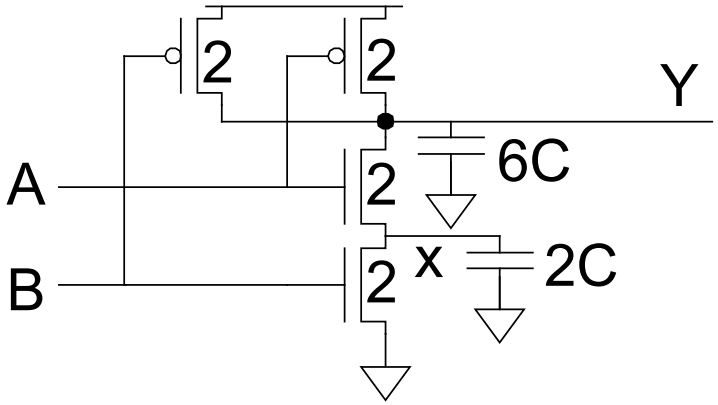
Combinational Circuits

Outline

- Input Ordering
- Symmetric Gates
- Asymmetric Gates
- Skewed Gates
- Circuit Families

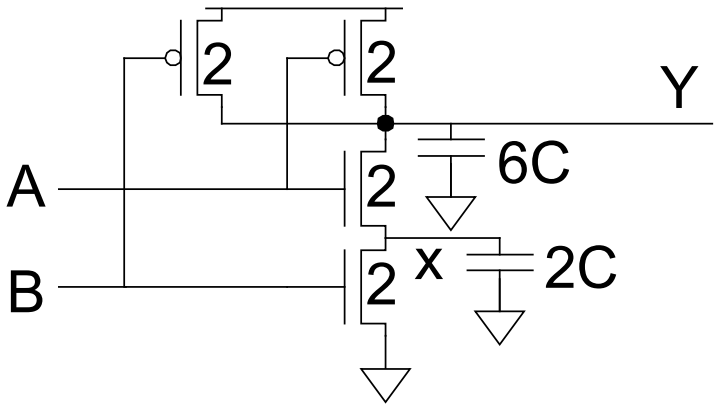
Input Order

- Our parasitic delay model was too simple to calculate parasitic delay for Y falling if
  - B arrives later than A?
  - A arrives later than B?



Input Order

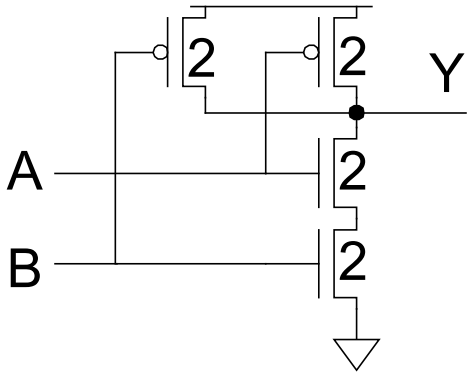
- Calculate parasitic delay for Y falling
  - If B arrives later?
    t_{pdf}=R/2*(2C)+(R/2+R/2)*6C=7RC=2.33τ
  - If A arrives later? t_{pdf}=(R/2+R/2)*6C=6RC= 2τ



Inner & Outer Inputs

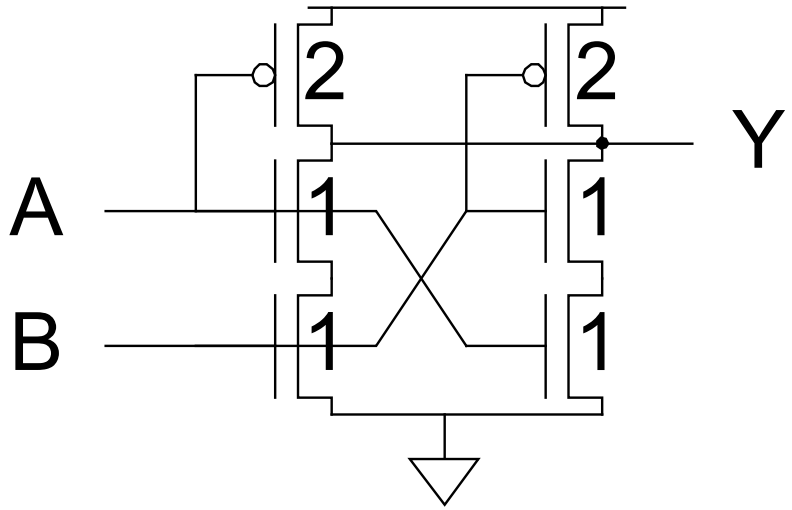
- ***Outer*** input is closest to rail (B) (Vdd/gnd rail)
- ***Inner*** input is closest to output (A)

- If input arrival time is known
  - Connect latest input to inner terminal



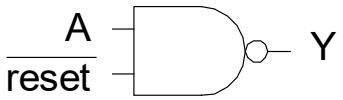
Symmetric Gates

- Inputs can be made perfectly symmetric by connecting each input to inner and outer

