Video Lecture 3
Week 2
Essential sequential designs

ENGN4213/6213

Digital Systems and Microprocessors

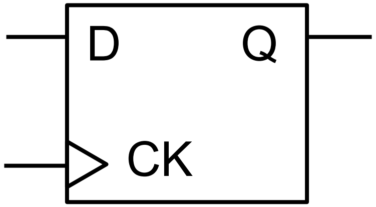
What’s in this lecture

- Essential sequential designs
  - Counters
  - Clock dividers
  - Shift registers

Resources

- Wakerly (5th edition) 11.1.1-11.1.2 for counters
- Wakerly (5th edition) 11.2.1 for shift registers

Some uses of flip-flops



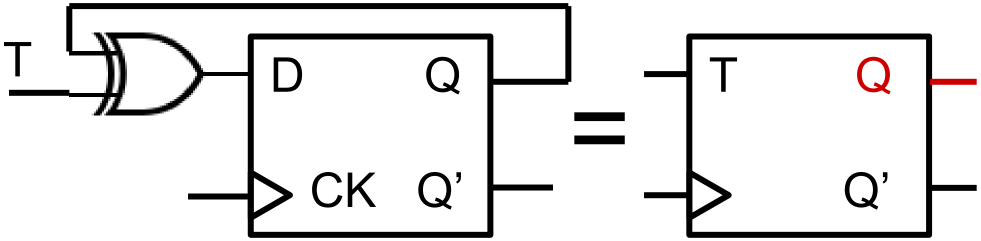
- Last week we discussed D flip-flops
  - Circuits which sample the value at their D input as an active edge is presented to their *clock* input port

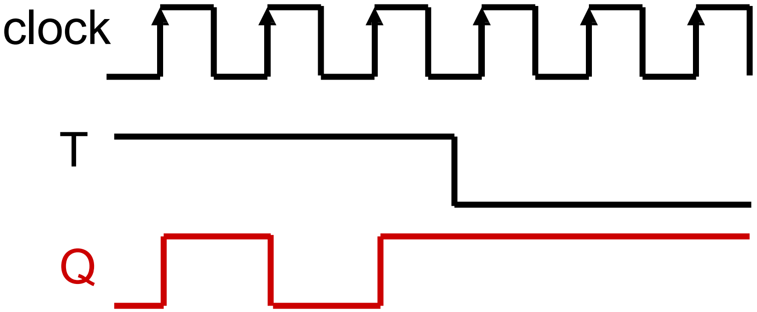
- D flip-flops are the type used in FPGA hardware

- Today we start off with some basic but essential and widespread designs which use flip-flops
  - Today we will only treat seemingly simple designs.
  - Bear in mind that sequential designs need not be much more difficult than this to design. Complexity is usually created through the interaction of many simple sub-units.

The “T” flip flop

- I promised that we would only learn D flip flops. This is not really a different type, just a variation.
- T stands for **Toggle**.
- If T is high, this design switches between logic values each time an active edge appears at the clock.
- If T is low, the flip-flop holds the previous value.

