

Digital electronics



**ASSIGN
BUSTER**

Use clock pulses in the inputs are called clocked-sequential circuits. They are stable and their timing can easily be broken down into independent discrete steps, each of which is considered separately. [pic] A clock signal is a periodic square wave that indefinitely switches from 0 to 1 and from 1 to 0 at fixed intervals. Clock cycle time or clock period: the time interval between two consecutive rising or falling edges of the clock. Clock Frequency = $1 / \text{clock cycle time}$ (measured in cycles per second or Hz) Example: Clock cycle time = 10ns clock frequency = 100MHz

Concept of Sequential Logic A sequential circuit as seen in the last page, is combinational logic with some feedback to maintain its current value, like a memory cell. To understand the basics let's consider the basic feedback logic circuit below, which is a simple NOT gate whose output is connected to its input. The effect is that output oscillates between HIGH and LOW (i. e. 1 and 0). Oscillation frequency depends on gate delay and wire delay. Assuming a wire delay of 0 and a gate delay of 10ns, then oscillation frequency would be (on time + off time = 20ns) 50MHz.

The basic idea of having the feedback is to store the value or hold the value, but in the above circuit, output keeps toggling. We can overcome this problem with the circuit below, which is basically cascading two inverters, so that the feedback is in-phase, thus avoids toggling. The equivalent circuit is the same as having a buffer with its output connected to its input. But there is a problem here too: each gate output value is stable, but what will it be? Or in other words buffer output can not be known. There is no way to tell.

If we could know or set the value we would have a simple 1-bit storage/memory element. The circuit below is the same as the inverters connected back to back with provision to set the state of each gate (NOR gate with both inputs shorted is like an inverter). I am not going to explain the operation, as it is clear from the truth table. S is called set and R is called Reset. IS 10 IQ 1 10 II There still seems to be some problem with the above configuration, we cannot control when the input should be sampled, in other words there is no enable signal to control when the input is sampled.

Normally input enable signals can be of two types. Level Sensitive or (LATCH) ? Edge Sensitive or (Flip-Flop) Level Sensitive: The circuit below is a modification of the above one to have level sensitive enable input. Enable, when LOW, masks the input S and R. When HIGH, presents S and R to the sequential logic input (the above circuit two NOR Gates). Thus Enable, when HIGH, transfers input S and R to the sequential cell transparently, so this kind of sequential circuits are called transparent Latch. The memory element we get is an RS Latch with active high Enable.

Edge Sensitive: The circuit below is a cascade of two level sensitive memory elements, with a phase shift in the enable input between first memory element and second memory element. The first RS latch (i. e. the first memory element) will be enabled when CLK input is HIGH and the second RS latch will be enabled when CLK is LOW. The net effect is input RS is moved to Q and Q' when CLK changes state from HIGH to LOW, this HIGH to LOW transition is called falling edge. So the Edge Sensitive element we get is called negative edge RS flip-flop.

Now that we know the sequential circuits basics, let's look at each of them in detail in accordance to what is taught in colleges. You are always welcome to suggest if this can be written better in any way. Latches and Flip-Flops There are two types types of sequential circuits. Asynchronous Circuits.

Synchronous Circuits. As seen in last section, Latches and Flip-flops are one and the same with a slight variation: Latches have level sensitive control signal input and Flip-flops have edge sensitive control signal input. Flip-flops and latches which use this control signals are called synchronous circuits.