Lab 11: Flip-Flops

Purpose: To acquaint the student with flip-flops and additional members of the 7400 family.

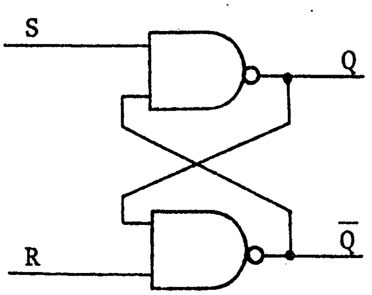
Concepts: This experiment covers R-S, D, and J-K flip-flops, latches and one-shots, edge triggering, master-slave flip-flops, data and shift registers, contact bounce, and clocked circuits.

References: J. L. Hughes (CLW) or Horowitz (Sect. 8.16-8.24) or Simpson (Sect. 13.1-13.2)

Introduction: Digital circuits are analyzed in terms of And, Or, Negation, and Memory operations. Experiment 6 dealt with the first three and showed how logical expressions can be represented by gating networks. This fact suggests that electronic logic should be useful in decision-making processes, especially where numerous variables have intricate interdependencies.

Logic circuits based solely on gates and parallel logic can be constructed, but there usually comes a point when a given task can be performed more efficiently if done in several steps, with intermediate results being stored in memory. This experiment deals with memory elements called flip-flops, binaries, bistables, or latches; whatever their name, they always exhibit only two stable states and a pair of complementary outputs Q and $\bar{Q}$. The first flip-flop we consider, the "R-S", is simple but imperfect. An improvement, the *clocked* R-S flip-flop, is better but it too has faults. The "type-D" flip-flop or "latch" represents an improvement although it lacks flexibility. The fourth FF that we consider, the "J-K", is more complicated but very useful.

Measurements. Part (A): Consider the R-S flip-flop consisting of the pair of *cross-coupled Nand gates* shown in Fig. 1. Although this flip-flop is simple, it has limited usefulness for two reasons. One is revealed in the first line of the truth table in Fig. 1: when S = R = 0, both the Q and $\bar{Q}$ outputs are Hi in contradiction to the requirement of flip-flops that their Q and $\bar{Q}$ outputs always be complementary. The other drawback concerns the *initializing* of this flip-flop. Construct the simple R-S flip-flop of Fig. 1 using a pair of Nands in a 7400 chip. Connect LOGIC SWITCHES to the S and R inputs and a MONITOR to each output. Verify the truth table of Fig. 1 by showing that (1) this particular R-S flip-flop is truly "flip-flop-like" (Q and $\bar{Q}$ are opposite) only if the combination S = 0 and R = 0 is avoided; (2) this R-S flip-flop can be initialized in its "set" state (Q = 1 and $\bar{Q}$ = 0) by providing inputs S = 0 and R = 1, after which S can be raised to a 1; and (3) this R-S flip-flop can be initialized in its "reset" state (Q = 0 and $\bar{Q}$ = 1) by providing S = 1 and R = 0, after which R can be raised to a 1. From these tests, we recognize the awkwardness of initializing this particular flip-flop.



Inputs		Outputs		Comment
S	R	Q	$\bar{Q}$	
0	0	1	1	Flip-flop?
0	1	1	0	OK
1	0	0	1	OK
1	1	0 1	1 0	Hysteresis

Fig. 1. Logic circuit and truth table for simple R-S flip-flop based upon Nand gates.

Next we discuss the response of this R-S flip-flop to a *pulsed* input. For pulsed operation, the inputs to this particular R-S flip-flop are normally Hi, and the device is "toggled" by applying *brief pulses* to Lo on the R or S inputs. For example, consider a pulse that takes the S input from its normal Hi to a temporary Lo (and back again) while leaving the R at its quiescent Hi. Judging from the circuit of Fig. 1, we conclude that this FF will "set" regardless of its original state. On the other hand, a pulse to Lo on the R input will "reset" the FF regardless of its initial state.

Verify this operation by connecting the "1" outputs of the PULSERS to the R and S inputs and MONITORS to the FF outputs. Fill in Table I, being careful to initialize the FF using procedures (2) and (3) above. To complete the last two lines of the table, connect a single PULSER *simultaneously* to both of the R and S inputs. Repeat the test several times checking for reproducibility. This circuit's indeterminacy for simultaneous input pulses is a major shortcoming.

