Jin-Soo Kim
(jinsoo.kim@snu.ac.kr)

Systems Software &
Architecture Lab.

Seoul National University

Fall 2019

# Logic Design

Chap. 4.2, Appendix A

# Logic Design Basics

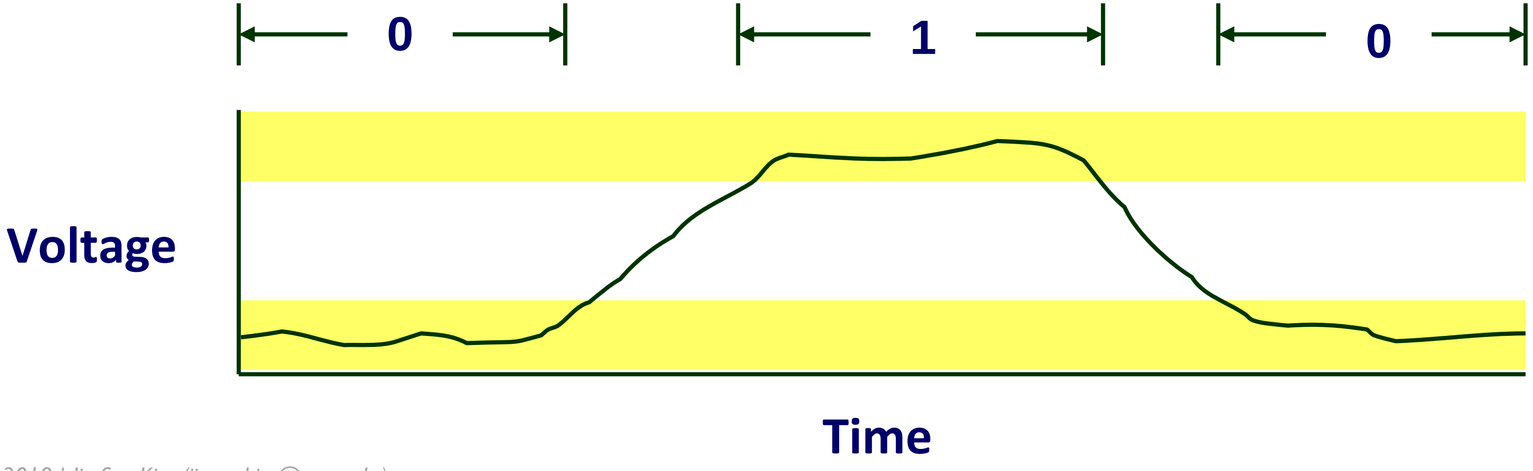
- Information encoded in binary
  - Low voltage = 0, High voltage = 1
  - One wire per bit
  - Multi-bit data encoded on multi-wire buses

- Combinational elements
  - Operate on data
  - Output is a function of input

- State (sequential) elements
  - Store information

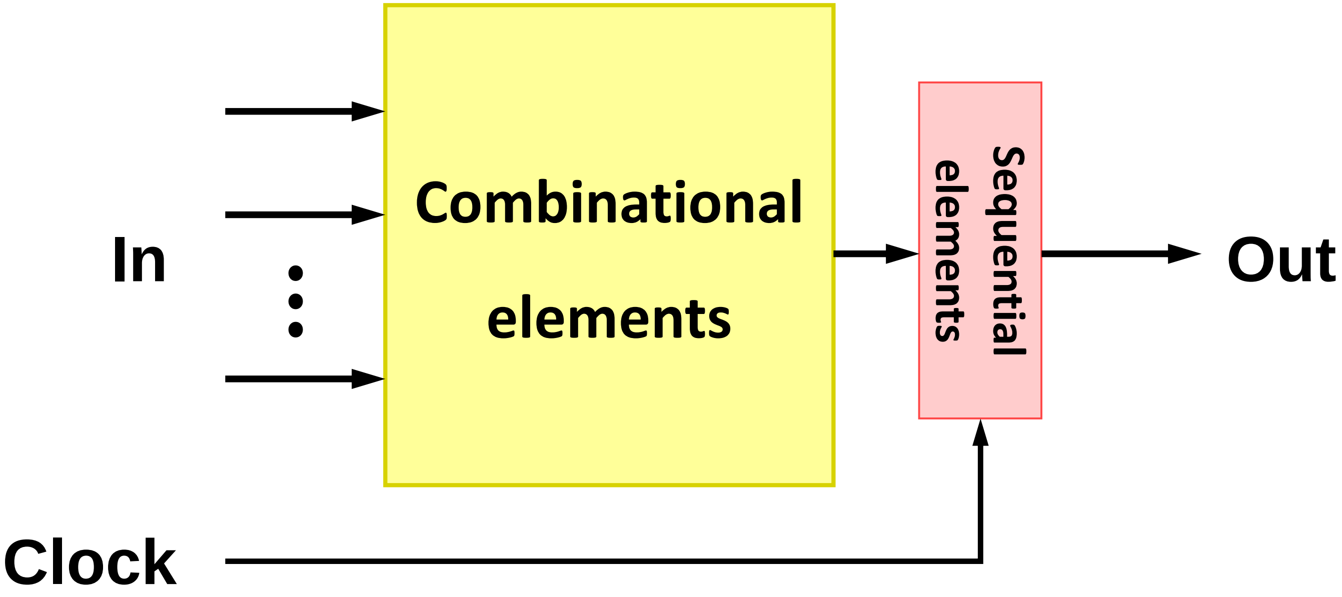
# Digital Signals

- Use voltage thresholds to extract discrete values from continuous signal
- Simplest version: 1-bit signal
  - Either high range (1) or low range (0)
  - With guard range between them
- Not strongly affected by noise or low quality circuit elements
  - Can make circuits simple, small, fast, and robust



