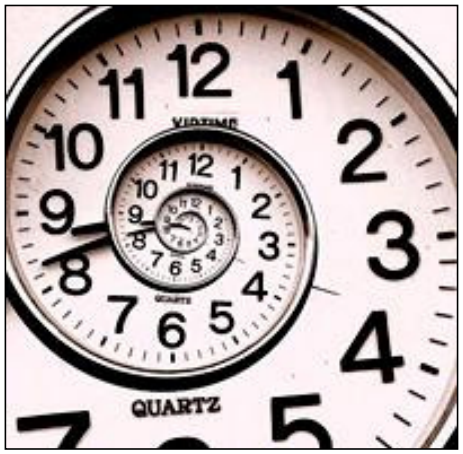
The New England Radio Discussion Society electronics course (Phase 5, cont'd.)

Clock your flip-flops!

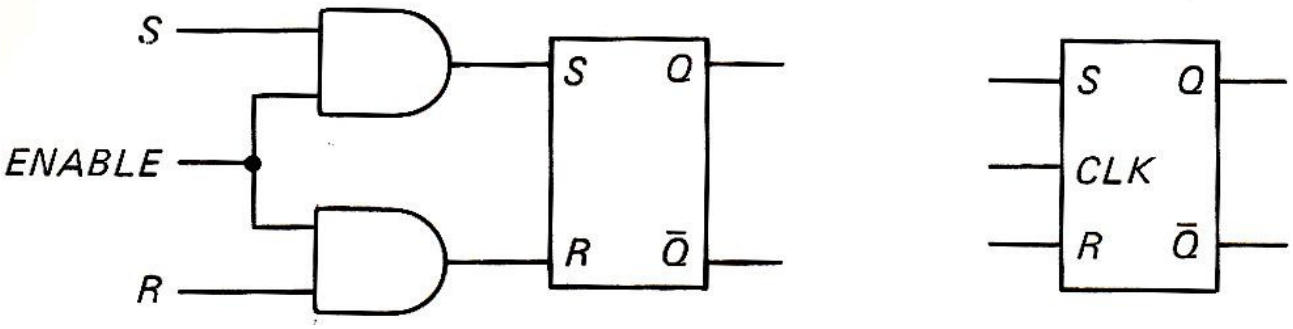


AI2Q – May 2017

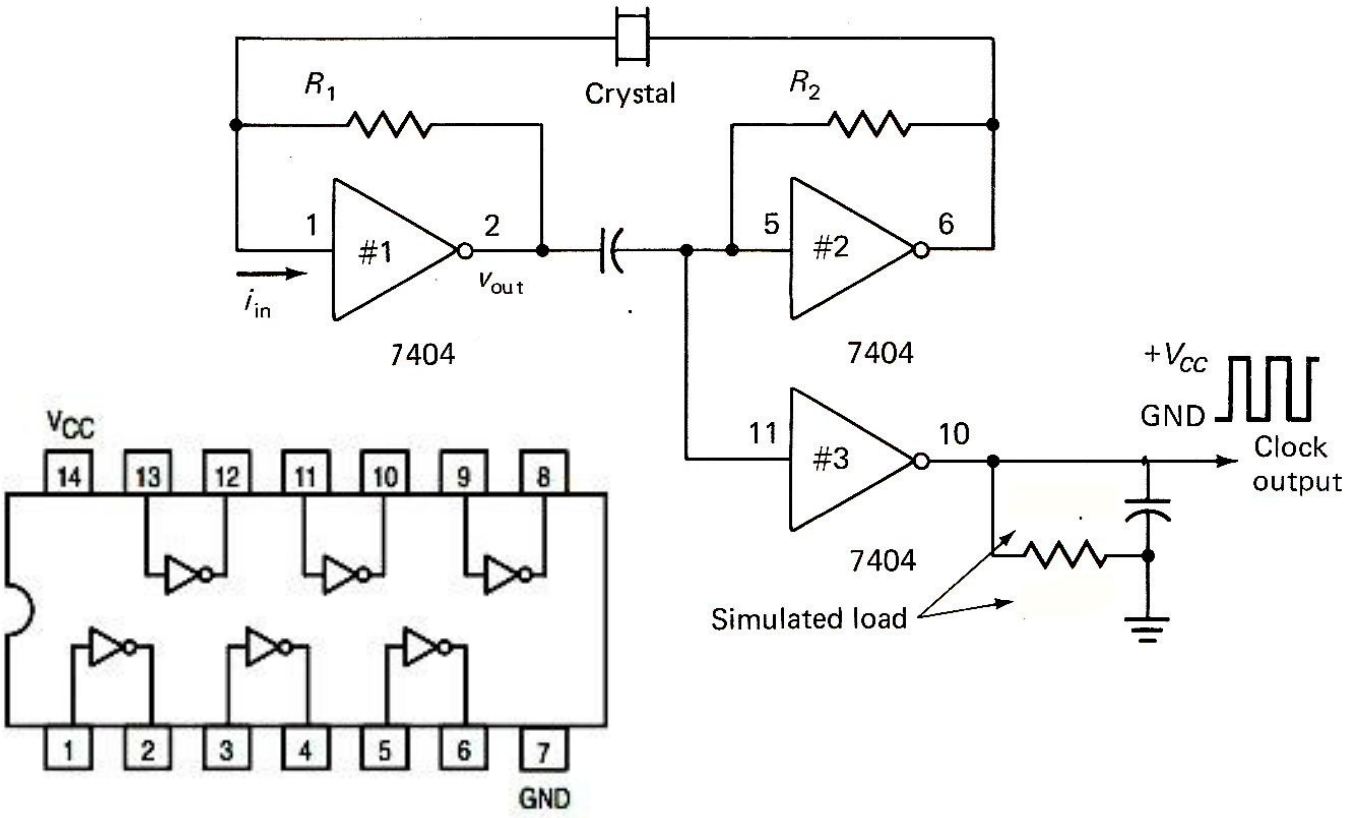
Adding two AND gates to a Set-Reset (RS) FF permits it to be enabled or disabled. If the *ENABLE* line is high, data from R or S can toggle the flip-flop. If *EN* is low, neither S nor R can pass data.