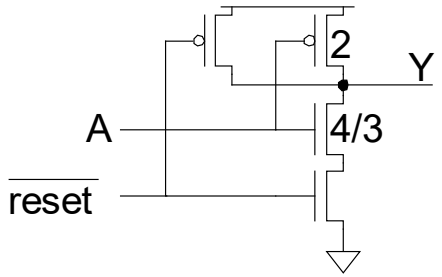


Asymmetric Gates

- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same



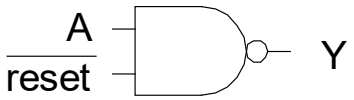
- g_A =
- g_B =
- $g_{total} = g_A + g_B$ =



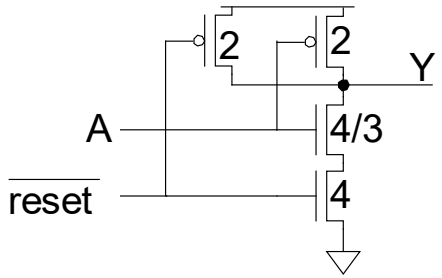
- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up

Asymmetric Gates

- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same



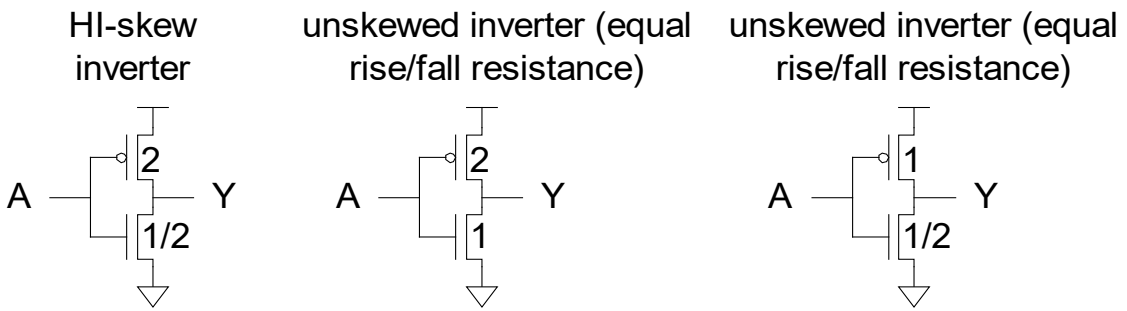
- $g_A = 10/9$
- $g_{reset} = 2$
- $g_{total} = g_A + g_{reset} = 28/9$



- Asymmetric gate approaches **g = 1 on critical input**
- But total logical effort goes up

Skewed Gates

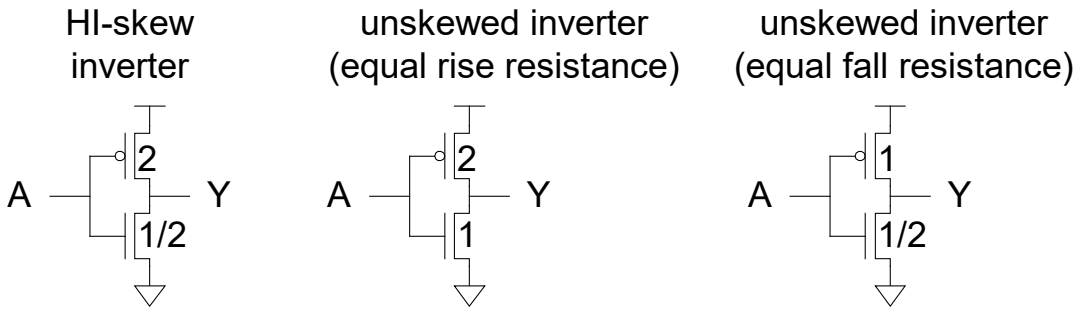
- Skewed gates favor one edge over another
- Ex: suppose rising output of inverter is most critical
  - Downsize noncritical nMOS transistor



- Calculate logical effort by comparing skewed gate to unskewed inverter with same effective resistance on that edge.
  - g_u =
  - g_d =

Skewed Gates

- Skewed gates favor one edge over another
- Ex: suppose rising output of inverter is most critical
  - Downsize noncritical nMOS transistor

