We can use a 2-to-1 multiplexer to build a settable storage element.

Recall that a MUX selects as its output value the value of one of its two data inputs.

The output of the MUX serves as the state output of the memory component.

Internally to the memory component we'll also connect the output of the MUX to its D0 data input.

The MUX's D1 data input will become the data input of the memory component.

And the select line of the MUX will become the memory component's load signal, here called the gate.

When the gate input is LOW, the MUX's output is looped back through MUX through the D0 data input, forming the bi-stable positive feedback loop discussed in the last section.

Note our circuit now has a cycle, so it no longer qualifies as a combinational circuit.

When the gate input is HIGH, the MUX's output is determined by the value of the D1 input, i.e., the data input of the memory component.

To load new data into the memory component, we set the gate input HIGH for long enough for the Q output to become valid and stable.

Looking at the truth table, we see that when G is 1, the Q output follows the D input.

While the G input is HIGH, any changes in the D input will be reflected as changes in the Q output, the timing being determined by the tPD of the MUX.

Then we can set the gate input LOW to switch the memory component into memory mode, where the stable Q value is maintained indefinitely by the positive feedback loop as shown in the first two rows of the truth table.

Our memory device is a called a D latch, or just a latch for short, with the schematic symbol shown here.

When the latch's gate is HIGH, the latch is open and information flows from the D input to the Q output.

When the latch's gate is LOW, the latch is closed and in "memory mode", remembering whatever value was on the D input when the gate transitioned from HIGH to LOW.

This is shown in the timing diagrams on the right.

The waveforms show when a signal is stable, i.e., a constant signal that's either LOW or HIGH, and when a signal is changing, shown as one or more transitions between LOW and HIGH.

When G is HIGH, we can see Q changing to a new stable output value no later than tPD after D reaches a new stable value.

Our theory is that after G transitions to a LOW value, Q will stay stable at whatever value was on D when G made the HIGH to LOW transition.

But, we know that in general, we can't assume anything about the output of a combinational device until tPD after the input transition.

The device is allowed to do whatever it wants in the interval between tCD and tPD after the input transition.

But how will our memory work if the 1-to-0 transition on G causes the Q output to become invalid for a brief interval?

After all it's the value on the Q output we're trying to remember!

We're going to have ensure that a 1-to-0 transition on G doesn't affect the Q output.

That's why we specified a lenient MUX for our memory component.

The truth table for a lenient MUX is shown here.

The output of a lenient MUX remains valid and stable even after an input transition under any of the following three conditions.

(1) When we're loading the latch by setting G HIGH, once the D input has been valid and stable for tPD, we are guaranteed that the Q output will be stable and valid with the same value as the D input, independently of Q's initial value.

Or (2) If both Q and D are valid and stable for tPD, the Q output will be unaffected by subsequent transitions on the G input.

This is the situation that will allow us to have a 1-to-0 transition on G without contaminating the Q output.

Or, finally, (3) if G is LOW and Q has been stable for at least tPD, the output will be unaffected by subsequent transitions on the D input.

Does lenience guarantee a working latch?

Well, only if we're careful about ensuring that signals are stable at the right times so we can leverage the lenient behavior of the MUX.

Here are the steps we need to follow in order to ensure the latch will work as we want.

First, while the G input is HIGH, set the D input to the value we wish store in the latch.

Then, after tPD, we're guaranteed that value will be stable and valid on the Q output.

This is condition (1) from the previous slide.

Now we wait another tPD so that the information about the new value on the Q' input propagates through the internal circuitry of the latch.

Now, both D *and* Q' have been stable for at least tPD, giving us condition (2) from the previous slide.

So if D is stable for 2*tPD, transitions on G will not affect the Q output.

This requirement on D is called the setup time of the latch: it's how long D must be stable and valid before the HIGH-to-LOW transition of G.

Now we can set G to LOW, still holding D stable and valid.

After another tPD to allow the new G value to propagate through the internal circuitry of the latch, we've satisfied condition (3) from the previous slide, and the Q output will be unaffected by subsequent transitions on D.

This further requirement on D's stability is called the hold time of the latch: it's how long after the transition on G that D must stay stable and valid.

Together the setup and hold time requirements are called the dynamic discipline, which must be followed if the latch is to operate correctly.

In summary, the dynamic discipline requires that the D input be stable and valid both before and after a transition on G.

If our circuit is designed to obey the dynamic discipline, we can guarantee that this memory component will reliably store the information on D when the gate makes a HIGH-to-LOW transition.