JKSM: State Machines Using J-K Flip-Flops

This section covers analysis and synthesis of state machines that are built using J-K flip-flops.

JKSM.1 Analysis of State Machines with J-K Flip-Flops

Clocked synchronous state machines built from J-K flip-flops can also be analyzed by the basic procedure in the preceding subsection. The only difference is that there are two excitation equations for each flip-flop—one for J and the other for K. To obtain the transition equations, both of these must be substituted into the J-K's characteristic equation, $Q* = J \cdot Q' + K' \cdot Q$.

Figure JKSM-1 is an example state machine using J-K flip-flops. Reading the logic diagram, we can derive the following excitation equations:

$$\begin{aligned} J0 &= X \cdot Y' \\ K0 &= X \cdot Y' + Y \cdot Q1 \\ J1 &= X \cdot Q0 + Y \\ K1 &= Y \cdot Q0' + X \cdot Y' \cdot Q0 \end{aligned}$$

Substituting into the characteristic equation for J-K flip-flops, we obtain the transition equations:

$$\begin{aligned} Q0* &= J0 \cdot Q0' + K0' \cdot Q0 \\ &= X \cdot Y' \cdot Q0' + (X \cdot Y' + Y \cdot Q1)' \cdot Q0 \\ &= X \cdot Y' \cdot Q0' + X' \cdot Y' \cdot Q0 + X' \cdot Q1' \cdot Q0 + Y \cdot Q1' \cdot Q0 \\ Q1* &= J1 \cdot Q1' + K1' \cdot Q1 \\ &= (X \cdot Q0 + Y) \cdot Q1' + (Y \cdot Q0' + X \cdot Y' \cdot Q0)' \cdot Q1 \\ &= X \cdot Q1' \cdot Q0 + Y \cdot Q1' + X' \cdot Y' \cdot Q1 + Y' \cdot Q1 \cdot Q0' + X' \cdot Q1 \cdot Q0 + Y \cdot Q1 \cdot Q0 \end{aligned}$$

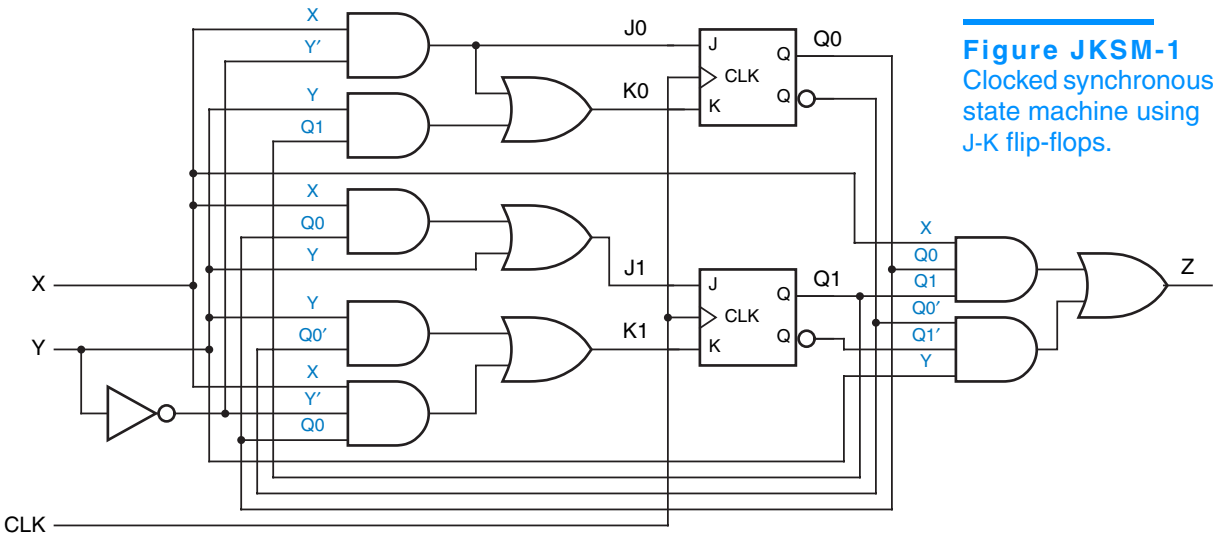


Figure JKSM-1
Clocked synchronous state machine using J-K flip-flops.

