Lab 03 – S’R’ Latch, a Programmable 1-Bit Memory Circuit

Students should work in pairs or, if there is an odd number enrolled in the lab session, in a triple.

Due: By the end of your lab session today.

Learning objectives: Gain experience with, and an understanding of, a programmable memory circuit.

Required equipment: Lab kit hardware and the Lab 03 Experiments Worksheet in Gradescope.

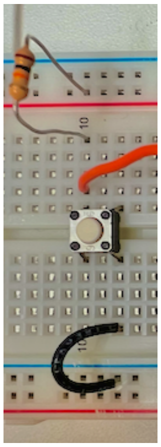
Bill of materials: Power-ready breadboard, 2x pushbutton switches, 2x 10 Kohm resistors, 1x 7400 DIP, 1x Green LED, 1x Red LED, 6 colors of wire, and wire stripper.

Step 1 – Build the lab circuit

Ensure that your breadboard is wired with the 10 Ohm limiting resistor and wires for enabling the red power and blue ground buses along the breadboard edges.

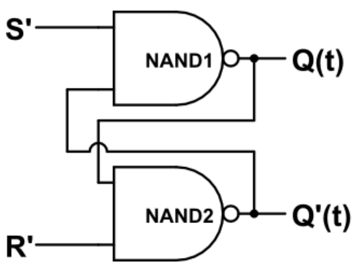
Parts Layout on a Breadboard. A good strategy is to place switches and DIP devices on your board first, before adding wires, resistors, and LEDs. Place switches where you can easily reach them. If there will be several, space the switches to give room for your fingers to reach each switch simultaneously. This will allow you easily press the switches in any combination that may be needed. DIP packages are larger, which is a good reason to place them on the breadboard next so that you do not run out of space for them.

The circuit for this lab requires two pushbutton switches and one 7400 DIP package. Place these three components on the breadboard now. Leave enough space between the two switches so that you will be able to push them independently.



Now, add wires and 10KOhm resistors to the board to build two active-low pushbutton logic signal sources, per this photo. The orange wire in the photo is the active-low logic output from this circuit which you will use as an input to the 7400 package. The left pushbutton circuit will produce the S’ logic signal, the right will produce R’.

The circuit diagram for the S’R’ latch (spoken as “S not R not Latch”) is shown below. When the label for an input or output of a logic circuit includes a not symbol, this means that input or output is true, or asserted, when the voltage on that wire is zero volts.



Q(t), the logic value of the Q output at time = t, is called the Current State of the latch.
Q(t+1) denotes the latch state at the next meaningful moment in time.

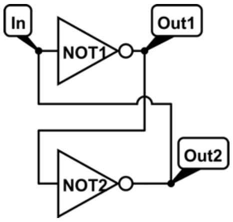
Add wire to power your 7400 chip and to build an S'R' latch circuit. Then connect a green LED to the Q(t) latch output, with the long leg of the LED to Q(t) and the short leg to the Ground bus. Connect the long leg of a red LED to the Q'(t) output and its short leg to Ground.

Finally, get a sheet of blank printer paper from the lab room printer to place centered underneath your breadboard. Then, write on the paper extending beyond the edges of your breadboard to clearly label which pushbutton in the S' input, which is the R' input, and which LED is the Q(t) output, and which the Q'(t) output.

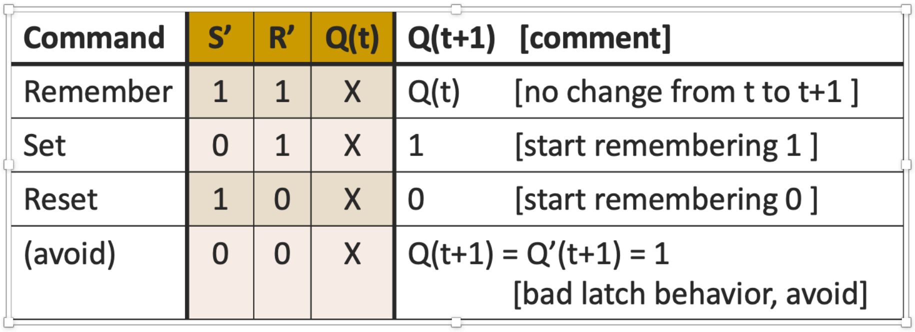
When your circuit is fully constructed, perform the experiments of the Lab 03 Experiments Notebook and record the data you gather.

S'R' Latch Operation

The S'R' Latch is an upgrade from the latch circuit that connects two NOT gates in a loop.

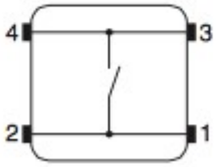


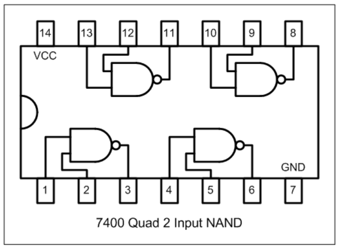
In the latch "upgrade" each NOT gate is implemented by a 2-input NAND that provides a user-controllable input. Now, Out1, of the two NOT gates circuit, is named Q(t), and Out2 is Q'(t). When inputs S'R' = 11, then both NAND gates emulate NOT and the latch stably remembers whatever bit value is present at Q(t) and, conveniently, the inverted value of the bit being remembered is available from output Q'(t). When S' = 0 and R' = 1, then NAND1 output, Q(t+1) is forced to 1 and after a gate delay, Q'(t+1) = 0, restoring the relationship intended for these two latch outputs. When S' = 1 but R' = 0, then Q'(t+1) is forced to 1, and Q(t+1) = 0 a gate delay later, there by re-setting, or clearing, the latch. If every S'R' = 00, then both Q(t+1) and Q'(t+1) are forced to 1, violating their intended relationship. The latch should never be given this input. Here is the Characteristic Table that describes the operation of the S'R' Latch given the inputs, S' and R', and the current state, Q(t), as the next latch output, Q(t+1):



Command	S'	R'	Q(t)	Q(t+1)	[comment]
Remember	1	1	X	Q(t)	[no change from t to t+1]
Set	0	1	X	1	[start remembering 1]
Reset	1	0	X	0	[start remembering 0]
(avoid)	0	0	X	Q(t+1) = Q'(t+1) = 1	[bad latch behavior, avoid]

Appendix: Reference Information for Circuit Building on Your Breadboard





7400 Quad 2 Input NAND

At any point if you want to test your construction, you can use an LED as a visual probe. Think of it as a hardware version of a PRINT statement in a programming language. Just connect the LED anode (longer lead) to the circuit node you wish to interrogate (variable to print out) and connect the other LED lead to breadboard ground. The LED will light when the circuit node is at logic 1 (high voltage) and not light if the circuit node is at low voltage or is not connected to anything (called an open circuit).