

The D flip-flops can be replaced by any of the other memory devices. In particular if any other gated flip-flop is used then the analysis techniques used to determine the functions governing state changes and outputs remains the same – only different characteristic functions are needed.

The textbook presents others: JK, SR, and T flip-flops.

- The textbook distinguishes between master-slave and edge-triggered flip-flops based on implementation detail.
- Both change state on the leading edge of the pulse.
- This is indicated in the logic diagrams by labeling the control input with $\triangleright C$.
- It is also possible to have a flip-flop trigger on the trailing edge.
- This is indicated in the logic diagrams by putting an inverter circle on the control input line labeled with $\triangleright C$.
- Other triggering schemes are possible.
- It is very important to check the flip-flop specifications in the logic diagram to determine the type of triggering.

In practice, D flip-flops are used most often because:

- They have the simplest implementations.
- They have the simplest characteristic function.

The first is important since this translates into performance and cost of the physical realization of the circuit.

The second means that the combinational logic that is used to compute the excitation variables D_i from the inputs and the current state variables y_i can in fact compute the next state value directly since the characteristic function is the identity – $Y_i^+ = D_i$. With the other flip-flops one needs to take into account the translation of the state variables into the terms of the excitation variables.

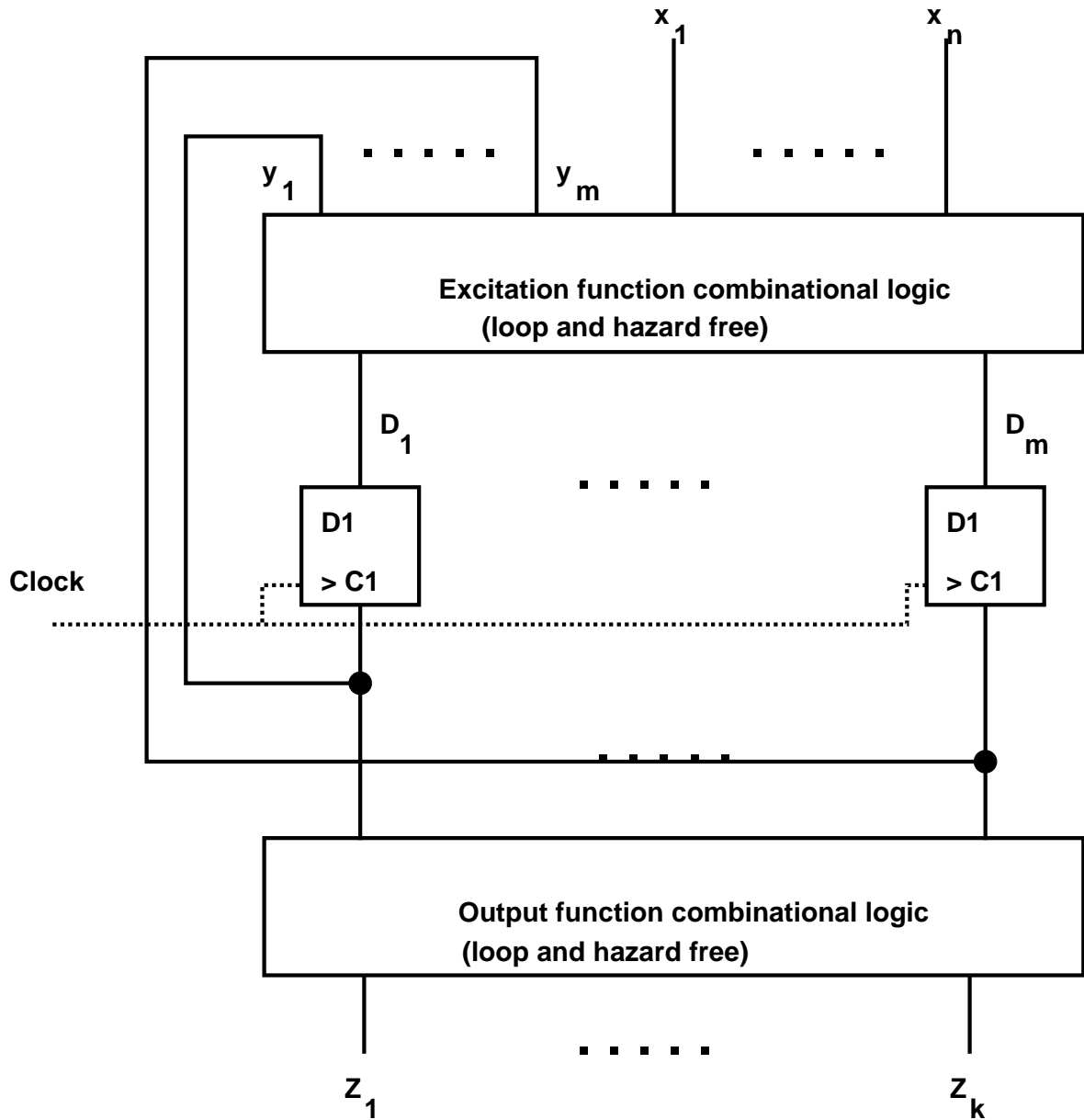
Suppose we build a sequential machine using D flip flops and an output that depends only on the state of the machine

How does it operate?

