

# **EEC 118 Lecture #13: Memories**

**Rajeevan Amirtharajah  
University of California, Davis**

**Jeff Parkhurst  
Intel Corporation**

# Announcements

---

- **Finish Lab 5 this week**
- **Quiz 3 Wednesday**
- **Homework 7 issued later this week, due next week**
- **Lab 6 next week, report due June 2**

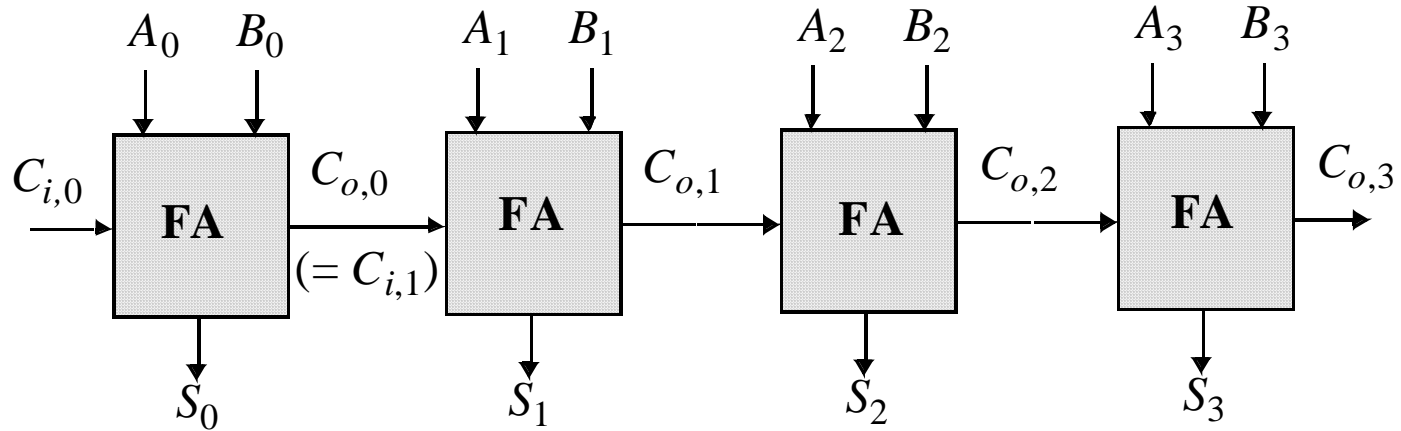
# Outline

---

- **Review: Adders (Rabaey 11.3)**
- **Multipliers: Rabaey 11.4**
- **Memories: Rabaey 12.1-12.2 (Kang & Leblebici, 10.1-10.6)**

# Review: Ripple Carry Adder

---



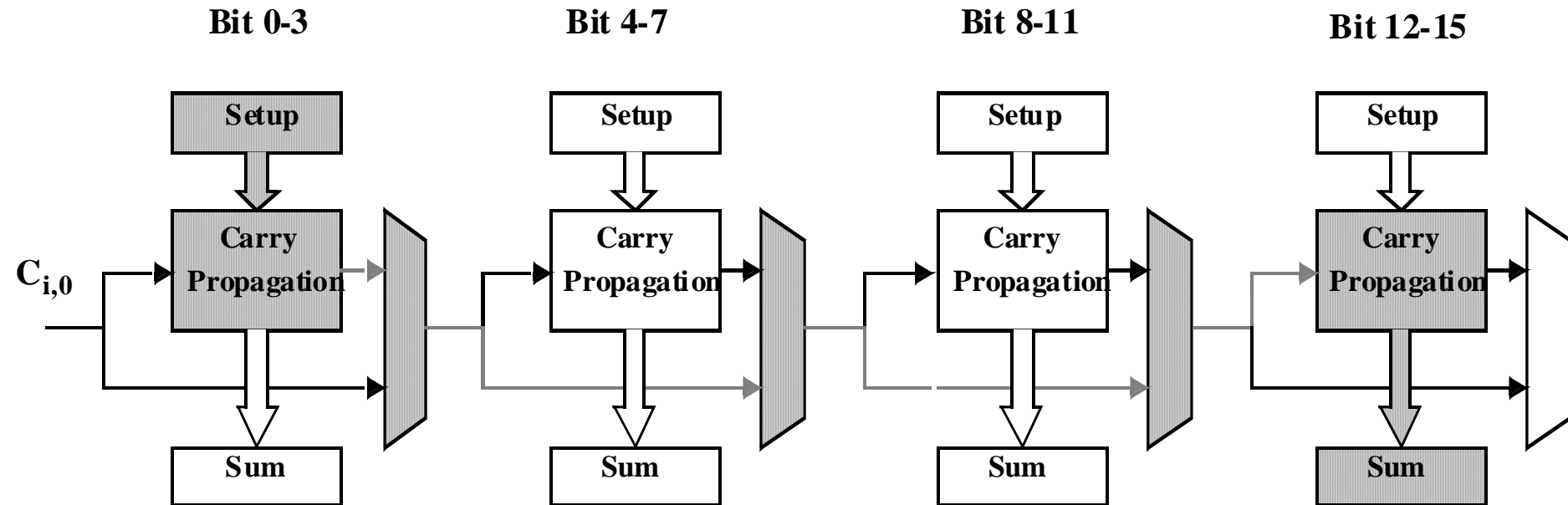
**Worst case delay linear with the number of bits**