(a)

	XY			
Q1 Q0	00	01	10	11
00	00, 0	10, 1	01, 0	10, 1
01	01, 0	11, 0	10, 0	11, 0
10	10, 0	00, 0	11, 0	00, 0
11	11, 0	10, 0	00, 1	10, 1
	Q1* Q0*, Z			

(b)

	XY			
S	00	01	10	11
A	A, 0	C, 1	B, 0	C, 1
B	B, 0	D, 0	C, 0	D, 0
C	C, 0	A, 0	D, 0	A, 0
D	D, 0	C, 0	A, 1	C, 1
	S*, Z			

Table JKSM-1 Transition/output and state/output tables for the state machine in Figure JKSM-1.

A transition table based on these equations is shown in Table JKSM-1(a). By reading the logic diagram, we can write the output equation:

$$Z = X \cdot Q1 \cdot Q0 + Y \cdot Q1' \cdot Q0'$$

The resulting output values are shown in each column of the table along with the next state. Assigning state names A–D, we obtain the state/output table shown in (b). A corresponding state diagram that uses transition expressions is shown in Figure JKSM-2.

JKSM.2 Synthesis Using J-K Flip-Flops

At one time, J-K flip-flops were popular for discrete SSI state-machine designs, since a J-K flip-flop embeds more functionality than a D flip-flop in the same size SSI package. By "more functionality" we mean that the combination of J and K inputs yields more possibilities for controlling the flip-flop than a single D input does. As a result, a state machine's excitation logic may be simpler using J-K flip-flops than using D flip-flops, which reduced package count when SSI gates were used for the excitation logic.

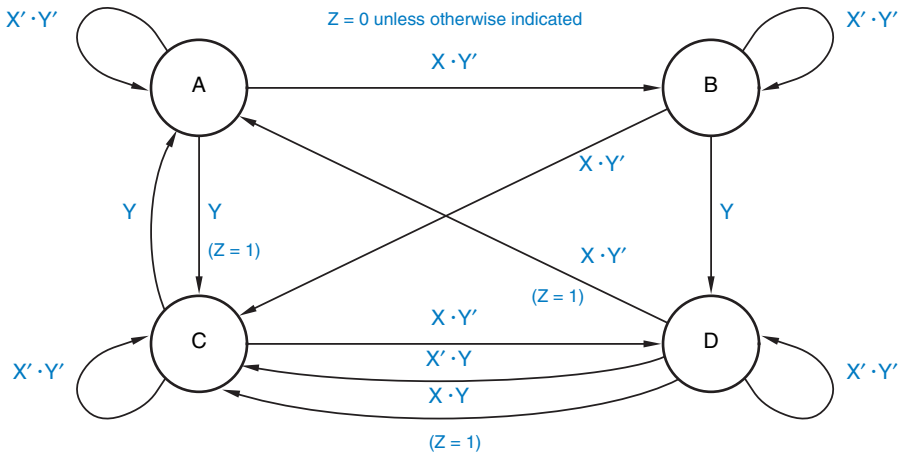


Figure JKSM-2 State diagram corresponding to the state machine of Table JKSM-1.

JUST FOR FUN While minimizing excitation logic was a big deal in the days of SSI-based design, the name of the game has changed with PLDs and ASICs. As you might guess from your knowledge of the AND-OR structure of combinational PLDs, the need to provide separate AND-OR arrays for the J and K inputs of a J-K flip-flop would be a distinct disadvantage in a sequential PLD.

In ASIC technologies, J-K flip-flops aren't so desirable either. For example, in LSI Logic Corp.'s LCA500K series of CMOS gate arrays, an FD1QP D flip-flop macrocell uses 7 "gate cells," while an FJK1QP J-K flip-flop macrocell uses 9 gate cells, over 25% more chip area. Therefore, a more cost-effective design usually results from sticking with D flip-flops and using the extra chip area for more complex excitation logic in just the cases where it's really needed.

Still, this subsection describes the J-K synthesis process "just for fun."

Up through the state-assignment step, the design procedure with J-K flip-flops is basically the same as with D flip-flops. The only difference is that a designer might select a slightly different state assignment, knowing the sort of behavior that can easily be obtained from J-K flip-flops (e.g., "toggling" by setting J and K to 1).

