# Topic X: Sequential Logic Design

ILOs for the Design Project: Sequential Logic Design

1. Understand how to produce decimal output on a 7 Segment Display from BCD using a decoder
2. Produce excitation tables from state transition tables
3. Sequential Logic Design: Use Karnaugh mapping to design a gate & flip-flop implementation that produces a desired timing diagram

Design Project Videos:

1. Sequential Logic Design: <https://www.youtube.com/watch?v=R0GszmzNrHg&list=PLhbHWgMknRJT_eKLFXB843NkaNHfJ37Pw&index=35>
2. Lab Info Overview: <https://www.youtube.com/watch?v=ZukvUoI_nZk&list=PLhbHWgMknRJT_eKLFXB843NkaNHfJ37Pw&index=36>
3. Hantek Lab Modification: N/A

## X.1 Sequential Digital Logic Design

### X.1.1 Sequential Logic Design

The truth table of a flip-flop tells you what the next output will be given the current inputs (J & K for a JK Flip-Flop):

JK Truth Table

|  |  |  |
| --- | --- | --- |
| J | K | Qnext |
| 0 | 0 | q (hold) |
| 0 | 1 | 0 (reset) |
| 1 | 0 | 1 (set) |
| 1 | 1 | !q (toggle) |

In contrast, an **excitation table** tells you which inputs (or "excitations") the flip-flop will need in order to make a desired state transition:

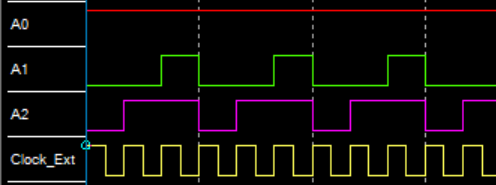
JK Excitation Table

|  |  |  |  |
| --- | --- | --- | --- |
| q (current state) | Q (next state) | J | K |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

Consider the following flip-flop circuit:



Taking A0 to be the output from the flip-flop on the left, A1 as the middle flip-flop output, and A2 as the output from the flip-flop on the right, the circuit has the following timing diagram.



Calling  the current "state" of the flip-flop circuit, it transitions through 001, 101, 111, and then repeats. A table of these state transitions is called a **state transition table:**

|  |  |  |
| --- | --- | --- |
| A2 | A1 | A0 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| 0 | 0 | 1 |

**Sequential logic design** means determining the inputs required to flip-flops to cause their outputs to move them through the states in the state transition table. We'll do this in the following steps:

1. Write a state transition table
2. Use the JK excitation table to determine what the J & K inputs to each flip-flop should be in each state in order to make it transition to the next state on the next clock triggering
3. For each J or K column, use a Karnaugh map to determine the minimum logic function for that output from the inputs.

Referring to the JK Excitation Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Qn | Qn+1 | J | K |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

... the completed state transition table is:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A2 | A1 | A0 | J2 | K2 | J1 | K1 | J0 | K0 |
| 0 | 0 | 1 | 1 | X | 0 | X | X | 0 |
| 1 | 0 | 1 | X | 0 | 1 | X | X | 0 |
| 1 | 1 | 1 | X | 1 | X | 1 | X | 0 |
| 0 | 0 | 1 |  |  |  |  |  |  |

Then fill in the known states for a K-map for each input; e.g., J2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A2\A1A0 | 00 | 01 | 11 | 10 |
| 0 |  | 1 |  |  |
| 1 |  | X | X |  |

Any state not in the table gets a don't care in the k-map, because it can't occur:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A2\A1A0 | 00 | 01 | 11 | 10 |
| 0 | X | 1 | X | X |
| 1 | X | X | X | X |

So, J2 = 1.

For K2 we get:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A2\A1A0 | 00 | 01 | 11 | 10 |
| 0 | X | X | X | X |
| 1 | X | 0 | 1 | X |

meaning K2 = A1

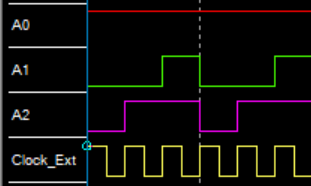
etc.

Repeating this for the other columns gives J1 = A2, K1 = 1, J0 = X, K0 = 0.

And we end up with a far simpler design:



Which nevertheless produces the same timing diagram:

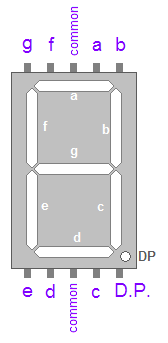


This example started with a known circuit and timing diagram and used sequential logic design to produce a simpler implementation, but it's also a general tool to produce a circuit which reproduces a given timing diagram from scratch.

### X.1.2 Seven-Segment Display (AKA 7SD or SSD)

A 7 segment display (or 7SD) is an array of 7 long LEDs made to show different numbers depending on which combination of LEDs is on.

7SD Common Cathode-style:



Src: <http://www.instructables.com/id/Seven-Segment-Display-Tutorial/>

e.g., to make a 0, you'd send a 1 to e, d, c, b, a, and f, and send a 0 to g. To make a 4, you'd send a 1 to f, g, b, and c and send a 0 to e, d, and a.

In order to display a BCD (binary coded decimal) nibble on a 7SD you need to convert it into the right configuration of 1s and 0s for the LEDs; e.g., convert 0100 (4) into abcdefg = 0110011, convert 0000 (0) into abcdefg = 1111110, convert 0011 (3) into 1111001, etc.

A BCD-to-7SD decoder chip is a chip which contains logic gates in the correct configuration to do this. e.g., the 4511 series:

<http://www.ti.com/lit/ds/symlink/cd74hc4511.pdf>

