ESE3700: Circuit-Level Modeling, Design, and Optimization for Digital Systems

Lec 16: April 2, 2025

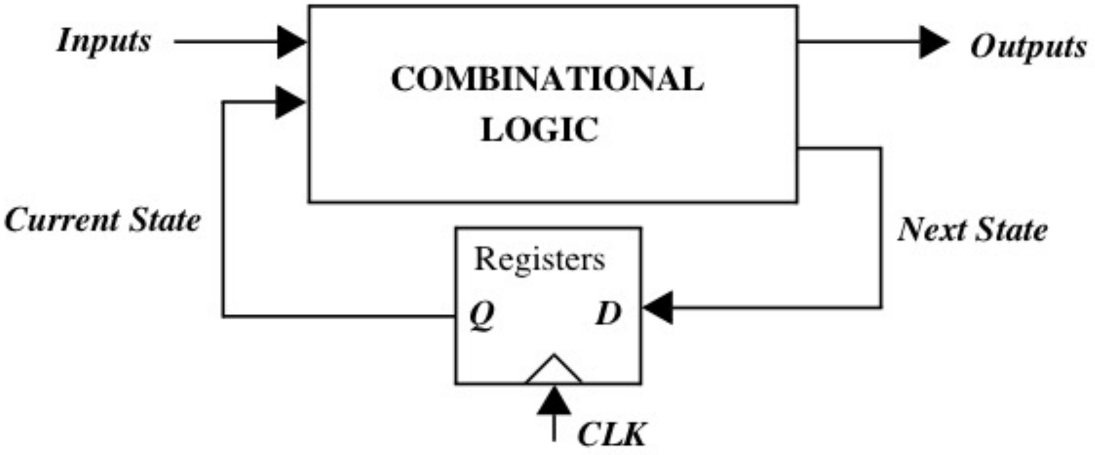
Dynamic Logic

Today

- Dynamic (Clocked) Logic
  - Strategy
  - Form
  - Compare CMOS

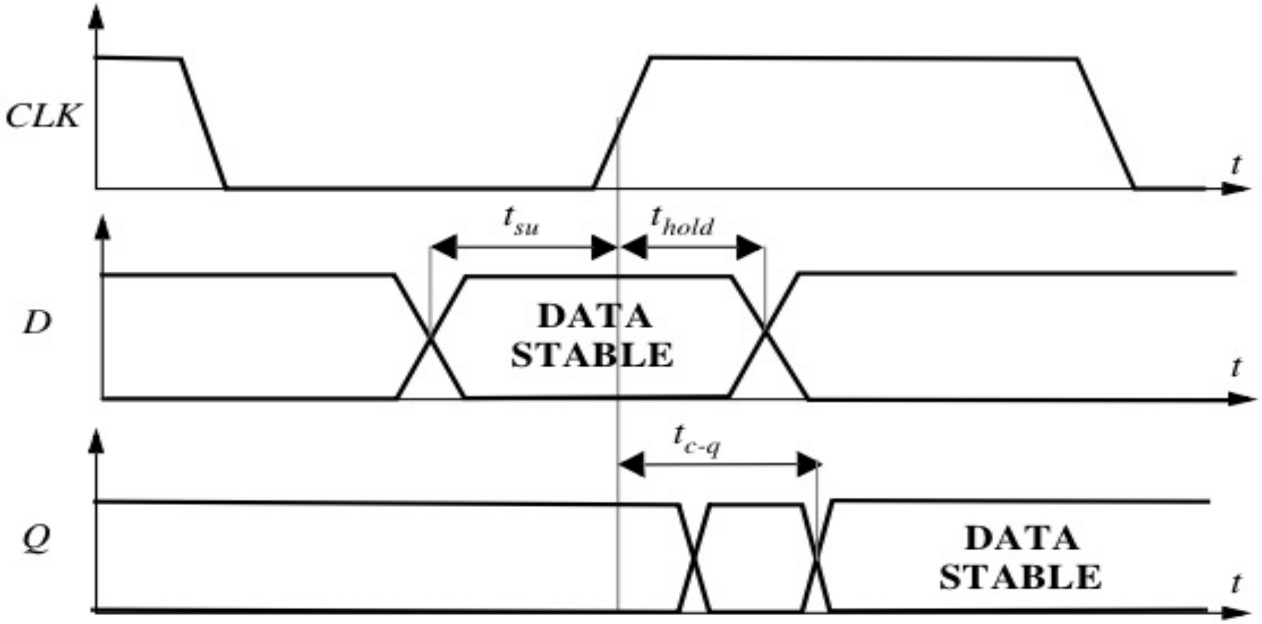
Clocking

Latch Timing Issues



$$T \geq t_{c\text{-}q} + t_{plogic} + t_{su}$$

$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$



Clocking Highlights

- Breaking logic up with registers allows circuit to run at high frequency
  - Inputs decoupled from outputs
- Clock discipline simplifies logic composition
  - Abstracts many internal timing details
  - Just concerned with making clock period long enough
- Design Discipline – keeping data stable around clock edge
  - Setup, hold time – determined by latch circuit
  - Worst case and minimum Clk→Q delay for latch

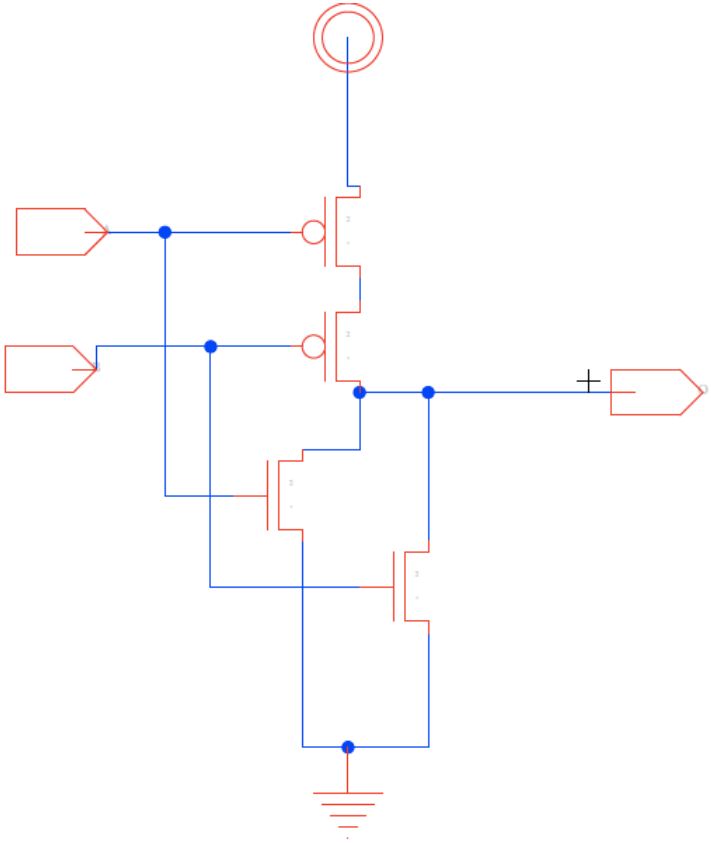
Clocking

- Circuits typically operate in a clocked environment
  - Synchronous circuits
- Gives some additional structure we can exploit → dynamic logic

Dynamic Logic

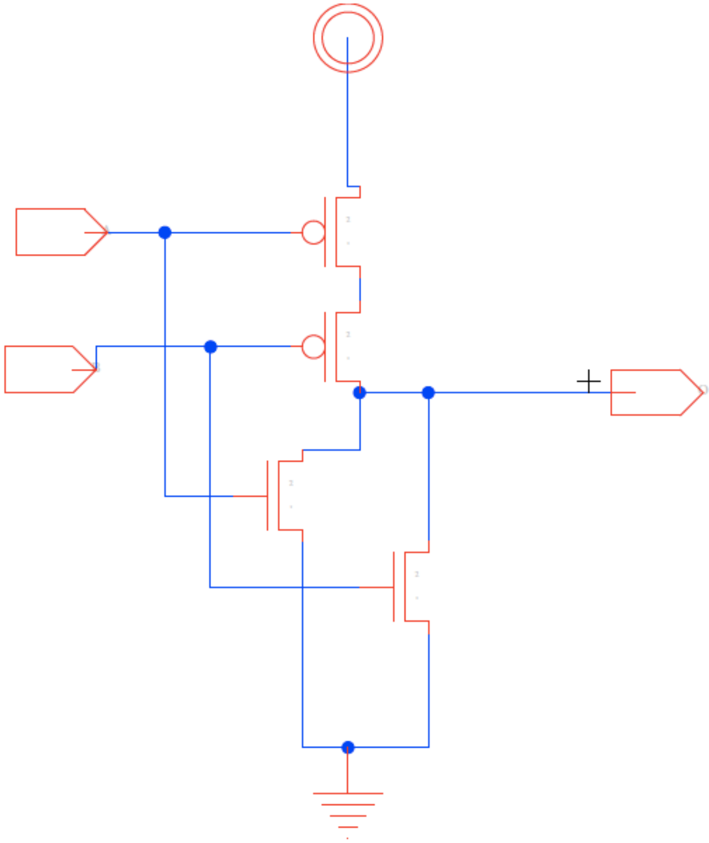
Motivation

- We would like to avoid driving both pullup/pulldown networks
  - reduce capacitive load
    - Power, delay



