EECE 238 Exam II

Name: Solutions

Problem 1	30 /30
Problem 2	20 /20
Problem 3	10 /10
Problem 4	15 /15
Problem 5	25 /25
Total:	100/100

Good Luck!

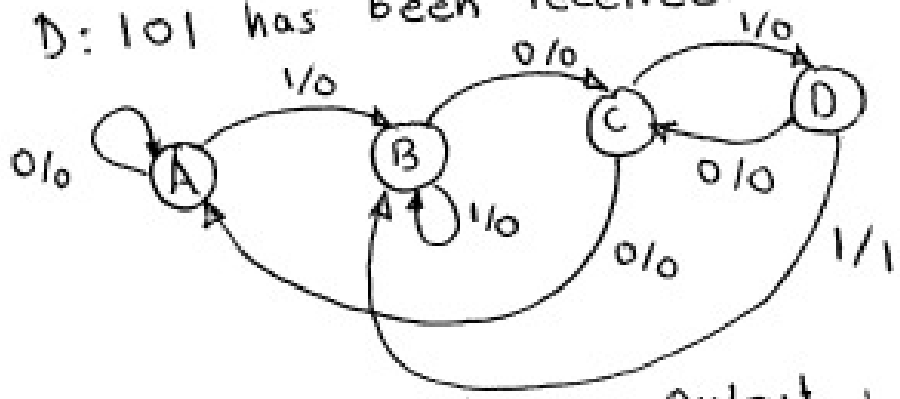
Problem 1 (30 points total) *Sequence Recognizer Design.*

Design a digital circuit to recognize the occurrence of the input sequence 1011. The circuit will output a 1 when the previous inputs were 101 and the current input is 1. Note that since the output depends on the input (as well as the current state), you need a ***Mealy*** solution to this problem.

1 (a) (12 points) Derive the state transition diagram.
1 (b) (10 points) Derive the state table and Flip-Flop inputs for J-K Flip-Flops.
1 (c) (5 points) Use Karnaugh maps to minimize the equations for the Flip-Flop inputs, and the output.
1 (d) (3 points) Draw the final circuit.

1(a)

A: No 1 received yet.
B: 1 has been received.
C: 10 has been received.
D: 101 has been received



1(b)

Current State	Next State I=0	Next State I=1	Output I=0	Output I=1	I=0: J_1 K_1	I=0: J_0 K_0	I=1: J_1 K_1	I=1: J_0 K_0
A=00	A=00	B=01	0	0	0X	0X	0X	1X
B=01	C=11	B=01	0	0	1X	X0	0X	X0
C=11	A=00	D=10	0	0	X1	X1	X0	X1
D=10	C=11	B=01	0	1	X0	1X	X1	1X

Recall JK FF excitation table:

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

apply here.

1(c)

"Copy" tables for k-maps:

