Under content/HW in your pilot, there are some home exercise questions along with their brief answers for you to practice.

Circuit Outline

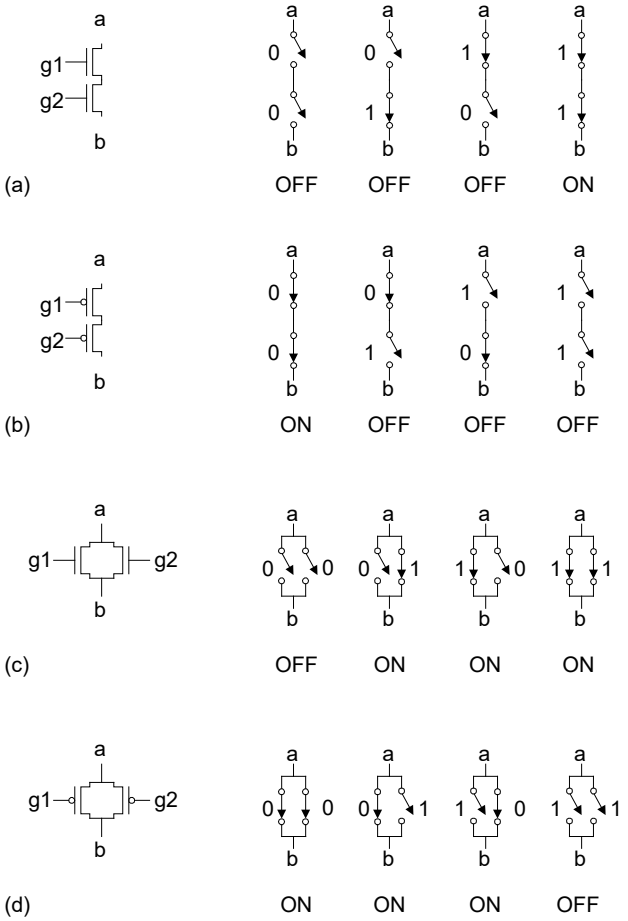
- CMOS Gate Design
- Pass Transistors
- Transmission gate
- Tristate
- Multiplexer
- CMOS Latches & Flip-Flops

CMOS Gate Design

- Activity:
  - Sketch a 4-input CMOS NOR gate

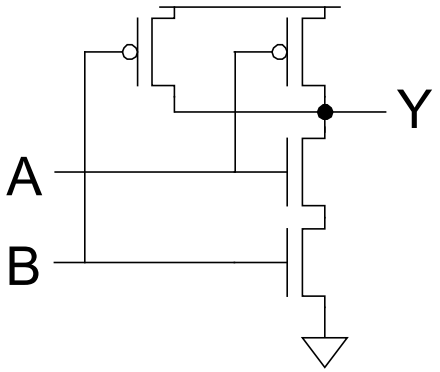
Series and Parallel

- nMOS: 1 = ON
- pMOS: 0 = ON
- *Series*: all must be ON
- *Parallel*: any can be ON



Conduction Complement

- Complementary CMOS gates always produce 0 or 1 for digital circuits
- Ex: NAND gate
  - Series nMOS: Y=0 when both inputs are 1
  - Thus Y=1 when either input is 0
  - Requires parallel pMOS



- Rule of *Conduction Complements*
  - Pull-up network is complement of pull-down
  - Parallel -> series, series -> parallel

Compound Gates—inverting circuit

Ex: $Y=\overline{((A.B)+(C.D))}$—this function is sometimes called AND-OR-INVERT-22 or AOI22 because it performs the NOR of a pair of 2-input ANDs

