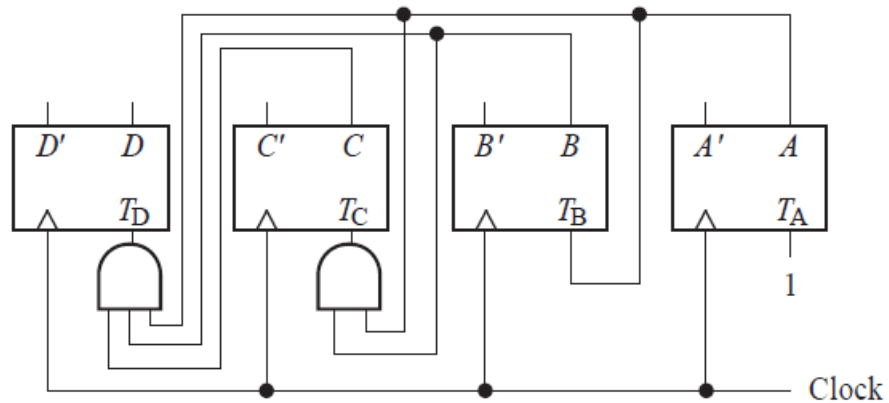
12.4 (a)

Present State $D\,C\,B\,A$	Next State $D^+C^+B^+A^+$	Flip-Flop Inputs $T_DT_CT_BT_A$
0 0 0 0	0 0 0 1	0 0 0 1
0 0 0 1	0 0 1 0	0 0 1 1
0 0 1 0	0 0 1 1	0 0 0 1
0 0 1 1	0 1 0 0	0 1 1 1
0 1 0 0	0 1 0 1	0 0 0 1
0 1 0 1	0 1 1 0	0 0 1 1
0 1 1 0	0 1 1 1	0 0 0 1
0 1 1 1	1 0 0 0	1 1 1 1
1 0 0 0	1 0 0 1	0 0 0 1
1 0 0 1	1 0 1 0	0 0 1 1
1 0 1 0	1 0 1 1	0 0 0 1
1 0 1 1	1 1 0 0	0 1 1 1
1 1 0 0	1 1 0 1	0 0 0 1
1 1 0 1	1 1 1 0	0 0 1 1
1 1 1 0	1 1 1 1	0 0 0 1
1 1 1 1	0 0 0 0	1 1 1 1



As explained in Section 12.3, it can be seen that A changes on every rising clock edge: $T_A = 1$

B changes only when $A = 1$: $T_B = A$

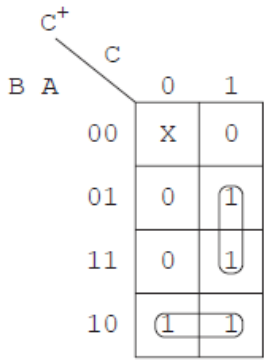
C changes only when both B and $A = 1$: $T_C = AB$

D changes only when A, B, and $C = 1$: $T_D = ABC$

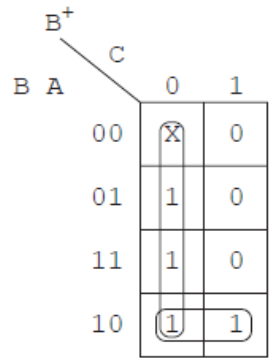
12.4 (b) The binary counter using D flip-flops is obtained by converting each T flip-flop to a D flip-flop by adding an XOR gate.

