Montogue

Quiz EL304

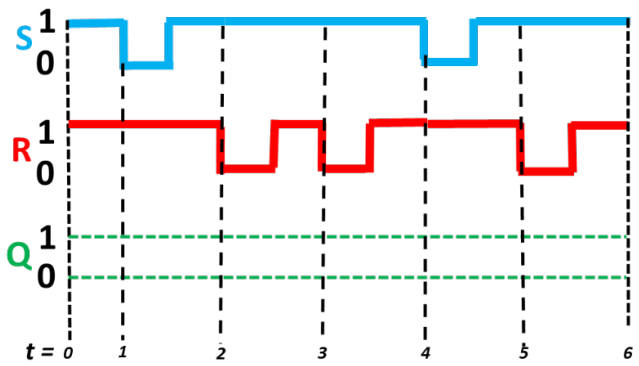
Latches and Flip-Flops

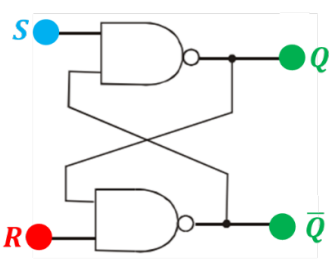
Lucas Monteiro Nogueira

PROBLEMS

Problem 1

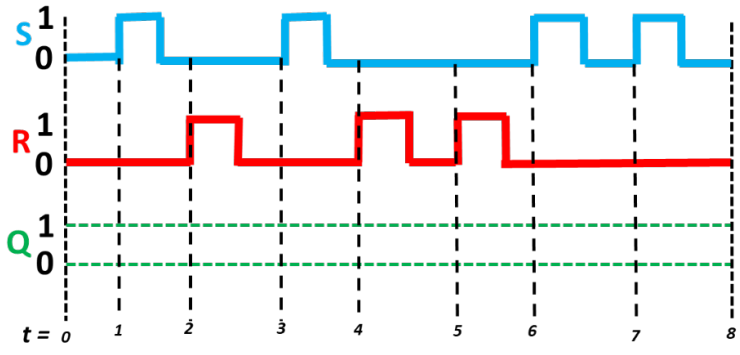
The waveforms illustrated below are applied to the inputs of a SR latch made up of NAND gates. Assuming that output $Q = 0$ initially, sketch the Q waveform.

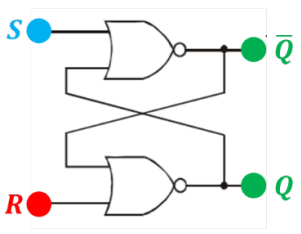




Problem 2

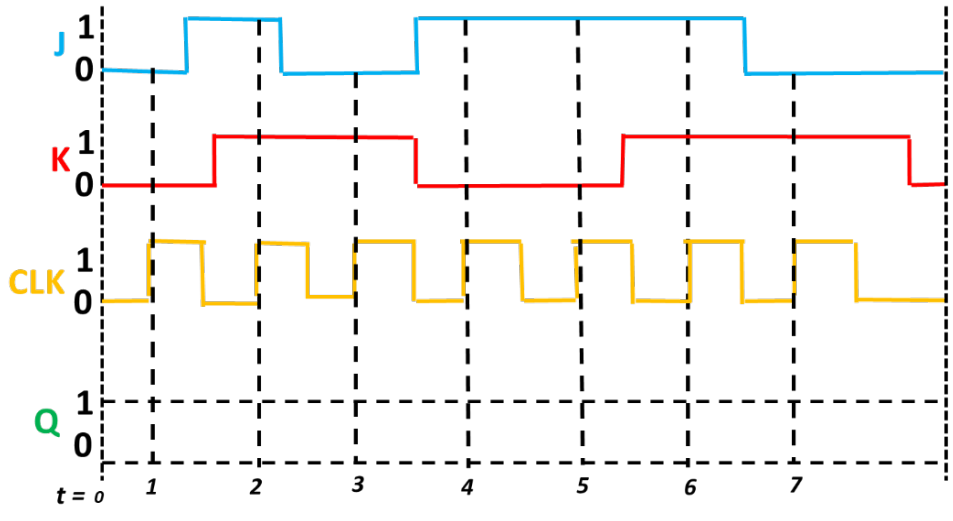
The waveforms illustrated below are applied to the inputs of a SR latch made up of NOR gates. Assuming that output $Q = 0$ initially, sketch the Q waveform.

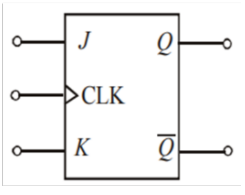




Problem 3

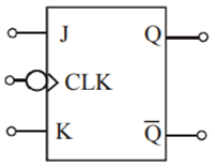
The waveforms illustrated below describe the J and K inputs of a rising-edge triggered JK flip-flop. There are no asynchronous inputs in this particular device. Assuming that output $Q = 0$ initially, sketch the Q waveform.