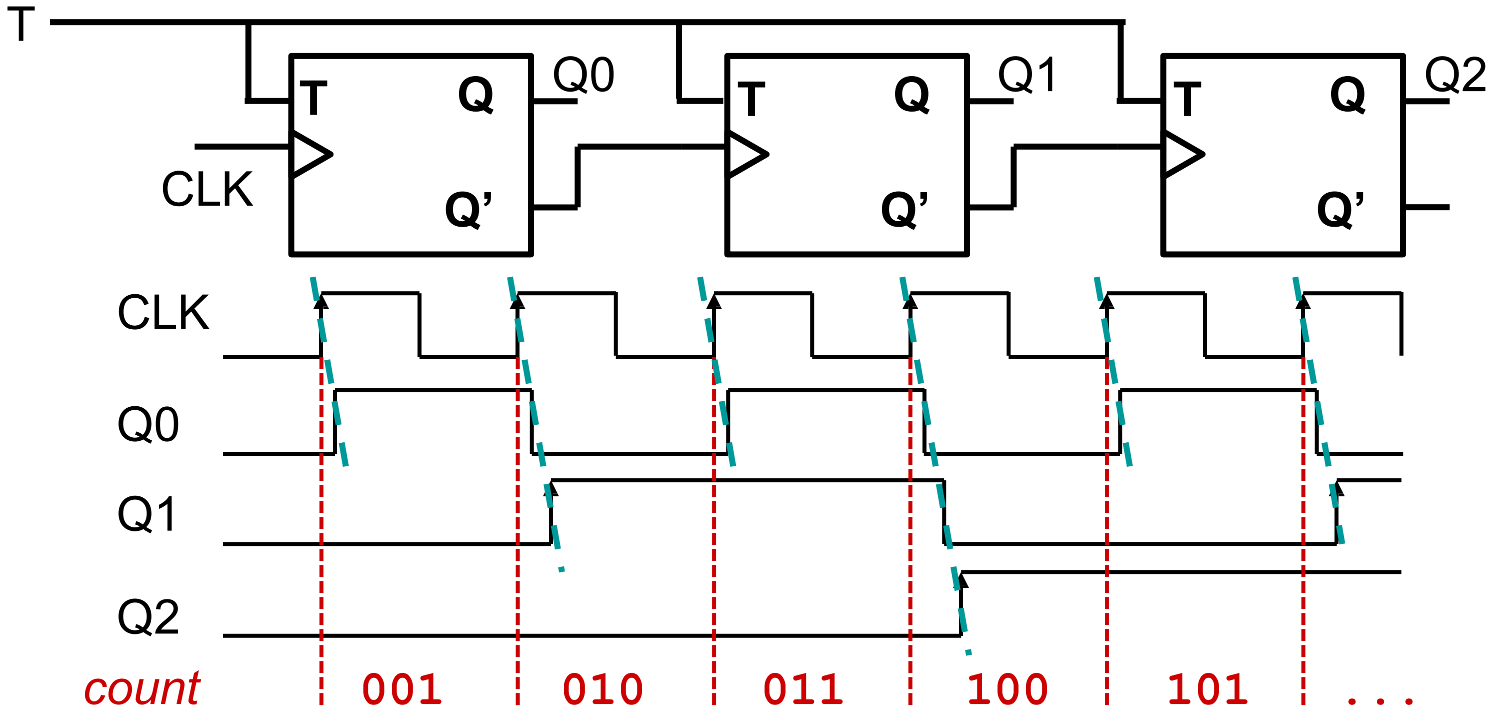




- This example, incidentally, shows that **feedback is allowed** in sequential circuits, can you explain why?

