Nadar Saraswathi College of Engineering & Technology

NSCET E-LEARNING PRESENTATION

LISTEN … LEARN… LEAD…

ELECTRONICS AND COMMUNICATION ENGINEERING

III YEAR / VI th SEMESTER

EC8095– VLSI DESIGN

Ms.Vivitha,M.E
Assistant Professor
Nadar Saraswathi College of & Technology,
Vadapudupatti, Annanji (po), Theni – 625531.

UNIT II COMBINATIONAL MOS LOGIC CIRCUITS– LECTURE 02

“A basic truth that the history of the creation of the transistor reveals is that the foundations of transistor electronics were created by making errors and following hunches that failed to give what was expected. ...”

-William Shockley

01
INTRODUCTION

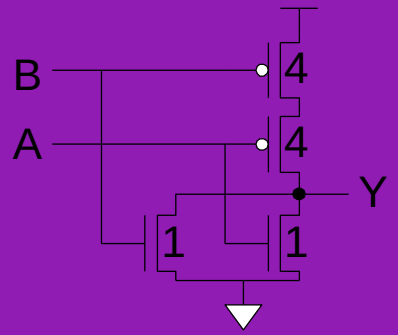
02
CIRCUIT
FAMILIES

03
POWER

01. INTRODUCTION

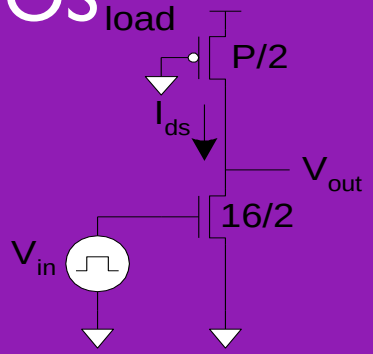
COMBINATIONAL CIRCUIT

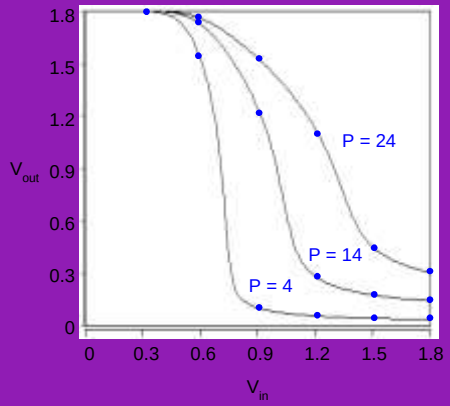
- What makes a circuit fast?
  - $I = C\,dV/dt \quad \rightarrow \quad t_{pd} \propto (C/I)\,\Delta V$
  - low capacitance
  - high current
  - small swing
- Logical effort is proportional to C/I
- pMOS are the enemy!
  - High capacitance for a given current
- Can we take the pMOS capacitance off the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



Pseudo-nMOS

- In the old days, nMOS processes had no pMOS load
  - 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