This permits *clocking* of the RS FF, so that it can <u>store</u> information for a designated period of time.

Notice the enable pin is referred to as the *CLK* pin on the schematic symbol (right).

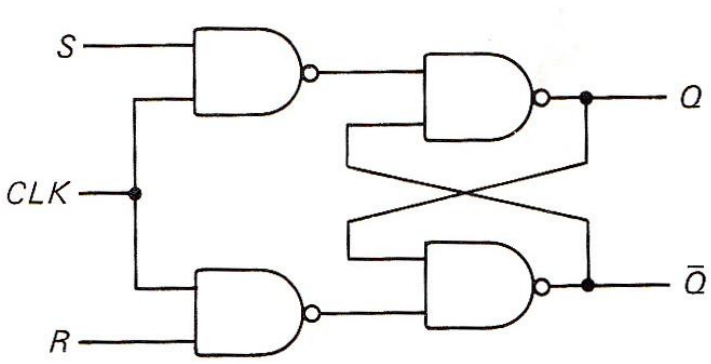


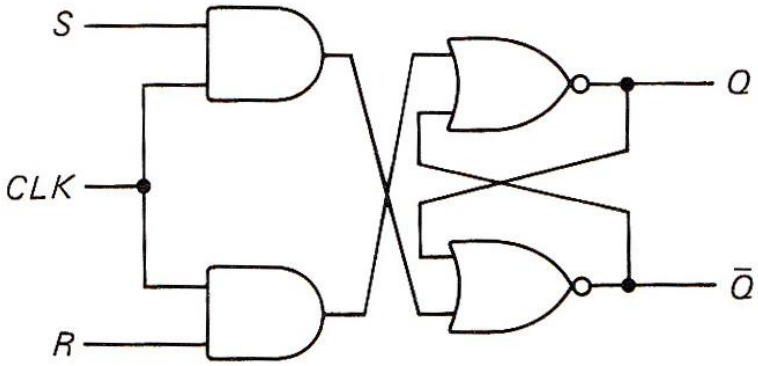
Here's a typical clock oscillator circuit based on a 7404 hex inverter 14-pin DIP-packaged TTL chip.