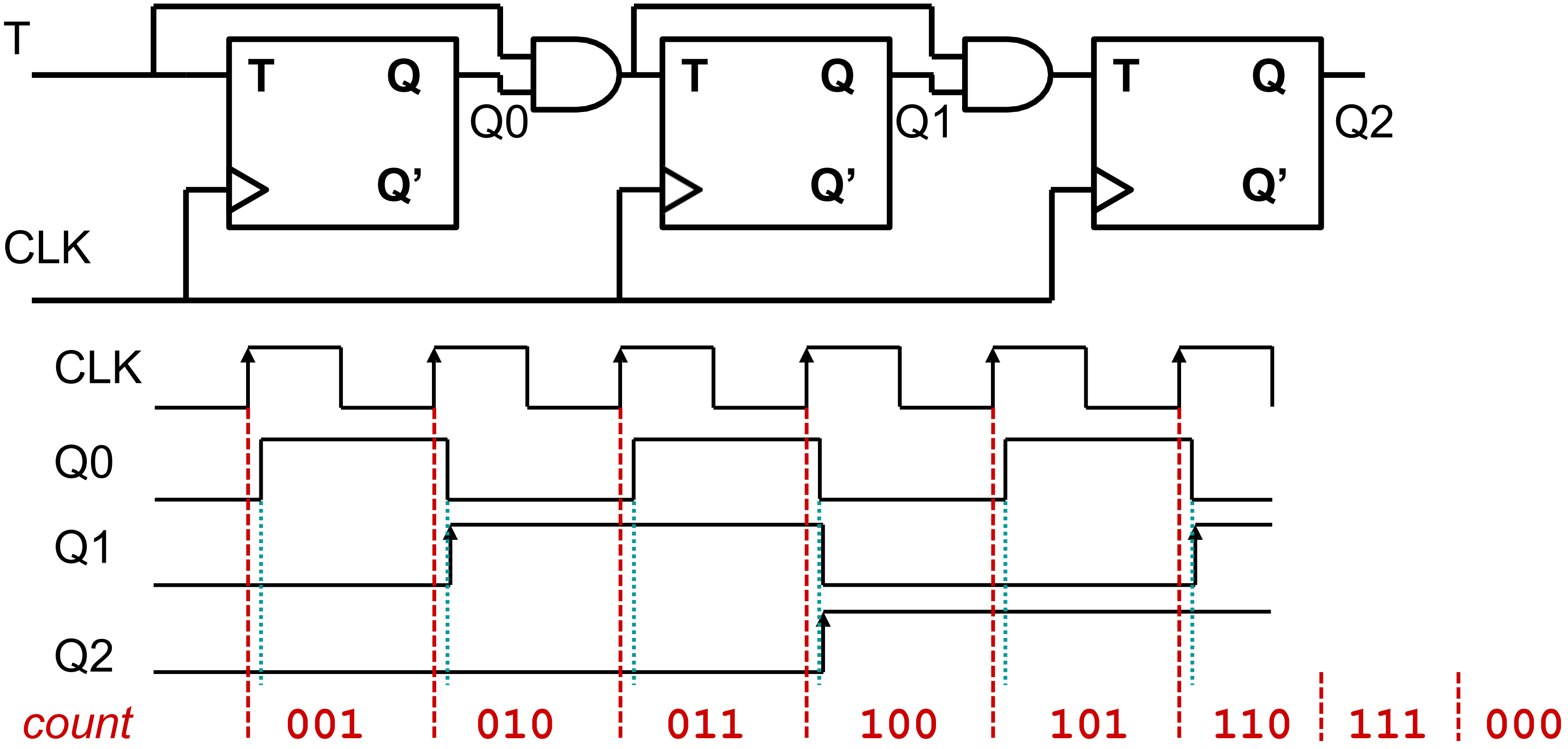
Asynchronous counter (“ripple” counter)

- **Counter:** a circuit which returns the number of active edges observed at the input



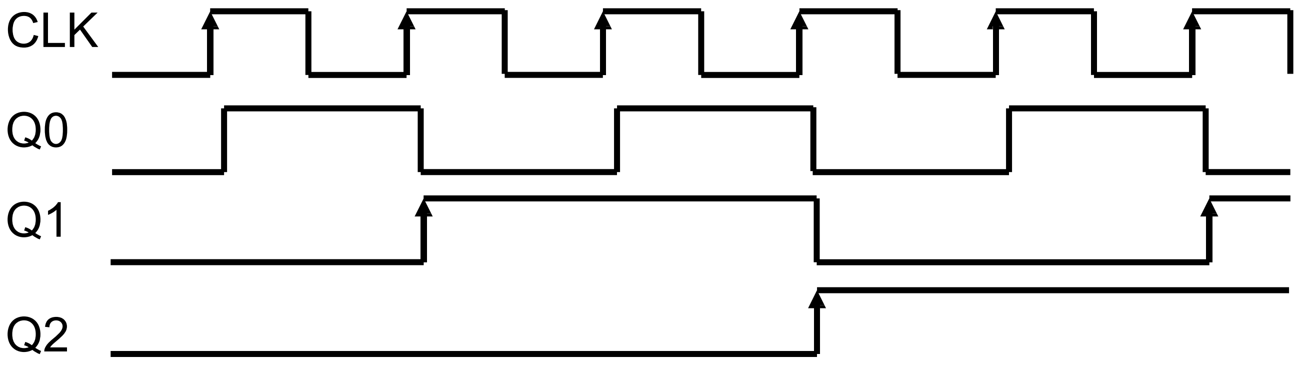
- ***Asynchronous:*** the update of the count bits is not simultaneous with the input clock (due to a “ripple” effect). This can be undesirable in some applications.