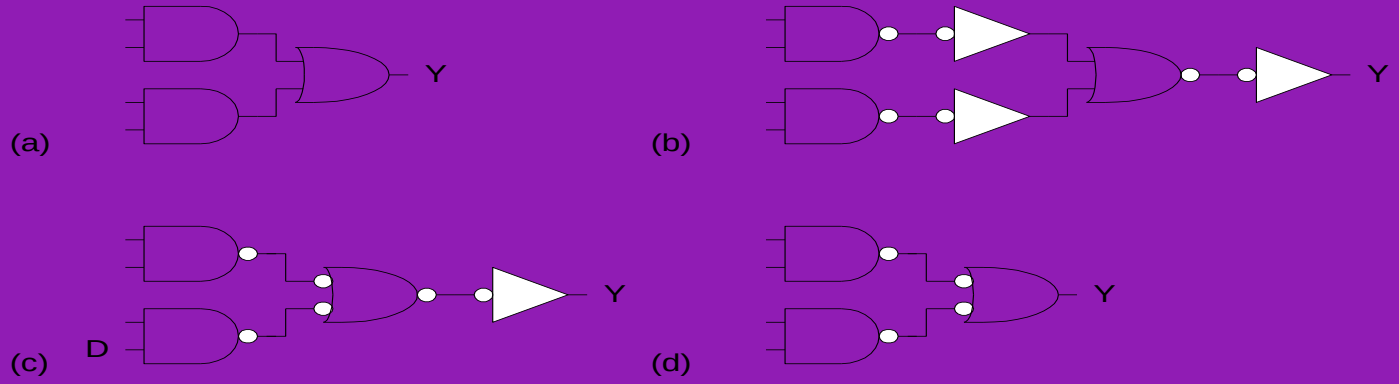


STATIC CMOS

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

Bubble Pushing

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

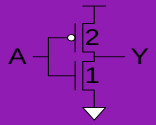


Compound Gates

- Logical Effort of compound gates

unit inverter

$Y = \overline{A}$

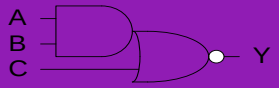


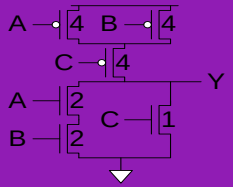
$g_A = 3/3$

$p = 3/3$

AOI21

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





$g_A = 6/3$

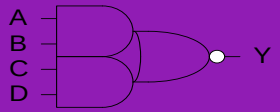
$g_B = 6/3$

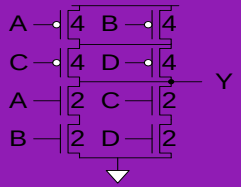
$g_C = 5/3$

$p = 7/3$

AOI22

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





$g_A = 6/3$

$g_B = 6/3$

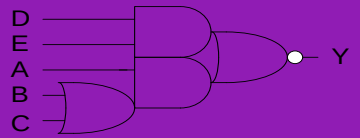
$g_C = 6/3$

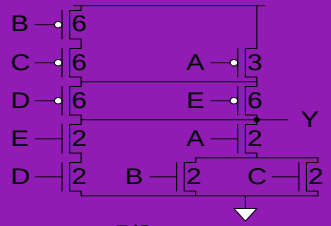
$g_D = 6/3$

$p = 12/3$

Complex AOI

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





$g_A = 5/3$

$g_B = 8/3$

$g_C = 8/3$

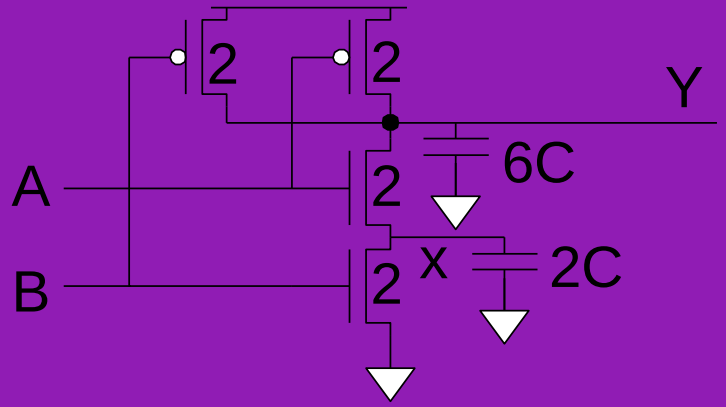
$g_D = 8/3$

$g_E = 8/3$

$p = 16/3$

Input Order

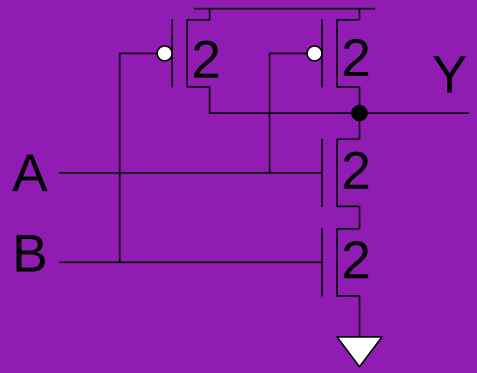
- Our parasitic delay model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest? 2τ
    - If B arrives latest? 2.33τ



Inner & Outer Inputs

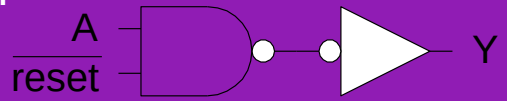
- *Inner* input is closest to output (A)
- *Outer* input is closest to rail (B)

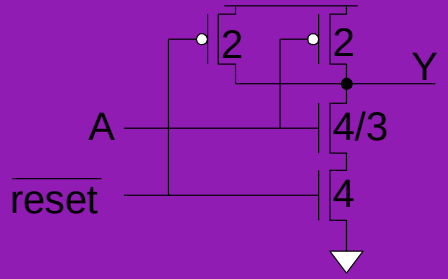
- If input arrival time is known
  - Connect latest input to inner terminal



Asymmetric Gates

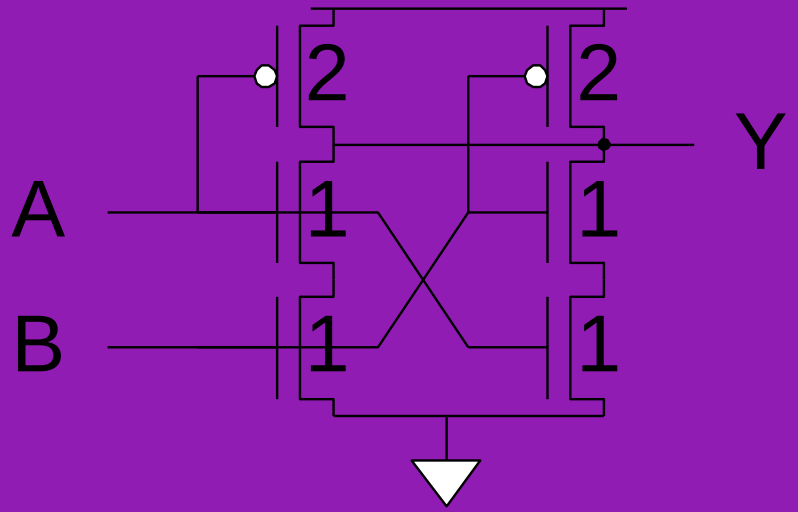
- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same
- $g_A = 10/9$
- $g_B = 2$
- $g_{total} = g_A + g_B = 28/9$
- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up