The big difference occurs in the derivation of an excitation table from the transition table. With D flip-flops, the two tables are identical; using the D's characteristic equation, $Q^* = D$, we simply substitute $D = Q^*$ for each entry. With J-K flip-flops, each entry in the excitation table has twice as many bits as in the transition table, since there are two excitation inputs per flip-flop.

J-K application table

A J-K flip-flop's characteristic equation, $Q^* = J \cdot Q' + K' \cdot Q$, cannot be rearranged to obtain independent equations for J and K. Instead, the required values for J and K are expressed as functions of Q and Q^* in a *J-K application table*, Table JKSM-2. According to the first row, if Q is currently 0, all that is required to obtain 0 as the next value of Q is to set J to 0; the value of K doesn't matter. Similarly, according to the third row, if Q is currently 1, the next value of Q will be 0 if K is 1, regardless of J's value. Each desired transition can be obtained by either of two different combinations on the J and K inputs, so we get a "don't-care" entry in each row of the application table.

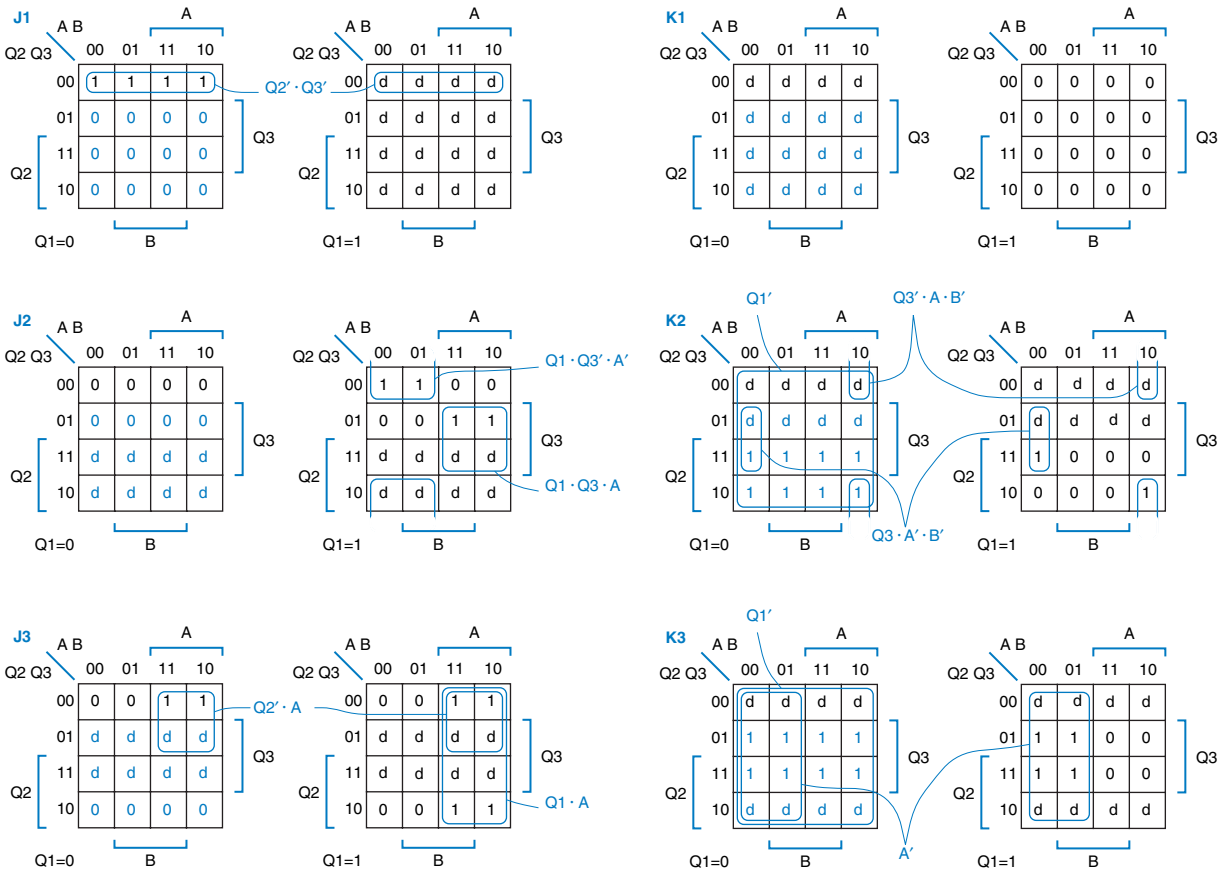
Table JKSM-2 Application table for J-K flip-flops.