There are multiple logical ways to realize a clocked RS flip-flop.

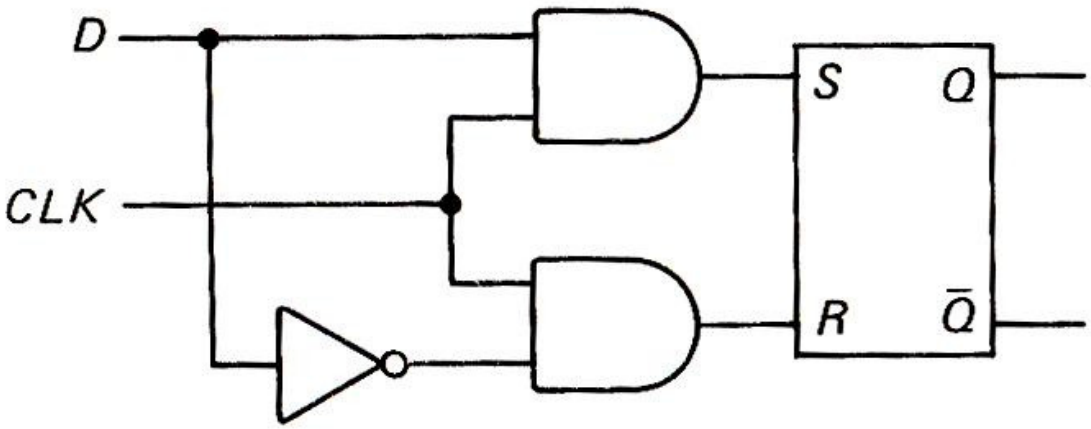
Many, many clocked flip-flops are implemented in very large-scale integrated circuits (*VLSI*) and ultra-large scale integrated circuits (*ULSI*).

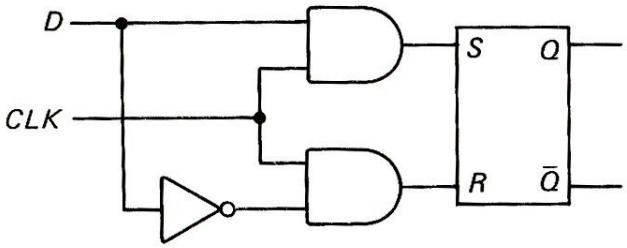




Generation of two signals to drive a FF can be a disadvantage in some applications. Also, the forbidden condition can happen where R and S occur together.

This is solved by the use of a *D-Type* FF, a circuit that needs only a single data input, D.





The D-type FF prevents the state of D from reaching Q output until a clock pulse occurs.

When the clock is low both AND gates are disabled. Therefore D can change without affecting the value of Q.

When the clock is high, both AND gates are enabled. That forces Q to equal the value of D.

When the clock then goes low again Q retains (stores) the value of D.

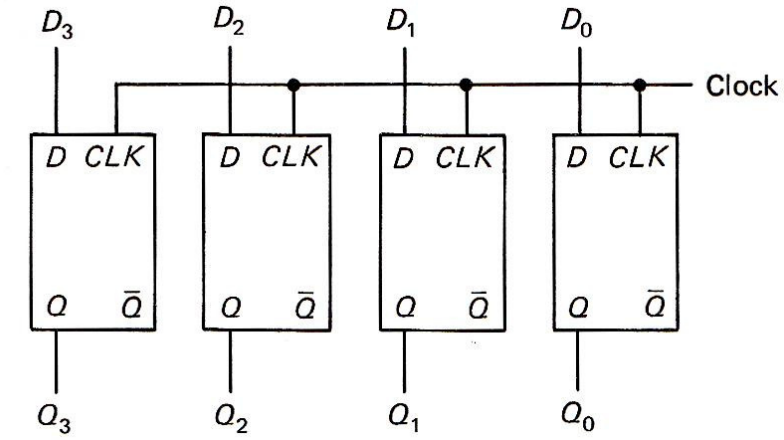
In short, D-type flip-flops transfer data to their outputs only when clock pulses are received.