12.7 (a)

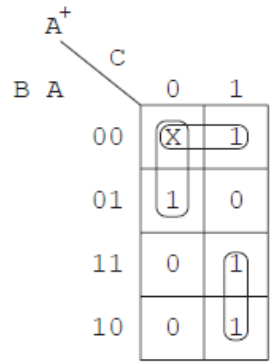
$C\,B\,A$	$C^+B^+A^+$
0 0 0	X X X
0 0 1	0 1 1
0 1 0	1 1 0
0 1 1	0 1 0
1 0 0	0 0 1
1 0 1	1 0 0
1 1 0	1 1 1
1 1 1	1 0 1



$C^+ = C\,A + B\,A'$



$B^+ = C' + B\,A'$

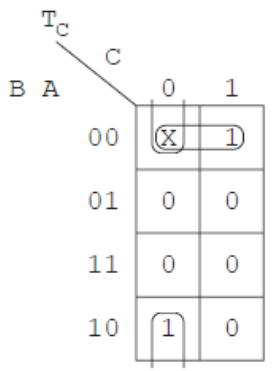


$A^+ = C'B' + C\,B + B'A'$

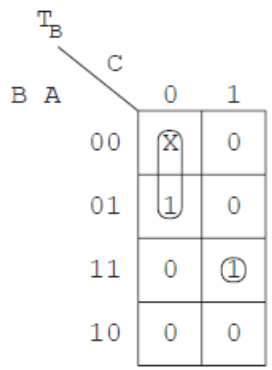
For D flip-flop: 000 goes to 011 because $D_C D_B D_A = 011$

$A^+ = C'B' + C\,B + C\,A'$

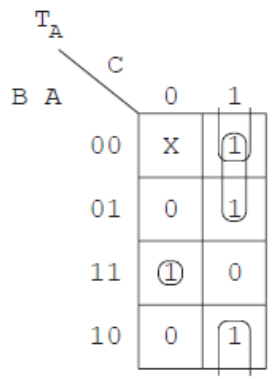
12.7 (b)



