

EECE 144
Fall 2011

Lab Report #10
Section 4
11/9/2011

Submitted by: Jeremiah Mahler

Signature

Printed Name

Date

| | | |
|--|-------------------------|--------------------|
| | Jeremiah Mahler | Nov 9, 2011 |
| | Marvanee Johnson | Nov 9, 2011 |

1 Description/Objectives

The objective of this lab is to design a three bit binary counter using at least one D flip-flop and at least one JK flip-flop. And then implement this design in hardware.

| n | Z_2 | Z_1 | Z_0 |
|---|-------|-------|-------|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

Table 1: The sequence of states for a 3-bit binary counter where Z_0 is the least significant bit. The counter should follow this order and when the end is reached it should loop to the beginning.

2 Procedure

The first step is to design the 3-bit counter. The strategy used here is to first solve the least significant bit and then proceed to the more significant bits. This is because the least significant bit does not depend on the state of any of the higher bits. The second bit depends on the state of the first bit. And the third bit depends on the state of the second and possibly the first.

Here we used one D flip-flop for the first bit and two JK flip-flops for subsequent bits. But other combinations are possible. In fact it is possible to use any register type for any bit position. It is even possible to implement a JK flip-flop out of a D flip-flop using only a few additional logic gates.

And after the design is created it will be implemented in hardware.

2.1 bit Z_0 , toggle using D flip-flop

The least significant bit, when it is triggered by the clock, toggles by setting the output (Q^+) to the opposite of the previous state (Q).¹

The relevant states for this bit (Z_0) are given in Table 2. It can be seen that when Q is 0 we want the next state, Q^+ , to be 1. And the only way a D flip-flop can produce a 1 as an output is if it has a 1 on its input (D). Similar reasoning can be used for the second row. To construct a function we need to decide what mapping from Q would produce the values in D . In this case it is simply $D = Q'$. What this function means in terms of the D flip-flop is that the output (Q') is connected to the input (D). This is similar in design to a Johnson Counter.

| Q | Q+ | D |
|---|----|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Table 2: States used to create a toggle using a D flip-flop.

2.2 bit Z_1 , using JK flip-flop

Bit Z_1 is more complicated because it depends on the state of Z_0 . Table 3 shows the relevant states for this bit. The value of Q^+ is the desired value of Z_1 during the next state given the current values of Z_0 and Z_1 . The values of each column of J and K are the values which will produce the desired value of Q^+ .

To find a mapping between the values of Z_0 and Z_1 to the inputs of J and K on the flip-flop, Karnaugh Maps were used as shown in Figure 1. And this resulted in the equations $J = Z_0$ and $K = Z_0$. Because they are both equal to Z_0 this means that both the J and K inputs are connected together and to Z_0 . And this will result in the desired output of Z_1 on Q of this JK flip-flop.

¹The notation used here denotes Q as the current state and Q^+ as the next state.

| (Q) | | (Z_1) | | | | | | |
|-------|-------|---------|---|---|---|---|---|---|
| Z_1 | Z_0 | Q^+ | J | K | J | K | J | K |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | X |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | X |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | X | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | X | 1 |

Table 3: States used for the JK flip-flop of bit Z_1 . The three columns of J K are equally valid solutions but the one containing "don't cares" (X) is the most general.

| | | | |
|-------|---|---|--|
| $J :$ | | Z_0 | |
| Z_1 | $\begin{array}{ c c } \hline 0 & 1 \\ \hline \end{array}$ | $\begin{array}{ c c } \hline 0 & 1 \\ \hline \end{array}$ | |
| | $\begin{array}{ c c } \hline X & X \\ \hline \end{array}$ | $\begin{array}{ c c } \hline X & X \\ \hline \end{array}$ | |

| | | | |
|-------|---|---|--|
| $K :$ | | Z_0 | |
| Z_1 | $\begin{array}{ c c } \hline X & X \\ \hline \end{array}$ | $\begin{array}{ c c } \hline X & X \\ \hline \end{array}$ | |
| | $\begin{array}{ c c } \hline 0 & 1 \\ \hline \end{array}$ | $\begin{array}{ c c } \hline 0 & 1 \\ \hline \end{array}$ | |

Figure 1: Karnaugh map for bit Z_1 resulting in the equations $J = Z_0$ and $K = Z_0$.

2.3 bit Z_2 , using JK flip-flop

Bit Z_2 is the most complicated because it depends on the state of Z_0 and Z_1 . Table 4 shows the relevant states for this bit.