Initial Conditions*	Lo-going Pulse on Input ...	Final State
set	S	
set	R	
reset	S	
reset	R	
set	R & S	
reset	R & S	

Table I. Truth table for simple R-S flip-flop operated under pulse mode. The * indicates the following: the FF is always initialized in its set or reset state, but then both inputs are raised to the normally quiescent Hi state before the trigger pulse to a momentary Lo comes along.

A common application of this FF is contact-bounce elimination. An ordinary switch cannot be used as a pulse generator for logic because of contact bounce — the random burst of level transitions that occur each time the switch is thrown. The output of a switch can be cleaned up, however, with an R-S FF. Figure 2 shows the circuit and facsimiles of the input voltages applied to the R and S inputs when the switch is thrown from the "0" position to the "1" position.

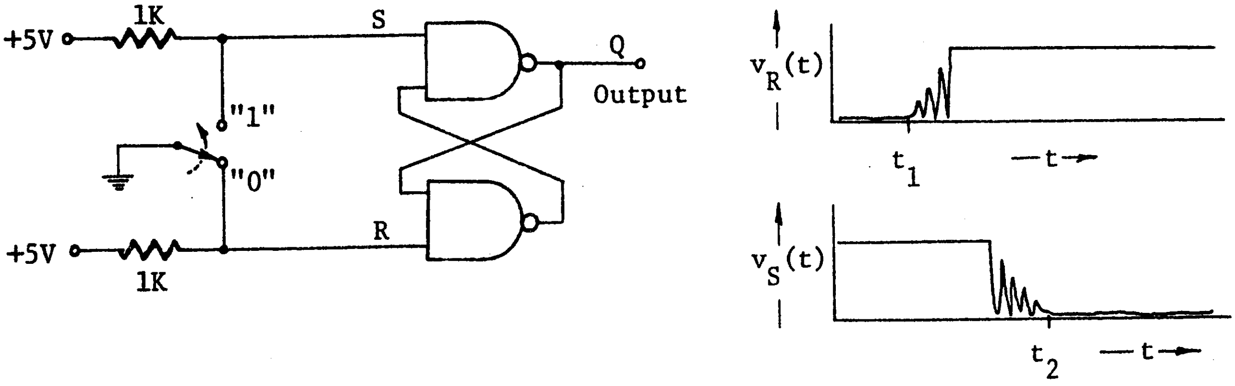


Fig. 2. Contact bounce eliminator based upon R-S FF. The curves depict the input voltages $v_R(t)$ and $v_S(t)$ applied to the R and S inputs respectively when the switch throws.

Using your 7400 chip, one slide switch, two 1KΩ resistors, and your scope, construct a contact-bounce eliminator and verify that it produces clean transitions. Be sure to observe the contact-bounce on the switch itself.

Part (B): Most logic circuits employ a clock that paces a sequence of operations. Such a clock produces a continuous stream of pulses made available to critical components. As a result, the more useful types of flip-flops have CLOCK inputs, and these FF's change state at either the "leading" (or "trailing") edge of a clock pulse depending upon the status of the R and S inputs. In this scheme, strictly logic *levels* as opposed to pulses are applied to the R and S inputs. Figure 3 shows a crude *clocked* R-S flip-flop constructed from Nand gates. The first two Nand gates mix the clock pulses with the prevailing levels at the R and S inputs, and the resulting signals are then fed on to the simple R-S flip-flop on the right. Although this circuit has the advantage of clocked inputs, it has the same shortcoming of the previous R-S flip-flop — when R = S = 1, the effect of a clock pulse is indeterminant. Without bothering to construct this clocked R-S flip-flop, fill in Table II to reinforce the idea of clocking action. Assume that the CLOCK input is normally Lo but goes Hi during the pulse interval. Would you expect this FF to trigger on the leading or trailing edge of the clock pulse? What would happen if the R or S input were changed during the pulse?

