LABORATORY 2 SEQUENTIAL LOGIC I Latches & Flip Flops

1. Learning objectives

To familiarise the student with

- the properties and limitations of the basic SC latch
- the concept of a gated or enabled latch
- the operation of D & JK flip-flops
- the operation of a ripple counter

2. Before the lab

Read through your class notes on latches and flip flops (Section 2 in class notes) and answer the questions below.

2.1 Show the construction of a basic NAND gate latch and use a truth table to show how the outputs depend on the inputs.

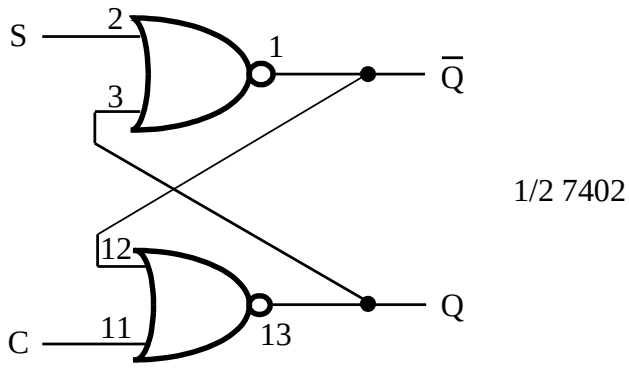
2.2 Identify at least three different limitations of this basic S-C latch and show how each of these will be solved to produce a practical J-K flip flop.

2.3 The J-K flip flop has two input; show how this design can be modified to produce a D flip flop with a single data input.

3. Lab Procedure

3.1 The Basic SC Latch

Construct the SC latch from two NOR gates on a 7402 package:-



Connect SC to logic switches (check V_{ext} = +5V), and Q, $\overline{Q}$ to LED indicators. Record the results of the following experiments and compare them with the truth table given earlier.

(a) Initialise the latch to $Q\overline{Q} = 01$ by setting S=0 and switching C momentarily to 1.

(b) Set S to 1, record $Q\overline{Q}$, return S to 0, and record $Q\overline{Q}$.

(c) Repeat procedure (b) for C, with S fixed at 0.

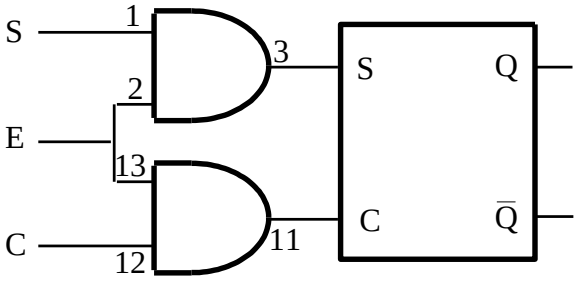
(d) Set SC to 11 and record $Q\overline{Q}$.

(e) Return SC to 00 through the two sequences $11 \rightarrow 10 \rightarrow 00$ and
(separately) $11 \rightarrow 01 \rightarrow 00$ and
record the final $Q\overline{Q}$ in each case.

(f) Set SC to 11 and then to 00 by flicking both switches at the same time. Repeat several times to see if the transition = $11 \rightarrow 00$ gives reproducible results.

3.2 The Enabled SC latch

Gate the inputs of your basic SC latch with an enable signal using 7408 AND gates to give an enabled SC latch:-

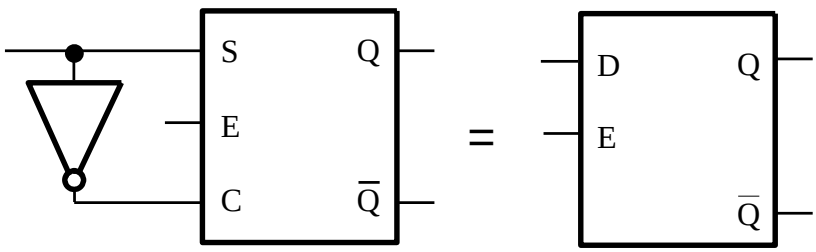


1/2 7408

Connect S, C, E to logic level switches and confirm that the latch is only sensitive to the SC inputs when E = 1.

3.3 The transparent D latch

Modify the enabled SC latch of 3.2 by using an inverter to force C = $\overline{S}$. The resulting configuration, called a transparent D latch, has a single input D = S.