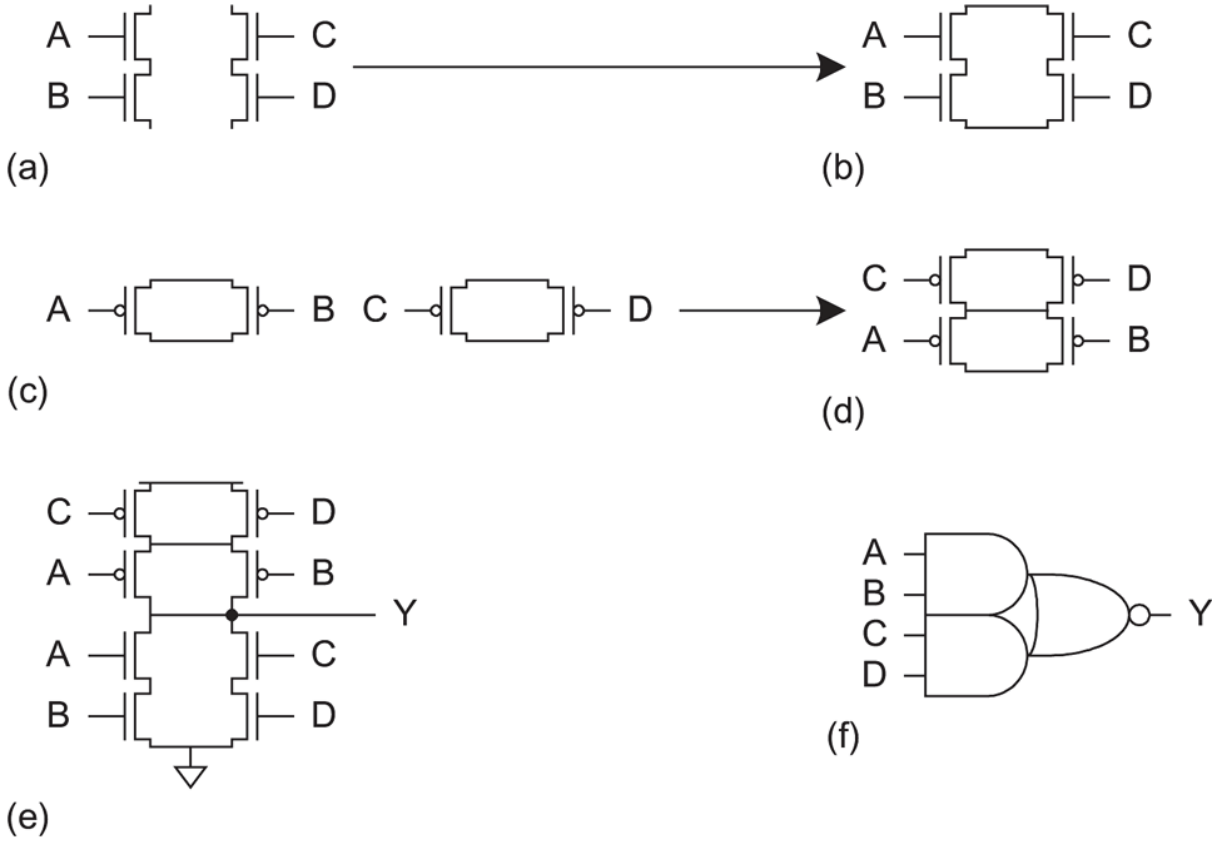


FIG 1.17 CMOS compound gate for function $Y = \overline{(A \bullet B) + (C \bullet D)}$

$Z=\overline{(abc+d)}$ is called AOI31; $Z=\overline{(ab+cd+e)}$ is called AOI221

Non-inverting circuits

- In general, the circuit outputs are not always the inverted inputs function
- Example 1 F=abc
  - Two ways
    1. Straight forward--adding an inverter to output

       let $\overline{G}=(abc)$, $F=\overline{G}=(abc)''$

       Total transistors =6+2=8
    2. Double inverter and use De Morgan—

       $F=\overline{(abc)}=\overline{a} + \overline{b} + \overline{c}$ =>

       Which required three inverters

       Total transistors =6+6=12

Non-inverting circuits (Cont)

- Example 2 $Z=(A+B)\bar{C}\,\bar{D}$

  A. Find ckt for G=$\bar{Z}$ and invert the output

  transistors

  14

  B. Double inverter and use De Morgan (no output inverter)

  transistors

  12

$\overline{\overline{Z}} = \overline{(A+B)\bar{C}\,\bar{D}}$

$Z = \overline{\overline{(A+B)\bar{C}\,\bar{D}}}$

$Z = \overline{\overline{(A+B)} + \overline{\bar{C}} + \overline{\bar{D}}}$

$= \overline{\bar{A}\bar{B} + C + D}$

Non-inverting circuits (Cont)

- Example 3 $F=(a'+b')(c+d)$

  A. Add inverter to output of G

  $F=G'$; $G'=\overline{((a'+b')(c+d))}$

  8+4+2=14 transistors

  B. De Morgan (no inverter output)

  $F'=((a'+b')(c+d))'=(a'+b')'+(c+d)'$

  $=ab+(c+d)'=ab+c'd'$

  8+4=12 transistors

- Which is a better design

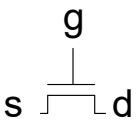
- Exercise for yourself
  1. Design both schematic and stick diagram for OR2/XOR2/NOR2/AND2 gates by using minimum CMOS transistors.
  2. Design schematic CMOS compound gates for functions Z=((a+b)(c+d))' and Z=(a'b'c+d) using least number of transistors.