- Since the last pulse completed the output network has settled to a function based only on the current state (in a Moore machine)
- Right before the next pulse, the excitation variables are available on the input to the flip-flops (in the case of D flip-flops these are also the next state bits)
- On the next pulse the flip-flops evaluate the new state based on old state and settled excitation variables.
- New state variables feed back into excitation network but cannot affect the state since the flip flop is opaque by design.
- New output values are computed based on new state.
- At some point new input variables are seen and the excitation network settles with the new excitation variable values and the new input and is therefore ready for the next pulse.

No unstable states, we need only know what the next state is given a set of input variable values and the old state.

Finite state machine based on D flip-flops
 (Moore machine : Z is a function of y only)



D flip-flops used as memory elements
 any other element can be used.

The sequential machine shown is an example of a **finite state machine**. The hardware components of sequential circuits can make machines that have more complicated functioning and timing behavior than finite state machines.

It is possible however to define restrictions to the way in which the hardware operates that results in mathematically equivalent behavior to the formal automata theory definition of finite state machines.

Three basic properties of the finite state machine.(McCluskey)

Single-edge Property All internal state changes occur in response to an input variable changing from 0 to 1.

For our example this is clearly the case. The state changes only on the leading edge of the clock signal. All flip-flops have the clock input in the same form, i.e., there are no flip-flops that receive the complement of the clock.

External-Trigger Property An internal state change never causes a subsequent internal state change.

For our example this is guaranteed by the fact that the combinational logic of the excitation functions is loop free and no flip-flop output is connected directly to the control of another flip-flop, i.e., to the clock line.

This eliminates the cascade of unstable intermediate states we saw for fundamental mode circuits such as latches. It is why pulse mode is so much easier to design and analyze.

Single-Clock Property There is one distinguished input called the **clock**. The output changes occur synchronized with the clock pulse.

For our example, this is clear. The state changes in response to the clock and then the new state ripples through the output combinational logic to set the Z_1, \dots, Z_r output signals.

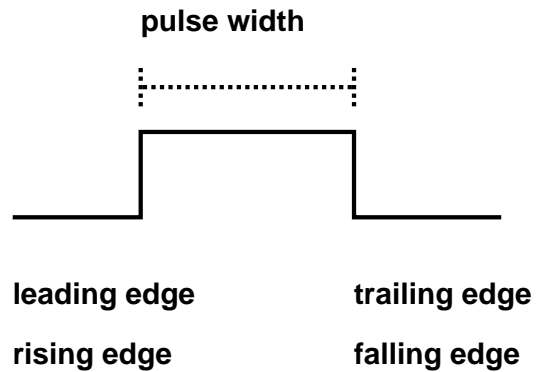
Note that Z_i is a function of **only** the state variables. Such a finite state machine is called a **Moore machine**. Since the state variables are levels rather than pulses, the output signals of a Moore machine are also levels rather than pulses.

The textbook has circuits which violates this property and yet which Gajski calls finite state machines. This is improper use of the terminology. This will be discussed in more detail later.

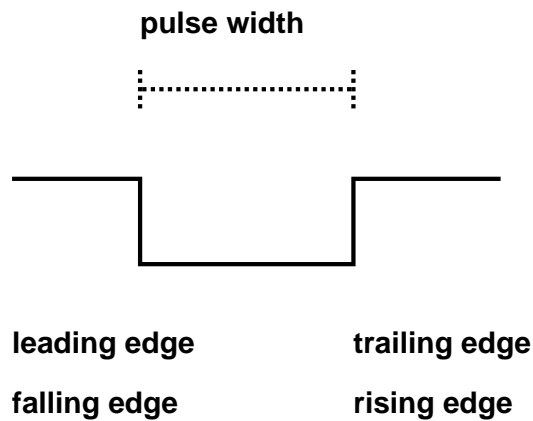
Pulse Mode Terminology

The circuit reacts only when a pulse is received on one of its input lines.

