FOURTH EDITION

CMOS VLSI DESIGN

A CIRCUITS AND SYSTEMS PERSPECTIVE

**NEIL H. E. WESTE**   **DAVID MONEY HARRIS**

Lecture 10: Circuit Families

Outline

- Pseudo-nMOS Logic
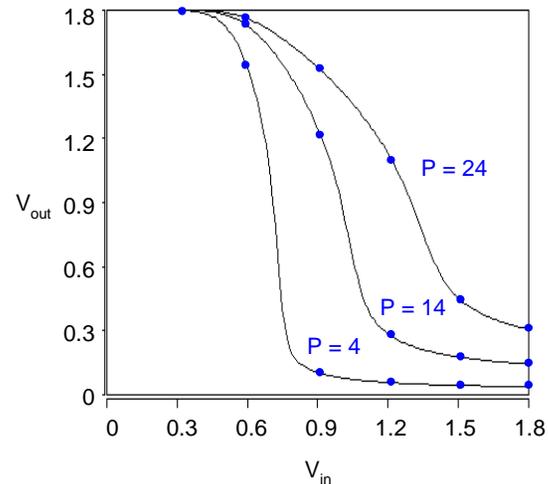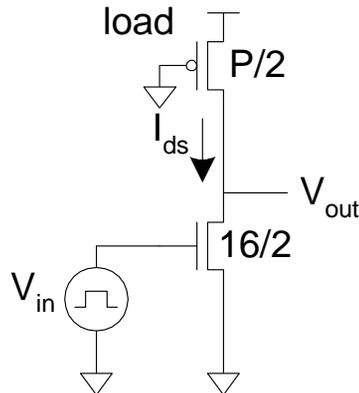- Dynamic Logic
- Pass Transistor Logic

Introduction

- ❑ What makes a circuit fast?
  - – $I = C\ dV/dt$ -> $t_{pd} \propto (C/I)\ \Delta V$
  - – 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?
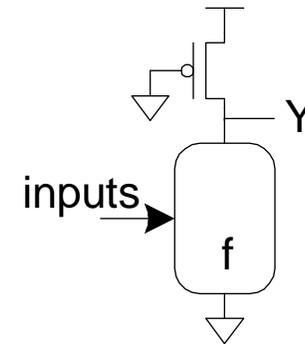- ❑ Various circuit families try to do this…
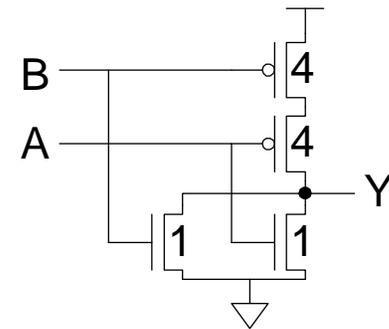
Pseudo-nMOS

- ❑ 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
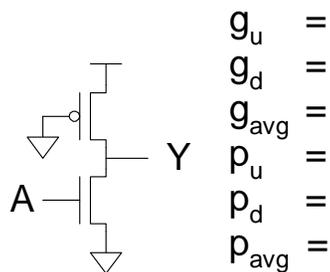   - – Make pMOS about ¼ effective strength of pulldown network

