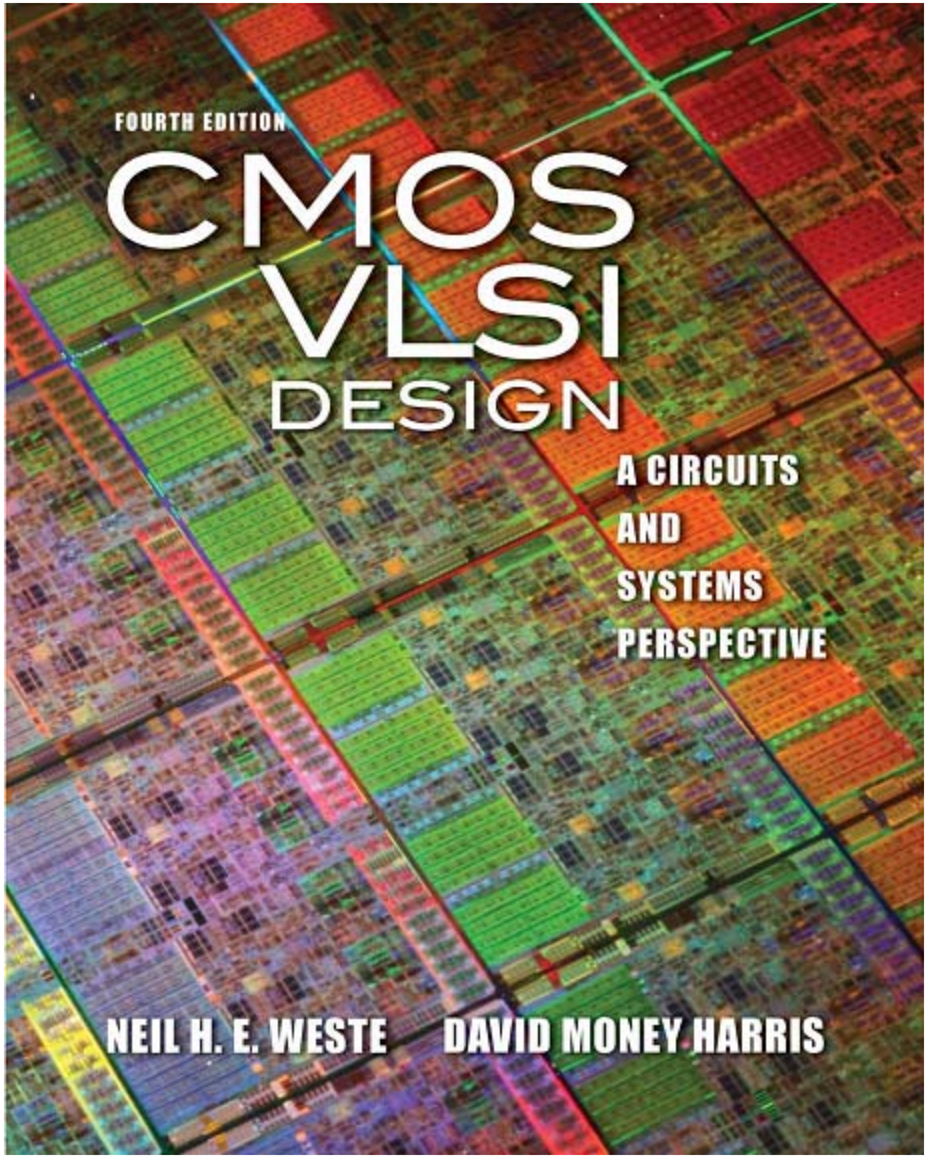


Lecture 9: Combinational Circuit Design

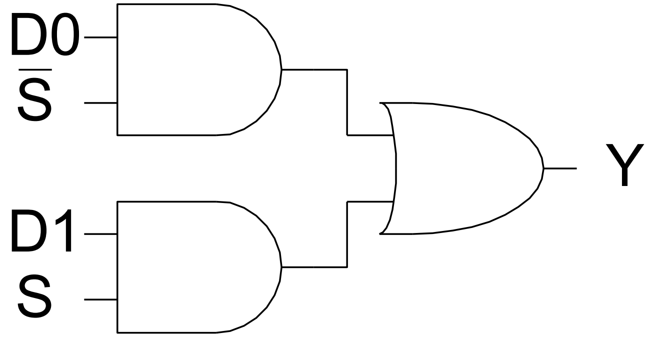
Outline

- Bubble Pushing
- Compound Gates
- Logical Effort Example
- Input Ordering
- Asymmetric Gates
- Skewed Gates
- Best P/N ratio