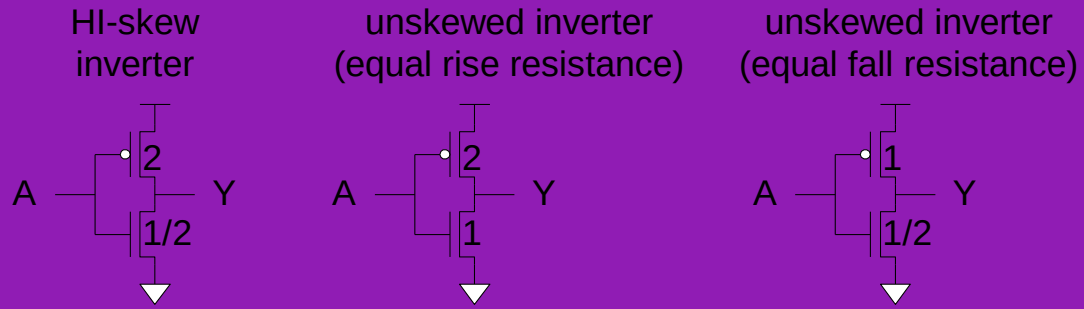
Symmetric Gates

- 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



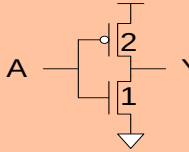
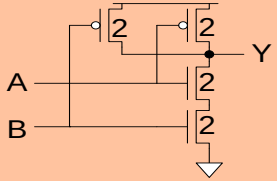
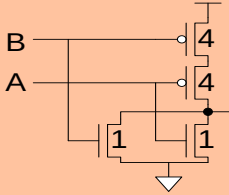
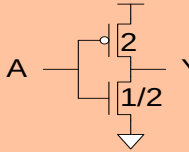
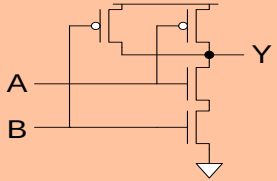
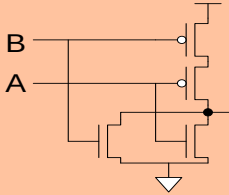
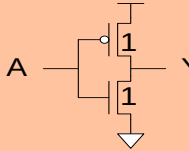
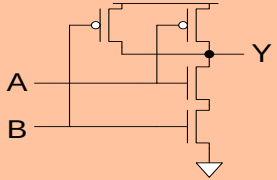
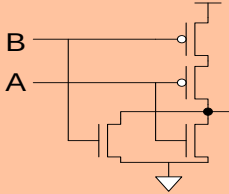
- Calculate logical effort by comparing to unskewed inverter with same effective resistance on that edge.
  - $g_u = 2.5 / 3 = 5/6$
  - $g_d = 2.5 / 1.5 = 5/3$

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.

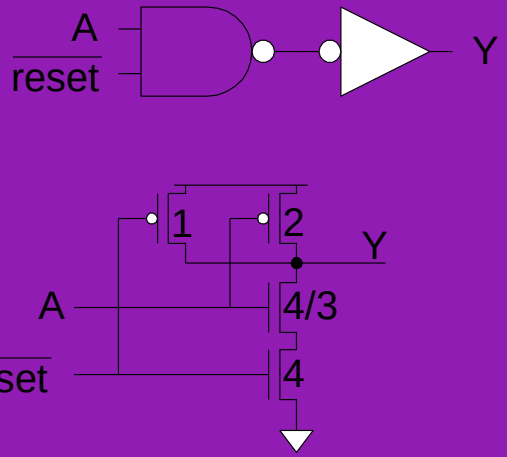
- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small nMOS)
  - LO-skew gates favor falling output (small pMOS)
- Logical effort is smaller for favored direction
- But larger for the other direction

Catalog of Skewed Gates

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

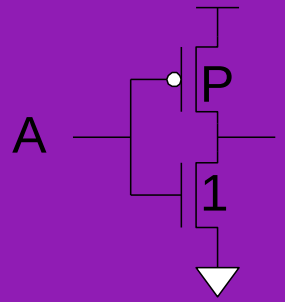
Asymmetric Skew

- 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
  - $t_{pdf} = (P+1)$
  - $t_{pdr} = (P+1)(\mu/P)$
  - $t_{pd} = (P+1)(1+\mu/P)/2 = (P + 1 + \mu + \mu/P)/2$
  - $dt_{pd} / dP = (1- \mu/P^2)/2 = 0$
  - Least delay for P

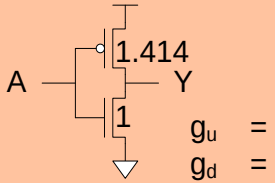


P/N Ratios

- 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

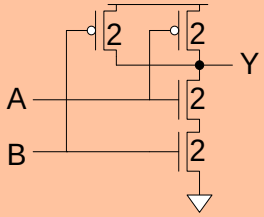
fastest P/N ratio

Inverter



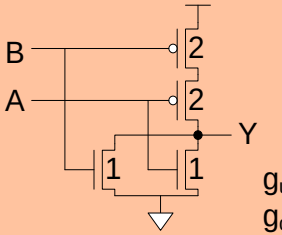
g_u =
g_d =
g_{avg} =

NAND2



g_u =
g_d =
g_{avg} =

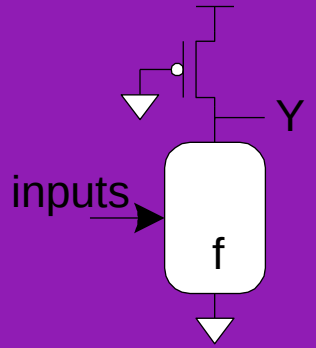
NOR2



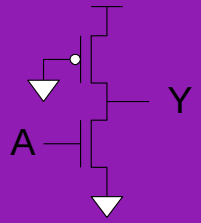
g_u =
g_d =
g_{avg} =

Pseudo-nMOS Gates

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

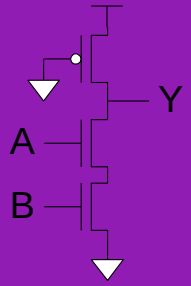


Inverter



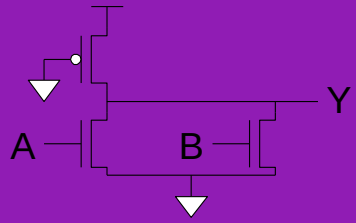
g_u =
g_d =
g_{avg} =
p_u =
p_d =
p_{avg} =

