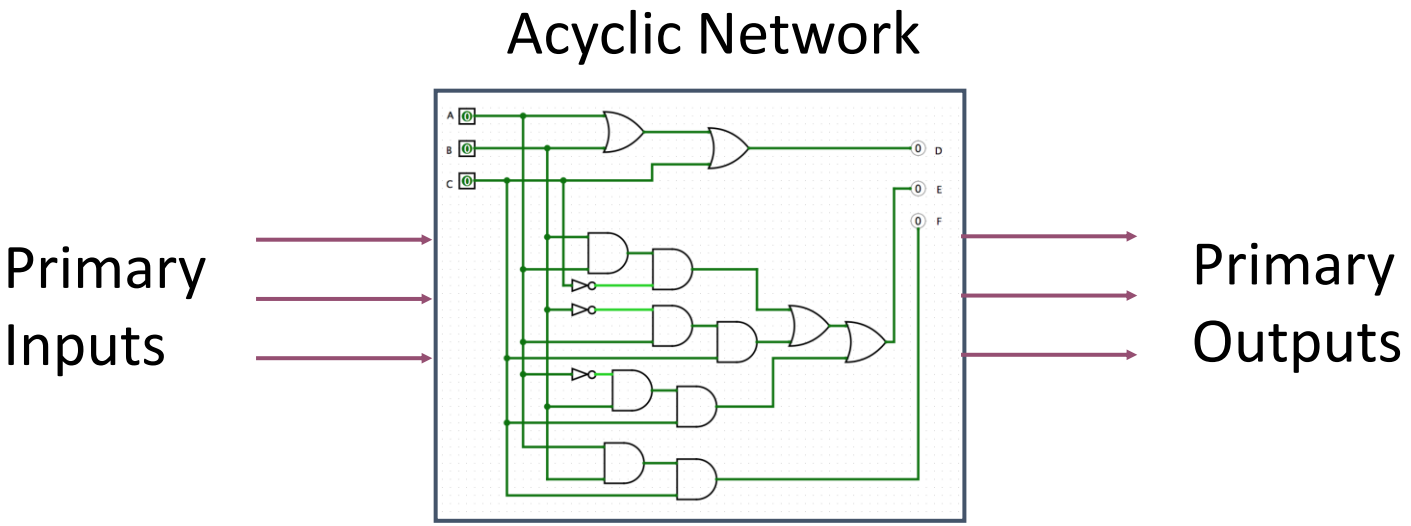
Sequential Logic

CMPU 224 – Computer Organization
Jason Waterman

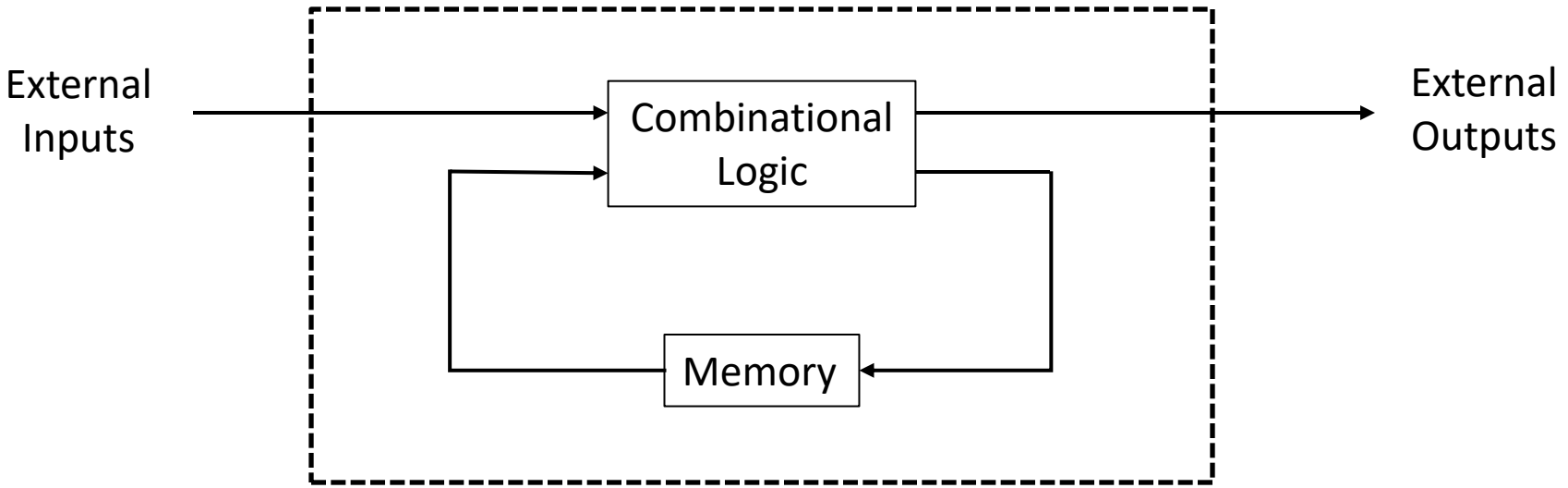
Sequential Logic Intro

- Last time: **Combinational Circuits**
  - Acyclic Network of Logic Gates
  - Continuously responds to changes on primary inputs
  - Primary outputs become (after some delay) Boolean functions of primary inputs

