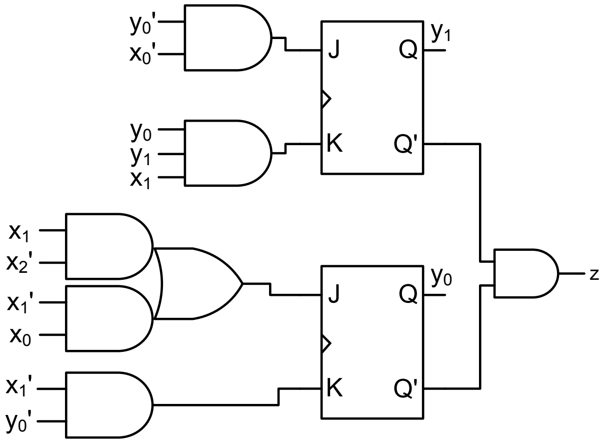
Quiz Problems (50 points total)

Problem 1 (12 points)

Answer the questions on the following gate network.



1. **(6 points)** Given the following timing parameters, calculate the setup time, hold time and propagation delay of the sequential network. All inverted inputs are available and there are no additional NOT gates.

   JK flip-flop characteristics: $t_{su}(\text{cell}) = 0.8$ ns, $t_h(\text{cell}) = 2.7$ ns, $t_p(\text{cell}) = 1.8$ ns

   Gate characteristics: $t_p(\text{AND2}) = t_p(\text{OR2}) = 3.5$ ns, $t_p(\text{AND3}) = 4.0$ ns

   Solution For $d1^x$, the worst case path is through the J input for y_0, where the signal needs to go through two gate levels. The logic for the output is just an AND gate.

$$\begin{aligned} d1^x &= t_p(\text{AND2}) + t_p(\text{OR2}) = 3.5 + 3.5 = 7.0(\text{ns}) \\ d2 &= t_p(\text{AND2}) = 3.5(\text{ns}) \end{aligned}$$

   Now using this we can calculate the following.

   Network setup time:

$$\begin{aligned} t_{su}(\text{net}) &= d1^x + t_{su}(\text{cell}) \\ &= 7.0 + 0.8 \\ &= 7.8 \text{ (ns)} \end{aligned}$$

   Network hold time $t_h(\text{net}) = t_h(\text{cell}) = 2.7$ (ns)

Network propagation delay

$$\begin{aligned} t_p(\text{net}) &= t_p(\text{cell}) + d2 \\ &= 1.8 + 3.5 = 5.3 \text{ (ns)} \end{aligned}$$

2. **(6 points)** $t_{in} = 2.5$ ns and $t_{out} = 1.5$ ns. Calculate T_x, T_y and T_z values for this system.

   Solution For $d1^y$, the worst case path is through the 3-input AND gate at the K input for y_1. Therefore,

$$d1^y = t_p(\text{AND3}) = 4.0(\text{ns})$$

With this and numbers from the previous part, we can write:

$$\begin{aligned} T_x &= t_{in} + d1^x + t_{su}(\text{cell}) = 2.5 + 7.0 + 0.8 = 10.3 \text{ (ns)} \\ T_y &= t_p(\text{cell}) + d1^y + t_{su}(\text{cell}) = 1.8 + 4.0 + 0.8 = 6.6 \text{ (ns)} \\ T_z &= t_p(\text{cell}) + d2 + t_{out} = 1.8 + 3.5 + 1.5 = 6.8 \text{ (ns)} \end{aligned}$$

Problem 2 (8 points)

We wish to create an SR flip-flop using a T flip-flop. The transition table for the SR flip-flop is:

$PS = Q(t)$	$S(t)R(t)$			
	00	01	10	11
0	0	0	1	-
1	1	0	1	-
	$NS = Q(t+1)$			

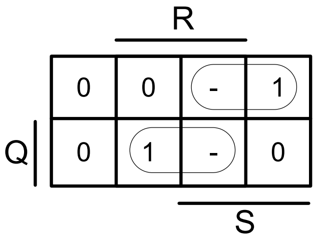
1. **(3 points)** Fill in the table below.

