CS 261
Fall 2017

Mike Lam, Professor

Sequential Circuits

Circuits

- Circuits are formed by linking gates together
  - Inputs and outputs
    - Link output of one gate to input of another
    - Some gates have multiple inputs and/or outputs
  - Combinational circuits: outputs are a boolean function of inputs
    - Not time-dependent
    - Used for computation
  - Sequential circuits: output is dependent on previous outputs
    - Time-dependent
    - Used for memory

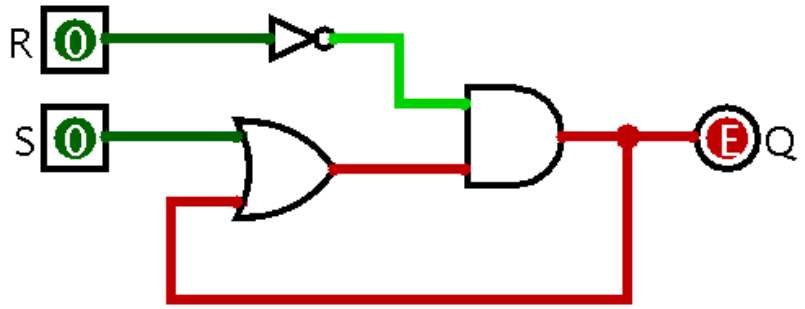
Circuit memory

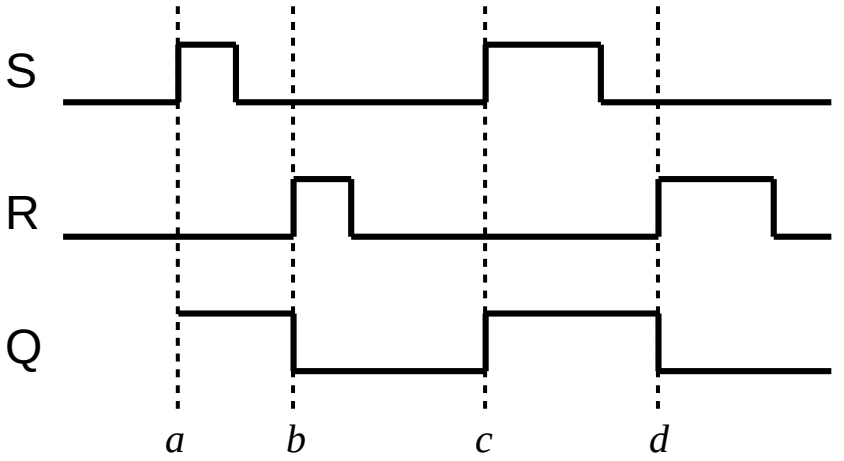
- Question: How do we make a circuit “remember” something?

Circuit memory

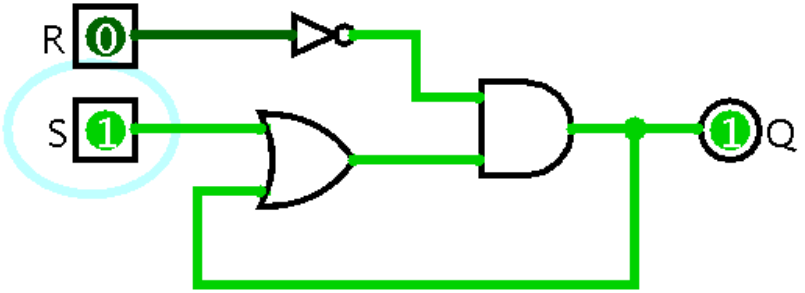
- Question: How do we make a circuit “remember” something?
  - Answer: Create a feedback loop!
  - Creates a “storage” circuit, often called a latch
  - Truth table must include previous state
  - Alternatively, draw a timing diagram
    - Shows how input/output signals change with respect to time
    - Given input signals in diagram, we can determine output signals