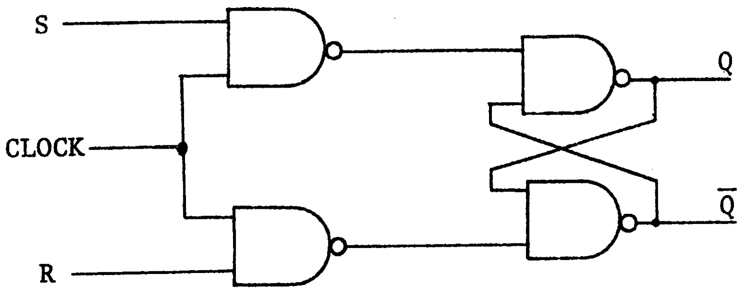
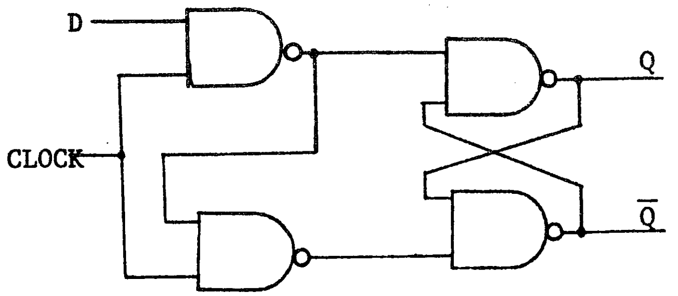


Fig. 3. Simple clocked R-S flip-flop.

Initial Conditions	Inputs S	R	After Clock Pulse
set	0	0	
set	1	0	
set	0	1	
set	1	1	
reset	0	0	
reset	1	0	
reset	0	1	
reset	1	1	

Table II. Truth table for clocked R-S flip-flop.

Part (C): A type-D flip-flop or "latch" resembles the R-S flip-flop of the previous section except that it has only one input, the "D" or data input. Figure 4 shows the D-type flip-flop and its truth table, the latter corresponding to the case where the CLOCK input is normally Lo but goes Hi during the pulse interval. The table shows that the type-D latch flips to whatever state the D input reads at the moment the clock pulse arrives. Note that the indeterminancy of the R-S flip-flop has been eliminated.



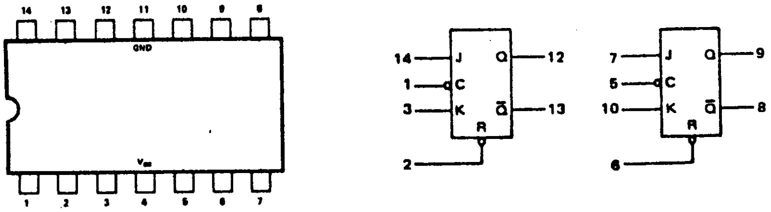
Initial Conditions	D Input	After Clock
set	0	0
set	1	1
reset	0	0
reset	1	1

Fig. 4: Circuit and truth table for D-type FF or latch.

Construct a type-D latch using a single 7400 package. Verify its operation by attaching a LOGIC SWITCH to the D input, the CLOCK to the CLOCK input, and LAMP MONITORS to the CLOCK and Q and $\overline{Q}$ outputs. Run the CLOCK at 1Hz and vary the position of the LOGIC SWITCH. Determine whether the toggling action coincides with the leading or trailing edge of the CLOCK pulse, and what happens when the D input is changed *during* a clock pulse.

Clusters of latches are commonly used in parallel to hold many bits of information until some succeeding stage of the system has a chance to digest it. Such clusters of latches are called *registers.*

Part (D): One of the most complicated and yet useful types of memory elements is the master-slave J-K flip-flop. This device is described in detail in Hughes' *Computer Lab Workbook*, and the reader is urged to study that treatment. A J-K flip-flop can be constructed from seven gates and a transistor steering network, but no one ever constructs them nowadays because pairs of TTL



t_n		t_{n+1}
J	K	Q
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q}_n$

Fig. 5. Package layout, pinout, and truth table for 7473 J-K FF. In the truth table, t_n corresponds to the time *before* the arrival of the Hi-going pulse. The time t_{n+1} is *after* the trailing edge of the pulse.

J-K FF's are available in a 7473 chip. Figure 5 shows the logic and pinout diagram for the 7473 "dual" J-K flip-flop and its truth table. The operation of this device can be summarized as follows:

(1). The outputs Q and $\overline{Q}$ are always complementary.