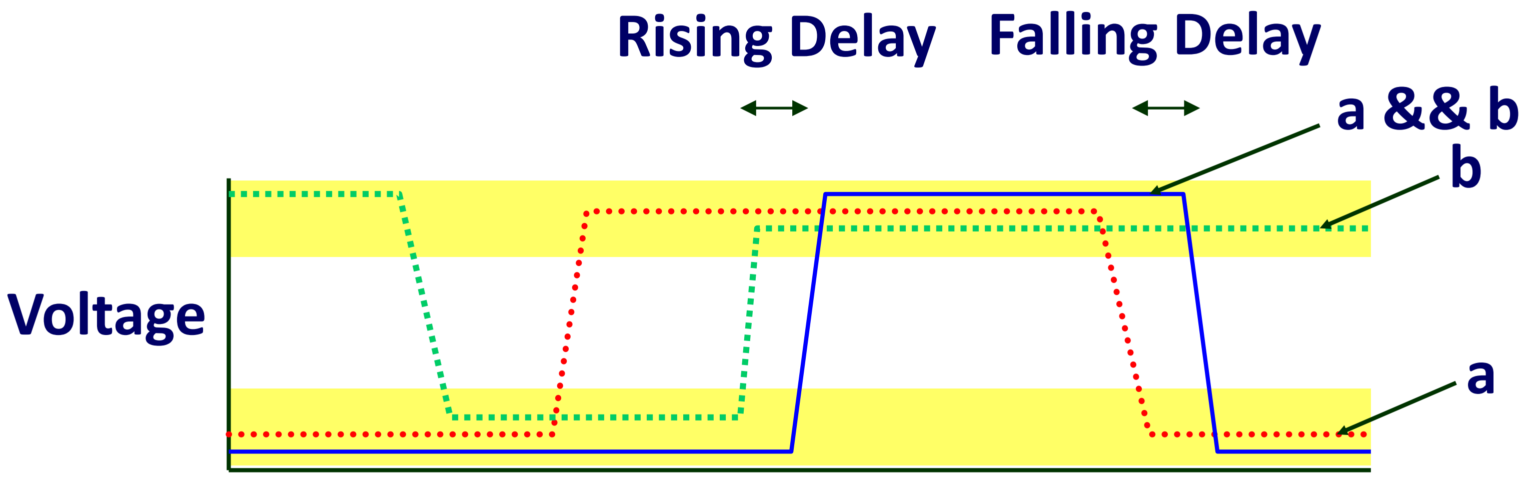
# Digital Systems

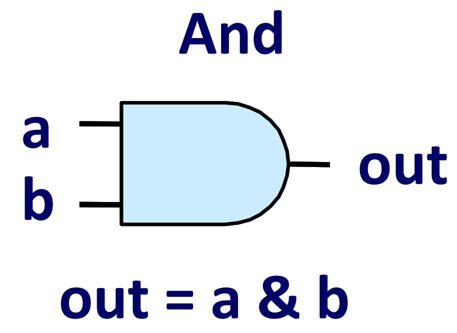
- Three components required to implement a digital system
  - Combinational elements to compute Boolean functions
  - Sequential elements to store bits
  - Clock signals to regulate the updating of the memory elements

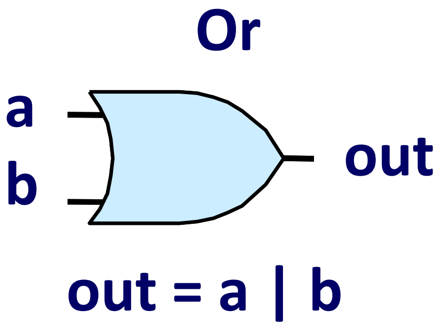


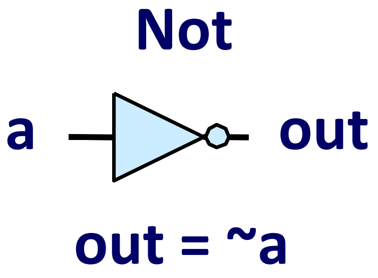
# Computing with Logic Gates

- Outputs are Boolean functions of inputs
- Respond continuously to changes in inputs (with some, small delay)