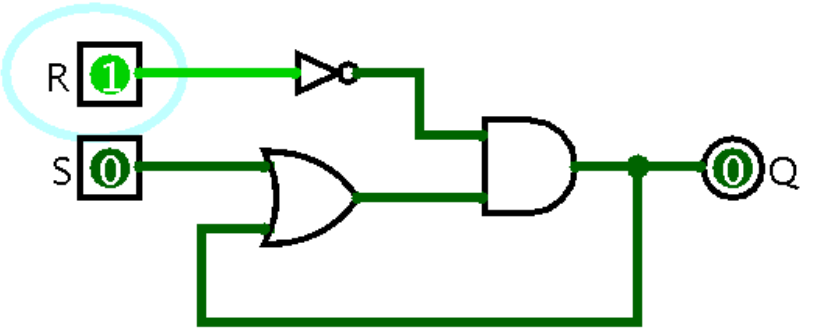
SR AND-OR latch

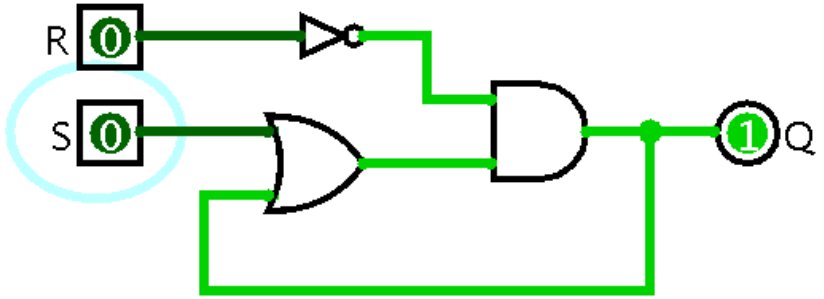


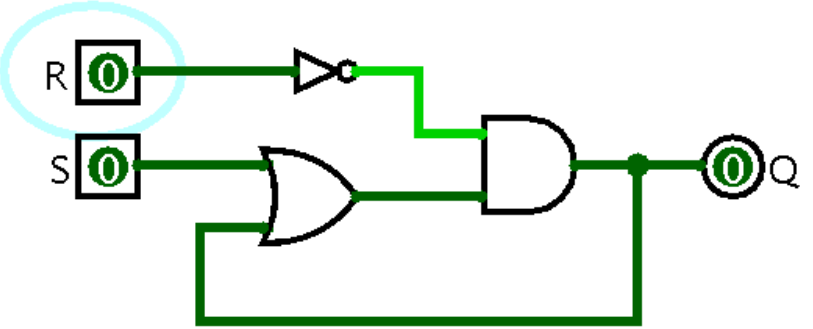


S = “set” R = “reset”