Signal Strength

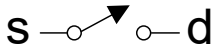
- *Strength* of signal
  - How close the signal approximates to ideal voltage source
- V_{DD} and GND rails are strongest ‘1’and ‘0’
- nMOS passes strong ‘0’
  - But degraded or weak ‘1’
- pMOS passes strong ‘1’
  - But degraded or weak ‘0’
- Thus nMOS are best for pull-down network and pMOS are best for pull-up network

Pass Transistors

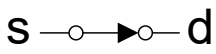
- Transistors can be used as switches



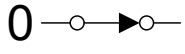
g = ‘0’

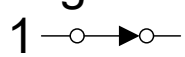


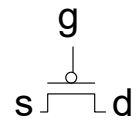
g = ‘1’



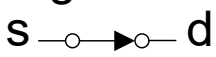
Input — g = 1 — Output



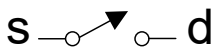




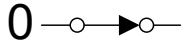
g = ‘0’

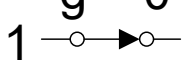


g = ‘1’



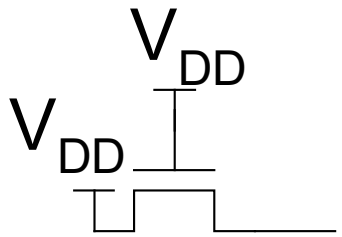
Input — g = 0 — Output





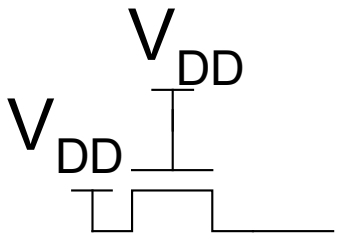
Pass Transistors

- We have assumed source is grounded
- What if source > 0?
  - e.g. pass transistor passing V_{DD}



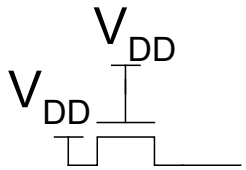
Pass Transistors

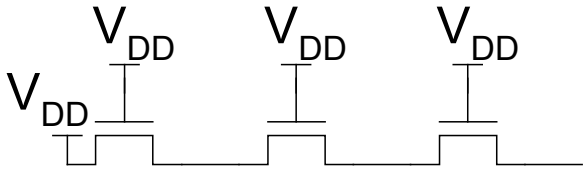
- We have assumed source is grounded
- What if source > 0?
  - e.g. pass transistor passing V_{DD}

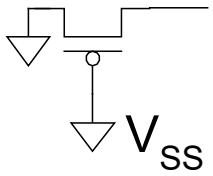


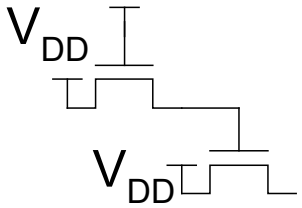
- $V_g = V_{DD}$
  - If $V_s > V_{DD}-V_t$, $V_{gs} < V_t$
  - Hence transistor would turn itself off
- nMOS pass transistors pull no higher than $V_{DD}-V_{tn}$
  - Called a degraded "1"
  - Approach degraded value slowly (low I_{ds})
- pMOS pass transistors pull no lower than $|V_{tp}|$

Pass Transistor Ckts

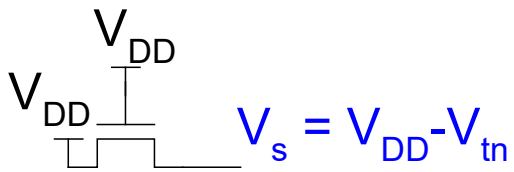


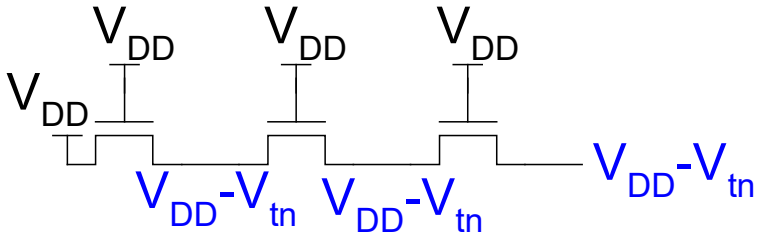


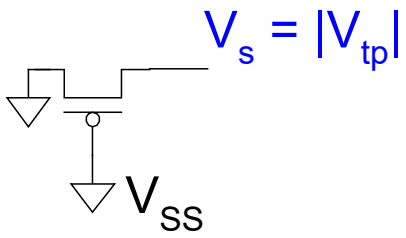


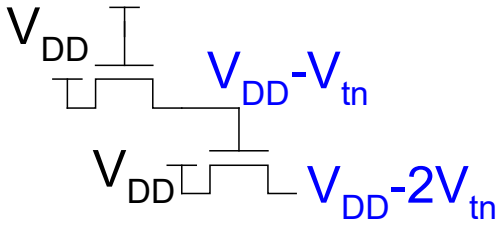


Pass Transistor Ckts



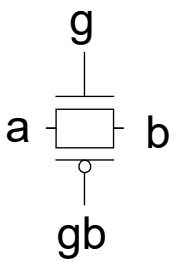




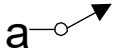


Transmission Gates

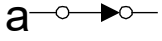
- Pass transistors produce degraded outputs
- *Transmission gates* pass both 0 and 1 well



