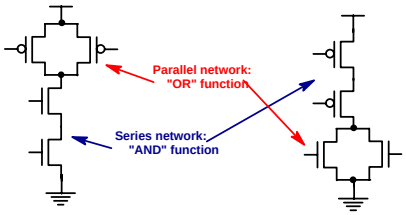
3. Implementing Logic in CMOS

- Last class: Layout, fabrication process, design rules
- This class: Implementing logic functions with CMOS transistors

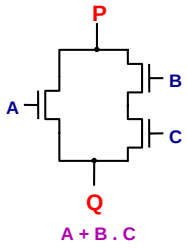
Static CMOS Circuits

- N and P channel **networks** implement logic functions
  - Each network connected between **Output** and **V_{DD} or V_{SS}**



Duality in CMOS Circuits

- N and P networks must implement **complementary** functions
- **Duality** sufficient for correct operation



Dual network? Look at values of A, B and C which will produce a connection between P and Q

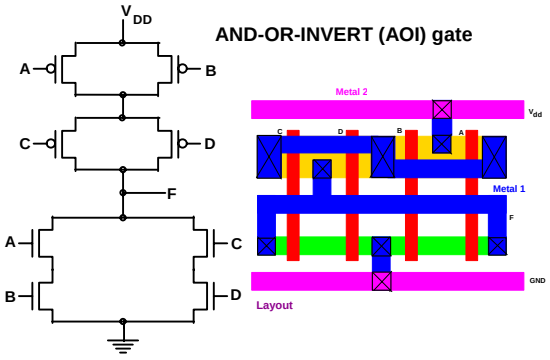
Constructing Complex Gates

- Example: $F = \overline{(A \cdot B) + (C \cdot D)}$
  1. Take **uninverted** function F = (AB + CD) and derive N-network
  2. Identify AND, OR components; F is OR of AB,CD
  3. Make connections of transistors
     - AND ⇔ Series connection, OR ⇔ Parallel

Construction of Complex Gates, Cont'd

4. Construct P-network by taking complement of N-expression (AB +CD), which gives the expression, $(\overline{A} + \overline{B}) \cdot (\overline{C} + \overline{D})$
5. Combine P and N circuits

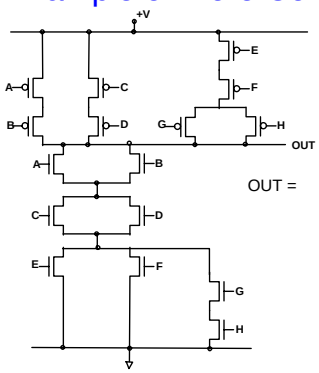
Layout of Complex Gate



Example of Compound Gate

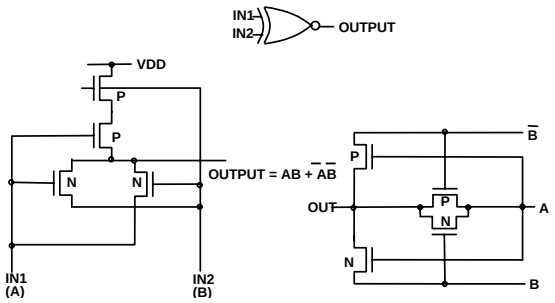
$F = \overline{(A + B + C) \cdot D}$

Example of More Complex Gate

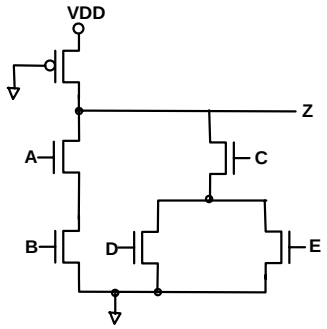


OUT =

Exclusive-NOR Gate in CMOS

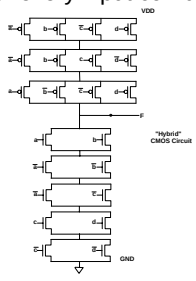


Pseudo nMOS Logic



Duality is not Necessary

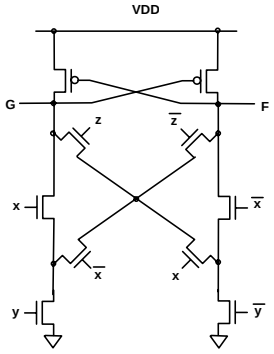
- Functions realized by N and P networks must be complementary, and one of them must conduct for every input combination



