Analysis and Design of Sequential Systems

How do we design a *combinational* system?

This approach will not work for *sequential* systems! Why not?

So, a different approach is needed to design sequential circuits.

But…good news… we have already dealt with many of the tools involved. We will need to use:

- State tables: a description of how the system behaves, showing the inputs and present state, and the next state (and possibly outputs).
- State diagrams: each state is a vertex, and arcs indicate how the inputs drive us from one state to another.

Think: How many states can a sequential circuit have?

The set of states corresponds to all possible values of the bits stored in the sequential logic circuit's memory. If there are three bits of storage available in the sequential circuit, there will be at most

If there are four bits of storage available in the sequential circuit, there will be at most

and so forth. Thus, our sequential logic circuit shifts from state to state, depending on the bits held in its memory, and the current inputs. Note that the number of states is *finite*.

So we can imagine our sequential circuit bouncing around from state to state as time progresses. It is common for the sequential circuit to produce an output during each transition

Again: What causes a sequential circuit to arrive in a specific state?

Again: What does the sequential circuit do based on this?

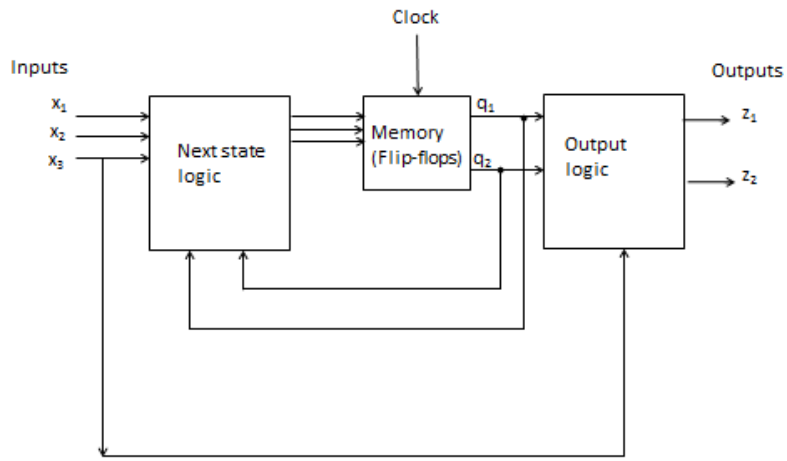
Note that if the sequential logic circuit is in a specific state, say *X*, and then has specific inputs, say *I*, it will always go to a next specific state as determined by its circuitry, say state *Y*. Again, if the circuit is in state *X* and has input *I*, it will be driven deterministically to state *Y*. It will ***not*** be driven to state *Y* today, but under the same conditions to some other state, say *Z*, tomorrow. So, the circuit behaves in a mechanical machine-like way.

Since a sequential logic circuit has a finite number of states and behaves like a machine, it is called a ***finite state machine.***

The pictorial state diagram abstraction for our sequential logic circuit is represented as real hardware, by considering two important functions:

- **The next state logic**. Given the inputs and the current state, how do we determine the next state?
- **The output logic**. This function determines the output. The output function might depend only on the state, or might depend on the inputs as well as the state.

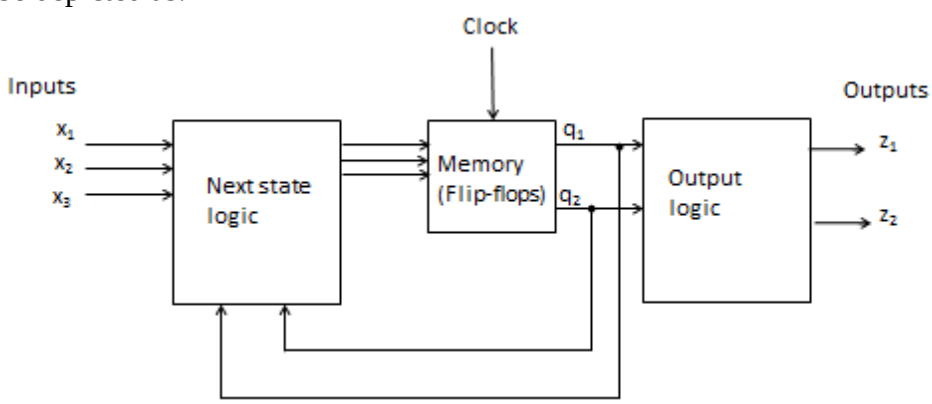
Thus, a circuit with three inputs, two flip flops and two outputs, where the output also depends on input x_3, would be depicted as:



A system such as the one above, where the output depends on the current input(s) in addition to the state, is termed a *Mealy system*.

<u>Simplifications</u>

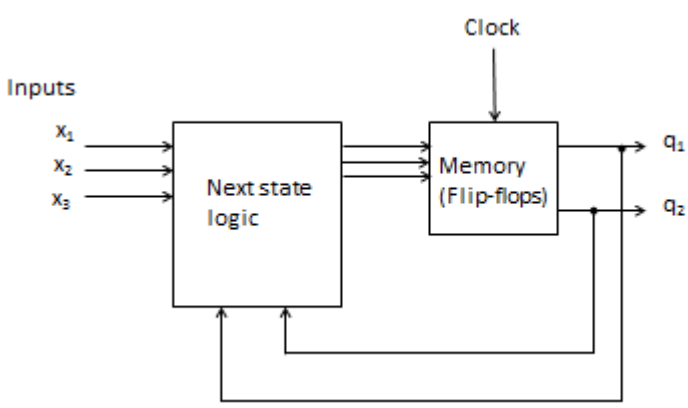
Quite often the output(s) of a sequential circuit will depend only on the state, and not on any of the current inputs. For example, a circuit with three inputs, two flip flops and two outputs, where the output depends only on the state, would be depicted as:



A system such as the one above, where the output depends only on the state (and not on the current input(s)) is termed a *Moore system*.