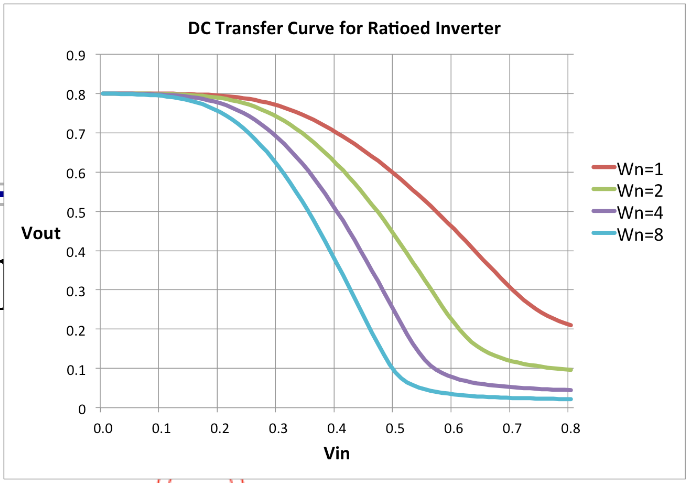
Motivation

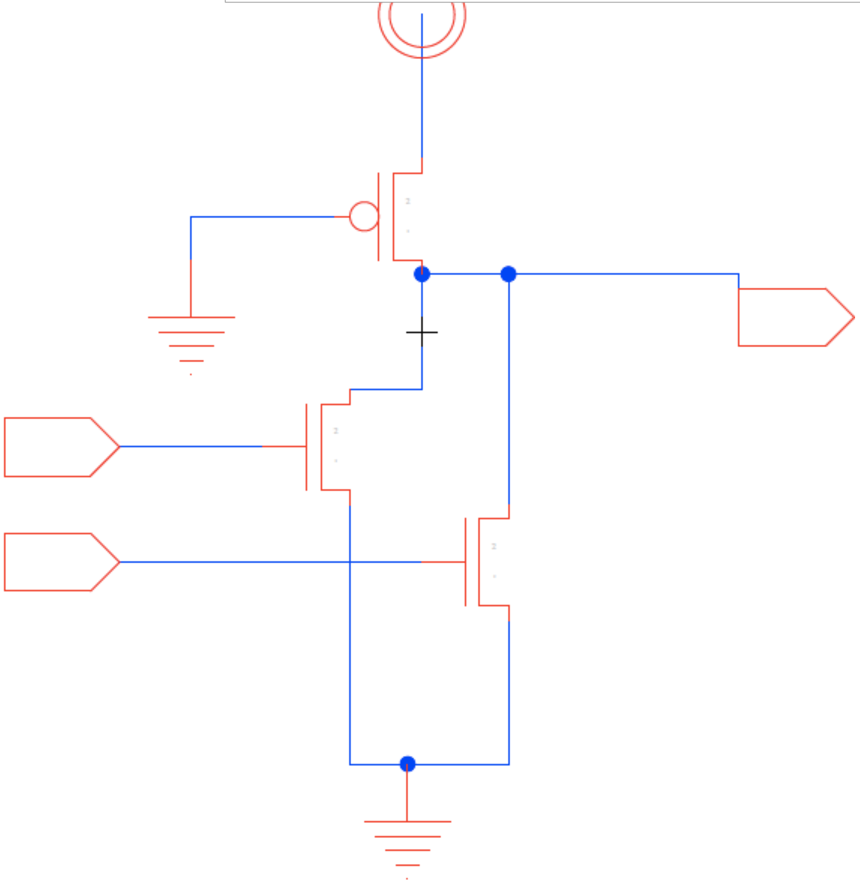
- We would like to avoid driving both pullup/pulldown networks
  - reduce capacitive load
    - Power, delay
- Ratioed Logic



Motivation

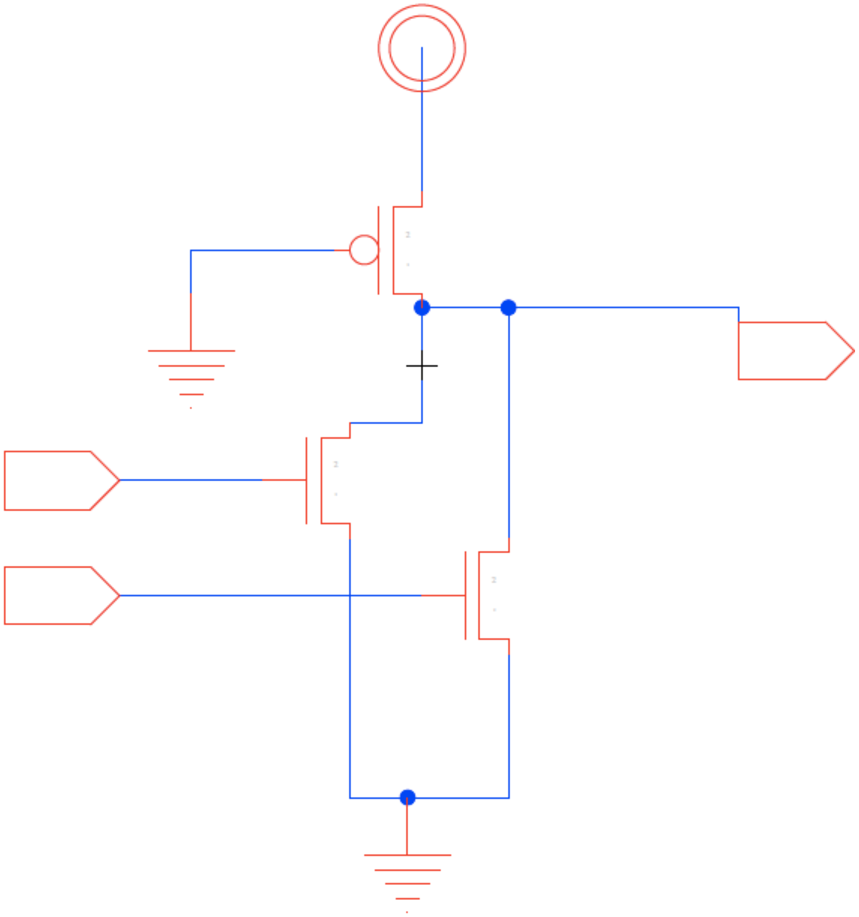
- We would like to avoid driving bot pullup/pulldown networks
  - reduce capacitive load
    - Power, delay
- Ratioed Logic cons:
  - Large devices for ratioing
    - Meeting noise margins
  - Slow pullup
  - Static power

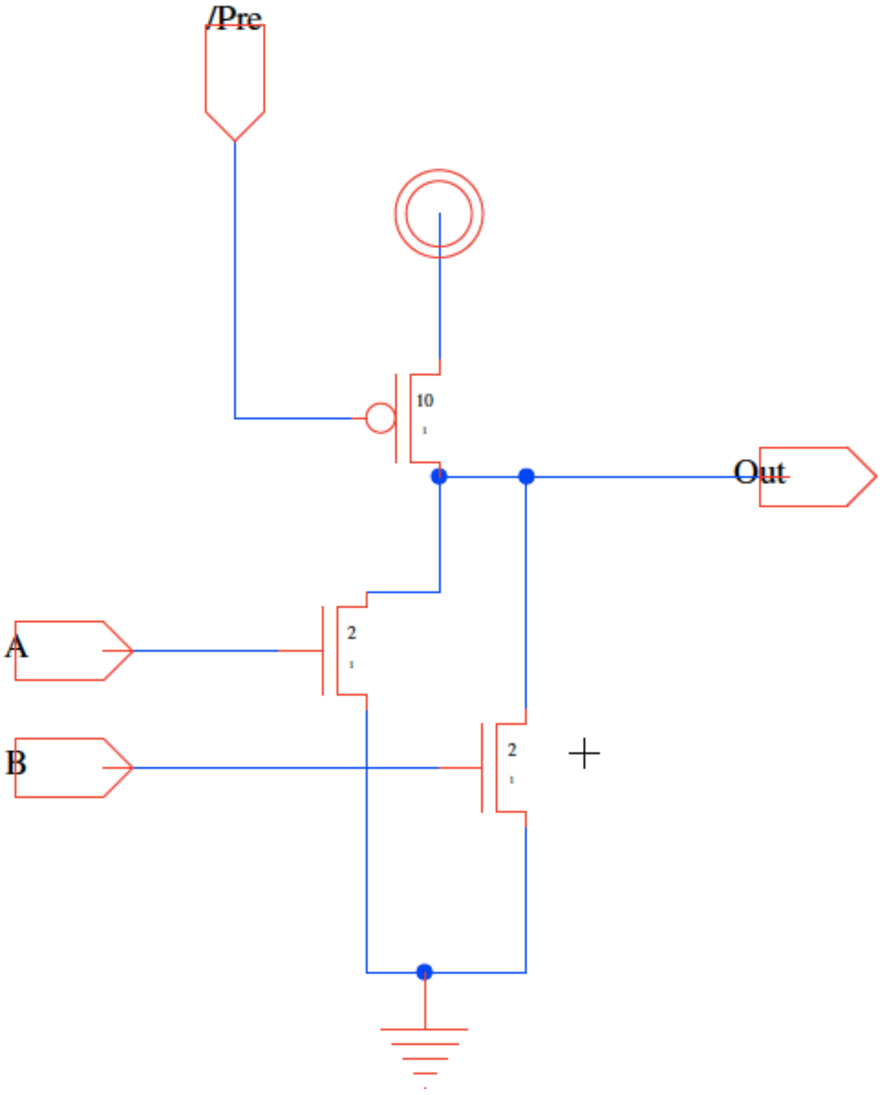




Idea

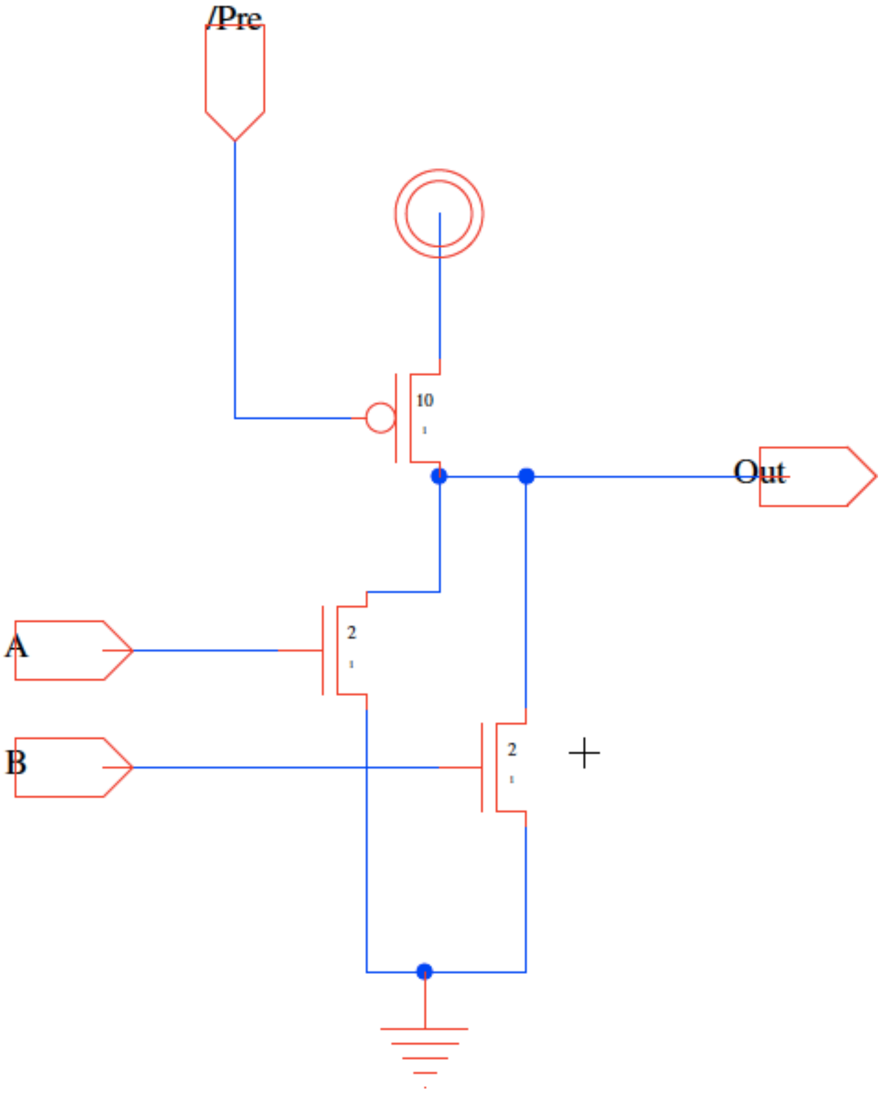
- Use clock to disable pullup network during logic evaluation