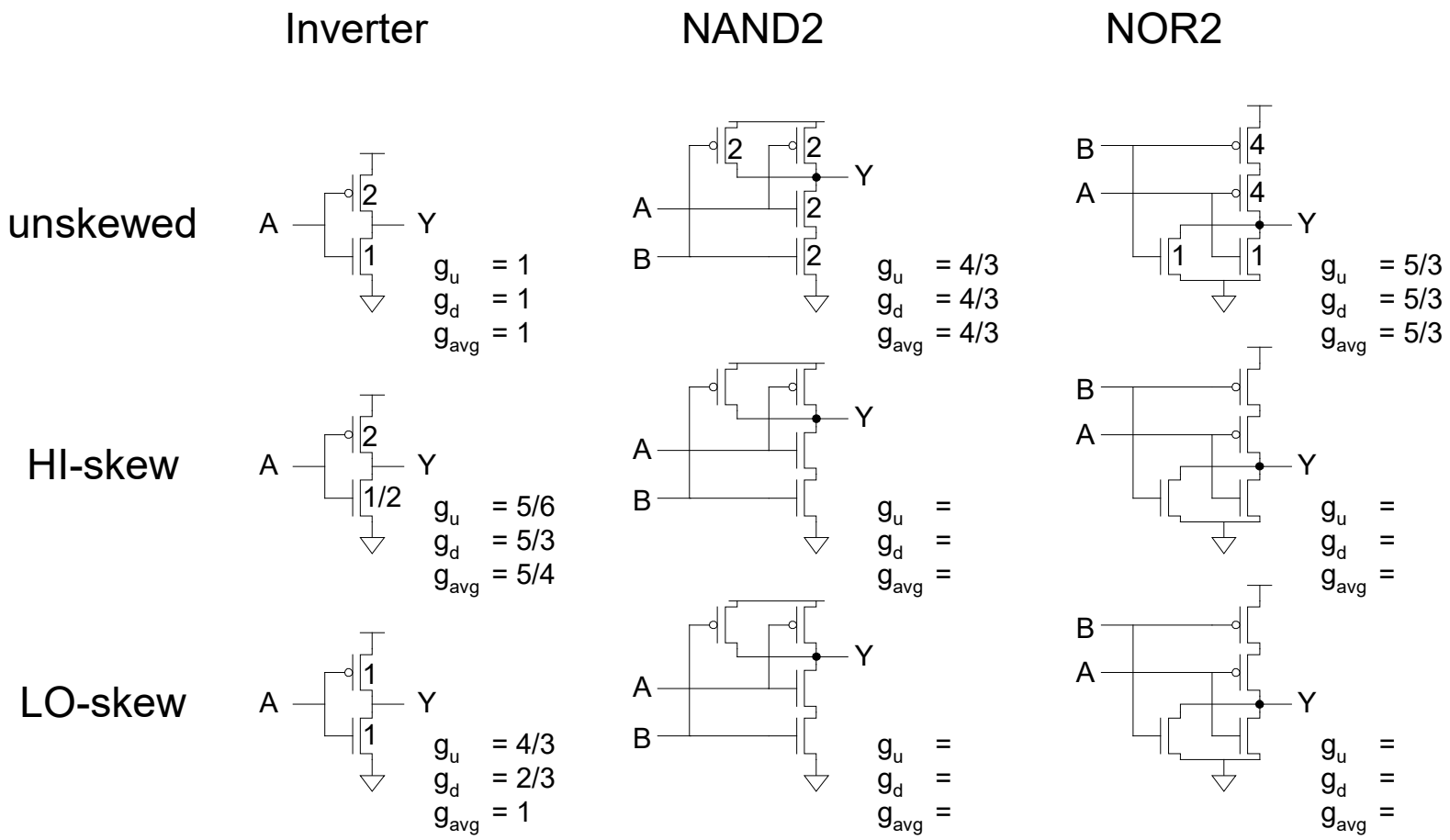


- Calculate logical effort by comparing skewed to unskewed inverter with same effective resistance on that edge.
  - $g_u = 2.5 / 3 = 5/6$ up logical effort
  - $g_d = 2.5 / 1.5 = 5/3$ down logical effort

