BROWN
School of Engineering

DIGITAL ELECTRONICS SYSTEM DESIGN

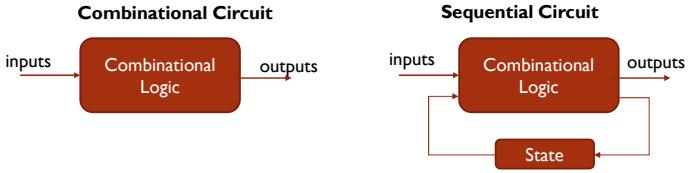
FALL 2019

PROF. IRIS BAHAR (TAUGHT BY JIWON CHOE)

OCTOBER 9, 2019

LECTURE 11: SEQUENTIAL LOGIC

COMBINATIONAL VS. SEQUENTIAL LOGIC



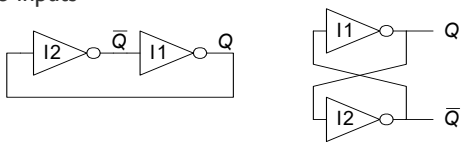
- Combinational:
  - Output depends only on current inputs
- Sequential:
  - Output depend on current inputs plus past history
  - Includes memory elements

SEQUENTIAL CIRCUITS

- Outputs depend on inputs and state variables
- The state variables embody the past
  - Storage elements hold the state variables
- A clock periodically advances the circuit

BISTABLE MEMORY STORAGE ELEMENT

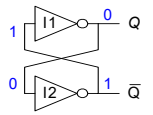
- Fundamental building block of other state elements
- Two outputs, Q, $\overline{Q}$
- No inputs



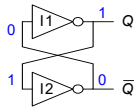
- What does the circuit do?

BISTABLE MEMORY ELEMENT

- Consider the two possible cases:
  - $Q = 0$: then $Q' = 1$ and $Q = 0$ (consistent)



  - $Q = 1$: then $Q' = 0$ and $Q = 1$ (consistent)



  - Bistable circuit stores 1 bit of state in the state variable, Q (or Q')
- But there are *no inputs to control the state*

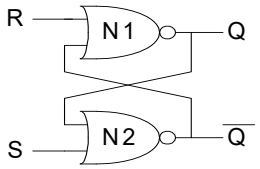
REVISIT NOR & NAND GATES

- Controlling inputs for NAND and NOR gates

- Implementing NOT with NAND/NOR using non-controlling inputs

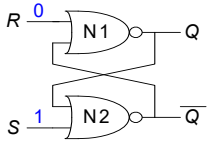
S-R (SET/RESET) LATCH

- Consider the four possible cases:
  - $S = 1, R = 0$
  - $S = 0, R = 1$
  - $S = 0, R = 0$
  - $S = 1, R = 1$

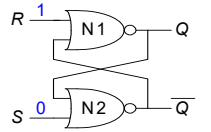


S-R LATCH ANALYSIS

- $S = 1, R = 0$: then $Q = 1$ and $\overline{Q} = 0$

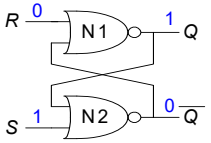


- $S = 0, R = 1$: then $Q = 0$ and $\overline{Q} = 1$

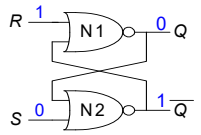


S-R LATCH ANALYSIS

– $S = 1, R = 0$: then $Q = 1$ and $\overline{Q} = 0$



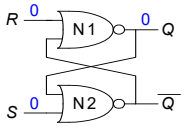
– $S = 0, R = 1$: then $Q = 0$ and $\overline{Q} = 1$



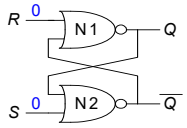
S-R LATCH ANALYSIS

– $S = 0, R = 0$: then $Q = Q_{prev}$

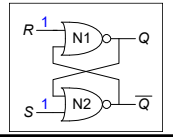
$Q_{prev} = 0$



$Q_{prev} = 1$

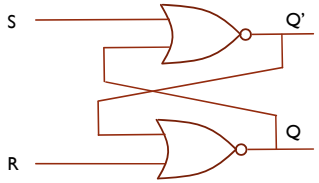


– $S = 1, R = 1$: then $Q = 0$ and $\overline{Q} = 0$



S-R LATCH AS MEMORY ELEMENT

- Boolean expression for S-R latch:



$Q' = (S + Q)'$

$Q_{next} = (R + Q')'$
$= (R + (S+Q)')'$
$= (R' \bullet (S+Q)$
$= R' \bullet S + R' \bullet Q$

S-R LATCH OPERATION

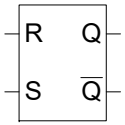
S	R	R' • S	R' • Q	Q_{next}
0	0	0	Q	Q
0	I	0	0	0
I	0	I	Q	I
I	I	0	0	0

- When S=0, R=0, latch holds its previous state
- When S=1, R=1, latch may become unstable
  - *What happens on a 11 → 00 transition?*

S-R LATCH SYMBOL

- SR stands for Set/Reset Latch
  - Stores one bit of state (Q)
- Control what value is being stored with S, R inputs
  - Set: Make the output 1 ($S = 1, R = 0, Q = 1$)
  - Reset: Make the output 0 ($S = 0, R = 1, Q = 0$)
- Must do something to avoid invalid state (when $S = R = 1$)