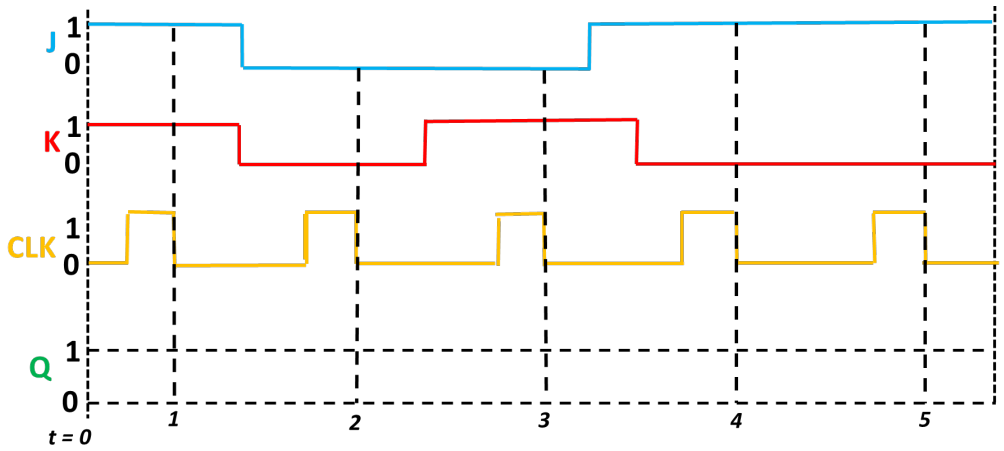




Problem 4

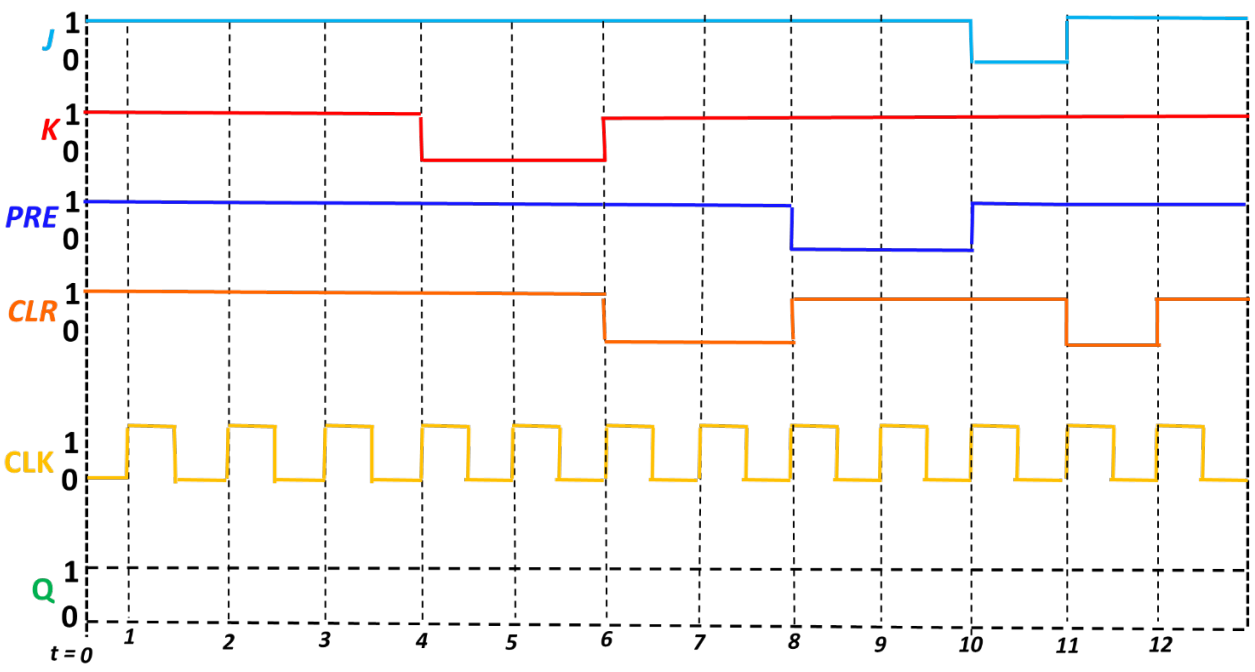
The waveforms illustrated below describe the *J* and *K* inputs of a falling-edge triggered JK flip-flop. There are no asynchronous inputs in this particular device. Assuming that output *Q* = 0 initially, sketch the *Q* waveform.





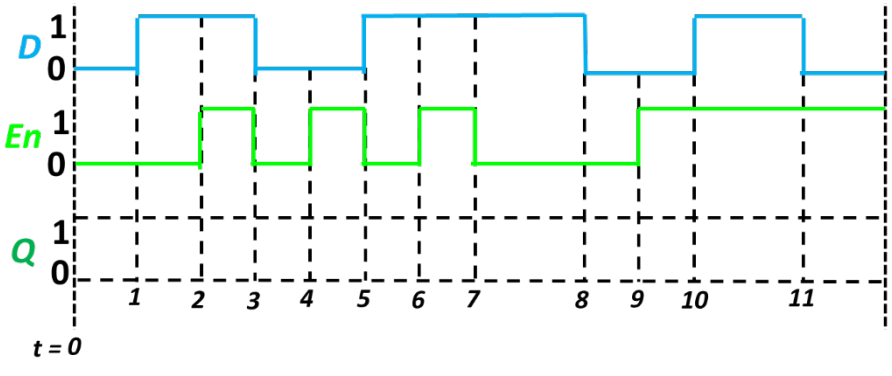
Problem 5

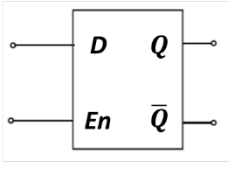
The waveforms illustrated below describe the *J* and *K* inputs of a rising-edge triggered JK flip-flop. Also provided are waveforms for the active-low PRESET and CLEAR asynchronous inputs. Assuming that output *Q* = 0 initially, sketch the *Q* waveform.



Problem 6

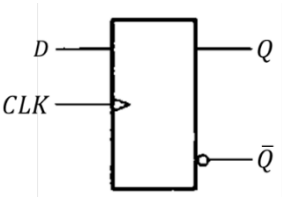
The following waveforms describe the input *D* and the enable signal *En* in a *D* latch. Sketch the normal output waveform *Q*. *Q* equals zero at *t* = 0.