Positive Pulse : positive transition then negative transition



Negative Pulse : negative transition then positive transition



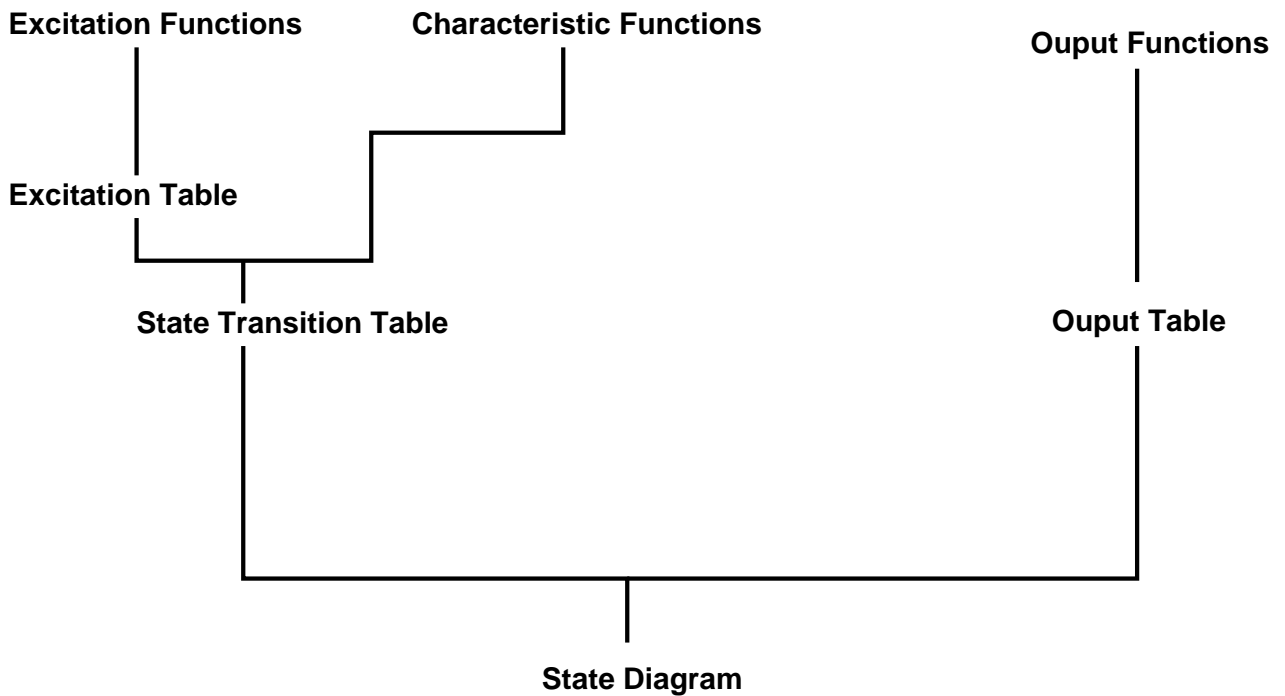
The pulse width has to be long enough for the devices ⁹ to detect the presences of a pulse. We will assume this is the case and that we always use positive pulses.

Analysis of FSM

Given a circuit diagram of a FSM, we must analyze its behavior just as we did for fundamental mode latches except things are much easier.

We proceed as follows:

FSM Analysis Procedure



- **Excitation functions** relate the input values and the current state to the new values that appear on the input lines of the flip-flops called the excitation variables, e.g., the S_i, R_i $i = 1, \dots, k$ lines for the k SR flip-flops.
- **Excitation table** is the union of the truth tables for the input lines to the flip-flops.
- Given the excitation variables, the action of the flip-flops to produce the next state can be determined by applying the characteristic functions of the flip-flops. The resulting values are the new values of the output of the flip-flops and encode the next state, e.g., the output of the i flip-flop q_i in the old state goes to Q_i in the new state after the clock pulse. The changes to all q_i as a function of the input and current state is given in the **state transition table**.
- **Output functions** relate the current state to the value on the output lines if it is a Moore machine. If the output depends also on the input then a careful consideration of which state, old or new, combines with the input to give the output must be done. This will be discussed later.
- **Output table** is the union of the truth tables for the output functions.

The excitation functions are

$$\begin{aligned}D_1 &= x \oplus y_2' \\D_2 &= x \oplus y_1\end{aligned}$$

The characteristic functions are those associated with a D flip-flop which we have seen before:

$$\begin{aligned}Y_1^+ &= D_1 \\Y_2^+ &= D_2\end{aligned}$$

The output functions depend only on the current state:

$$\begin{aligned}Z_1 &= y_1 \\Z_2 &= y_2\end{aligned}$$

The Excitation and Transition tables are identical since $Y_i^+ = D_i$.