Example 1

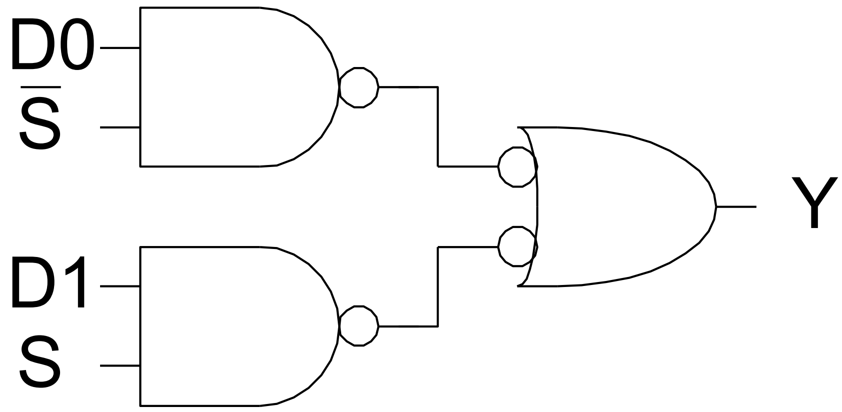
```
module mux(input  s, d0, d1,
              output y);

   assign y = s ? d1 : d0;
endmodule
```

1) Sketch a design using AND, OR, and NOT gates.



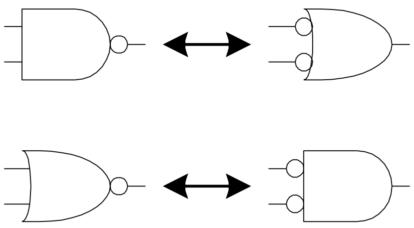
Example 2

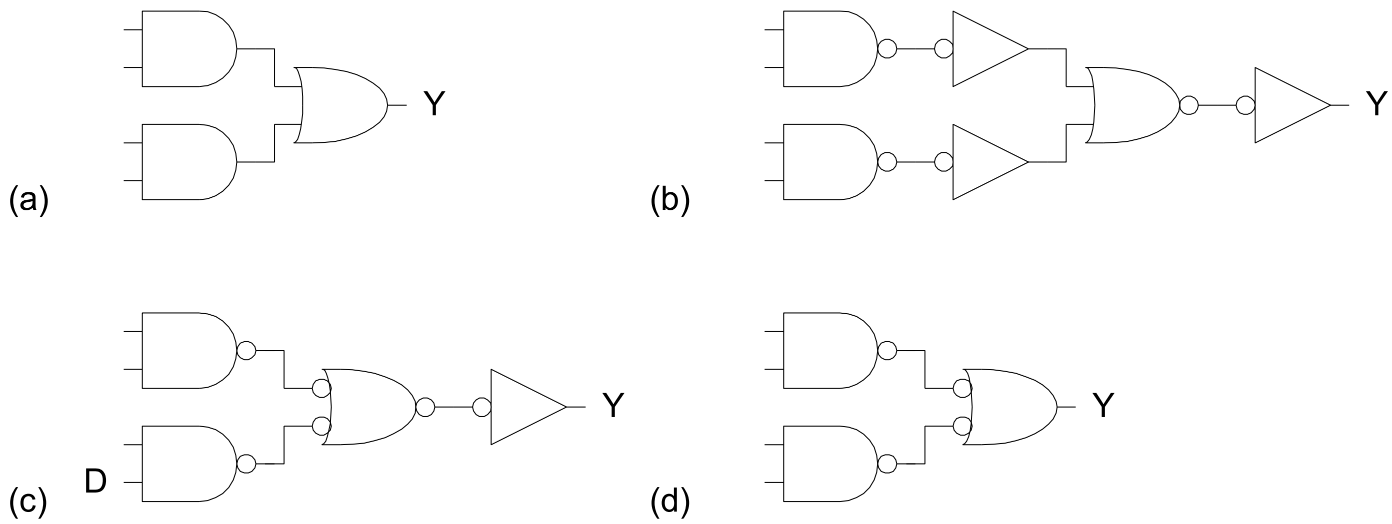
2) Sketch a design using NAND, NOR, and NOT gates.
Assume ~S is available.