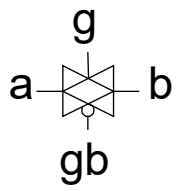
g = 0, gb = 1

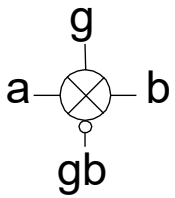


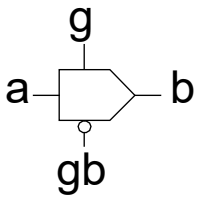
g = 1, gb = 0



Input	Output
g = 1, gb = 0 0	strong 0
g = 1, gb = 0 1	strong 1



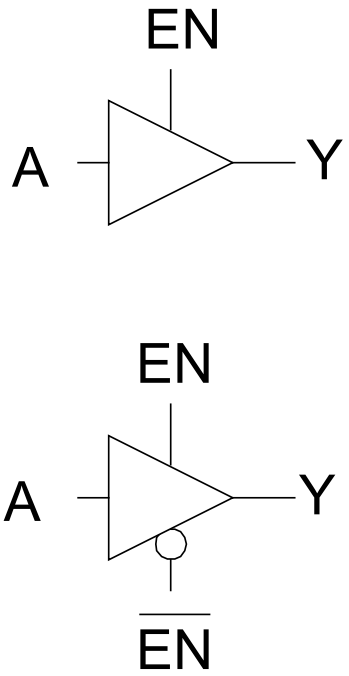




Tristates

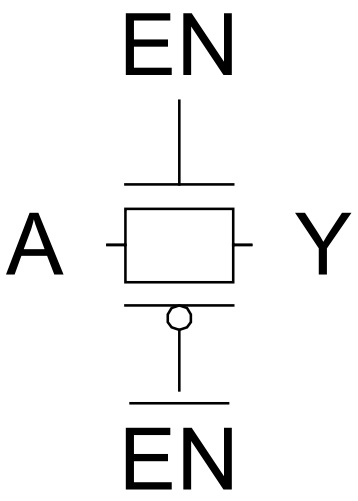
- *Tristate buffer* produces Z when not enabled

EN	A	Y
0	0	
0	1	
1	0	
1	1	



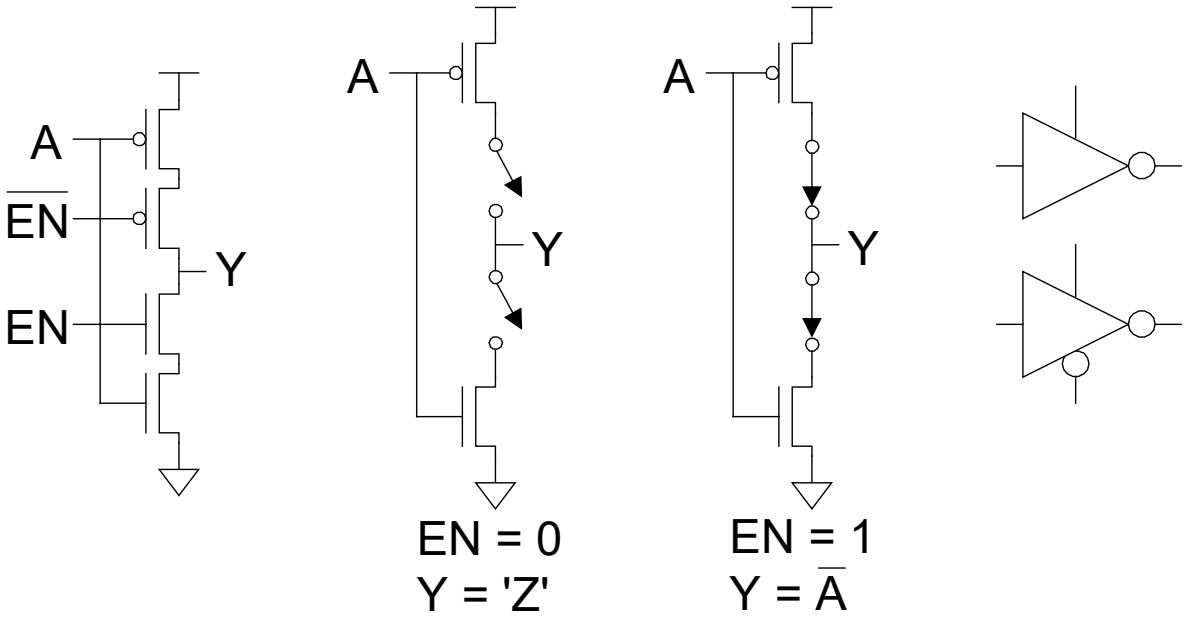
Nonrestoring Tristate

- Transmission gate acts as tristate buffer
  - Only two transistors
  - But *nonrestoring*
    - Noise on A is passed on to Y