$$t_d = O(N)$$

$$t_{\text{adder}} \approx (N - 1)t_{\text{carry}} + t_{\text{sum}}$$

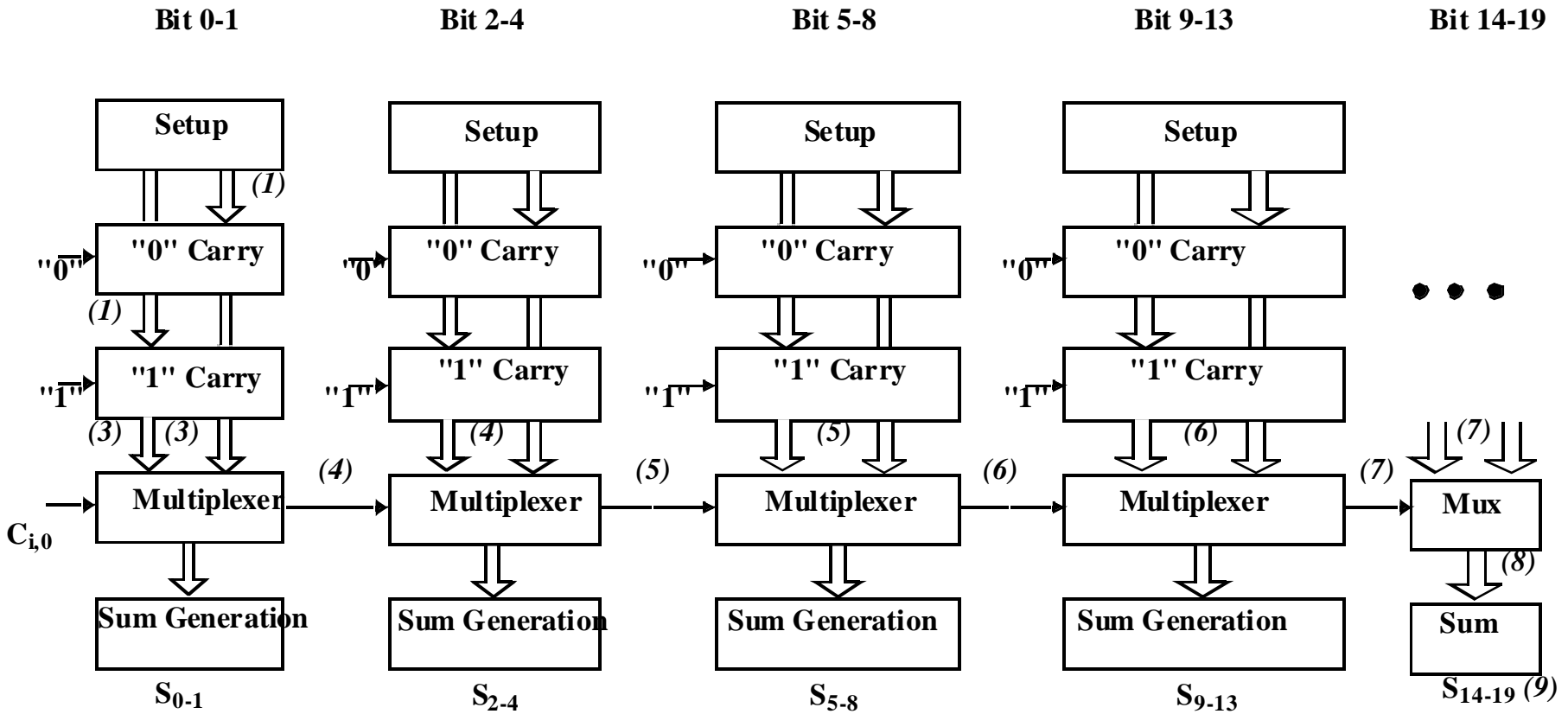
Goal: Make the fastest possible carry path circuit

# Review: Carry Bypass Adder



- Note that this is done at the expense of a MUX in the carry delay path !!
- Delay increases linearly but with smaller slope than RCA.

# Review: Carry Select Adder



$$t_{add} = t_{setup} + P \cdot t_{carry} + (\sqrt{2N})t_{mux} + t_{sum}$$

# Memory and Performance Trend I

---

- **Memory is becoming a key factor in the performance of a computer**
- **1<sup>st</sup> generation computers had just system memory**
  - Very slow DRAM
  - As microprocessors got faster, the bottleneck in performance was data access
- **2<sup>nd</sup> generation computers added cache memory**
  - This provided faster access to small localized memory that was being read or written
    - Memory placed on front side bus (off-chip)
    - Performance increased
  - As processors got faster memory access again became the bottleneck

# Memory and Performance Trend II

---

- **3<sup>rd</sup> generation computers added on chip cache**
  - These caches started out 16K and were termed level 1 cache
  - Soon we had level 1 and level 2 cache
  - Currently we have three levels of cache on chip that run at processor frequency, then access main memory if data can't be found in any of these caches
  - Moving to stacking memory die on top of processor



# Types of Memory I

---

- **ROM: read-only memory**
  - Non-volatile – mask programmed
- **RWM: read-write memory (RAM, random access memory)**
  - SRAM: static memory
    - Data is stored as the state of a bistable circuit
    - State is retained without refresh as long as power is supplied
  - DRAM: dynamic memory
    - Data is stored as a charge on a capacitor
    - State leaks away, refresh is required

# Types of Memory II

---

- **NVRWM: non-volatile read-write memory (also called NVRAM, non-volatile random access memory)**
  - Flash (EEPROM): ROM at low voltages, writable at high voltages (Electrically Erasable Programmable Read-Only Memory)
  - EPROM: ROM, but erasable with UV light (falling out of common usage)