Q	*Q**	*J*	*K*
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

To obtain a J-K excitation table, the designer must look at both the current and desired next value of each state bit in the transition table and substitute the corresponding pair of J and K values from the application table. For the transition table in Table 7-7 on page 563, these substitutions produce the excitation table in Table JKSM-3. For example, in state 100 under input combination 00, Q1 is 1 and the required Q1* is 1; therefore, "d0" is entered for J1 K1. For the same state/input combination, Q2 is 0 and Q2* is 1, so "1d" is entered for J2 K2. Obviously, it takes quite a bit of patience and care to fill in the entire excitation table (a job best left to a computer).

As in the D synthesis example of Section 7.4.4, the excitation table is *almost* a truth table for the excitation functions. This information is transferred to Karnaugh maps in Figure JKSM-3.

Figure JKSM-3 Excitation maps for J1, K1, J2, K2, J3, and K3, assuming that unused states go to state 000.



	A B				
Q1 Q2 Q3	00	01	11	10	Z
000	1d, 0d, 0d	1d, 0d, 0d	1d, 0d, 1d	1d, 0d, 1d	0
100	d0, 1d, 0d	d0, 1d, 0d	d0, 0d, 1d	d0, 0d, 1d	0
101	d0, 0d, d1	d0, 0d, d1	d0, 1d, d0	d0, 1d, d0	0
110	d0, d0, 0d	d0, d0, 0d	d0, d0, 1d	d0, d1, 1d	1
111	d0, d1, d1	d0, d0, d1	d0, d0, d0	d0, d0, d0	1
	J1 K1, J2 K2, J3 K3				

Table JKSM-3 Excitation and output table for the state machine of Table 7-7, using J-K flip-flops.

The excitation table does not specify next states for the unused states, so once again we must choose between the minimal-risk and minimal-cost approaches. The colored entries in the Karnaugh maps result from taking the minimal-risk approach.

Note that even though the "safe" next state for unused states is 000, we didn't just put 0s in the corresponding map cells, as we were able to do in the D case. Instead, we still had to work with the application table to determine the proper combination of J and K needed to get $Q* = 0$ for each unused state entry, once again a tedious and error-prone process.

Using the maps in Figure JKSM-3, we can derive sum-of-products excitation equations:

$$J1 = Q2' \cdot Q3' \qquad K1 = 0$$

$$J2 = Q1 \cdot Q3' \cdot A' + Q1 \cdot Q3 \cdot A \qquad K2 = Q1' + Q3' \cdot A \cdot B' + Q3 \cdot A' \cdot B'$$

$$J3 = Q2' \cdot A + Q1 \cdot A \qquad K3 = Q1' + A'$$

MINIMAL-COST SOLUTION

In the preceding design example, excitation maps for the minimal-cost approach would have been somewhat easier to construct, since we could have just put d's in all of the unused state entries. Sum-of-products excitation equations obtained from the minimal-cost maps (shown in Section Min.2) are as follows:

$$J1 = 1 \qquad K1 = 0$$

$$J2 = Q1 \cdot Q3' \cdot A' + Q3 \cdot A \qquad K2 = Q3' \cdot A \cdot B' + Q3 \cdot A' \cdot B'$$

$$J3 = A \qquad K3 = A'$$

The state encoding for the J-K circuit is the same as in the D circuit, so the output equation is the same, $Z = Q1 \cdot Q2$ for minimal risk, $Z = Q2$ for minimal cost.

A logic diagram based on the minimal-cost equations above is shown in Figure JKSM-4. This circuit has two more gates than by the minimal-cost D flip-flop circuit (see box on page 566), so J-K flip-flops still didn't save us anything.