Synchronous counter



- ***Synchronous***: all flip-flops in the chain switch at the same time.

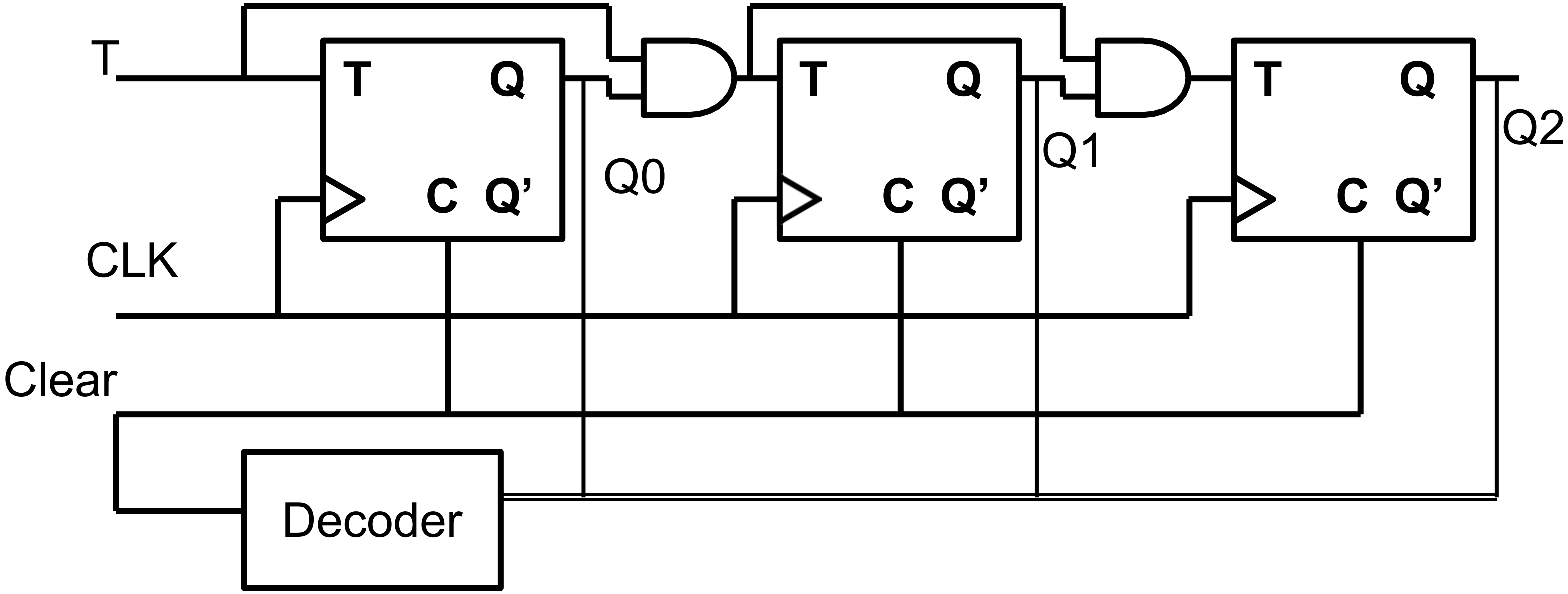
Counters as clock dividers



- A binary counter can be used as a 2^n ***clock divider***. You can see that the Q0 output toggles at ½ the frequency of CLK, Q1 at ¼, etc.
  - Sometimes called *modulo-m counters* as they count up to *m*
- We will use this in some applications where we want the system (or parts of it) to run at slower clock speeds.