$T_C = C'A' + B'A'$



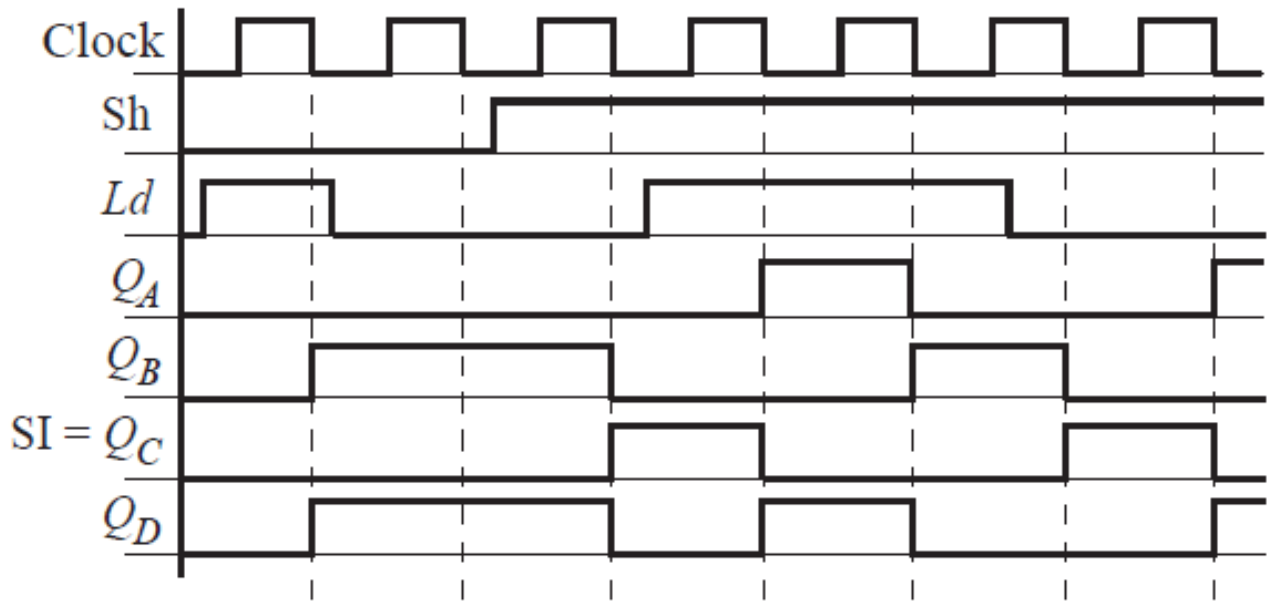
$T_B = C'B' + C\,B\,A$



$T_A = C'B\,A + C\,B' + C\,A'$

For T flip-flop: 000 goes to 110 because $T_A T_B T_C = 110$

12.13 Notice that *Sh* overrides *Ld* when *Sh* = *Ld* = 1



12.20

$A\ B\ C$	$A^+B^+C^+$
0 0 0	X X X
0 0 1	1 0 0
0 1 0	0 1 1
0 1 1	0 0 1
1 0 0	1 0 1
1 0 1	1 1 1
1 1 0	0 1 0
1 1 1	1 1 0

12.20 (a) $D_A = B' + AC;\ D_B = AC + BC';\ D_C = A'B + AB'$

12.20 (b) $J_A = B',\ K_A = BC';\ J_B = AC,\ K_B = A'C;\ J_C = A' + B',\ K_C = A'B' + AB$

12.20 (c) $T_A = A'B' + ABC';\ T_B = A'BC + AB'C;\ T_C = A'B' + A'C' + B'C' + ABC$

12.20 (d) $S_A = B',\ R_A = BC';\ S_B = AC,\ R_B = A'C;\ S_C = A'B + AB',\ R_C = A'B' + AB$

12.20 (e) State 000 goes to 100, because $D_A D_B D_C = 100$.

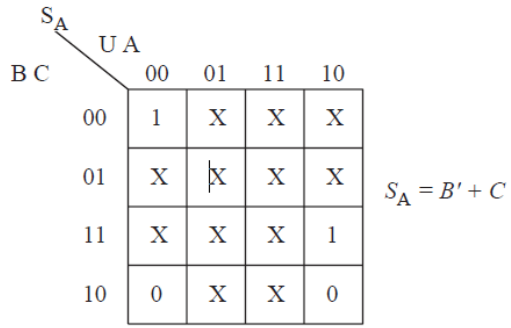
12.26

$Q_3Q_2Q_1Q_0$	*ClrN Ld*
0000	1 0
0001	1 0
0010	1 0
0011	1 0
0100	1 1
0101	x x
0110	x x
0111	x x
1000	x x
1001	x x
1010	x x
1011	1 0
1100	1 0
1101	1 0
1110	1 0
1111	1 0

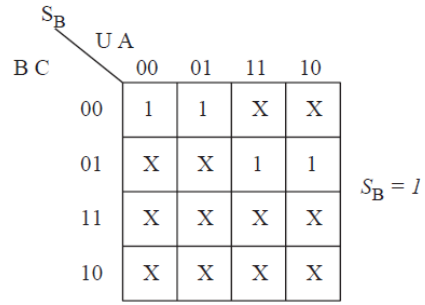
The transition from state 1111 to state 0000 can be effected using Clear, Parallel Load or increment. The latter gives the simplest equations. Then *ClrN* = *1*, $Ld = Q_3'Q_2$, and $P_3P_2P_1P_0 = 1011$.

12.32 (a)

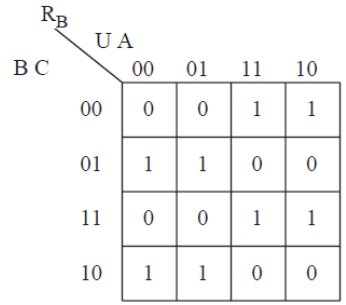
UABC	$S_A R_A$	$S_B R_B$	$S_C R_C$
0000	10	10	10
0001	0x, x1	0x, x1	x1
0010	0x, x1	x1	10
0011	0x, x1	x0	x1
0100	x1	10	10
0101	x0	0x, x1	x1
0110	x0	x1	10
0111	x0	x0	x1
1000	0x, x1	0x, x1	10
1001	0x, x1	10	x1
1010	0x, x1	x0	10
1011	10	x1	x1
1100	x0	0x, x1	10
1101	x0	10	x1
1110	x0	x0	10
1111	x1	x1	x1