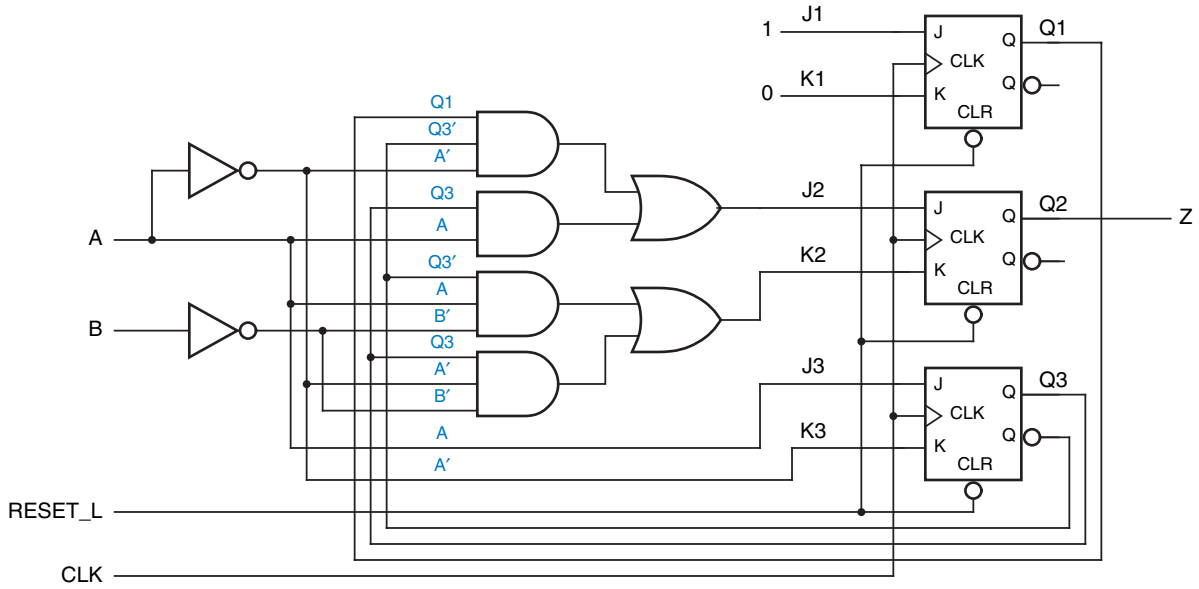


Figure JKSM-4 Logic diagram for example state machine using J-K flip-flops and minimal-cost excitation logic.

These equations take two more gates to realize than do the preceding subsection's minimal-risk equations using D flip-flops, so J-K flip-flops didn't save us anything in this example, least of all design time.

Exercises

JKSM.1 Analyze the clocked synchronous state machine in Figure xJKSM.1. Write excitation equations, transition equations, transition table, and state/output table (use state names A–D for Q1 Q2 = 00–11). Draw a state diagram, and draw a timing diagram for CLK, X, Q1, and Q2 for 10 clock ticks, assuming that the machine starts in state 00 and X is continuously 1.

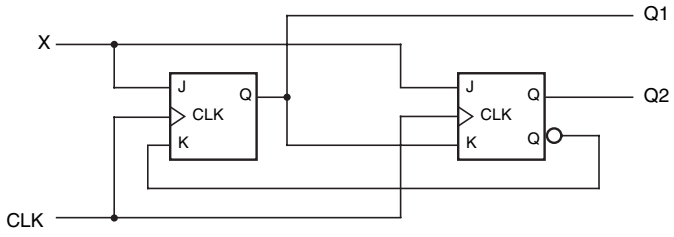


Figure xJKSM.1

JKSM.2 Analyze the clocked synchronous state machine in Figure xJKSM.2. Write excitation equations, transition equations, transition table, and state/output table (use state names A–D for Q1 Q0 = 00–11). Draw a state diagram, and draw a timing diagram for CLK, EN, Q1, and Q0 for 10 clock ticks, assuming that the machine starts in state 00 and EN is continuously 1.

