K-map for the state variables in which each cell of the map denotes a combination of the binary digits that can be assigned to a state of the sequential network.

Q_1 \ Q_2Q_3	00	01	11	10
0	A	B	C	H
1	E	G	F	D

Figure 7.23 A state-assignment map for the state table of Table 7.17*a*.

Transition Table

- Using the state assignment map and state table, a transition table is constructed.

Q_1 \ Q_2Q_3	00	01	11	10
0	*A*	*B*	*C*	*H*
1	*E*	*G*	*F*	*D*

Figure 7.23 A state-assignment map for the state table of Table 7.17*a*.

Table 7.17 Illustrations of state assignments. (*a*) State table. (*b*) Transition table for state assignment in binary order. (*c*) Transition table for state assignment based on guidelines

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	**1**	**0**	**1**
**A*	*A*	*B*	0	0
B	*B*	*C*	0	0
C	*D*	*E*	0	0
D	*F*	*G*	1	0
E	*C*	*B*	0	1
F	*D*	*H*	1	0
G	*B*	*C*	0	1
H	*F*	*G*	0	0

(*a*)

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
*$A \rightarrow 000$	000	001	0	0
$B \rightarrow 001$	001	011	0	0
$C \rightarrow 011$	110	100	0	0
$D \rightarrow 110$	111	101	1	0
$E \rightarrow 100$	011	001	0	1
$F \rightarrow 111$	110	010	1	0
$G \rightarrow 101$	001	011	0	1
$H \rightarrow 010$	111	101	0	0

Unused States

- With p bits, the number of states s that can be coded is given by
$$2^{p-1} < s \leq 2^{p}$$
- In general, when coding s states with p bits some binary combinations are not assigned to any state.

Unused States

- Approach 1:
  - The corresponding entries in the K-maps are don’t cares.
  - This provides greater flexibility when obtaining minimal expressions for next-state and output functions.
- Approach 2:
  - The network may enter one of the unused states (when first turned on, due to noise, hardware failure, etc.)
  - It may be desirable that the network go to some well-defined state at the end of the clock period.
  - Next state entries for each of the unused states should be specified.

Illustrating Approach 1

Present state	Next state		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
$*A$	A	B	0	0
B	C	D	1	0
C	A	D	0	1
D	E	A	1	1
E	C	B	0	0

(*a*)

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
$A \rightarrow$ 000	000	001	0	0
$B \rightarrow$ 001	010	011	1	0
$C \rightarrow$ 010	000	011	0	1
$D \rightarrow$ 011	100	000	1	1
$E \rightarrow$ 100	010	001	0	0
101	–	–	–	–
110	–	–	–	–
111	–	–	–	–

(*b*)

Illustrating Approach 2

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
$A \rightarrow$ 000	000	001	0	0
$B \rightarrow$ 001	010	011	1	0
$C \rightarrow$ 010	000	011	0	1
$D \rightarrow$ 011	100	000	1	1
$E \rightarrow$ 100	010	001	0	0
101	000	000	0	0
110	000	000	0	0
111	000	000	0	0

(d)

Completing the Design

- Choose which type of clocked flip-flops should be used for memory.
- Depending on this choice, appropriate excitation signals must be generated by the combinational logic that precedes the input terminals of the flip-flops.
- Excitation table can be constructed from transition table once flip-flop type is selected.

Application tables for Flip-Flops

Table 7.18 Application tables. (*a*) *D* flip-flop. (*b*) *JK* flip-flop. (*c*) *T* flip-flop. (*d*) *SR* flip-flop

Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

(*a*)

Q	Q^+	J	K
0	0	0	–
0	1	1	–
1	0	–	1
1	1	–	0

(*b*)

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

(*c*)

Q	Q^+	S	R
0	0	0	–
0	1	1	0
1	0	0	1
1	1	–	0

(*d*)

