

Lab 3: Flip-flops

A flip-flop (FF) is a simple digital circuit element which represents one bit of electronic *memory*. A FF can be put into one of two possible states (“writing” into memory) and kept in that state until retrieved at a later time (“read” from memory). In this lab we will investigate properties of flip-flops, home-built from standard gates, as well as pre-packaged IC D-type and J-K type flip-flops. We will also build a standard application circuit.

3.1 Flip-flops from Standard Gates

Refer to the relevant data sheets from the Lab 1 handout or from the in-lab references to get the pin assignments for the 7400 and 7404 ICs.

3.1.1

Build a flip-flop from two inverters, as shown in Fig. 1. Read the output state with an LED. See if you can drive the FF into the opposite state by forcing a LOW point HIGH or a HIGH point LOW using a shorting wire to +5 V or to ground (shown as switches in the figure).

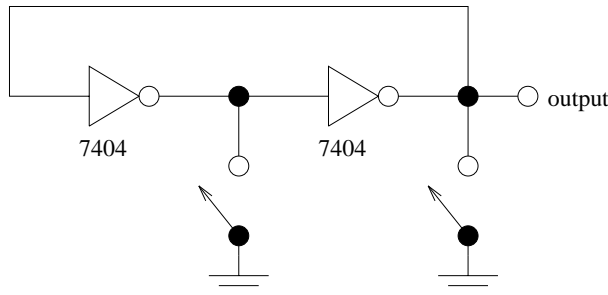


Figure 1: Flip-flop based on two inverters from a 7404 IC.

Now put a $\sim 500 \Omega$ resistor in series with the shorting wire and repeat. Can you now reliably change the FF state? (Unlike inverters, other types of gate inputs do not interact, thus avoiding this kind of problem, and are used in preference to inverters.)

3.1.2

Build a flip-flop from two 7400 IC NAND gates, as shown in Fig. 2. Determine the truth table for this FF. What happens when both inputs are simultaneously grounded? Determining which of the two inputs gives rise to the result in this case depends upon circuit details which are not readily controlled. Hence, this input state is not used — it is disallowed.

3.1.3

Based on the previous circuit, construct a debounced switch, as shown in Fig. 3. Use a wire to ground as the switch shown in the figure. Try bouncing your wire “switch” connection. Does the debouncer work?

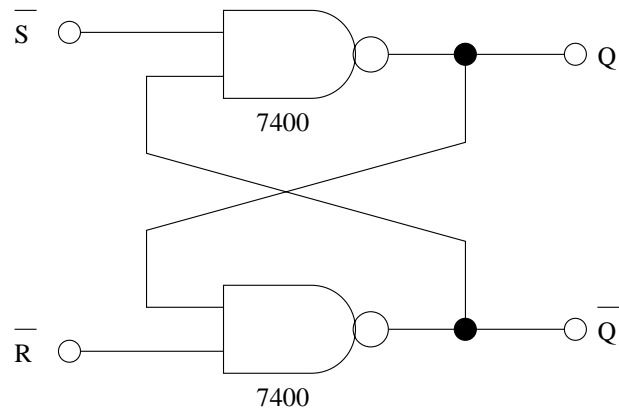


Figure 2: Flip-flop based on two NAND gates from a 7400 IC.

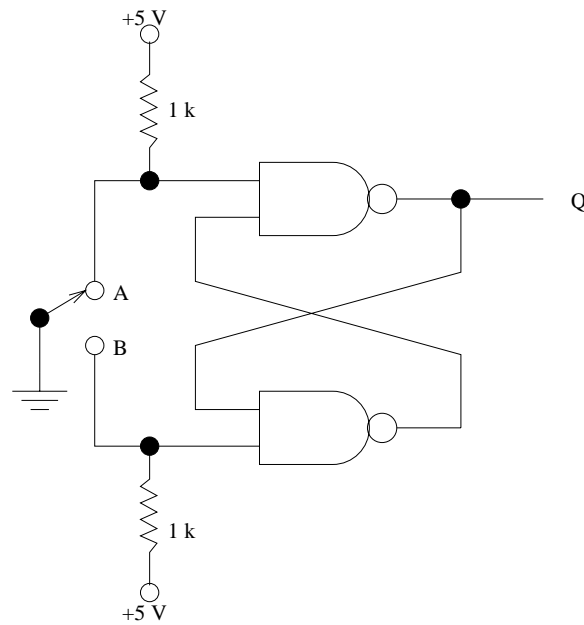


Figure 3: A debounced switch built from the previous circuit. The pull-up resistors can fall in the range 500Ω to $2 \text{ k}\Omega$.