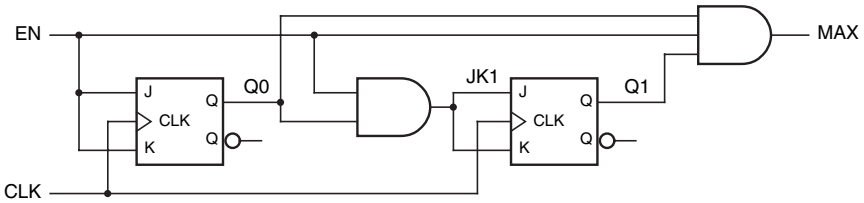


Figure xJKSM.2

JKSM.3 Design a clocked synchronous state machine with the state/output table shown in Table xJKSM.3, using J-K flip-flops. Use two state variables, Q1 Q2, with the state assignment A = 00, B = 01, C = 11, D = 10.

Table xJKSM.3

	X		
S	0	1	Z
A	B	D	0
B	C	B	0
C	B	A	1
D	B	C	0
	S*		

JKSM.4 Write new transition and excitation tables and derive minimal-cost excitation and output equations for the state table in Table 7-5 using the "simplest" state assignment in Table 7-6 and J-K flip-flops. Compare the cost of your excitation and output logic (when realized with a two-level AND-OR circuit) with the circuit in Figure JKSM-4.

JKSM.5 Write new transition and excitation tables and derive minimal-cost excitation and output equations for the state table in Table 7-5 using the "almost one-hot" state assignment in Table 7-6 and J-K flip-flops. Compare the cost of your excitation and output logic (when realized with a two-level AND-OR circuit) with the circuit in Figure JKSM-4.

JKSM.6 Determine the full 8-state table for the state machine of Figure JKSM-4. Use the names U1, U2, and U3 for the states that are unused in the original state table (001, 010, and 011). Draw a state diagram and explain the behavior of the unused states.

JKSM.7 In many applications, the outputs produced by a state machine during or shortly after reset are irrelevant, as long as the machine begins to behave correctly a short time after the reset signal is removed. If this idea is applied to Table 7-5, the INIT state can be removed and only two state variables are needed to code the remaining four states. Redesign the state machine using this idea. Write a new state table, transition table, excitation table for J-K flip-flops, minimal-cost excitation and output equations, and logic diagram. Compare the cost of the new circuit with that of Figure JKSM-4.

JKSM.8 Redesign the 1s-counting machine of Table 7-9, assigning the states in binary counting order (S0–S3 = 00, 01, 10, 11) and using J-K flip-flops. Compare the cost of the resulting sum-of-products excitation equations with the ones derived in the text.

JKSM.9 Given the transition equations for a clocked synchronous state machine that is to be built using J-K flip-flops, how can the excitation equations for the J and K inputs be derived? (*Hint:* Show that any transition equation, Qi* = expr, can be written in the form Qi* = Qi · expr1 + Qi′ · expr2, and see where that leads.) How can the "don't-cares" that are possible in a J-K design be specified?

JKSM.10 Design a circuit that meets the specifications of Exercise 7.89 using edge-triggered J-K flip-flops (74x109) and NAND and NOR gates without feedback loops. Give a complete circuit diagram and word description of how your circuit achieves the desired behavior.

JKSM.11 Is the circuit in the preceding exercise subject to metastability, and if so, under what conditions?

JKSM.12 Construct an application table similar to Table JKSM-2 for each of the following flip-flop types: (a) S-R; (b) T with enable; (c) D with enable. Discuss the unique problem that you encounter when trying to make the most efficient use of don't-cares with one of these flip-flops.

JKSM.13 Construct a new excitation table and derive minimal-cost excitation and output equations for the state machine of Table 7-7 using T flip-flops with enable inputs (Figure 7-33). Compare the cost of your excitation and output logic (when realized with a two-level AND-OR circuit) with the circuit in Figure Min-3 in Section Min.

JKSM.14 Write the application table for a T flip-flop with enable.

JKSM.15 Redesign the 1s-counting machine of Table 7-9, assigning the states in binary counting order (S0–S3 = 00, 01, 10, 11) and using T flip-flops with enable. Compare the cost of the resulting sum-of-products excitation equations with the ones derived in the text.

JKSM.16 Design a clocked synchronous state machine that checks a serial data line for even parity. The circuit should have two inputs, SYNC and DATA, in addition to CLOCK, and one Moore-type output, ERROR. Devise a state/output table that does the job using just four states and include a description of each state's meaning in the table. Choose a 2-bit state assignment, write transition and excitation equations, and draw the logic diagram. Your circuit should use one D and one J-K flip-flop.