- Class exercise. How could we design a counter that counts up to an arbitrary number? (not a power of 2)

Arbitrary *m* up-counter

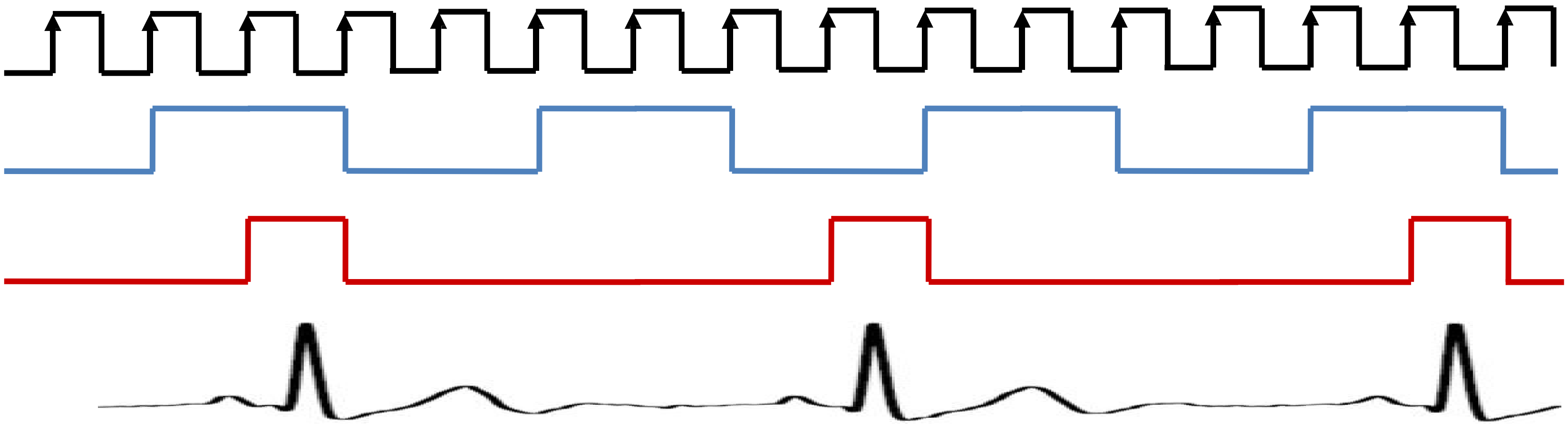


- The decoder associates the binary sequence for *m* with a 1 output.

- Instead of connecting the equality result to the counter reset, it can be used it to activate some other circuit event. Such use is called an ***event sequencer***.

The heartbeat generator

- A system that gives a 1 output for 1 clock period in duration every *n* clock cycles (arbitrary *n*)
  - Compare below: a clock, clockdivider and heartbeat generator output (and the beat of a real heart).

