Supplement to

Logic and Computer Design Fundamentals
4th Edition[1]

Design and Analysis Using JK and T Flip-flops

Selected topics not covered in the fourth edition of *Logic and Computer Design Fundamentals* are provided here for optional coverage and for self-study if desired. This material fits well with the desired coverage in some programs but not may not fit within others due to time constraints or local preferences. This supplement is referenced in Chapter 5 as a part of the coverage of design and analysis of sequential circuits. Although JK and T flip-flops have declined in use, they still appear as components in design libraries and in older designs.

Analysis with Other Flip-Flop Types

So far we have considered the state table for sequential circuits that employ *D*-type flip-flops, in which case the next-state values are obtained directly from the input equations. For circuits with other types of flip-flops, such as *JK*, the next-state values are obtained by following a two-step procedure:

1. Obtain the binary values of each flip-flop input equation in terms of the present-state and input variables.
2. Use the corresponding flip-flop characteristic from Table 6-7 in the text (See Reference 1) to determine the next state.

To illustrate this procedure, consider the sequential circuit with two *JK* flip-flops *A* and *B* and one input *X* specified by the following input equations:

$$J_A = B \qquad K_A = B\overline{X}$$

$$J_B = \overline{X} \qquad K_B = A\overline{X} + \overline{A}X$$

The state table for this circuit is shown in Table 1. The binary values listed under the columns spanned by the header "Flip-flop inputs" are not part of the state table. They are needed for the purpose of evaluating the next state as specified in step 2 of the procedure. These binary values are obtained directly from the four input equations in a manner similar to that for obtaining a truth table from an algebraic expression. The next state of each flip-flop is evaluated from the corresponding J and K inputs and the characteristic table of the JK flip-flop in Table 6-7 of the text. There are four cases to consider. When $J = 1$ and $K = 0$, the next state is 1. When $J = 0$ and $K = 1$, the next state is 0. When $J = K = 0$, there is no change of state, and the next-state value is the same as that of the present state. When $J = K = 1$, the next-state bit is the complement of the present-state bit. An example of the $J = K = 0$ case occurs in the table when the present-state and input (A, B, X) are 100. J_A and K_A are both equal to 0, and the present state of A is 1. Therefore, the next state of A remains the same and is equal to 1. In the same row of the table, J_B and K_B are both equal to 1. Since the present state of B is 0, the next state of B is complemented and changes to 1.

TABLE 1
State Table for Circuit with *JK* Flip-Flops

Present state		Input	Next state		Flip-flop inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	1	0	0	1
1	0	0	1	1	0	0	1	1
1	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0

An alternative analysis procedure uses the characteristic equation for the JK flip-flop from Table 6-7

$$Q(t+1) = J\overline{Q(t)} + \overline{K}Q(t)$$

We substitute the equations for J_A, K_A, J_B, and K_B into copies of the above equation for $A(t + 1)$ and $B(t + 1)$. The resulting equations for $A(t + 1)$ and $B(t + 1)$ are

$$A(t+1) = B\overline{A} + B\overline{\overline{X}}A = \overline{A}B + A\overline{B} + AX$$

$$B(t+1) = \overline{X}\overline{B} + \overline{(A\overline{X} + \overline{A}X)}B = \overline{B}\overline{X} + \overline{A}B\overline{X} + ABX$$

Evaluating these two equations gives the same values as those in the next state columns in Table 1.

Design with Other Flip-Flop Types

The design of a sequential circuit with flip-flops other than the D type is complicated by the fact that the flip-flop input equations for the circuit must be derived indirectly from the state table. When D-type flip-flops are employed, the input equations are obtained directly from the next state. This is not the case for the JK and other types of flip-flops. In order to determine the input equations for these flip-flops, it is necessary to derive a functional relationship between the state table and the input equations.