F = ab + a'b' + ac + cd + c'd'

The N and P networks are NOT duals, but the switching functions they implement are complementary

Example of Complex CMOS Gate



F =

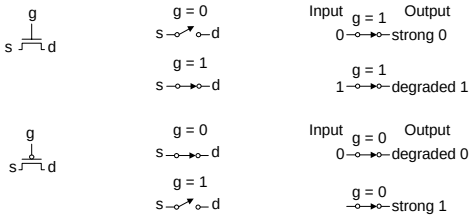
G =

Signal Strength

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

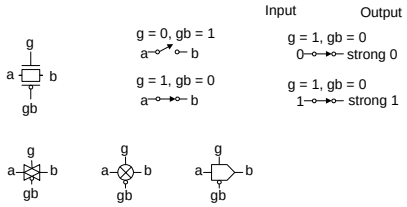
Pass Transistors

- Transistors can be used as switches



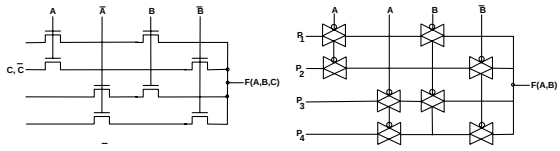
Transmission Gates

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



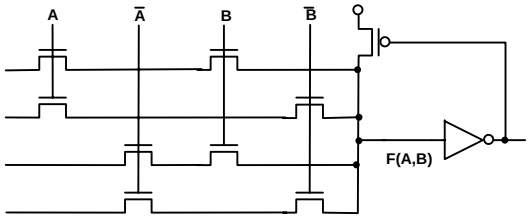
Pass Transistor Logic

- What is the difference between the two circuits?



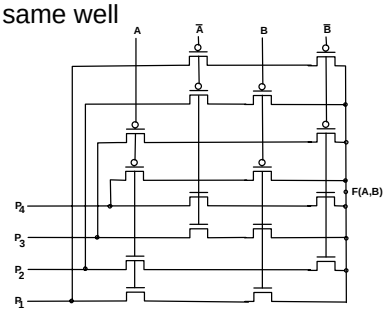
Pass Transistor Logic Pull-Up Version

- How do voltage levels at the output of this gate differ from that of the pass-transistor multiplexer in the previous foil?



Pass Transistor Logic -- Better Layout

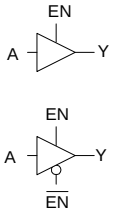
- Group similar transistors, so they can be in the same well



Tristates

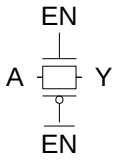
- *Tristate buffer* produces Z when not enabled

EN	A	Y
0	0	
0	1	
1	0	
1	1	



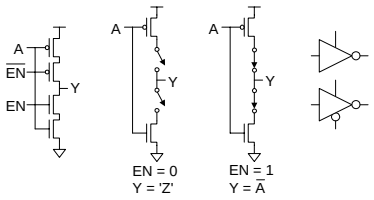
Nonrestoring Tristate

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



Tristate Inverter

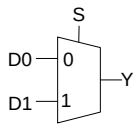
- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output