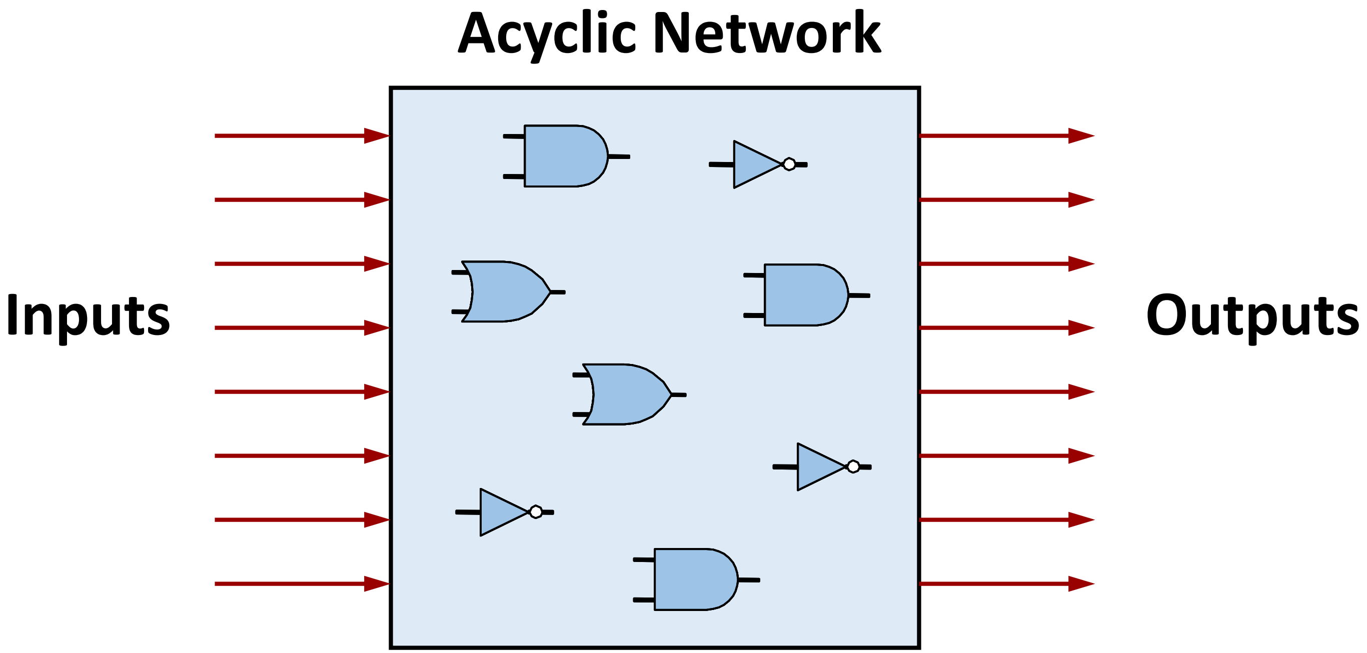






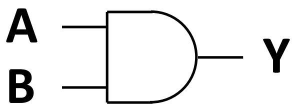
# Combinational Circuits

- Acyclic network of logic gates
  - Continuously responds to changes on primary inputs
  - Primary outputs become (after some delay) Boolean functions of primary inputs

