Lab 10: Three-out-of-Four Detection

Arturo Salinas-Aguayo
CSE 2301: Principles and Practice of Digital Logic Design
Dr. Mohammad Khan, Section 003L-1248
Electrical and Computer Engineering Department

College of Engineering, University of Connecticut
Coded in LaTeX

Theory

A '101' or '010' Serial Sequence Detector

This task involved designing a Moore machine from the problem statement. The sucessfull implementation will yield a 1 if either the detected sequence was 101 or 010.

The Difference between Combinational and Sequential Logic

In order to properly define these two, here are their definitions.

- **Combinational** - This is a circuit, which is just a network that processes discrete-valued variables. The outputs of combinational circuit depend only on the inputs to the circuit. It is *memoryless*.
- **Sequential** - The outputs of sequential circuits depend on the current input and the past inputs. These circuits are said to *have memory*. However, sequential circuits may either remember directly or distill the information as the *state*.

Discussion

Practice Questions

Example 1 A Counter Utilizing the J-K Flip-Flop
Using JK flip flops, design a counter that counts from DCBA=0000 sequentially to DCBA=1011 and then returns to 0000. Complete the table below.

#	D	C	B	A	D^*	C^*	B^*	A^*	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	0	0	1	?	?	?	?	?	?	?	?
1	0	0	0	1	0	0	0	1	?	?	?	?	?	?	?	?
2	0	0	1	0	0	0	0	1	?	?	?	?	?	?	?	?
3	0	0	1	1	0	0	0	1	?	?	?	?	?	?	?	?
4	0	1	0	0	0	0	0	1	?	?	?	?	?	?	?	?
5	0	1	0	1	0	0	0	1	?	?	?	?	?	?	?	?
6	0	1	1	0	0	0	0	1	?	?	?	?	?	?	?	?
7	0	1	1	1	0	0	0	1	?	?	?	?	?	?	?	?
8	1	0	0	0	0	0	0	1	?	?	?	?	?	?	?	?
9	1	0	0	1	0	0	0	1	?	?	?	?	?	?	?	?
10	1	0	1	0	0	0	0	1	?	?	?	?	?	?	?	?
11	1	0	1	1	0	0	0	1	?	?	?	?	?	?	?	?
12	1	1	0	0	0	0	0	1	?	?	?	?	?	?	?	?
13	1	1	0	1	0	0	0	1	?	?	?	?	?	?	?	?
14	1	1	1	0	0	0	0	1	?	?	?	?	?	?	?	?
15	1	1	1	1	0	0	0	1	?	?	?	?	?	?	?	?

Table 1: J-K Flip Flop Sequence

Example 2 The Difference between Mealy and Moore Machines
In sequential logic, there are two ways to illustrate the forms of how a circuit operates called *Finite State Machines*. These forms show the *next state logic* and the *output logic*.

In general, a FSM contains 2^k *finite* output states such that there are M inputs, N outputs, and k bits of state.

In a **Moore Machine**, the outputs depend exclusively on the *current state* of the machine. This means that the output remains consistent as long as the system stays in a particular state, regardless of changes in the inputs. Consequently:

- **Predictable Outputs**: Because the outputs are tied solely to the state, they are stable and do not change unexpectedly due to momentary fluctuations in the inputs.
- **Design Simplicity**: A Moore Machine is often simpler to design because the output logic only needs to account for the state, not the inputs.

In contrast, a **Mealy Machine**'s outputs are determined by a combination of the *current state* **and** the *current input*. This makes the output sensitive to changes in the input, allowing for faster responses:

- **Responsive Outputs**: Since outputs are based on both the state and input, a Mealy Machine can react immediately to changes in inputs, making it more dynamic.

- **Complexity**: The dual dependency on both inputs and state can make a Mealy Machine more complex to design and predict. However, it can also lead to fewer states being required, as the inputs directly influence the outputs.

The ticket-to-ride for this concept, is that Moore Machines only rely on Current State. Mealy Machines rely on Current State and the Input. This allows the Mealy Machine FSM to be one clock cycle ahead of the Moore since it actively senses the input.

The *D Flip-Flop* or *Delay Flip-Flop* are widely used to form shift or storage registers. This only has one data input D, a clock CLK, and the outputs Q and $\overline{Q}$. But first, some terms to describe these *things*.

- Transparent - When Data, D flows to Output Q.
- Opaque - When Data, D is blocked from flowing and Q retains its old value.
- Master (Leader) - When two back to back flip-flops or latches are controlled by complimentary clocks and the output of the Master(Leader) Q flows into the input of the Slave(Follower) D
- Slave (Follower) - The second latch or flip-flop in the complimentary clock chain. This follows what the Master does.
- Edge-Triggered - When the rising or falling "edge" of a signal causes the logic to advance
- Level-Triggered - When the input being high or low causes the signal to advance. More in later example.

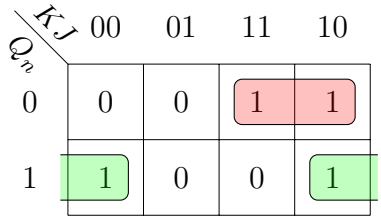
The D Flip Flop is just two D latches tied together which are simply SR Latches that are clocked. More information on the distinction between these can be found in the next example.

To convert from one to the other, the the table describing Q_n, our current state, and Q_{n+1}, the next state is populated logically. This will form the inputs to the D Flip-Flop. Recall that the J and K inputs correspond to Set and Reset from the SR Latch.

Q_n	J	K	Q_{n+1}	D_i
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Table 2: Mapping J-K Logic to D Logic

I use a Karnaugh Map to simplify this mapping:



Therefore,

$$D = Q_n\overline{K} + \overline{Q_n}J$$