EEC180A DIGITAL SYSTEMS I Winter, 2006.

Solutions for Homework # 8

15.1 (a) Implication chart after one pass:

	A	B	C	D	E	F	G
B	X						
C	X	F-H					
D	X	C-E	B-E F-H				
E	X	C-D A-F	B-D A-H	A-F			
F	~~A-B~~ ~~E-F~~	X	X	X	X		
G	X	C-D F-H	B-D	D-E F-H	A-H	X	
H	E-G	X	X	X	X	~~B-H~~ ~~F-G~~	X

Reduced state table:

State	Next State $X=0$	Next State $X=1$	Output $X=0$	Output $X=1$
A	A	C	1	0
B	C	F	0	0
C	B	A	0	0
F	B	F	1	0

Complete implication chart

	A	B	C	D	E	F	G
B	X						
C	X	~~F-H~~					
D	X	C-E	~~B-E~~ ~~F-H~~				
E	X	~~C-D~~ ~~A-F~~	B-D A-H	~~A-F~~			
F	~~A-B~~ ~~E-F~~	X	X	X	X		
G	X	~~C-D~~ ~~F-H~~	B-D	~~D-E~~ ~~F-H~~	A-H	X	
H	E-G	X	X	X	X	~~B-H~~ ~~F-G~~	X

$A \equiv H$
$C \equiv E \equiv G$
$B \equiv D$

15.1 (b) $B \not\equiv C$ because $F \not\equiv H$, (and also because $C \not\equiv D$)
$F \not\equiv H$ because $B \not\equiv H$, (and also because $F \not\equiv G$), and
$B \not\equiv H$ because the output differs for $X = 0$.
So use the sequence $\underline{X} = 100$.

Input:	X:	1	0	0
Starting in B:	Z:	0	1	0
	State: (B)	F	B	
Starting in G:	Z:	0	1	1
	State: (G)	H	H	

So $\lambda_1(B, 100) = 010 \neq 011 = \lambda_2(G, 100)$, and $B \not\equiv G$.
(Alternative: $\lambda_1(B, 110) = 001 \neq 000 = \lambda_2(G, 110)$.
Also, $\lambda_1(B, 00101) \not\equiv \lambda_2(G, 00101)$, but this requires an $\underline{X}$ of length 5.

15.2

	a	b	c	d	e	f	g	h
b	c-e							
c	X	X						
d	~~e-h~~ ~~e-a~~	~~c-h~~ ~~e-a~~	X					
e	X	X	h-f	X				
f	X	X	~~i-e~~ ~~h-g~~	X	~~i-e~~ ~~f-g~~			
g	~~e-h~~ ~~e-b~~	~~c-h~~ ~~e-b~~	X	a-b	X	X		
h	X	X	~~c-i~~ ~~d-h~~	X	~~c-i~~ ~~f-d~~	c-e g-d	X	
i	~~e-f~~ ~~e-b~~	~~c-f~~ ~~b-e~~	X	h-f a-b	X	X	h-f	X

$a \equiv b$
$c \equiv e$
$h \equiv f$
$d \equiv g \equiv i$

See FLD p. 660 for reduced state table.

15.3

S_0	$S_5 - a$ $S_1 - b$	~~$S_5 - a$~~ ~~$S_1 - c$~~	
S_1	~~$S_5 - a$~~ ~~$S_6 - b$~~	$S_5 - a$ $S_6 - c$	
S_2	~~$S_2 - a$~~ ~~$S_6 - b$~~	~~$S_2 - a$~~ ~~$S_6 - c$~~	
S_3			$S_0 - a$ $S_1 - b$
S_4	~~$S_4 - a$~~ ~~$S_1 - b$~~	~~$S_4 - a$~~ ~~$S_1 - c$~~	
S_5	$S_0 - a$ $S_1 - b$	~~$S_0 - a$~~ ~~$S_1 - c$~~	
S_6			$S_5 - a$ $S_1 - b$
	a	b	c

$S_0 \equiv a$
$S_1 \equiv b$
$S_3 \equiv c$
$S_5 \equiv a$
$S_6 \equiv c$
S_2 and S_4 have no equivalent states.

15.3 (a) $a \equiv S_0, S_5$
$b \equiv S_1$
$c \equiv S_3, S_6$

Since S_2 and S_4 do not have corresponding states, the circuits are *not* equivalent.

15.3 (b) Starting from S_0, it is not possible to reach S_2 or S_4. So then the circuits would perform the same.