$\langle ck \rangle \uparrow$		
y_1y_2	$x = 0$	$x = 1$
00	10	01
01	00	11
11	01	10
10	11	00

The output table is trivial:

y_1y_2	Z_1Z_2
00	00
01	01
11	11
10	10

This is a Moore machine.

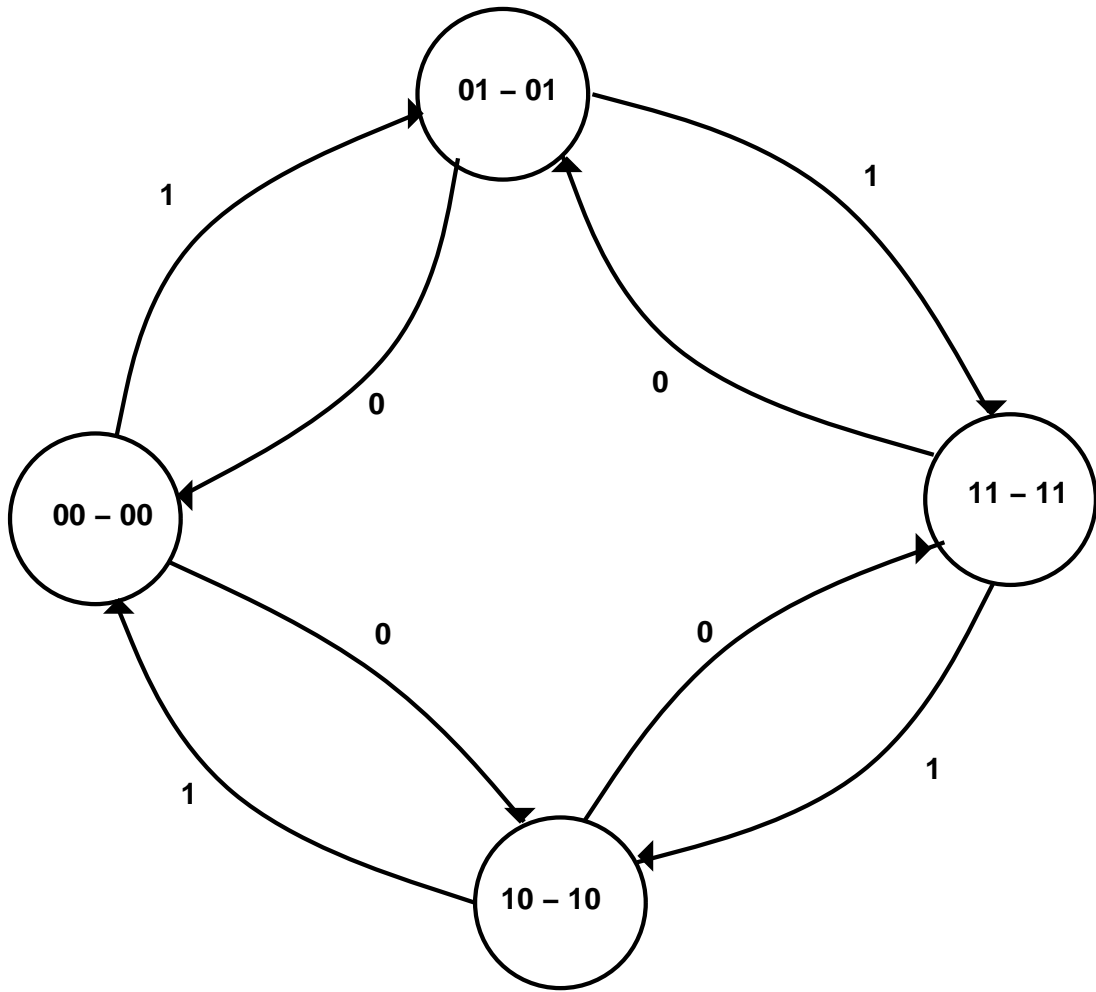
Note all states are stable so the table only lists the next state for each state/input combination. There are no intermediate unstable states as one might see when looking at the transition table of a fundamental mode circuit.

The $\langle ck \rangle \uparrow$ indicates that the machine operates in pulse mode using the leading edge of the clock.

$\langle ck' \rangle \uparrow$ or $\langle ck \rangle \downarrow$ would be used to indicate that the machine triggers on the trailing edge.