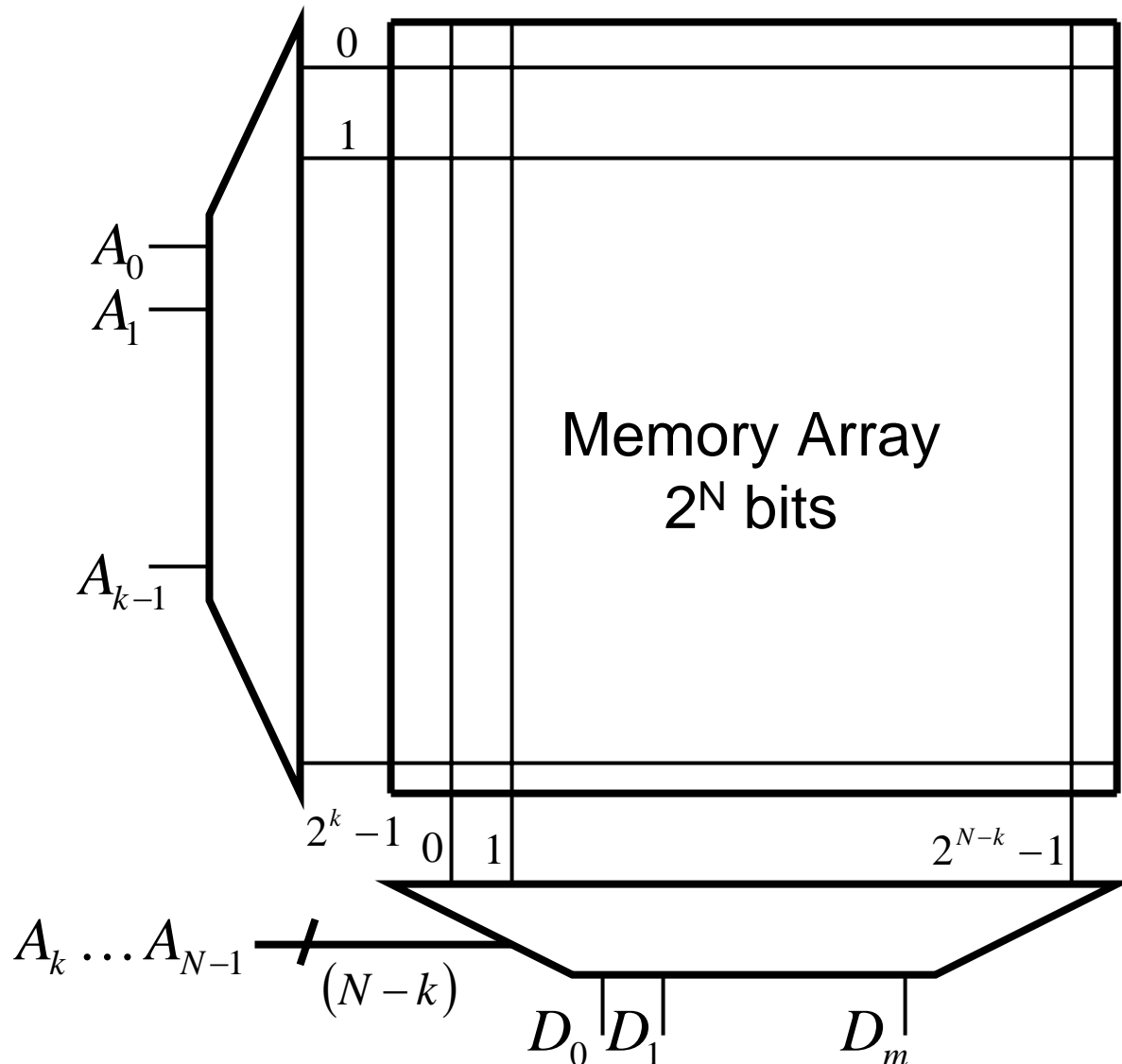
# Memory Usage in Computers

---

- **DRAM Memory**
  - Main memory storage. Used for data and programs
- **SRAM Memory**
  - Faster than DRAM, however, uses more transistors
    - Used to be used for external cache
    - Variant used in internal cache (on chip cache)
- **FLASH Memory and ROM**
  - Used for BIOS data storage in PCs
  - Also used to store pictures, MP3 files for digital cameras and MP3 players – eventually for hard disk

# Basic Memory Array Structure

- **Memory cells arranged in a rectangular array**
- **Rows correspond to data words**
  - Accessed through a row decoder
- **Columns to individual bits**
  - Selected through a column mux
- **Bit voltage amplified by sense amplifier**



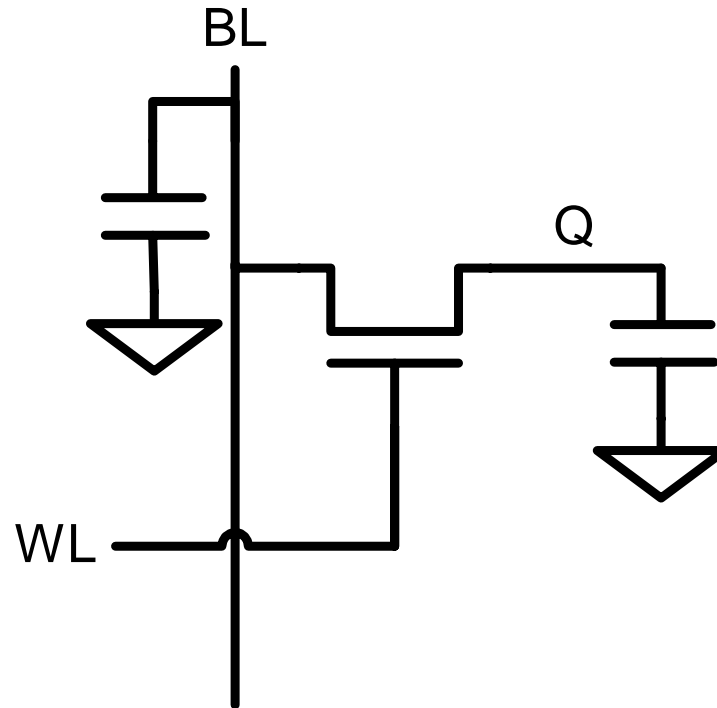
# Memory Circuit Operation

---

- **Wordlines (WL) control row (word) access**
  - Usually control gates of pass transistors
- **Bitlines (BL) route column data (individual bits)**
  - Bitlines usually precharged high (like dynamic logic)
  - Memory cells discharge bitline depending on stored data (bitline left high if cell stores 1, bitline discharged if cell stores 0)
  - Bitline swings usually small (10s – 100s of mV) and must be amplified by sense amplifiers
  - Synchronous or asynchronous timing can be used
- **Memory cells store data value**
  - Static vs. dynamic, single or multiple bits, etc.

# DRAM

---



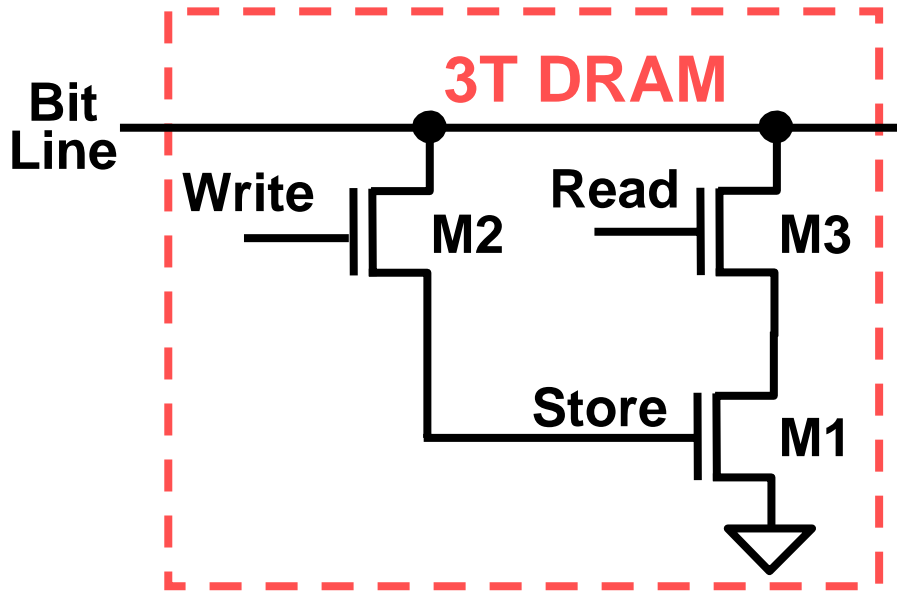
- **Smaller cell size (1 transistor or 1T cell)**
  - Reason for inexpensive memory in computers
  - Tradeoff of area (memory density) vs. speed and complexity (refreshing)