Tristate Inverter

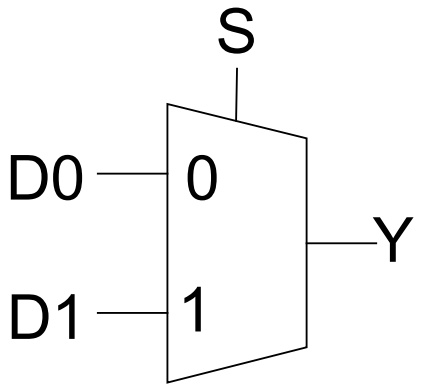
- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output



Multiplexers

- 2:1 multiplexer chooses between two inputs

S	D1	D0	Y
0	X	0	
0	X	1	
1	0	X	
1	1	X	

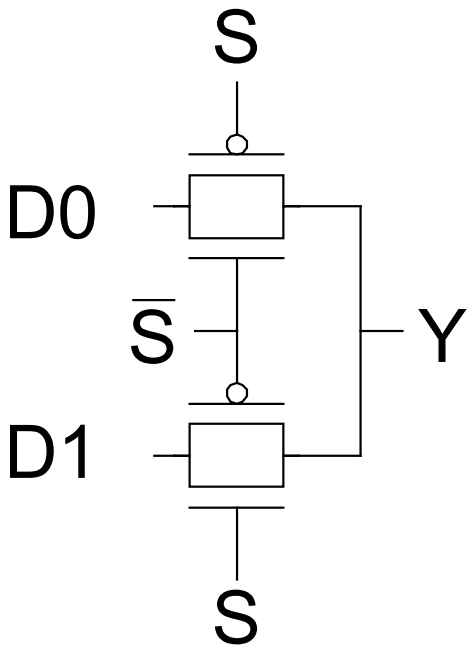


Gate-Level Mux Design

- $Y = SD_1 + \overline{S}D_0$ (too many transistors)
- How many transistors are needed?

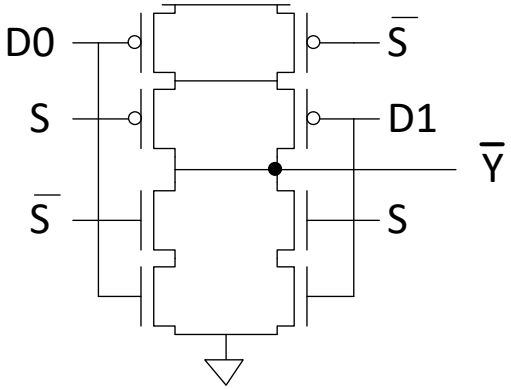
Transmission Gate Mux

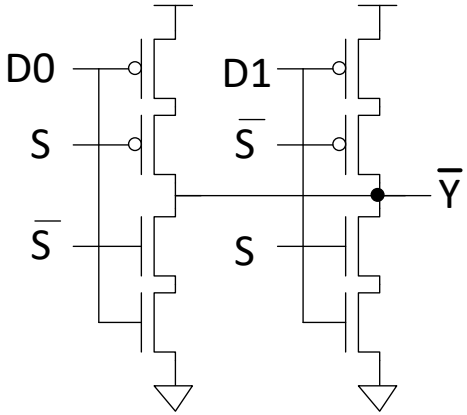
- Nonrestoring mux uses two transmission gates
  - Only 6 transistors

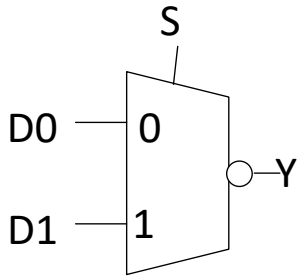


Inverting Mux

- Inverting multiplexer
  - Use compound AOI22
  - Or pair of tristate inverters
  - Essentially the same thing
- Noninverting multiplexer adds an inverter