Flip-Flop Excitation Tables The flip-flop characteristic tables presented in Table 6-7 of the text provide the value of the next state when the values of the inputs and the present state are known. These tables are used to analyze sequential circuits and to define the operation of the flip-flops. During the design process, we usually know the transition from the present state to the next state and wish to find the flip-flop input conditions that will cause this transition. Accordingly, we need a table that lists the required inputs for a given change of state. Such a table is called an *excitation table*. Table 6-7 presents the excitation tables for four different types of flip-flops. Each table has a column for the present state $Q(t)$, a column for the next state $Q(t + 1)$, and a column for each flip-flop input to show how the required transition is achieved. There are four possible transitions from the present state to the next state. The required input conditions for each of these transitions are derived from the information available in the characteristic table. The symbol X in the table represents a don't-care condition, which means, again, that it does not matter whether the input is 0 or 1.

The excitation table for the JK flip-flop can be derived from the knowledge of how the flip-flop operates. Consider the first of the entries in the excitation table for the JK flip-flop in Table 6-7. The transition from a present state of 0 to a next state of 0 can be accomplished in two ways. If $J = 0$ and $K = 0$, there is no change of state, and the flip-flop stays at 0. If $J = 0$ and $K = 1$, the flip-flop resets to 0. This dictates that J must be equal to 0, but K can be either 0 or 1, and in either case, the required transition occurs. This is indicated in the first row of the table by a 0 under J and a don't-care symbol X under K. The transformation from a present state of 0 to a next state of 1 can also be done in two ways. Letting $J = 1$ and $K = 0$ sets the flip-flop to 1. Letting $J = 1$ and $K = 1$ complements the flip-flop from 0 to 1. For this case it is necessary that J be equal to 1, but it does not matter whether K is 0 or 1. Either way, the proper transition happens. This information is listed in the second row of the table. In a similar manner, it is possible to derive the rest of the entries in the excitation tables for the JK flip-flop and the other three flip-flops. The excitation tables for the T and D flip-flops have no don't-care conditions and can be specified with an excitation function. The excitation table for the D flip-flop shows that the next state is always equal to the D input and is independent of the present state. This can be represented algebraically by the excitation function

$$D = Q(t+1)$$

Therefore, the values for input D can be taken directly from the values in the next-state column. The excitation table for the T flip-flop shows that the T input is

TABLE 2
State Table with *JK* Flip-Flop Inputs

Present State		Input	Next State		Flip-Flop Inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

equal to the exclusive-OR of the present state and the next state. This can be expressed by the excitation function

$$T = Q(t) \oplus Q(t+1)$$

DESIGN PROCEDURE The design procedure for sequential circuits with *JK* flip-flops is the same as that for sequential circuits with *D* flip-flops, except that the flip-flop input equations must be evaluated from the present-state to next-state transition derived from the excitation table. To illustrate the procedure, we design the sequential circuit specified by Table 2. Assume that A and B are the outputs. In addition to having columns for the present state, input, and next state, as in a conventional state table, the table also shows the flip-flop input values, from which the input equations are derived. The flip-flop inputs in the table are derived from the state table in conjunction with the excitation table for the *JK* flip-flop. For example, in the first row of Table 2, a transition for flip-flop *A* occurs from 0 as the present state to 0 as the next state. In the exitation table for the *JK* flip-flop in Table 6-7, a transition from present state 0 to next state 0 requires that input *J* be 0 and input *K* be a don't-care condition. So 0 and X are entered in the first row under J_A and K_A. Since the first row also shows a transition for flip-flop *B* from 0 in the present state to 0 in the next state, 0 and X are inserted in the first row under J_B and K_B, respectively. The second row of the table shows a transition for flip-flop *B* from 0 as the present state to 1 as the next state. From the excitation table, we find that a transition from 0 to 1 requires that *J* be 1 and *K* be a don't-care condition, so 1 and X are copied in the second row under J_B and K_B, respectively. This process is continued for each row in the table and for each flip-flop, with the input conditions from the excitation table copied into the proper row of values for the particular flip-flop being considered.