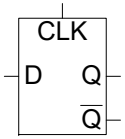
SR Latch Symbol



THE D LATCH

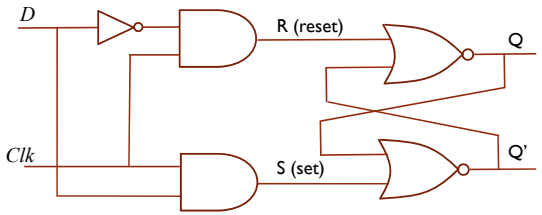
- D latch: builds on the S-R latch, where S and R cannot be both 1
  - Output "follows" input
- D latch captures input data (what to set) when certain condition holds (when to set)
- Operates in 2 modes:
  1. Open (transparent): input flows through to output
  2. Closed (opaque): output does not change

D Latch Symbol



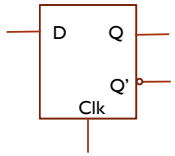
D LATCH

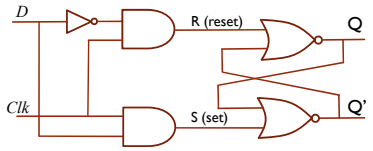
- Clock enabled: Q output follows D input
- Clock disabled: Q output retains last state



D LATCH

Clk	D	R	S	Q	Q'
1	1	0	1	1	0
1	0	1	0	0	1
0	X	0	0	Q_{prev}	Q'_{prev}

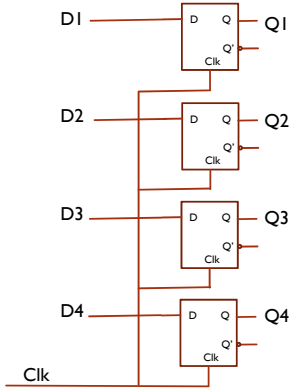




- Circuit guarantees R=S=1 will never occur

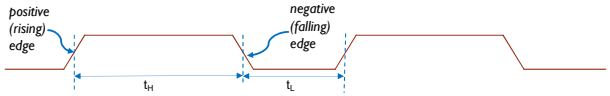
MULTI-BIT LATCH

- Simultaneously latch multiple bits
- A Latch may refer to a 1-bit latch or multi-bit latch



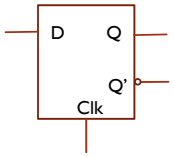
CLOCK

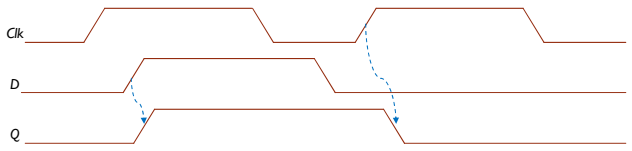
- **Clock:** An input to a sequential circuit that changes output and state value at a predetermined rate



- **Triggering edge:** Transition of the clock (L→H or H→L) that captures input data
  - *positive-edge* or *negative-edge*
  - Clock period (cycle time): time between successive transitions in the same direction (L→H or H→L)
  - Clock frequency = 1/clock period

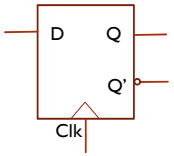
D LATCH TIMING



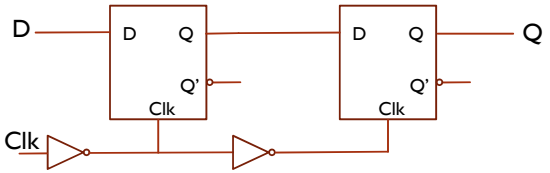


FLIP-FLOP

- Flip-flop: Samples input on triggering edge of clock
  - Rising edge → *positive edge-triggered* flip-flop
  - Falling edge → *negative edge-triggered* flip-flop
- D flip-flop: Two D latches back-to-back



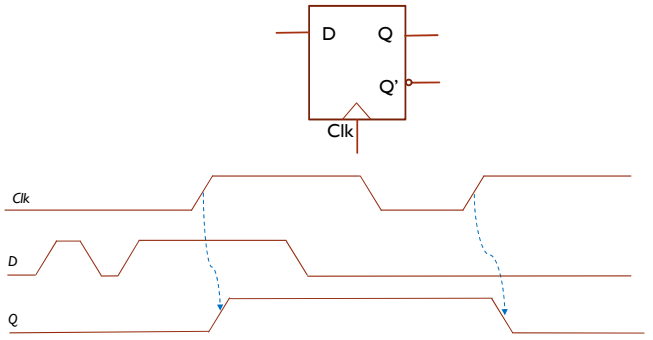
FLIP-FLOP MADE FROM D-LATCHES



D	Clk	Q
0	↑	0
1	↑	1
X	0	Q_{prev}
X	1	Q_{prev}

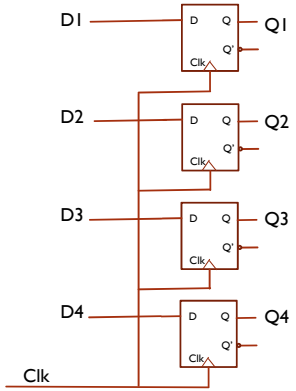
- Copies D to Q on the *rising edge* of the clock

D FLIP-FLOP TIMING



REGISTER

- Register: a collection of FFs operating off a common clock
- A single D flip-flop is a 1-bit register



FLIP-FLOPS VS. LATCHES

- Why use a flip flop when it just takes twice as much logic?
- When would you want to use a flip-flop instead of a latch?