load P/2

I_{ds}

V_{out}

16/2

V_{in}

1.8
1.5
1.2
V_{out} 0.9
0.6
0.3
0

P = 24
P = 14
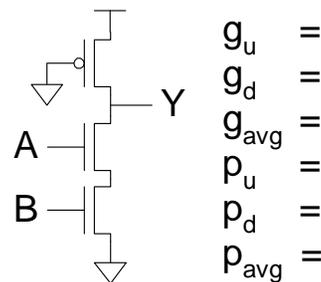P = 4

0   0.3   0.6   0.9   1.2   1.5   1.8

V_{in}
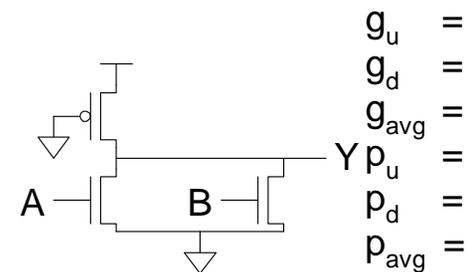
Pseudo-nMOS Gates

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

inputs → f — Y

Inverter

A — Y

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

NAND2

A
B — Y

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

NOR2

A   B — Y
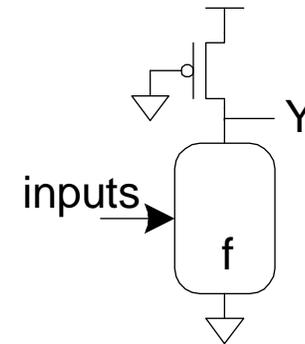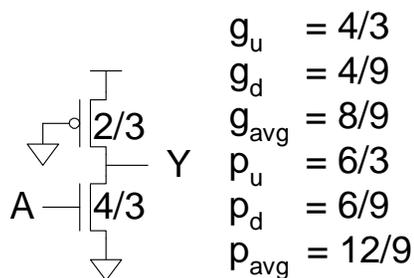
g_u =
g_d =
g_{avg} =
p_u =
p_d =
p_{avg} =

Pseudo-nMOS Gates

❑ Design for unit current on output
   to compare with unit inverter.

❑ pMOS fights nMOS

Y

inputs →

f

Inverter

2/3

A — 4/3

Y

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

2/3

Y

A — 8/3

B — 8/3

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

2/3

A — 4/3   B — 4/3

Y

g_u = 4/3
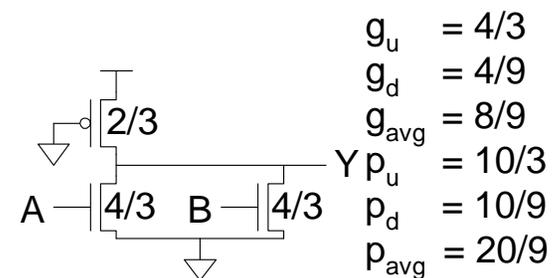g_d = 4/9
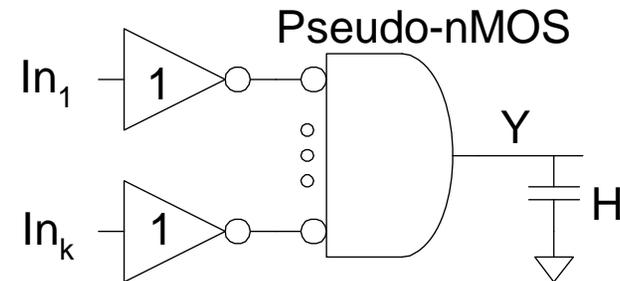g_{avg} = 8/9
p_u = 10/3
p_d = 10/9
p_{avg} = 20/9
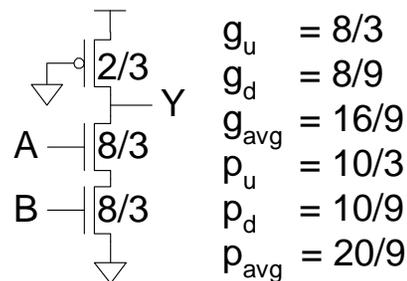
Pseudo-nMOS Design

❑ Ex: Design a k-input AND gate using pseudo-nMOS. Estimate the delay driving a fanout of H