NAND2



g_u =
g_d =
g_{avg} =
p_u =
p_d =
p_{avg} =

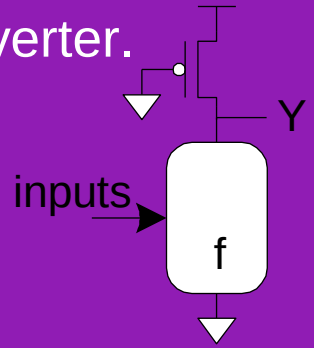
NOR2



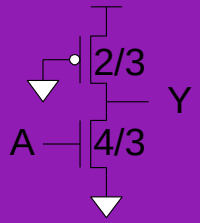
g_u =
g_d =
g_{avg} =
p_u =
p_d =
p_{avg} =

Pseudo-nMOS Gates

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

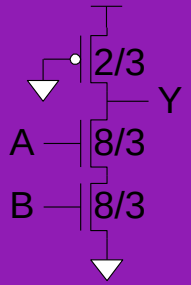


Inverter



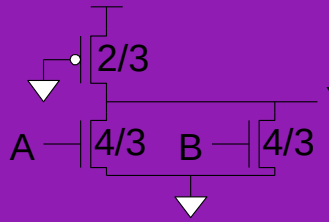
$g_u = 4/3$
$g_d = 4/9$
$g_{avg} = 8/9$
$p_u = 6/3$
$p_d = 6/9$
$p_{avg} = 12/9$

NAND2



$g_u = 8/3$
$g_d = 8/9$
$g_{avg} = 16/9$
$p_u = 10/3$
$p_d = 10/9$
$p_{avg} = 20/9$

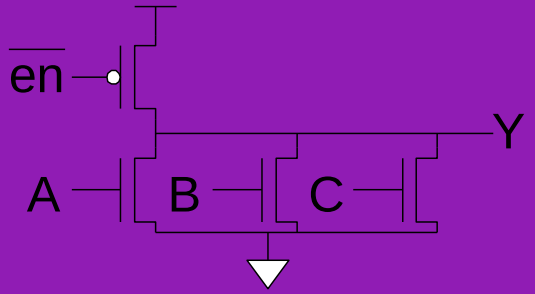
NOR2



$g_u = 4/3$
$g_d = 4/9$
$g_{avg} = 8/9$
$p_u = 10/3$
$p_d = 10/9$
$p_{avg} = 20/9$

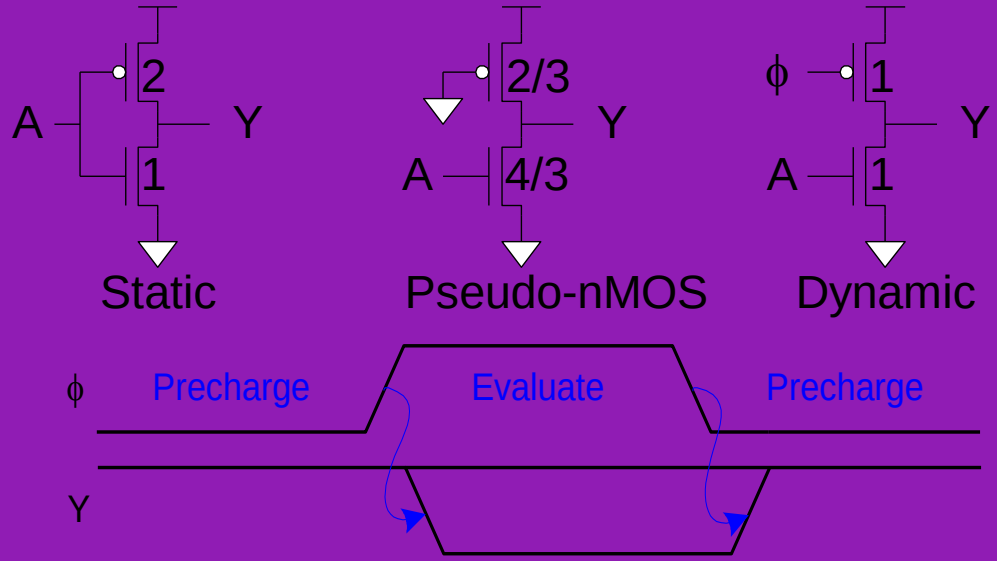
Pseudo-nMOS Power

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



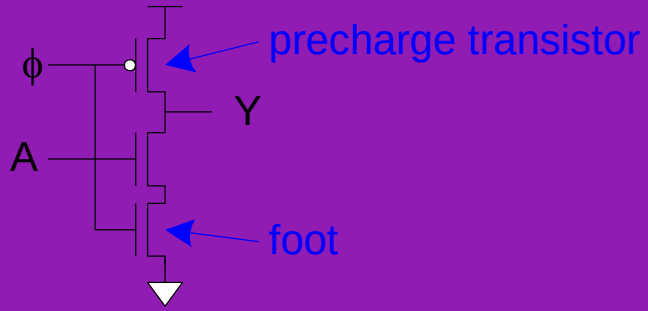
Dynamic Logic

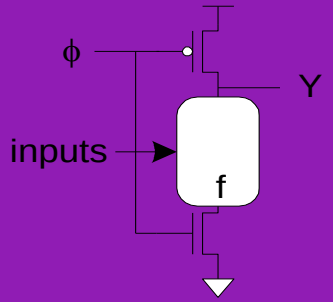
- *Dynamic* gates uses a clocked pMOS pullup
- Two modes: *precharge* and *evaluate*