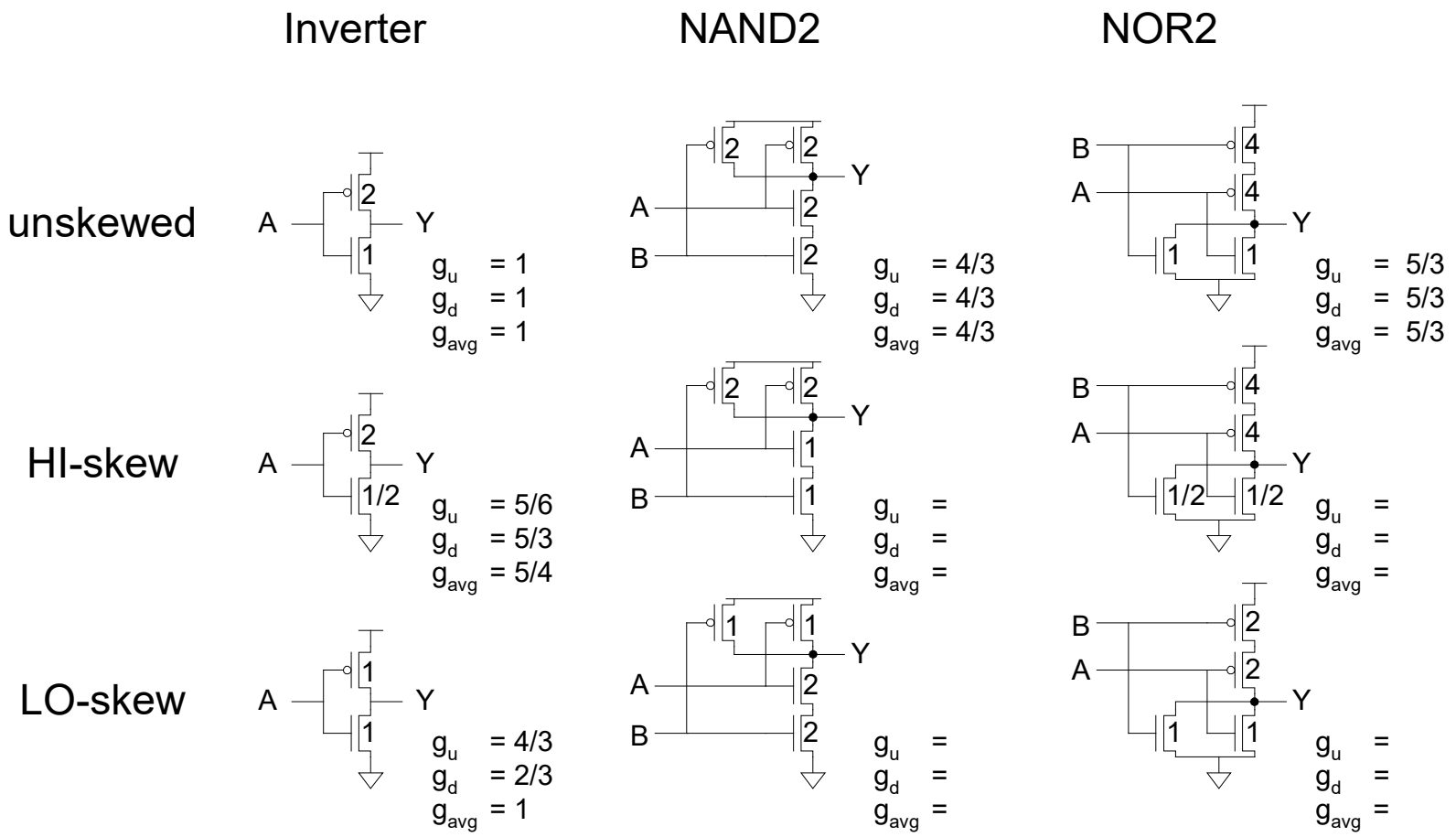
HI- and LO-Skew Summary

- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.
- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small nMOS)
  - LO-skew gates favor falling output (small pMOS)
- Logical effort is smaller for favored direction
- But larger for the other direction

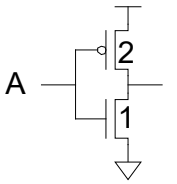
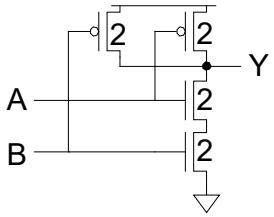
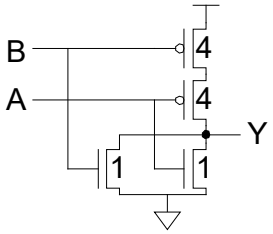
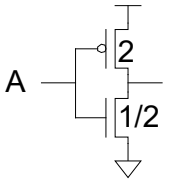
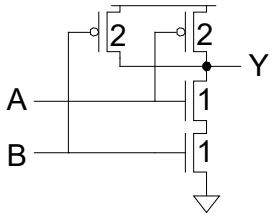
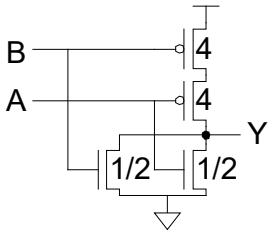
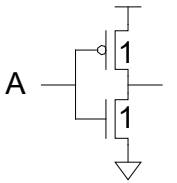
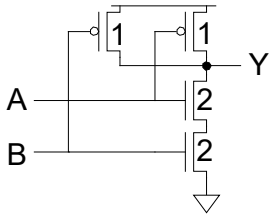
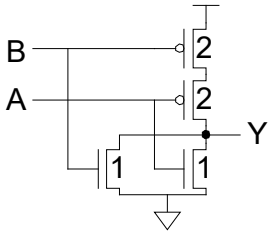
Catalog of Skewed Gates



Catalog of Skewed Gates

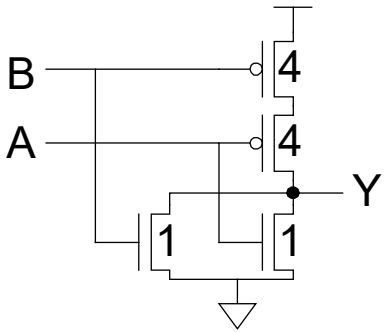


Catalog of Skewed Gates

	Inverter	NAND2	NOR2
unskewed	 $g_u = 1$ $g_d = 1$ $g_{avg} = 1$	 $g_u = 4/3$ $g_d = 4/3$ $g_{avg} = 4/3$	 $g_u = 5/3$ $g_d = 5/3$ $g_{avg} = 5/3$
HI-skew	 $g_u = 5/6$ $g_d = 5/3$ $g_{avg} = 5/4$	 $g_u = 1$ $g_d = 2$ $g_{avg} = 3/2$	 $g_u = 3/2$ $g_d = 3$ $g_{avg} = 9/4$
LO-skew	 $g_u = 4/3$ $g_d = 2/3$ $g_{avg} = 1$	 $g_u = 2$ $g_d = 1$ $g_{avg} = 3/2$	 $g_u = 2$ $g_d = 1$ $g_{avg} = 3/2$