❑ G =

❑ F =

❑ P =

❑ N =

❑ D =

Pseudo-nMOS

In_1 — 1

In_k — 1

Y

H

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
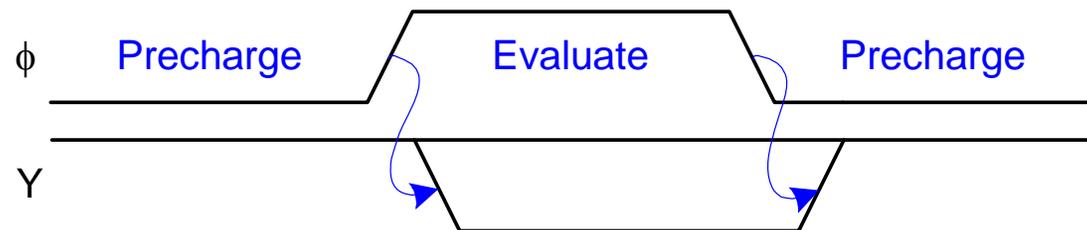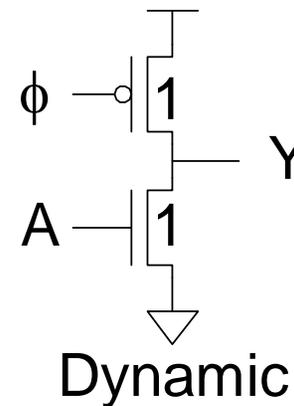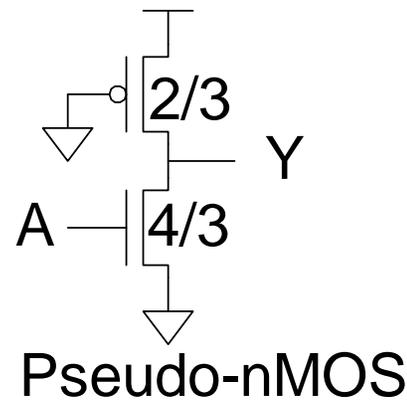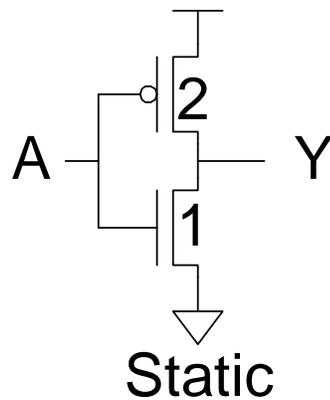- ❑ Turn off pMOS when not in use
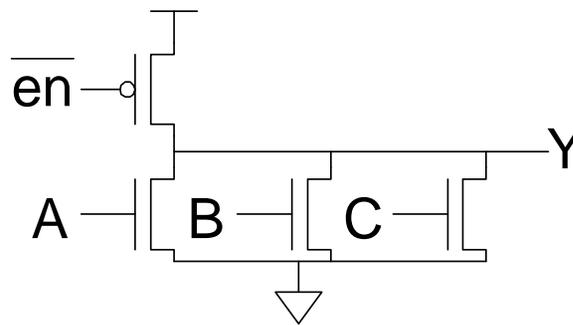
Ratio Example

❑ 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
- – $I_{on-p} = 36 \ \mu A$

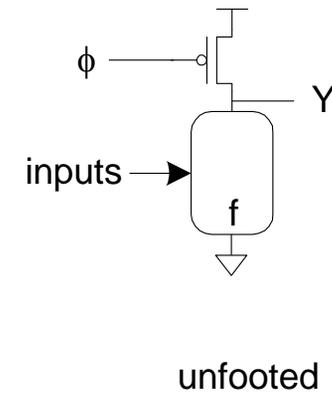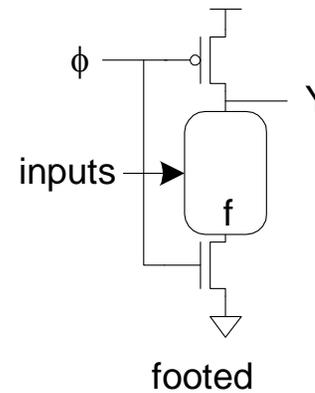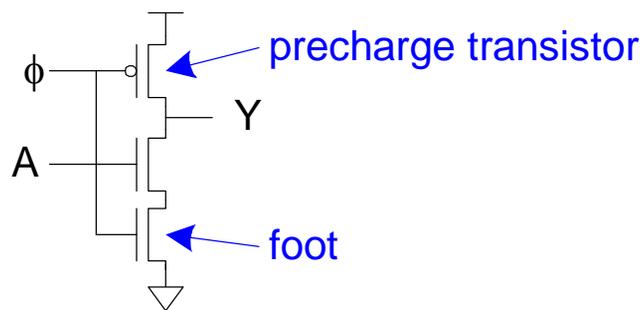❑ Solution:

$$P_{\text{pull-up}} =$$

$$P_{\text{static}} =$$

Dynamic Logic

❑ *Dynamic* gates uses a clocked pMOS pullup

❑ Two modes: *precharge* and *evaluate*

A ——|[2] [1] Y

Static

A ——|[4/3] [2/3] Y

Pseudo-nMOS

φ ——|[1] A ——|[1] Y
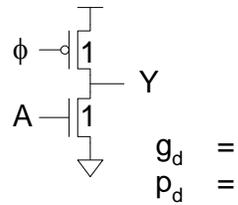
Dynamic

φ   Precharge   Evaluate   Precharge

Y

The Foot

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

precharge transistor
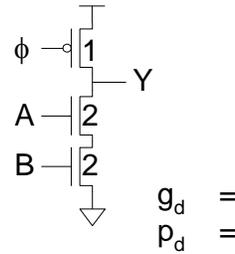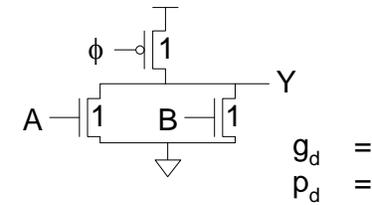
foot