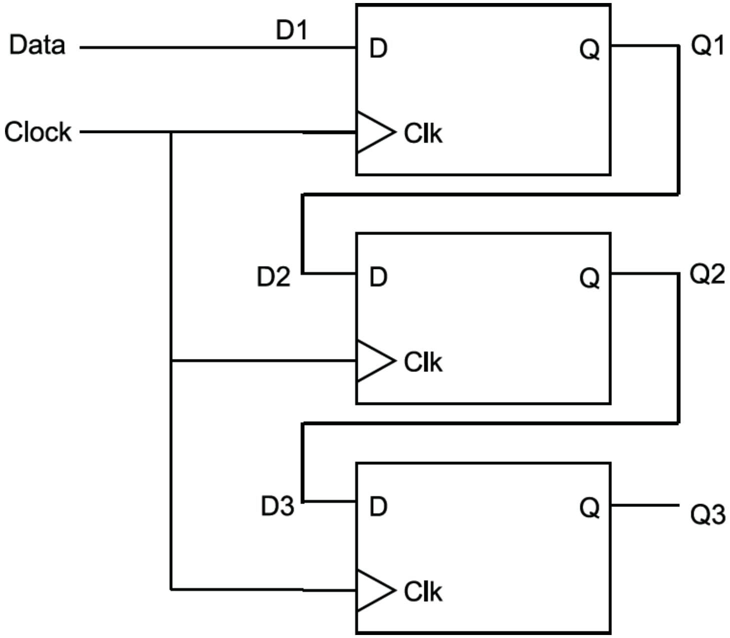


- How would you modify the sequential design shown on the previous slide to achieve this behaviour?

Shift registers

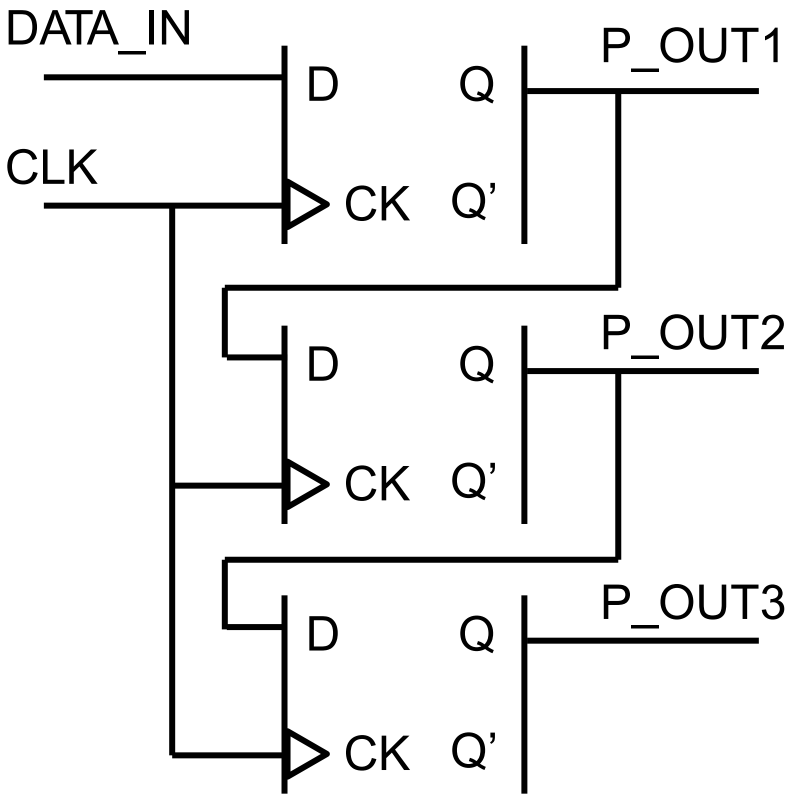
- An assembly of *n* flip-flops with a provision for shifting data between them. E.g.,

- At each active clock edge, data is shifted from a FF to the next
  - The one on the right is called a ***serial-in, serial-out*** shift register. Data words are read at the input and produced at the output one bit at a time.



Shift registers (2)

- As well as serial-in serial-out (SISO) register, there are also others where the data word is read in or out with all bits at once (***parallel***).
  - Parallel operation is faster but requires more wires.
  - The circuit on the right can be used either as SISO or as SIPO (serial-in, parallel-out).
  - What would PIPO and PISO look like?