Present State Q_1Q_0	I = 0	I = 1
00	0	0
01	0	0
11	X	X
10	X	X

$J_1 = I'Q_0$

Present State Q_1Q_0	I = 0	I = 1
00	X	X
01	X	X
11	0	0
10	0	0

$K_1 = I'Q_0 + IQ_0'$

Present State Q_1Q_0	I = 0	I = 1
00	0	1
01	X	X
11	X	X
10	1	1

$J_0 = I + Q_1$

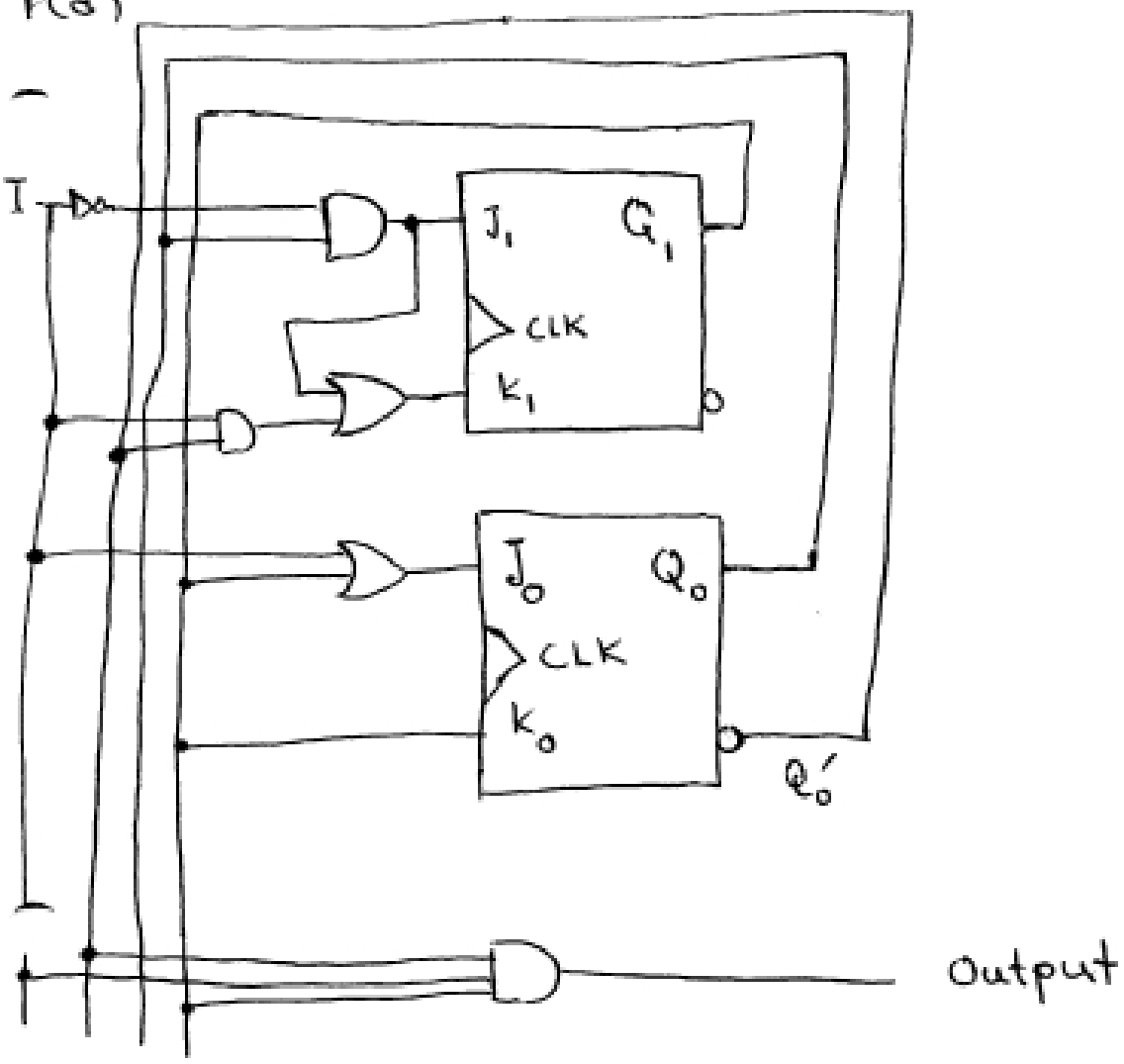
Present State Q_1Q_0	I = 0	I = 1
00	X	X
01	0	0
11	1	1
10	X	X

$K_0 = Q_1$

Present State Q_1Q_0	I = 0	I = 1
00	0	0
01	0	0
11	0	0
10	0	1

Output $= IQ_1Q_0'$

1(d)



I Q_0' Q_0 Q_1

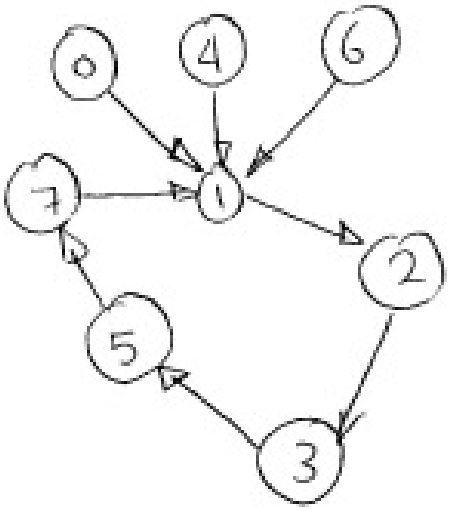
Problem 2 (20 points total) *Synchronous Counter Design.*

Design a binary counter that counts through the prime numbers less than 10. This means that your counter is to count: $1, 2, 3, 5, 7$ and then go back to 1. For your design, assume that there is no reset signal and you are thus forced to send any of the states $0, 4, 6$ back to 1.