Multiple flip-flops can be used to store data *words*.

When the clock goes high, input data words are loaded, and data appears at the outputs.

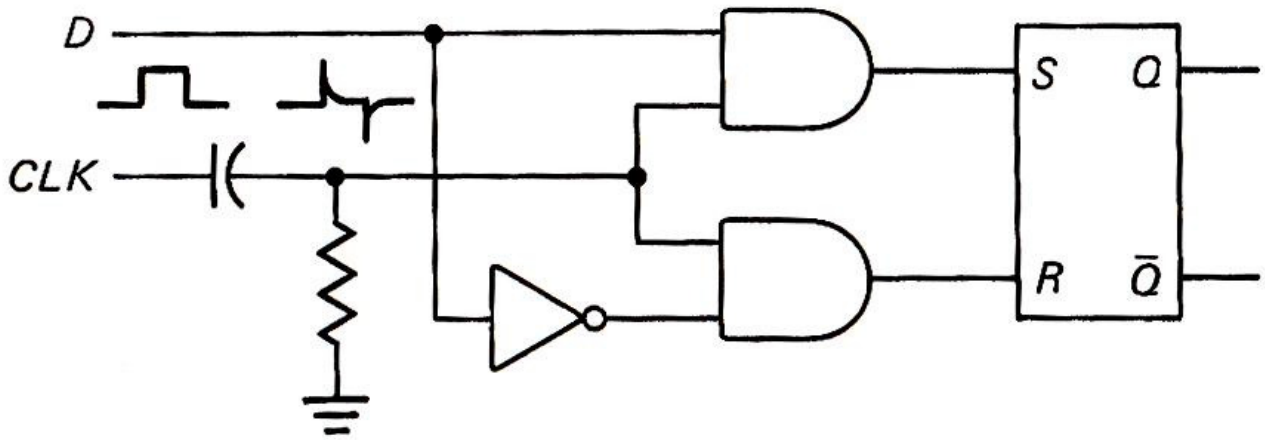
When the clock goes low, the outputs latch to retain the data. As long as the clock is low D values can change without affecting the Q values.

Put two of these quad latches together and you can store a *byte*!