Multiplexers

- 2:1 *multiplexer* chooses between two inputs

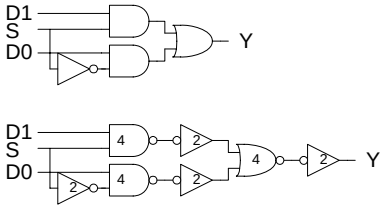
S	D1	D0	Y
0	X	0	
0	X	1	
1	0	X	
1	1	X	



Gate-Level Mux Design

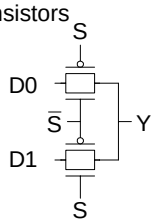
- How many transistors are needed? 20

$Y = SD_1 + \overline{S}D_0$ (too many transistors)



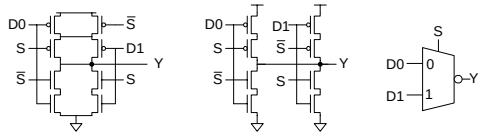
Transmission Gate Mux

- Nonrestoring mux uses two transmission gates
  - Only 4 transistors



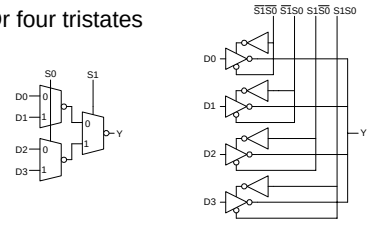
Inverting Mux

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



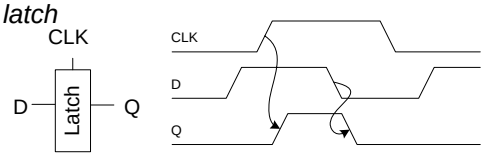
4:1 Multiplexer

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



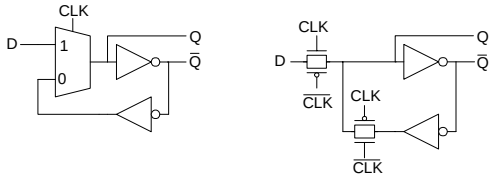
D Latch

- 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*

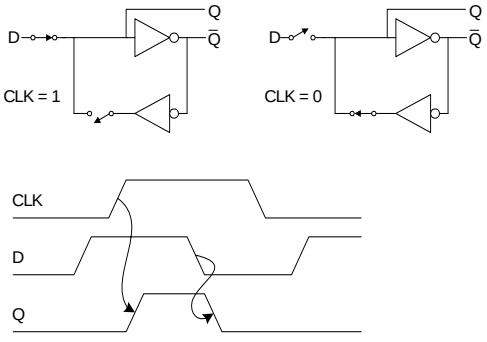


D Latch Design

- Multiplexer chooses D or old Q

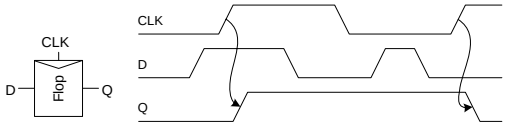


D Latch Operation



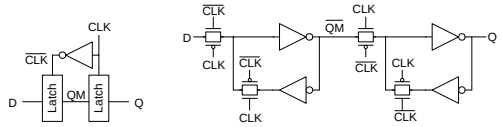
D Flip-flop

- 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*

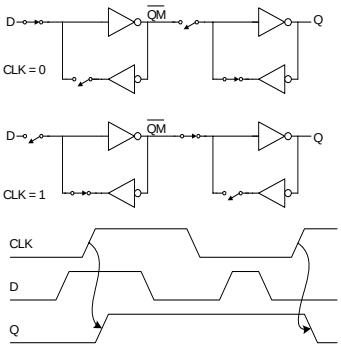


D Flip-flop Design

- Built from master and slave D latches

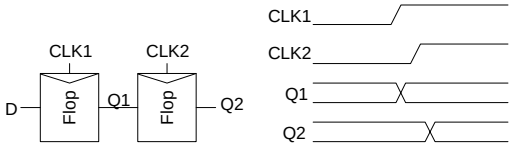


D Flip-flop Operation



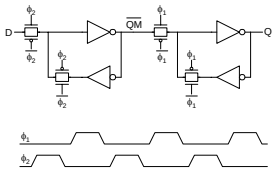
Race Condition

- Back-to-back flops can malfunction from clock skew
  - Second flip-flop fires late
  - Sees first flip-flop change and captures its result
  - Called *hold-time failure* or *race condition*