3.2 Latch Comparisons

We wish to compare three standard types of IC-packaged flip-flops using the test circuit depicted in Fig. 4:

1. 7474 D-type (positive edge-triggered)
2. 7475 latch (level sensing)
3. 7476 J-K type (positive edge-triggered)

Refer to the attached data sheets for pin assignments. Note that the $\overline{\text{CLR}}$ and $\overline{\text{PRE}}$ inputs should be connected to +5 V when not in use. A 7404 inverter channel is needed for the 7476 K input. Also note that if you happen to use a 74LS76A, this is a *negative* edge-triggered version of the 7476.

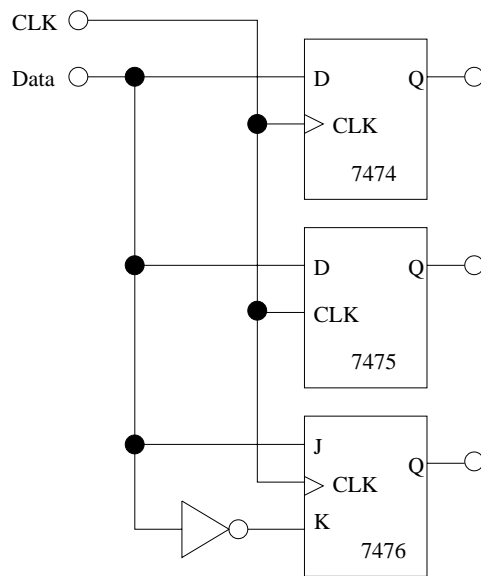


Figure 4: A test circuit for comparing different flip-flop types. Note that the 7476 $\overline{\text{CLR}}$ and $\overline{\text{PRE}}$ inputs should be connected to +5 V when not in use.

3.2.1

Use the LEDs to determine the Q outputs of each FF. Use separate debounced switches for the data and clock inputs. After each test, initialize by cycling the clock until all LEDs are off. Setup the following input states and compare the responses of the three devices:

1. Clock (CLK) HIGH or LOW; no D action.
2. D HIGH or LOW; no clock action.
3. Hold CLK HIGH; then make D HIGH, then LOW; release CLK.
4. Hold CLK HIGH; hold D HIGH; release CLK; release D.
5. Hold D HIGH; push CLK HIGH, then release; release D.

3.2.2

Explore the SET and CLEAR inputs of the 7474 and 7476, also known as “jam set” and “jam clear”. (Note that terminologies differ, so that “set” is sometimes called “preset”, and “clear” is often called “reset”.) You can do this by simply using a wire connected to ground. Do the “jam” functions take precedence over the D and CLK inputs ?

3.2.3

For the 7476 J-K FF, remove the inverter and determine the truth table when J and K are both connected to LOW or HIGH. In particular, note that when both J and K are HIGH, the FF toggles between HIGH and LOW for every clock cycle. For any input combination, when in the clock cycle do changes occur?

3.3 Latching a Number on Command

One of the applications illustrated by this experiment is that of “glitch” detection, which is a common use of flip-flops. Glitches are fast pulses in a circuit which arise through a problem in circuit operation or timing. Because TTL is fast, it can respond to a glitch, causing unexpected results which are difficult to debug. A glitch detector can hold and determine the time of a glitch. Of course, such a circuit can also be used to record intentionally fast, irregular signals.