footed

unfooted

Logical Effort

	Inverter	NAND2	NOR2

unfooted

Inverter (unfooted):
ϕ — 1
A — 1
Y
g_d =
p_d =

NAND2 (unfooted):
ϕ — 1
A — 2
B — 2
Y
g_d =
p_d =

NOR2 (unfooted):
ϕ — 1
A — 1 B — 1
Y
g_d =
p_d =

footed

Inverter (footed):
ϕ — 1
A — 2
2
Y
g_d =
p_d =

NAND2 (footed):
ϕ — 1
A — 3
B — 3
3
Y
g_d =
p_d =

NOR2 (footed):
ϕ — 1
A — 2 B — 2
2
Y
g_d =
p_d =
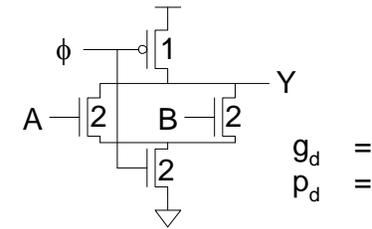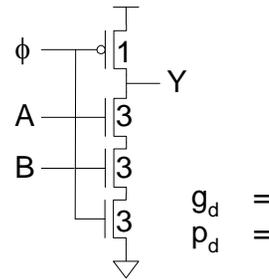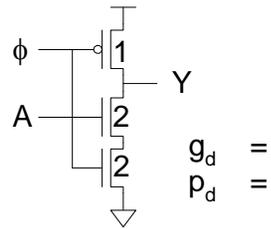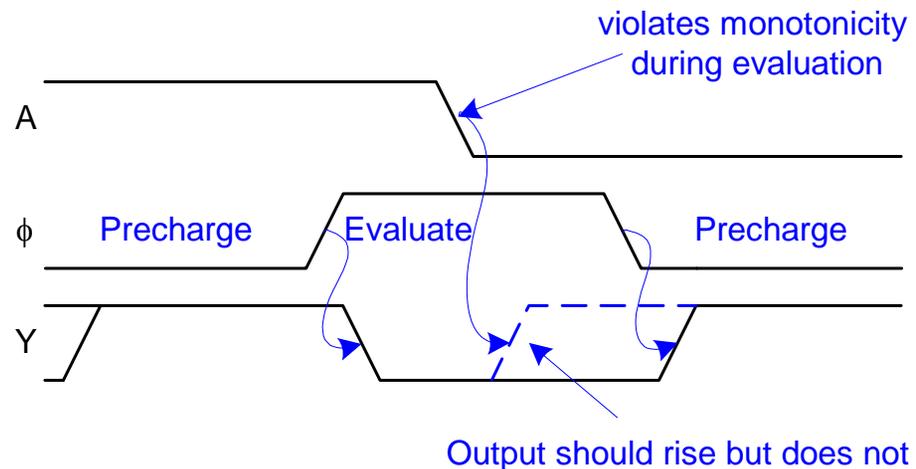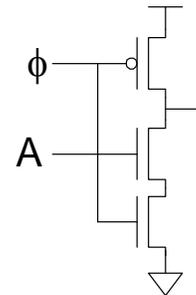
Monotonicity

❑ Dynamic gates require *monotonically rising* inputs during evaluation

- – 0 -> 0
- – 0 -> 1
- – 1 -> 1
- – But not 1 -> 0

violates monotonicity during evaluation

A

φ Precharge   Evaluate   Precharge

Y

Output should rise but does not
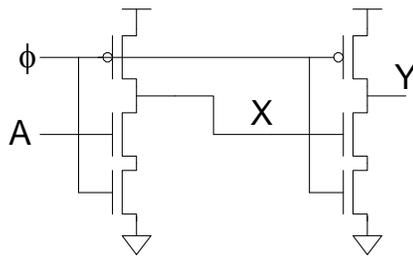
Monotonicity Woes

- ❑ But dynamic gates produce monotonically falling outputs during evaluation

- ❑ Illegal for one dynamic gate to drive another!