# DRAM Issues

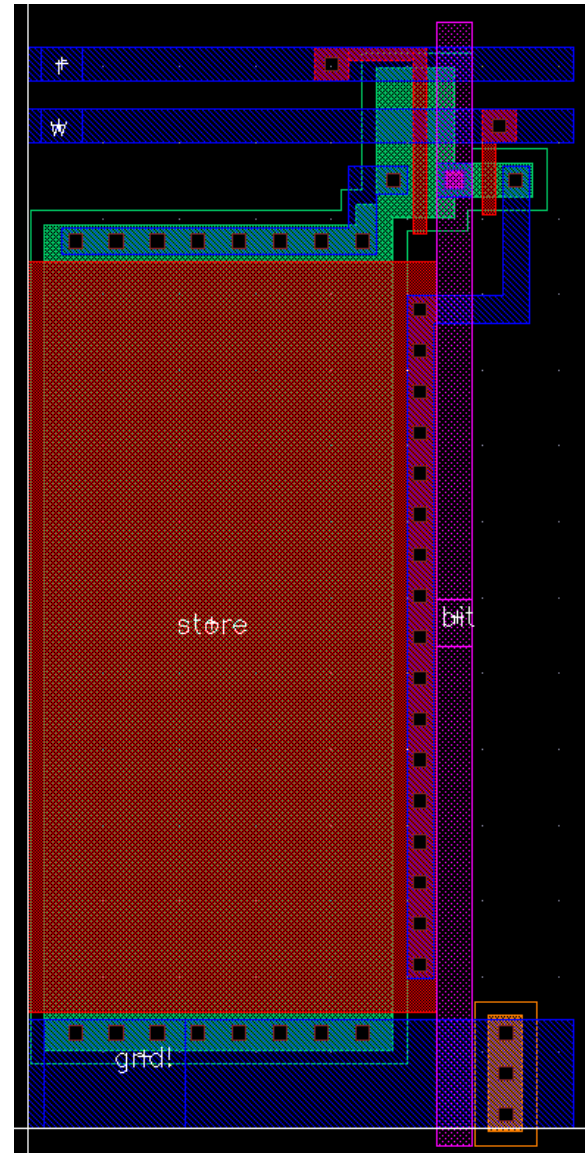
---

- **Must be periodically refreshed**
  - Reads are destructive (modify voltage stored on capacitor)
  - Every read followed by a refresh of the bit (write back of read value)
- **No static power dissipation**
- **Output voltage is charge sharing result of storage capacitor and bitline capacitance**
  - More complex sense amplifiers
  - Higher noise susceptibility
- **Requires different CMOS process than high performance logic**
  - Not compatible with cache in microprocessors

# 3T DRAM Cell



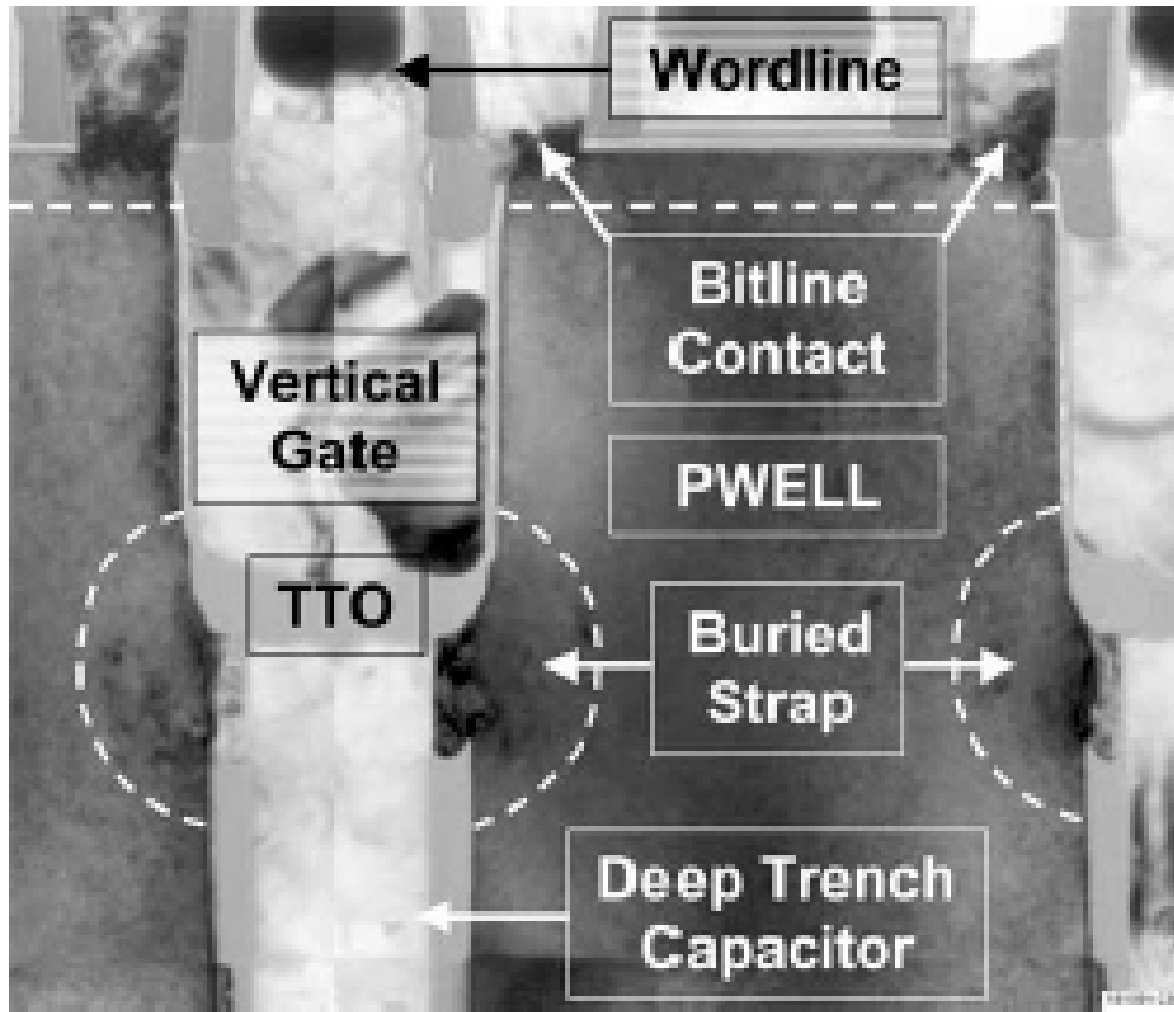
- Early DRAM technology
- Gate cap of M1 stores bit
- Nondestructive reads
- Storage node voltage  $< V_{DD}$ 
  - Compensate with boosted wordline