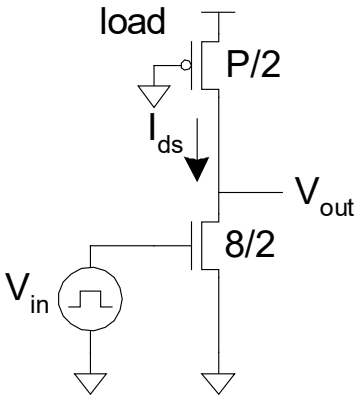
Circuit Families Introduction

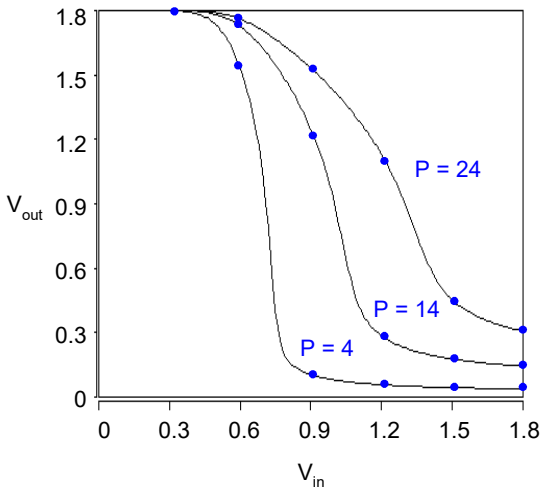
- What makes a circuit fast?
  - $I = C\,dV/dt \rightarrow t_{pd} \propto (C/I)\,\Delta V$
  - low capacitance
  - high current
  - small swing
- Logical effort is proportional to C/I
- pMOS are the enemy!
  - High capacitance for a given current
- Can we take the pMOS capacitance off the input?
- Various circuit families try to do this…



Pseudo-nMOS

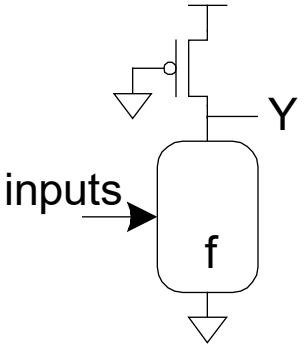
- In the old days, nMOS processes had no pMOS
  - Instead, use pull-up transistor that is always ON
- In CMOS, use a pMOS that is always ON
  - *Ratio* issue
  - Make pMOS about ¼ effective strength of pulldown network (1/3—1/6)



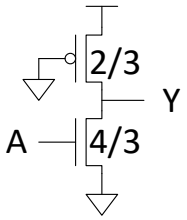


Pseudo-nMOS Gates

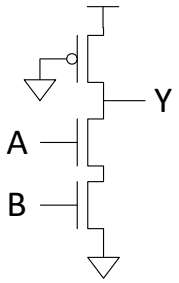
- Design for unit current on output to compare with unit inverter.
- pMOS fights nMOS



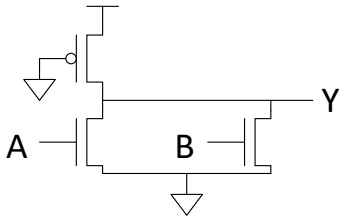
Inverter



NAND2

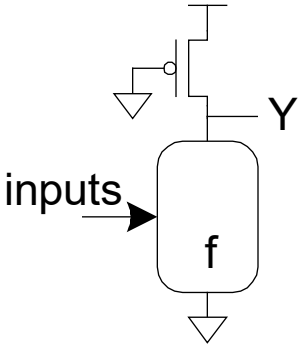


NOR2

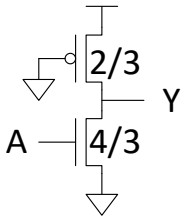


Pseudo-nMOS Gates

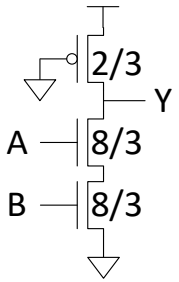
- Design for unit current on output to compare with unit inverter.
- pMOS fights nMOS



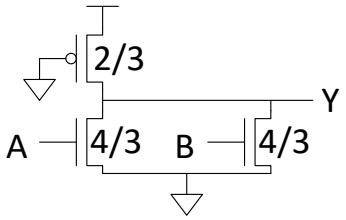
Inverter



NAND2

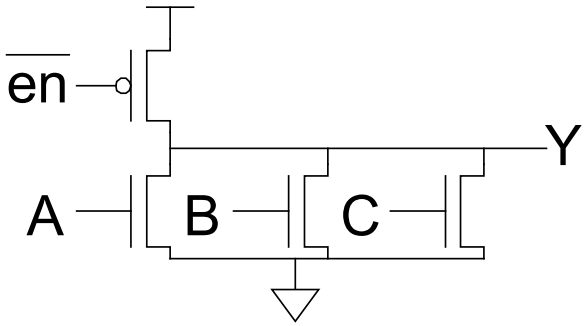


NOR2



Pseudo-nMOS Power

- Pseudo-nMOS draws power whenever Y = 0
  - Called static power $P = I \cdot V_{DD}$
  - A few uA / gate * 1M gates would be a problem
- Use pseudo-nMOS sparingly for wide NORs
- Turn off pMOS when not in use