Nonoverlapping Clocks

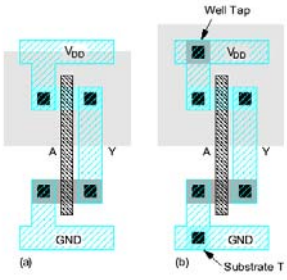
- Nonoverlapping clocks can prevent races
  - As long as nonoverlap exceeds clock skew
- Use in safe (conservative) designs
  - Industry manages skew more carefully instead



Gate Layout

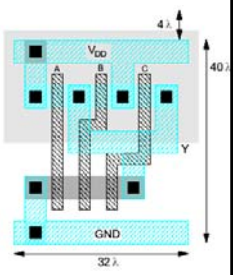
- Layout can be very time consuming
  - Design gates to fit together nicely
  - Build a library of standard cells
- Standard cell design methodology
  - V_{DD} and GND should abut (standard height)
  - Adjacent gates should satisfy design rules
  - nMOS at bottom and pMOS at top
  - All gates include well and substrate contacts

Example: Inverter



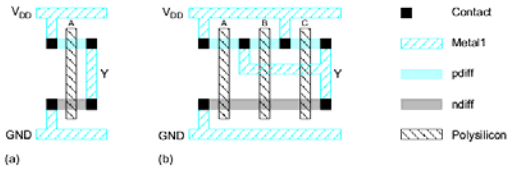
Example: NAND3

- Horizontal N-diffusion and p-diffusion strips
- Vertical polysilicon gates
- Metal1 V_{DD} rail at top
- Metal1 GND rail at bottom
- 32 λ by 40 λ



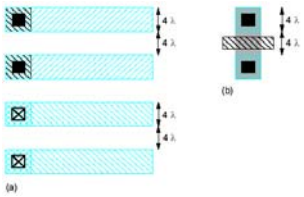
Stick Diagrams

- *Stick diagrams* help plan layout quickly
  - Need not be to scale
  - Draw with color pencils or dry-erase markers



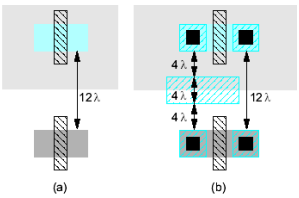
Wiring Tracks

- A *wiring track* is the space required for a wire
  - 4 λ width, 4 λ spacing from neighbor = 8 λ pitch
- Transistors also consume one wiring track



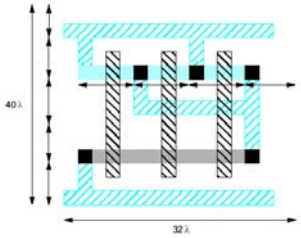
Well spacing

- Wells must surround transistors by 6 λ
  - Implies 12 λ between opposite transistor flavors
  - Leaves room for one wire track



Area Estimation

- Estimate area by counting wiring tracks
  - Multiply by 8 to express in λ

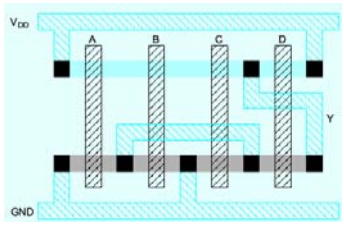


Example: O3AI

- Sketch a stick diagram for O3AI and estimate area
  - $Y = \overline{(A + B + C) \cdot D}$

Example: O3AI

- Sketch a stick diagram for O3AI and estimate area
  - $Y = \overline{(A + B + C) \cdot D}$



Example: O3AI

- Sketch a stick diagram for O3AI and estimate area
  - $Y = \overline{(A + B + C) \cdot D}$