# Advanced DRAM Process

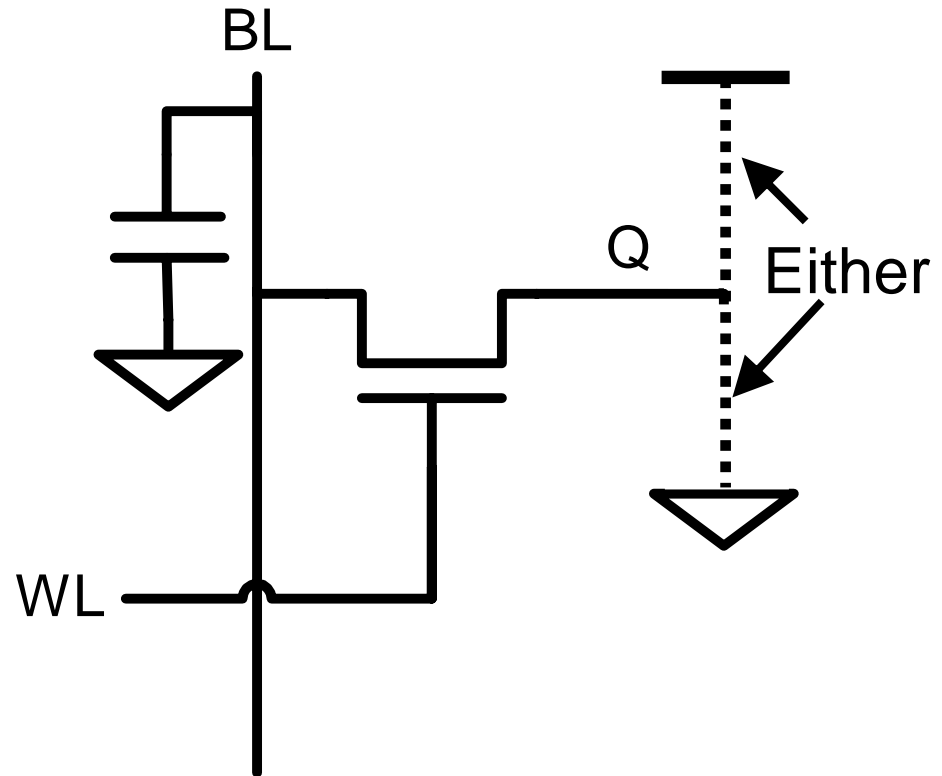
---



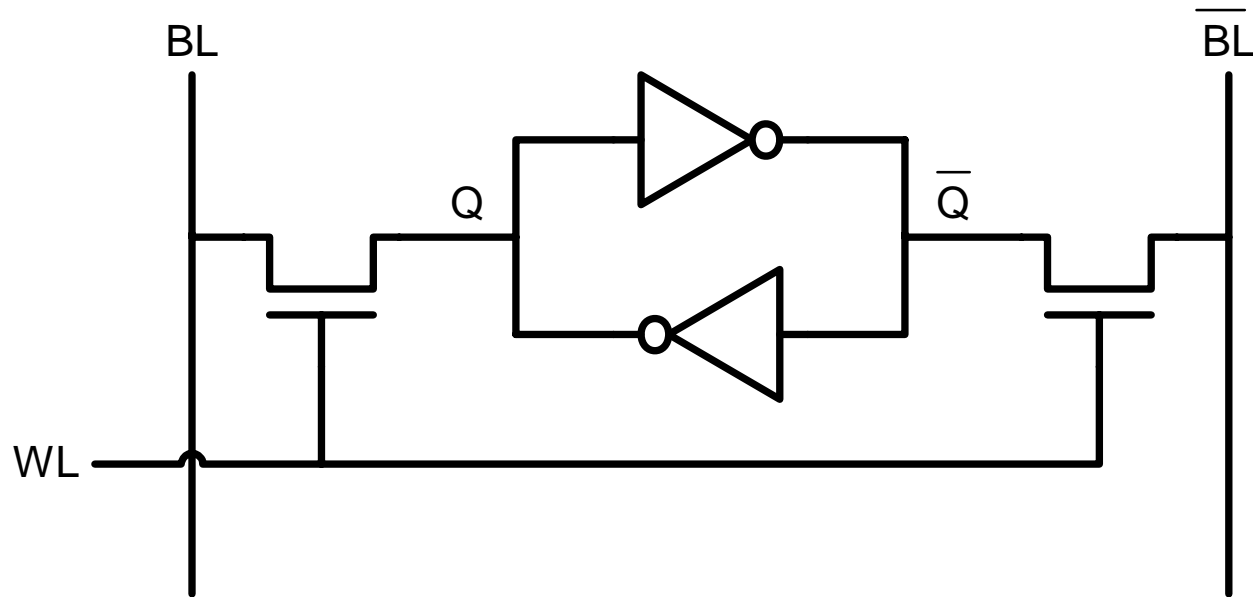
- **Vertical transistor, trench capacitor (Beintner, JSSC 04)**

# ROM

- **Dotted lines refer to either set at '1' or '0'**
  - PROM: Replace dotted lines with fuses
- **Small cell size (1T cell)**
- **Not necessary to refresh**
- **No static power dissipation**
- **Output voltage is set by WL duration**

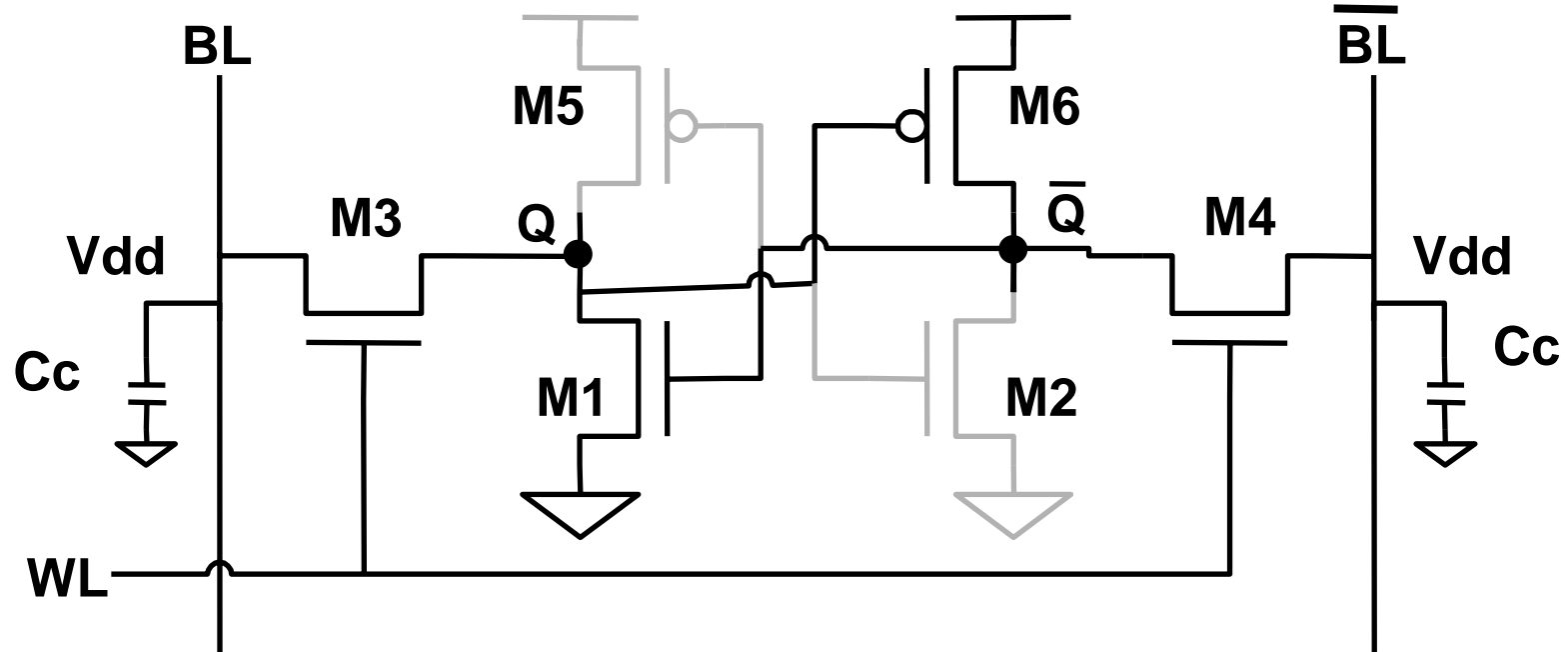


# SRAM Cell



- **Cross-coupled inverters: bistable element**
- **Density is important in memories**
  - Single NMOS pass transistor used for reading/writing
  - Transistor sizes should take up minimum area
- **Faster than DRAM since typically fewer cells**
- **No refresh required (nondestructive reads)**

