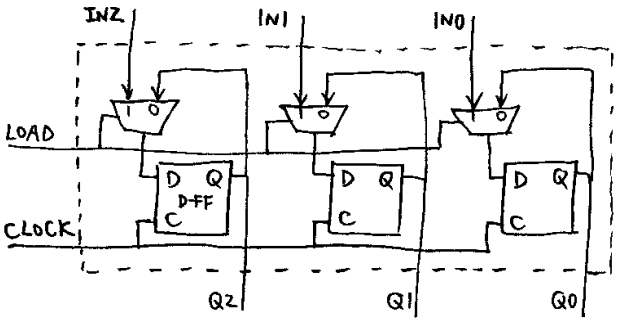
CPRE 281 – Solutions to Practice Questions for Exam #3

1.



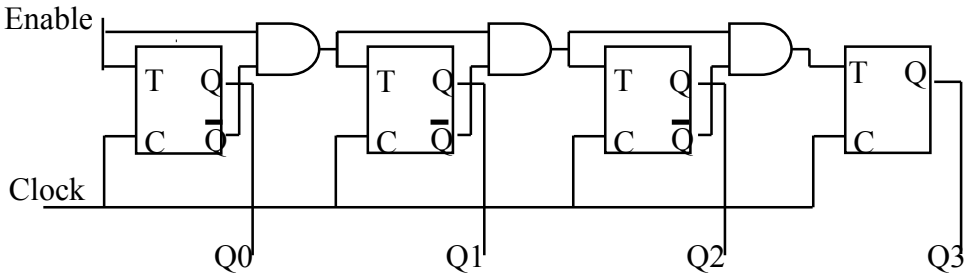
2. Bit Q_i is toggled when Q_{i-1}=0, …, Q_0=0 and ENABLE =1. Therefore,

T_0 = ENABLE

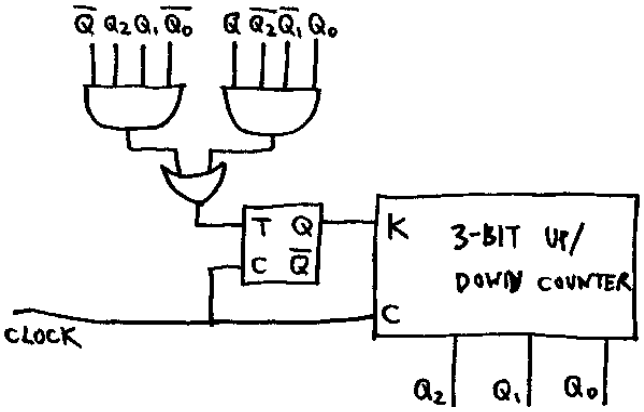
$T_1 = Q_0'$.ENABLE

$T_2 = Q_1'.Q_0'$.ENABLE

$T_3 = Q_2'.Q_1'.Q_0'$.ENABLE

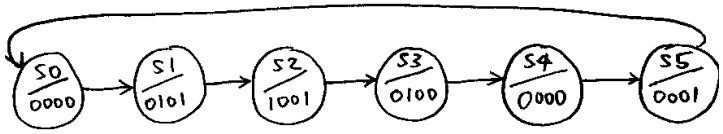


3.



We use the T flip-flop to store the direction of the counting. Notice that we need to change the input of the T flop-flop when Q = 0 and Q2 Q1 Q0 = 110 or Q = 1 and Q2 Q1 Q0 = 001. In other words, T = Q' Q2 Q1 Q0' + Q Q2' Q1' Q0. Then the input to K will be toggled and the counting direction will change when Q2 Q1 Q0 = 111 or 000.

4.



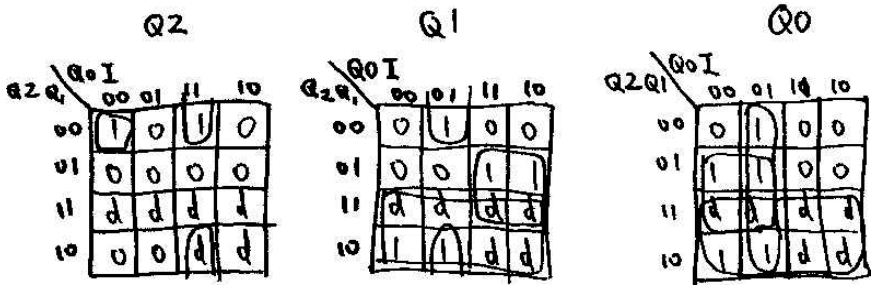
At least 3 bits.