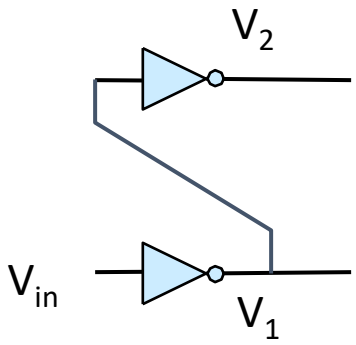


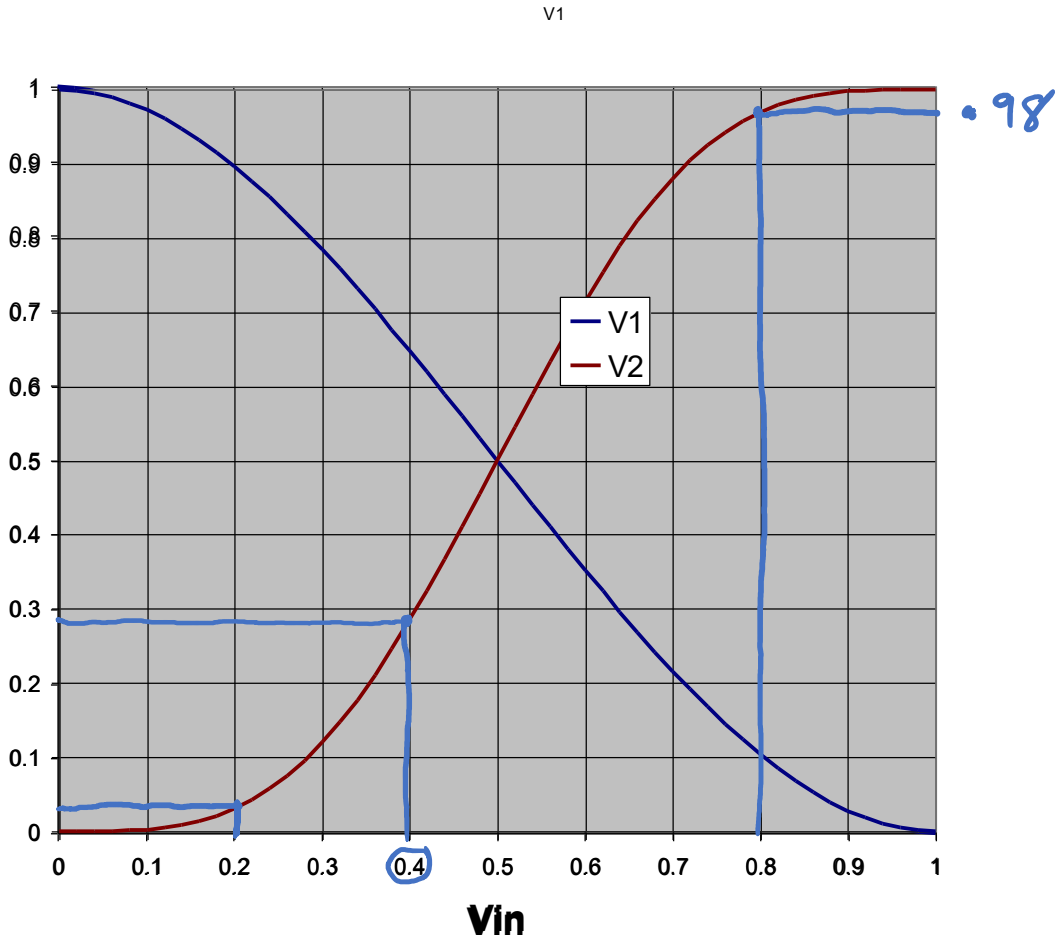
Sequential Logic

- Sequential logic circuits are those whose outputs are also dependent upon past inputs
- In other words, the output of a sequential circuit may depend upon its previous outputs
- In effect, it has some form of "memory"