Bubble Pushing

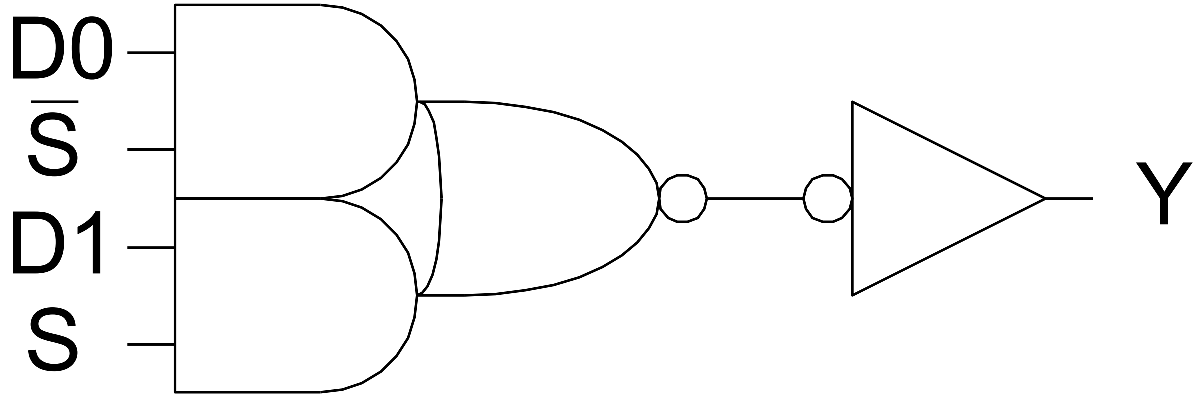
- Start with network of AND / OR gates
- Convert to NAND / NOR + inverters
- Push bubbles around to simplify logic
  - Remember DeMorgan's Law





Example 3

3) Sketch a design using one compound gate and one NOT gate. Assume ~S is available.