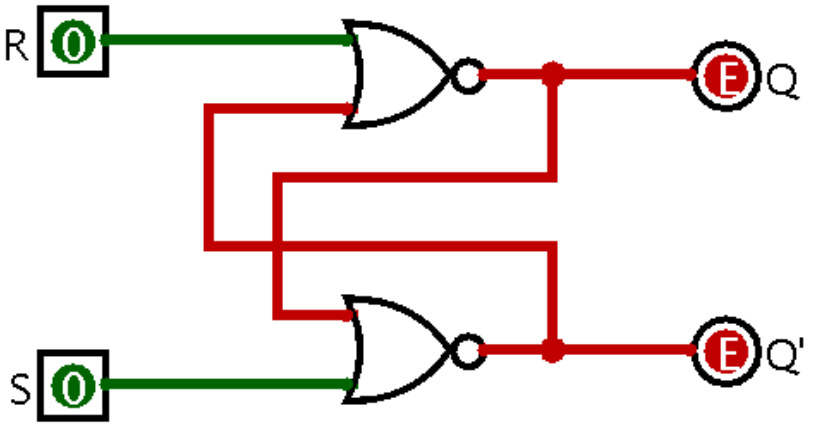


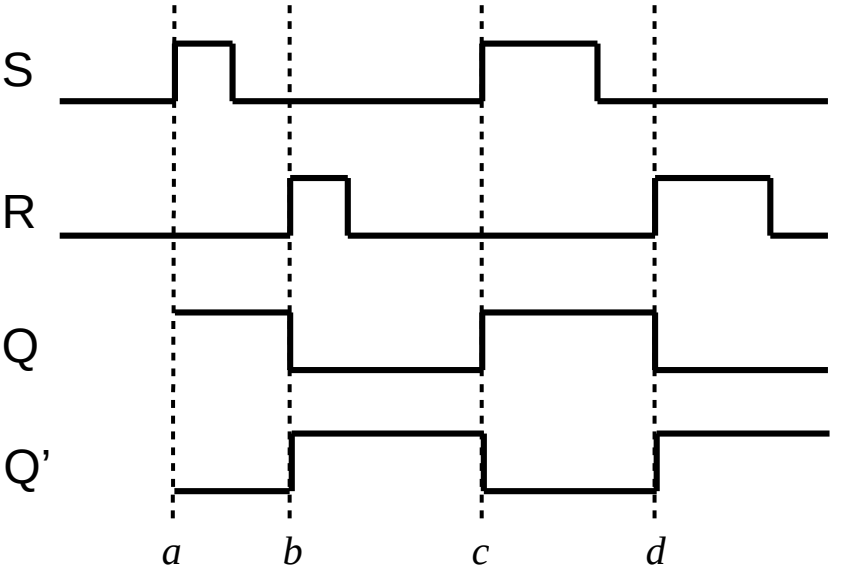


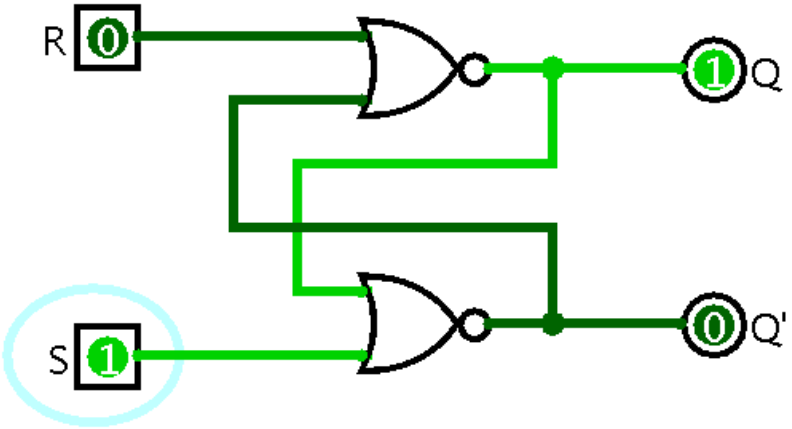


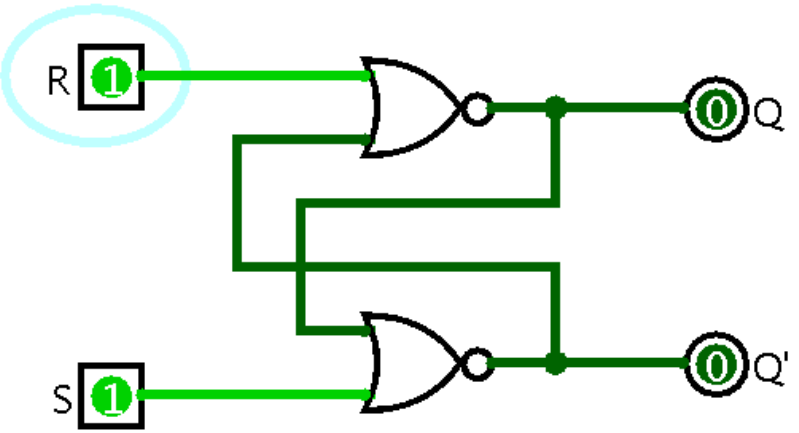


SR NOR latch



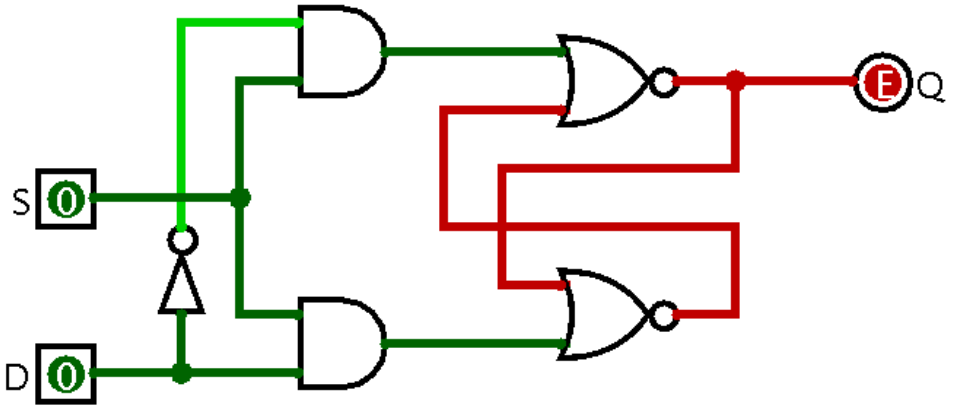