The flip-flop inputs in Table 2 specify the truth table for the flip-flop input equations as a function of the present-state variables *A* and *B* and input *X*. The input equations are simplified in the maps of Figure 1. The next-state values are not used during the simplification, since the input equations are a function of the

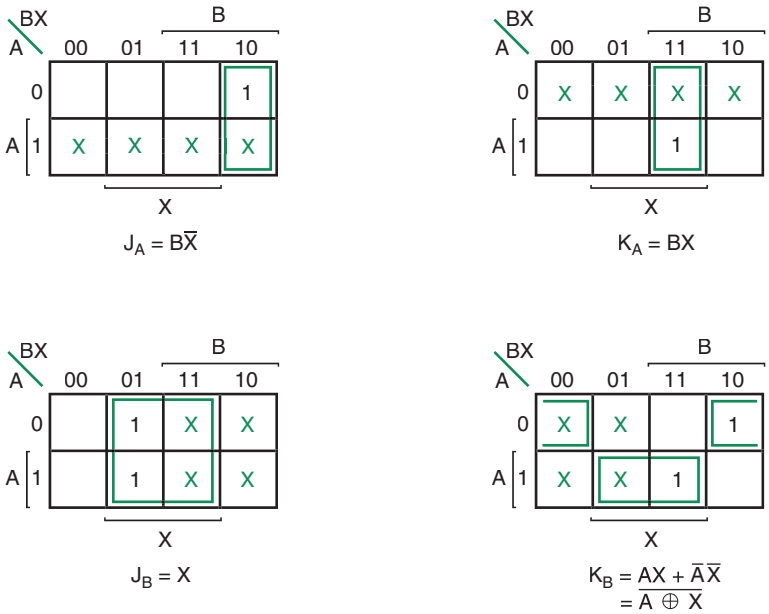

FIGURE 1
Maps for *J* and *K* Input Equations

present state and input only. Note the advantage of using *JK*-type flip-flops when designing sequential circuits. The fact that there are so many don't-care entries indicates that the combinational circuit for the input equations is likely to be simpler, because don't-care minterms usually help in obtaining simpler expressions. If there are unused states in the state table, there may be additional don't-care conditions in the map. The four input equations for the two *JK* flip-flops are listed under the maps of Figure 1. The logic diagram of the sequential circuit is drawn in Figure 2 implements the equations for flip-flop inputs contains an asynchronous reset input *R* which for a 1 applied, resets the two flip-flops to 0, initializing the state.

As a final example, consider the design of a sequential circuit with *T* flip-flops. Using the state table portion of Table 1, we obtain the binary values for the *T* flip-flop inputs from the excitation functions for *A* and *B*:

$$T_A = A(t) \oplus A(t+1)$$

$$T_B = B(t) \oplus B(t+1)$$

Going over the present-state and next-state values for *A* and *B* in the table, we determine the binary values for T_A and T_B in the following manner. For each present-state to next-state transition from 0 to 1 or from 1 to 0, we place a 1 on the corresponding *T* input; for each case where there is no change (from 0 to 0 or from 1 to 1), we place a 0 on the corresponding *T* input. The resulting values appear in