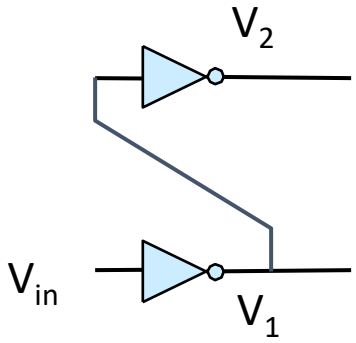


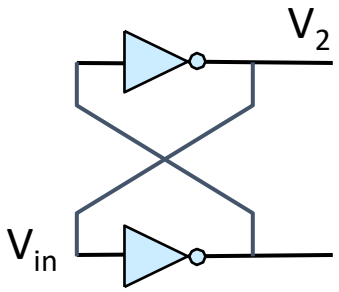
Storing 1 Bit

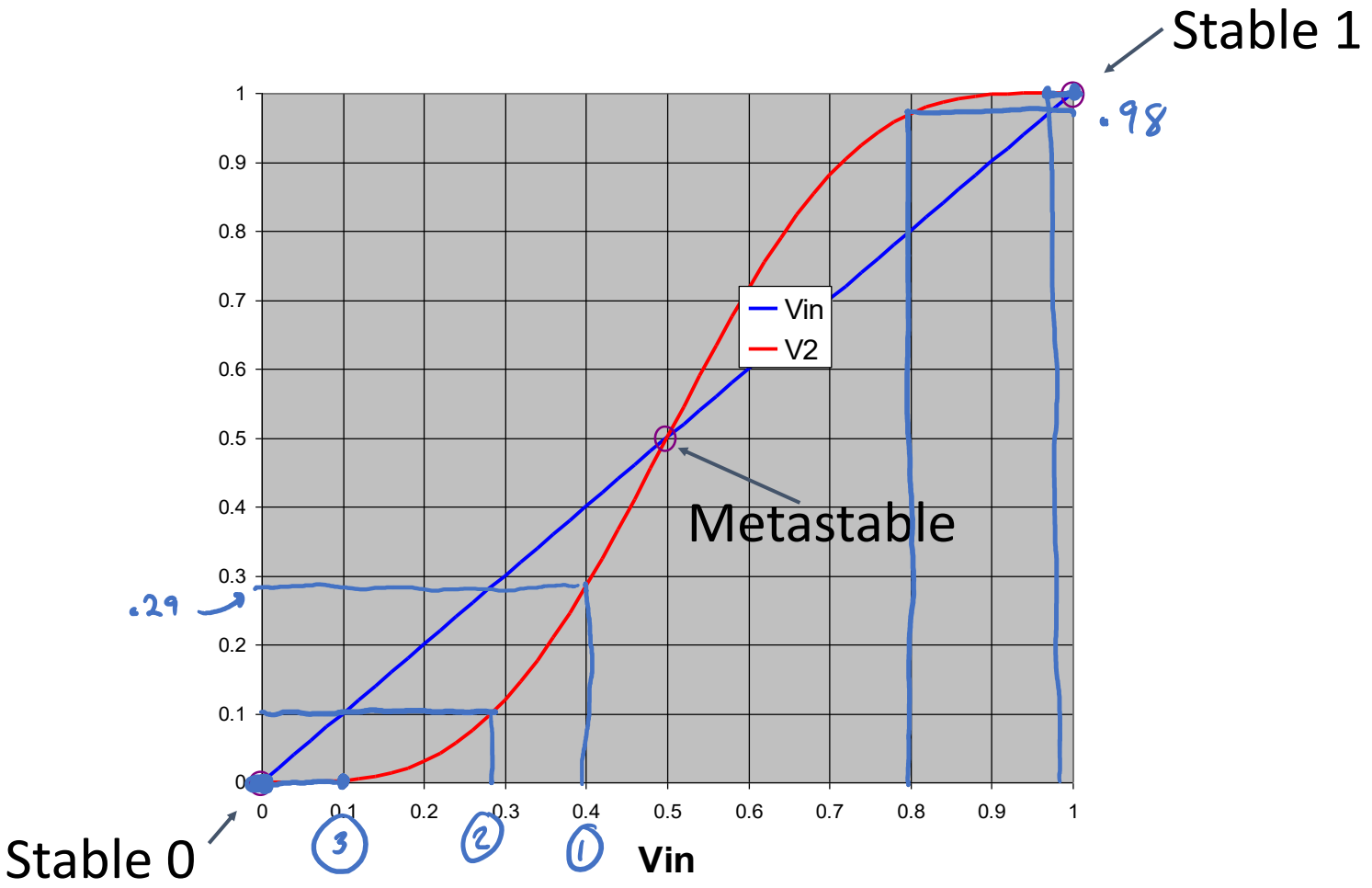




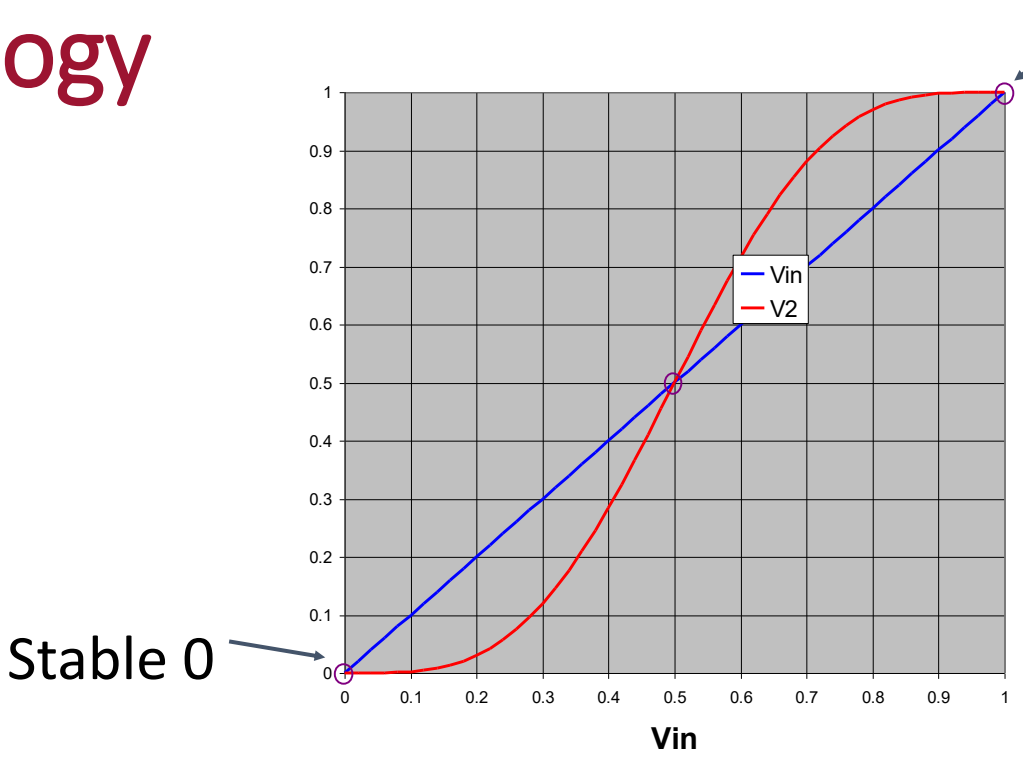
Storing 1 Bit

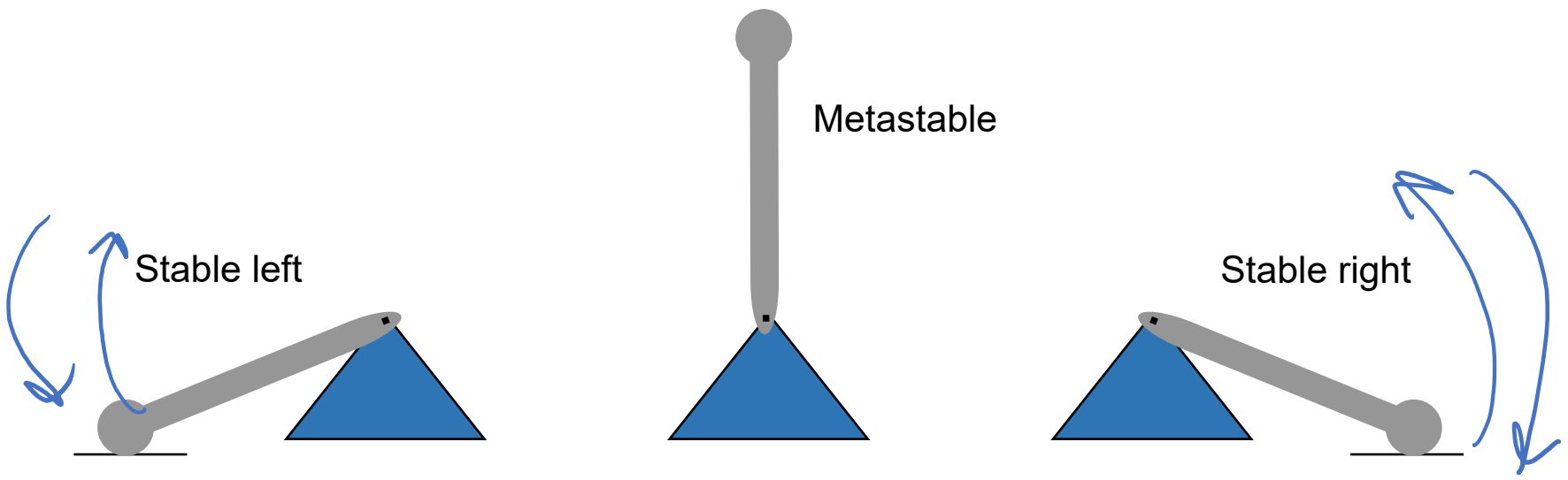




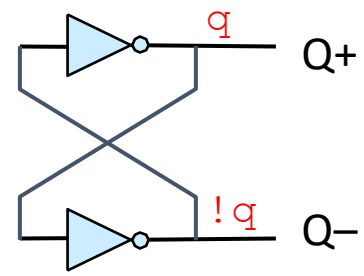


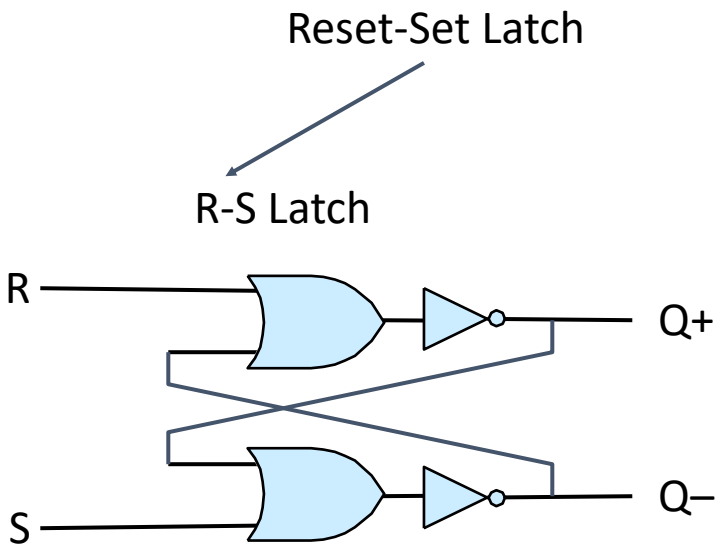
Physical Analogy

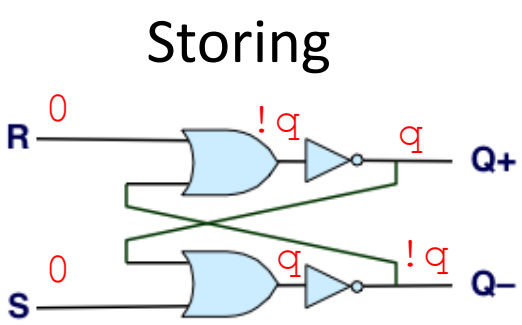