2 (a) (3 points) Draw the state transition diagram.
2 (b) (7 points) Derive the state table for implementing the counter using D Flip-Flops.
2 (c) (7 points) Use K-maps to minimize the inputs to the D Flip-Flops.
2 (d) (3 points) Indicate the final circuit.

Problem 2 (regular assignment)

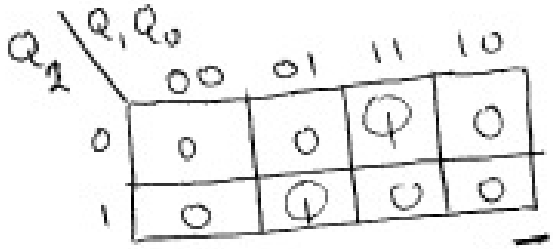
2(a)



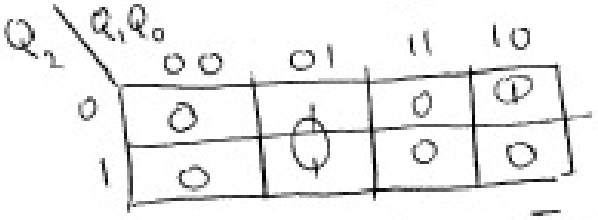
2(b)

Present state $Q_2Q_1Q_0$	D-FF inputs Next state $Q_2Q_1Q_0$
0 0 0	0 0 1
0 0 1	0 1 0
0 1 0	0 1 1
1 0 0	0 0 1
1 0 1	1 1 1
1 1 0	0 0 1
1 1 1	0 0 1

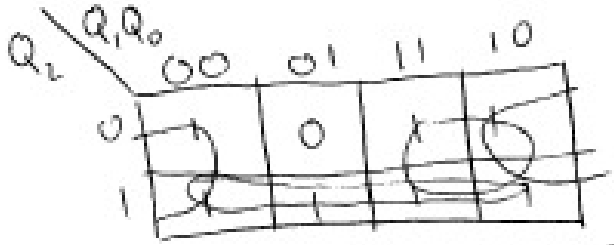
2(c)