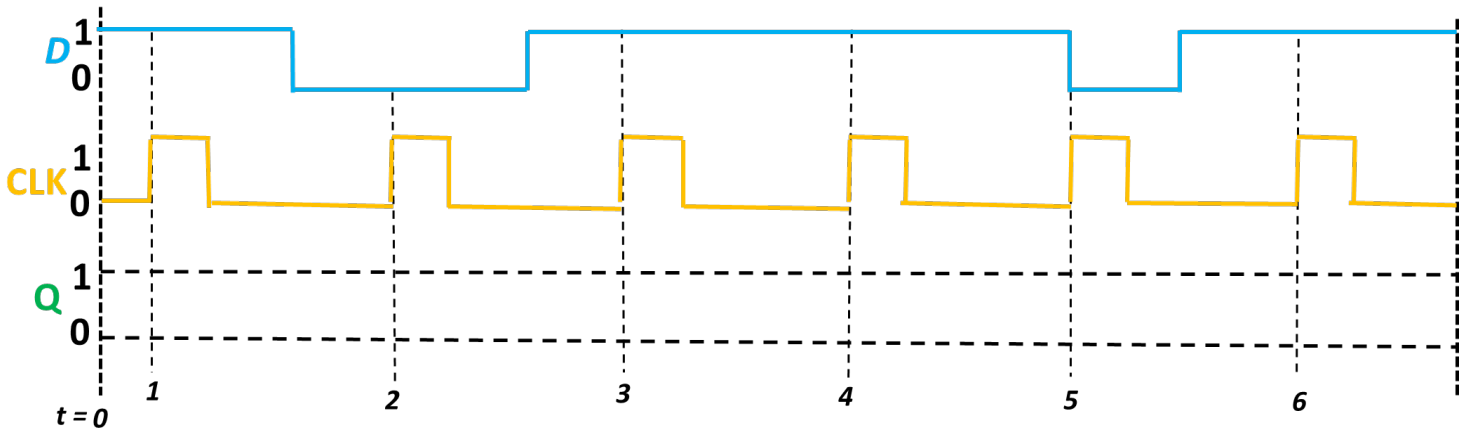




Problem 7

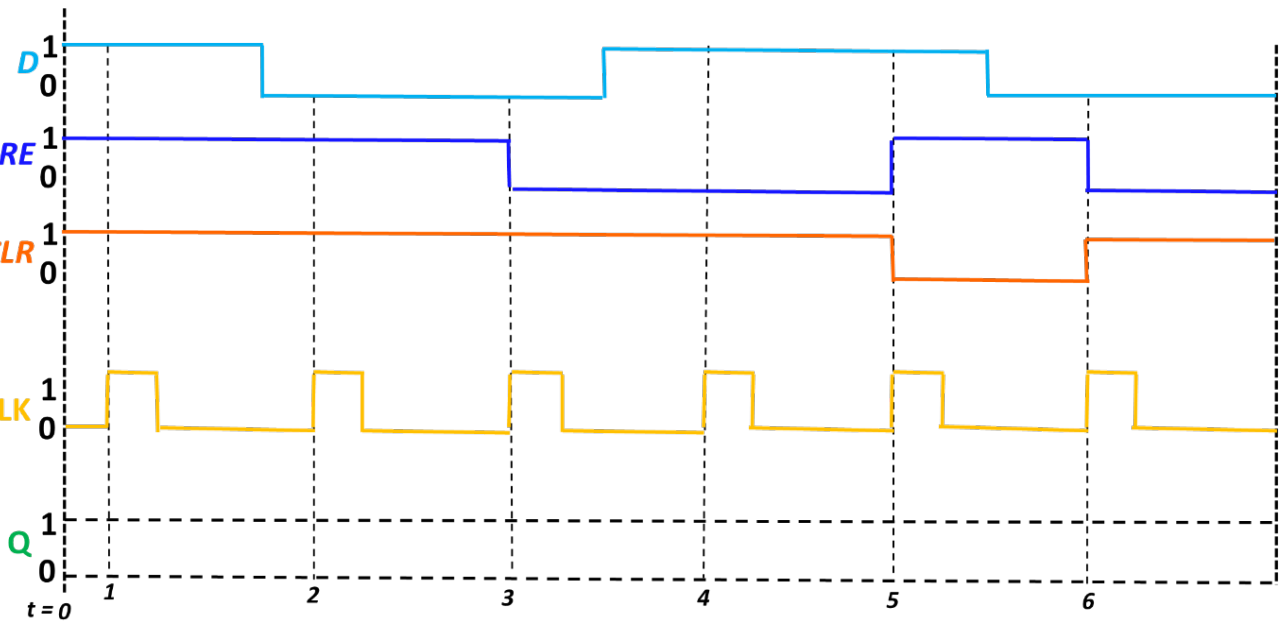
A rising edge-triggered D flip-flop has its input *D* and clock described by the following waveforms. Delays are negligible and the device has no asynchronous inputs. Determine the waveform of normal output *Q*, assuming that *Q* = 0 initially.





Problem 8

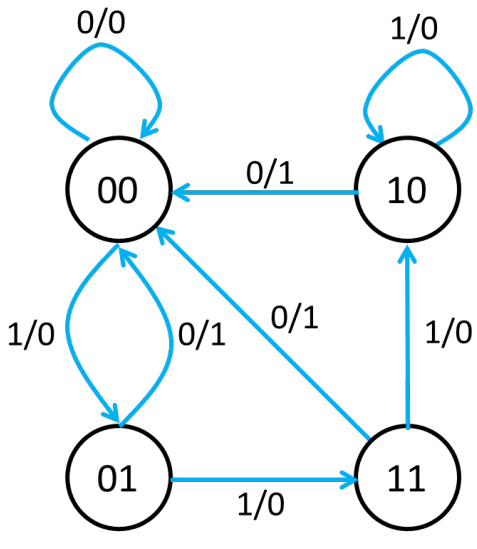
A rising edge-triggered D flip-flop has its input *D* and clock described by the following waveforms. Also provided are waveforms for PRESET and CLEAR asynchronous inputs. Assuming delays are negligible, determine the waveform of normal output *Q*, with *Q* = 0 initially.



Problem 9 (Modified from Mano and Ciletti, 2004, w/ permission)

Starting from state 00 in the state diagram illustrated below, determine the state transitions and output sequence that will be generated when an input sequence described by the following string is applied.

Input: 0 1 1 0 1 1 1 1 0



Problem 10 (Mano and Ciletti, 2004, w/ permission)

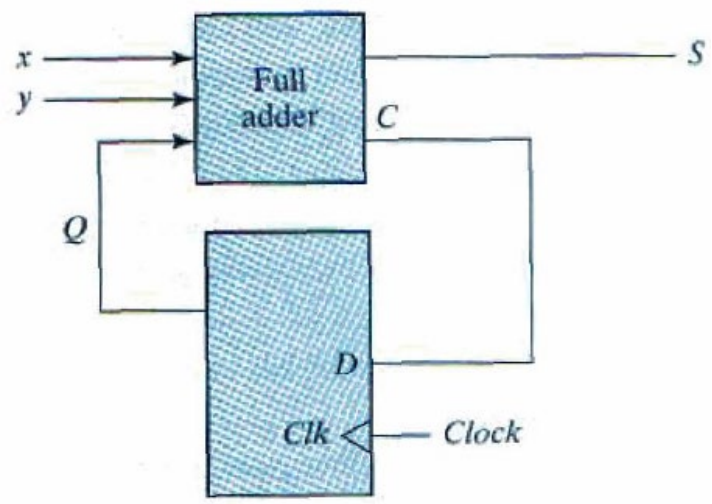
A sequential circuit with two *D* flip-flops *A* and *B*, two inputs *x* and *y*, and one output *z* is specified by the following next-state and output equations:

$$A(t+1) = x'y + xB$$
$$B(t+1) = x'A + xB$$
$$z = A$$

Problem 10.1: Draw the logic diagram of the circuit.
Problem 10.2: List the state table for the sequential circuit.
Problem 10.3: Draw the corresponding state diagram.