A = 1

φ   Precharge   Evaluate   Precharge

X

Y
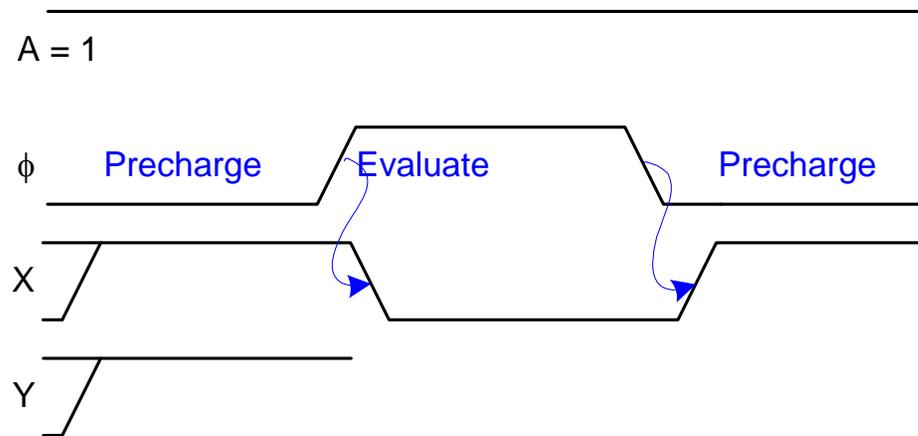
Domino Gates

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

domino AND

dynamic
NAND

static
inverter

Precharge   Evaluate   Precharge
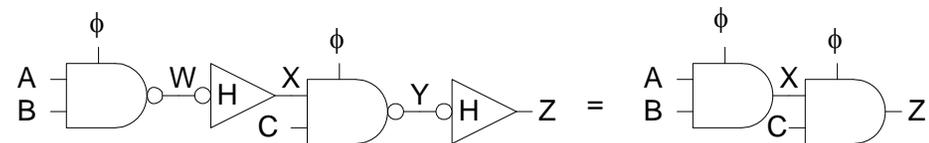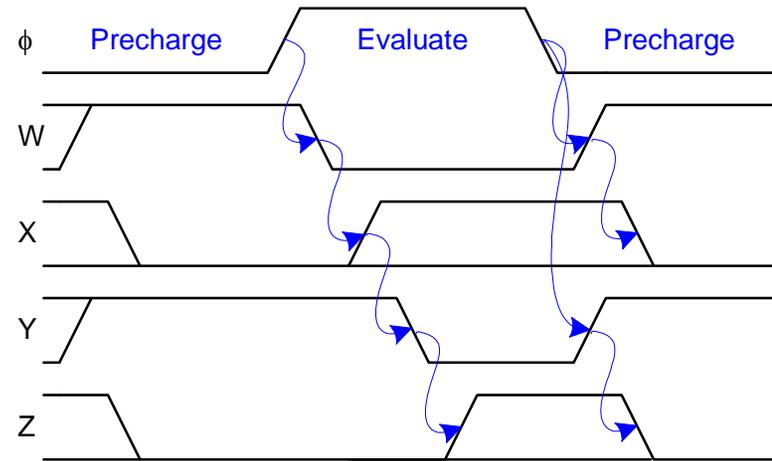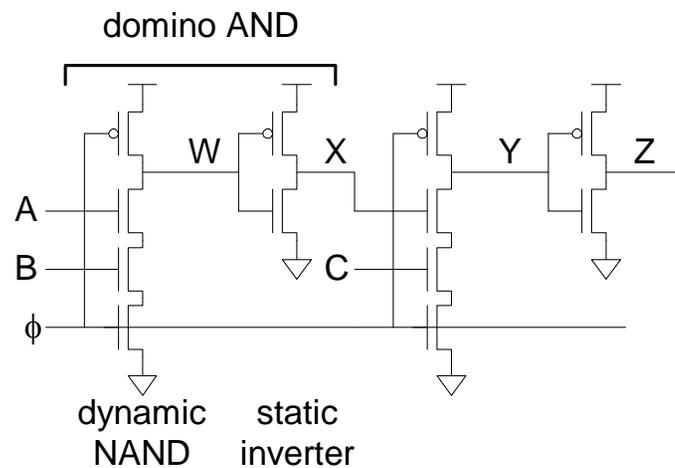
Domino Optimizations

- ❑ 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
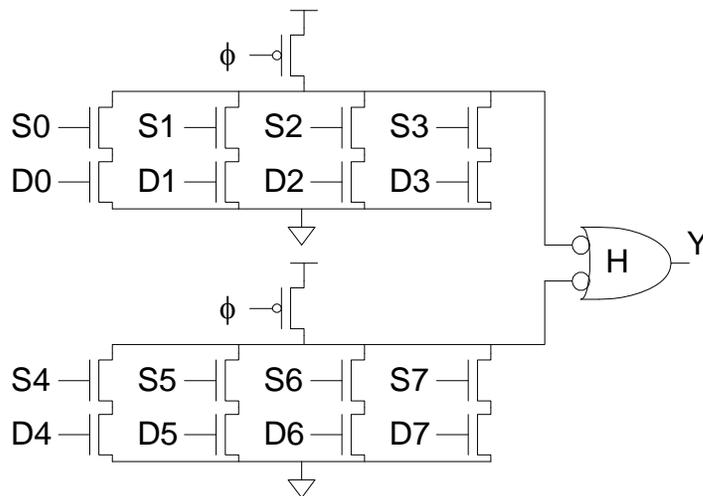
Dual-Rail Domino