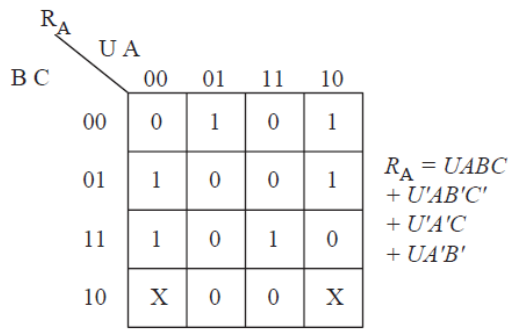
$S_A = B' + C$



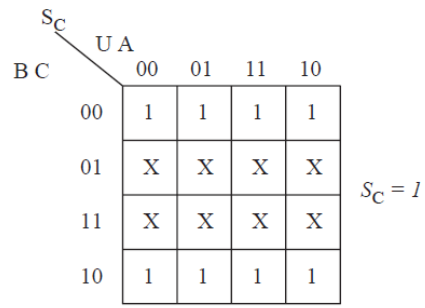
$S_B = 1$



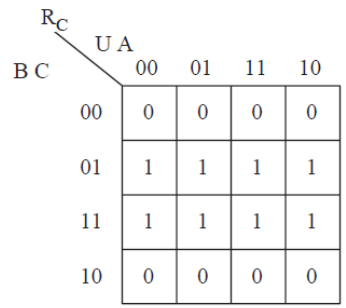
$R_B = U'B'C + U'BC' + UB'C' + UBC$



$R_A = UABC + U'AB'C' + U'A'C + UA'B'$



$S_C = 1$



$R_C = C$

12.32 (b)

$UABC$	CE_AD_A	CE_BD_B	CE_CD_C
0000	11	11	11
0001	0x, 10	0x, 10	10
0010	0x, 10	10	11
0011	0x, 10	0x, 11	10
0100	10	11	11
0101	0x, 11	0x, 10	10
0110	0x, 11	10	11
0111	0x, 11	0x, 11	10
1000	0x, 10	0x, 10	11
1001	0x, 10	11	10
1010	0x, 10	0x, 11	11
1011	11	10	10
1100	0x, 11	0x, 10	11
1101	0x, 11	11	10
1110	0x, 11	0x, 11	11
1111	10	10	10

CE_A

B C \ U A	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	0	0	1	1
10	0	0	0	0

$CE_A = UBC + U'B'C'$

D_A

B C \ U A	00	01	11	10
00	1	0	X	X
01	X	X	X	X
11	X	X	0	1
10	X	X	X	X

$D_A = A'$

CE_B

B C \ U A	00	01	11	10
00	1	1	0	0
01	0	0	1	1
11	0	0	1	1
10	1	1	0	0

$CE_B = U'C' + UC$

D_B

B C \ U A	00	01	11	10
00	1	1	X	X
01	X	X	1	1
11	X	X	0	0
10	0	0	X	X

$D_B = B'$

CE_C

B C \ U A	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$CE_C = 1$

D_C

B C \ U A	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	1	1

$D_C = C'$

12.39

Clock Cycle	Input Data	EnIn	EnAd	LdAc	LdAd	Accumulator Register	Addend Register	Bus	Description
0	18	1	0	1	0	0	0	18	Input to accumulator
1	13	1	0	0	1	18	0	13	Input to addend
2	15	0	1	1	0	18	13	31	Sum to accumulator
3	93	1	0	0	1	31	13	93	Input to addend
4	47	0	1	1	0	31	93	124	Sum to accumulator
5	22	1	0	0	1	124	93	22	Input to addend
6	0	0	1	0	0	124	22	146	Sum on bus

Note: Register values change *after* the clock edge. So a value loaded from the bus appears in the register on the next clock cycle after the load signal and bus value are present.