15.4 (a)

X3 Q \ X1 X2	00	01	11	10
00	0	1	0	1
01	0	1	1	0
11	1	0	0	1
10	0	1	0	1

D = X2'X3 Q + X1'X2 Q' + X1 X2'Q' + X2 X3'Q
$Z = Q$

15.4 (b)

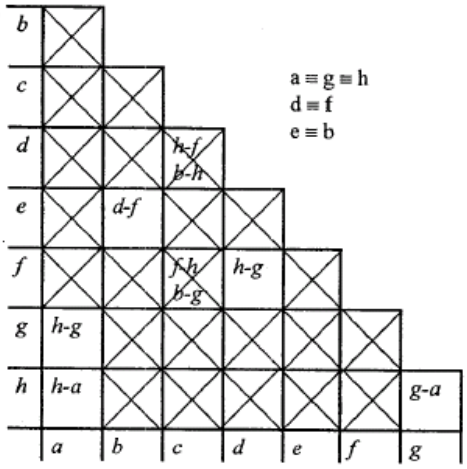
X3 Q \ X1 X2	00	01	11	10
00	X	0	X	0
01	1	0	0	1
11	0	1	1	0
10	X	0	0	0

R = X2 X3 Q + X2'X3'Q
$Z = Q$

X3 Q \ X1 X2	00	01	11	10
00	0	1	0	1
01	0	X	X	0
11	X	0	0	X
10	0	1	0	1

S = X1'X2 Q' + X1 X2'Q'

15.10 (a)



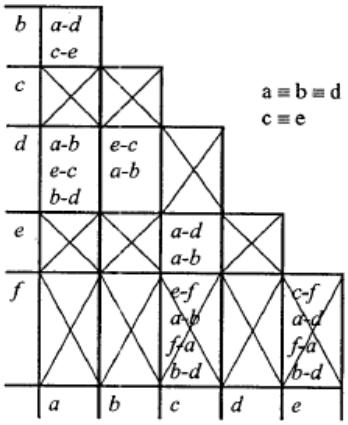
State	Next State $X=0$	Next State $X=1$	Output $X=0$	Output $X=1$
a	a	c	1	0
b	c	d	0	1
c	a	b	0	0
d	d	a	0	0

15.10 (b) Input: 00

Output starting in state c: 01 (state $c \xrightarrow{0}$ state $a \xrightarrow{0}$ state a)

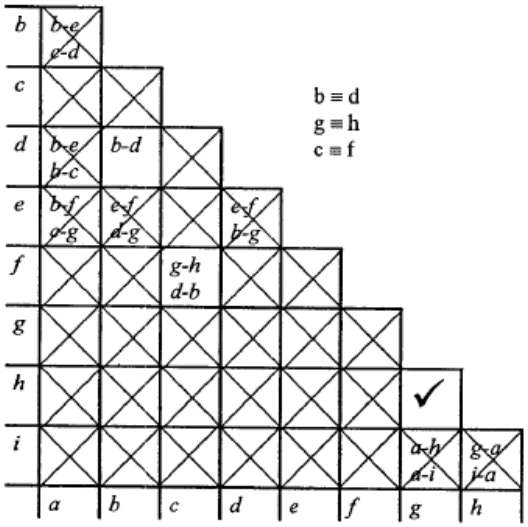
Output starting in state d: 00 (state $d \xrightarrow{0}$ state $d \xrightarrow{0}$ state d)

15.11 (a)



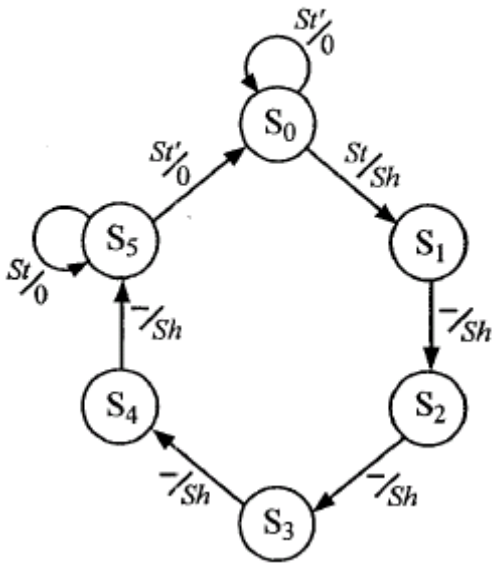
Present State	Next State 00	01	11	10	Z
a	a	c	c	a	0
c	c	a	f	a	1
f	f	a	a	a	1

15.11 (b)



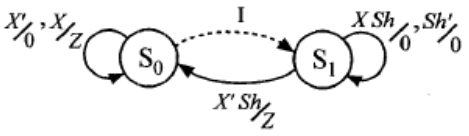
State	Next State $X=0$	Next State $X=1$	Z $X=0$	Z $X=1$
a	b	c	1	0
b	e	b	1	0
c	g	b	1	1
e	c	g	1	0
g	g	i	0	1
i	a	a	0	1

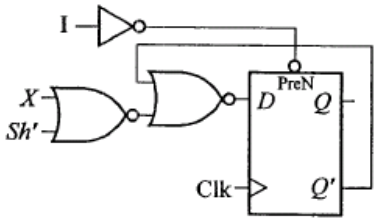
18.3 *See FLD p. 669 for circuit. Notice that the Q output of the flip-flop is b_{in}, while the D input is b_{out}.*

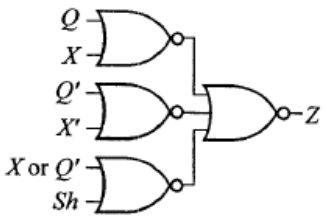


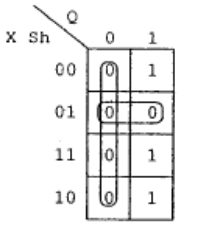
18.9 The ONE ADDER is similar to a serial adder, except that there is only one input. This means that the carry will be added to X. Thus, if the carry flip-flop is initially set to 1, 1 will be added to the input. The signal I can be used to preset the carry flip-flop to 1.