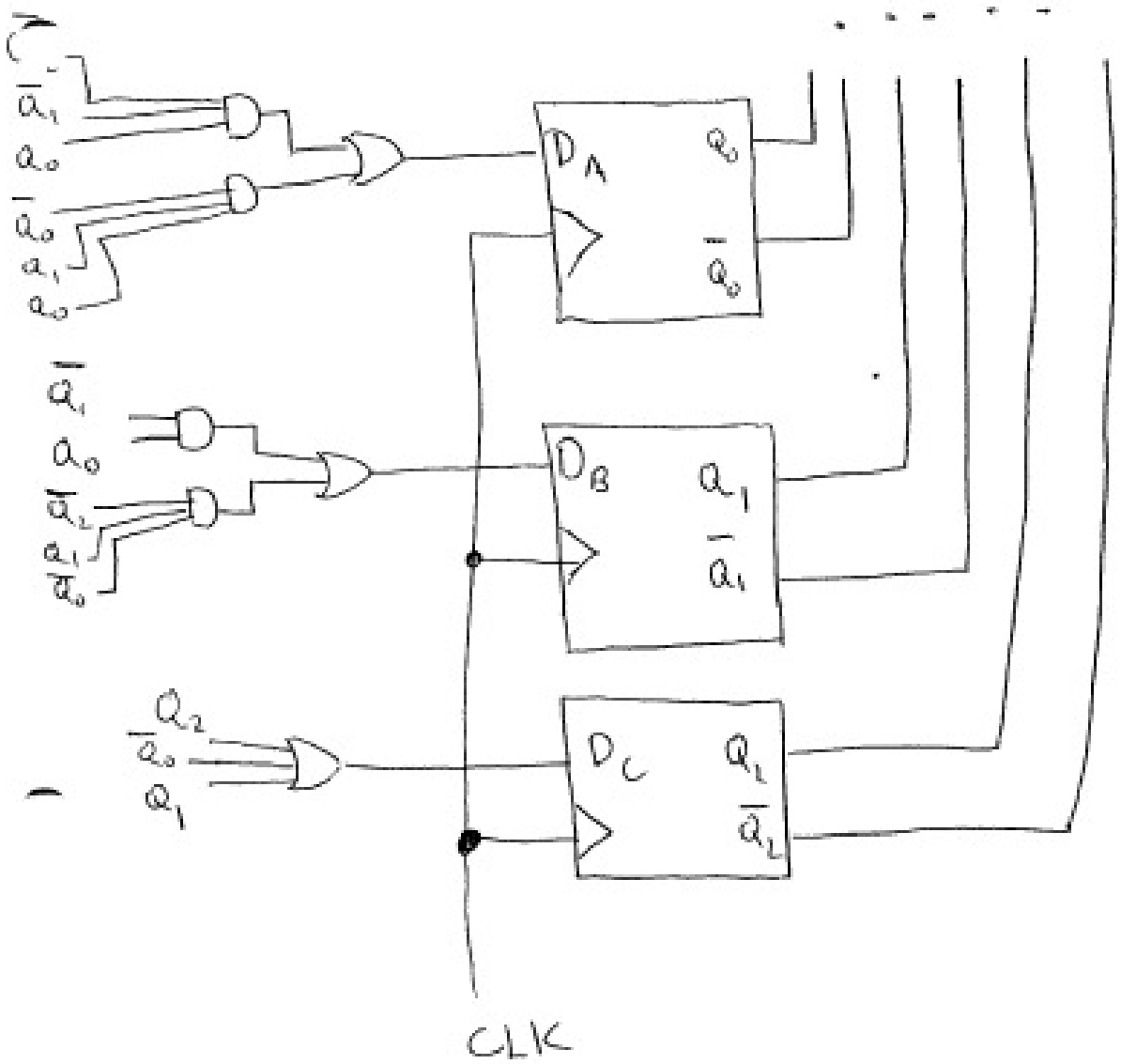
$$D_2 = Q_2\bar{Q}_1Q_0 + \bar{Q}_0Q_1Q_0$$