Problem 11 (Mano and Ciletti, 2004, w/ permission)

A sequential circuit has one flip-flop *Q*, two inputs *x* and *y*, and one output *S*. It consists of a full-adder circuit connected to a *D* flip-flop, as illustrated below. Derive the state table and the state diagram of the sequential circuit.



Problem 12 (Mano and Ciletti, 2004, w/ permission)

A sequential circuit has two JK flip-flops *A* and *B* and one input *x*. The circuit is described by the following flip-flop input equations:

$$J_A = x \;;\; K_A = B'$$
$$J_B = x \;;\; K_B = A$$

Problem 12.1: Derive the state equations $A(t+1)$ and $B(t+1)$ by substituting the input equations for the *J* and *K* variables.
Problem 12.2: Draw the state diagram of the circuit.

Problem 13 (Mano and Ciletti, 2004, w/ permission)

A sequential circuit has two JK flip-flops *A* and *B*, two inputs *x* and *y*, and one output *z*. The flip-flop input equations and circuit output equation are given below.

$$J_A = Bx + B'y' \;;\; K_A = B'x'y'$$
$$J_B = A'x \;;\; K_B = A + xy'$$
$$z = Ax'y' + Bx'y'$$

Tabulate the state table for this circuit.

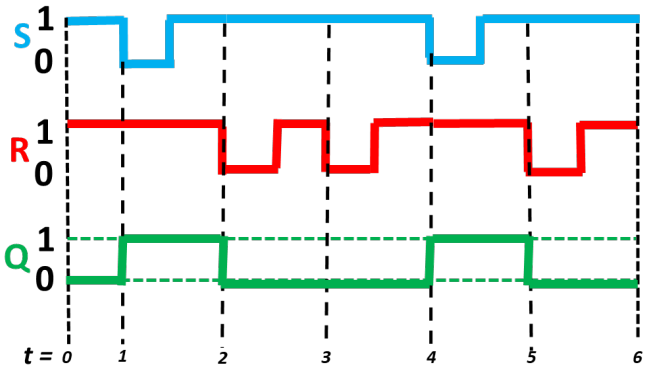
SOLUTIONS

P.1 ➡ *Solution*

The operation of a SR latch made up of NAND gates can be summarized by the following truth table.

Input		Output
S	***R***	
1	1	No change
0	1	$Q = 1$
1	0	$Q = 0$
0	0	$Q = \bar{Q} = 1$ (Invalid)

With these rules in mind, refer to the input waveforms we were given. From $t = 0$ to $t = 1$, both S and R inputs are high and $Q = 0$. At $t = 1$, S changes to 0 and output Q responds by changing to 1. At $t = 2$, R transitions to zero and causes the output Q to return to 0. At $t = 3$, R goes low again but, because Q is already set to 0 and there is no change in input S, the output waveform remains unchanged. At $t = 4$, S returns to 0 and Q responds by going high. At $t = 5$, R transitions to 0 and Q responds by returning to 0. The output waveform then remains unchanged until the final instant $t = 6$. The output waveform is sketched below.

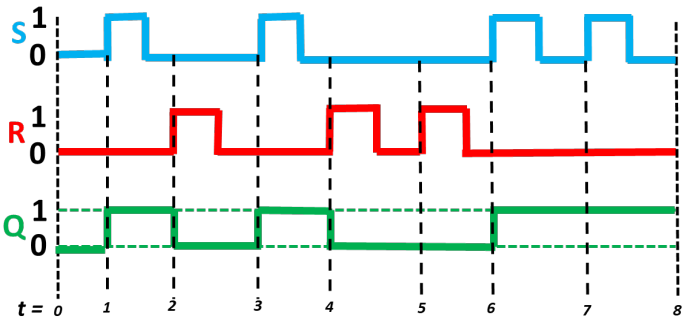


P.2 ➡ *Solution*

The operation of a SR flip-flop made up of NOR gates can be summarized by the following truth table.

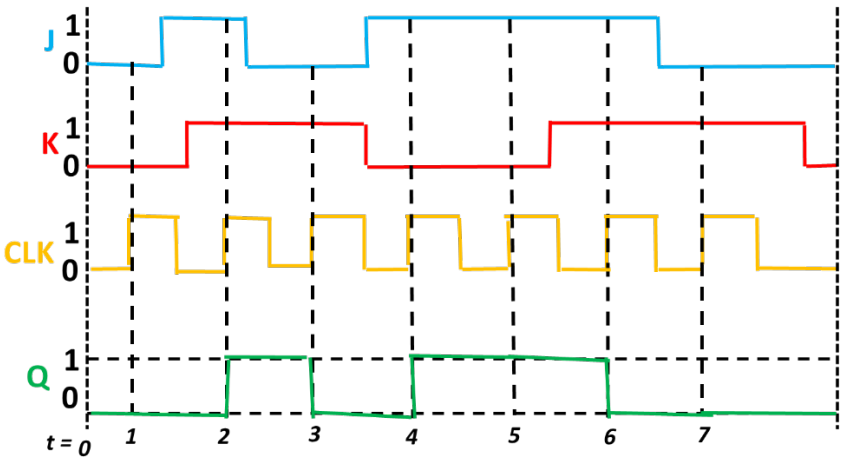
Input		Output
S	***R***	
0	0	No change
1	0	$Q = 1$
0	1	$Q = 0$
1	1	$Q = \bar{Q} = 1$ (Invalid)

With these rules in mind, sketching the output waveform follows the same reasoning as Problem 1.



P.3 ➡ *Solution*

Since the output is initially zero, $Q = 0$ from $t = 0$ to $t = 1$. In the first rising edge of the clock, at $t = 1$, inputs J and K are both zero and the output remains unchanged. In the second rising edge, at $t = 2$, inputs J and K are both high and the output toggles. In the third rising edge, at $t = 3$, input J is zero and K is 1, which causes the output Q to clear to the 0 state. At $t = 4$, with $J = 1$ and $K = 0$, output Q is set to the logic-1 state. At $t = 5$, J and K remain at 1 and 0 respectively, therefore there is no change in output. At $t = 6$, both J and K are equal to 1 and Q responds by toggling to back to 0. Lastly, at $t = 7$, with $J = 0$ and $K = 1$, output Q remains 0.