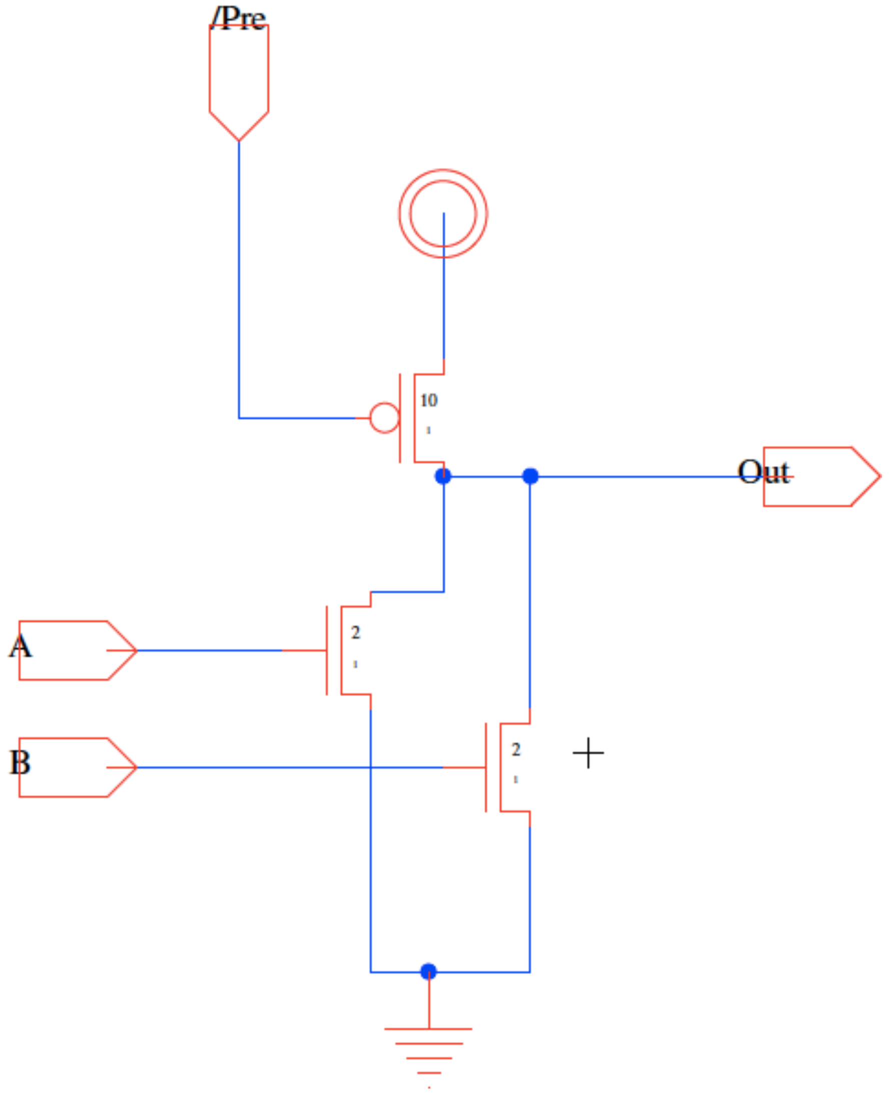
Idea

- Use clock to disable pullup network during logic evaluation
- Define two phases
  - Pre-charge
    - Output pre-charged
  - Evaluation
    - Pulldown network evaluates gate logic



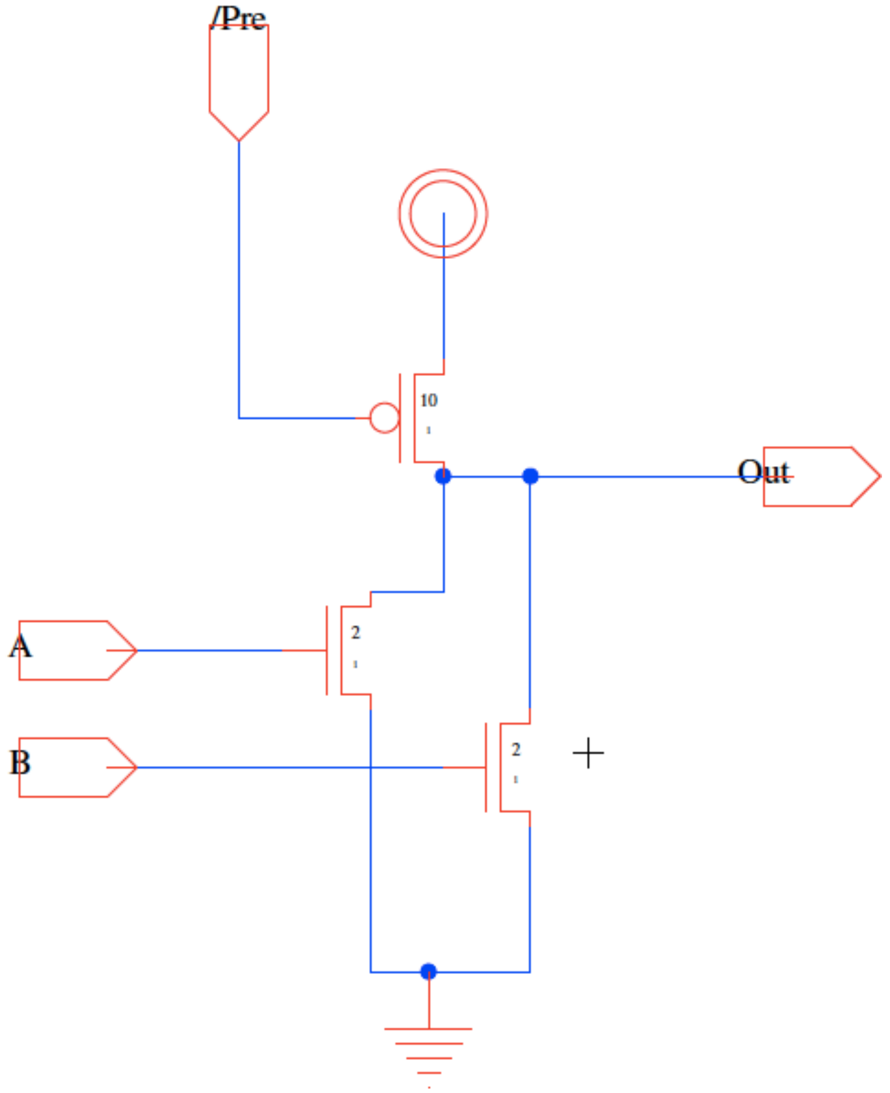
Discuss (Preclass 1)

- Use CLK to disable pullup during evaluation
- What is Vout when:
  - /Pre=0, A=B=0?
  - /Pre=0→1, A=B=0?
  - /Pre=1, A=0, B=0→1 ?



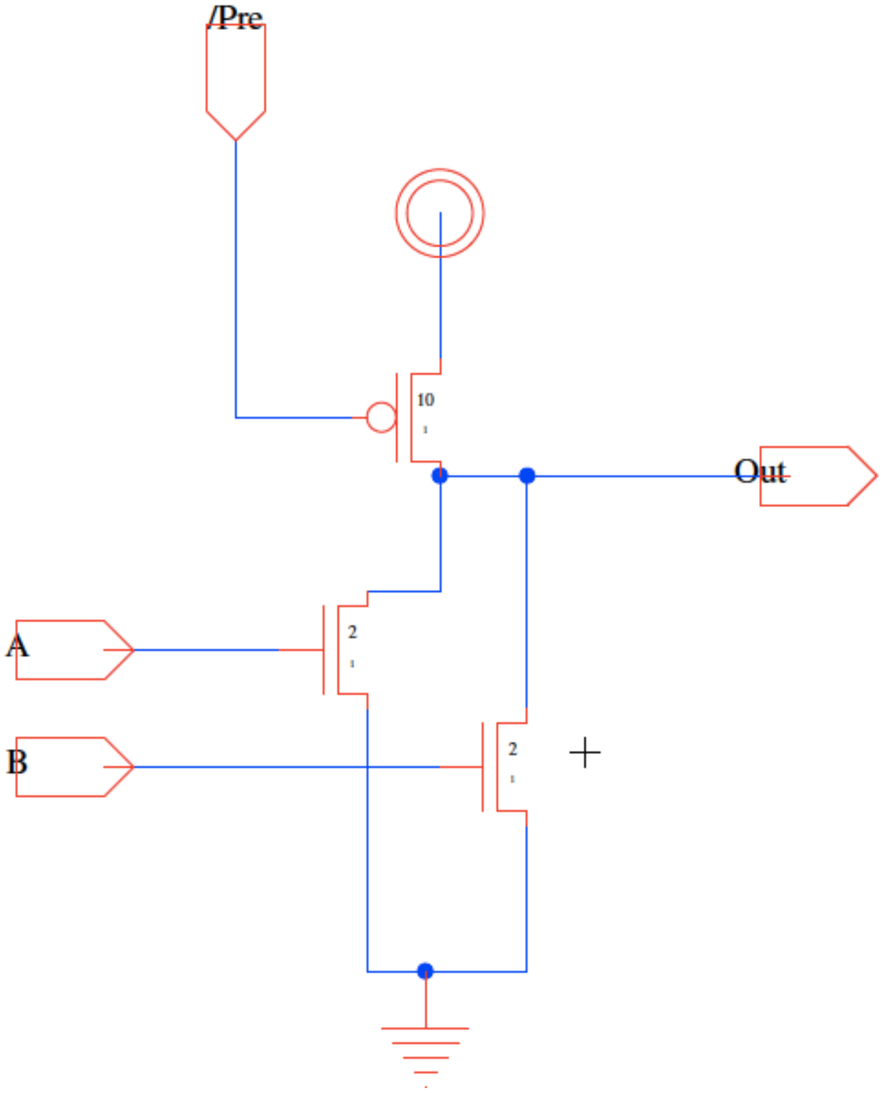
Discuss (Preclass 1)

- Use CLK to disable pullup during evaluation
- What is Vout when:
  - /Pre=0, A=B=0?
  - /Pre=0→1, A=B=0?
  - /Pre=1, A=0, B=0→1 ?
- Sizing implication?
- Concerns?
- Requirements?