$$D_1 = \bar{Q}_1Q_0 + \bar{Q}_2Q_1\bar{Q}_0$$



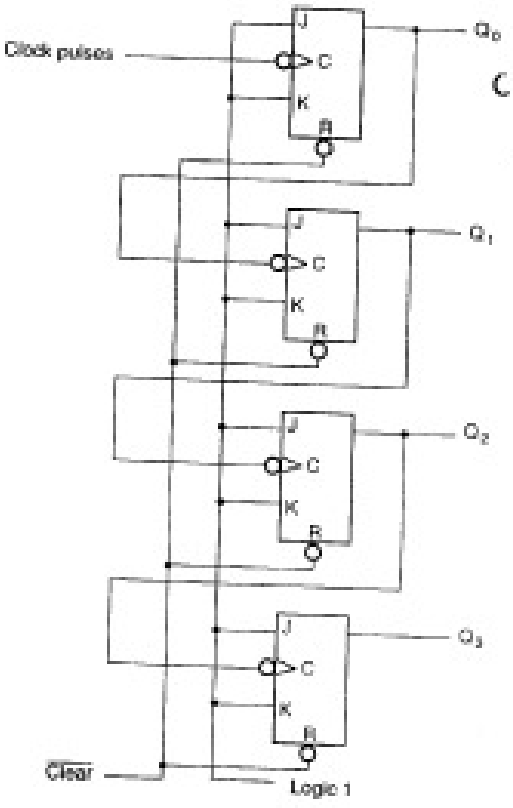
$$D_0 = Q_2 + \bar{Q}_0 + Q_1$$

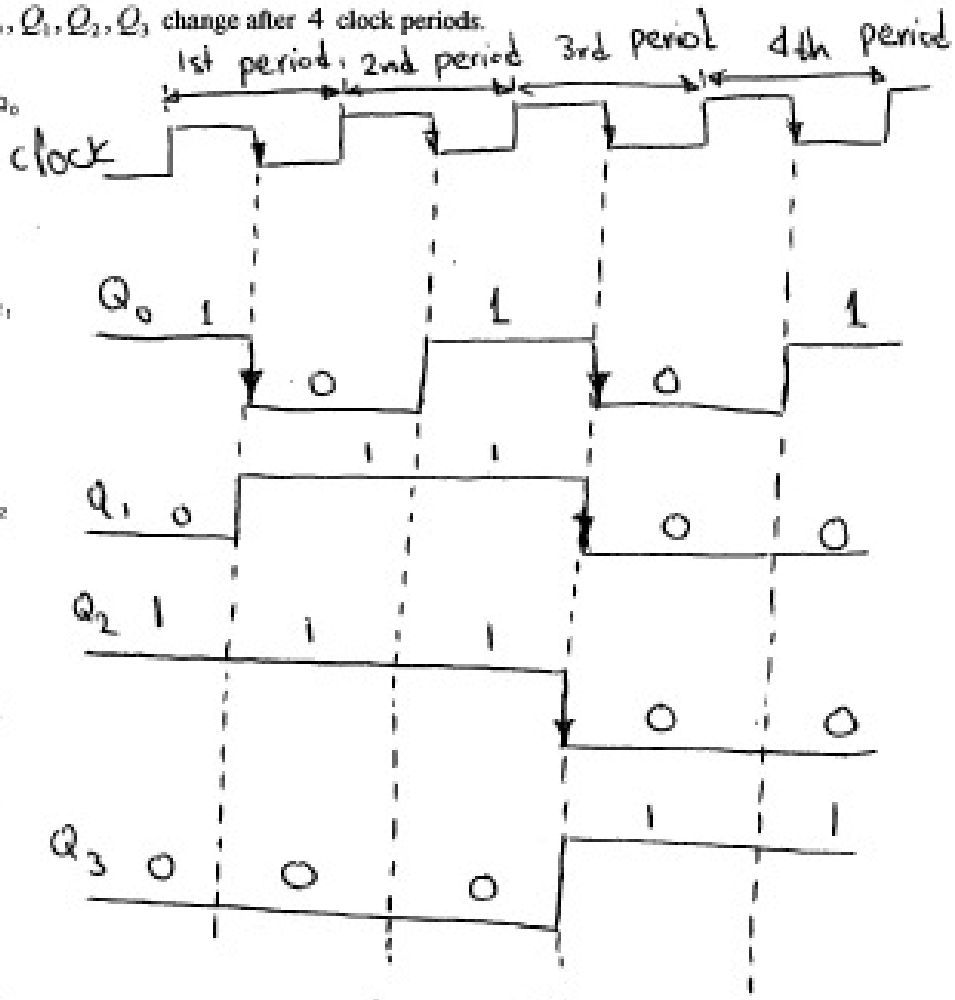

$\bar{Q}_1$
Q_0
$\bar{Q}_0$
Q_1
Q_0
D_A
Q_0
$\bar{Q}_0$
$\bar{Q}_1$
Q_0
$\bar{Q}_2$
Q_1
Q_0
D_B
Q_1
$\bar{Q}_1$
Q_2
$\bar{Q}_0$
Q_1
D_C
Q_2
$\bar{Q}_2$
CLK

Problem 3 (10 points total) *Ripple Counting*

Consider the 4-bit Ripple Counter shown below. Assume that: $Q_0 = 1, Q_1 = 0, Q_2 = 1, Q_3 = 0$.

Clearly indicate how the states of Q_0, Q_1, Q_2, Q_3 change after 4 clock periods.





Notes: ① changes occur only with falling edges.

② All Q_s oscillate: ...0→1→0→1→0...

③ (a) Q_0 changes with the ↓clock.
(b) Q_1 changes with ↓Q_0
(c) Q_2 changes with ↓Q_1
(d) Q_3 changes with ↓Q_2

} ripple counting

Problem 4 (15 points total) *Even Parity Detector*

Using T Flip-Flops, design an even parity detector that outputs a 1 after an even number of 1s have been received. Note that since the output does not depend on the current input, you will need a ***Moore*** solution to this problem.