# SRAM Design: Read "0"



- Prior to read operation, voltage at node Q = 0V and  $\overline{Q}$  = Vdd, bit lines precharged to Vdd
- Transistors M3 and M4 are turned on by word line (WL) select circuitry

# SRAM Design: Read “0”

---

- **Transistors M3 and M4 are turned on by WL line select circuitry**
  - $C_c = V_{dd}$  to start...capacitance discharges through M1.
  - Need to make sure the ratio between M1 and M3 does not allow Q to go above  $V_{tn}$ .
    - Otherwise node  $\bar{Q}$  accidentally discharged
    - Conservative since there will also be charge sharing at that node as well between small internal node capacitance and large bitline capacitance
  - Sense amp detects that node Q was a stored 0 due to the minor drop of voltage on the bitline

# SRAM Design: Read “0”

---

- **Data must not be destroyed when bitline voltage different than storage node**
  - $V_Q$  must not exceed the threshold of the inverter (assumed to be  $V_{DD}/2$ ), more conservative to keep it below  $V_{TN}$

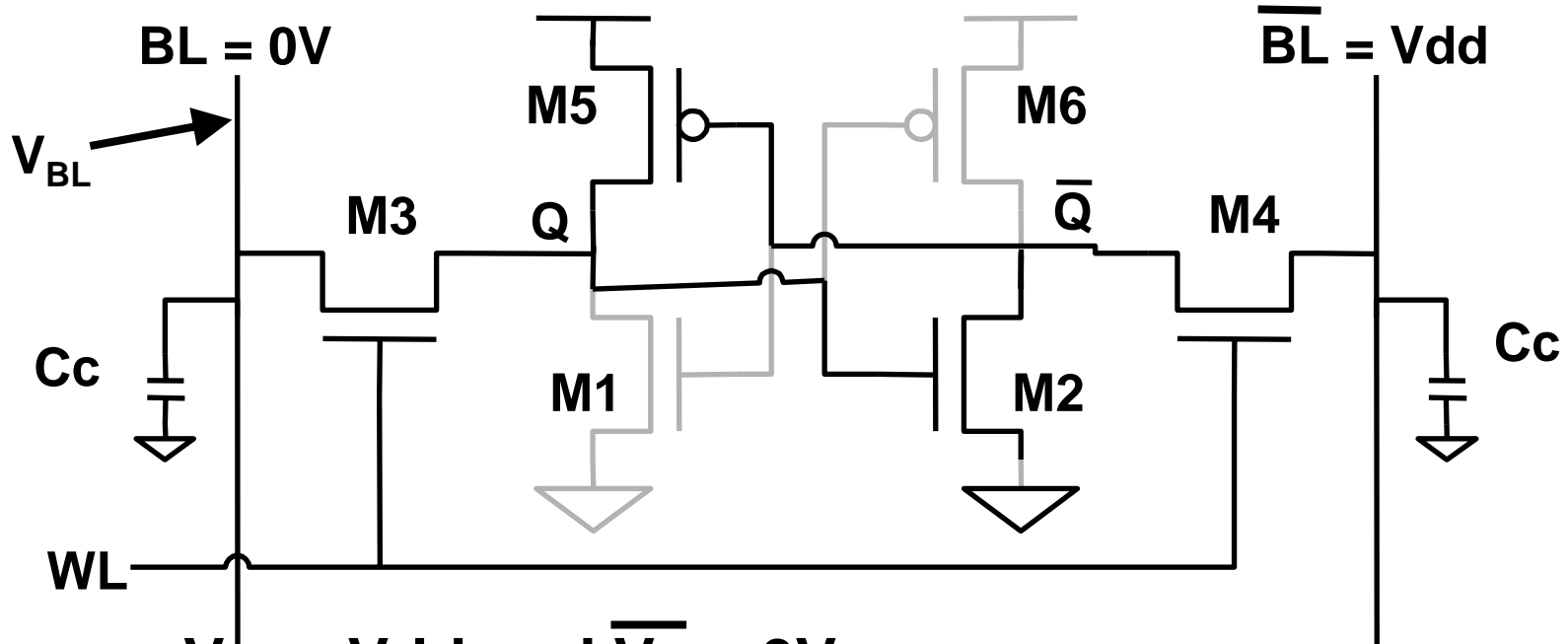
- **Assume  $V_{BL}$  initially remains at  $V_{DD}$ : M3 in saturation, M1 in linear**

$$\frac{k_{n,3}}{2} (V_{DD} - V_Q - V_{TN})^2 = \frac{k_{n,1}}{2} (2(V_{DD} - V_{TN})V_Q - V_Q^2)$$

$$\frac{k_{n,3}}{k_{n,1}} = \frac{\left(\frac{W}{L}\right)_3}{\left(\frac{W}{L}\right)_1} < \frac{2(V_{DD} - 1.5V_{TN})V_{TN}}{(V_{DD} - 2V_{TN})^2}$$