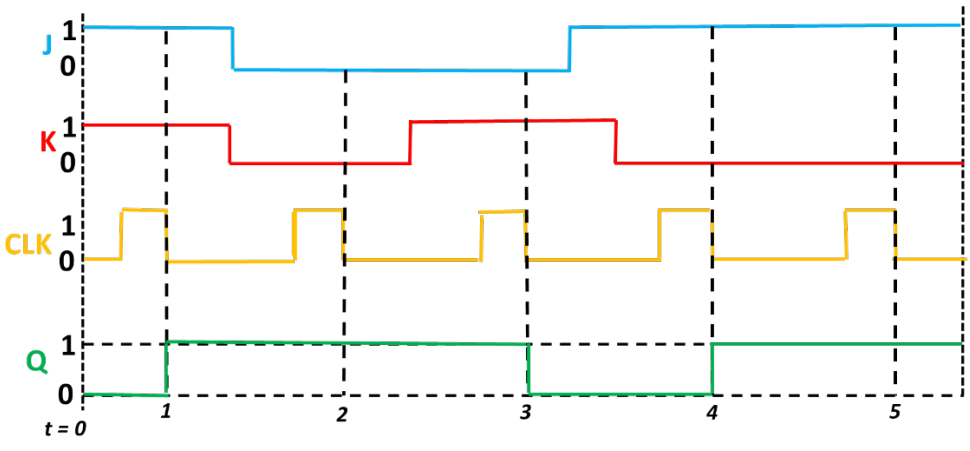


P.4 ➡ *Solution*

The functioning of a falling-edge JK flip-flop is described in the following table. Notice that the operation is similar to that of the JK flip-flop of Problem 3, the difference being that output Q is updated at the falling edge of clock pulses.

Input			Output
J	***K***	***CLK***	
0	0	↓	Q_0 (output remains unchanged)
1	0	↓	1
0	1	↓	0
1	1	↓	Q_0 (toggles)

Since the output is initially zero, $Q = 0$ from $t = 0$ to $t = 1$. In the first falling edge of the clock, at $t = 1$, inputs J and K are both 1 and the output toggles to 1. In the second falling edge, at $t = 2$, both J and K are set to zero and no change in output occurs. In the third falling edge, at $t = 3$, J equals 0 and K equals 1, which causes the output to become 0. In the fourth falling edge, at $t = 4$, J equals 1 and K equals 0, which causes the output to become 1 again. Lastly, in the falling edge at $t = 5$, J and K are still 1 and 0 respectively, and output Q remains equal to logic-1.



P.5 ➡ *Solution*

If *PRESET* = 1 and *CLEAR* = 1, a clocked JK flip-flop with asynchronous inputs will behave like a clocked device no different from the ones analyzed in Problems 3 and 4. Differences arise when one or both asynchronous inputs are not high. In general, there are four scenarios:

1. *PRESET = 1* and *CLEAR = 1*: As mentioned above, such a configuration enables the flip-flop to behave as a normal clocked JK flip-flop.

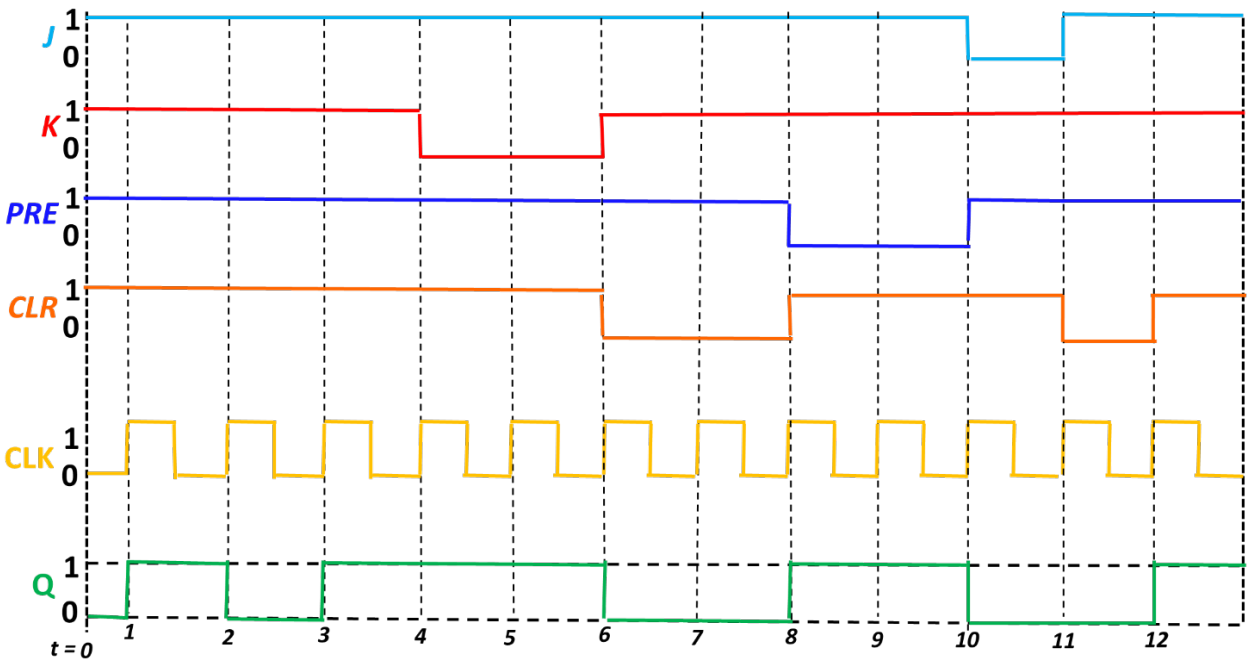
2. *PRESET* = 0 and *CLEAR* = 1: This condition indicates that the asynchronous active-low input PRESET is activated. In this case, the output Q is immediately set to 1, no matter what conditions are present at the J, K, and *CLK* inputs. Importantly, the *CLK* input cannot affect the flip-flop while *PRESET* = 0.

3. *PRESET* = 1 and *CLEAR* = 0: This condition indicates that the asynchronous input CLEAR is activated. In this case, the output Q is immediately set to 0, no matter what conditions are present at the J, K, and CLK inputs. Importantly, the CLK input cannot affect the flip-flop while *CLEAR* = 0.