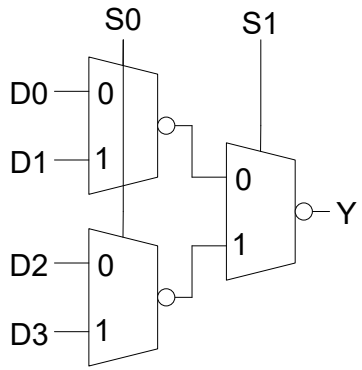


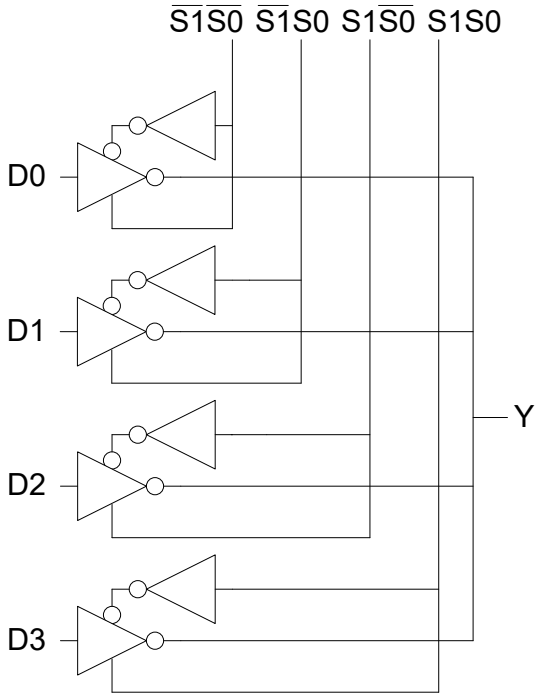




4:1 Multiplexer

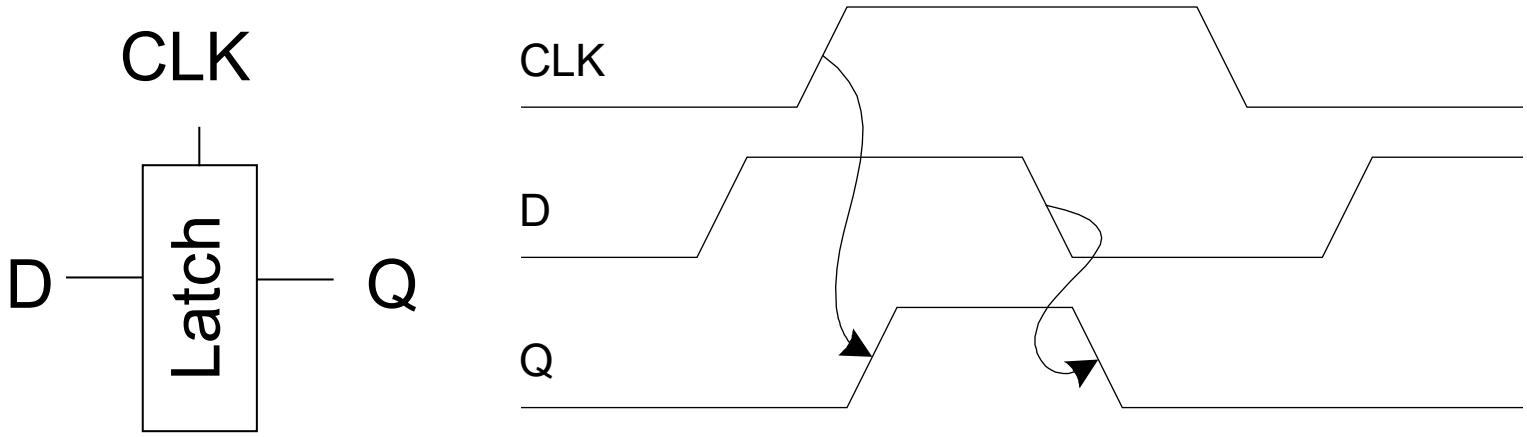
- 4:1 mux chooses one of 4 inputs using two selects
  - Two levels of 2:1 muxes
  - Or four tristates