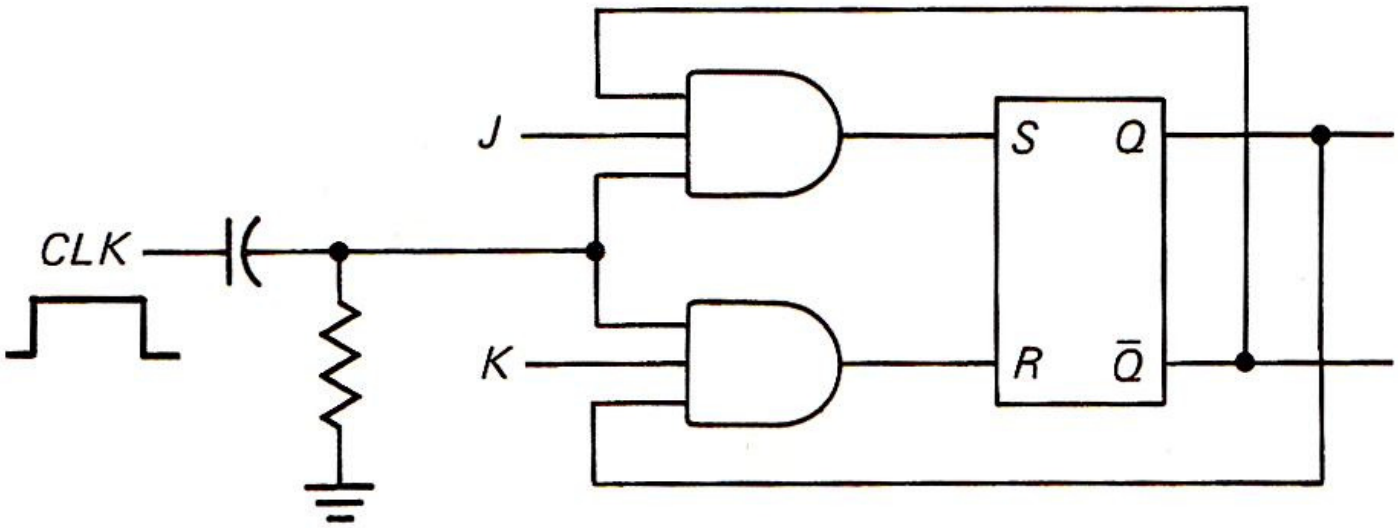


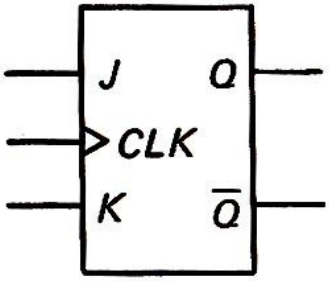
There are also so-called *edge-triggered* flip-flops that sample data at a unique point in time. An RC time-constant at the device's input is selected to be shorter than the clock's pulse width.

A resulting positive spike enables the AND gate for an instant. At this point in time D hits the flip-flop, forcing Q to Set or Re-set.

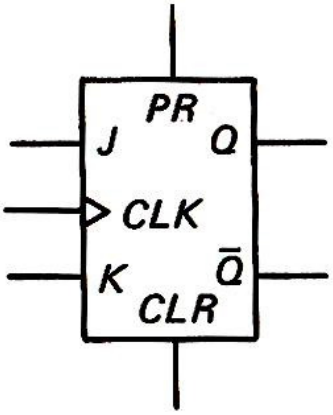


The *J-K flip-flop* is also used for counting. J and K inputs are control inputs; they determine what the flip-flop does when a positive clock edge arrives.