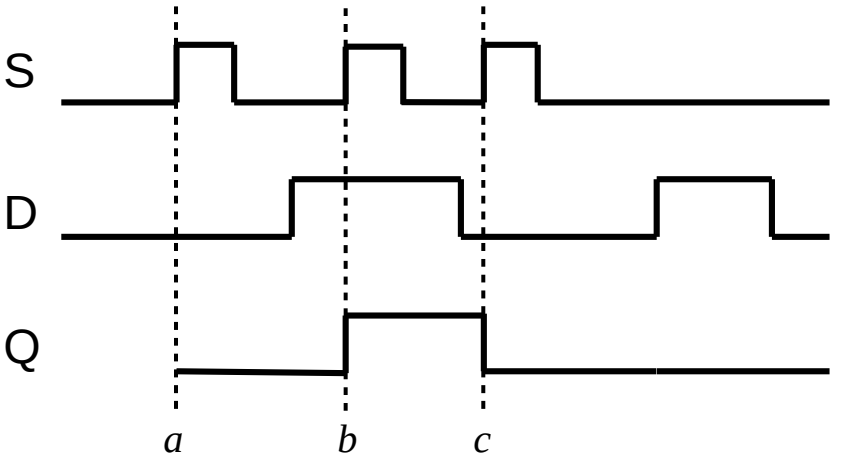




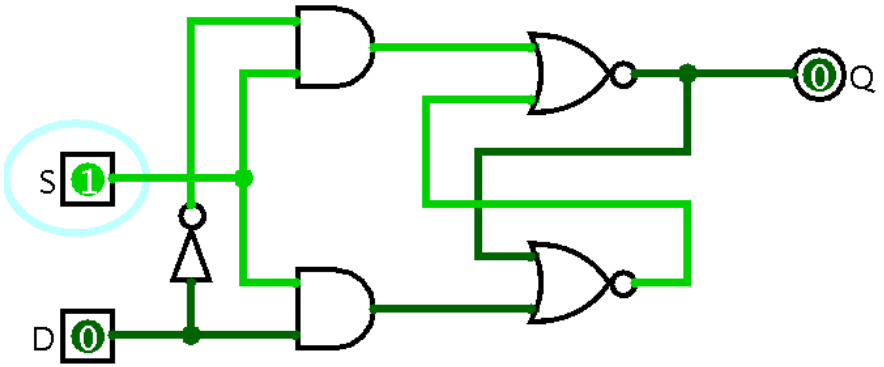
Disallow S=1, R=1 because Q’ ≠ !Q

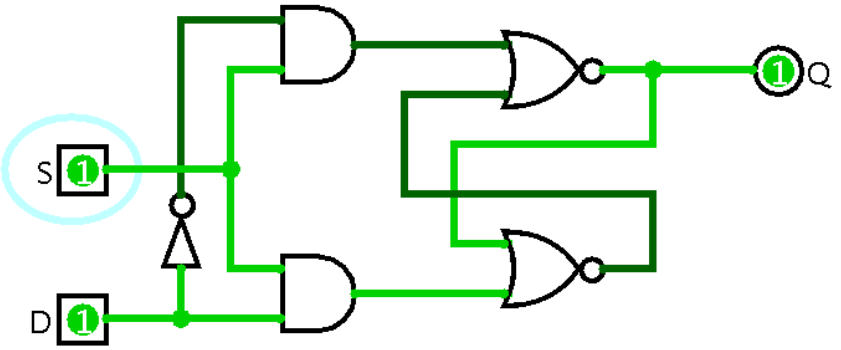
D latch





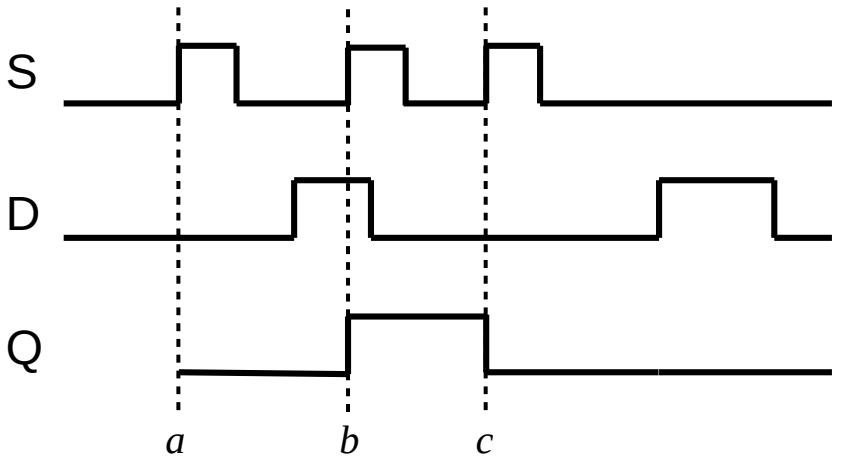
From “Code” book: S = “Save that bit!”