4. *PRESET* = 0 and *CLEAR* = 0: This condition is not used because it can generate an ambiguous response.

Using these rules, we can sketch the output waveform Q. From $t = 0$ to $t = 6$, the asynchronous inputs PRESET and CLEAR are both set to 1 and hence the device behaves like a *JK* flip-flop subject to the same rules studied in Problems 3 and 4. At $t = 6$, CLEAR goes low and causes Q to become zero. The output remains zero at $t = 7$ because CLEAR is still low at this rising edge and overrides the synchronous inputs J and K. At $t = 8$, CLEAR returns to 1 and PRESET goes low, causing the output to go high. The output remains 1 at $t = 8$ because PRESET is still low at this rising edge and overrides the synchronous inputs J and K. At $t = 10$, the two asynchronous inputs PRESET and CLEAR are back to 1 and the device can once again function as a typical clocked flip-flop; since $J = 0$ and $K = 1$ at this point, Q goes low. At $t = 11$, CLEAR goes low yet again and would override asynchronous inputs by changing Q to logic-0; however, Q is already zero at this instant and no change occurs. At $t = 12$, the

asynchronous inputs return to their high states and regular clocked flip-flop functioning resumes. The output waveform is shown below.



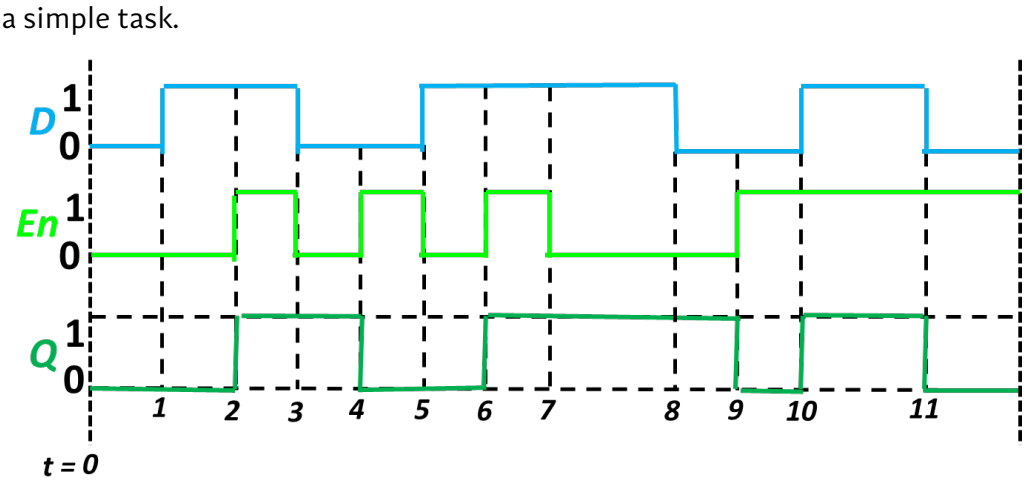
P.6 ➡ *Solution*

The functioning of a *D* latch is straightforward and can be described by two rules:

1. When the enable signal *En* is high, the *Q* output will attain the same level as input *D*. If input *D* changes from zero to 1 or vice versa, *Q* will reproduce these changes.

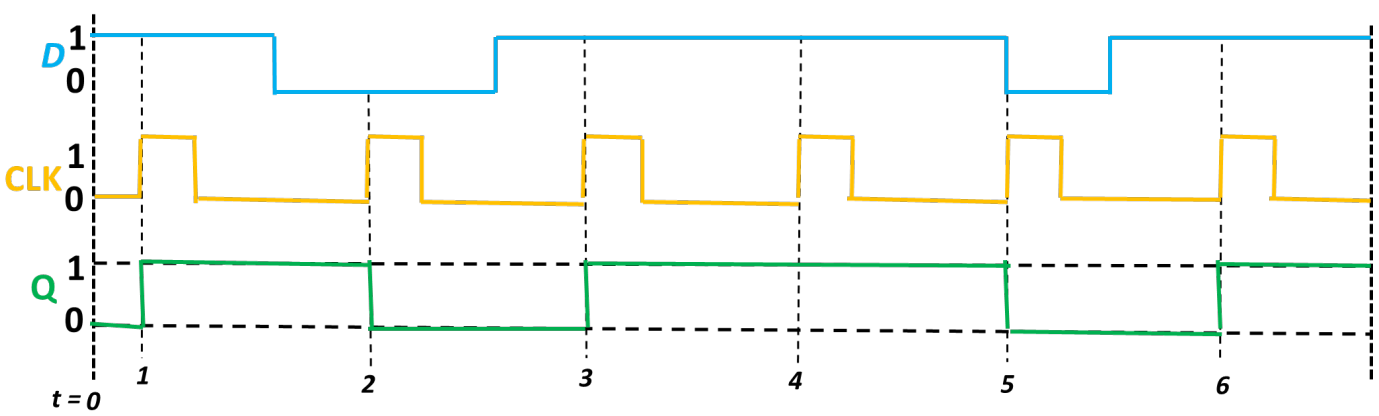
2. When the enable signal *En* is low, the *Q* output will remain at the level it had just before *En* went to zero. *Q* will remain unchanged for variations in input *D* so long as *En* remains low.

With these two rules in mind, sketching the waveform of output *Q* is a simple task.



P.7 ➡ *Solution*

Functioning of a *D* flip-flop hinges primarily on the clock pulse and the level of input *D* at the rising edge of each pulse. Specifically, at each rising edge the output *Q* will assume the state of input *D* at that instant, and will only update at the next rising edge. In the case at hand, the input *D* is high at $t = 1$; accordingly, output *Q* will go high at this instant. Notice that input *D* becomes zero at a point between $t = 1$ and $t = 2$. However, output *Q* will not change to logic-0 itself until the next clock rising-edge, that is, at $t = 2$, as shown. The same reasoning applies to the rest of the clock pulse train.