5. (a)

State	State Assignment
S0	000
S1	001
S2	010
S3	011
S4	100

Current State Q2 Q1 Q0	Input X	Next State Q2 Q1 Q0
000	0	100
000	1	011
001	0	000
001	1	100
010	0	001
010	1	001
011	0	010
011	1	010
100	0	011
100	1	011
101	0	ddd
101	1	ddd
110	0	ddd
110	1	ddd
111	0	ddd
111	1	ddd

(b)

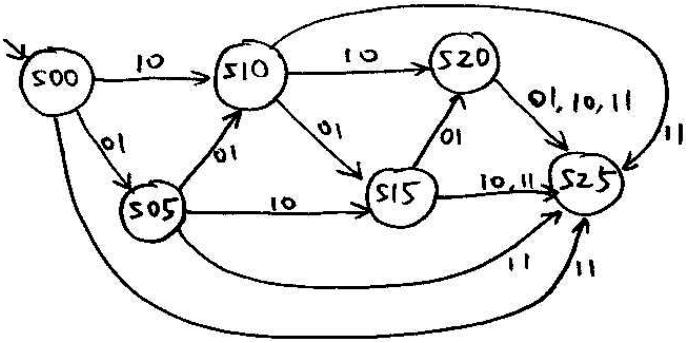


Next Q2 = Q2' Q1' Q0' X' + Q1' Q0 X
Next Q1 = Q2 + Q1 Q0 + Q1' Q0' X
Next Q0 = Q2 + Q1 Q0' + Q0' X

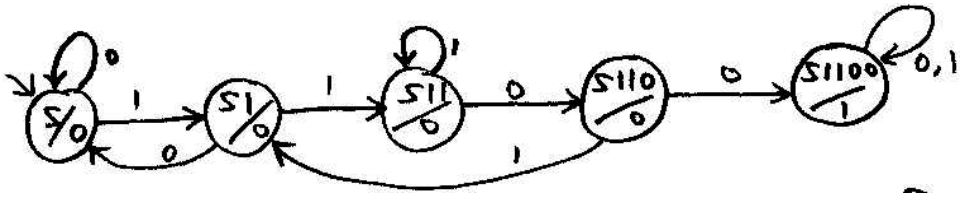
6. S1 (The sequence is S0 S3 S2 S1 S0 S4 S3 S2 S1.)

7. The machine is a Moore machine with the following outputs:

State	Outputs
S00	0000 0000
S05	0000 0101
S10	0001 0000
S15	0001 0101
S20	0010 0000
S25	0010 0101



8.



State table:

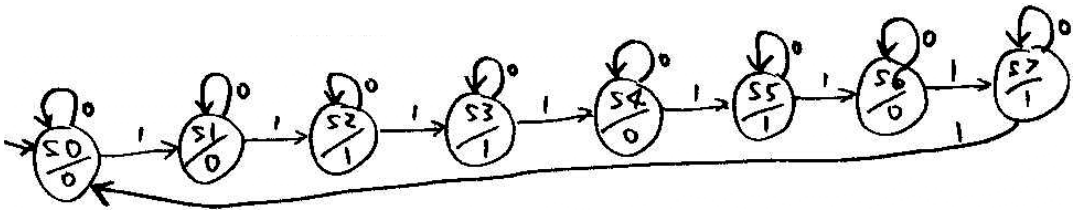
Present state	Next state		Output z
	w=0	w=1	
S	S	S1	0
S1	S	S11	0
S11	S110	S11	0
S110	S1100	S1	0
S1100	S1100	S1100	1

State-assigned table:

Present state y2 y1 y0	Next state w=0 Y2 Y1 Y0	 w=1 Y2 Y1 Y0	Output z
000	000	001	0
001	000	010	0
010	011	010	0
011	100	001	0
100	100	100	1
101	ddd	ddd	d
110	ddd	ddd	d
111	ddd	ddd	d

Expressions for Y2, Y1, Y0 and z can be derived. Then the circuit diagram can be drawn. They are skipped here.