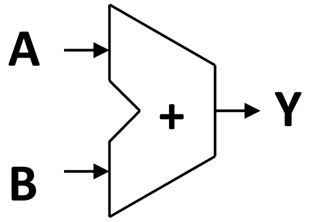


# Combinational Elements: Examples

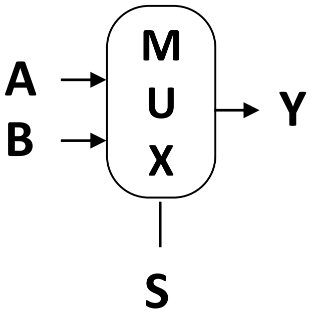
- AND-gate
  - `Y = A & B`



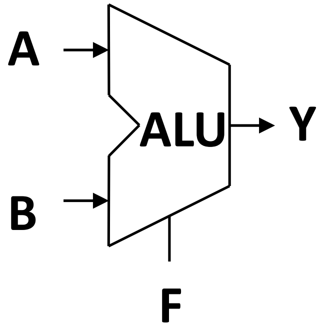
- Adder
  - `Y = A + B`



- Multiplexer
  - `Y = S? A : B`

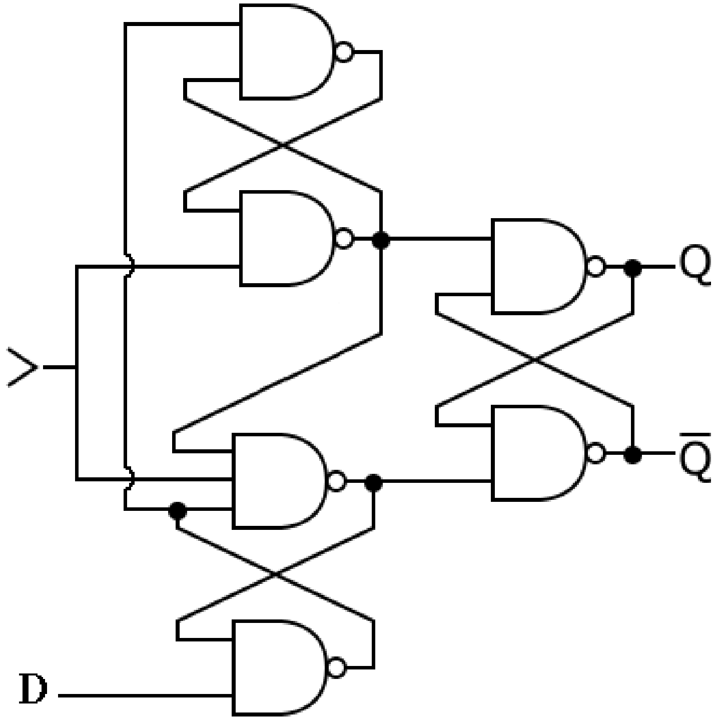


- Arithmetic/Logic Unit
  - `Y = F(A, B)`

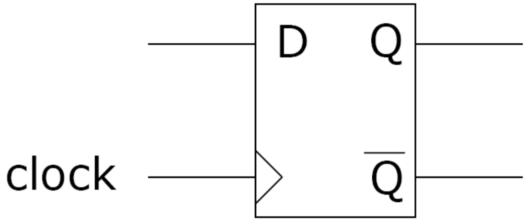


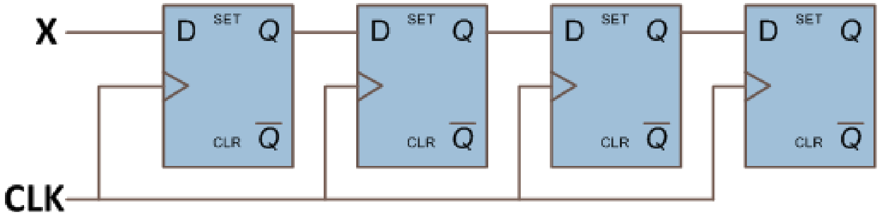
# Sequential Elements: Examples

- Flip-flops

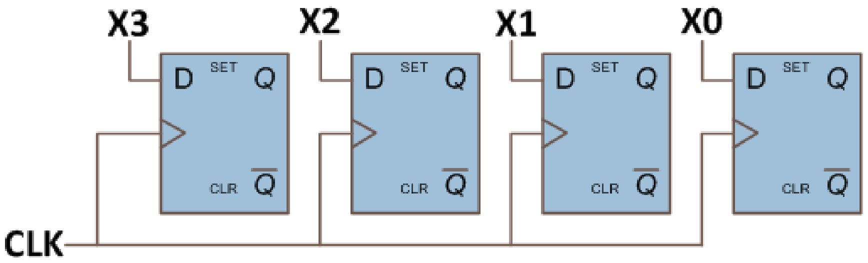


Edge-triggered D flip-flop





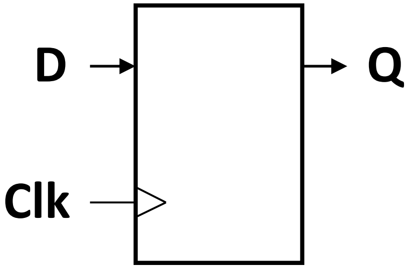
Shifter

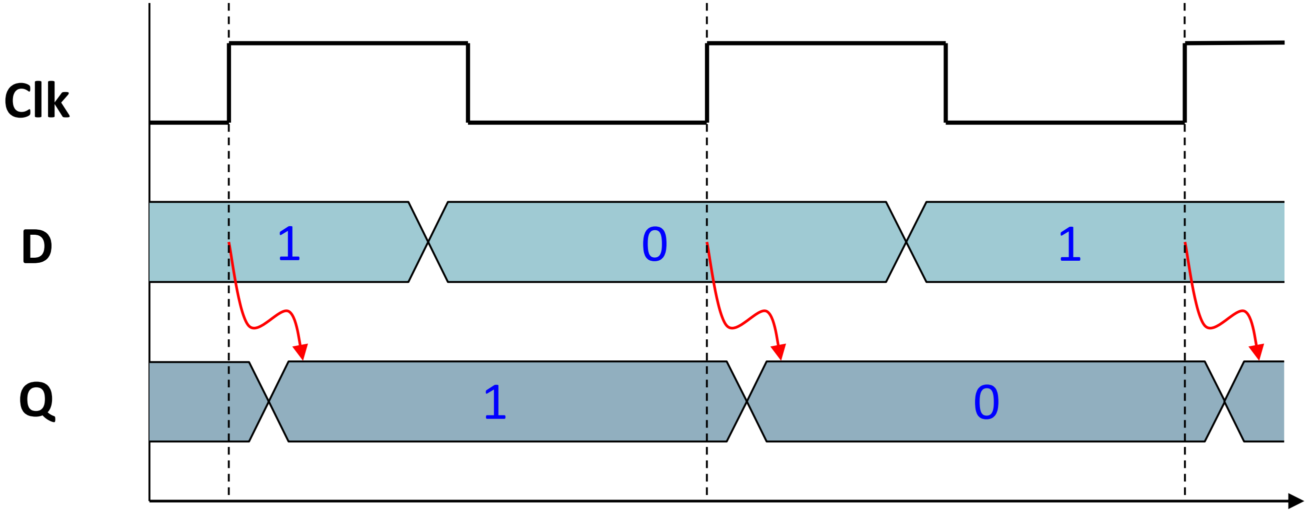


4-bit register

# Sequential Elements: Storing 1 bit

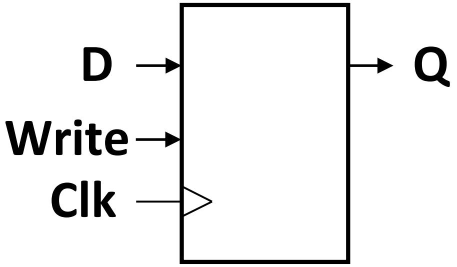
- Register: stores data in a circuit
  - Uses a clock signal to determine when to update the stored value
  - Edge-triggered: update when Clk changes from 0 to 1

