Semiconductor Technology from A to Z

Digital technology

Sequential Logic: Flip-Flops, Latches, and Clock Trees

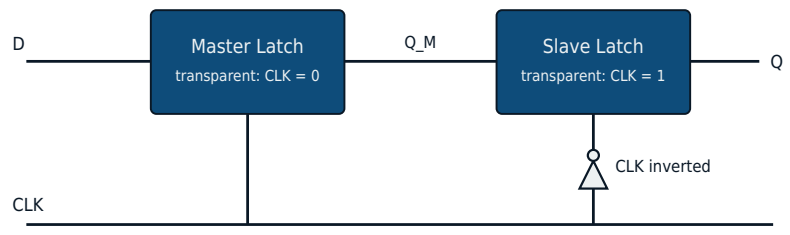
Latches and Flip-Flops

A latch is a level-sensitive storage element: as long as the clock or enable signal is active, the output directly follows the input (transparent state); once the clock signal switches to the inactive level, the last applied value is frozen. A simple D-latch can be built from two cross-coupled inverters whose feedback path is opened and closed through a transmission gate; in modern standard-cell libraries, such a latch typically requires 6 to 8 transistors.

A flip-flop, by contrast, is edge-triggered: the stored value changes only at the moment of the rising or falling clock edge. The most common CMOS flip-flop is the master-slave flip-flop, in which two latches clocked in opposite phase are connected in series – the master latch tracks the input while the clock is low, and the slave latch captures the value of the master latch on the rising edge and holds it for the remainder of the clock period.

Master-Slave Flip-Flop Built from Two D-Latches

Latches clocked in opposite phase prevent pass-through

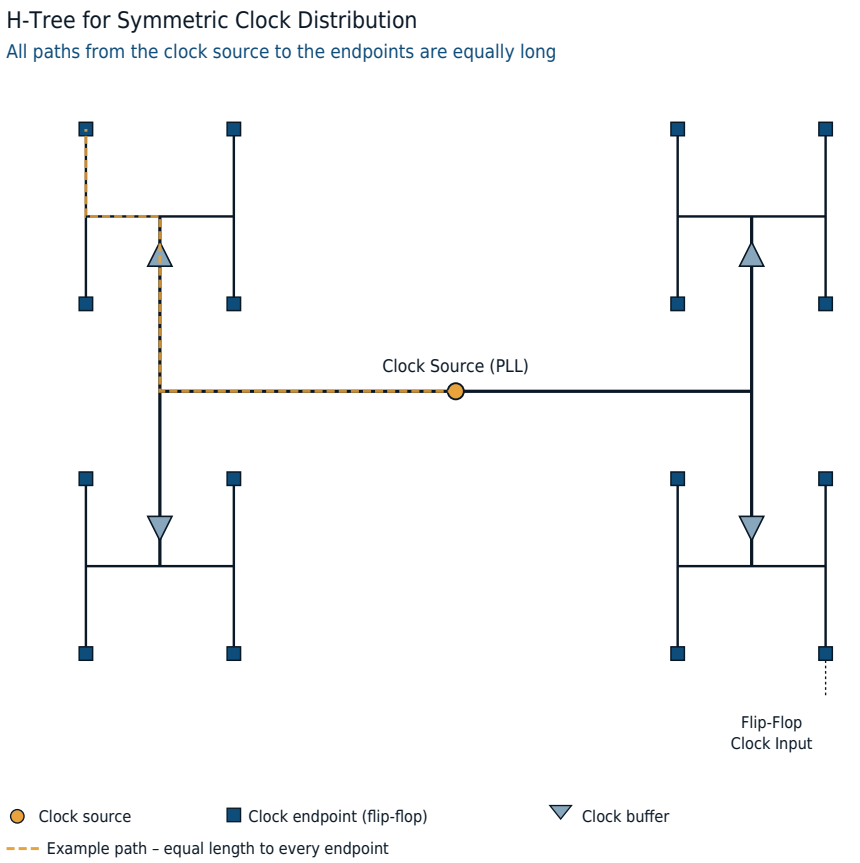


At CLK = 0 the master latch is transparent and the slave latch holds – at CLK = 1 it is reversed

Because of this cascading, at most one of the two latches is ever transparent at a given time, which prevents the input from passing directly to the output within a single clock period. Such a master-slave flip-flop typically consists of around 20 to 24 transistors; modern processor dies contain several hundred million to over a billion such flip-flops.

The Clock Tree

In a synchronous circuit, every flip-flop must receive a common clock signal that arrives at tens of thousands to millions of gates at nearly the same time. Since a single clock source cannot drive the parasitic capacitance of all connected flip-flops, the clock is distributed through a clock tree: a hierarchy of typically 5 to 8 buffer stages that amplify the signal in stages while fanning it out to an increasing number of endpoints.



The timing difference with which the clock arrives at different flip-flops is called clock skew and arises from unequal wire lengths and buffer chains; in well-optimized designs, skew is usually kept to within 2 to 5 % of the clock period. In a symmetric clock tree such as an H-tree, the wire length from the root to every endpoint is deliberately kept identical to minimize skew. Because clock wiring and its buffers alone can account for 20 to 30 % of a digital chip total dynamic power, optimizing the clock tree (clock tree synthesis) is one of the most demanding steps in digital backend design.