9.



10. (a) Step 1: B ← e^x // A = x, B = e^x
Step 2: A ← A * B // A = x * e^x, B = e^x
Step 3: B ← A + B // A = x * e^x, B = x * e^x + e^x

(b)

Step	ALE	BLE	e0	e1	e2	e3	e4
0	0	1	0	0	1	0	0
1	1	0	0	0	0	0	1
2	0	1	0	0	0	1	0

11.

Step	LD1	LD2	M1	M2	OpCode	enable0	enable1
(1)	1	0	d	d	dd	0	1
(2)	0	1	d	d	dd	0	1
(3)	1	0	0	1	00	1	0
(4)	0	1	0	1	01	1	0
(5)	1	0	0	1	01	1	0

Step	A after step	B after step
(1)	3	????
(2)	3	5
(3)	8	5
(4)	8	3
(5)	5	3

12. (a)

Operation	LD1	LD2	M1	M2	OpCode	enable0	enable1
i)	1	0	d	d	d	0	1
ii)	0	1	d	d	d	0	1
iii)	1	0	0	1	0	1	0
iv)	0	1	0	1	0	1	0
v)	1	0	0	1	0	1	0

(b)

Operation	Register A	Register B
i)	0101	????
ii)	0101	1100
iii)	1001	1100
iv)	1001	0101
v)	1100	0101

(c) This way is better for the following reasons:

1) There is no overflow problem if XOR is used.
2) XOR is faster and less expensive than ADD and SUB.

13. The state table is given below.

Curr. State	CC	W	INDATA	LDA	LDB	OP	FOUND	Next State
S0	xx	0	1110	1	0	d	0	S1
S1	xx	0	0100	0	1	d	0	S2
S2	01	1	dddd	1	0	0	0	S2
S2	00	1	dddd	1	0	0	0	S2
S2	10	d	dddd	0	0	d	1	S2

14. Note that 1MHz = 10^6Hz.

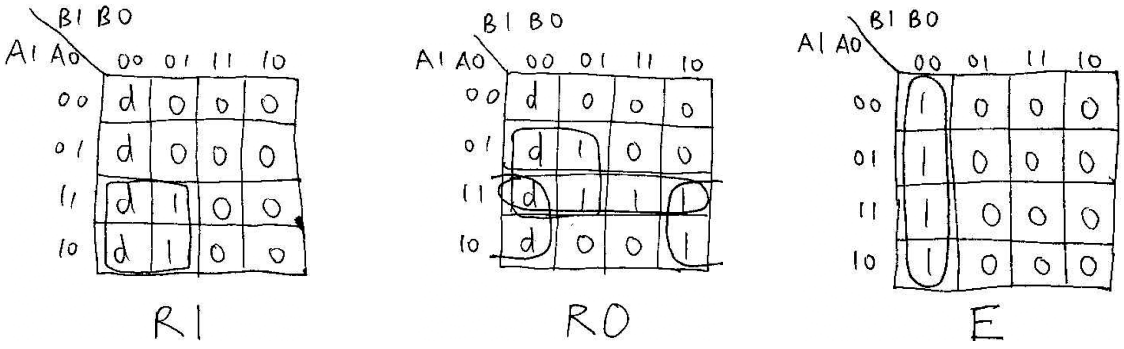
(a) Clock period = 1/(1000*10^6Hz) = 1 nsec.
Clock period >= FF set-up time + Next state compute time + FF propagation delay
So Next state compute time <= 1 – 0.15 – 0.25 = 0.6 nsec

(b) Clock period = 1/(1333*10^6Hz) = 0.75 nsec.
Clock period >= FF set-up time + Next state compute time + FF propagation delay
So Next state compute time <= 0.75 – 0.15 – 0.25 = 0.35 nsec

15. (a)

A1	A0	B1	B0	R1	R0	E
0	0	0	0	d	d	1
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	d	d	1
0	1	0	1	0	1	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	d	d	1
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	0
1	1	0	0	d	d	1
1	1	0	1	1	1	0
1	1	1	0	0	1	0
1	1	1	1	0	1	0

(b)



R1=A1 B1'
R0=A1 A0 + A1 B0' + A0 B1'
E=B1' B0'

(c)