Compound Gates

- Logical Effort of compound gates

unit inverter

$$Y = \overline{A}$$

AOI21

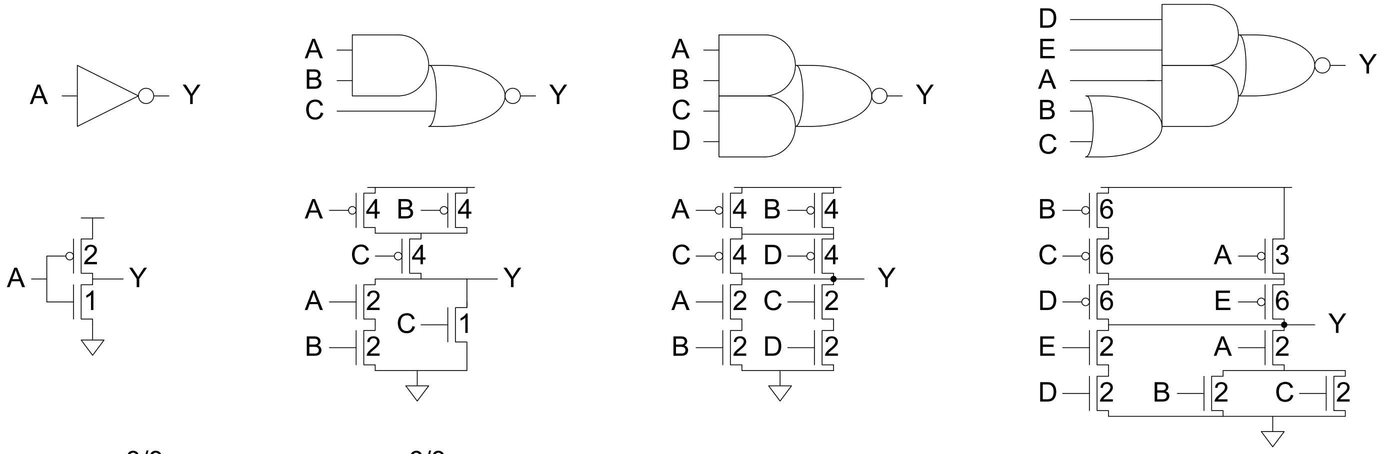
$$Y = \overline{A \bullet B + C}$$

AOI22

$$Y = \overline{A \bullet B + C \bullet D}$$

Complex AOI

$$Y = \overline{A \bullet (B + C) + D \bullet E}$$



unit inverter:

$g_A = 3/3$

$p = 3/3$

AOI21:

$g_A = 6/3$

$g_B = 6/3$

$g_C = 5/3$

$p = 7/3$

AOI22:

$g_A =$

$g_B =$

$g_C =$

$g_D =$

$p =$

Complex AOI:

$g_A =$

$g_B =$

$g_C =$

$g_D =$

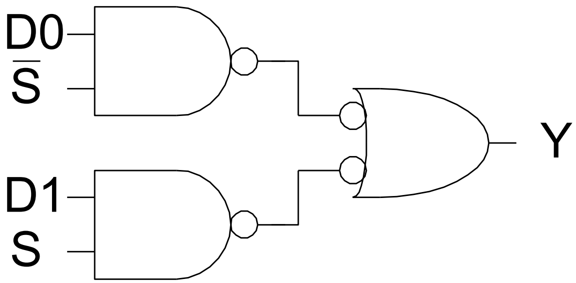
$g_E =$

$p =$

Example 4

- The multiplexer has a maximum input capacitance of 16 units on each input. It must drive a load of 160 units. Estimate the delay of the two designs.