Let S_0 represent $Carry = 0$, and let S_1 represent $Carry = 1$. The state graph is as follows:

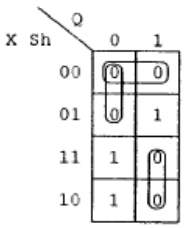








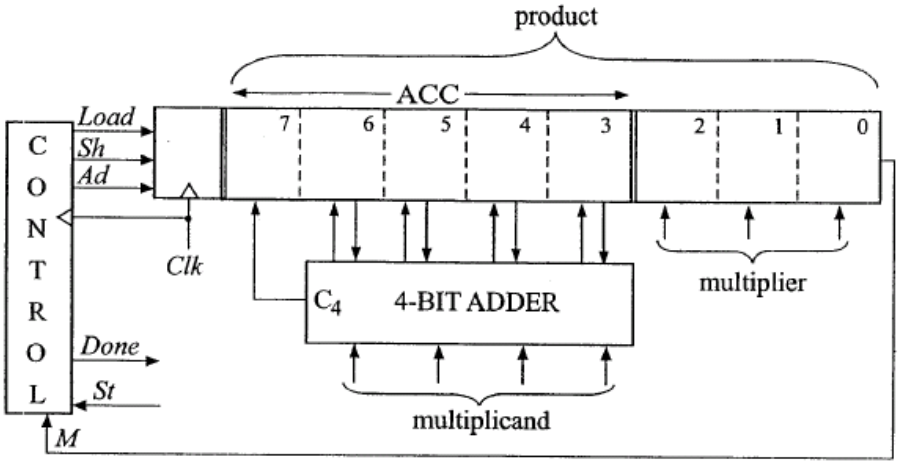
$Q^+ = Q\,(Sh' + X)$



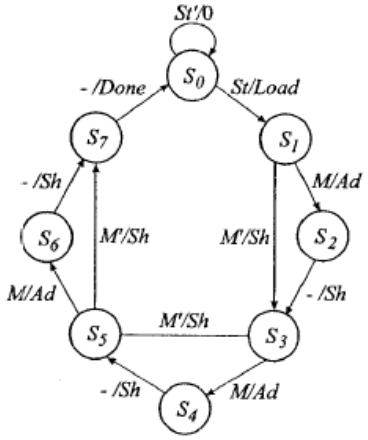
$Z = (Q + X)\,(Q' + X')\,(X + Sh)$

$Z = (Q + X)\,(Q' + X')\,(Q' + Sh)$

18.10 (a)



18.10 (b)



18.10 (c)

0	0	0	0	0	1	0	1		add
	1	0	1	1					
0	1	0	1	1	1	0	1		shift
0	0	1	0	1	1	1	0		shift
0	0	0	1	0	1	1	1		add
	1	0	1	1					
0	1	1	0	1	1	1	1		shift
0	0	1	1	0	1	1	1		

18.10 (d)

Present State	Next State *StM*: 00	01	10	11	*Ad Sh Load Done* 00	01	10	11
S_0	S_0	S_0	S_1	S_1	0000	0000	0010	0010
S_1	S_3	S_2	S_3	S_2	0100	1000	0100	1000
S_2	S_3	S_3	S_3	S_3	0100	0100	0100	0100
S_3	S_5	S_4	S_5	S_4	0100	1000	0100	1000
S_4	S_5	S_5	S_5	S_5	0100	0100	0100	0100
S_5	S_7	S_6	S_7	S_6	0100	1000	0100	1000
S_6	S_7	S_7	S_7	S_7	0100	0100	0100	0100
S_7	S_0	S_0	S_0	S_0	0001	0001	0001	0001

I. (S_0, S_7) (S_1, S_2) (S_3, S_4) (S_5, S_6)
II. (S_0, S_1) (S_2, S_3) (S_4, S_5) (S_6, S_7)
III. (S_1, S_3, S_5) (S_2, S_4, S_6) etc.

B C \ A	0	1
00	S_0	S_1
01	S_3	S_2
11	S_5	S_4
10	S_7	S_6

(Other assignments are possible.)

18.10 (d) For this assignment, from LogicAid:

(contd) $J_A = StB'C' + MC;$ $K_A = M' + B + C;$ $J_B = A'C;$ $K_B = A'C';$ $J_C = AB';$ $K_C = A'B;$ $Ad = MAB'C' + MA'C$
$Sh = M'A + M'C + AB + AC;$ $Load = StA'B'C';$ $Done = A'BC'$