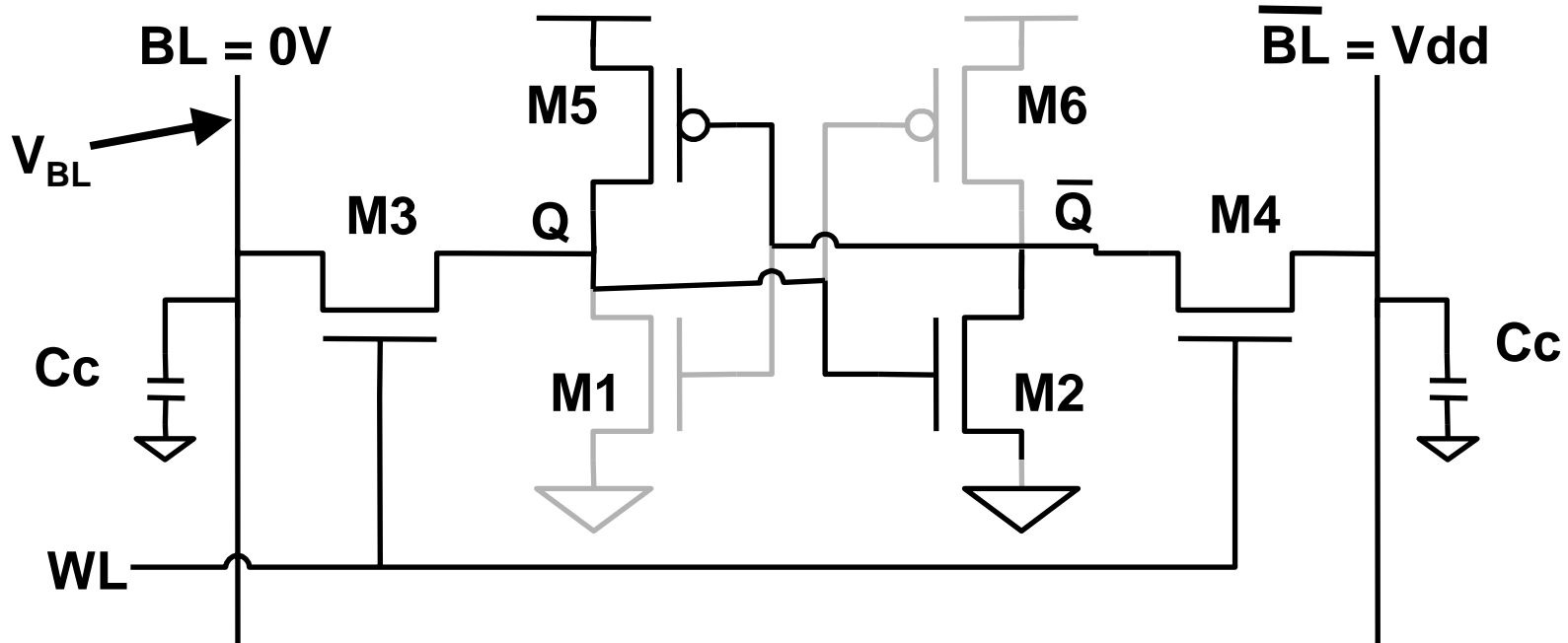
**Guarantee  $V_Q < V_{TN}$ ,  
plug into  $I_D$  equations:  
 $I_{D3} < I_{D1}$  at  $V_Q = V_{TN}$**

# SRAM Design: Write "0" (1<sup>st</sup> Analysis)



- Assume  $V_Q = V_{DD}$  and  $\overline{V}_Q = 0V$
- Data must be forced into the cell
  - $V_Q$  must fall below the threshold of the inverter to turn M2 off.
  - This allows  $\overline{V}_Q$  to go high enough to go above the  $V_t$  of M1
    - This discharges node Q and stores a 0
- Assume  $V_{BL}$  remains at 0V: M3 linear, M5 linear ( $V_Q = V_{DD}/2$ )

# SRAM Design: Write "0" (1<sup>st</sup> Analysis)

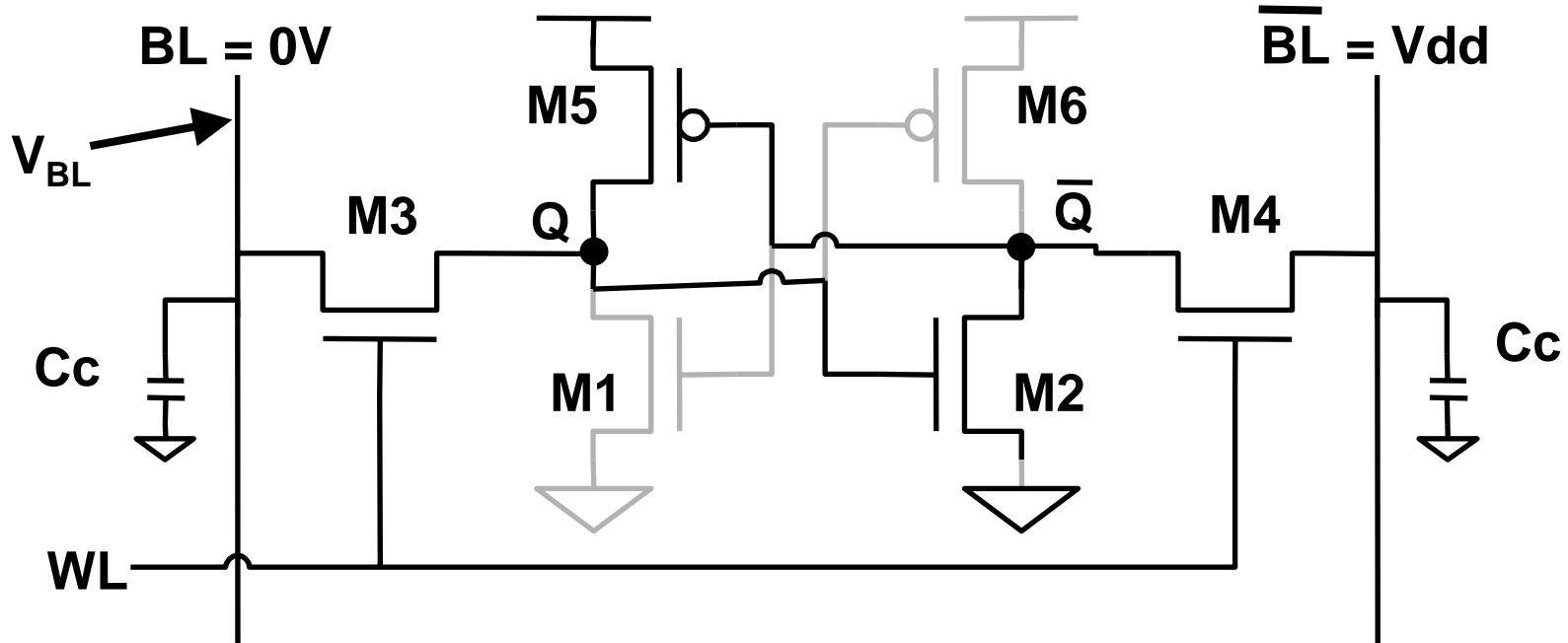


- **Conditions for this to happen: requires M5 to M3 ratio to be relatively small ( $V_{DS} = V_Q = V_{DD}/2$ )**

$$\frac{k_{p,5}}{2} \left( (V_{DD} - |V_{TP}|) V_{DD} - \frac{V_{DD}^2}{4} \right) = \frac{k_{n,3}}{2} \left( (V_{DD} - V_{TN}) V_{DD} - \frac{V_{DD}^2}{4} \right)$$