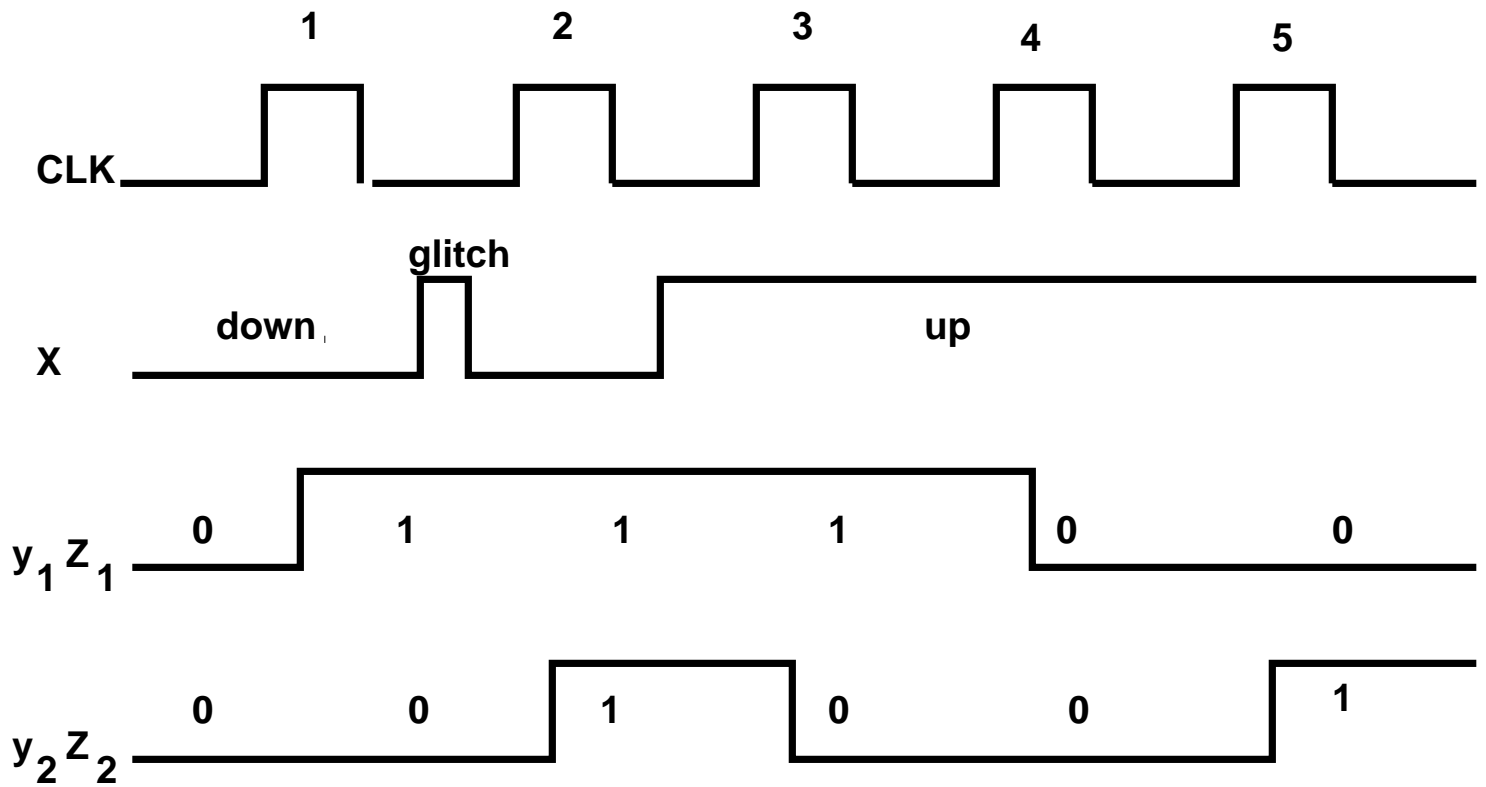
The operation is easily seen from a state diagram where each state is associated with a node. Inside the node is listed the state identifier which can be y_1y_2 or some unique letter or number, and the output pair Z_1Z_2 (this is the case for a Moore machine).

Transitions are indicated by an arc from the old to new state with the input value of x labelling the arc.



x = 1 count up in Gray code (move clockwise)

x = 0 count down in Gray code (move counterclockwise)



Note state and output do not respond to x signal glitching up to 1 during first clock cycle due to pulse mode operation

- only the input x and Z_1Z_2 are visible to the external world.
- from a mathematical point of view this example performs the mapping of an input sequence on x to an output sequence on Z_1Z_2 .
- $x = 0, 0, 1, 1, 1$ the glitch to 1 is not included since only the value of x at the clock pulse count.
- $Z_1Z_2 = 00, 10, 11, 10, 00, 01$.