DIGITAL CIRCUITS

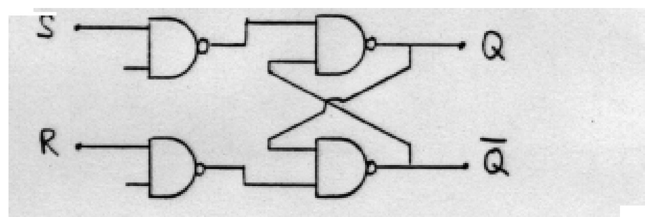
EXPERIMENT 14: DIGITAL CIRCUITS FLIP-FLOPS AND THE UART (12/9/02)

In this experiment we will construct a few simple flip-flop circuits, and use JK flip-flops to carry out some complex operations. We will also use a Universal Asynchronous Receiver-Transmitter, or UART (a large scale integrated circuit communications device which is constructed from many flip-flop based circuits) to send messages from one logic board to another.

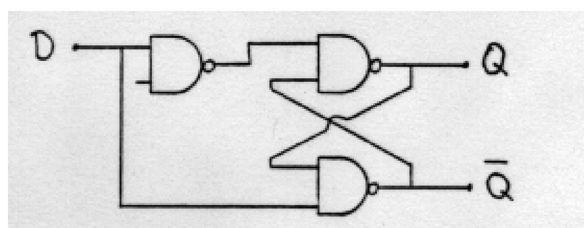
When introducing signals into the logic board from an external source (the function generator or another logic board) the grounds of the two circuits need to be connected. It will also be necessary to bring the input in through a NAND gate (or through two NAND gates connected in series if the input should not be inverted).

1. Use NAND gates to construct the following flip-flops. In each case sketch the circuit and experimentally determine the function table.

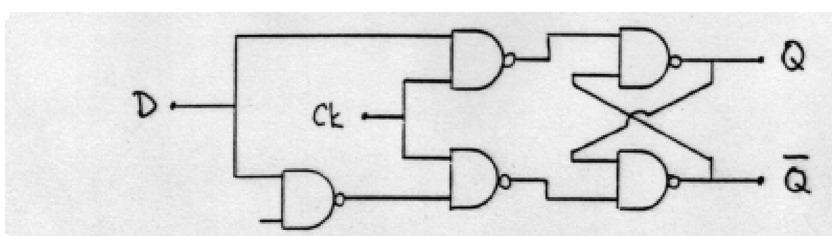
a) set-reset flip-flop (RS FF)



b) data-type flip-flop (D FF)

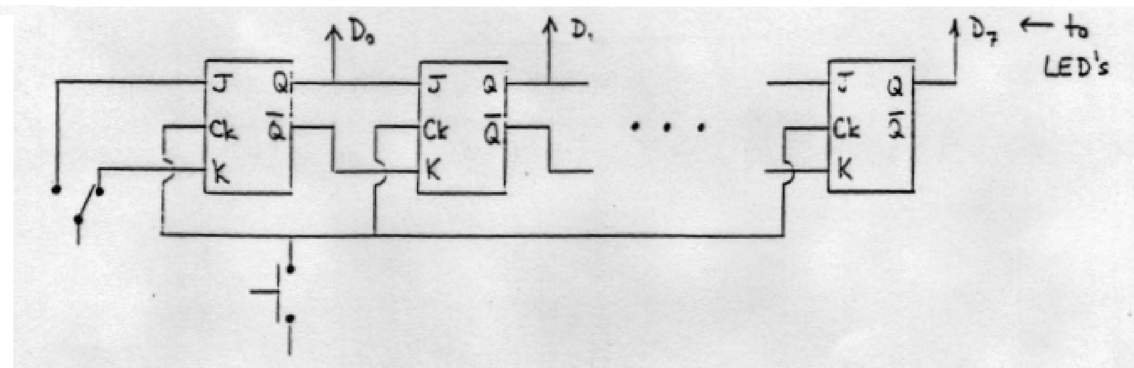


c) clocked D FF



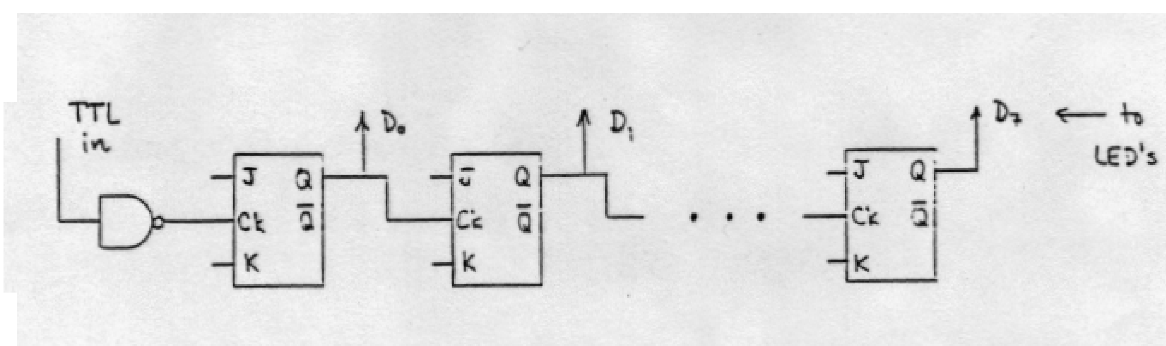
2. Investigate the behavior of the J-K flip-flops located on the logic boards. First determine how the flip-flop is triggered; i.e., does the output switch a) any time the clock pulse is positive, b) on the negative edge of the clock pulse, or c) on the positive edge of the clock pulse?
Then experimentally determine the function table for the flip-flop. Be sure to observe what final state is obtained for all possible inputs (J,K) = (0,0), (0,1), (1,0), and (1,1).

