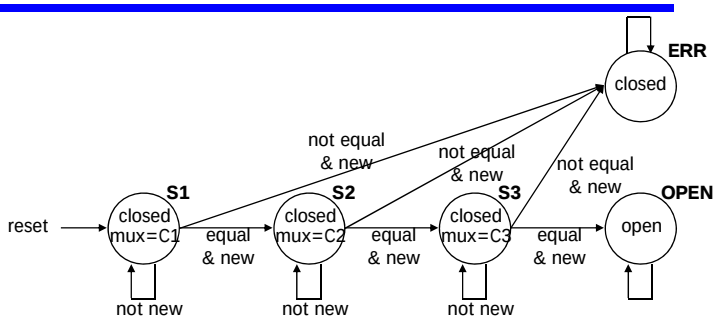
Lecture 13

- Logistics
  - HW5 due Wednesday
- Last lecture
  - Finished combinational logic
  - Introduction to sequential logic and systems
- Today
  - Finish the example from last time
  - Latches
  - Flip-flops

Example from last time: Combination lock

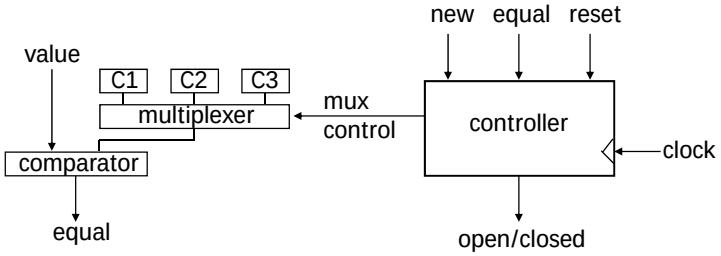
- Door combination lock
  - Enter 3 numbers in sequence and the door opens
  - If there is an error the lock must be reset
  - After the door opens the lock must be reset
  - Inputs: Sequence of numbers, reset
  - Outputs: Door open/close
  - Memory: Must remember the combination

State diagram and state table



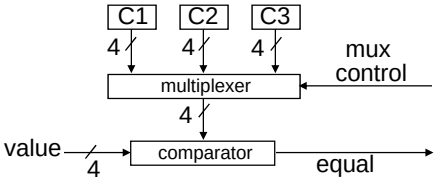
reset	new	equal	state	next state	mux	open/closed
1	–	–	–	S1	C1	closed
0	0	–	S1	S1	C1	closed
0	1	0	S1	ERR	–	closed
0	1	1	S1	S2	C2	closed
…						
0	1	1	S3	OPEN	–	open
…						

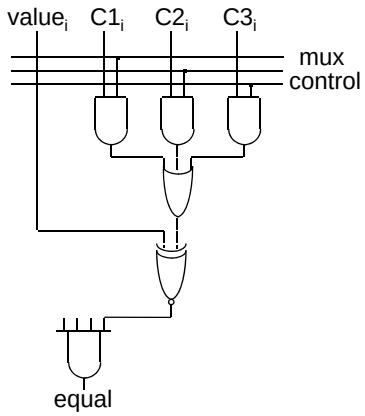
Combination lock system design



Designing the datapath

- Four 3:1 multiplexers
  - 2-input ANDs and 3-input OR
- Four single-bit comparators
  - 2-input XNORs
- 4-input AND

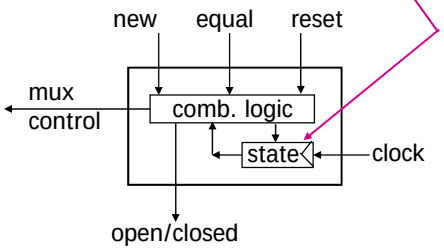




Designing the controller

- We will learn how to design the controller given the encoded state-transition table

special circuit element, called a register, for storing inputs when told to by the clock

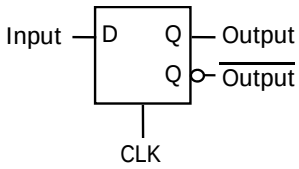


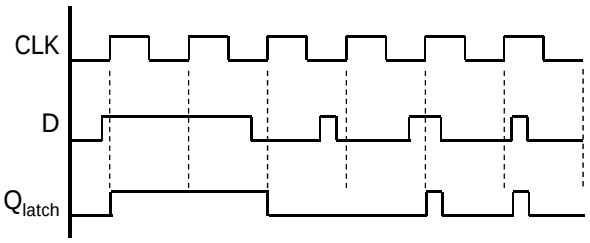
Now we will learn how to store things!

- First, we will learn several different logic elements (like how you had to learn AND and OR before you could do functional things)
  - D latch
  - D flip flop
  - T flip flop
  - SR latch
- Be patient --- once you know these components, you can do a lot in combinational logic!

The D latch

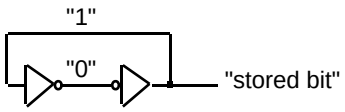
- Output depends on clock
  - Clock high: Input passes to output
  - Clock low: Latch holds its output
- Latch are level sensitive and transparent



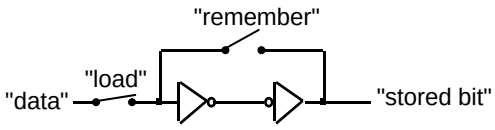


How do we make a latch?