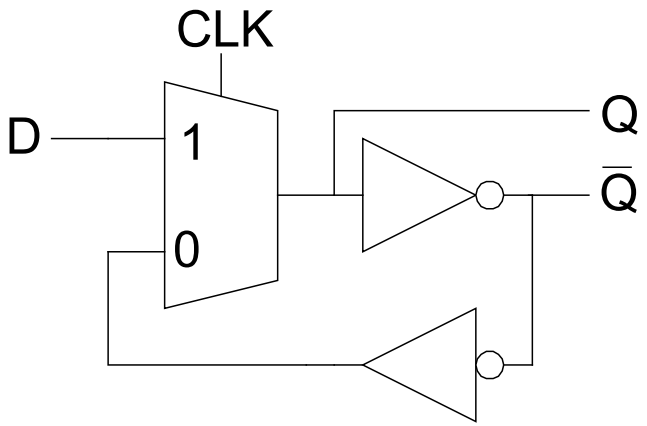
D Latch

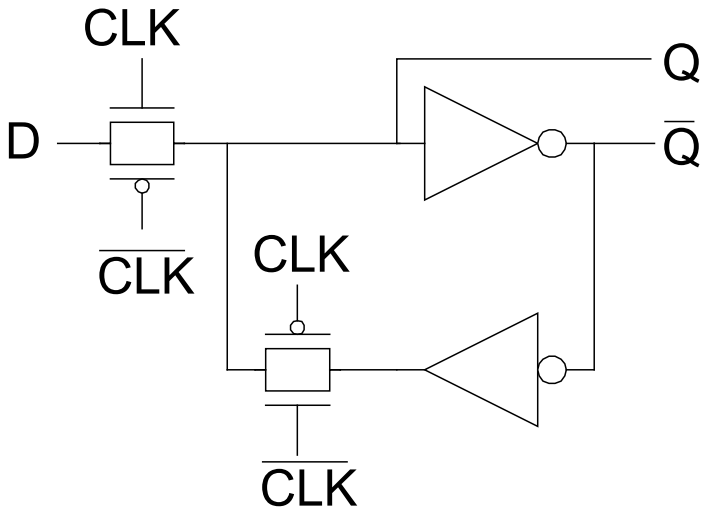
- When CLK = ‘1’, latch is *transparent*
  - D flows through to Q like a buffer
- When CLK = ‘0’, the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*