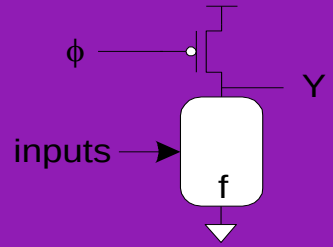
The Foot

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



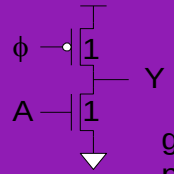
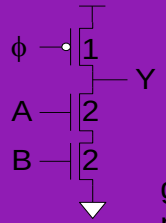
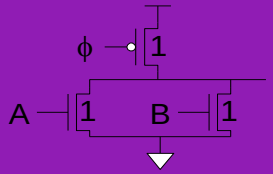
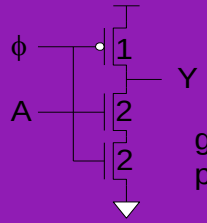
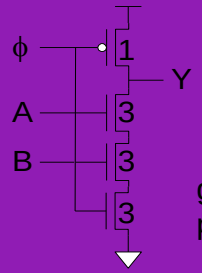
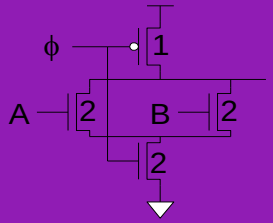


footed



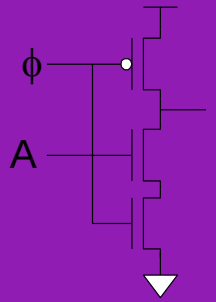
unfooted

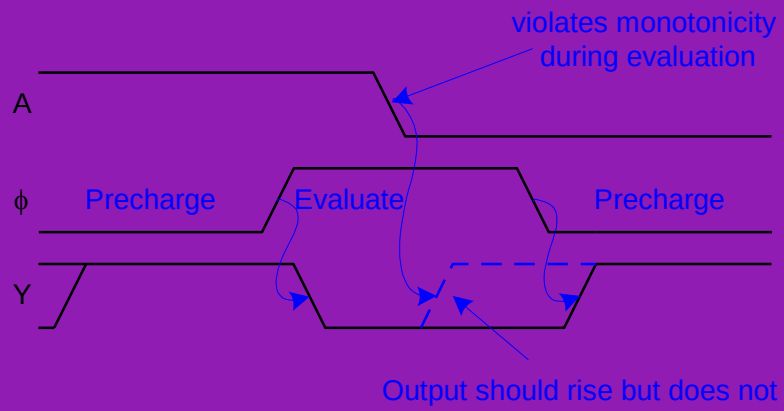
Logical Effort

	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

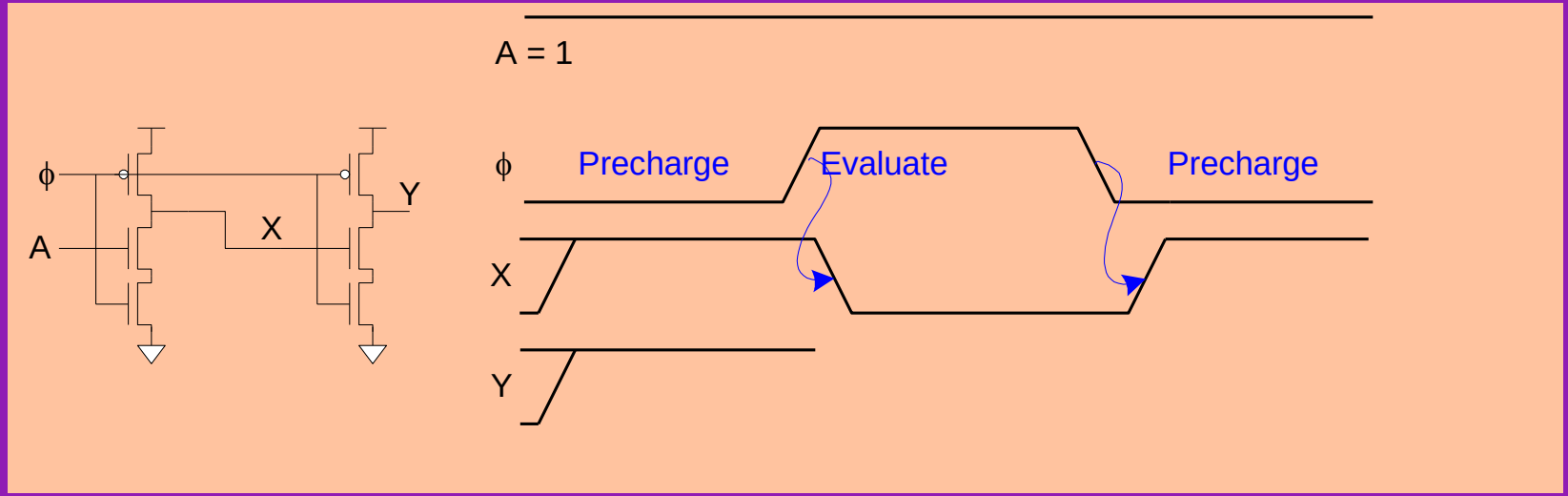
- Dynamic gates require *monotonically rising* inputs during evaluation
  - 0 -> 0
  - 0 -> 1
  - 1 -> 1
  - But not 1 -> 0