H = B = N =



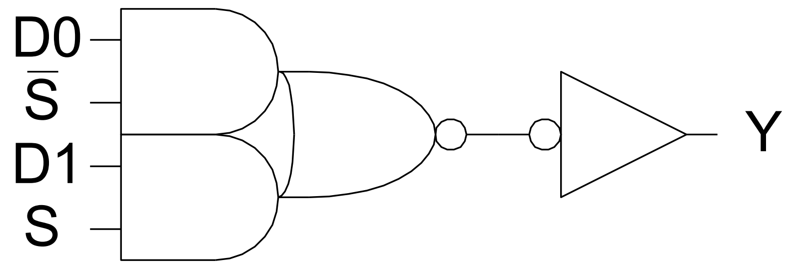
$P =$

$G =$

$F =$

$\hat{f} =$

$D =$



$P =$

$G =$

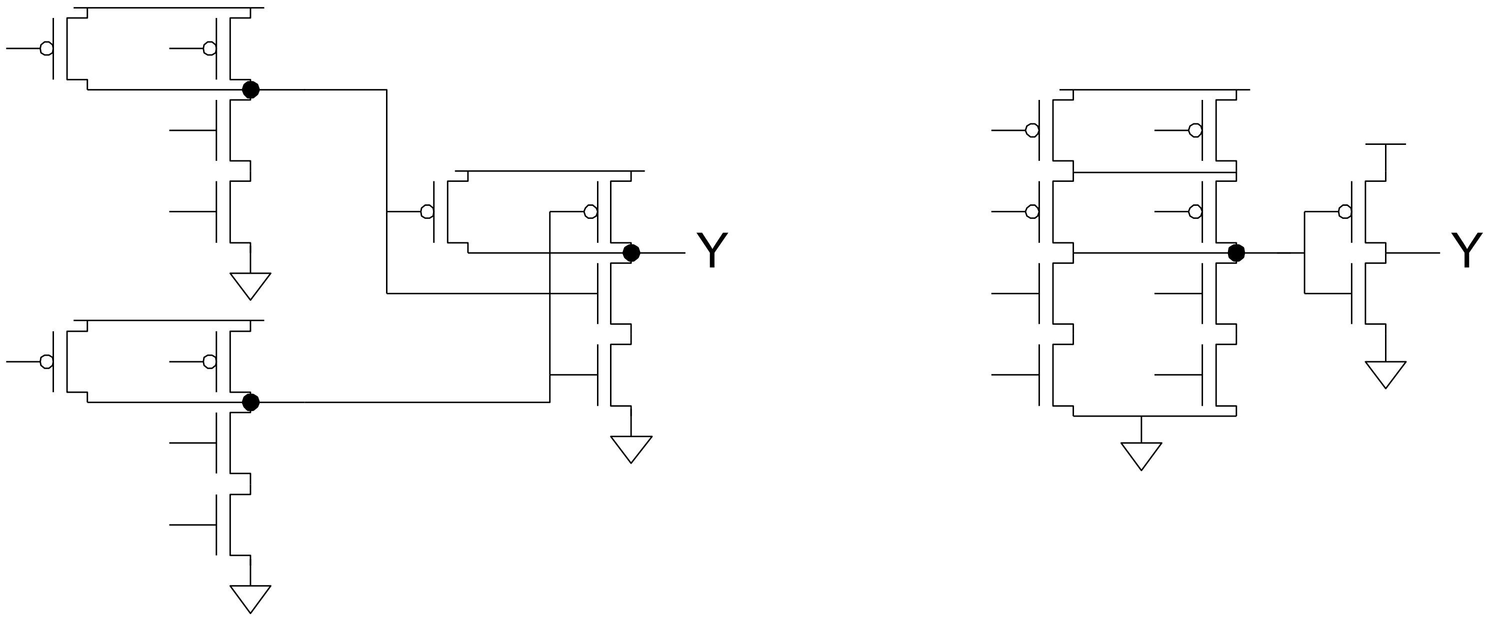
$F =$

$\hat{f} =$

$D =$

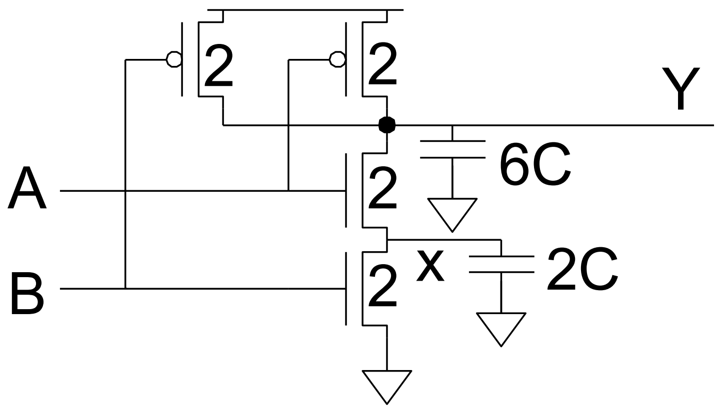
Example 5

- Annotate your designs with transistor sizes that achieve this delay.



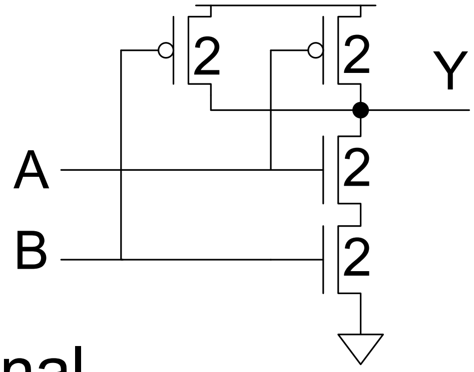
Input Order

- Our parasitic delay model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest?
    - If B arrives latest?



Inner & Outer Inputs

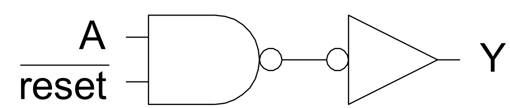
- *Inner* input is closest to output (A)
- *Outer* input is closest to rail (B)