Ratio Example

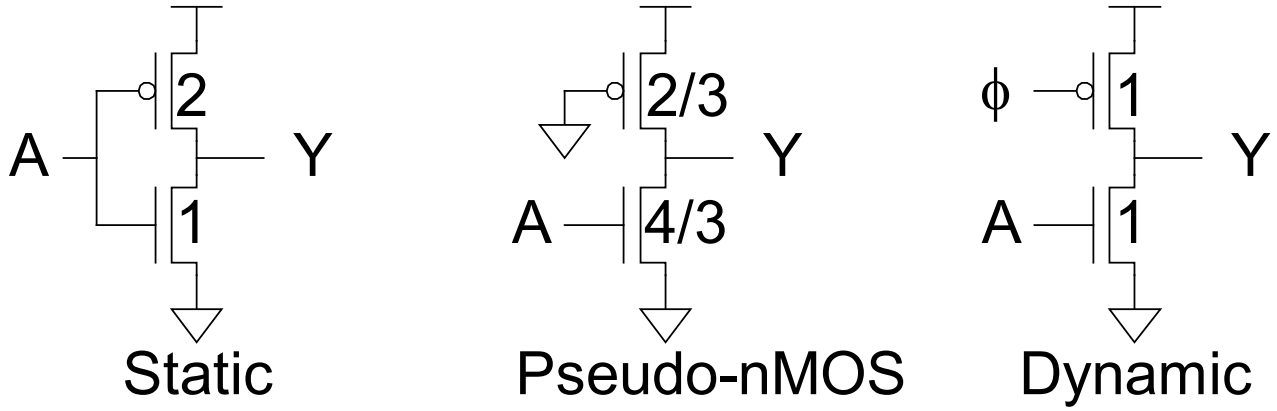
- The chip contains a 32 word x 48 bit ROM
  - Uses pseudo-nMOS decoder and bitline pullups
  - On average, one wordline and 24 bitlines are high
- Find static power drawn by the ROM
  - $I_{on\text{-}p}$ = 36 μA, V_{DD} = 1.0 V
- Solution:

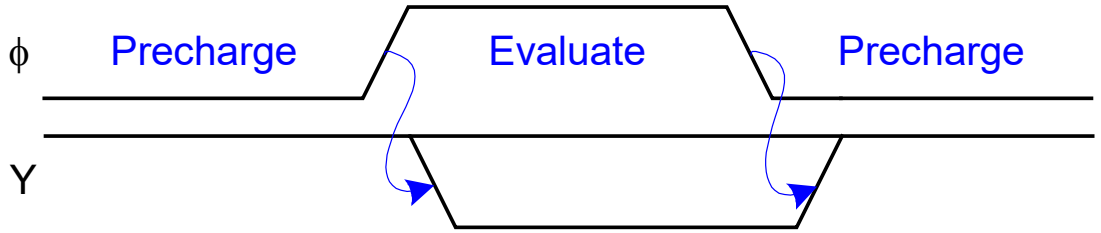
$$P_{\text{pull-up}} = V_{DD} I_{\text{pull-up}} = 36\ \mu\text{W}$$

$$P_{\text{static}} = (31+24) P_{\text{pull-up}} = 1.98\ \text{mW}$$

Dynamic Logic

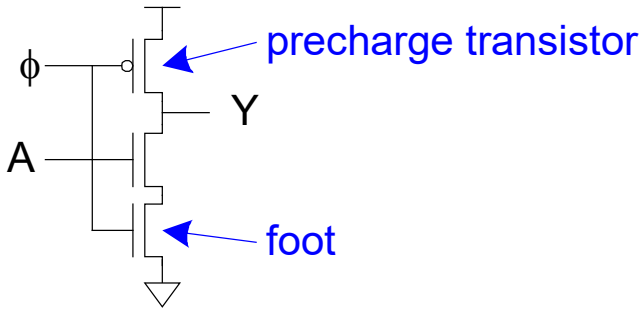
- *Dynamic* gates uses a clocked pMOS pullup
- Two modes: *precharge* and *evaluate*

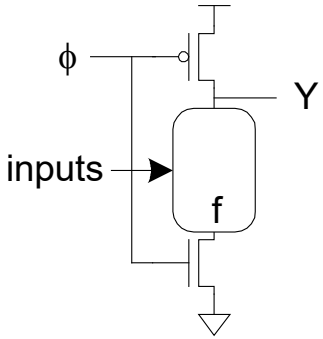




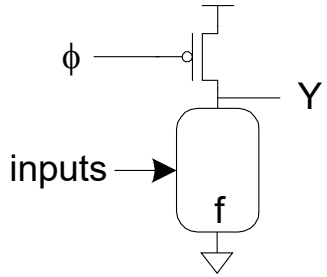
The Foot

- What if pulldown network is ON during precharge?
- Use series evaluation transistor to prevent fight.



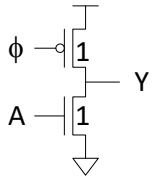
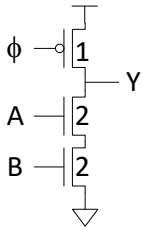
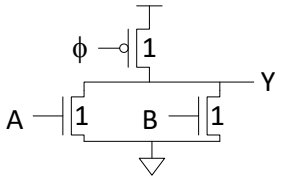
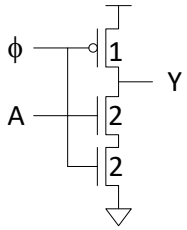
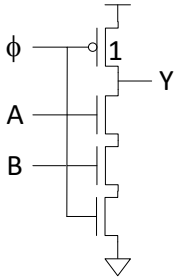
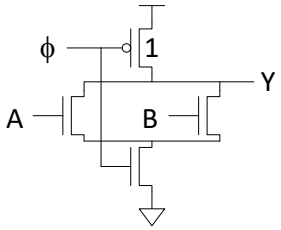