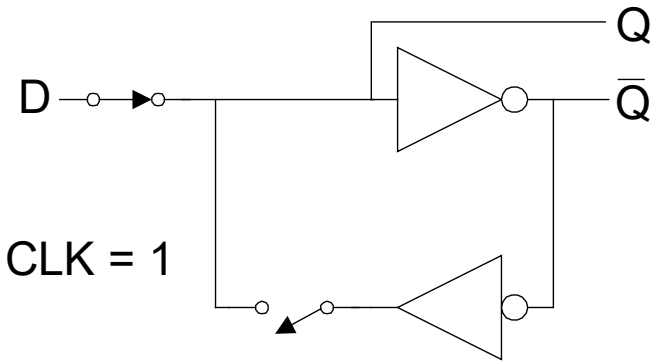
D Latch Design

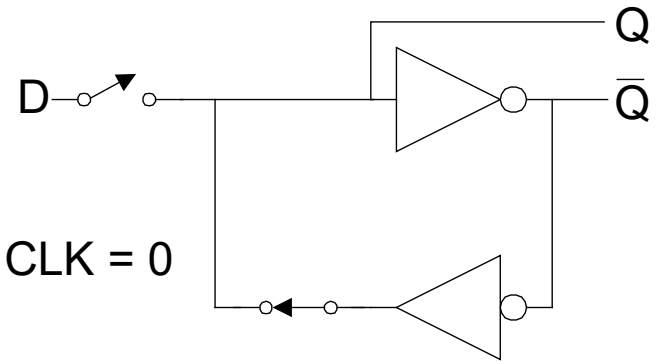
- Multiplexer chooses D or old Q

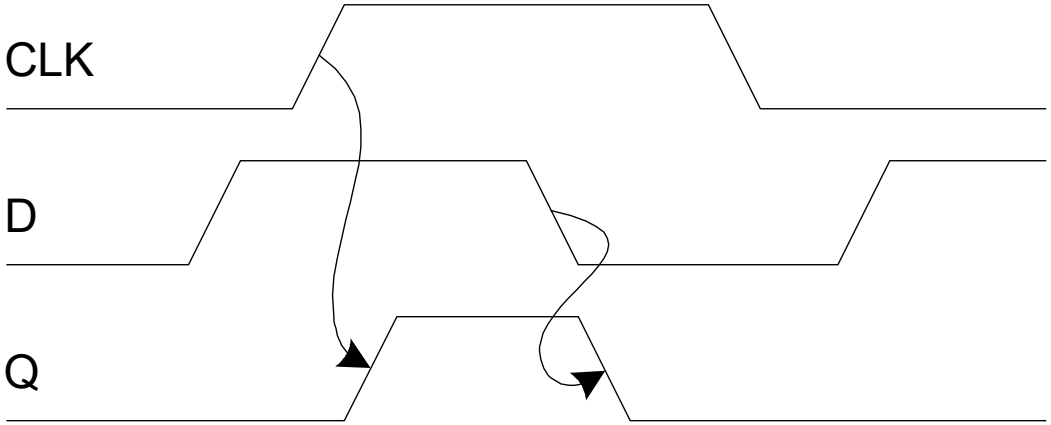


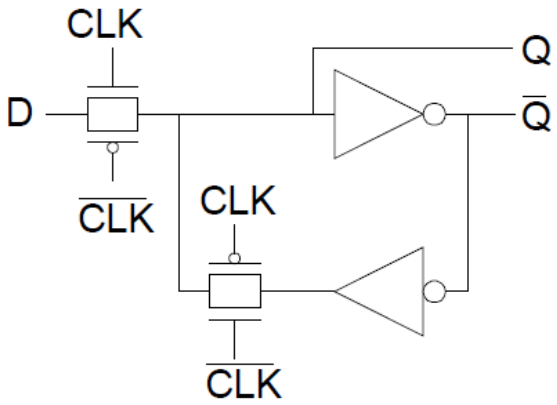


D Latch Operation



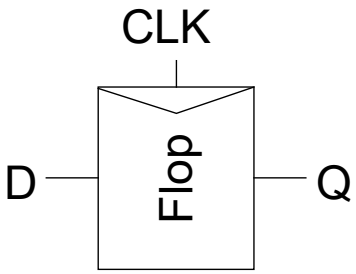


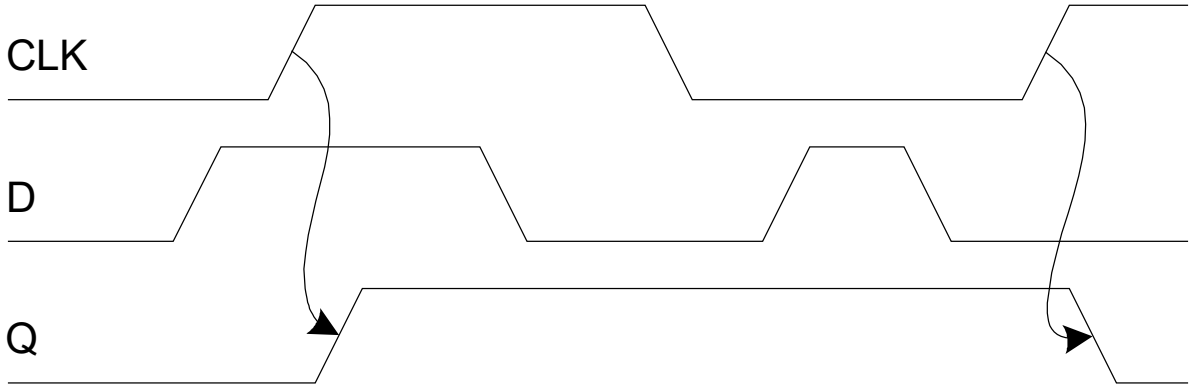




D Flip-flop

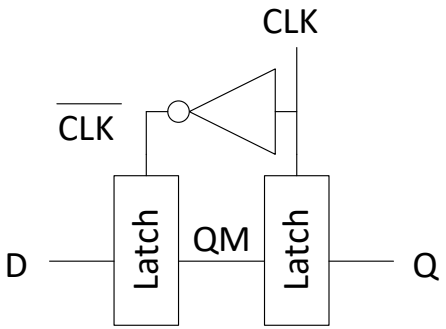
- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop*, *master-slave flip-flop*

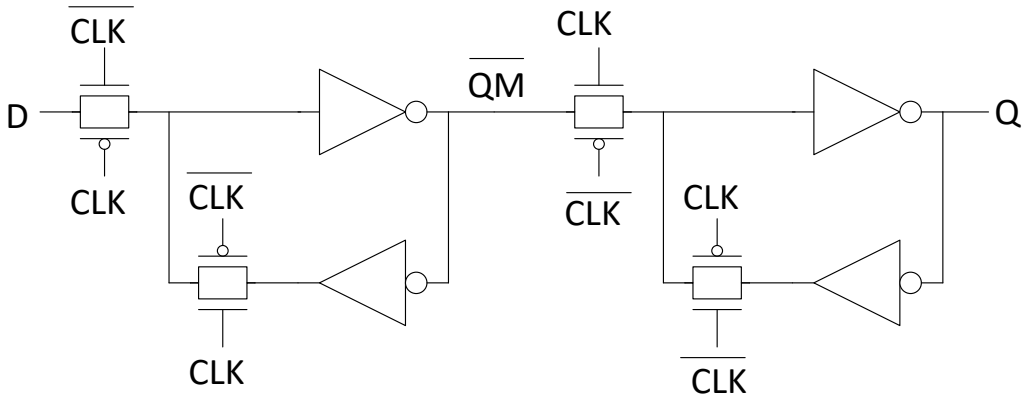




D Flip-flop Design

- Built from master and slave D latches





D Flip-flop Operation