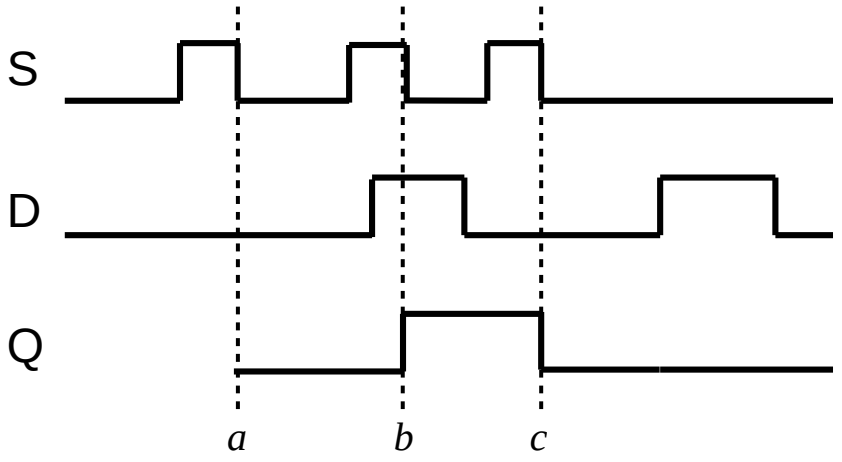


Signal changes

- The original D latch reflects D input on Q as long as “set” is on
- Edge-triggered latches change Q on rising edge of “set” signal
- Master-slave latches change Q on falling edge of “set” signal



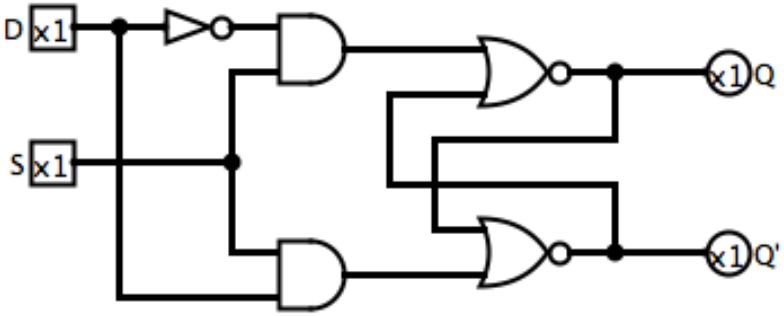
Edge-triggered D latch

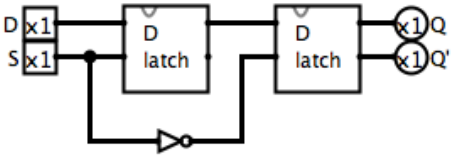


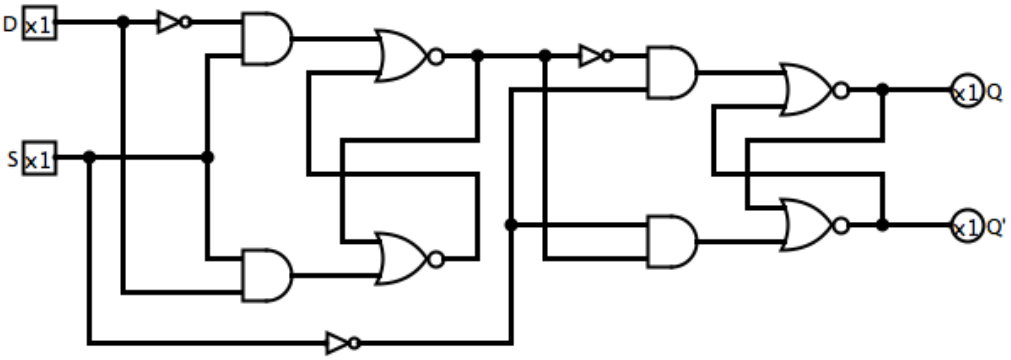
Master-slave D latch

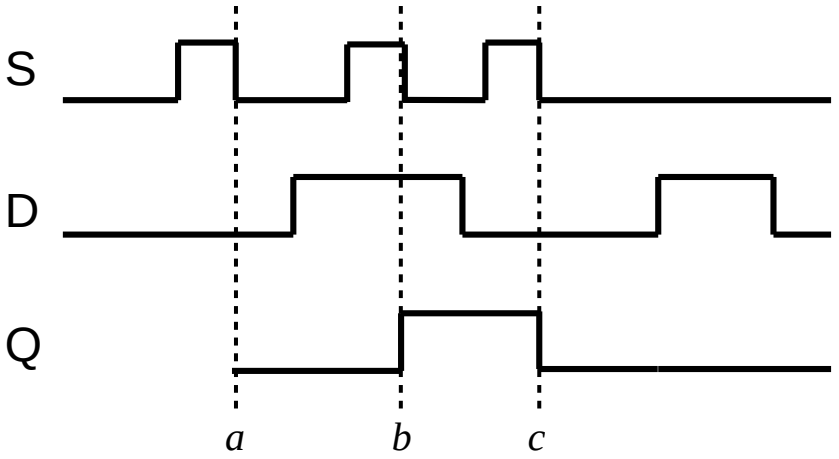
Master-slave D latch

Original D latch:



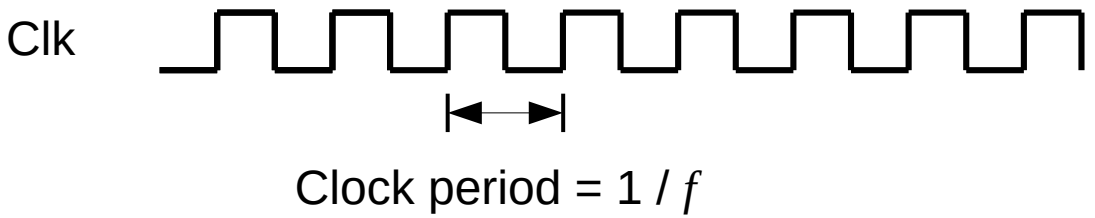






Clocks

- Provide oscillating signal
- Often used as “set” signal for latches
- Keeps computation and memory in sync
- Clocked latches are called flip-flops
- The clock period is the inverse of the frequency (measured in *hertz*)
- The length of a clock period determines the minimum time an instruction takes to execute

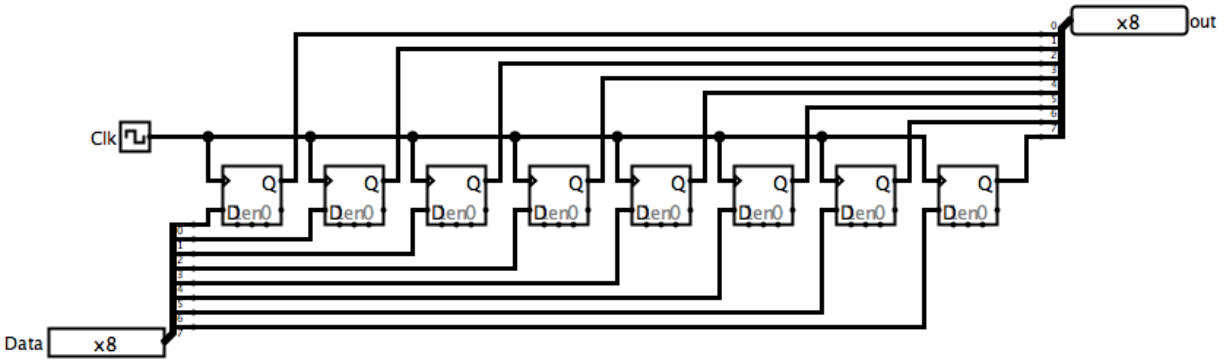


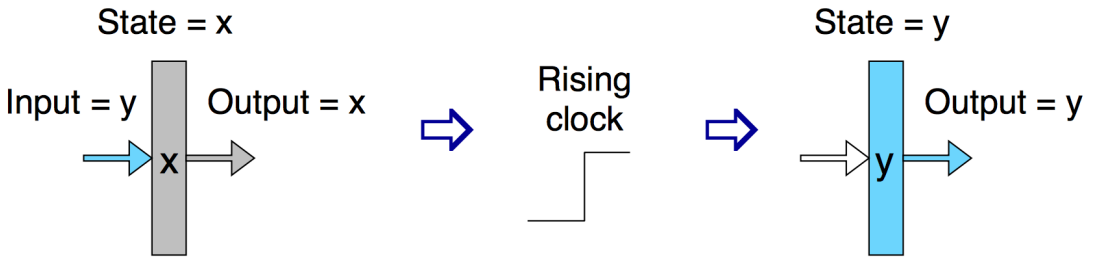
Flip-flop types

- SR: “set-reset”
- D: “data” bit + clock
- T: “toggle”
- JK: like SR + T (toggle when S=1, R=1)
  - J is S, K is R
- Any of these can be used to build the others
- Also can be built from basic logic gates in multiple ways