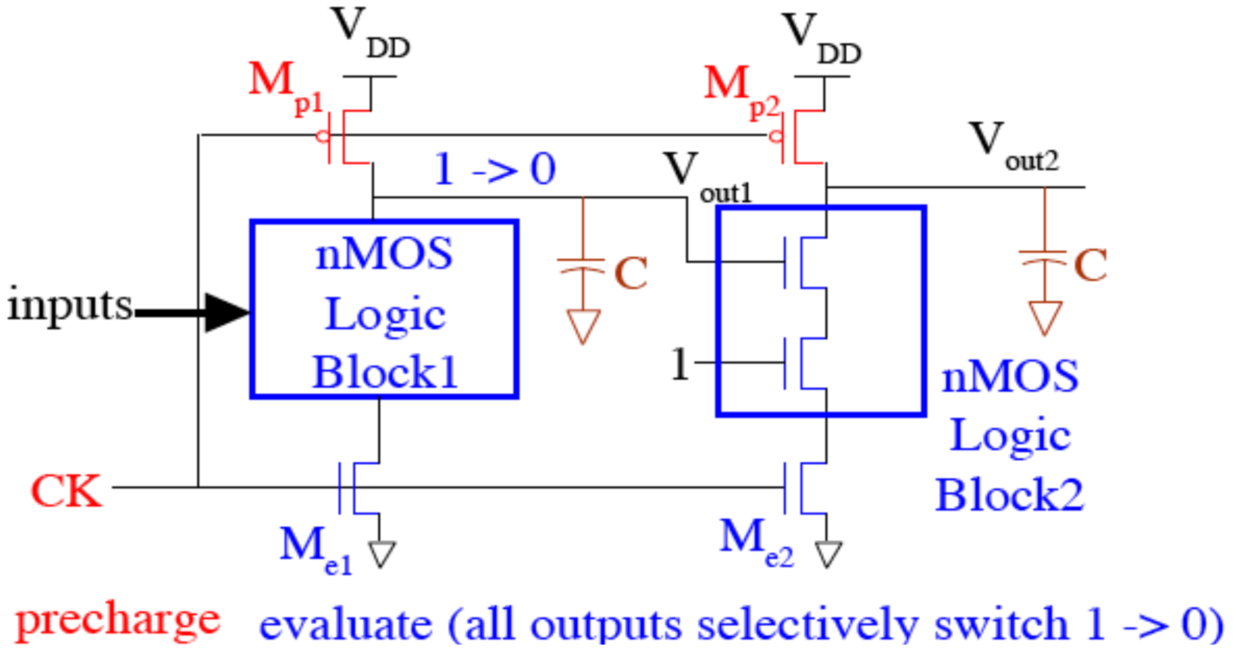


Advantages

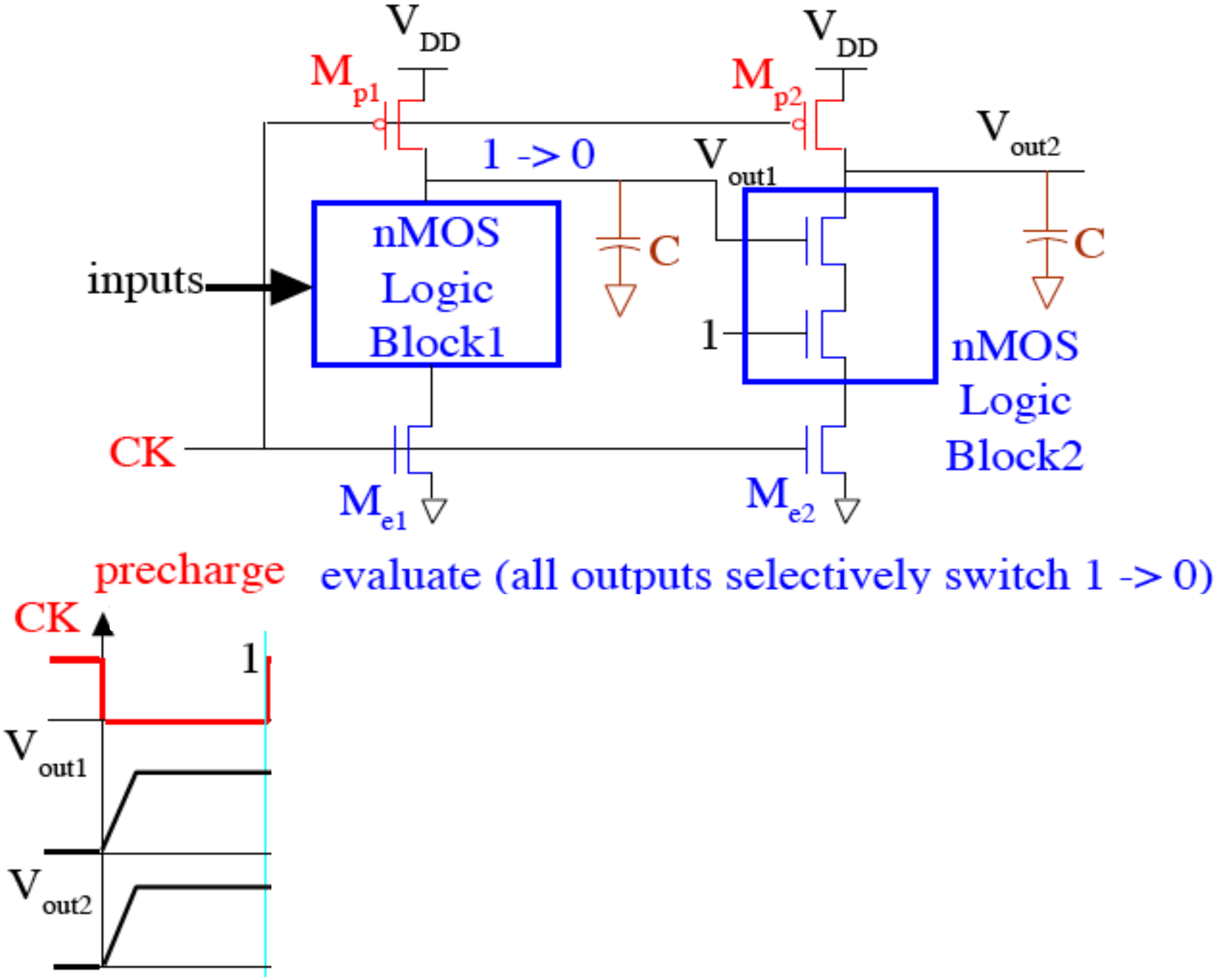
- Large load device
  - Driven by CLK—not data
  - Can pullup quickly without putting load on logic
- Single pulldown network
  - Don’t have to size for ratio with pullup
  - Swings rail-to-rail