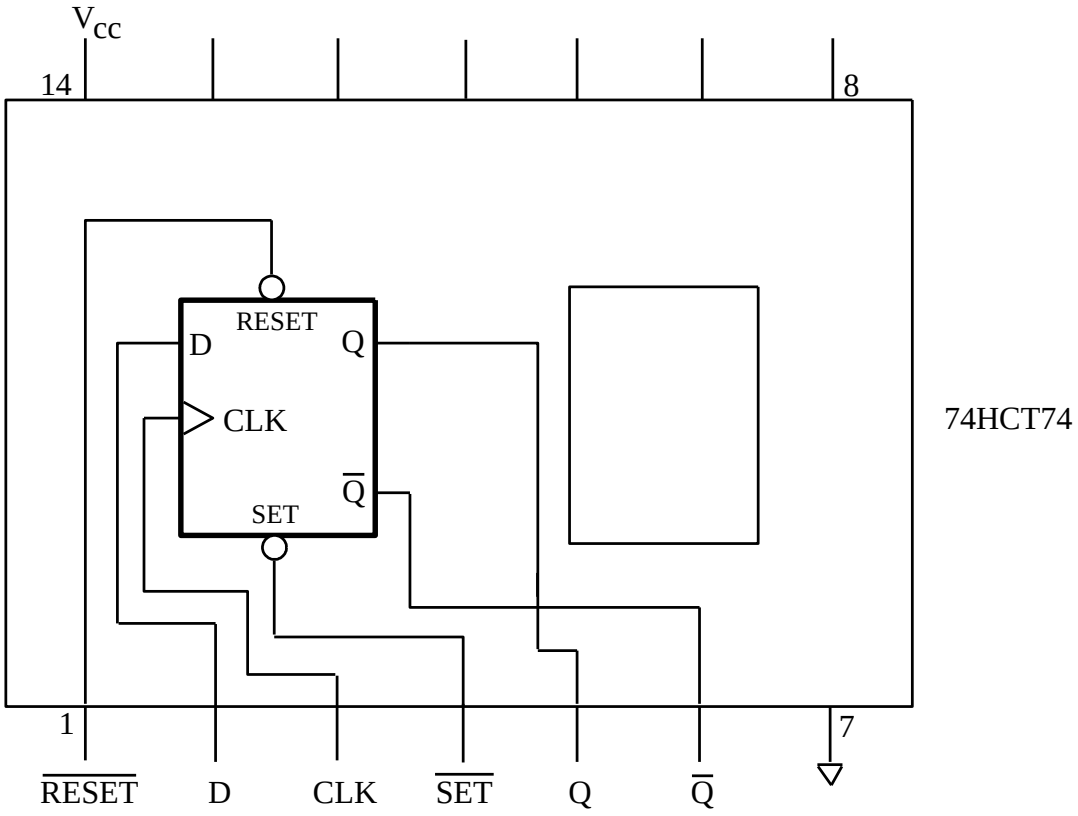
One of the main applications of this device is to sample a digital signal at a particular time, and to store the value for later processing. Observe this by connecting D to the SDS square wave generator, set for ~ 0.1 Hz pulses, and E to the normally LO pulser. Use LED monitors on both lines, and on the Q output. Investigate this digital "sample

& hold" application by momentarily operating the pulser when the input D is LO, and when it is HI.

3.4 The Edge-Triggered D flip-flop

A flip flop is normally set in synchronization with a clock pulse. The outputs are then only sensitive to the inputs only for a very short period (~ns) around the leading (+ve) or falling (-ve) edge of the enable or clock pulse. The 74HCT74 is a dual, positive edge-triggered D flip flop as shown below.

Construct the following circuit:-



Connect D, $\overline{\text{RESET}}$, $\overline{\text{SET}}$ to logic switches, Q and $\overline{Q}$ to logic indicators and CLK to the SDS square wave generator set to 0.1 Hz and to a logic indicator.

First investigate the properties of the $\overline{\text{RESET}}$ and $\overline{\text{SET}}$ lines. Does their operation depend upon the clock status? With both lines HI, investigate whether the flip flop is sensitive to its inputs during the CLK = HI pulse, rather than just the LO→ HI edge.

Finally connect CLK to a SDS square wave generator at ~ 1kHz and D to $\overline{Q}$, as in section 4.3, and check if the problems apparent there have been overcome. Note the timing relation between Q and the CLK signal using the oscilloscope.

3.5 The JK edge-triggered flip-flop

In the next two Sections we will use the 74HCT112 dual JK negative-edge-triggered

FF's. Each has 5 inputs (J, K, CLK, $\overline{\text{RESET}}$, $\overline{\text{SET}}$ and two outputs (Q,$\overline{Q}$) .