Registers

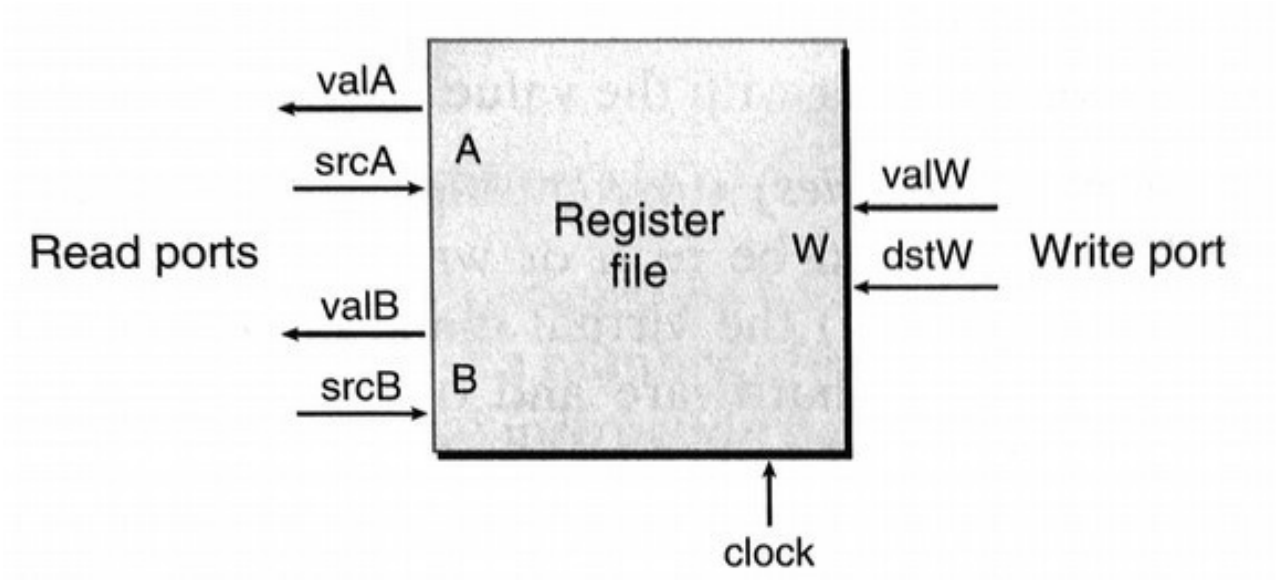
- Registers: arrays of flip-flops with a single set/clock input
- Connected by buses (groups of wires) to other components
- Edge triggering allows computation to stabilize before results are saved
- Caveat: difference between hardware registers and program registers
  - Former are physical, latter are logical (and stored in a register file)





Register files

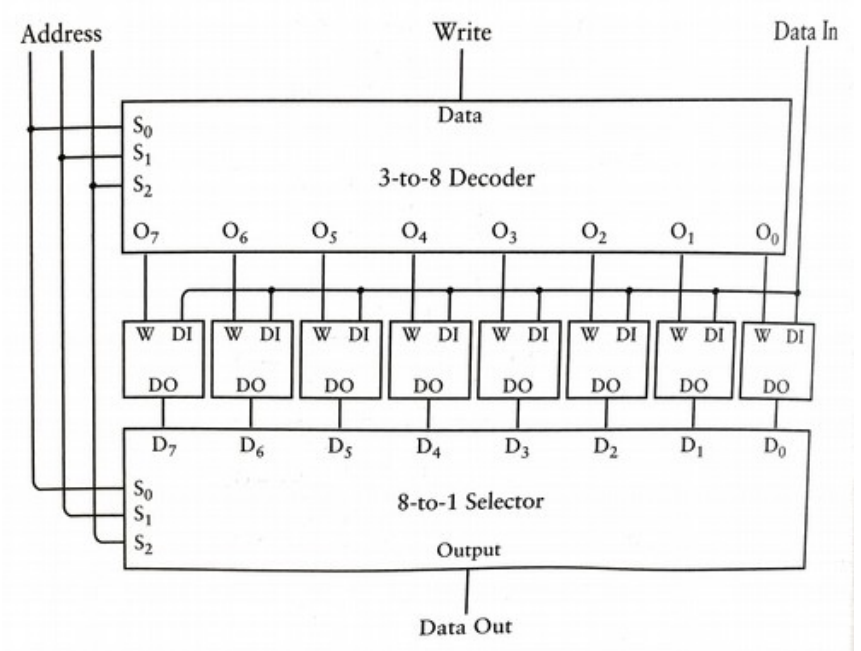
- Register files: multiple registers w/ selector inputs
  - Use multiplexors to differentiate