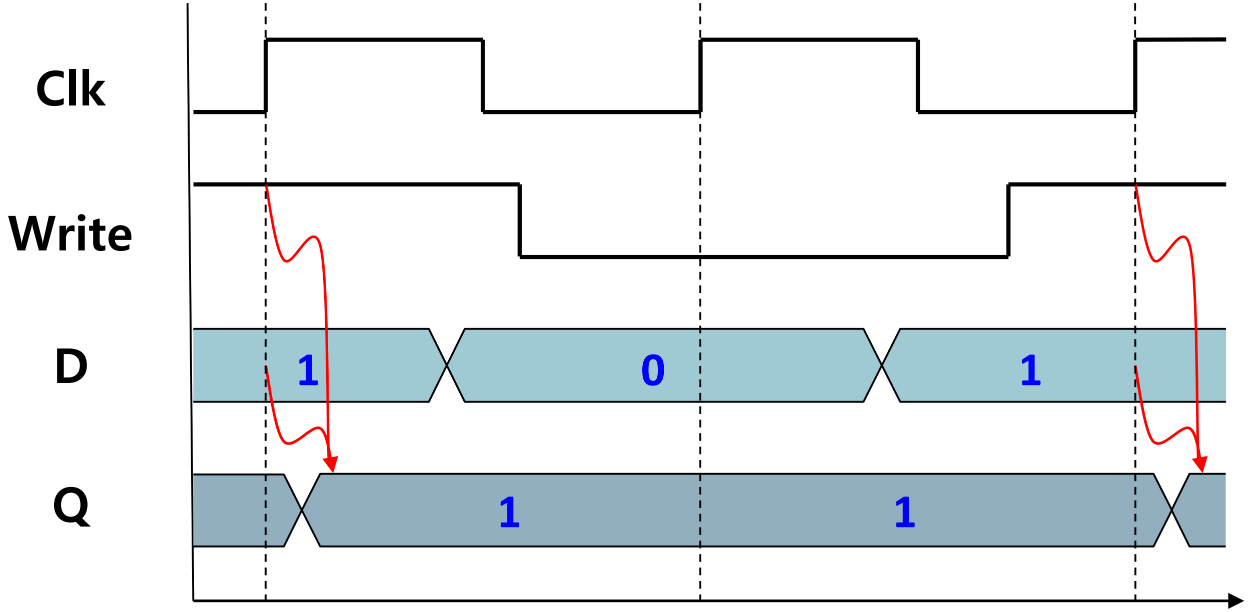




# Sequential Elements: Storing 1 bit (cont'd)

- Register with write control
  - Only updates on clock edge when write control input is 1
  - Used when stored value is required later

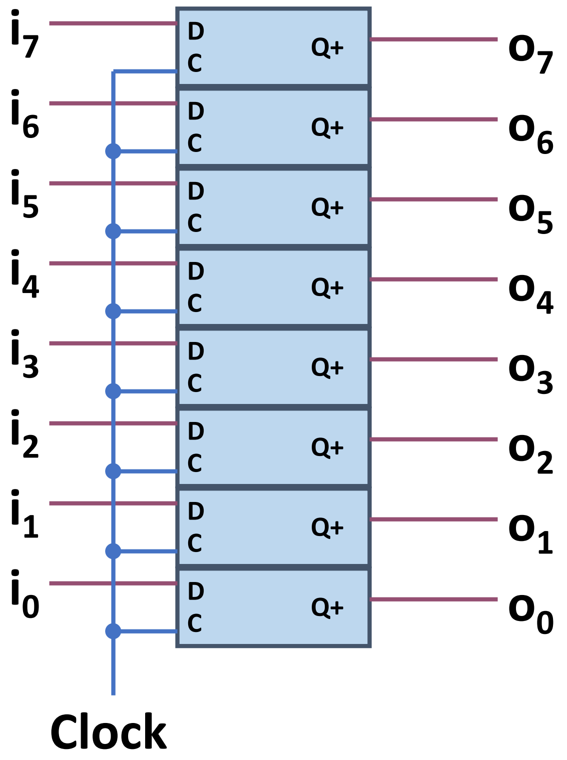


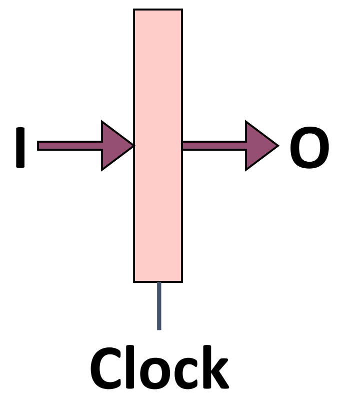


# (Hardware) Registers

- Stores word of data
  - Different from program registers seen in assembly code
- Collection of edge-triggered latches
- Loads input on rising edge of clock