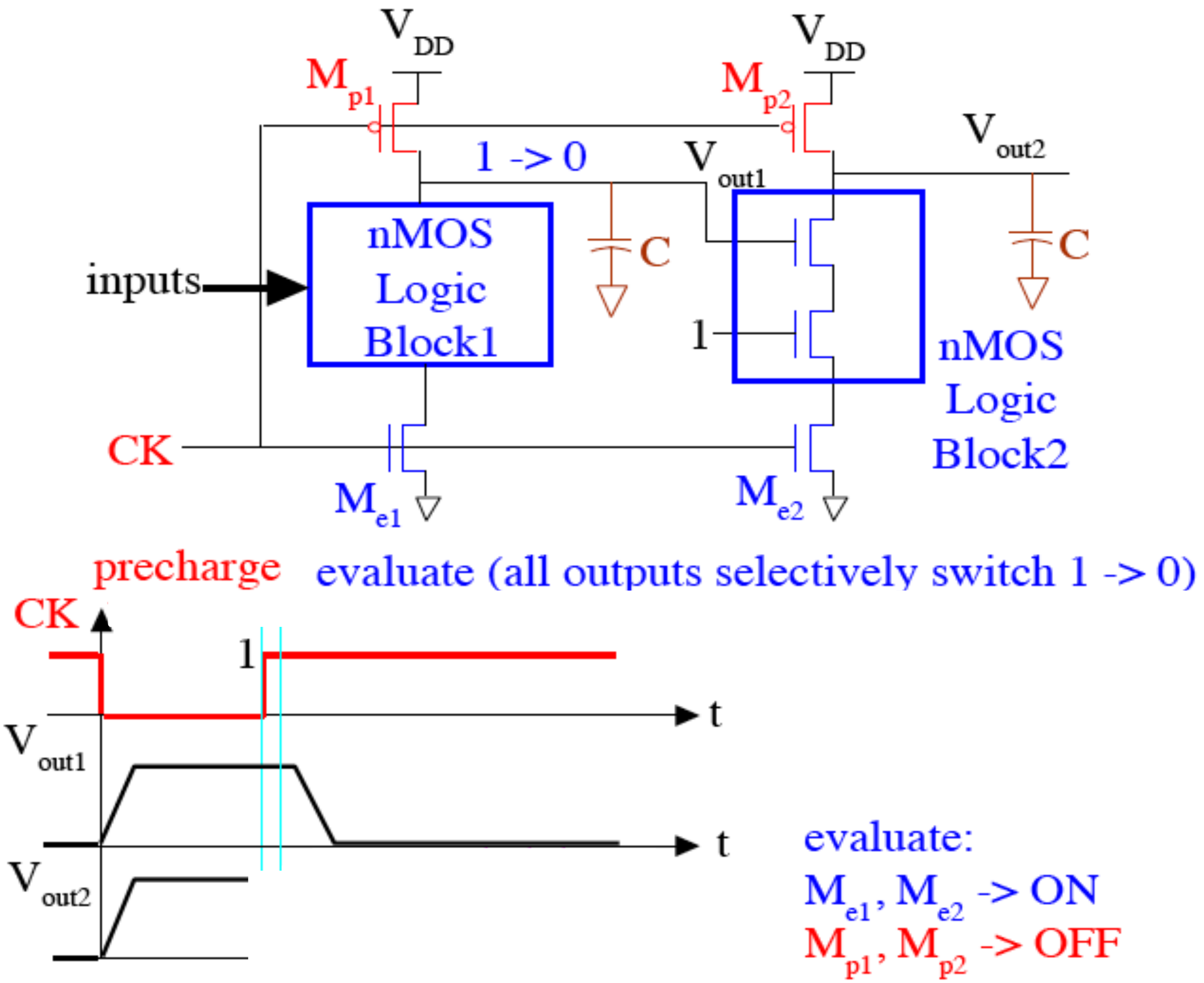
Cascaded Dynamic Logic



Cascaded Dynamic Logic



Cascaded Dynamic Logic

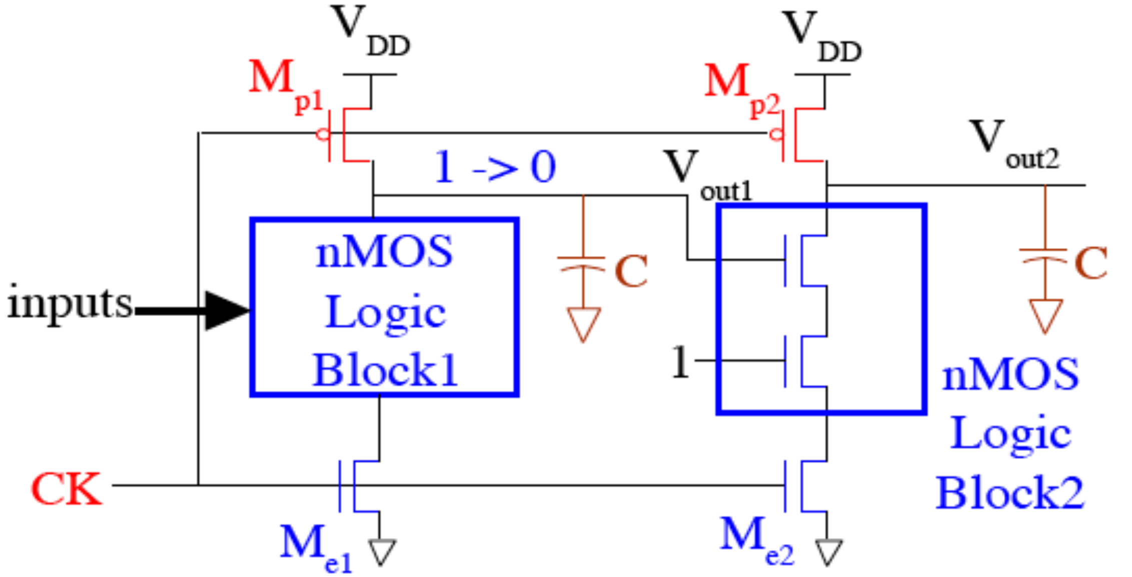


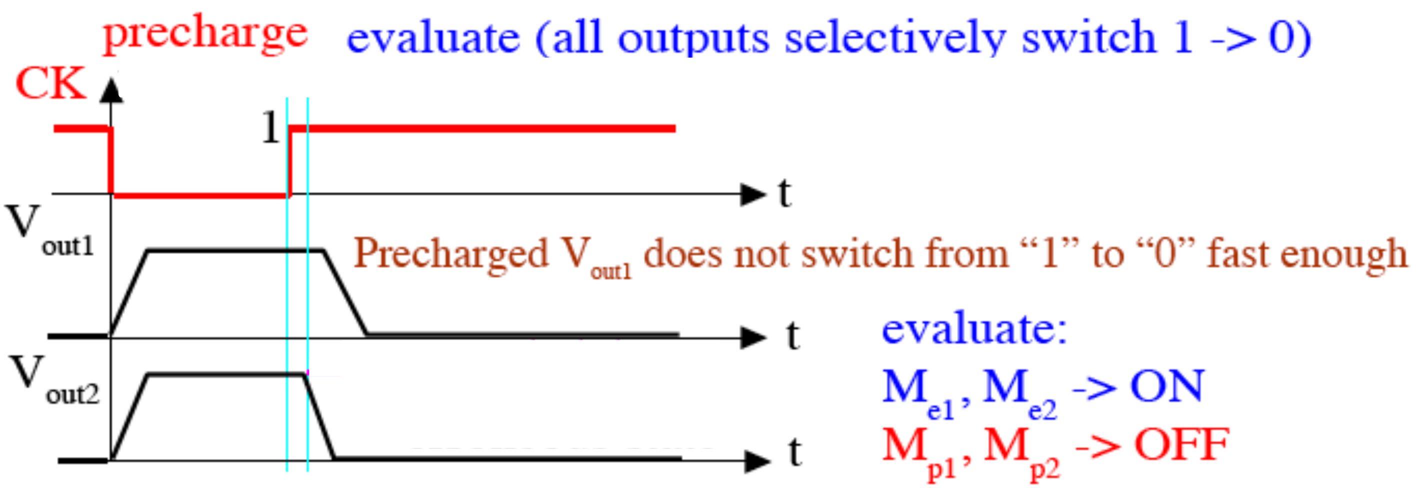
evaluate:

M_{e1}, M_{e2} -> ON

M_{p1}, M_{p2} -> OFF

Cascaded Dynamic Logic





precharge evaluate (all outputs selectively switch 1 -> 0)

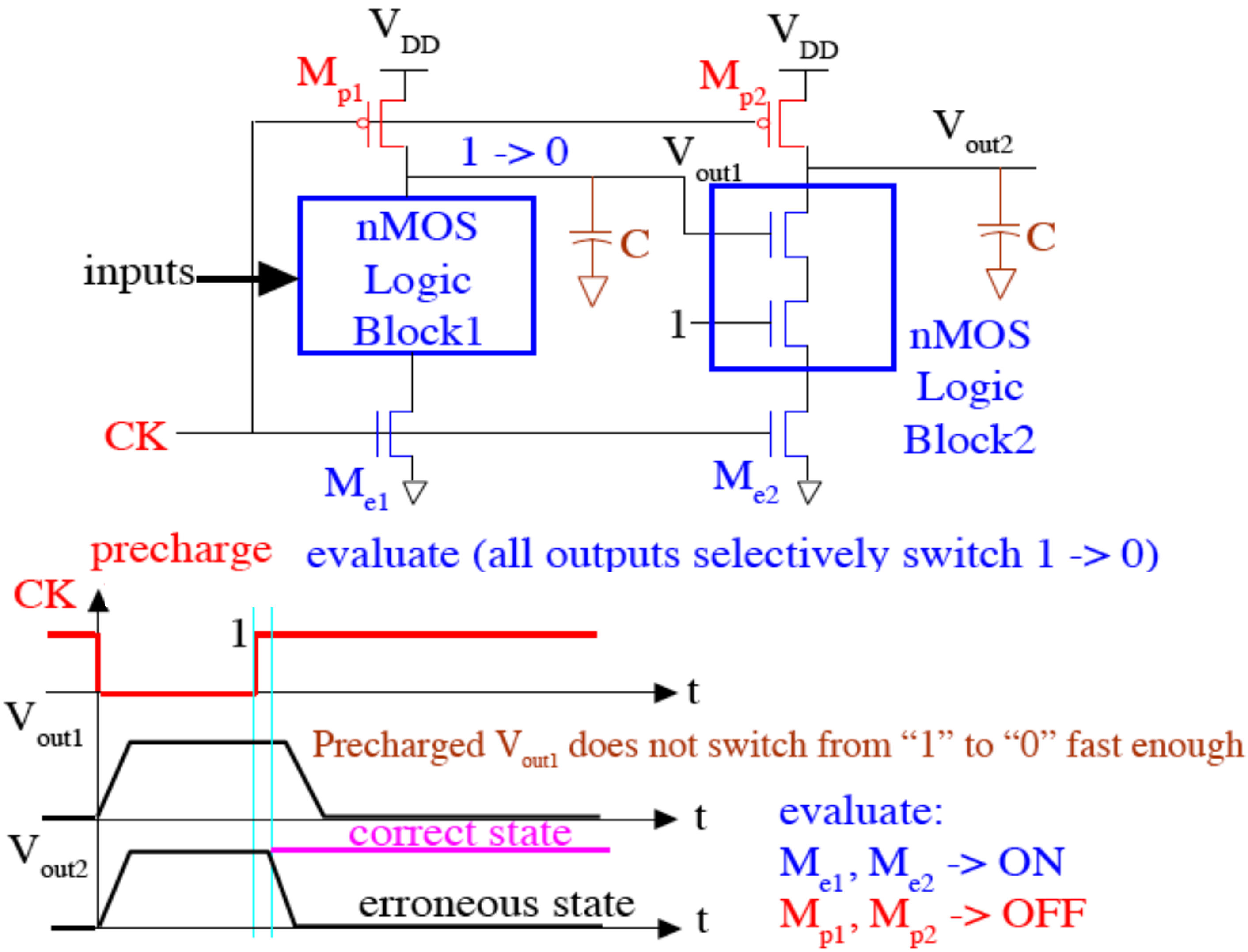
Precharged V_{out1} does not switch from “1” to “0” fast enough

evaluate:

M_{e1}, M_{e2} -> ON

M_{p1}, M_{p2} -> OFF

Cascaded Dynamic Logic



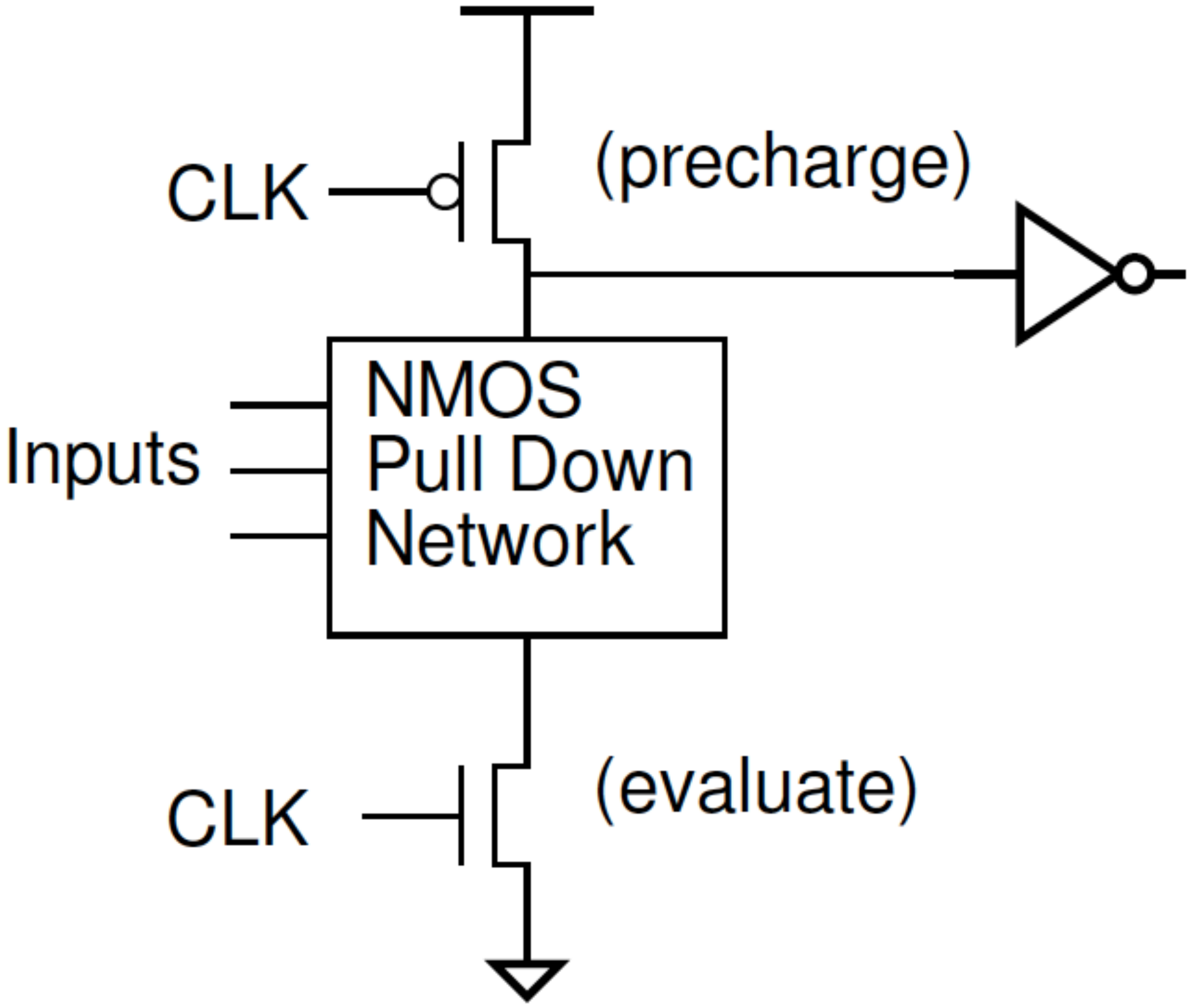
precharge evaluate (all outputs selectively switch 1 -> 0)