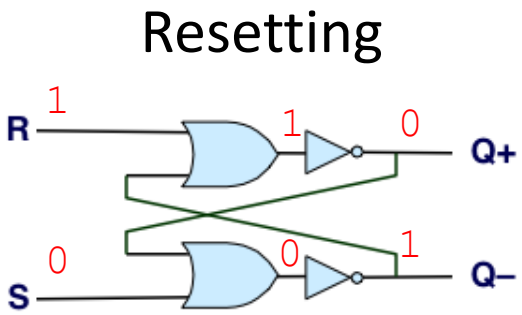


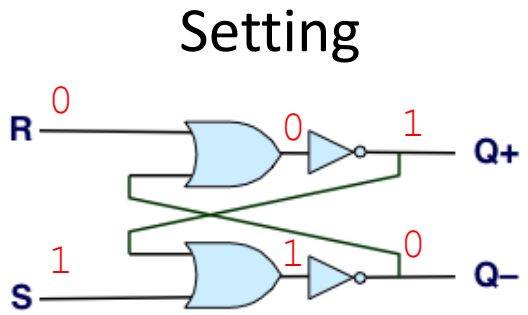
Storing and Accessing 1 Bit



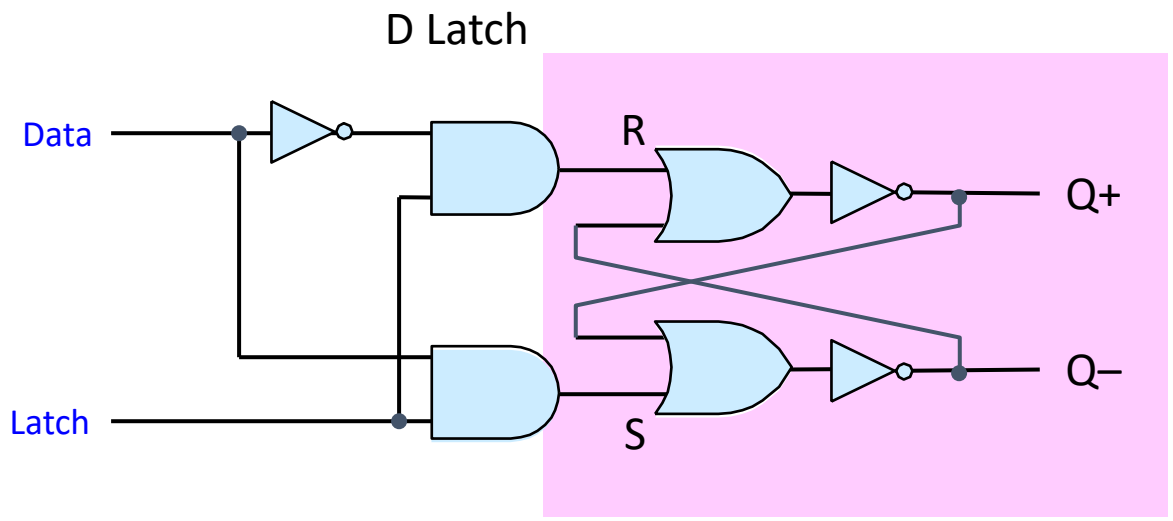




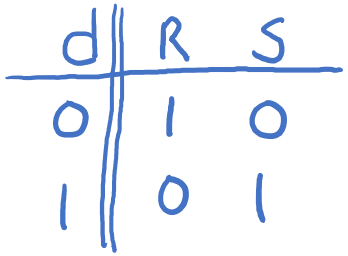




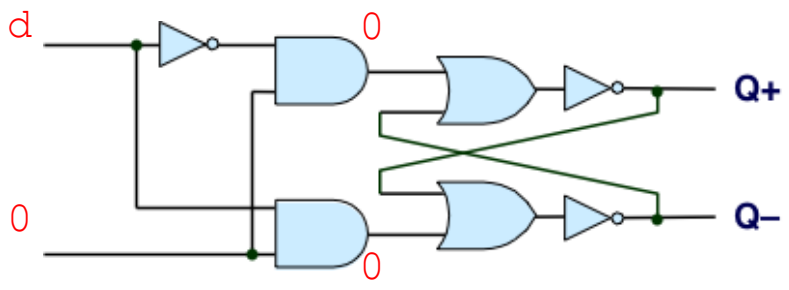
1-Bit Latch



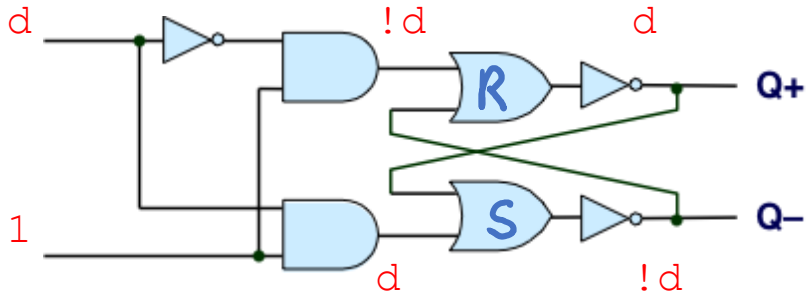
R	S	State
1	0	Reset to 0
0	1	Set to 1
0	0	Storing



