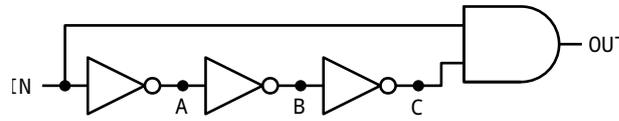


Worksheet: D Flip Flop

Gate Delay

Evaluate the output of the circuit to the right.



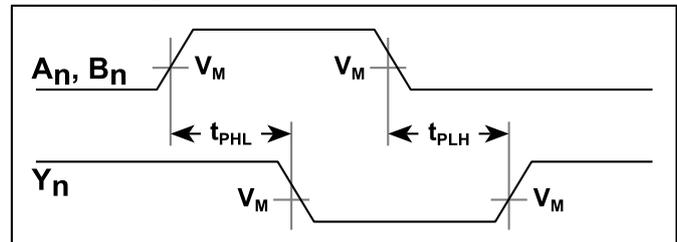
IN	A	B	C	OUT
0				
1				

The truth table shows a steady-state evaluation – it looks at the state of the circuit after it has settled down to a final value. However, with circuits, as the signal flows through each gate, there is a delay and it takes time for the steady-state to be achieved.

The table below left and diagram below right describes the propagation delay through an example NAND gate. When either input (A or B) changes and this results in a change of the output from high to low, there is a propagation delay, t_{PHL} , of somewhere between 1 ns and 3.9 ns. When a change in input results in a change of the output from low to high, there is a propagation delay, t_{PLH} , of between 1.0 ns and 4.1 ns.

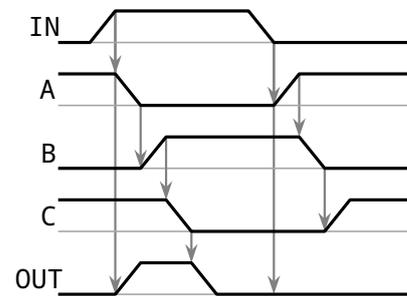
	MIN	TYP	MAX
t_{PLH}	1.0	2.7	4.1
t_{PHL}	1.0	2.7	3.9

AC WAVEFORMS
 $V_M = 1.5V, V_{IN} = GND \text{ to } 2.7V$



The *timing diagram* to the right examines how the circuit at the top of the page changes over time.

- The signal on IN starts with a value of low.
- The signal IN is set high.
- After a propagation delay through the first inverter, signal A, the input to the second inverter, has completed the transition to a low value.
- After another propagation delay through the second inverter, signal B, the input to the third inverter, is high.
- After a third propagation delay through the their inverter, signal C, the input to the NAND gate, is low.

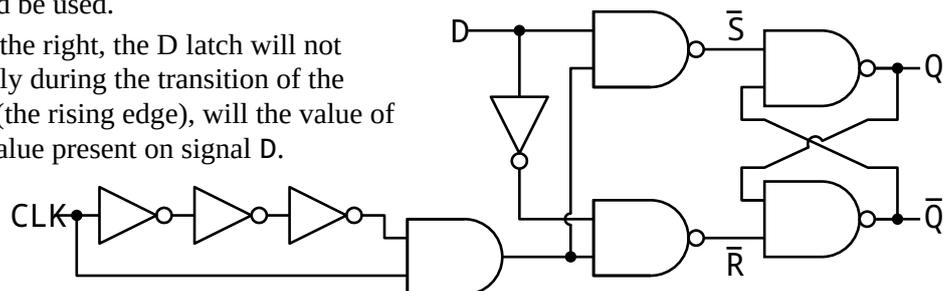


To start, the signal IN at the AND gate is low and the signal C at the AND gate is high, leading to the output being low. When the IN signal first changes to high, signal C at the NAND gate does not change. As both IN and C signals are high, the output of the AND gate changes to high as well. Some time later (the time of two more propagation delays), the signal C changes to low, resulting in the output signal returning low.

This pulse is called a **glitch**. Glitches may occur as signals propagate differently through different parts of a logic circuit. In certain situations, this may cause unwanted side effects in circuits. However, there are situations where a glitch could be used.

For example, in the circuit to the right, the D latch will not follow the input signal, D. Only during the transition of the CLK signal from low to high (the rising edge), will the value of the output, Q, change to the value present on signal D.

This circuit is one implementation of what is called a D flip flop.



Worksheet: D Flip Flop

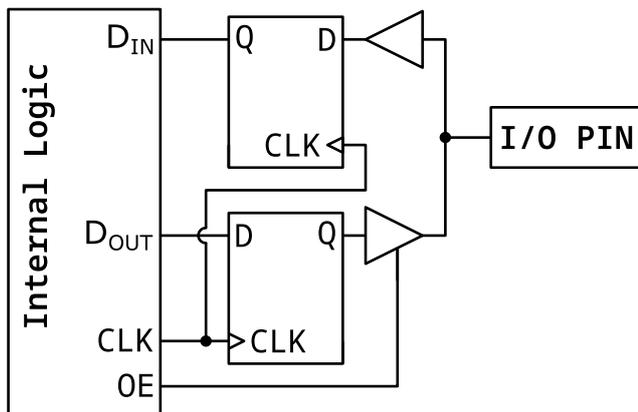
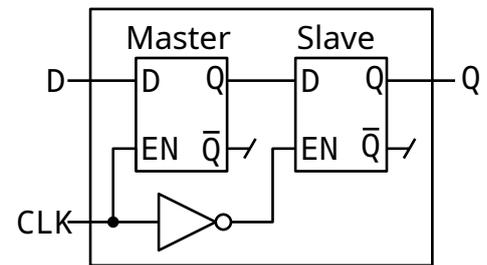
D Flip Flop

The D latch is able to copy the value of an input data line while the enable line is set active, and store the value while the enable line is set inactive. The D latch is a “level-triggered” device: the output follows the input as long as the level of the enable signal is in the active state.

The next improvement on this circuit is to make the device edge triggered. This means, the output will not follow the input, but only change when a clock input changes from low to high. This is called the *rising edge* of the clock signal. The edge-triggered version of a D latch is called a D flip flop.

Although the glitch implementation of a D flip flop is possible, it is very robust, as the delay through a gate will likely change from device to device, and especially with any change in the manufacturing process.

The more common implementation of an edge-triggered D flip flop is called a master-slave implementation, which uses two D latches. The circuit is shown diagrammatically to the right. When the CLK signal is set low, the master D latch output follows the input signal, D. When the clock transitions from low to high, the output of the master D latch stops changing, and the slave D latch follows the value saved by the master D latch. However, as stated, while the clock is high, the master D latch does not change. So, effectively, the circuit will always output the value that was present on the D input at the time when the clock signal transitioned from low to high.



One common place to use D flip flops is on input/output (I/O) pins of chips, including I/O pins on microprocessors and memory.

The figure to the left is a diagram of the circuit that interfaces between the internal logic of a chip and the I/O pin.

It becomes easier to correctly design and implement logic when signals are kept steady and only change synchronously with a clock.

Another use of D flip flops is as **registers** used to store computer data. In modern processors, registers are typically 32 bits or 64 bits. The figure below shows a circuit diagram of a 4-bit register. In the diagram, the register can read and store the data that is presented on the data bus if the EN signal is set high on the rising edge of the clock signal (when CLK transitions from low to high). The register can also drive the data onto the data bus by setting the OE signal high.