   Solution From the given transition table, we can write:

$PS = Q(t)$	$S(t)R(t)$				$S(t)R(t)$			
	00	01	10	11	00	01	10	11
0	0	0	1	-	0	0	1	-
1	1	0	1	-	0	1	0	-
	$NS = Q(t+1)$				T			

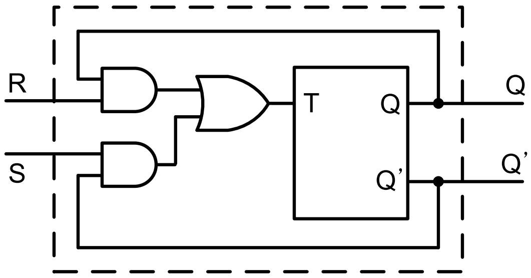
2. **(5 points)** Using the table, obtain the expression for T and draw the final circuit.

   Solution From the completed table, we can get the following K-map:



0	0	-	1
0	1	-	0

and from this we get $T = QR + Q'S$.
The final circuit looks like:



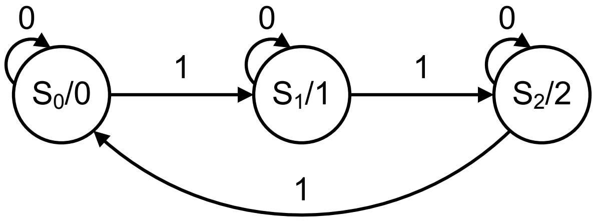
Problem 3 (30 points)

Design a modulo-3 counter using JK flip-flops. Use a Moore machine. The counter specifications are as shown here:

Input:	$x(t) \in \{0, 1\}$
Output:	$z(t) \in \{0, 1, 2\}$
State:	$s(t) \in \{S_0, S_1, S_2\}$
Initial state:	$s(0) = S_0$
Function:	In modulo-3, the system counts the number of 1's in the input sequence $x(0, t-1)$

1. **(6 points)** Draw the state transition diagram for the counter. Clearly show **ALL** transitions from each state. Show the output of each state.

   Solution The state transition diagram is shown.



2. **(8 points)** Using the following **unconventional** encoding for the states, complete the table below.

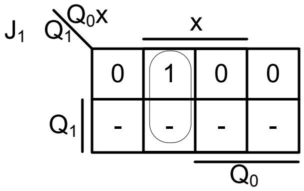
	Q_1	Q_0
S_0	0	0
S_1	1	1
S_2	0	1

Solution

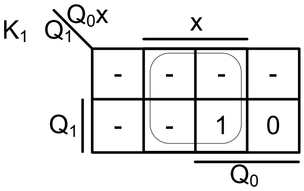
Q_1Q_0	$x=0$	$x=1$	$x=0$	$x=1$
00	00	11	0– 0–	1– 1–
01	01	00	0– –0	0– –1
10	– –	– –	– – – –	– – – –
11	11	01	–0 –0	–1 –0
	NS		$J_1K_1J_0K_0$	

3. **(12 points)** Complete the K-maps and obtain the minimal expressions for J_1, K_1, J_0 and K_0.

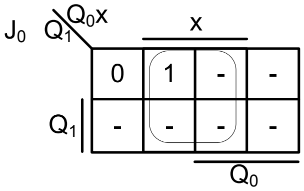
   Solution



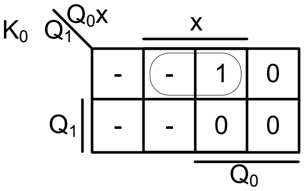
0	1	0	0
-	-	-	-



-	-	-	-
-	-	1	0



0	1	-	-
-	-	-	-



-	-	1	0
-	-	0	0

From these we get:

$$\begin{aligned} J_1 &= Q_0{}'x \\ K_1 &= x \\ J_0 &= x \\ K_0 &= Q_1{}'x \end{aligned}$$

4. **(4 points)** The output is encoded using conventional binary numbers as shown.

	z_1	z_0
0	0	0
1	0	1
2	1	0

Obtain the expressions for the output bits z_1 and z_0. Since this is a Moore machine, show them in terms of the state bits Q_1 and Q_0. If possible, try to obtain the simplest expression by utilizing don't-cares.

Solution From the state encoding and output bit encoding, we can get:

Q_1	Q_0	State	Output value	z_1	z_0
0	0	S_0	0	0	0
0	1	S_2	2	1	0
1	0	–	–	–	–
1	1	S_1	1	0	1

From here, we can write $z_1 = Q_1{}'Q_0$ and $z_0 = Q_1$.