Asynchronous Control Inputs and Scan Flip-Flops

In addition to the clocked data input, most flip-flops have separate, asynchronously acting control inputs for setting or resetting the stored value. These act immediately, independent of the clock edge, and are used, for example, to force a defined initial state when the circuit is powered on.

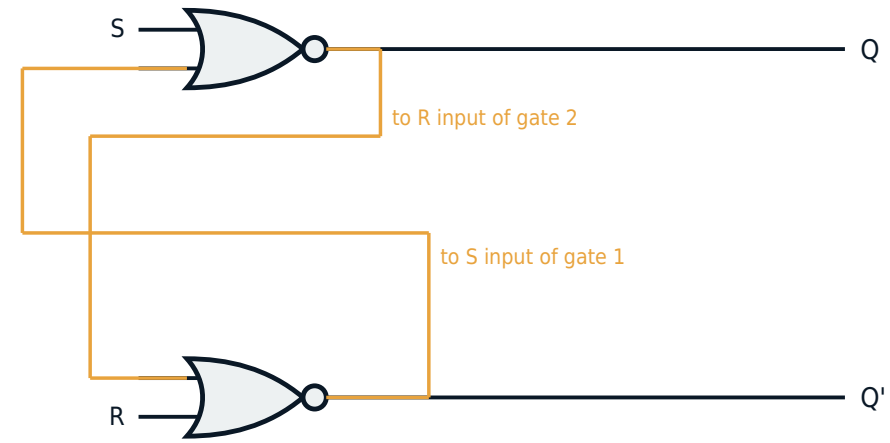
For manufacturing test, regular flip-flops are frequently replaced with scan flip-flops, which have an additional data input and a selection multiplexer and typically require only about 5 to 10 % more area than a regular flip-flop. In test mode, all scan flip-flops on a chip are connected into one or more long shift-register chains, often with several thousand elements per chain, through which arbitrary test patterns can be shifted in and the resulting states shifted back out. This approach, known as design for test (DFT), typically achieves fault coverage above 98 % in modern designs and makes logic states deep inside a chip observable and controllable from the outside without requiring additional test pins.

The SR Latch: The Simplest Bistable Element

The simplest bistable element can be built from two cross-coupled NOR or NAND gates and is called an SR latch (set-reset latch). In the NOR version, the circuit holds its state as long as both inputs S (set) and R (reset) are 0; a high level on S sets the output Q to 1, a high level on R resets it to 0, and the combination S=R=1 is defined as a forbidden state, since both outputs Q and the inverted Q̄ then fall to 0 simultaneously, leaving the subsequent state undefined once both inputs return to 0 together – which of the two gates "wins" depends on tiny propagation delay differences.

SR Latch Built from Two Cross-Coupled NOR Gates

Positive feedback stores the most recently set state



Feedback paths – each gate reinforces the other gate output

The feedback that gives the latch its name arises because the output of each gate is fed back to the input of the other: once a state is set, it holds itself, since both gates mutually "support" each other as long as S and R remain 0 – this same principle of positive feedback also underlies the latches and flip-flops discussed in the following sections. An equivalent NAND-based version, in which the inputs S and R are active-low, is equally common in practice, for example for debouncing mechanical switches, where a single button press should produce only one clean state change despite multiple contact bounces.

The SR latch is therefore the simplest element capable of storing a single bit of state without requiring a clock signal – it responds immediately (asynchronously) to changes on S and R, which also makes it susceptible to brief glitches. Adding a clock signal and a gating circuit that only lets S and R through during an active clock phase turns it into a clocked SR latch – the precursor to the edge-triggered flip-flops discussed in the following sections.

Flip-Flop Types Overview: SR, JK, T, and D

Building on the SR principle, four basic flip-flop types have become established in digital logic, differing in their input wiring and therefore in their behavior,

even though internally they are ultimately all built from the same bistable element. The SR flip-flop inherits the behavior of the SR latch, including the forbidden state, only edge-triggered rather than level-sensitive. The JK flip-flop closes this gap: the combination J=K=1 is not forbidden but instead toggles the output on every active clock edge (Q switches to its complement); for J=1, K=0 it sets like the S input, for J=0, K=1 it resets like the R input, and for J=K=0 it holds its state.

The T (toggle) flip-flop is a special case of the JK flip-flop with a single input T that toggles on every clock cycle when T=1 and holds its state when T=0 – a function especially needed in frequency dividers and asynchronous counter circuits, since a chain of T flip-flops with T=1 at each output successively halves the clock frequency. Historically, SR, JK, and T flip-flops were built directly from discrete gates, before a more unified approach took hold with the spread of dense integrated circuits.

The D flip-flop (data or delay flip-flop), by far the most widely used in modern CMOS logic, simply captures the value present at the D input on every active clock edge, without the ambiguity of the SR flip-flop or the switching logic of the JK flip-flop – its characteristic equation is simply Q(t+1) = D. Because SR, JK, and T behavior can, if needed, also be reproduced with additional combinational logic in front of a D flip-flop (for example T = D XOR Q for toggle behavior), modern standard-cell libraries provide essentially only the D flip-flop as a base cell, and the following sections accordingly focus on its structure and timing behavior.