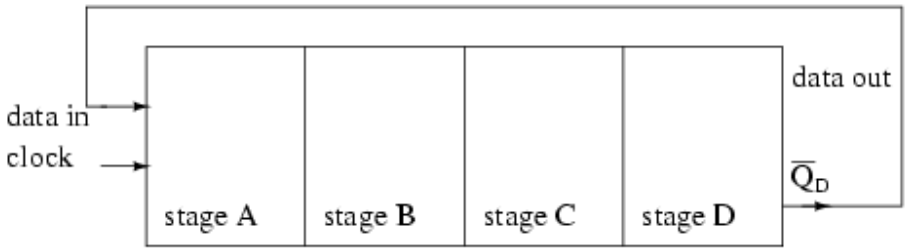
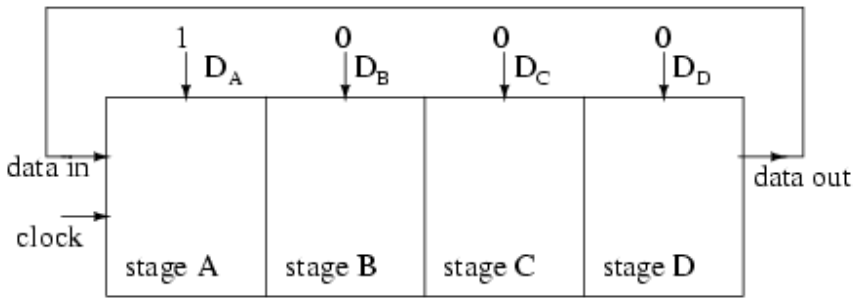
# RING COUNTERS

If the output of a shift register is fed back to the input. a ring counter results. The data pattern contained within the shift register will re-circulate as long as clock pulses are applied. For example, the data pattern will repeat every four clock pulses in the figure below. However, we must load a data pattern. All **0**'s or all **1**'s doesn't count. Is a continuous logic level from such a condition useful?



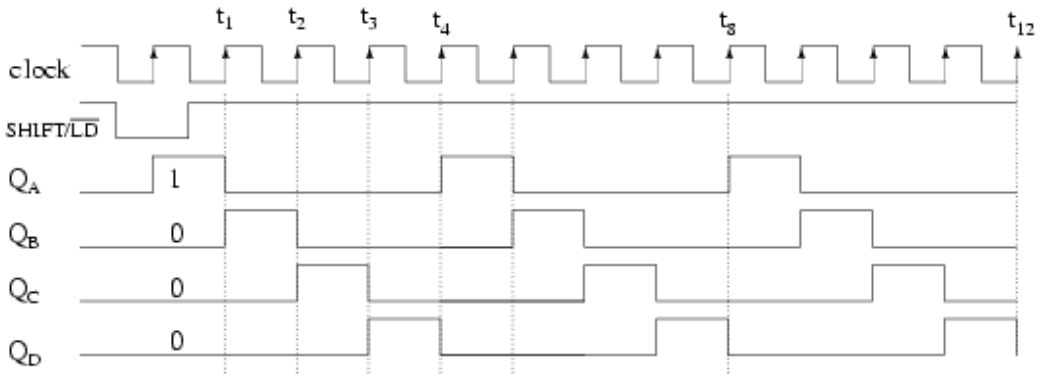

Ring Counter, shift register output fed back to input

We make provisions for loading data into the parallel-in/ serial-out shift register configured as a ring counter below. Any random pattern may be loaded. The most generally useful pattern is a single **1**.



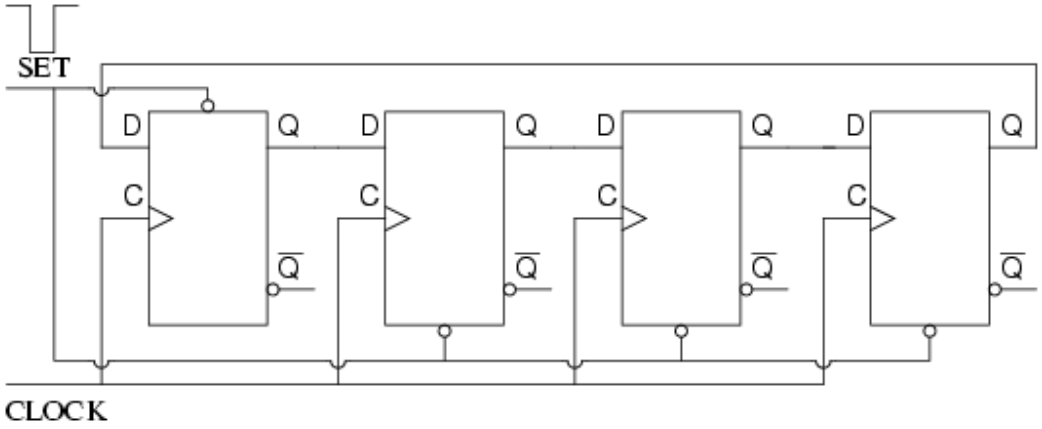

Parallel-in, serial-out shift register configured as a ring counter

Loading binary **1000** into the ring counter, above, prior to shifting yields a viewable pattern. The data pattern for a single stage repeats every four clock pulses in our 4-stage example. The waveforms for all four stages look the same, except for the one clock time delay from one stage to the next. See figure below.