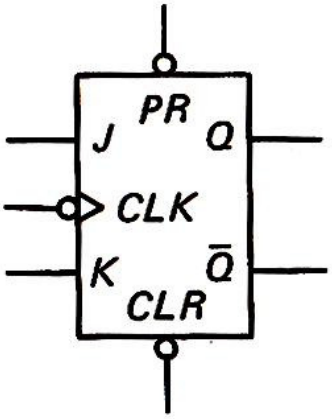




Basic symbol



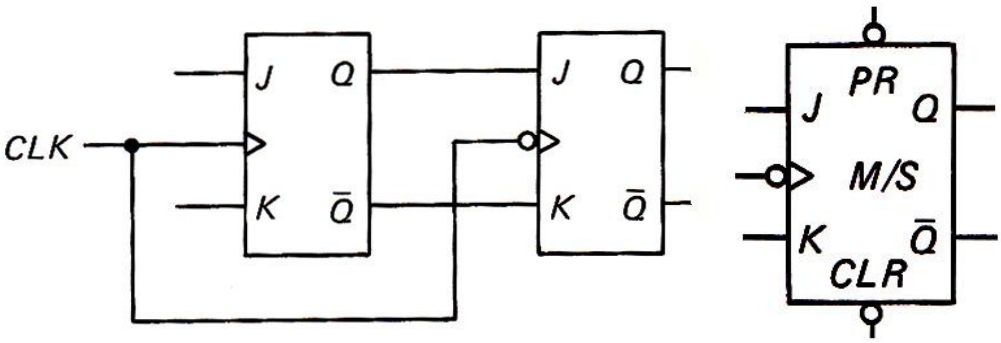
Positive preset and clear



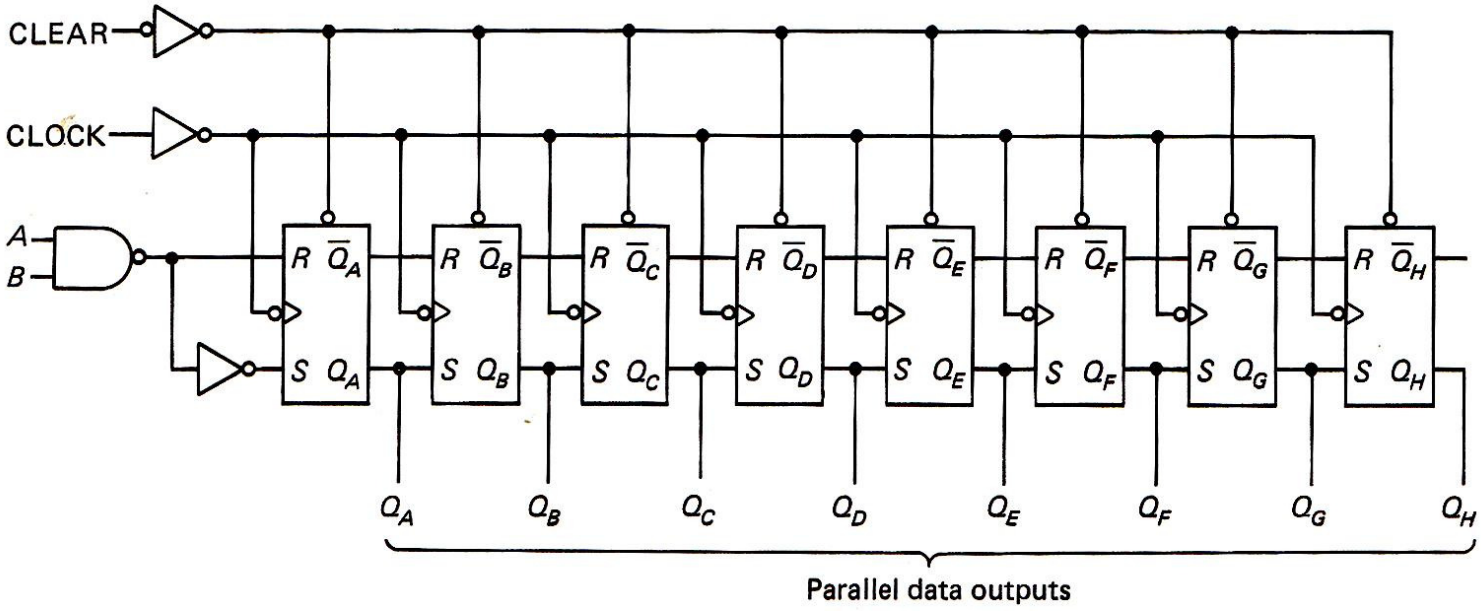
Inverted preset and clear

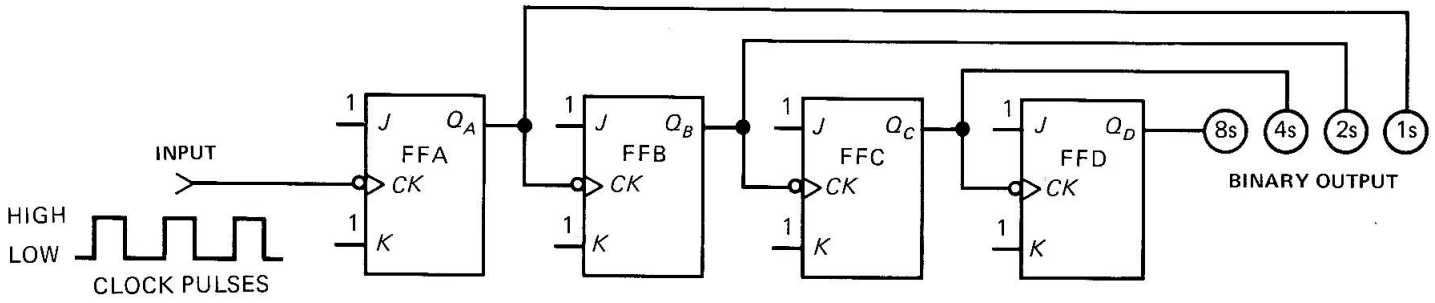
As you can see, there are a number of types of J-K flip-flops.