P.8 ➡ *Solution*

If PRESET = 1 and CLEAR = 1, a *D* flip-flop with asynchronous inputs will behave like a clocked device no different from the one analyzed in Problem 7. Differences arise when one or both asynchronous inputs are not high. In general, there are four scenarios:

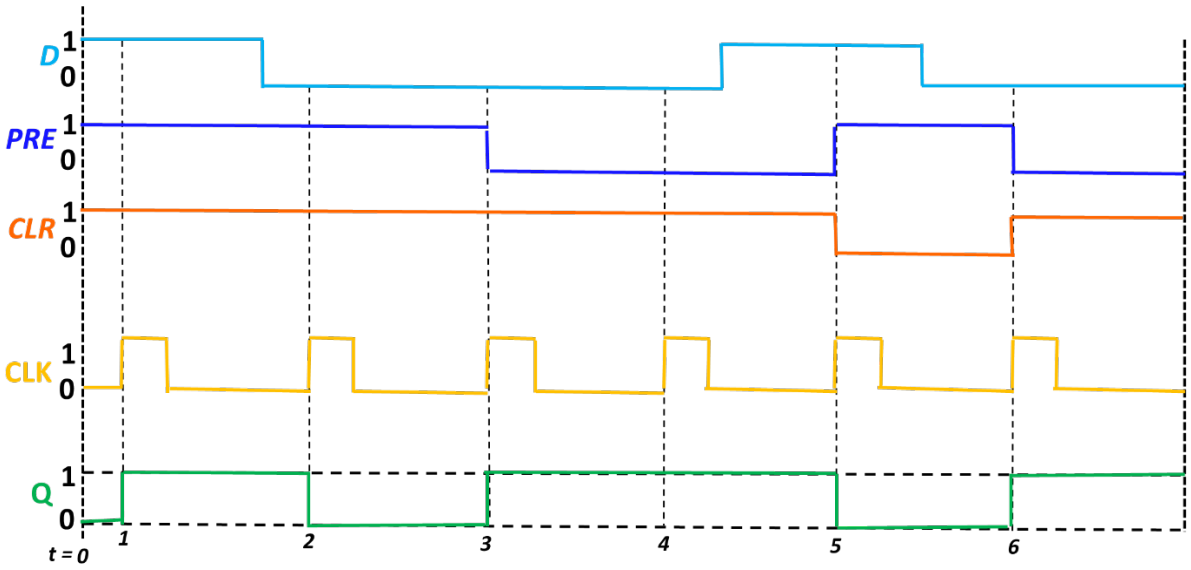
1. *PRESET* = 1 and *CLEAR* = 1: As mentioned above, such a configuration enables the flip-flop to behave as a normal clocked *D* flip-flop with no asynchronous inputs.

2. *PRESET* = 0 and *CLEAR* = 1: This condition indicates that the asynchronous input PRESET is activated. In this case, the output *Q* is immediately set to 1, no matter what conditions are present at the *D* and *CLK* inputs. Importantly, the CLK pulse and the *D* input cannot affect the flip-flop while *PRESET* = 0.

3. *PRESET* = 1 and *CLEAR* = 0: This condition indicates that the asynchronous input CLEAR is activated. In this case, the output *Q* is immediately set to 0, no matter what conditions are present at the *D* and *CLK* inputs. Importantly, the CLK pulse and the *D* input cannot affect the flip-flop while *CLEAR* = 0.

4. *PRESET* = 0 and *CLEAR* = 0: This condition is not used because it can generate an ambiguous response.

With these rules in mind, sketching the output waveform *Q* is quite straightforward. From *t* = 0 to *t* = 3, PRESET and CLEAR are both high and the device can function like a clocked flip-flop with no asynchronous inputs. At *t* = 3, PRESET goes low and the output *Q* responds by becoming 1. In the next rising-edge, at *t* = 4, the *D* signal is low and, were there no asynchronous inputs, the output would respond by assuming a low value as well; however, the low PRESET input overrides *D* and *CLK*, so *Q* remains equal to 1 from *t* = 4 onwards. Then, at *t* = 5, PRESET returns to 1 and CLEAR goes low, which, in accordance with scenario 3 above, causes *Q* to become zero. Lastly, at *t* = 6, CLEAR becomes 1 and PRESET goes low, changing the output *Q* to 1.



P.9 ➡ *Solution*

As the reader may know, the two-digit number inside a circle of a given state diagram indicates the state of the flip-flops in a circuit. Each directed line is accompanied by two numbers separated by a slash; the number to the left of the slash is an input, while the number to the right of the slash is the output attained when the system receives this input, assuming that the circuit was originally in the state at the base of the directed line. For example, the state diagram at hand has a directed line that stems from the circle 10 and points to the circle 00; the line is labeled "0/1," which indicates that an input of 0, applied when the circuit was in state 10, will generate an output of 1. Since the directed line points to the 00 state circle, we surmise that after the next clock cycle the system will transition to state 00.

With these rules in mind and equipped with the state diagram, we can establish the output sequence for the input string we were given. The initial state of the system is 00 and the first input is 0. There is a directed line labeled "0/0" that begins and ends at the 00 state circle; this indicates that an input of 0 will yield an output of 0, and at the next clock pulse the system will remain at state 00. Thus, we can write down the following:

Input: 011011110
Output: 0

Next state: 00

The next input is a 1; a directed line labeled “1/0” joins circle 00 and circle 01, indicating that the output is 0 and the next state is 01. The table is updated accordingly:

Input: 011011110
Output: 00
Next state: 00 01

The next input is a 1; a directed line labeled “1/0” joins circle 01 and circle 11, indicating that the output is 0 and the next state is 11. The table is updated accordingly:

Input: 011011110
Output: 000
Next state: 00 01 11

The next input is a 0; a directed line labeled “0/1” joins circle 11 and circle 00, indicating that the output is 1 and the next state is 00. The table is updated accordingly:

Input: 11011110
Output: 0001
Next state: 00 01 11 00

The next input is a 1; a directed line labeled “1/0” joins circle 00 and circle 01, indicating that the output is 0 and the next state is 01. The table is updated accordingly:

Input: 11011110
Output: 00010
Next state: 00 01 11 00 01

The next input is a 1; a directed line labeled “1/0” joins circle 01 and circle 11, indicating that the output is 0 and the next state is 11. The table is updated accordingly:

Input: 11011110
Output: 000100
Next state: 00 01 11 00 01 11

The next input is a 1; a directed line labeled “1/0” joins circle 11 and circle 10, indicating that the output is 0 and the next state is 10. The table is updated accordingly:

Input: 11011110
Output: 0001000
Next state: 00 01 11 00 01 11 10

The next input is a 1; a directed line labeled “1/0” ends and begins on circle 10, indicating that the output is 0 and the state remains unchanged at 10. The table is updated accordingly:

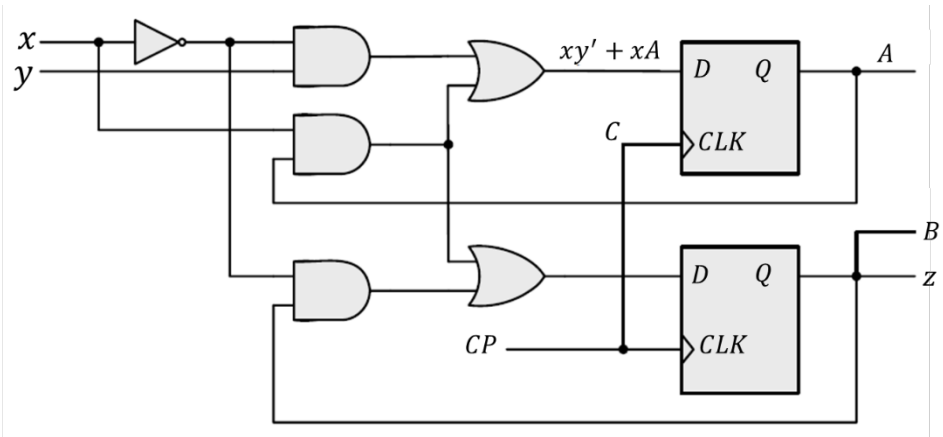
Input: 11011110
Output: 00010000
Next state: 00 01 11 00 01 11 10 10

The last input is a 0; a directed line labeled “0/1” e joins circle 10 and circle 00, indicating that the output is 1 and the next state is 00. The table is updated accordingly:

Input: 011011110
Output: 000100001
Next state: 00 01 11 00 01 11 10 10 00

P.10 ➡ *Solution*

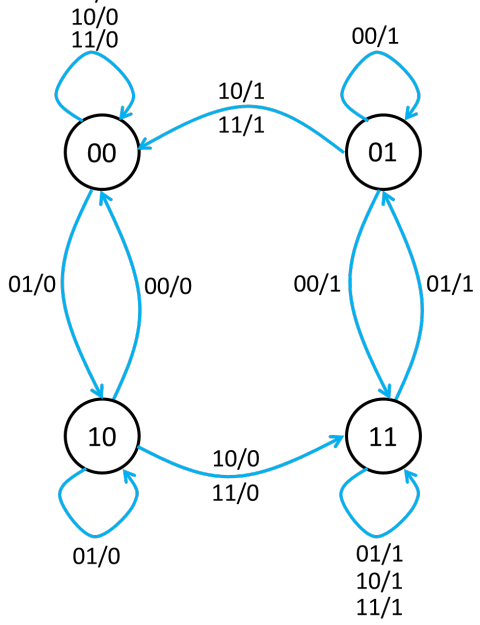
Problem 10.1: The logic diagram is shown below.



Problem 10.2: The state table of the circuit is drawn up as follows.

Present state		Inputs		Next state		Output
A	***B***	***x***	***y***	***A***	***B***	***z***
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	1	1	1
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Problem 10.3: The state diagram is shown below.



P.11 ➡ *Solution*

The state table for the circuit at hand is drawn up as follows.

Present State	Inputs		Next State	Output
Q	***x***	***y***	***Q***	***S***
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

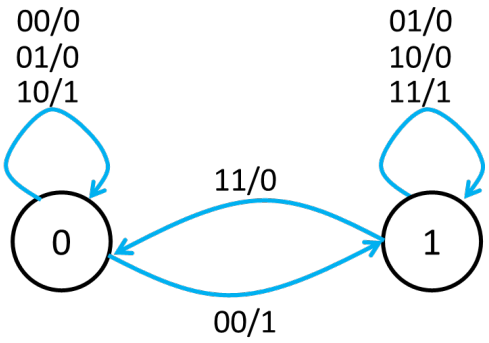
Note that the state equation for Q is

$$Q(t+1) = xy + xQ + yQ$$

Further, the output S is described by

$$S = x \oplus y \oplus Q$$

The state diagram is shown below.



P.12 ➡ *Solution*

Problem 12.1: Since the devices at hand are JK flip-flops, states A and B can be described by the equations

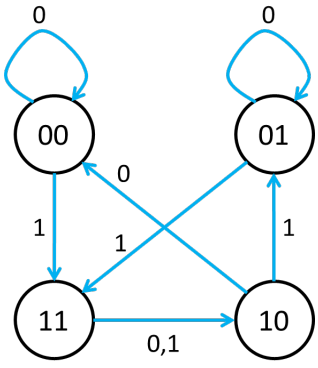
$$A(t+1) = J_A A' + K'_A A$$
$$B(t+1) = J_B B' + K'_B B$$

Problem 12.2: Substituting the inputs J_A, K_A, J_B and K_B, we get

$$A(t+1) = xA' + (B')' A = xA' + BA$$

$$B(t+1) = xB' + (A)' B = xB' + A'B$$

The state diagram is shown below.



P.13 ➡ *Solution*

The state table is drawn up in continuation.

Present State		Inputs		Next State		Outputs	Flip-Flop Outputs			
A	*B*	*x*	*y*	*A*	*B*	z	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1	1	1
0	0	1	1	0	1	0	0	0	1	0
0	1	0	0	0	1	1	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0
0	1	1	0	1	0	0	1	0	1	0
0	1	1	1	1	1	0	1	0	1	0
1	0	0	0	1	0	0	1	0	0	1
1	0	0	1	1	0	0	0	0	0	1
1	0	1	0	0	0	0	1	1	0	1
1	0	1	1	1	0	0	0	0	0	1
1	1	0	0	1	0	1	0	0	0	1
1	1	0	1	1	0	0	0	0	0	1
1	1	1	0	1	0	0	1	0	0	1
1	1	1	1	1	0	1	1	0	0	1

REFERENCES

- FLOYD, T.L. (2015). *Digital Fundamentals*. 11th edition. Upper Saddle River: Pearson.
- MANO, M.M. and CILETTI, M.D. (2004). *Digital Design*. 4th edition. Upper Saddle River: Pearson Prentice-Hall.
- SEDHA, R.S. (2004). *A Textbook of Digital Electronics*. New Delhi: S. Chand and Company.