Shift registers (3)

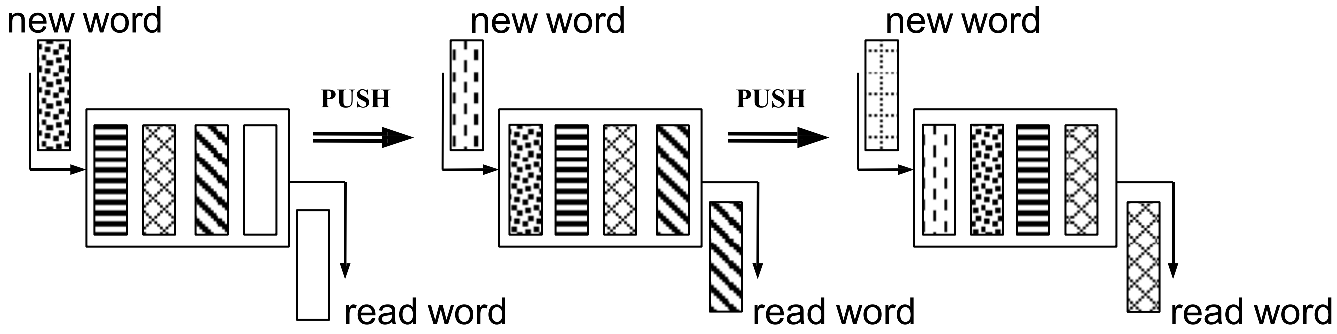
- Another distinction between registers is given by the possible shift directions of data.
  - In *first-in, first-out* (FIFO) registers, data flows through in a queue fashion
  - In *last-in, first-out* (LIFO) registers, you can imagine data as stacked in a pile, with the last added item being on top

  - You will encounter FIFO and LIFO registers in LAB3

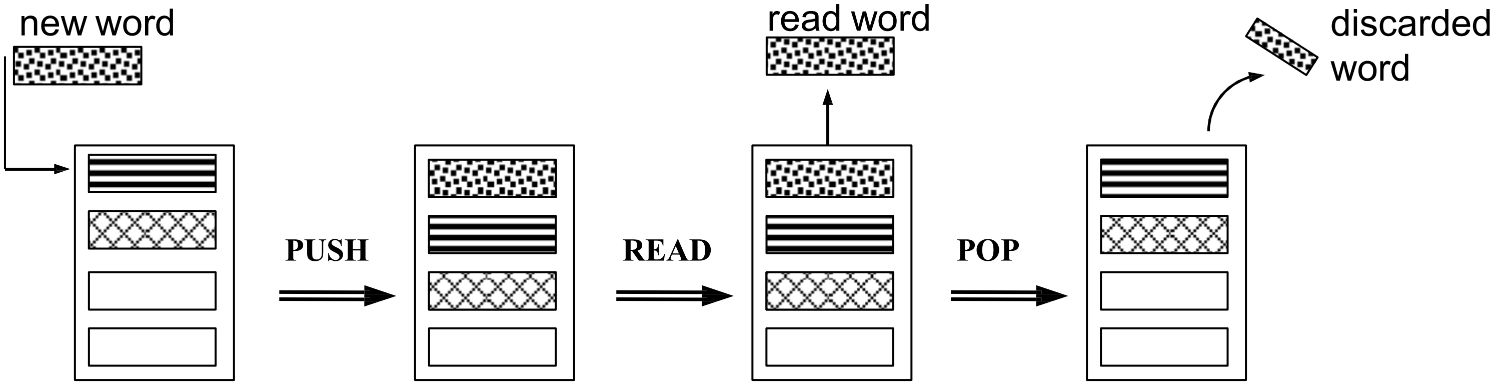
- Exercise before class... Given one of the types above, could you design the relevant register?

FIFO and LIFO

- **FIFO**



- **LIFO**



Summing up

- We have seen and discussed some basic uses of flip-flops
  - Counters
  - Clock dividers
  - Shift registers

- We have seen that **feedback is admissible** in sequential circuits and in fact it can be rather useful.

- We have again encountered matters of **synchrony and timing**. Synchrony errors are by far the most common issue with designs so I will insist on timing-related concepts over and over throughout the course.