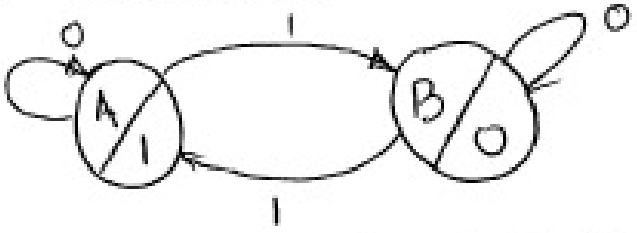
4 (a) **(3 points)** Derive the state transition diagram.
4 (b) **(3 points)** Derive the state table and T Flip-Flop input(s).
4 (c) **(3 points)** Indicate the final Circuit.
4 (d) **(3 points)** Re-implement the Circuit using D Flip-Flop(s).
4 (e) **(3 points)** Re-implement the Circuit using JK Flip-Flop(s).

4(a) The hint given during the exam is that zero 1s is considered to be an even number of 1s.

Let the states be:

A: No or even 1s received

B: Odd 1s received.



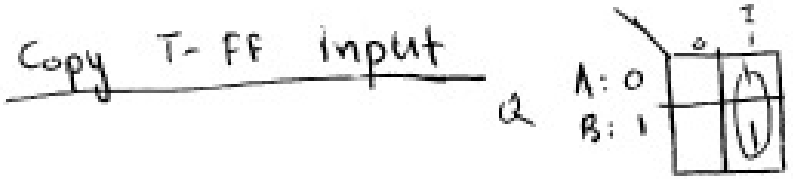
4(b) A: 0, B: 1 } state assignment. Initially, apply Flip-flop reset to start in A.

Present State	Next State I=0	Next State I=1	T-FF Input I=0	T-FF Input I=1	Output
A: 0	A: 0	B: 1	0	1	1
B: 1	B: 1	A: 0	0	1	0

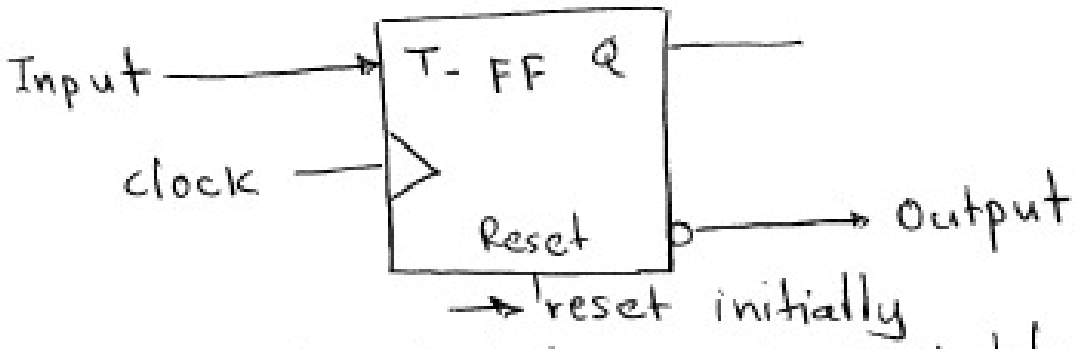
0s for no state change

1s for changing state.

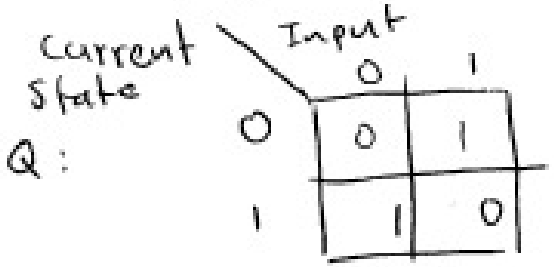
Copy T-FF input



4(c)