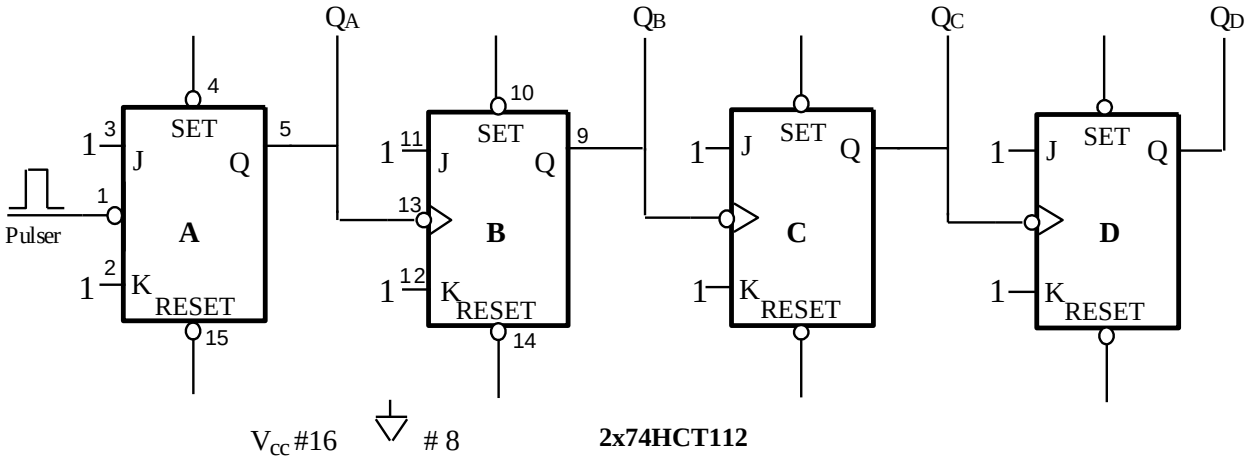
Confirm the basic operation of this device by connecting the CLK input to a logic pulser, connect Q, and $\overline{Q}$ to logic level indicators, and connect J and K to logic level switches. Construct the truth table for this flip-flop:-

Inputs before nth Clock Pulse		Outputs after nth Clock Pulse	
J	K	Q_{n+1}	$\overline{Q}_{n+1}$

Finally, set JK to 11, the pulse generator to ~ 1 kHz, and observe both CLK & Q on the oscilloscope. Sketch the waveforms.

3.6 A 4-bit ripple counter

One of the many applications of J-K flip flops is in counters and we will now look at a four-bit ripple counter. Construct the circuit shown below. Decide on a wiring colour code to use for all of these functions and adhere rigidly to it during the experiment (i.e. all SETS blue etc). This practice will save considerable time in debugging.



Connect all the J,K inputs to a logic 1 by means of a **single** 1 kΩ resistor to the V_{cc} rail. Connect the sets and resets to individual logic level switches, the clock input of the first FF to the HI pulser, and the FF outputs to LED indicators **in left to right order DCBA**. Investigate the effect of the set and reset control lines and then reset the counter to 0000. By exercising the pulser, complete the following table:

Pulses in	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1				
2				
3				
.				
.				

Connect the input to the pulse generator set for 1 kHz pulses and examine the frequency relationship between Q_D, Q_C, Q_B, Q_A and the clock input on the oscilloscope. (Note that precise measurements of periods are **not** required here - it is the ratios which are important).

4. Post Lab (or if time allows during the lab)

4.1 Setting the modulus of a counter

The simple 4-bit ripple counter that you built in Section 3.6 progresses through 16 states and so is a counter of modulus 16. The four flip flop stages in the counter also means that it can be used to construct a counter with modulus = $2^{stages} = 2^4 = 16$. When this counter reaches the maximum count of 15 (or 1111) it will automatically reset to the minimum state of zero (0000) and start counting up again. However, we do not always want to use the full mod number of a counter and we may want it to reset when it reaches a specific maximum number. This can be done by using external logic gates to reset when the specified maximum count is reached.

Show what additional gates will be needed in order to turn your mod 16 counter into:
(i) A mod 12 counter
(ii) A mod 10 counter.

4.2 Adding more flexibility to your counter

The counter you built in Section 3.6 counts up on every clock pulse. Show by using logic diagrams how you can change your counter (you may add additional logic gates) to:
(i) Act as a down counter
(ii) be flexible so that you can choose whether it is counting down or up by means of a logic input.

5. Report

Put all your results from Section 3 and 4 together in a short report.