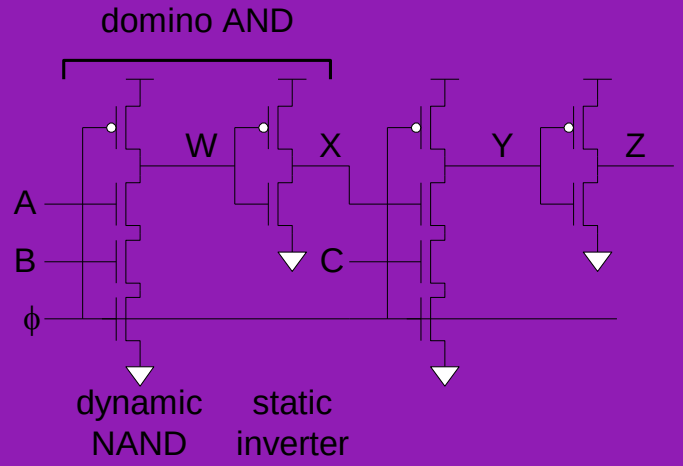
Monotonicity Woes

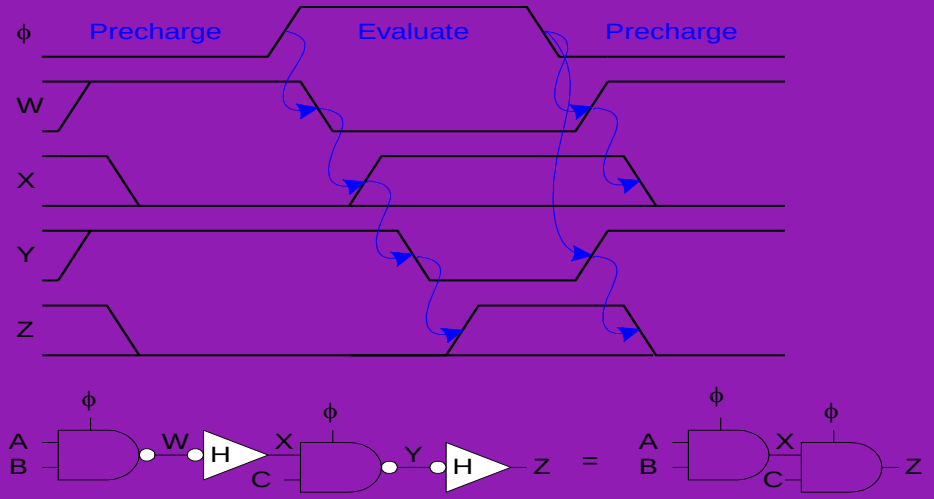
- But dynamic gates produce monotonically falling outputs during evaluation
- Illegal for one dynamic gate to drive another!



Domino Gates

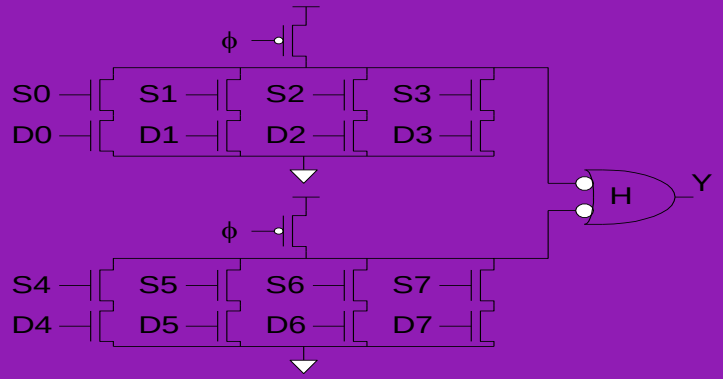
- Follow dynamic stage with inverting static gate
  - Dynamic / static pair is called domino gate
  - Produces monotonic outputs





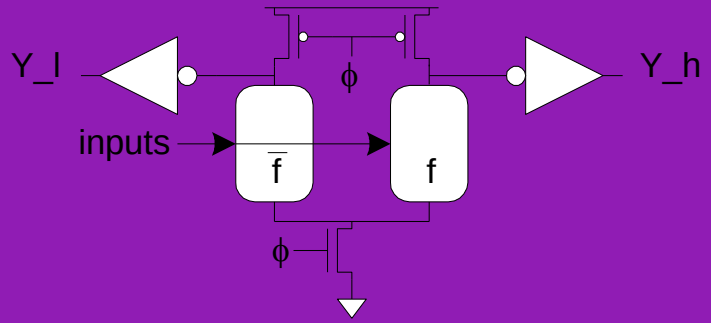
Domino Optimizations

- Each domino gate triggers next one, like a string of dominos toppling over
- Gates evaluate sequentially but precharge in parallel
- Thus evaluation is more critical than precharge
- HI-skewed static stages can perform logic



Dual-Rail Domino

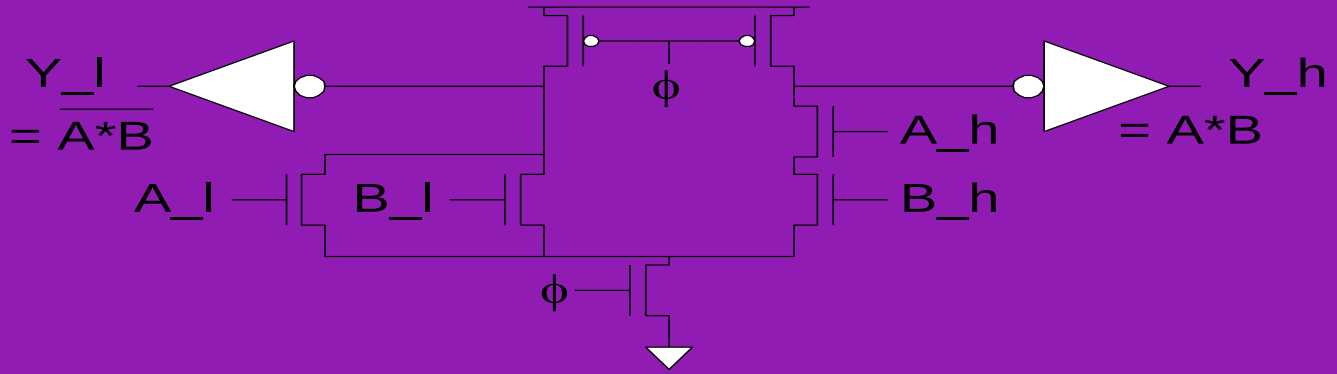
- Domino only performs noninverting functions:
  - AND, OR but not NAND, NOR, or XOR
- Dual-rail domino solves this problem
  - Takes true and complementary inputs
  - Produces true and complementary outputs