This is a *master-slave* JK flip-flop. →

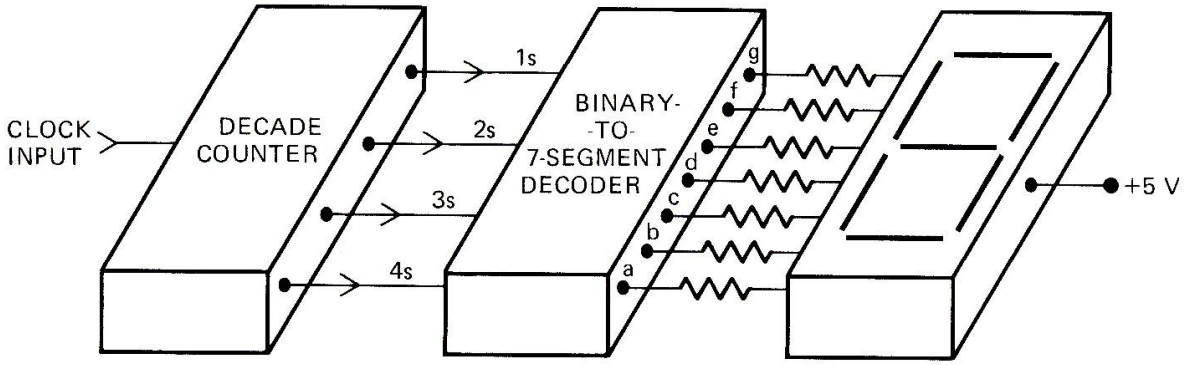


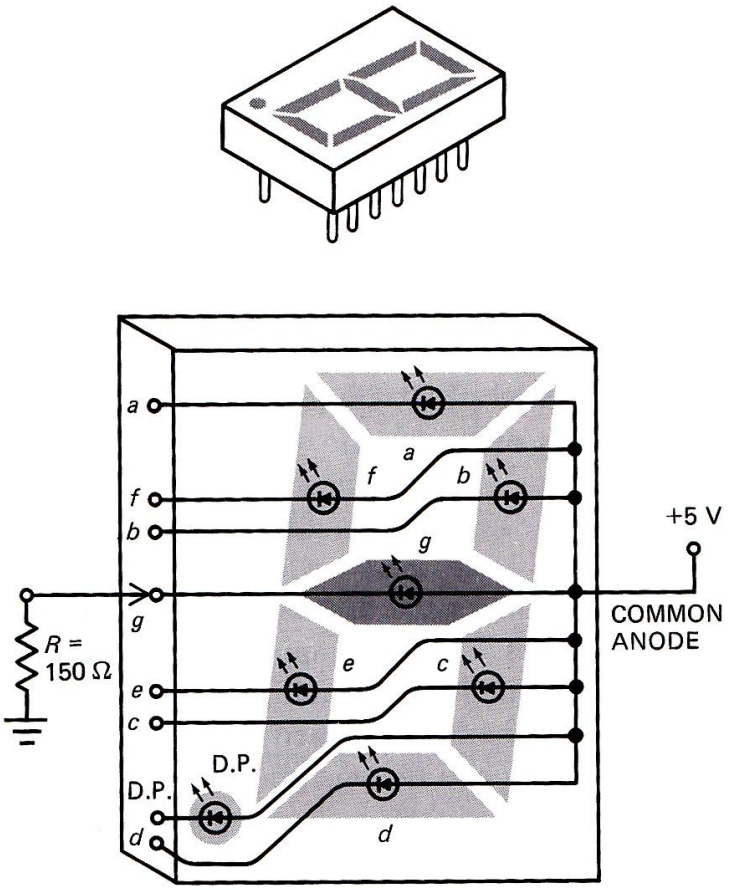
This is a typical application circuit using an 8-bit *74164* shift-register TTL chip. It accepts a serial clock signal at its input, and generates a parallel binary 8-bit word, or byte, at its output.





A decoder chip can convert a parallel binary output to something human-readable, such as a 7-segment LED.