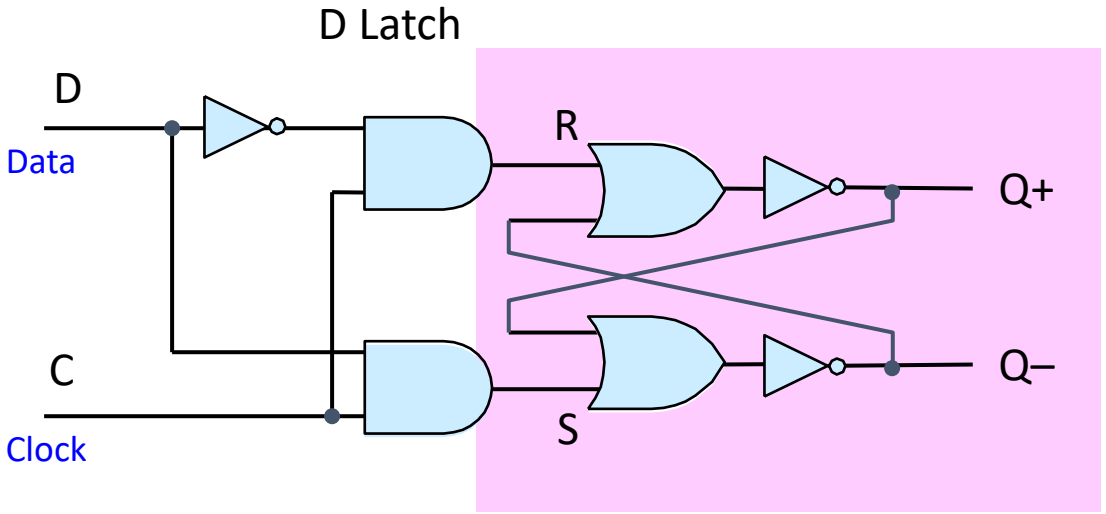
Storing



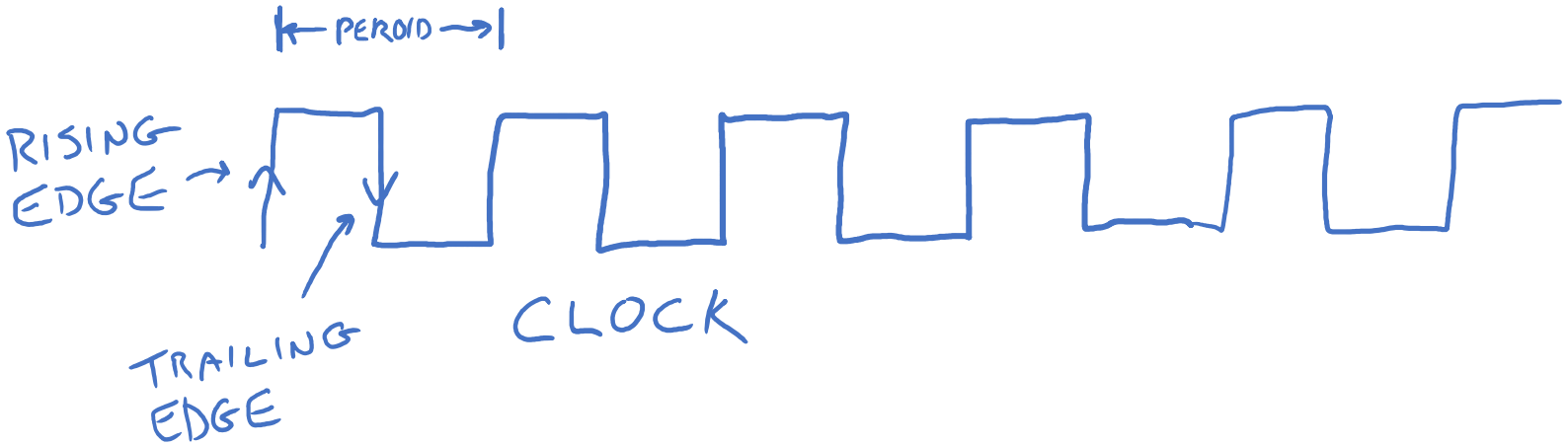
Latching



Clocking



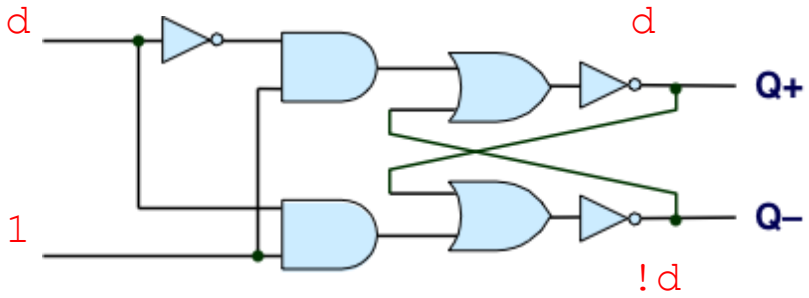
R	S	State
1	0	Reset to 0
0	1	Set to 1
0	0	Storing



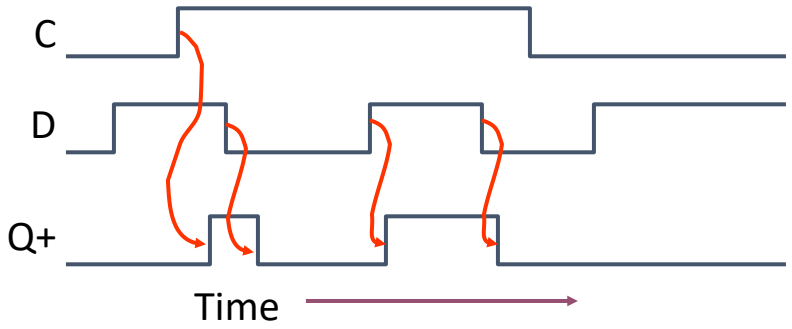
Transparent 1-Bit Latch

- When in latching mode, combinational propagation from D to Q+ and Q–
- Value latched depends on value of D as C falls