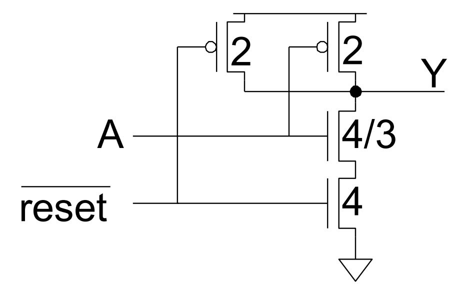


- If input arrival time is known
  - Connect latest input to inner terminal

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

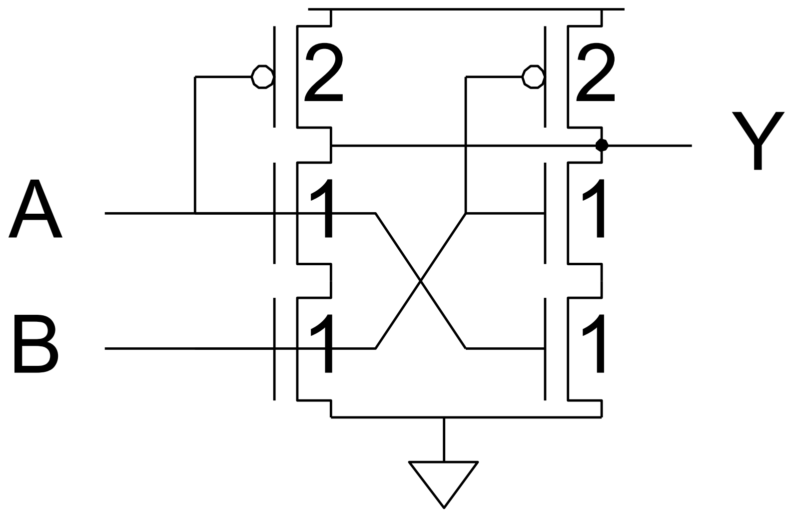




- $g_A =$
- $g_B =$
- $g_{total} = g_A + g_B =$
- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up

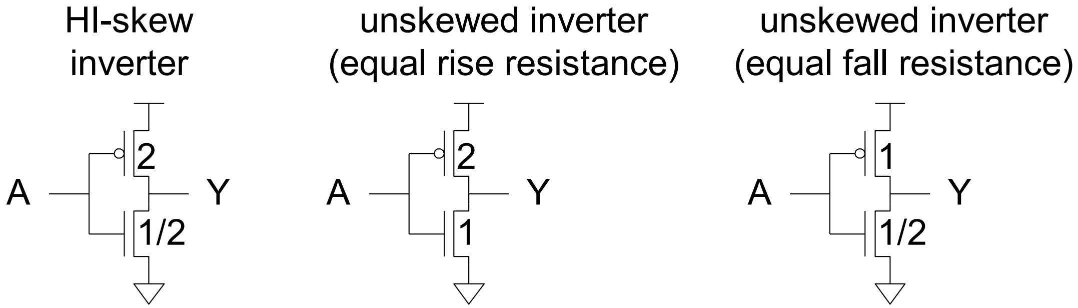
Symmetric Gates

- Inputs can be made perfectly symmetric



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 to unskewed inverter with same effective resistance on that edge.
  - g_u =
  - g_d =

HI- and LO-Skew

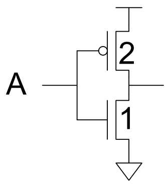
- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.
- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor 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