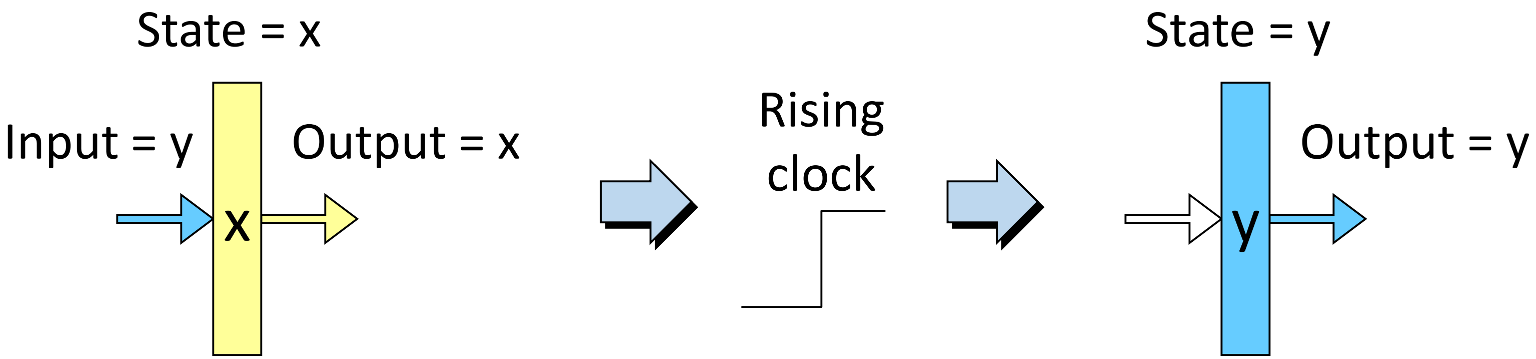
Structure





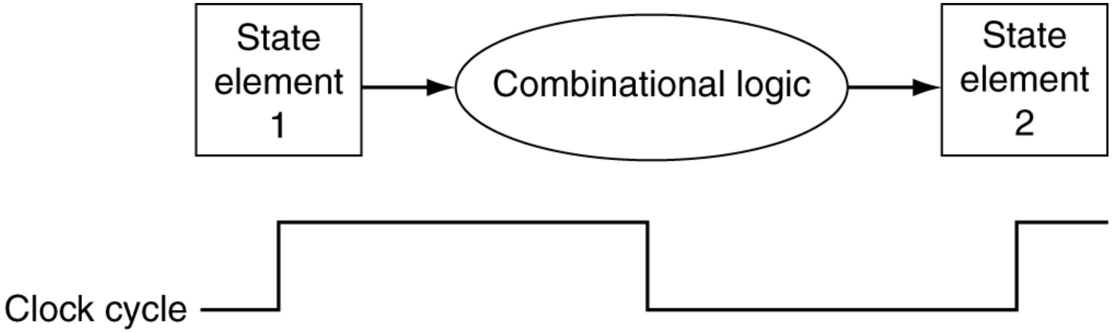
# Register Operation

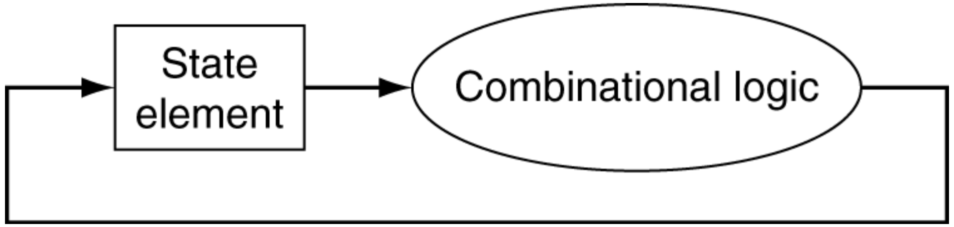
- Stores data bits
- For most of time acts as barrier between input and output
- As clock rises, loads input



# Clocking Methodology

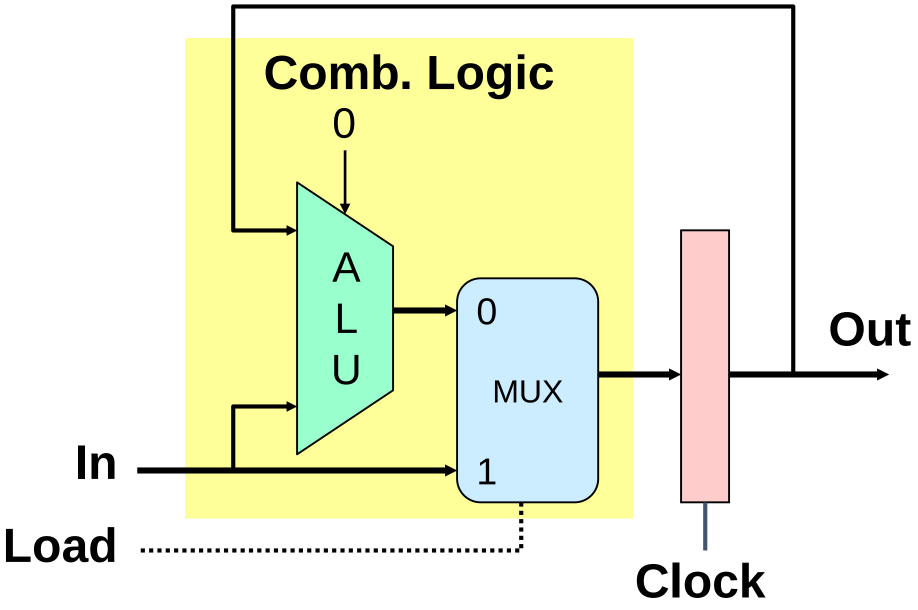
- Combinational logic transforms data during clock cycles
  - Between clock edges
  - Input from state elements, output to state element
  - Longest delay determines clock period