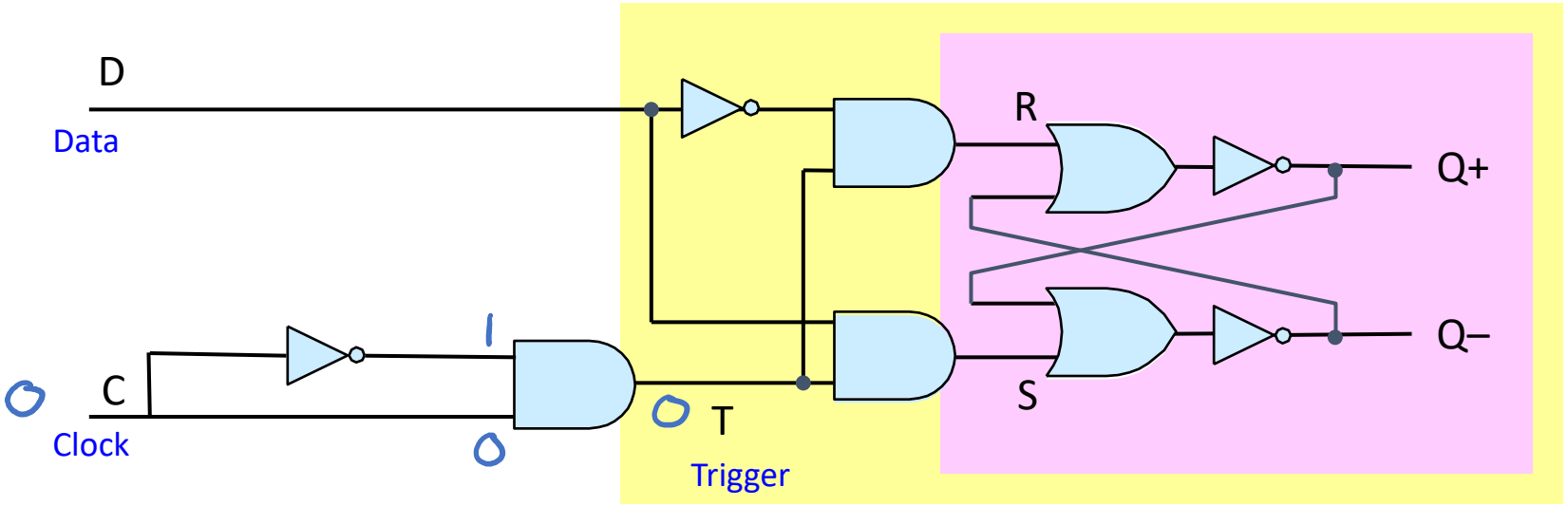
Latching



Changing D



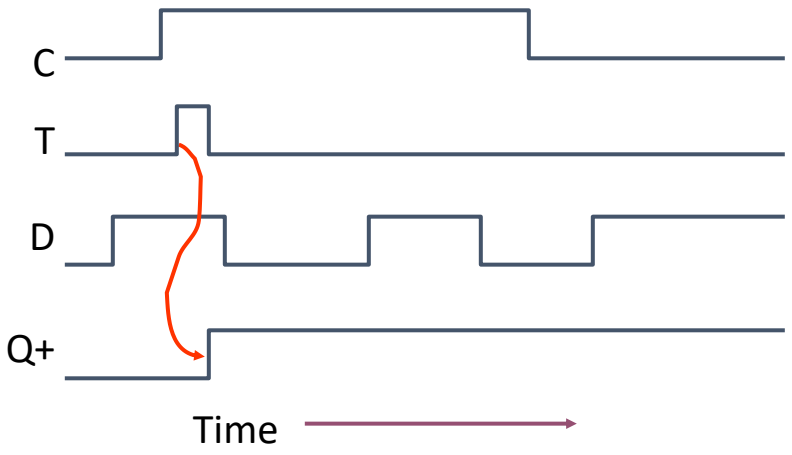
Edge-Triggered Latch



CLOCK LOW

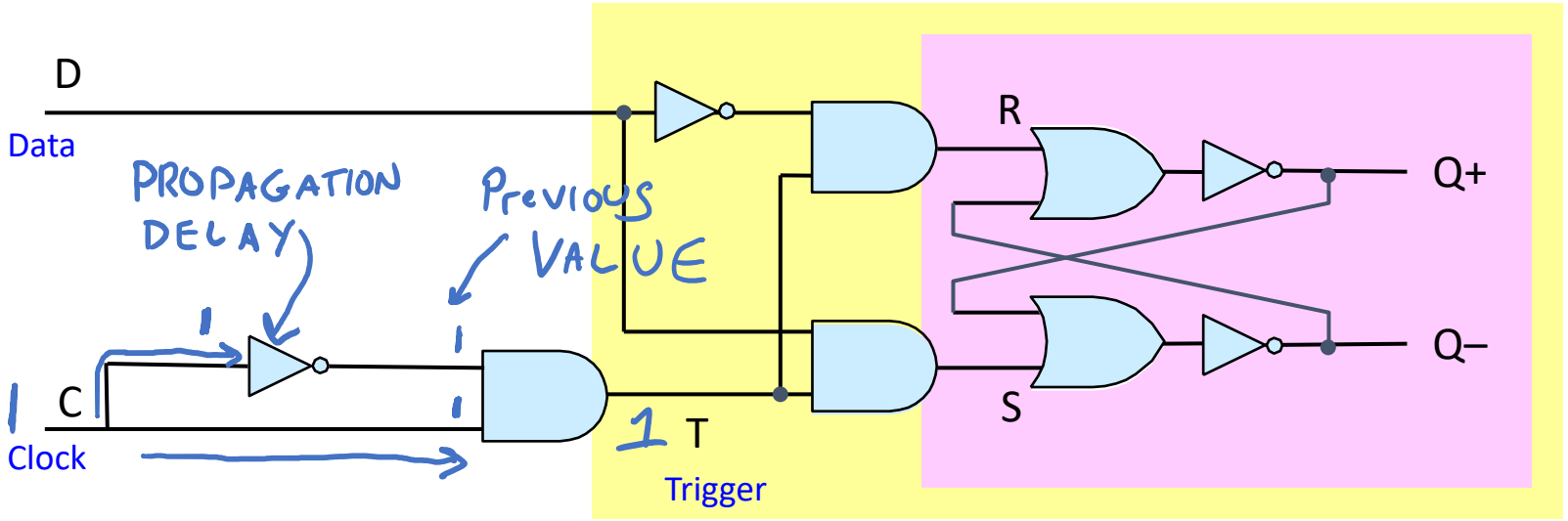
STORAGE MODE

INPUT D IS IGNORED



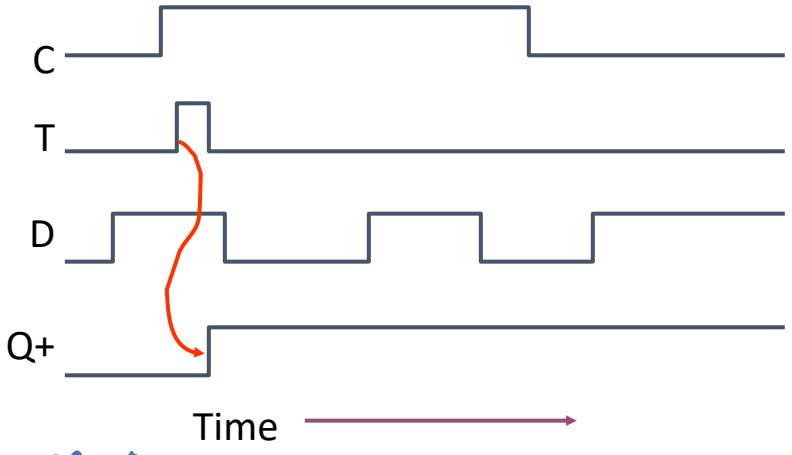
- Only in latching mode for brief period
  - Rising clock edge
- Value latched depends on the data as the clock rises
- Output remains stable at all other times