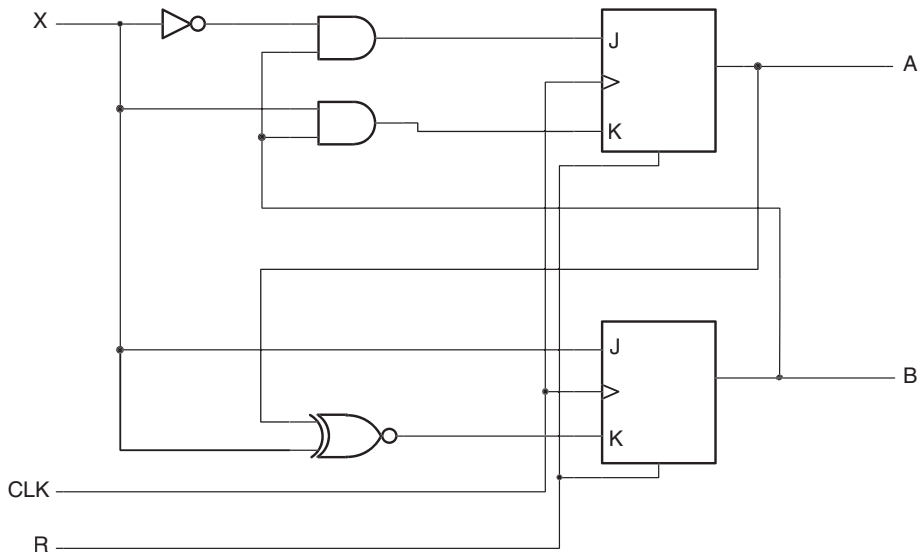

FIGURE 2
Logic Diagram for Circuit Represented by Table 2

Table 3. The input equations can be obtained by using the minterms that produce a 1 in the flip-flop input columns for T_A and T_B:

$$T_A(A,B,X) = \Sigma\, m(2,7) = \overline{A}B\overline{X} + ABX$$

$$T_B(A,B,X) = \Sigma\, m(1,2,5,7) = \overline{A}B\overline{X} + ABX + \overline{B}X$$

The algebraic expressions are the simplified input equations from which one can draw the gates of the sequential circuit. The circuit consists of two T flip-flops, three AND gates (due to the common terms $\overline{A}B\overline{X}$ and ABX in T_A and T_B), two OR gates, and an inverter. Remember that a T flip-flop can be constructed from a JK flip-flop with inputs J and K tied together to form a single input T.

References

1. MANO, M. M. AND C. R. KIME. *Logic and Computer Design Fundamentals, 4th* ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2008.
2. MANO, M. M. *Digital Design,* 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
3. WAKERLY, J. F. *Digital Design: Principles and Practices,* 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2000.

Problems

The plus (+) indicates a more advanced problem.

1. *A set-dominant flip-flop has set and reset inputs. It differs from a conventional *SR* flip-flop in that, when both *S* and *R* are equal to 1, the flip-flop is set.
   (a) Obtain the characteristic table of the set-dominant flip-flop.
   (b) Find the state diagram for the set-dominant flip-flop.
   (c) Design the set-dominant flip-flop by using a J-K flip-flop and AND gates, OR gates and inverters.

2. A *JN* flip-flop has two inputs *J* and *N*. Input *J* behaves like the *J* input of a *JK* flip-flop, and input *N* behaves like the complement of the *K* input of a *JK* flip-flop (that is $N = \overline{K}$).
   (a) Obtain the characteristic table of the flip-flop.
   (b) Show that, by connecting the two inputs *J* and *N* together, one obtains a *D*-type flip-flop.

3. **(a)** Derive an excitation table for the set-dominant flip-flop defined in Problem 1.
   (b) Derive the excitation table for *JN* flip-flop defined in Problem 2.

4. *Design a sequential circuit for the state diagram given in Figure 6-40 of the text (see Reference 1) using *JK* flip-flops.

5. *Design a sequential circuit with two *JK* flip-flops *A* and *B* and two inputs *E* and *X*. If $E = 0$, the circuit remains in the same state, regardless of the value of *X*. When $E = 1$ and $X = 1$, the circuit goes through the state transitions from 00 to 01 to l0 to 11, back to 00, and then repeats. When $E = 1$ and $X = 0$, the circuit goes through the state transitions from 00 to 1 to 10 to 01, back to 00, and then repeats.

6. Using the state table part of Table 3 below, derive a new list of binary values for the inputs of two *T* flip-flops T_A and T_B. Find the flip-flop input equations for T flip-flops and draw the logic diagram of the circuit with *T* flip-flops assuming that A and B are outputs.

TABLE 3
State Table with *JK* Flip-Flop Inputs

Present State		Input	Next State	
A	B	X	A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

7. Write a VHDL description for a J-K negative edge-triggered flip-flop with clock CLK. Compile and simulate your description. Apply a sequence that causes all eight combinations of inputs J and K and stored value Q to be applied in some clock cycle.

8. Write a Verilog description for a J-K negative edge-triggered flip-flop with clock CLK. Compile and simulate your description. Apply a sequence that causes all eight combinations of inputs J and K and stored value Q to be applied in some clock cycle.