footed

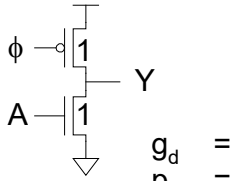
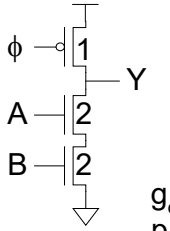
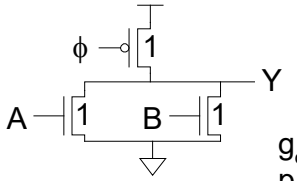
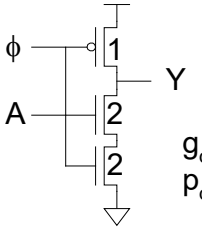
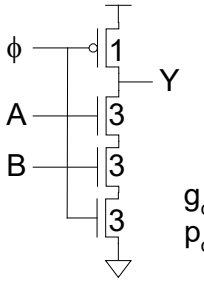
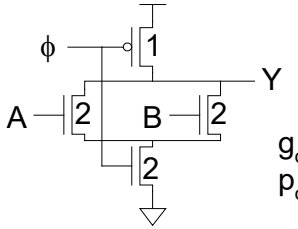


unfooted

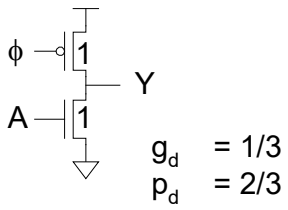
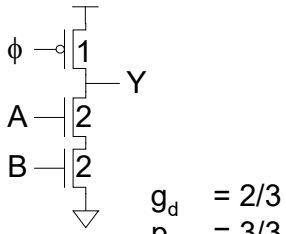
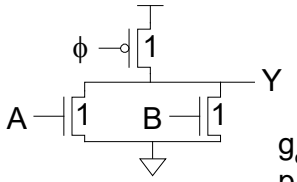
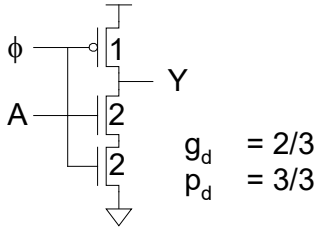
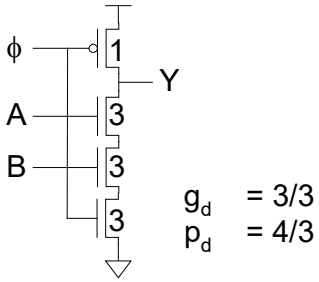
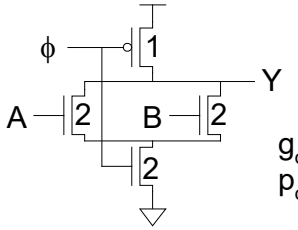
Dynamic Logical Effort

	Inverter	NAND2	NOR2
unfooted			
footed			

Dynamic Logical Effort

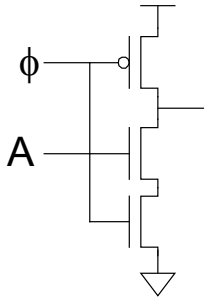
	Inverter	NAND2	NOR2
unfooted	 g_d = p_d =	 g_d = p_d =	 g_d = p_d =
footed	 g_d = p_d =	 g_d = p_d =	 g_d = p_d =

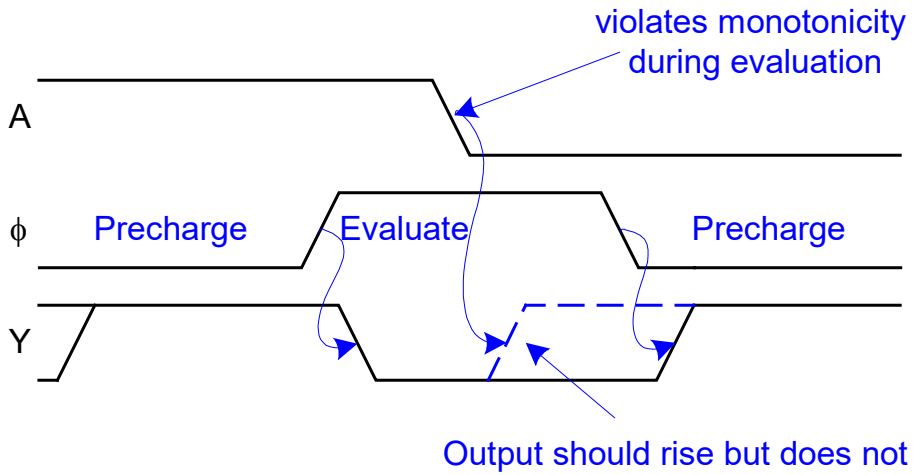
Dynamic Logical Effort

	Inverter	NAND2	NOR2
unfooted	 $g_d = 1/3$ $p_d = 2/3$	 $g_d = 2/3$ $p_d = 3/3$	 $g_d = 1/3$ $p_d = 3/3$
footed	 $g_d = 2/3$ $p_d = 3/3$	 $g_d = 3/3$ $p_d = 4/3$	 $g_d = 2/3$ $p_d = 5/3$