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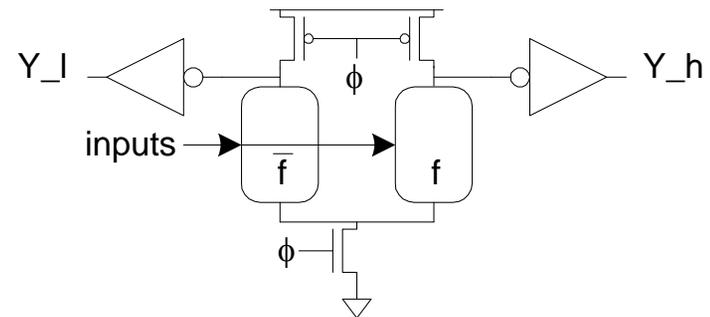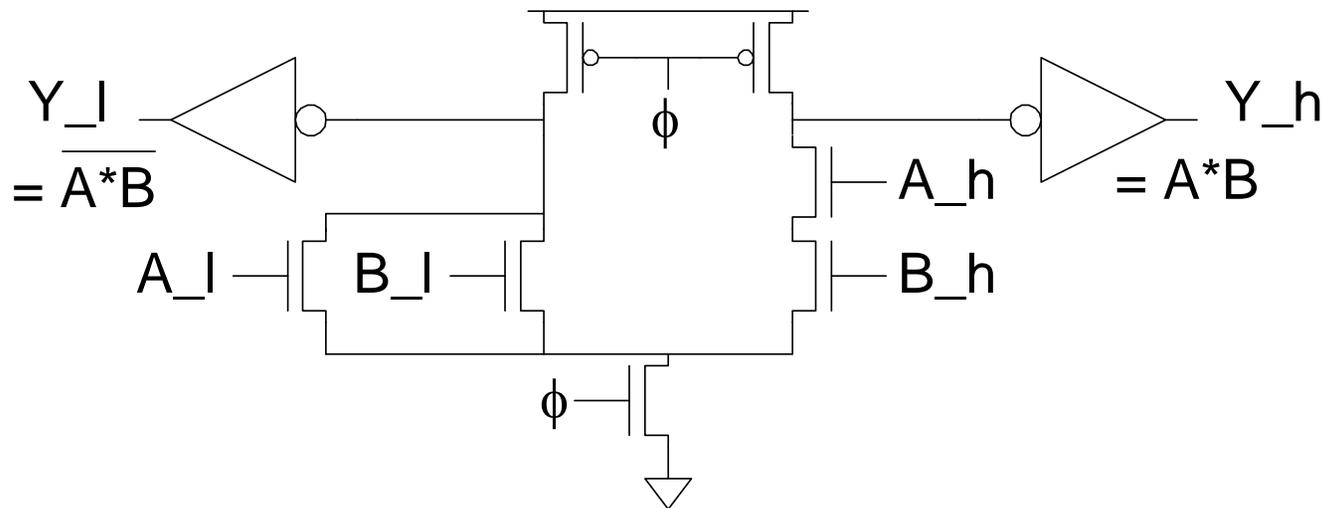 – AND, OR but not NAND, NOR, or XOR
❑ Dual-rail domino solves this problem
  – Takes true and complementary inputs
  – Produces true and complementary outputs