(2). As the clock input begins to rise with the arrival of a Hi clock pulse, the slave flip-flop is disconnected from the master. Once the Hi is attained, information on the J and K inputs is transferred to the master. When the clock returns to Lo, the information in the master is transferred to the slave. The truth table is correct only if the information on the J and K is stable just before and during the clock pulse.

(3). A Lo on the RESET input returns Q to 0 and $\overline{Q}$ to 1 and holds the FF in its reset state. Only a renewed Hi on the RESET line releases the FF to respond to further pulses.

(4). For J = K = 0, clock pulses produce no change.

(5). For (J = 0, K = 1) and (J = 1, K = 0), the J-K flip-flop behaves like an R-S flip-flop where the J input corresponds to the S input, etc.

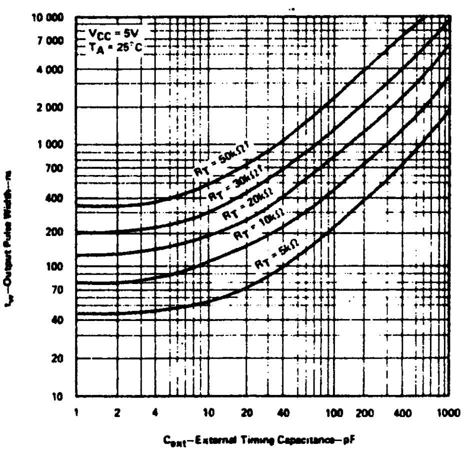
(6). For J = K = 1, a clock pulse produces a complementation of existing outputs.

(7). Changes at the output coincide with the *trailing edge* of a Hi-going clock pulse.

Using a pair of LOGIC SWITCHES, a pair of LAMP MONITORS, a PULSER to simulate a slow clock, and a PULSER on the RESET input, verify the truth table of Fig. 5 and points (1) through (7) above.

Part (E): Construct Hughes' shift register (pages 20-21) and complete his truth table on page 21.

Part (F): Until now we have considered true bistables — flip-flops that remain in one state or the other indefinitely. We conclude this experiment with an investigation of the monostable multivibrator or *one-shot,* a flip-flop with a *preferred* state (the reset state). The one-shot can be set, but after a programmable characteristic time T, it spontaneously reverts back to its reset state where it then remains indefinitely (until another clock pulse comes along). The one-shot is sometimes called a *pulse stretcher*; it is useful for introducing delays in digital circuits and converting ragged pulses of arbitrary shape and length into square pulses of constant amplitude and known duration. The time interval that the one-shot dwells in the set state can range from nanoseconds to seconds; it is programmed by using an external resistor and capacitor which form a simple RC discharge circuit. In the case of the TTL 74123 dual monostable multivibrator, the dwell time is given in Fig. 6 providing that the external capacitor C is less than 1000pF. For C > 1000pF, the dwell time is given by $T = 0.28 \cdot RC\,(1 + 0.7/R)$ [nsec] , where R and C are understood to be measured in KΩ and pF respectively, and 5KΩ < R < 50KΩ. Figure 6 also shows the truth table for the 74123, its pinout, and the external wiring schematic. The first line of the truth table shows that a Hi on the "A" input will hold the one-shot in its reset state and inhibit triggering on the "B" input.



INPUTS		OUTPUTS	
A	B	Q	$\overline{Q}$
H	X	L	H
X	L	L	H
L	↑	⊓	⊔
↓	H	⊓	⊔

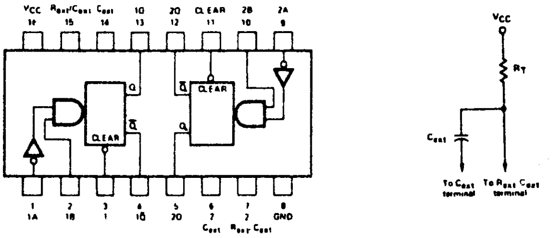


Fig. 6. Dwell time versus R and C for 74123 dual one-shot. Truth table, pinout, and external wiring schematic are also shown.