Edge-Triggered Latch



RISING CLOCK EDGE

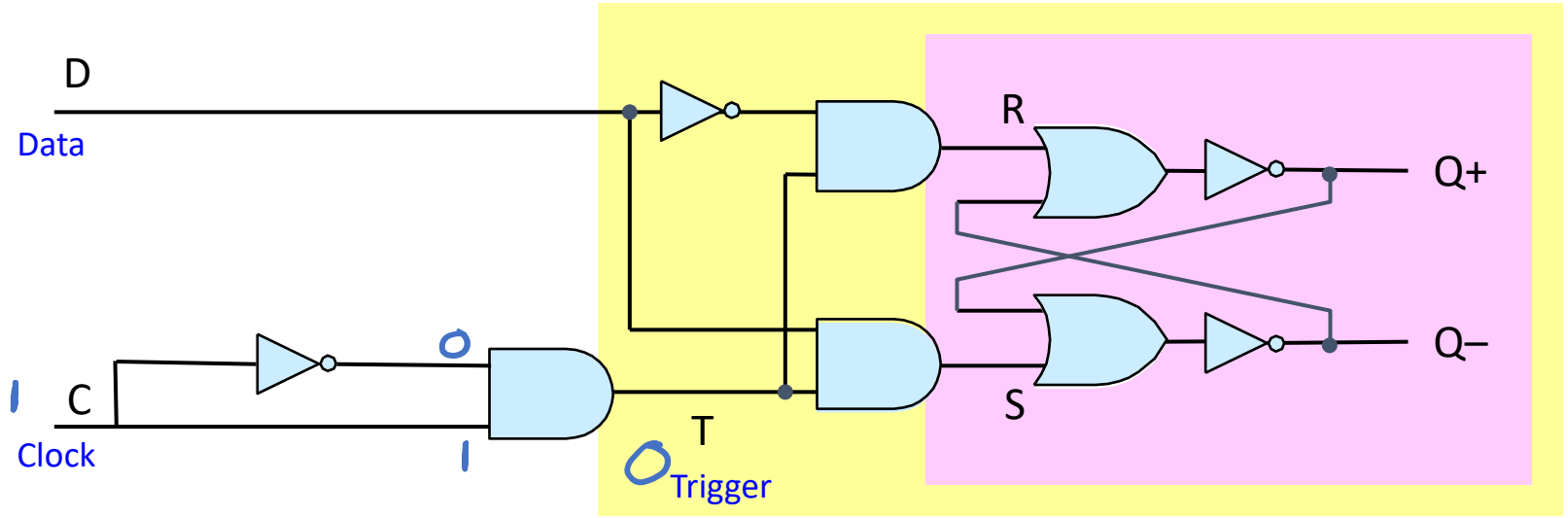
0 → 1



TRIGGER IS BRIEFLY 1: LATCHING MODE

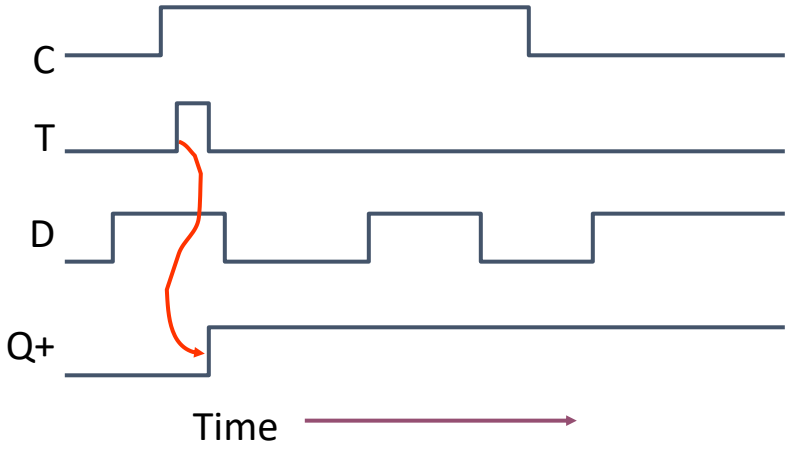
- Only in latching mode for brief period
  - Rising clock edge
- Value latched depends on the data as the clock rises
- Output remains stable at all other times

Edge-Triggered Latch



CLOCK HIGH

INPUT D IS IGNORED

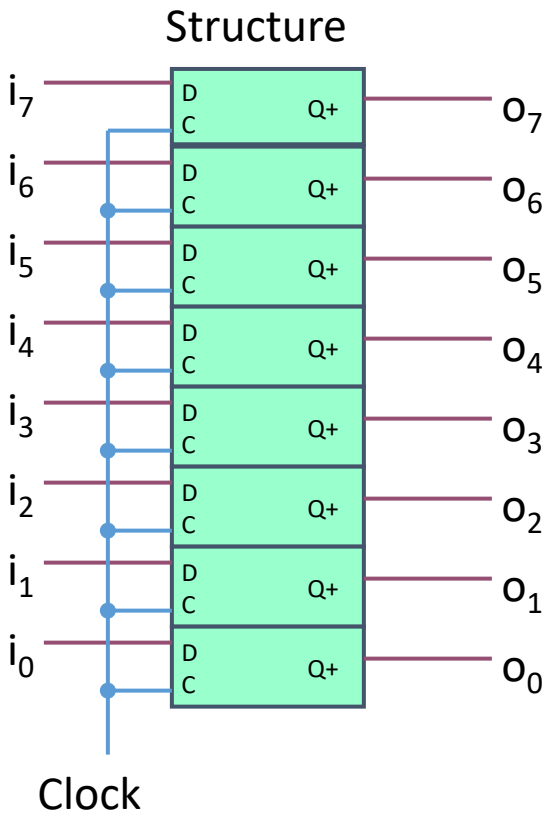


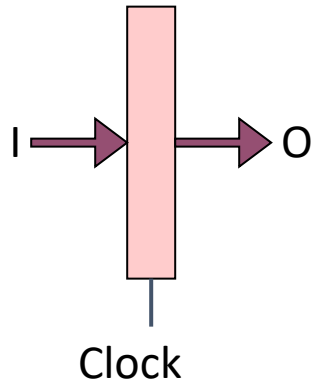
- Only in latching mode for brief period
  - Rising clock edge
- Value latched depends on the data as the clock rises
- Output remains stable at all other times

FOR THE REST OF THE CLOCK CYCLE !!!