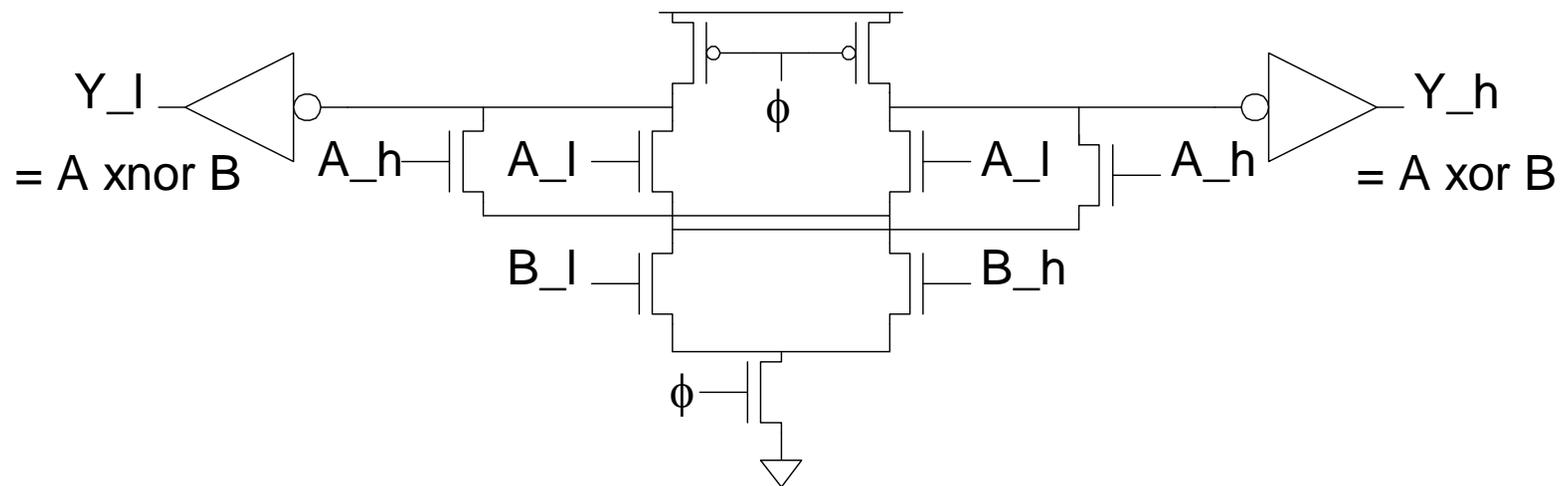| sig_h	sig_l	Meaning
0	0	Precharged
0	1	'0'
1	0	'1'
1	1	invalid

Example: AND/NAND

- ❑ Given A_h, A_l, B_h, B_l
- ❑ Compute Y_h = AB, Y_l = $\overline{AB}$
- ❑ Pulldown networks are conduction complements

Example: XOR/XNOR

❑ Sometimes possible to share transistors

Y_l = A xnor B   A_h   A_l   φ   A_l   A_h   Y_h = A xor B
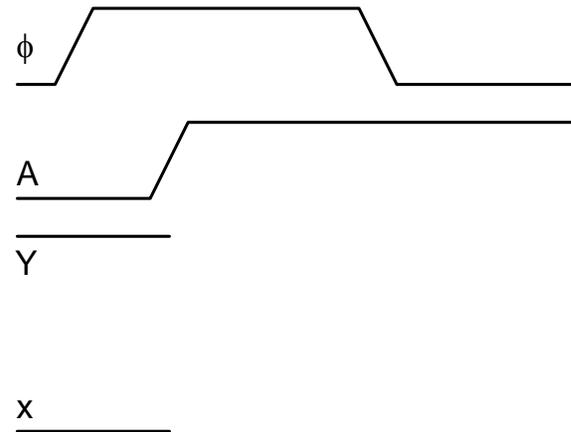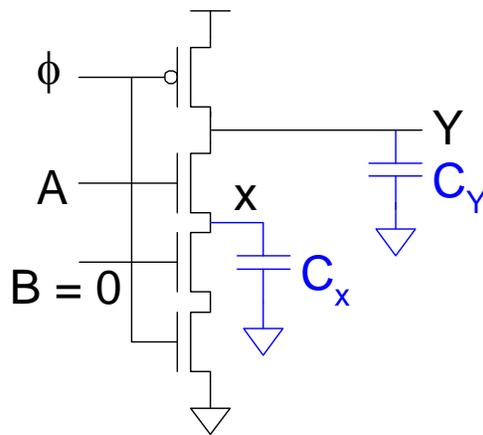
B_l   B_h