Monotonicity

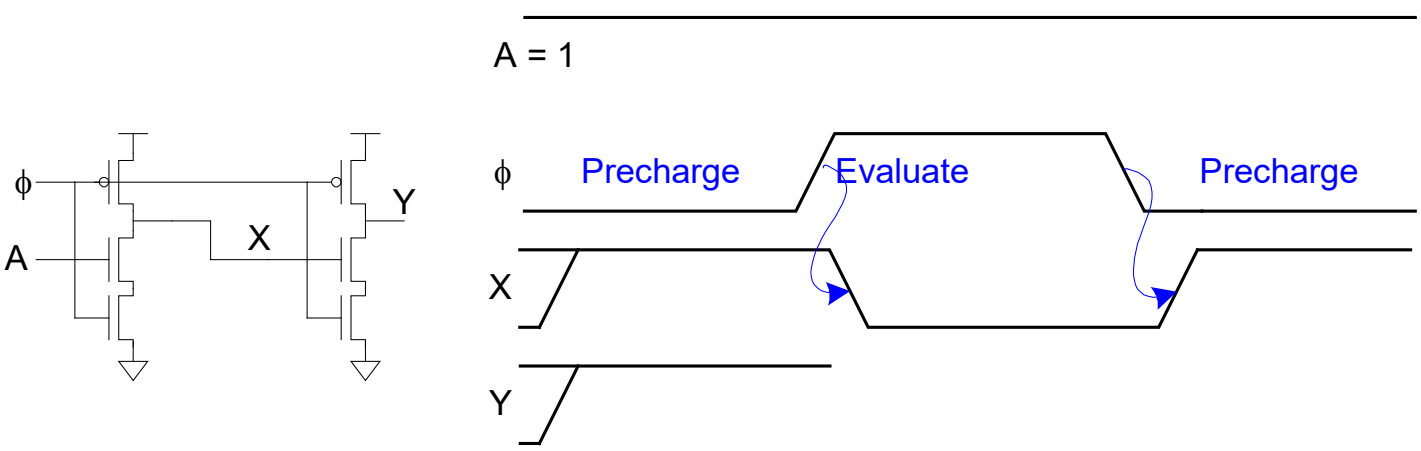
- Dynamic gates require *monotonically rising* inputs during evaluation
  - 0 -> 0
  - 0 -> 1
  - 1 -> 1
  - But not 1 -> 0





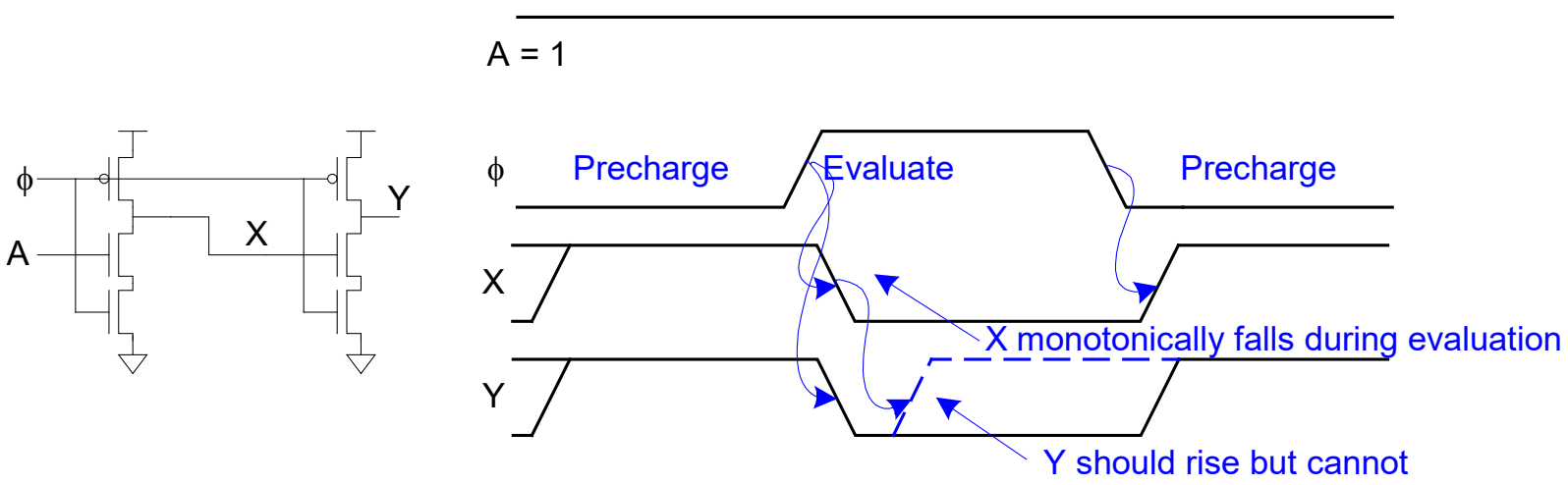
Monotonicity Woes

- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!



Monotonicity Woes

- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!



Domino Gates

- Follow dynamic stage with inverting static gate
  - Dynamic / static pair is called domino gate
  - Produces monotonic outputs