sig_h	sig_l	Meaning
0	0	Precharged
0	1	‘0’
1	0	‘1’
1	1	invalid

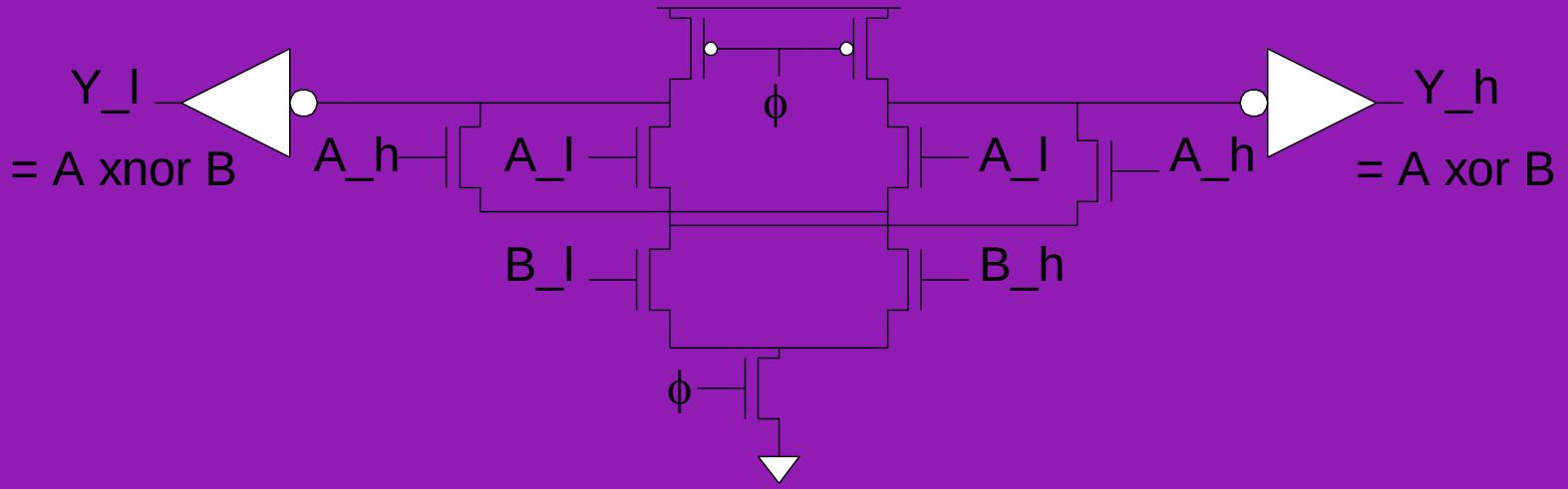
Example: AND/NAND

- Given A_h, A_l, B_h, B_l
- Compute Y_h = AB, Y_l = AB
- Pulldown networks are conduction complements



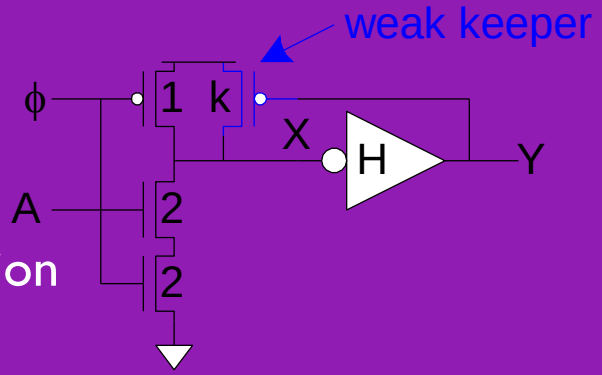
Example: XOR/XNOR

- Sometimes possible to share transistors



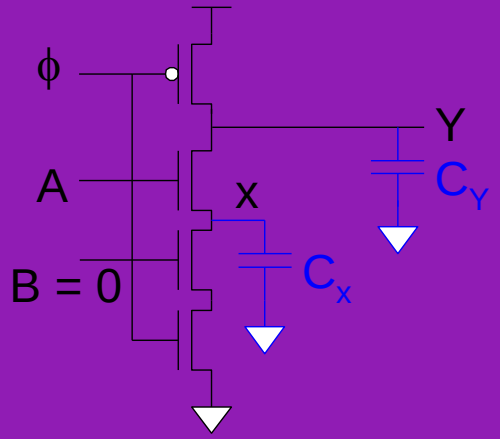
Leakage

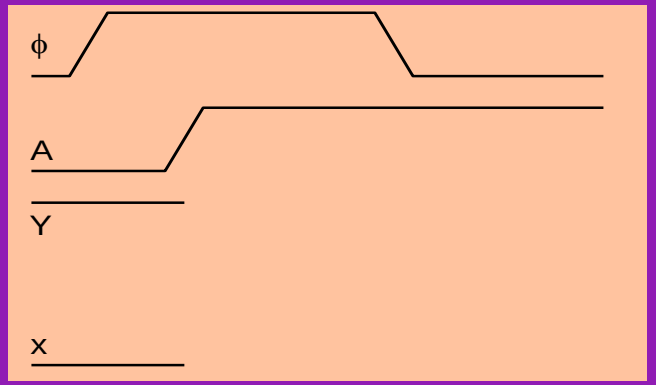
- Dynamic node floats high during evaluation
  - Transistors are leaky ($I_{OFF} \neq 0$)
  - Dynamic value will leak away over time
  - Formerly miliseconds, now nanoseconds
- Use keeper to hold dynamic node
  - Must be weak enough not to fight evaluation