Precharged V_{out1} does not switch from "1" to "0" fast enough

evaluate:
M_{e1}, M_{e2} -> ON
M_{p1}, M_{p2} -> OFF

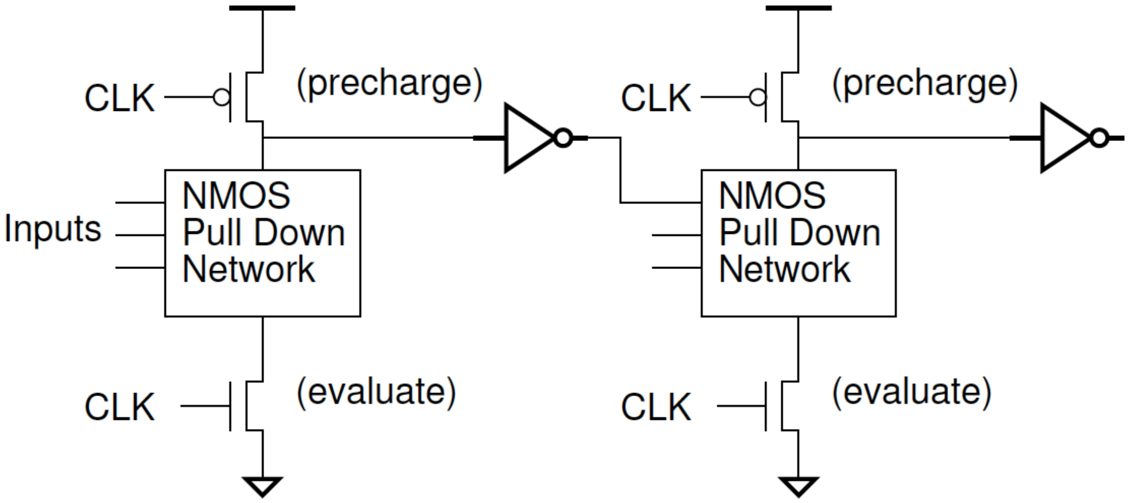
PROBLEM: ALL STAGES MUST EVALUATE SIMULTANEOUSLY
SINGLE CLOCK DOES NOT PERMIT PIPELINING OF STAGES

Domino Logic

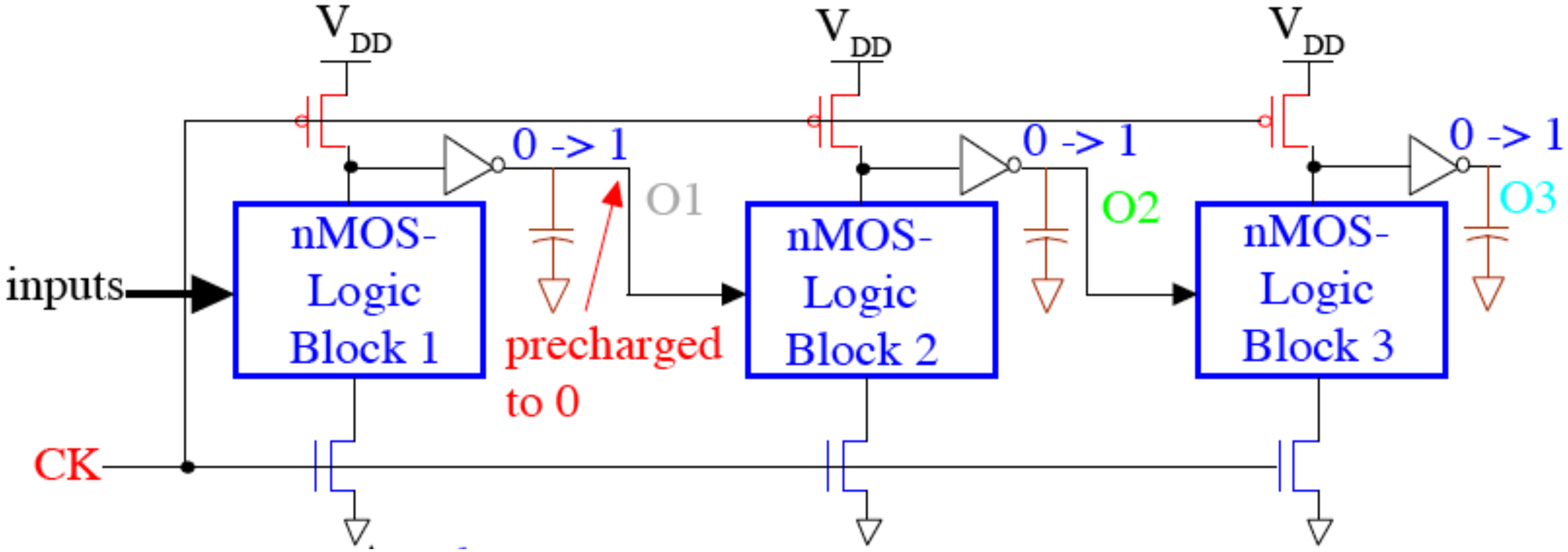


Requirements

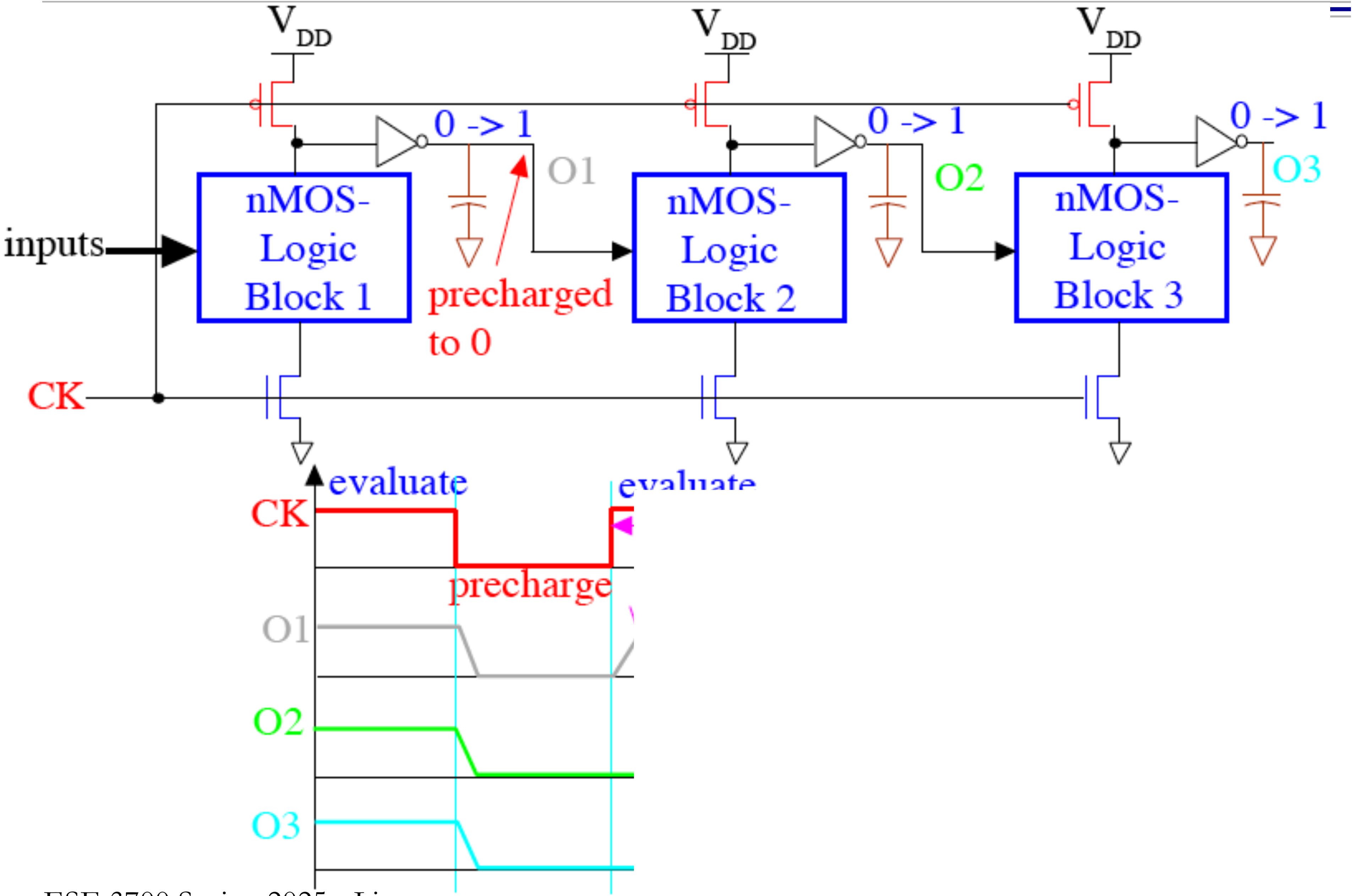
- Single transition
  - Once transitioned, it is done → like domino falling
- All inputs at 0 during precharge
  - “Outputs” pre-charged to 1 then inverted to 0
- Non-inverting gates fundamental gate