Registers

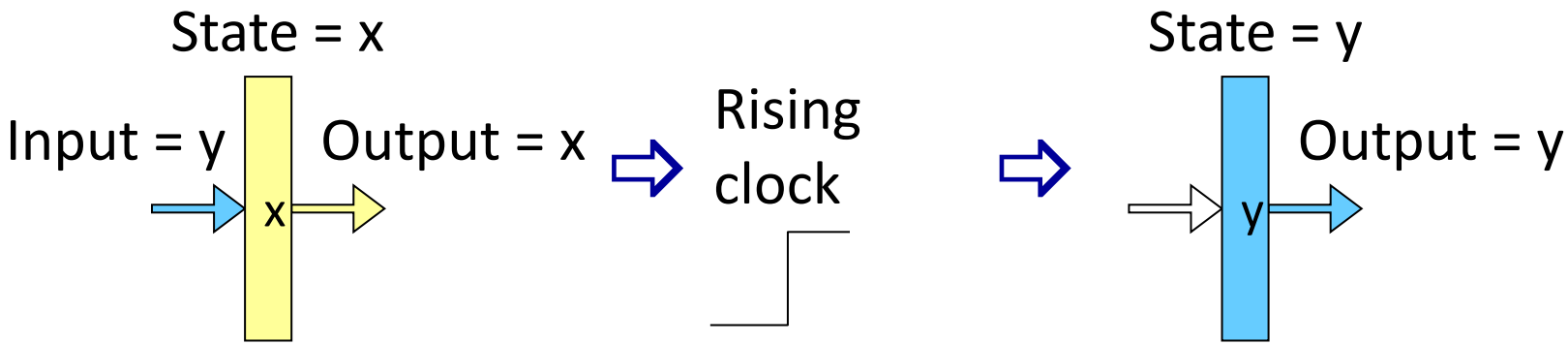
- Stores a word of data
- Collection of edge-triggered latches
- Loads input on rising edge of clock





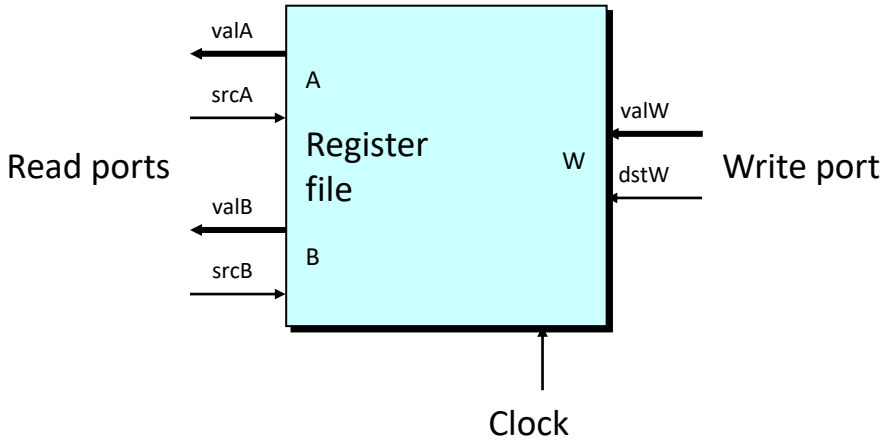
Register Operation

- Stores data bits
- Most of the time it acts as barrier between input and output
- At the rising edge of the clock, it loads its input



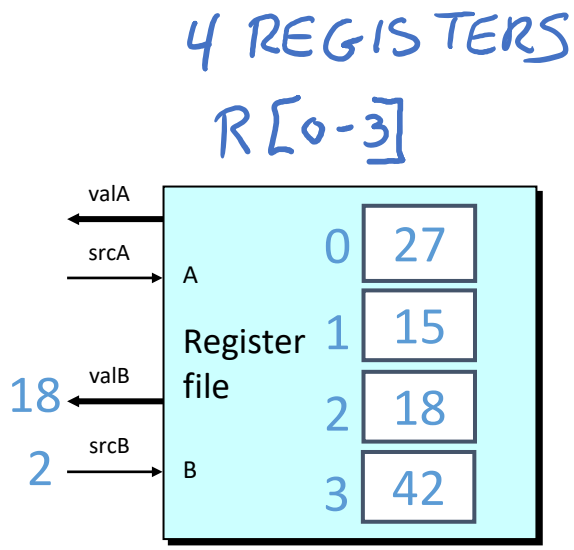
Random-Access Memory

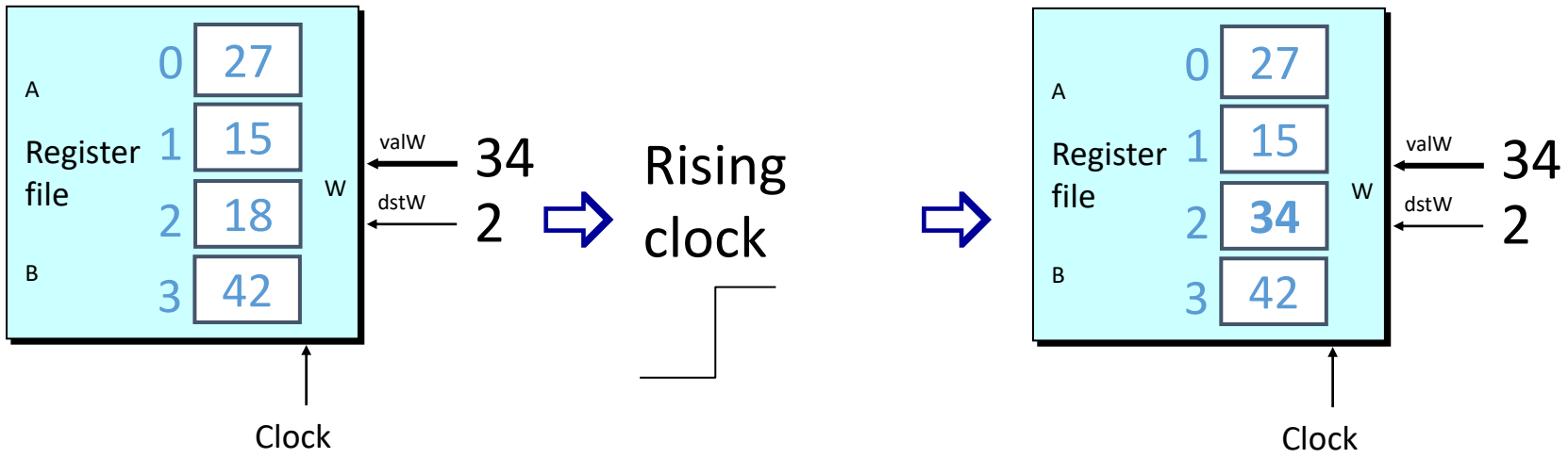
- Stores multiple words of memory
  - Address input specifies which word to read or write
- Register file
  - Holds values of program registers
  - `a0`, `ra`, etc.
  - Register identifier serves as index into the register file
- Multiple Ports
  - Can read and/or write multiple words in one cycle
    - Each has separate address and data input/output



Register File Timing

- Reading
  - Like combinational logic
  - Output data generated based on input address
    - After some delay
- Writing
  - Like a register
  - Update only as clock rises





Summary

- Computation
  - Performed by combinational logic
  - Computes Boolean functions
  - Continuously reacts to input changes
- Storage
  - Registers
    - Hold single words
    - Loaded as clock rises
  - Random-access memories
    - Hold multiple words
    - Possible multiple read or write ports
    - Read word when address input changes
    - Write word as clock rises