From Transition Table to Excitation Table

Present state ($Q_1Q_2Q_3$)	Next state ($Q_1^+Q_2^+Q_3^+$)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
$A \rightarrow$ 000	000	001	0	0
$B \rightarrow$ 001	010	011	1	0
$C \rightarrow$ 010	000	011	0	1
$D \rightarrow$ 011	100	000	1	1
$E \rightarrow$ 100	010	001	0	0
101	–	–	–	–
110	–	–	–	–
111	–	–	–	–

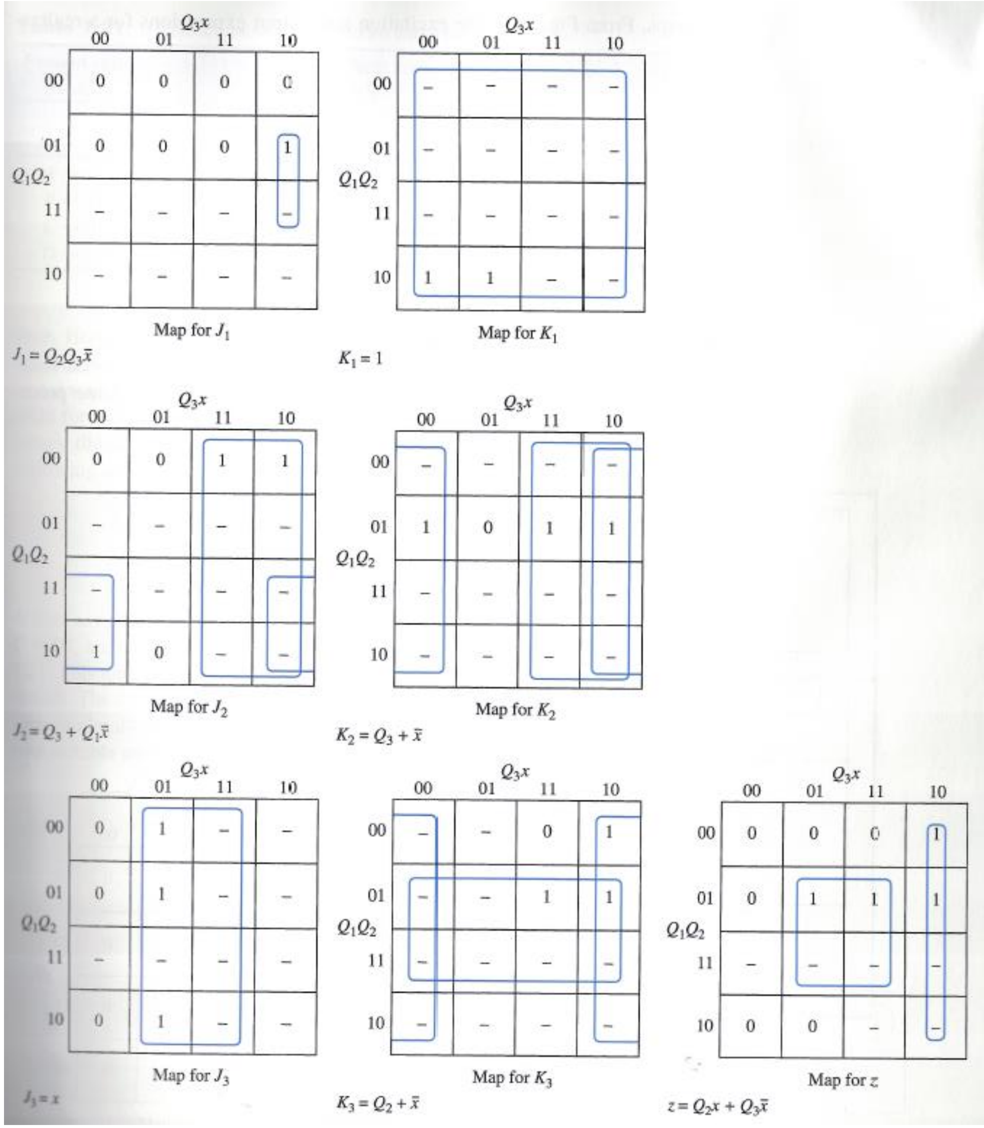
(b)