φ

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
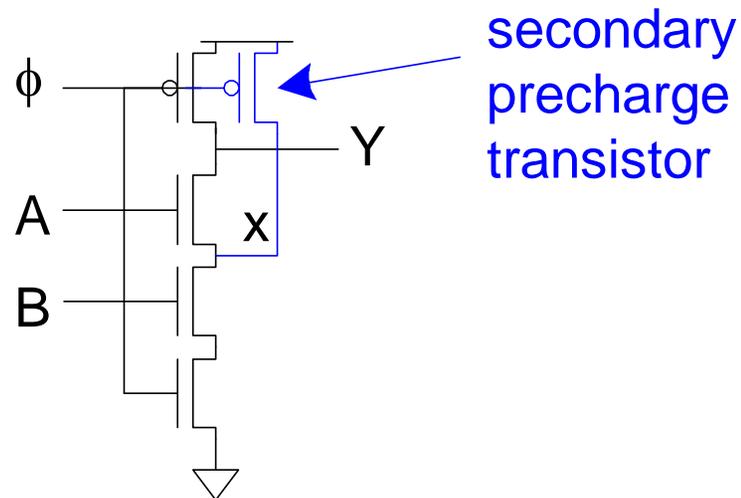
weak keeper
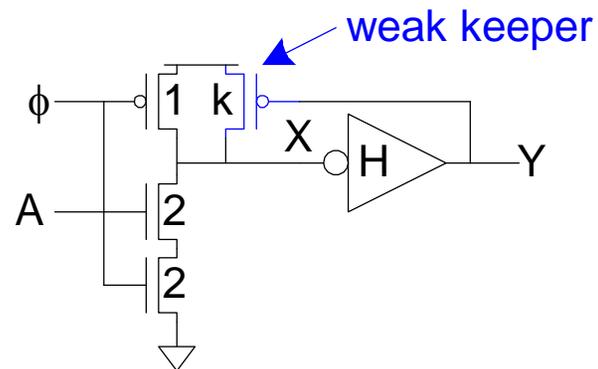
Charge Sharing

❑ Dynamic gates suffer from charge sharing

$$V_x = V_Y =$$

Secondary Precharge

❑ Solution: add secondary precharge transistors

    – Typically need to precharge every other node

❑ Big load capacitance C_Y helps as well

secondary
precharge
transistor

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:
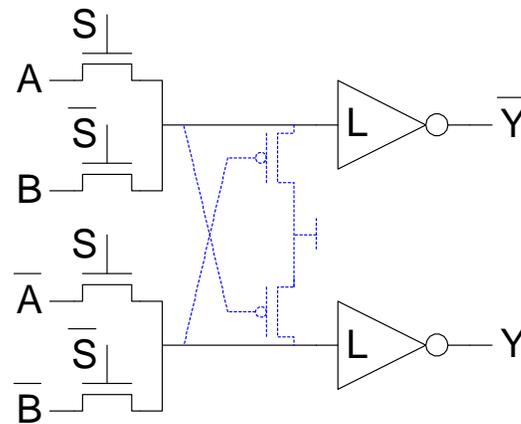- 2-input multiplexer
- Gates should be restoring
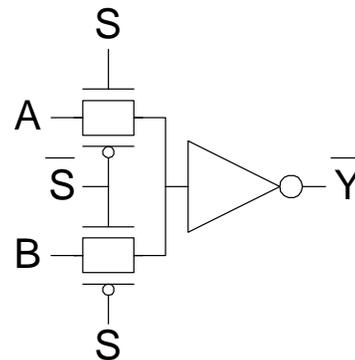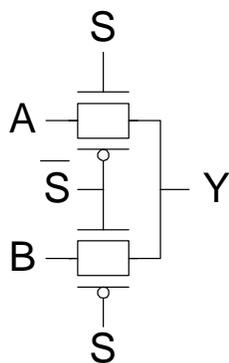
LEAP

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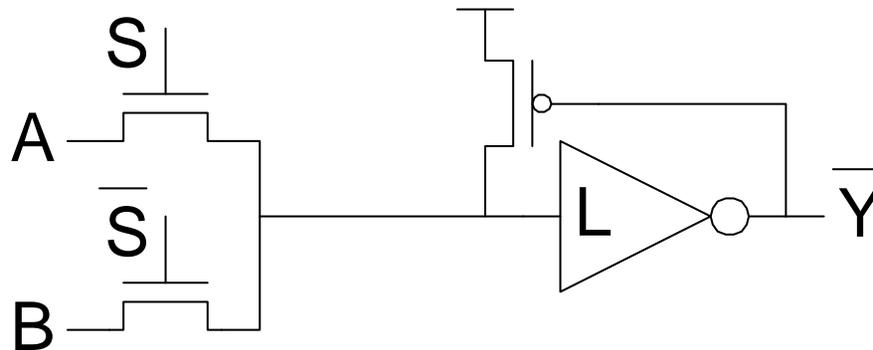
CPL

- **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