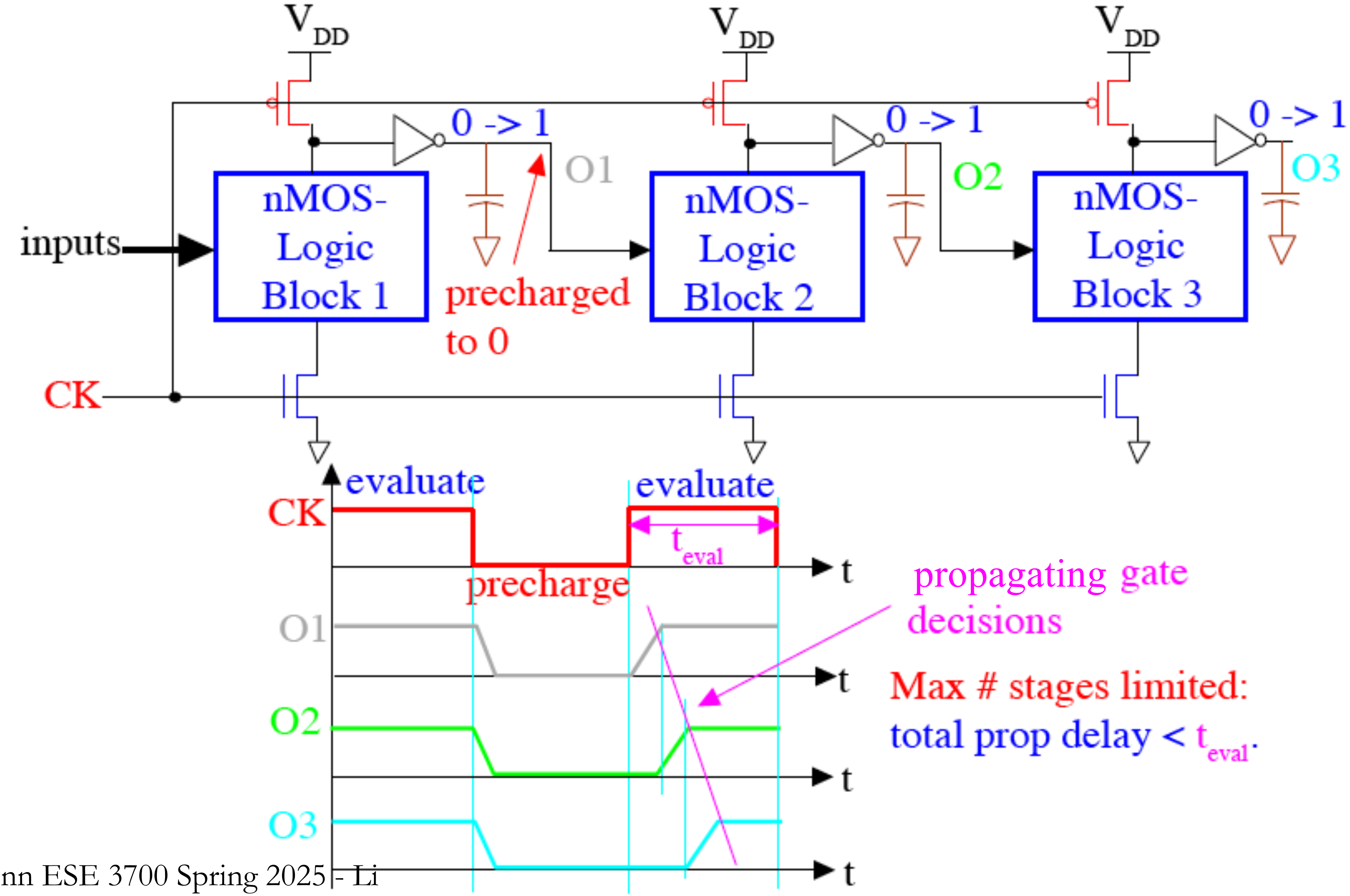
Cascaded Domino CMOS Logic Gates



Cascaded Domino CMOS Logic Gates

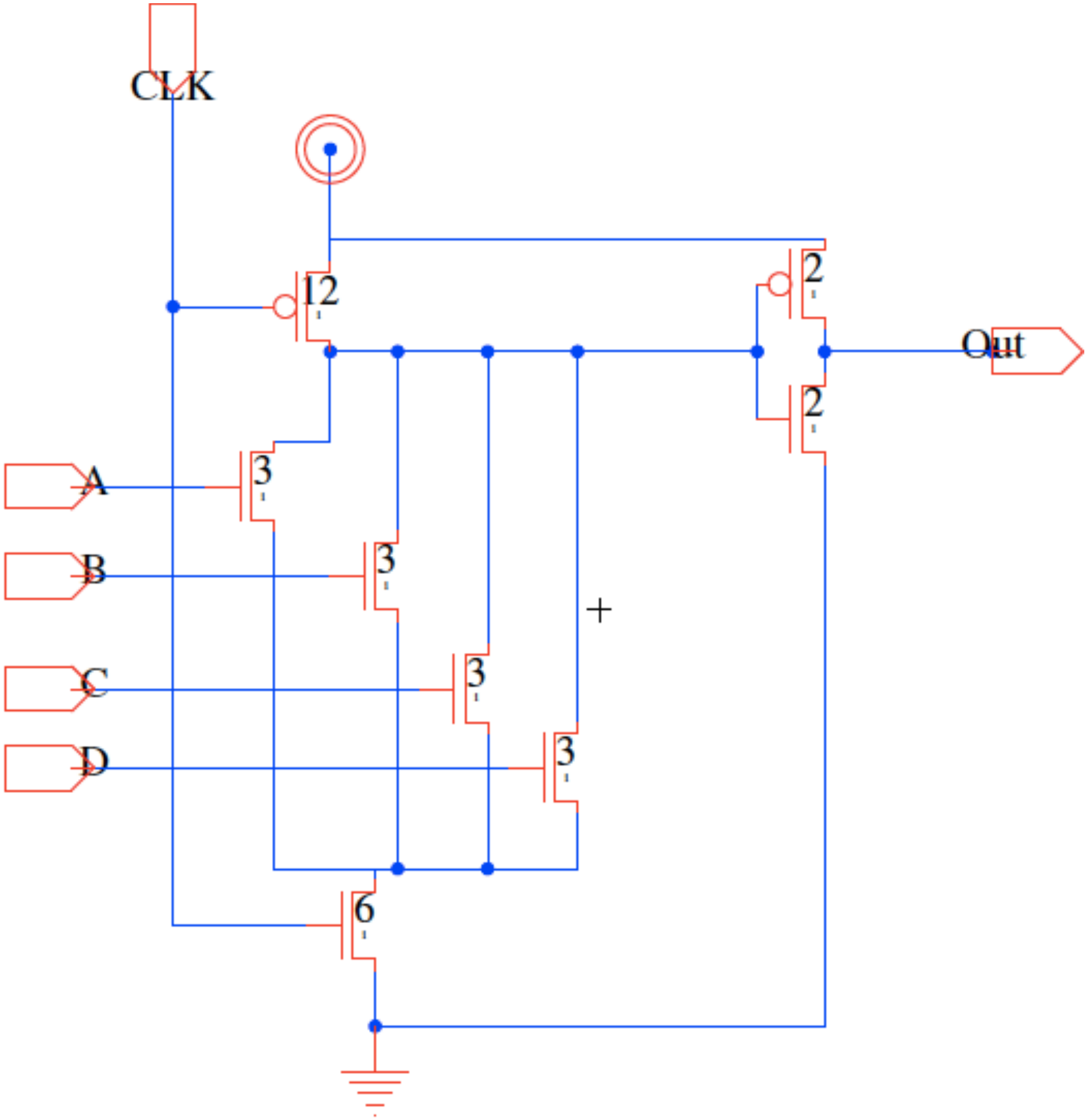


Cascaded Domino CMOS Logic Gates



Max # stages limited: total prop delay < t_{eval}.

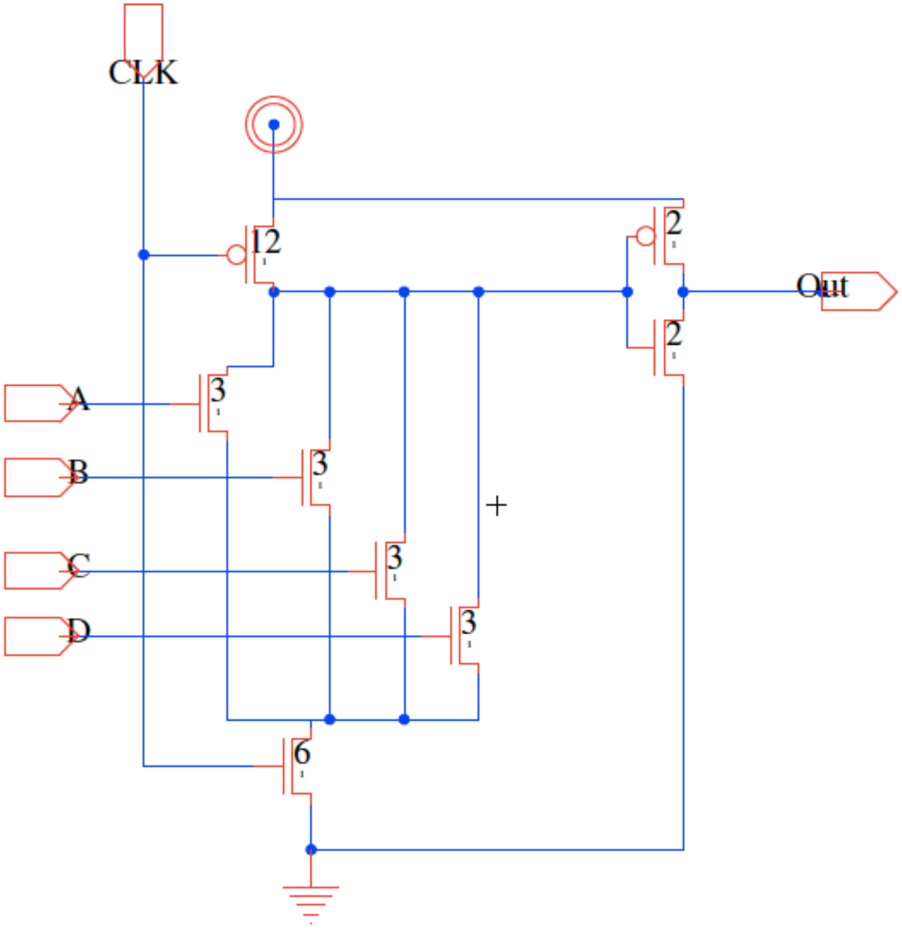
Domino or4 (Preclass 2)



Domino Logic (Preclass 2)