Similarly, a Lo on the "B" input will hold the one-shot in its reset state and inhibit triggering via the "A" input. The more common situation is to hold "A" Lo while applying a pulse to "B" (line three of the truth table) or hold "B" Hi while applying a pulse to "A". The arrows in the truth table indicate that the 74123 actually requires only a transition from Lo to Hi (upward pointing arrow) or

Hi to Lo (downward pointing arrow) for triggering. Obviously a pulse provides both types of transitions, but the truth table permits one to determine whether the one-shot will trigger off the leading or trailing edge of a pulse, depending upon the type of pulse and the selected input. The CLEAR input permits one to reset the one-shot (regardless of its state) with a Lo-going pulse; such a pulse can be used to terminate the dwell period prematurely. Termination coincides with the leading edge of the clear pulse.

To become familiar with the operation of the 74123, verify Eq. (1) by using an external capacitance of approximately 0.019μF, a resistance of 15K, a PULSER for clearing operations, the CLOCK on the Digi-Designer to toggle the "B" input, and your scope to monitor the output waveform and measure the dwell time. Careful adjustment of the scope's trigger controls may be necessary to produce a stable waveform. Start with the CLOCK running at 1KHz, but once you have verified that the circuit is operating properly, decrease the clock rate to 100 Hz. Do you understand the resultant effect on the 74123's output? Is the 74123 operating as a pulse-stretcher or pulse shortener? The 74123 is retriggerable, meaning that if a second clock pulse is received before the end of the dwell period, a new dwell period will be initiated automatically. Increase the CLOCK rate to 10 KHz and verify this property of the 74123.

As a final exercise, we construct a crude coincidence circuit that tests for near coincidence between two successive pulses. The circuit is shown in Fig. 7. The one-shot on the left generates a continuous train of pulses of approximately 50μsec in duration, with a repetition rate equal to that of the Digi-Designer's CLOCK. Reuse the 0.019μF capacitor and a 6K resistor to establish a 50μsec dwell time for this one-shot. The second one-shot triggers on the trailing edges of these 50μsec pulses, and its dwell time should be programmed at about 200μsec (use another 0.019μF capacitor and a 25K or 30K resistor). When the second one-shot is set, it "enables" the J input of the 7473 flip-flop, which proceeds to perform the test for "nearness" between successive pulses in the input pulse stream. The dwell time of the second one-shot suggests that two adjacent pulses separated by as much as 200μsec will be judged "coincident" by this circuit. Hence if the Digi-Designer's CLOCK is set at 1KHz, implying pulse separations on the order of 1000μsec, the circuit will detect no coincidences. If, on the other hand, the CLOCK runs at 10 KHz, one should get "coincidences." Note in Fig. 7 that the circuit announces the detection of coincidences with a LAMP MONITOR. CLEARing capabilities for initialization are provided; use the "1" output of the pulser to perform CLEARing. Before constructing the circuit, make sure that you understand its operation. Use your scope to observe the circuit's behavior at various stages.

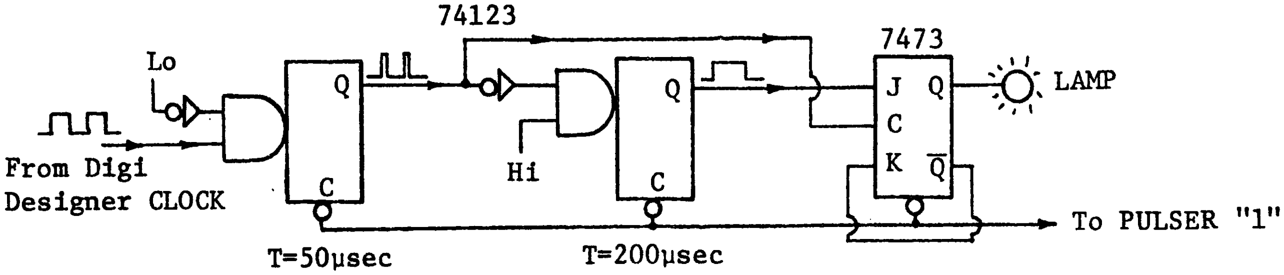


Fig. 7. Simple "coincidence circuit" based upon 74123 chip.