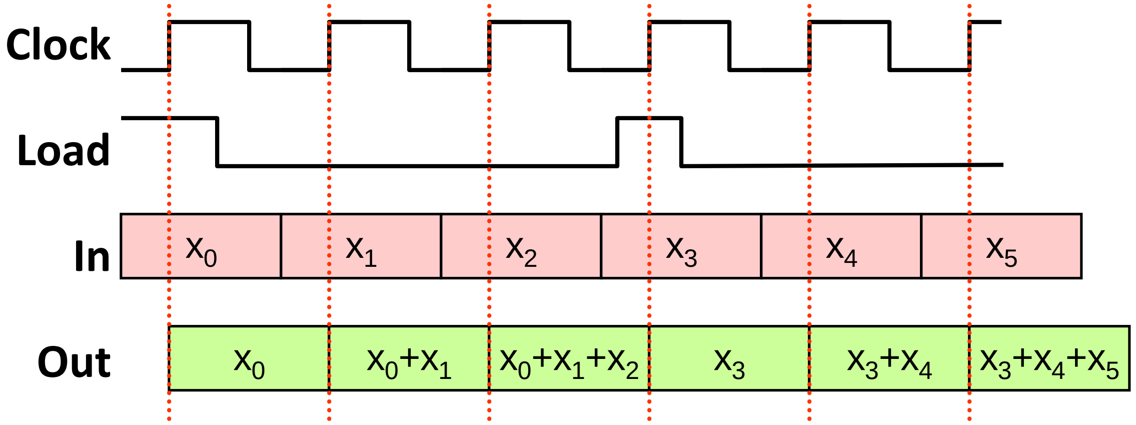




# State Machine Example

- Accumulator circuit
- Load or accumulate on each cycle

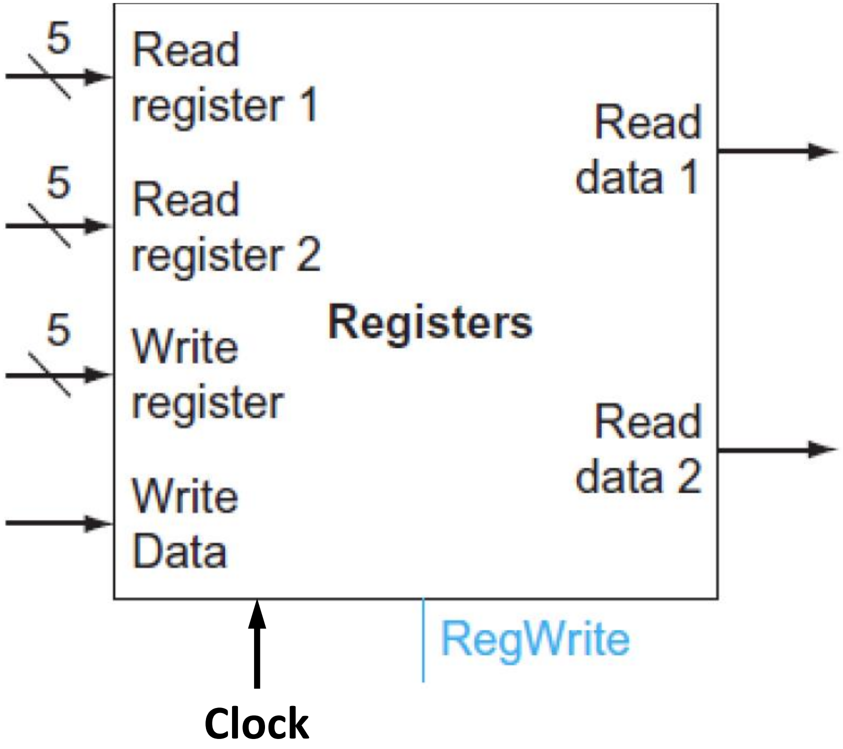




# Register File

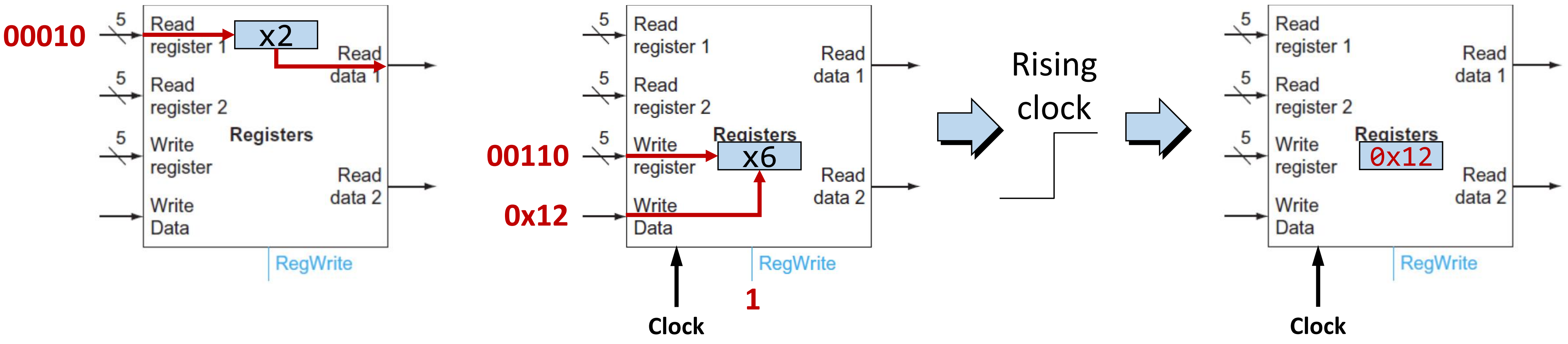
- A collection of registers
  - Holds values of "program registers"
  - `x0` ~ `x31` in RISC-V
  - Register identifier (5 bits) serves as address

- Multiple ports
  - Can read and/or write multiple words in one cycle
  - Each has separate address and data input/output
  - Data is written to the register only when RegWrite signal is enabled



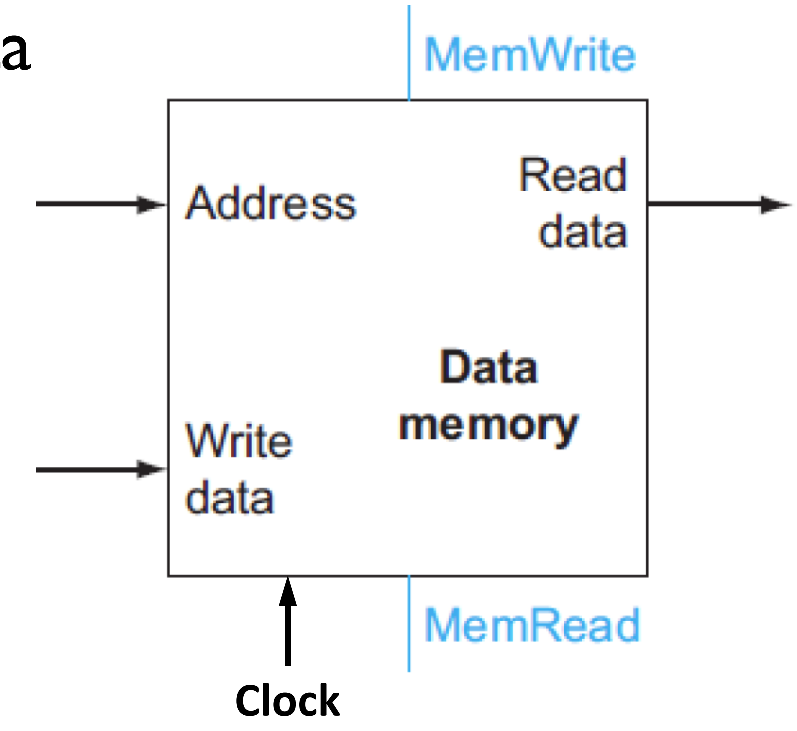
# Register File Timing

- **Reading**
  - Like combinational logic
  - Output data generated based on input address (after some delay)
- **Writing**