Construct the circuit shown below in Fig. 5. How does this circuit work? Here are some of its key features:

- Pulse 1 drives the 7474 FF clock. Since its D input is kept HIGH, the 7474 Q will go HIGH whenever it detects a clock pulse.
- Pulse 2 provides the 7474 with a SET (=PRESET) input. Since this is actually $\overline{\text{PRESET}}$ on the 7474, pulse 2 needs to be normally HIGH, but goes LOW to perform the SET function. (Use the “NO” (normally open) debounced switch with a pull-up resistor to +5 V for this.)
- The four 7475 latches are enabled and follow the input when pins 13 and 4 are HIGH. The outputs latch when the clock inputs go LOW. These clock inputs are driven by the 7474 Q output, which is HIGH whenever either its clock or preset inputs is HIGH.
- The RESET (=CLEAR) input of the 7474 is driven by the selected state of a 7442 decoder, which is connected to a 7490 to make a sequencer, as in Lab 2. Since the CLEAR is activated by a LOW, the FF will clear when the selected state (0) is reached.
- The 7490 is reset (set to zero) when the \overline{Q} of the 7474 goes HIGH.

Given these hints, circuit operation should become clear by performing the following experiments:

- When power is turned on, what do you observe?

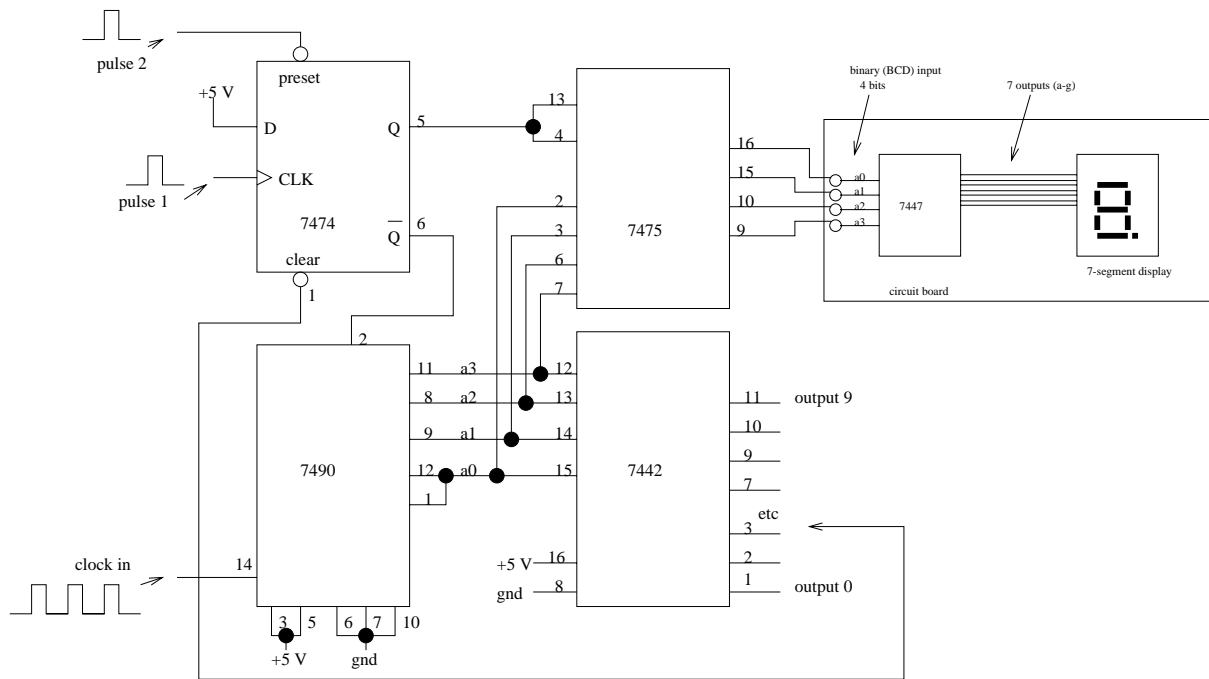


Figure 5: A circuit to latch a number.

- Press and release pulse 1. The counter should reset and cycle until the number selected by the 7442 (which drives the 7474 CLEAR) is reached, whereupon it will latch that number.
- Press and hold pulse 2. This inhibits latching, and the display will be observed to cycle through a number of states determined by the selected decoder output.

Repeat and/or test with LED readouts the various portions of the circuit until the answers to the following become clear:

1. Why does pulse 1 cause a cycle which stops at the selected number?
2. Why does pulse 2 inhibit the latching but still allow the counter to reset?
3. What happens to the 7490 counter after latching occurs?
4. What happens to the latch when the counter is reset?