Q	Q^+	J	K
0	0	0	–
0	1	1	–
1	0	–	1
1	1	–	0

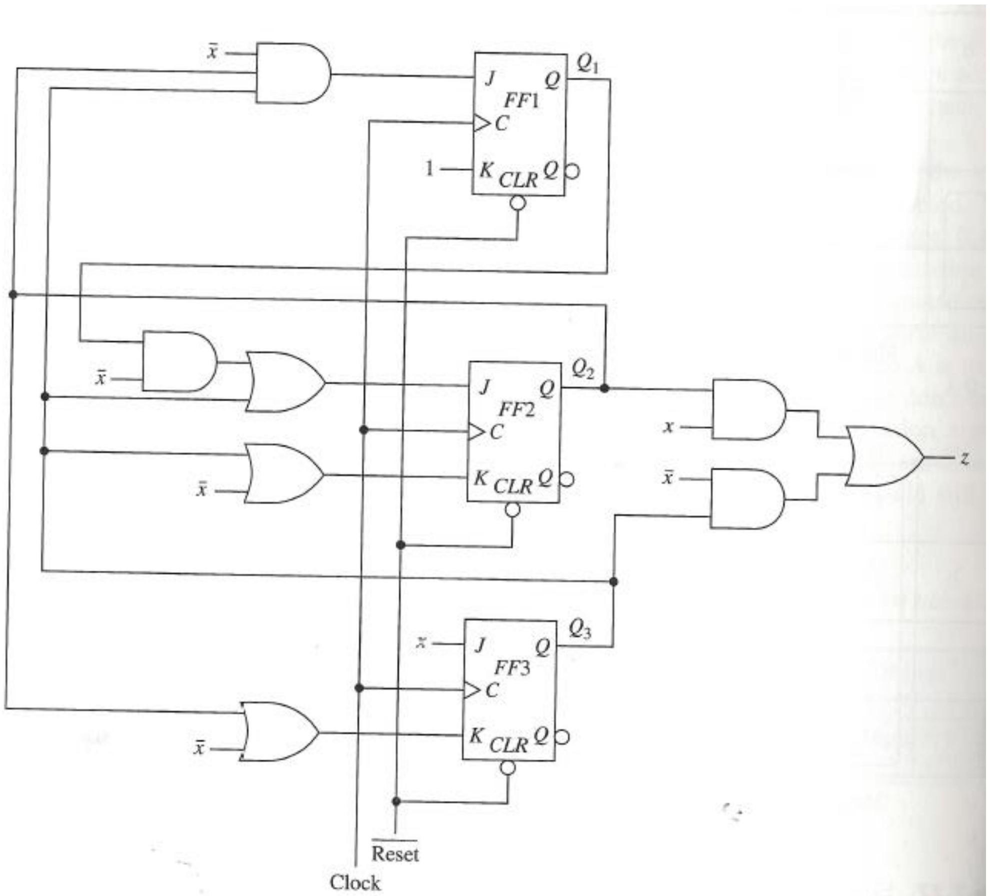
(b)

Present state ($Q_1Q_2Q_3$)	Excitation (J_1K_1, J_2K_2, J_3K_3)		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
000	0–, 0–, 0–	0–, 0–, 1–	0	0
001	0–, 1–, –1	0–, 1–, –0	1	0
010	0–, –1, 0–	0–, –0, 1–	0	1
011	1–, –1, –1	0–, –1, –1	1	1
100	–1, 1–, 0–	–1, 0–, 1–	0	0

K-Maps for Excitation and Output



Completing the Design with D-flip-flops



$$J_1 = Q_2 Q_3 \bar{x}$$
$$K_1 = 1$$
$$J_2 = Q_3 + Q_1 \bar{x}$$
$$K_2 = Q_3 + \bar{x}$$
$$J_3 = x$$
$$K_3 = Q_2 + \bar{x}$$
$$z = Q_2 x + Q_3 \bar{x}$$