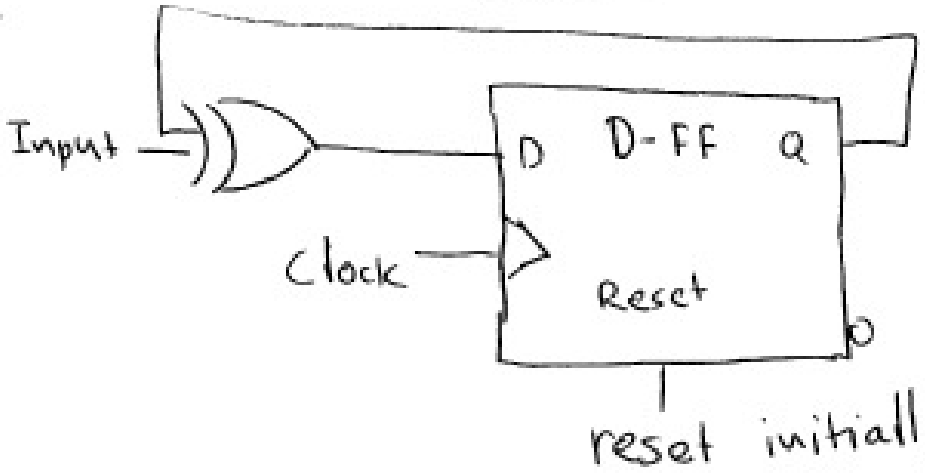


4(d) Copy Next-State part of table into a K-map:

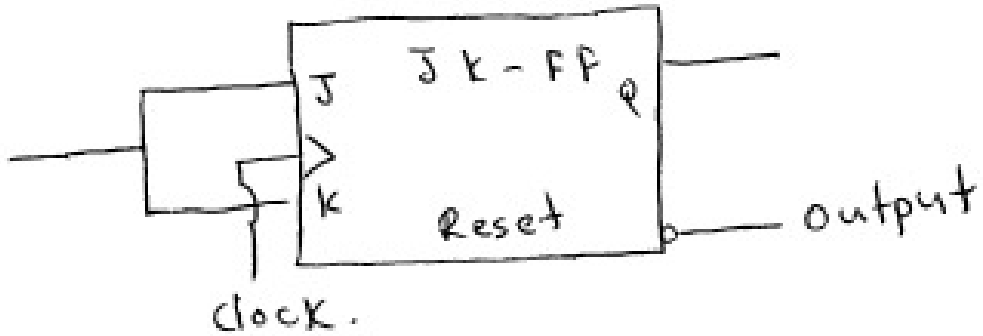


$D = Q'I + QI'$

$\Rightarrow D = Q \oplus I$



4(e) Recall that J-K FF implements a T-FF using $J = K = T$. From 4(c):



Problem 5 (25 points total) *Serial Addition*

Consider the Serial-Adder Circuit indicated below.

We want to use the adder to compute $1+2-4$.

Note that the following parts are independent (except for the last part). Hence, if you do not know how to do a particular part, simply move to the next part.

For full credit, you must clearly indicate:
(i) the values for all the inputs, and
(ii) how the relevant registers *A*, *B* and the Carry are affected with respect to the clock.
If the state of the registers or the Carry cannot be determined, simply mark them by X.

5(a) (3 points) Initially, we do not know what value is stored in register A. Indicate how to initialize register A to zero.

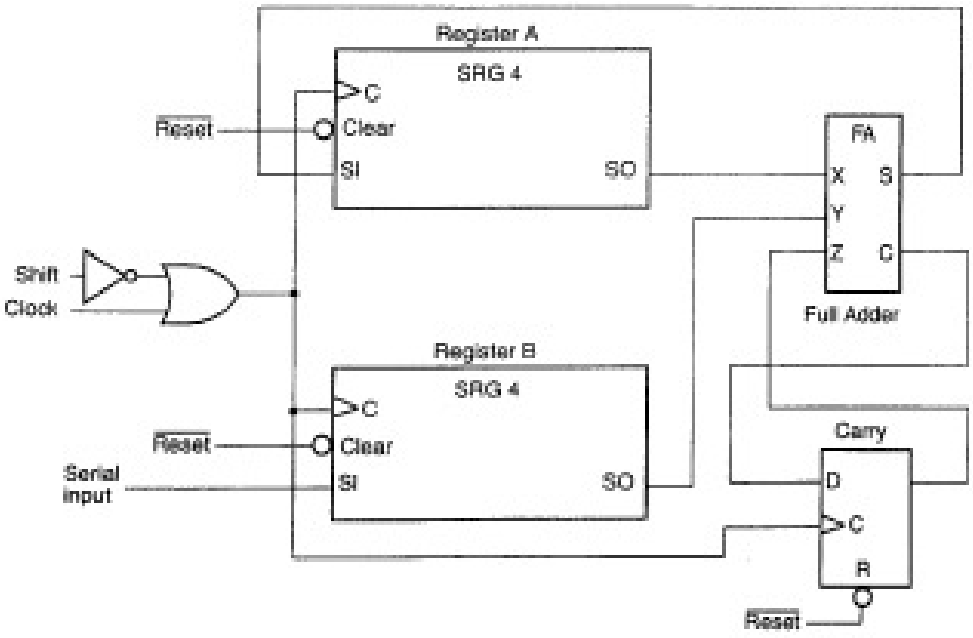
5(b) (7 points) Indicate how to (i) compute $A = 0+1$, assuming that $A = 0$, while at the same time, leave register B with the value: $B = 2$. How many clock cycles did it take to perform the addition?