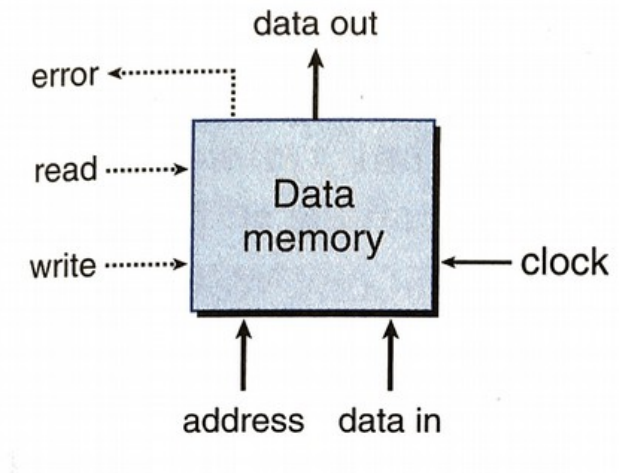
Use multiplexors to read/write from the appropriate register using srcA, srcB, or dstW signals (set dstW to 0xF to read only)

Memory

- Memory: multiple flip-flops w/ address input
  - Random access memory (RAM) - can access any address at any time
  - Use decoder (translates 3-bit number to 8 “set” signals) to write data
  - Use selector (multiplexor) to read data



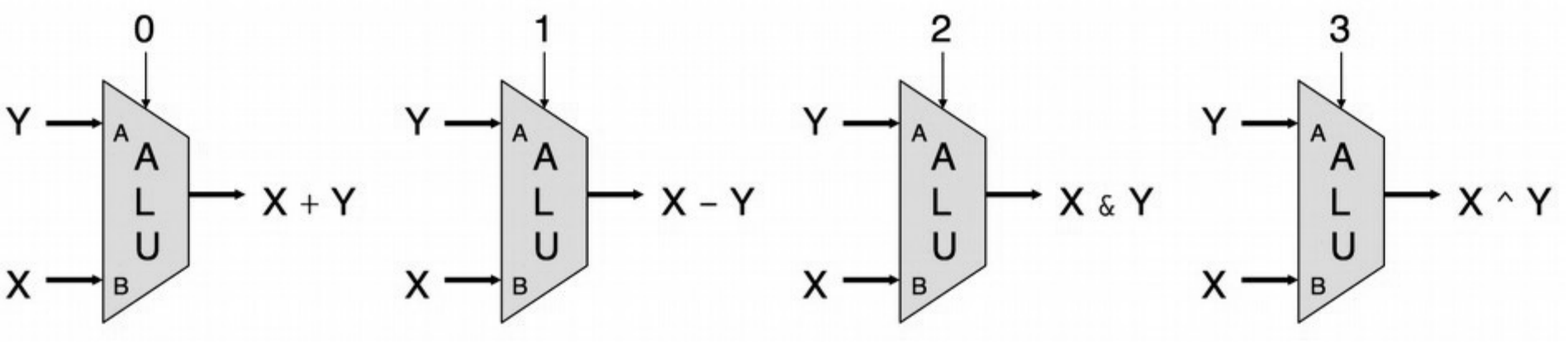
Single RAM array



Abstraction of multiple RAM arrays

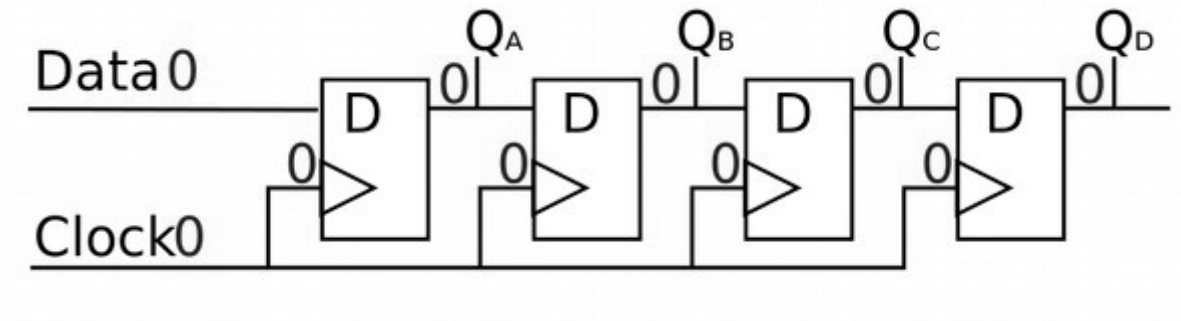
ALUs and memory

- Combine adders and multiplexors to make arithmetic/logic units
- Combine flip-flops to make register files and main memory



Basic Arithmetic Logic Unit (ALU)

4-bit Register



CPUs

- Combine ALU with registers and memory to make CPUs