The beautiful thing is this: The next state logic and the output logic are entirely combinational logic!

Sometimes, the output of a system might just be the values of the states of the various flip flops, and no explicit output logic is needed. In such cases, the sequential circuit would be depicted as:



Analysis of Sequential Systems

As an engineer, you will be primarily concerned with the design of sequential systems. But, to become more comfortable with sequential systems, we will first analyze systems that were designed by others.

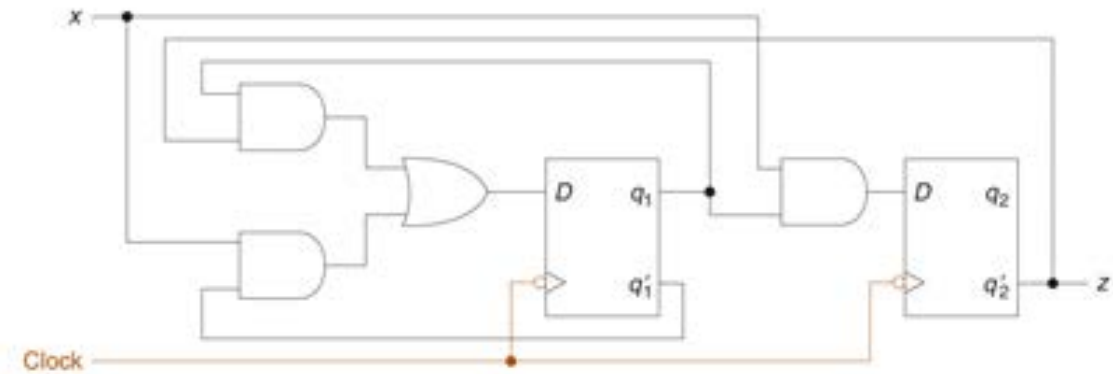
It will be helpful to remember the next state equations for the *D*, *JK* and *T* flip flops:

D flip flop:

JK flip flop

T flip flop:

Example Consider the circuit below, which was designed by your friend who was fired yesterday and left behind no notes on what this circuit does (but your boss thinks the circuit is important!).



From A. Marcovitz, *Introduction to Logic Design*, McGraw Hill, 2010

Question: Is this a Mealy circuit or a Moore circuit?

Question: How many states can this circuit take on?

Question: Write an expression for D_1, the input to the left flip flop.

Question: What is the value of q_1* ?

Question: Write an expression for D_2, the input to the right flip flop.

Question: What is the value of q_2* ?

Question: Write an expression for the output *z*.

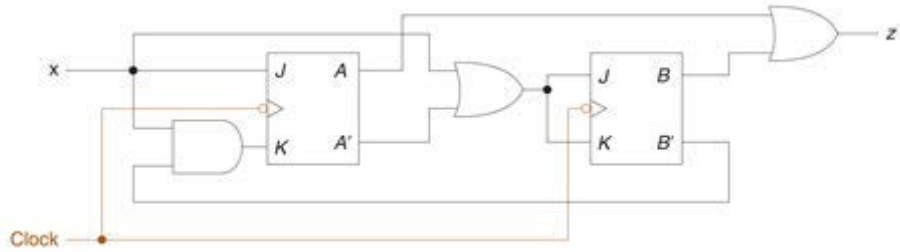
Question: Complete the state table using this format:

$q_1\ q_2$	$q_1^*q_2^*$ $x = 0$	$x = 1$	z
00			
01			
10			
11			

Question: Sketch the state diagram for this logic circuit.

Why do you think we draw the outputs within the circle representing the state for Moore circuits such as this?

Example. Consider the Moore circuit shown below.



From A. Marcovitz, *Introduction to Logic Design*, McGraw Hill, 2010

Question: Write input equations J_A, K_A, J_B, K_B.

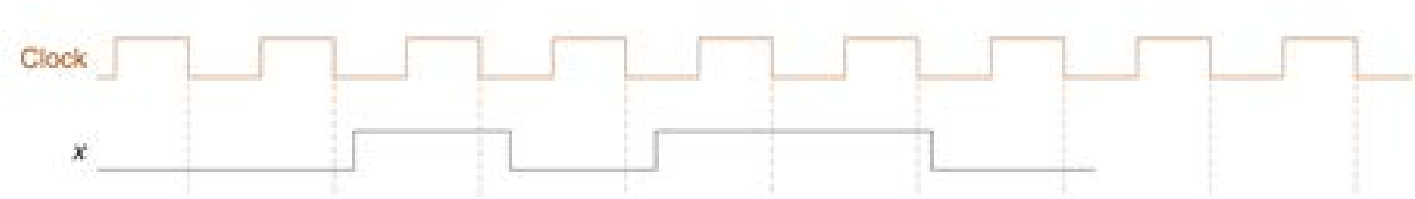
For the left flip flop:

For the right flip flop:

Remember what a J-K flip flop does:

Question: Construct the state table that has the heading below.

	A^*B^*		
AB	x = 0	$x = 1$	z

Question: Sketch the state diagram for this logic circuit

Question: Sketch a timing diagram showing *A*, *B* and *z* for the clock and input signals below. Assume $A = B = 0$ prior to the first clock training edge.