5(c) (6 points) Indicate how to compute (i) $A = 1+2$, while at the same time, leave register B with $B = -4$. Assume that register B contains $B = 2$ and register A contains $A = 1$. How many clock cycles did it take to perform the addition?

5(d) (3 points) Indicate how to use the circuit to finish the computation and store the final result of $1+2-4$ in A.

5(e) (3 points) In general, suppose that we want to add N four-bit numbers (assuming that all the results will actually fit in our 4-bit register A without causing overflows). How many clock cycles would we need?

5(f) (3 points) Suppose that only unsigned numbers are to be added. Can you think of a simple way to detect overflows?
Hint: what happens to the Carry bit?



5(a) To make sure that Register A is set to 0, and stays that way:

1. Apply Shift = 0 to avoid accumulating a sum in register A.
2. Apply logic 0 to $\overline{\text{Reset}}$ of register A.

5(b)

1. Apply shift = 0 to avoid shifts, additions into register A.
2. Apply $\overline{\text{Reset}} = 0$ for register B.
3. Apply $\overline{\text{Reset}} = 0$ for carry Flip-Flop.
4. Apply shift = 1 and apply (before rising edge of the clock), the 4 bits in 1_2: 1, 0, 0, 0 to SI.
5. Apply shift = 1 and apply 0, 1, 0, 0 (2 in reverse) to SI.
6. Apply shift = 0 to stop the process.

	shift	SI	B	A	Carry
initially:	1	1	0000	0000	0
After ↑	1	0 } 1	1000	0000	0
After ↑	1	0	0100	0000	0
After ↑	1	0	0010	0000	0
After ↑	1	0	0001	0000	0
After ↑	1	1 } 2	0000	1000	0
After ↑	1	0	1000	0100	0
After ↑	1	0	0100	0010	0
After ↑	1	0	0010	0001	0
After ↑	0	1X	0010	0001	0

Zero in shift stops everything

this input is ignored

B=2

A=1 to =1

Total = 8 clock cycles.

5(c) Eventhough it is not necessary here, it is always a good idea to clear the carry before any new addition: $\overline{Reset} = 0$ to carry FF.

For -4, recall that the Full Adder will work with 2's compliment:

$$4 = (0100)_2$$

$$\begin{array}{r} 1011 \\ +\quad 1 \\ \hline 1100 \end{array}$$ 1's comp. / 2's comp.

All we have to do is shift-in -4:

	shift	SI	B	A	carry
Initially	1	0	0010	0001	0
After ↑	1	0	0001	1000	0
After ↑	1	1	0000	1100	0
After ↑	1		1000	0110	0
After ↑	1	1	1100	0011	0
After ↑	0	X			

stop everything

input is ignored

B = -4

A = 1+2 = 3

Total = 4 clock cycles.

Note shift values change after ↑

5(d) To finish the computation, shift four times (after resetting the carry):

	Shift	SI	B	A	Carry
Initially	1	0	1100	0011	0
After ↑	1	0	0110	1001	0
After ↑	1	0	0011	1100	0
After ↑	1	0	0001	1110	0
After ↑	0	0	0000	1111	0
	0				

A = −1

I left B = 0 (not reqd)

After holding shift = 1 for 4 ↑s, stop.

5(e) It takes: 8 cycles for ~~adding~~ setting A to the initial, first number. After that, a new number gets added every 4 cycles ⇒ (N+1)4 for N numbers

5(f) In ~~unsigned~~ arithmetic, Carry = 1 at the end of an addition signifies an overflow.