- Two inverters hold a bit
  - As long as power is applied

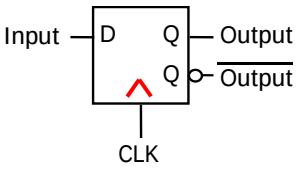


- Storing a new memory
  - Temporarily break the feedback path

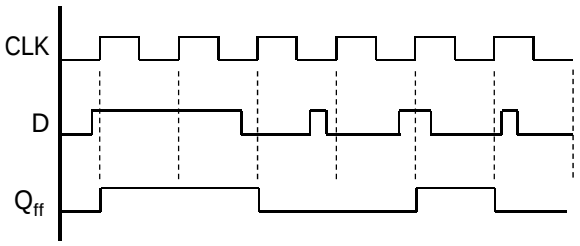


The D flip-flop

- Input sampled at clock edge
  - Rising edge: Input passes to output
  - Otherwise: Flip-flop holds its output

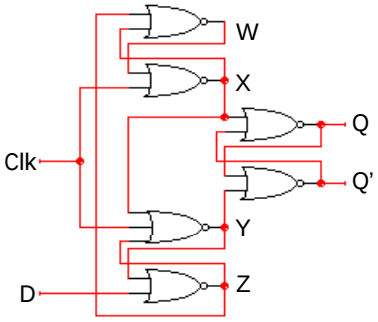


- Flip-flops are rising-edge triggered, falling-edge triggered, or master-slave



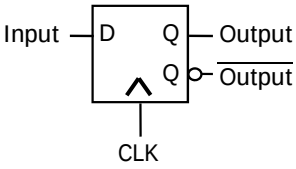
How do we make a D flip flop?

- Edge triggering is difficult
  - You can do this at home:
    - Label the internal nodes
    - Draw a timing diagram
    - Start with Clk=1

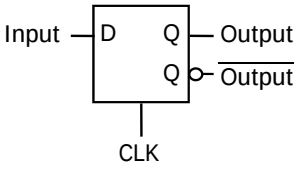


Terminology & notation

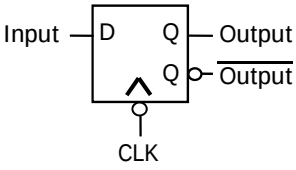
Rising-edge triggered D flip-flop



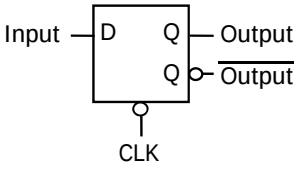
Positive D latch



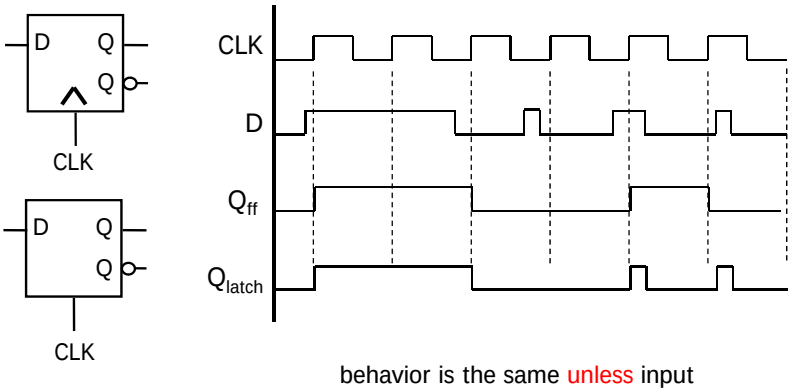
Falling-edge triggered D flip-flop



Negative D latch



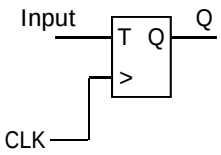
Latches versus flip-flops



behavior is the same unless input changes while the clock is high

T flip-flop

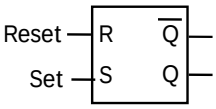
- Full name: Toggle flip-flop
- Output toggles when input is asserted
  - If T=1, then Q → Q' when CLK ↑
  - If T=0, then Q → Q when CLK ↑

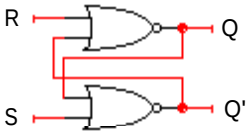


Input(t)	Q(t)	Q($t + \Delta t$)
0	0	0
0	1	1
1	0	1
1	1	0

The SR latch

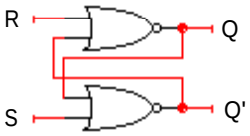
- Cross-coupled NOR gates
  - Can set (S=1, R=0) or reset (R=1, S=0) the output



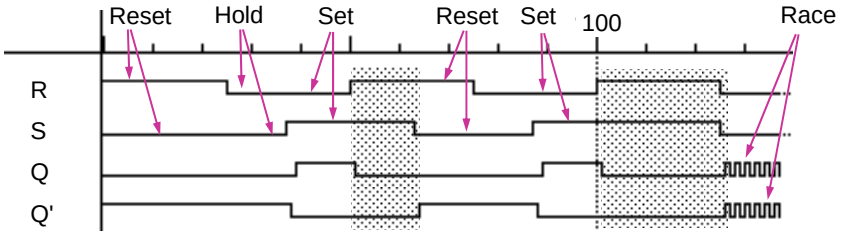


SR latch behavior

- Truth table and timing



S	R	Q
0	0	hold
0	1	0
1	0	1
1	1	disallow



SR latch is glitch sensitive

- Static 0 hazards can set/reset latch
  - Glitch on S input sets latch
  - Glitch on R input resets latch