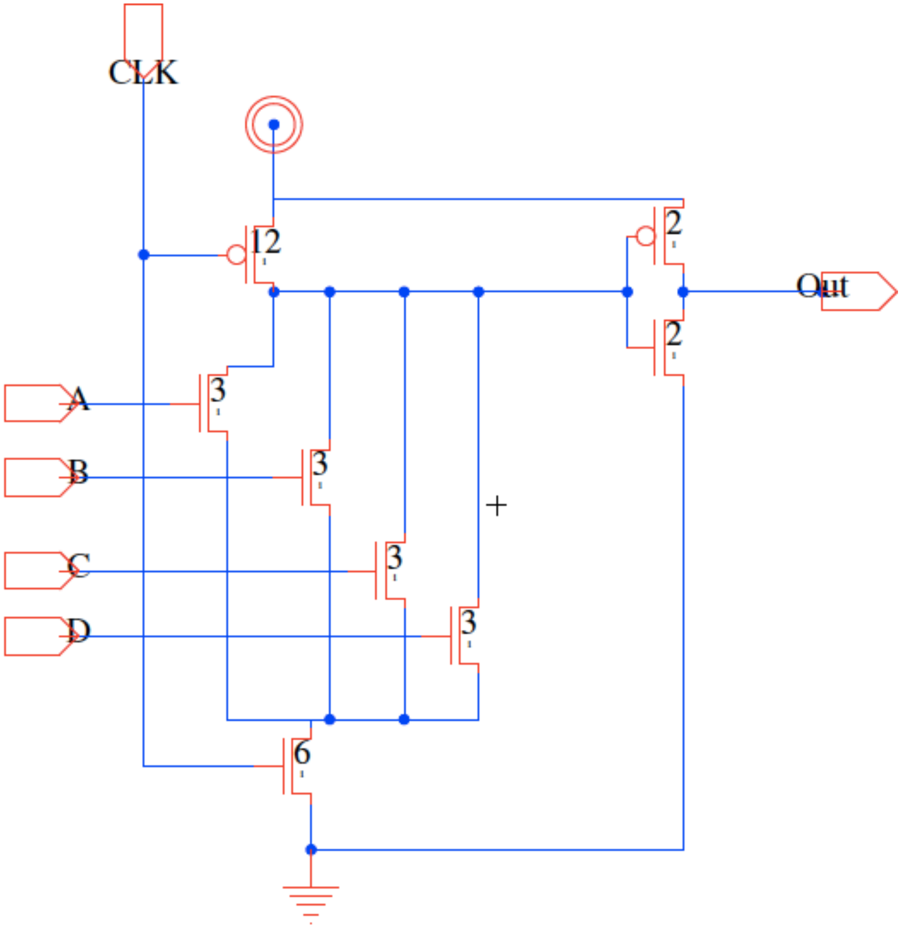
- Performance
  - $R_0/2$ input

- Compare to CMOS cases?
  - nor4
  - or4
  - nand4



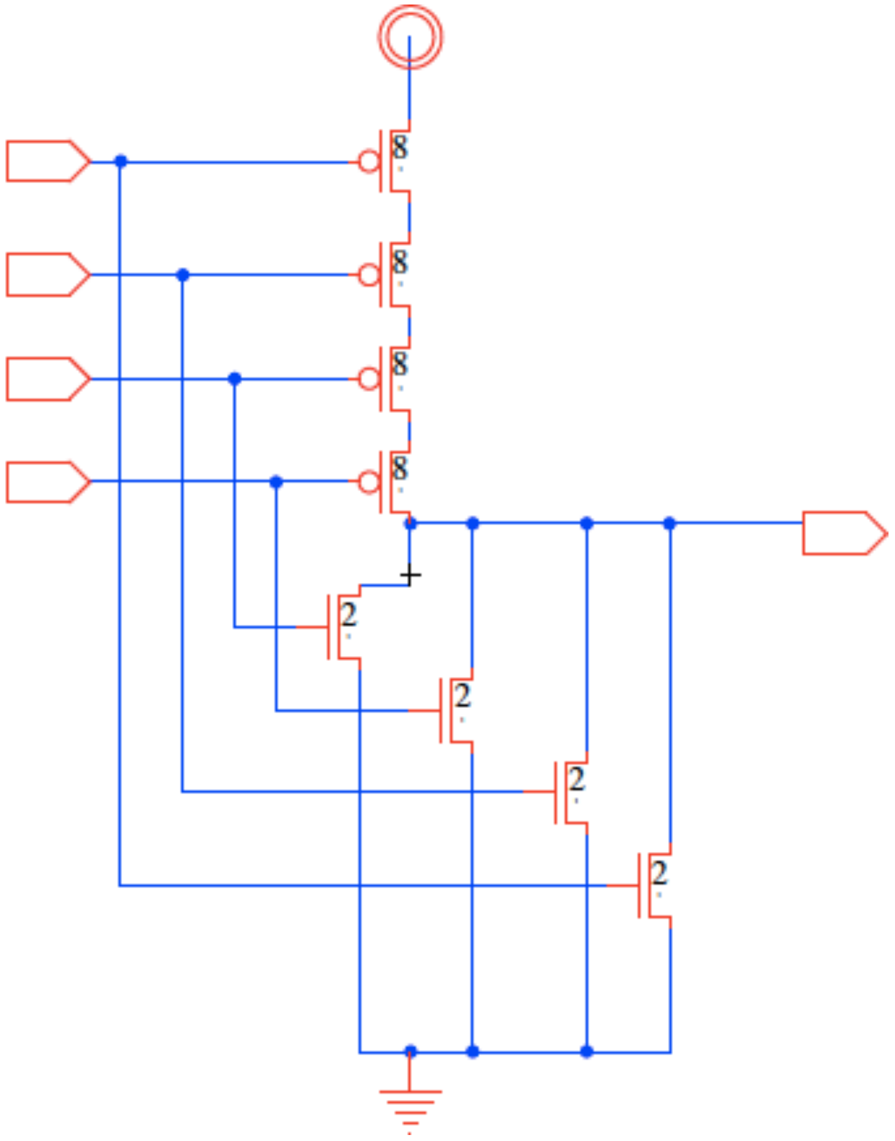
Dynamic OR4 (Preclass 2)

- Precharge time?
- Driving input
  - With $R_0/2$ inverter
- Driving inverter?
- Self output Delay?



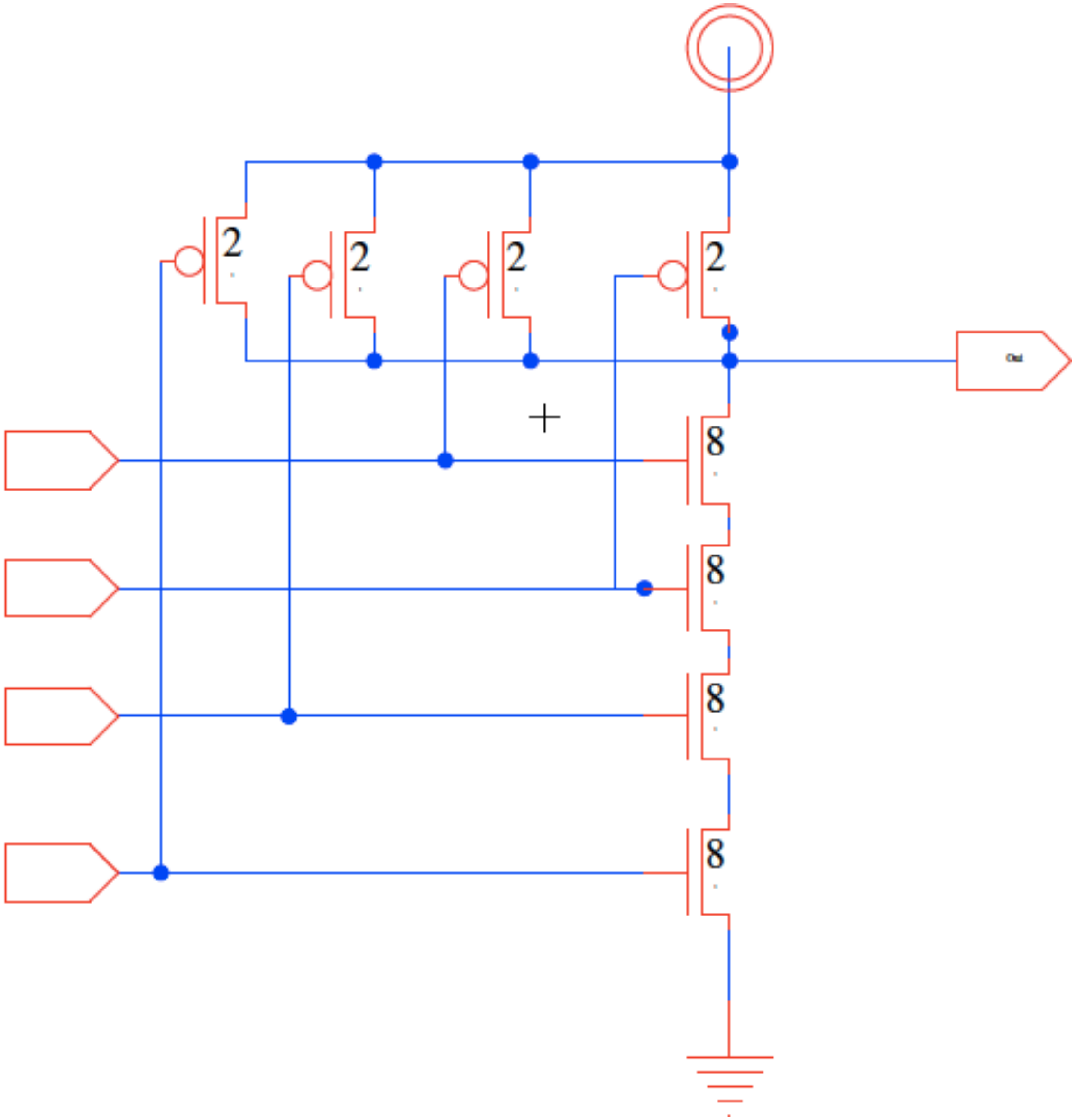
CMOS NOR4 (Preclass 2)

- Driving input
  - With $R_0/2$ inverter
- Self output Delay?



CMOS NAND4 (Preclass 2)

- Driving input
  - With $R_0/2$
- Driving self cap?



Dynamic Logic Issues

- Noise sensitive
  - During evaluation phase, when output is high it’s floating and therefore more susceptible to noise
- Power
  - Eliminates static current
  - Higher activity factor—always a 0→1 transition, large pre-charge device dissipates extra switching power

Observe

- Better (lower) ratio of input capacitance to drive strength
- Particularly good for
  - Driving large loads
  - Large fanin gates
- Harder to design with
  - Timing and polarity restrictions
  - Avoiding noise
    - Especially with today's high variation tech
- Can consume more energy

Idea

- Clock discipline simplifies logic composition
  - Breaking logic up with registers allows circuit to run at high frequency
  - Abstracts many internal timing details
    - Setup/Hold time, clk→q delay
  - Just concerned with making clock period long enough
- Dynamic/clocked logic
  - Only build/drive one pulldown network
    - Domino Logic
  - Fast transition propagation
  - Spend delay (capacitance) on pullup of critical path of logic
  - More complicated design, power dissipation
    - Reserve for when most needed

Admin

- Homework 6
  - Due **Friday 4/11**
- Wednesday 4/16 Midterm 2
  - 1:45pm-3:45pm **in class**
  - **Midterm 2 Review session (4/16) - in class**
  - Lectures 11-18
  - Closed note, calculator allowed
  - All old exams online
    - 2015-2024
  - **Do review your preclass!!**

Acknowledgement

- Prof. André DeHon (University of Pennsylvania)
- Prof. Tania Khanna (University of Pennsylvania)