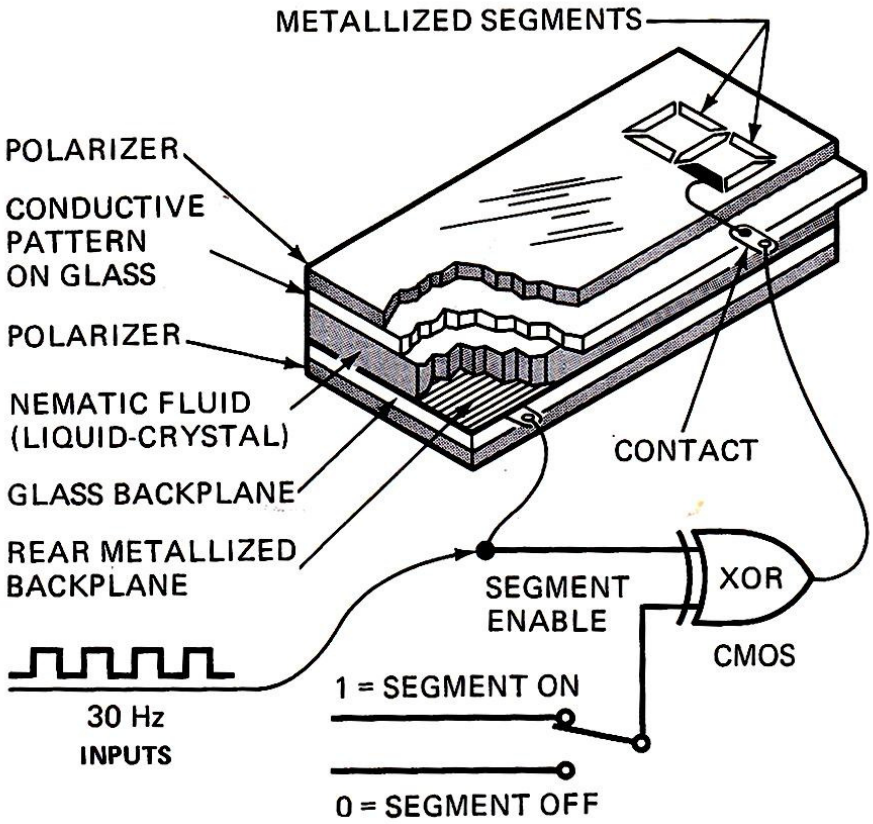




Each part of the 7-segment LED's display is based on a separate internal light-emitting diode, all sharing a common anode.

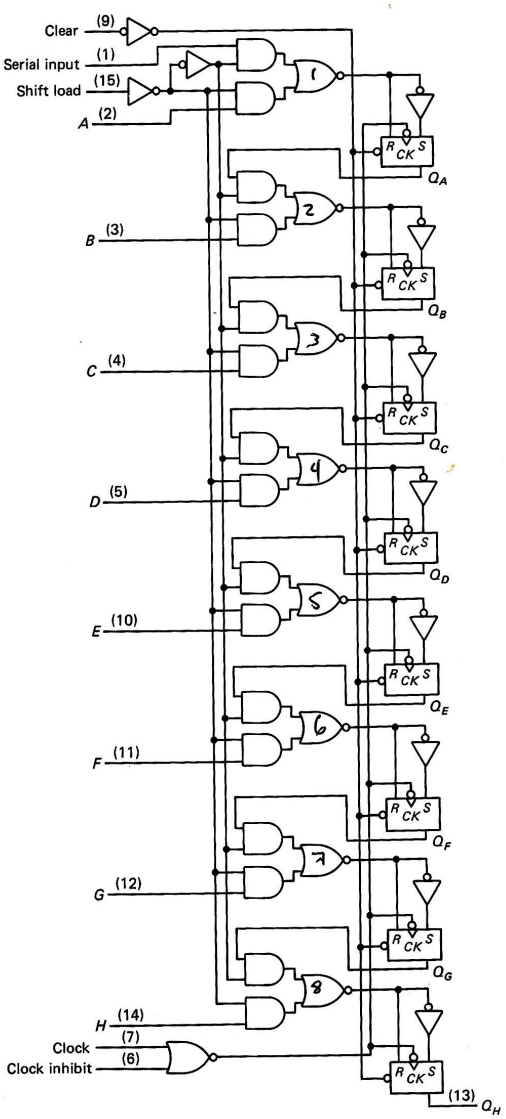
Most 7-segment LEDs run on 5V dc supplies.

This example 7-segment *liquid crystal display* (LCD) is clocked, and uses an exclusive OR gate at its input.



Parallel data in ->

This circuit accepts parallel digital data and generates a serial data train at its output.



<- Serial data out

It’s logical, eh?

Vy 73, Al2Q, Alex