Inverter | NAND2 | NOR2

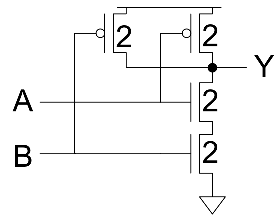
unskewed

Inverter



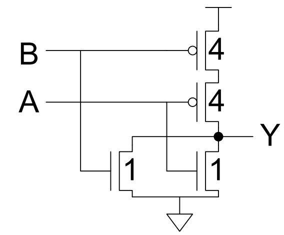
g_u = 1
g_d = 1
g_{avg} = 1

NAND2



g_u = 4/3
g_d = 4/3
g_{avg} = 4/3

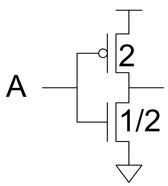
NOR2



g_u = 5/3
g_d = 5/3
g_{avg} = 5/3

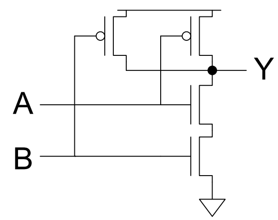
HI-skew

Inverter



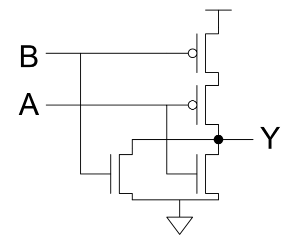
g_u = 5/6
g_d = 5/3
g_{avg} = 5/4

NAND2



g_u =
g_d =
g_{avg} =

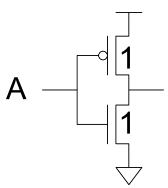
NOR2



g_u =
g_d =
g_{avg} =

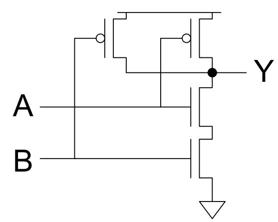
LO-skew

Inverter



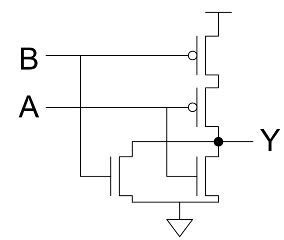
g_u = 4/3
g_d = 2/3
g_{avg} = 1

NAND2



g_u =
g_d =
g_{avg} =

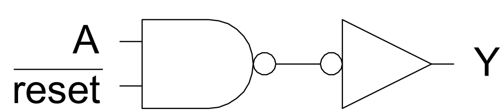
NOR2

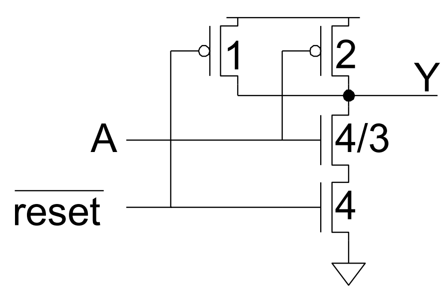


g_u =
g_d =
g_{avg} =

Asymmetric Skew

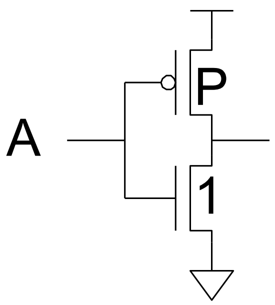
- Combine asymmetric and skewed gates
  - Downsize noncritical transistor on unimportant input
  - Reduces parasitic delay for critical input





Best P/N Ratio

- We have selected P/N ratio for unit rise and fall resistance (μ = 2-3 for an inverter).
- Alternative: choose ratio for least average delay
- Ex: inverter
  - Delay driving identical inverter
  - t_{pdf} =
  - t_{pdr} =
  - t_{pd} =
  - dt_{pd} / dP =
  - Least delay for P =

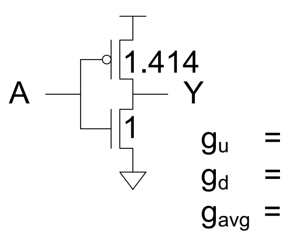


P/N Ratios

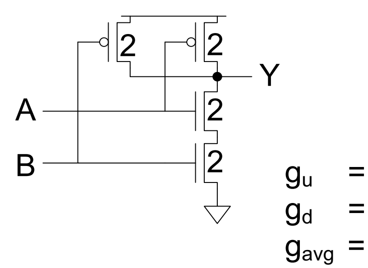
- In general, best P/N ratio is sqrt of equal delay ratio.
  - Only improves average delay slightly for inverters
  - But significantly decreases area and power

fastest P/N ratio

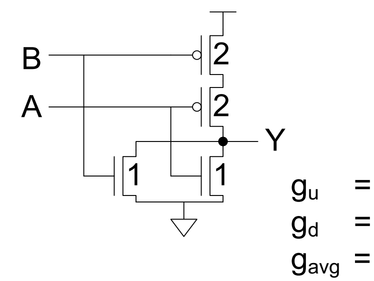
Inverter



NAND2



NOR2



Observations

- For speed:
  - NAND vs. NOR
  - Many simple stages vs. fewer high fan-in stages
  - Latest-arriving input
- For area and power:
  - Many simple stages vs. fewer high fan-in stages