The value of Q^+ is the desired value of Z_2 during the next state given the current values are Z_0 , Z_1 and Z_2 . The values of each column of J and K are the values which will produce the desired value of Q^+ .

| (Q) | | | (Z_2) | | | | | | |
|-------|-------|-------|---------|---|---|---|---|---|---|
| Z_2 | Z_1 | Z_0 | Q^+ | J | K | J | K | J | K |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | X |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | X |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | X |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | X |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | X | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | X | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | X | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | X | 1 |

Table 4: States used for the JK flip-flop of bit Z_2 .

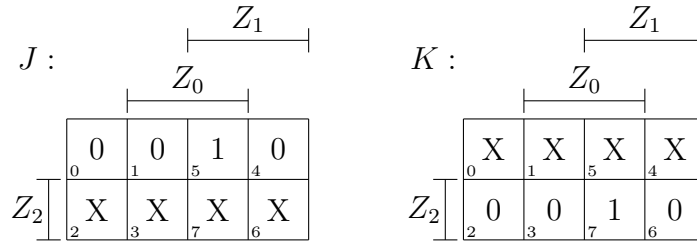


Figure 2: Karnaugh map for bit Z_2 resulting in the equations $J = Z_1Z_0$ and $K = Z_1Z_0$.

To find a mapping between the values of Z_0 , Z_1 and Z_2 to the inputs of J and K on the flip-flop, Karnaugh Maps were used as shown in Figure 2. And this resulted in the equations $J = Z_1Z_0$ and $K = Z_1Z_0$. In this case the inputs to J and K are the same but they also involve another gate. Z_1 and Z_0 must be anded together and then input to J and K . And this will result in the desired output of Z_2 on Q of this JK flip-flop.

2.4 Implementation in hardware

Combining the design for each individual bit results in the circuit shown in Figure 3.

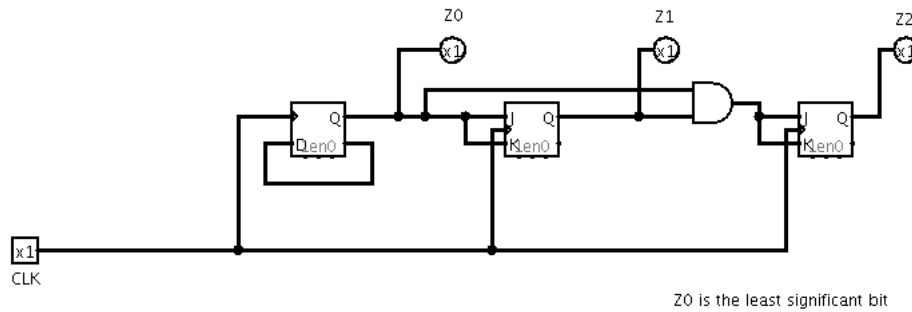


Figure 3: A 3 bit counter using 1 D flip-flop and 2 JK flip-flops with bit Z_0 as the least significant bit.

To implement this design the circuit diagram should be followed along with the relevant data sheets for the particular chips.

Table 5 list the chips that were used during this lab. Other chips can be substituted but they must all be from the same family. CMOS chips cannot be mixed with TTL chips.

On the flip-flops there are additional pins for "reset" and "clear". These are used to momentarily reset and/or clear the state of the chip. When the chips power up the count will be in some random state. But since the counter goes through all possible states there will be no erroneous states and it is not necessary to clear/reset the counter. The pins can be connected to hi/low for normal operation (refer to the data sheet).

| | |
|---------|------------------------|
| 74HC74 | dual D flip-flop |
| 74HC109 | dual JK flip-flop |
| 74HC08 | quad 2 input AND gates |
| 74HC04 | hex inverting gates |

Table 5: List of chips used in the lab.

All the chips used the same clock inputs. And this can be produced from a mechanical switch or a function generator. If a function generator is used the frequency should be reduced to a value such as 1 hz so that it does not change too fast to be seen.

3 Observations

The 3 bit counter implemented in hardware sequentially counted from 0 to 7 as expected. On power up the initial state was not always the same. This was not a problem in this case since all possibilities were accounted for but it could be a problem in other situations. The reset and clear pins could also be used to reset the state if this was necessary.

4 Conclusion

This lab was a success in designing and implementing a 3-bit binary counter in hardware using D and JK flip-flops. It is indeed possible to build a counter using D or JK flip-flops in any of the bit positions, although it may necessitate a few additional gates. And this procedure could be used to design counters with more bits or different features (up/down).