3. Construct the 8 bit shift register shown below. Use a toggle switch to vary the input to the first stage and a pushbutton to provide the clock signal. Observe the behavior of the shift register and write a brief description of what the circuit does.



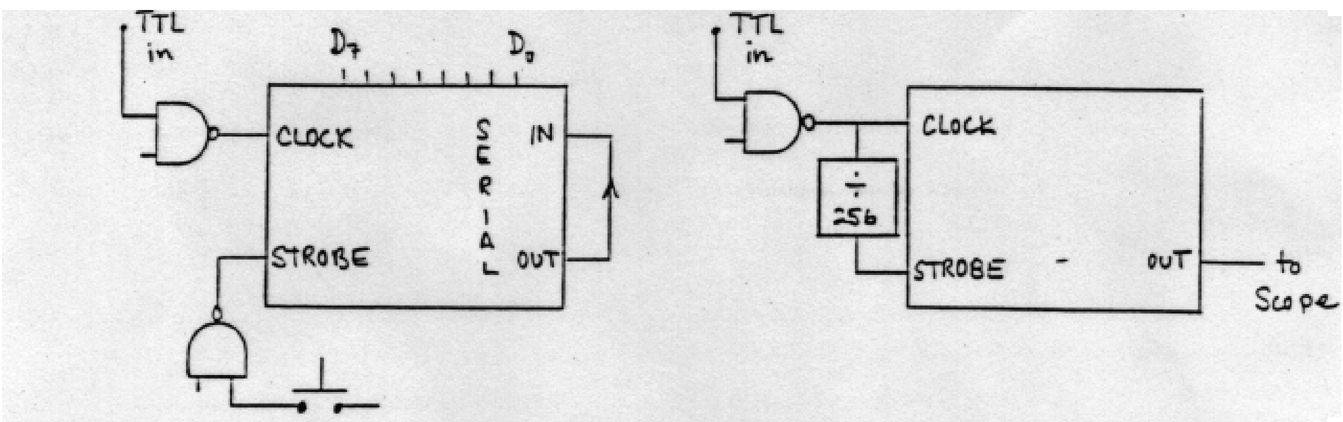
Next connect the output to the input to make a circular shift register, and use the TTL output from the function generator for the clock (you will need to bring the signal in through a NAND gate). Observe the Q output from the last flip-flop with the scope. Note that what you see in this case is a serial representation of the information stored in the register.

4. Construct the 8-bit (0 - 255) ripple counter shown below. Use the TTL output from the function generator as the input and observe the Q outputs with the scope. Sketch the input and the output from the first three flip-flops. Leave the circuit set up, since it will be used again in step 5.



5. The final step is to observe the operation of the UART. The UART is capable of both sending and receiving information in a serial form. To use the UART, an 8-bit data word representing a number or letter is entered at the parallel inputs D_0 through D_7. To transmit this information one enters a "0" pulse at the strobe input. This causes the UART to transmit the data word, one bit at a time, from the serial output. The data word is always preceded by a "0" and followed by two "1"s. Each transmitted bit has a length of 16 clock pulses. Any data received by the UART at the serial input is decoded in a similar manner, and the result appears on a set of parallel outputs (these are connected to LED's on the logic board).

   (a) Start by setting up the UART circuit as shown in the diagram on the left below. Set the clock rate to about 100 kHz, and use switches to set the input bits (D_0 to D_7) to either "0" or "1". Now use the push-button to strobe the UART. This causes the UART to send the data word to itself, and thus the information you entered should now appear on the LED's.



   (b) Next try using the ripple counter to generate the strobe inputs as shown in the right-hand diagram. In this case we get one strobe for each 256 clock pulses, and the UART will transmit the same data word over and over. Use the scope to look at the serial output, and observe what happens as you change various input bits between "0" and "1". Sketch some of the output waveforms in your notebook, showing where the information contained in each of the input bits appears in the serial output.

   (c) Connect the serial input and output of the UART to the UART on one of the other logic boards (for example, the one being used by a neighboring group). Note that on each board you will need to bring the input in through two NAND gates connected in series. You will also need to connect the grounds of the two boards, and to set the two clocks to approximately the same frequency. Send a message (using ASCII code) from one board to the other.

ASCII CODE

Keyboard Character	ASCII	Keyboard Character	ASCII
A	100 0001	S	101 0011
a	110 0001	s	111 0011
B	100 0010	T	101 0100
b	110 0010	t	111 0100
C	100 0011	U	101 0101
c	110 0011	u	111 0101
D	100 0100	V	101 0110
d	110 0100	v	111 0110
E	100 0101	W	101 0111
e	110 0101	w	111 0111
F	100 0110	X	101 1000
F	110 0110	x	111 1000
G	100 0111	Y	101 1001
g	110 0111	y	111 1001
H	100 1000	Z	101 1010
h	110 1000	z	111 1010
I	100 1001	0	011 0000
I	110 1001	1	011 0001
J	100 1010	2	011 0010
j	110 1010	3	011 0011
K	100 1011	4	011 0100
k	110 1011	5	011 0101
L	100 1100	6	011 0110
L	110 1100	7	011 0111
M	100 1101	8	011 1000
m	110 1101	9	011 1001
N	100 1110	,	010 1100
n	110 1110	.	010 1110
0	100 1111	+	010 1011
o	110 1111	—	010 1101
P	101 0000	(	010 1000
p	110 0000	)	010 1001
Q	101 0001		
q	111 0001		
R	101 0010		
r	111 0010		