Charge Sharing

- Dynamic gates suffer from charge sharing

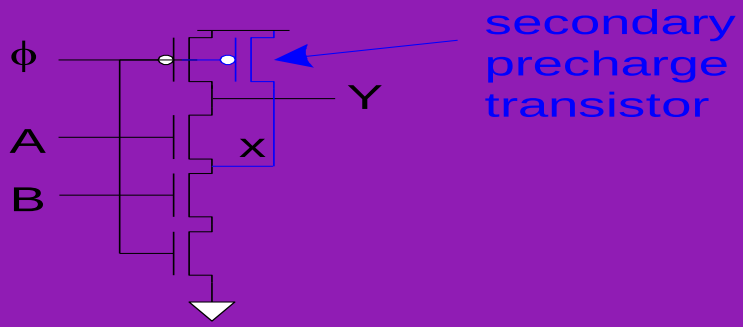




$$V_x = V_Y = \frac{C_Y}{C_x + C_Y} V_{DD}$$

Secondary Precharge

- Solution: add secondary precharge transistors
  - Typically need to precharge every other node
- Big load capacitance C_Y helps as well



Noise Sensitivity

- Dynamic gates are very sensitive to noise
  - Inputs: $V_{IH} \approx V_{tn}$
  - Outputs: floating output susceptible noise
- Noise sources
  - Capacitive crosstalk
  - Charge sharing
  - Power supply noise
  - Feedthrough noise
  - And more!

Power

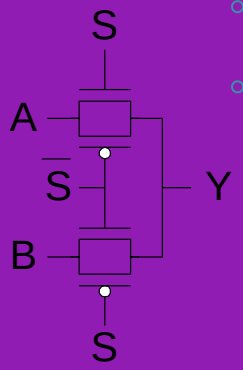
- Domino gates have high activity factors
  - Output evaluates and precharges
    - If output probability = 0.5, α = 0.5
      - Output rises and falls on half the cycles
  - Clocked transistors have α = 1
- Leads to very high power consumption

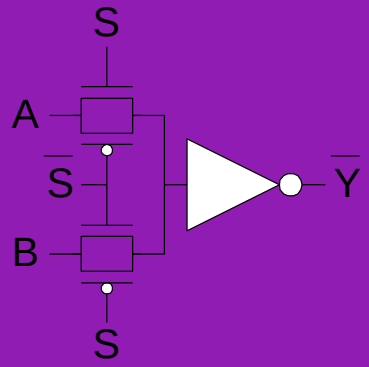
Domino Summary

- Domino logic is attractive for high-speed circuits
  - 1.3 – 2x faster than static CMOS
  - But many challenges:
    - Monotonicity, leakage, charge sharing, noise
- Widely used in high-performance microprocessors in 1990s when speed was king
- Largely displaced by static CMOS now that power is the limiter
- Still used in memories for area efficiency

Pass Transistor Circuits

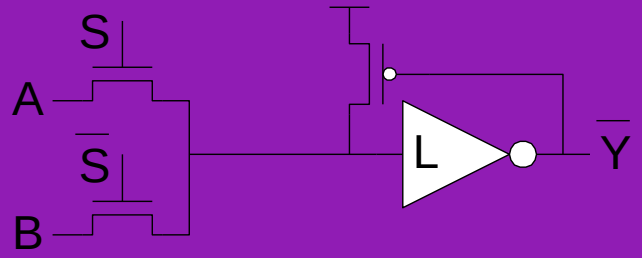
- Use pass transistors like switches to do logic
- Inputs drive diffusion terminals as well as gates
- CMOS + Transmission Gates:
  - 2-input multiplexer
  - Gates should be restoring





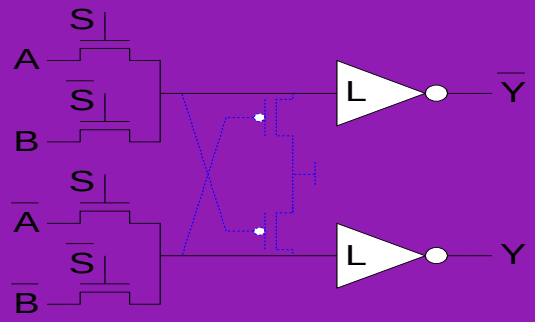
LEAP

- **LEA**n integration with **P**ass transistors
- Get rid of pMOS transistors
  - Use weak pMOS feedback to pull fully high
  - Ratio constraint



CPL

- **C**omplementary **P**ass-transistor **L**ogic
  - Dual-rail form of pass transistor logic
  - Avoids need for ratioed feedback
  - Optional cross-coupling for rail-to-rail swing



Pass Transistor Summary

- Researchers investigated pass transistor logic for general purpose applications in the 1990's
  - Benefits over static CMOS were small or negative
  - No longer generally used
- However, pass transistors still have a niche in special circuits such as memories where they offer small size and the threshold drops can be managed

THANK YOU