  - Like register
  - Update only as clock rises



# Data Memory

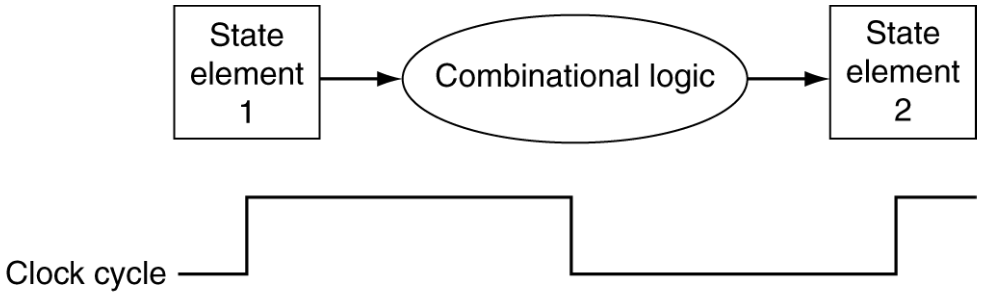
- Random access memory for storing program data
- Operations similar to registers
- Reading: like combinational logic
- Writing: update only as clock rises
- Read/Write controlled using MemWrite and MemRead signals



- Another read-only memory needed for instructions
- Dual-port memory: single memory for both instructions and data
  - One read port for instructions, another read or write port for data

# 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
    - Possibly multiple read or write ports
    - Read word when address input changes
    - Write word as clock rises


State element 1
Combinational logic
State element 2
Clock cycle