Load 1000 into 4-stage ring counter and shift

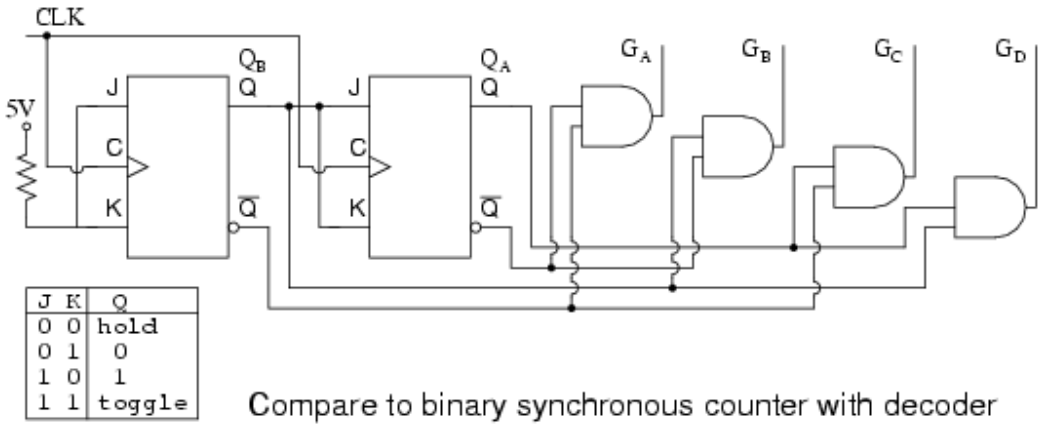
The circuit above is a divide by **4** counter. Comparing the clock input to any one of the outputs, shows a frequency ratio of 4:1. How may stages would we need for a divide by 10 ring counter? Ten stages would recirculate the **1** every **10** clock pulses.




Set one stage, clear three stages

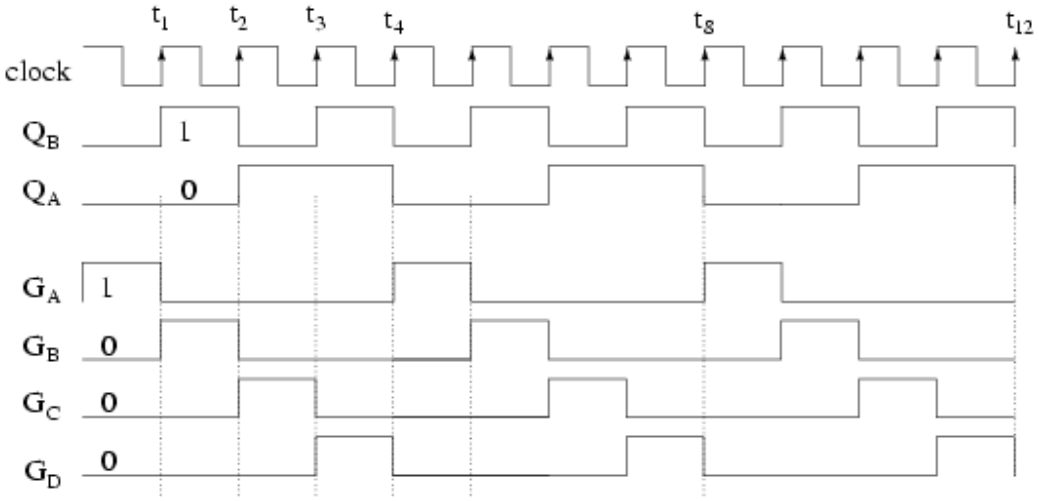
An alternate method of initializing the ring counter to **1000** is shown above. The shift waveforms are identical to those above, repeating every fourth clock pulse. The requirement for initialization is a disadvantage of the ring counter over a conventional counter. At a minimum, it must be initialized at power-up since there is no way to predict what state flip-flops will power up in. In theory, initialization should never be required again. In actual practice, the flip-flops could eventually be corrupted by noise, destroying

the data pattern. A "self correcting" counter, like a conventional synchronous binary counter would be more reliable.




Compare to binary synchronous counter with decoder

The above binary synchronous counter needs only two stages, but requires decoder gates. The ring counter had more stages, but was self decoding, saving the decode gates above. Another disadvantage of the ring counter is that it is not "self starting". If we need the decoded outputs, the ring counter looks attractive, in particular, if most of the logic is in a single shift register package. If not, the conventional binary counter is less complex without the decoder.




Compare to binary synchronous counter with decode. waveforms

The waveforms decoded from the synchronous binary counter are identical to the previous ring counter waveforms. The counter sequence is ($\mathbf{Q_A\ Q_B}$) = (**00 01 10 11**).