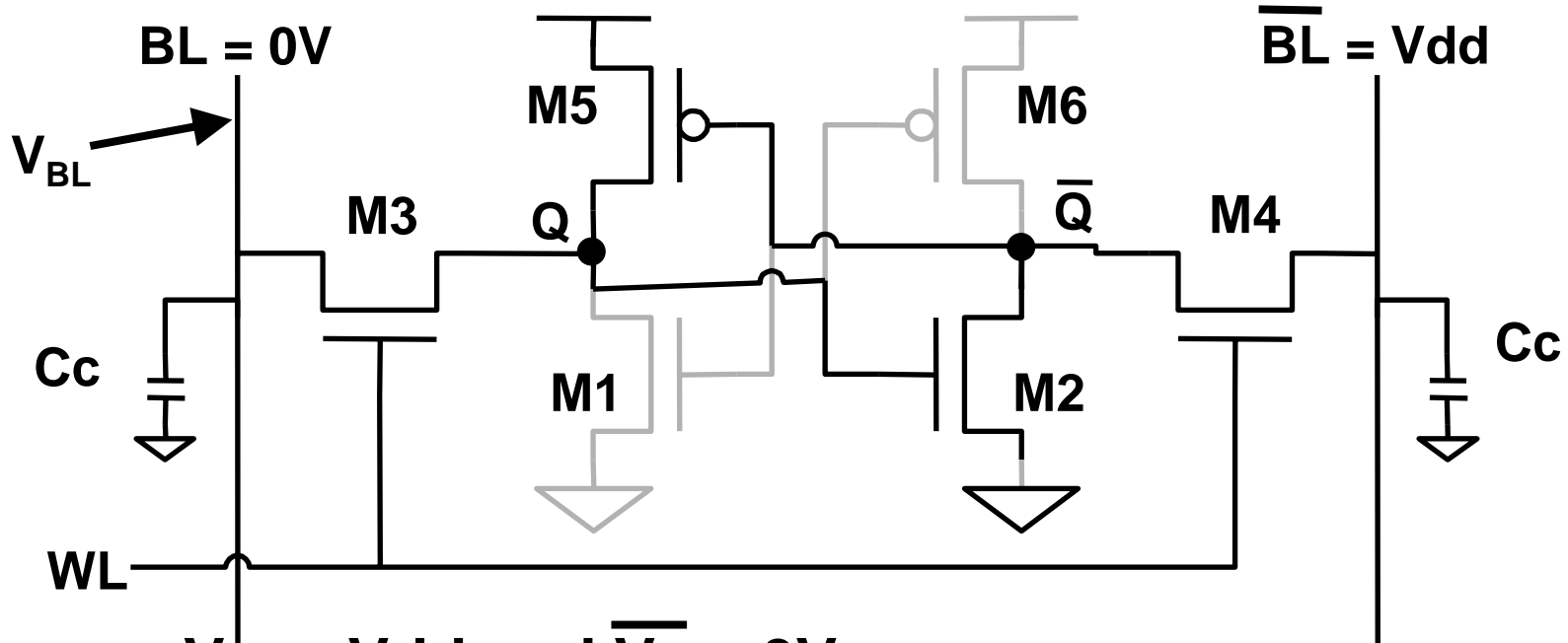
# SRAM Design: Write "0" (1<sup>st</sup> Analysis)



- Required sizing (size M5 below mobility ratio):

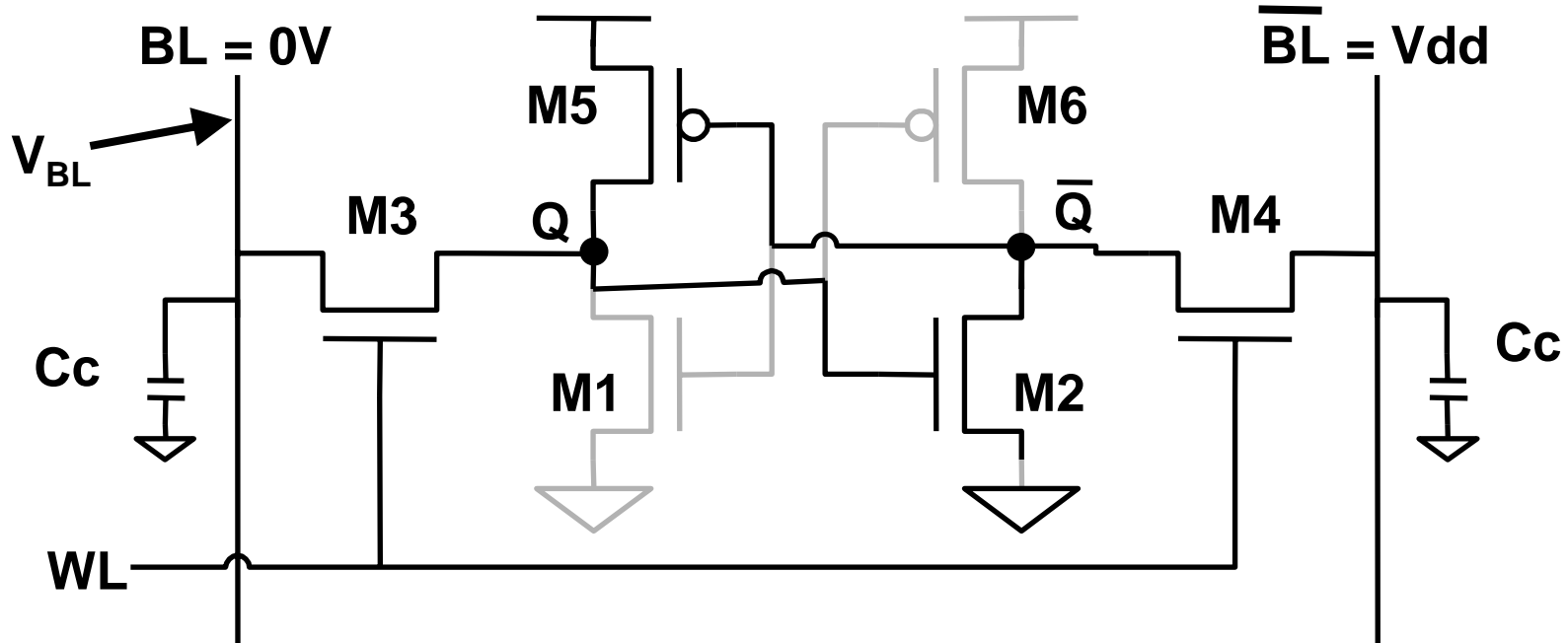
$$\frac{\left(\frac{W}{L}\right)_5}{\left(\frac{W}{L}\right)_3} < \frac{k'_n \left( (V_{DD} - V_{TN})V_{DD} - \frac{V_{DD}^2}{4} \right)}{k'_p \left( (V_{DD} - |V_{TP}|)V_{DD} - \frac{V_{DD}^2}{4} \right)} = \frac{k'_n}{k'_p} \quad \text{If } V_{TN} = |V_{TP}|$$

# SRAM Design: Write "0" (2<sup>nd</sup> Analysis)



- Assume  $V_Q = V_{dd}$  and  $\overline{V}_Q = 0V$
- Data must be forced into the cell
  - $V_Q$  must fall below the threshold of the NMOS (turns M2 off).
  - This allows  $\overline{V}_Q$  to go high enough to go above the  $V_t$  of M1
    - This discharges node Q and stores a 0
- Assume  $V_{BL}$  remains at 0V: M5 sat., M3 linear ( $V_Q = V_{TN}$ )

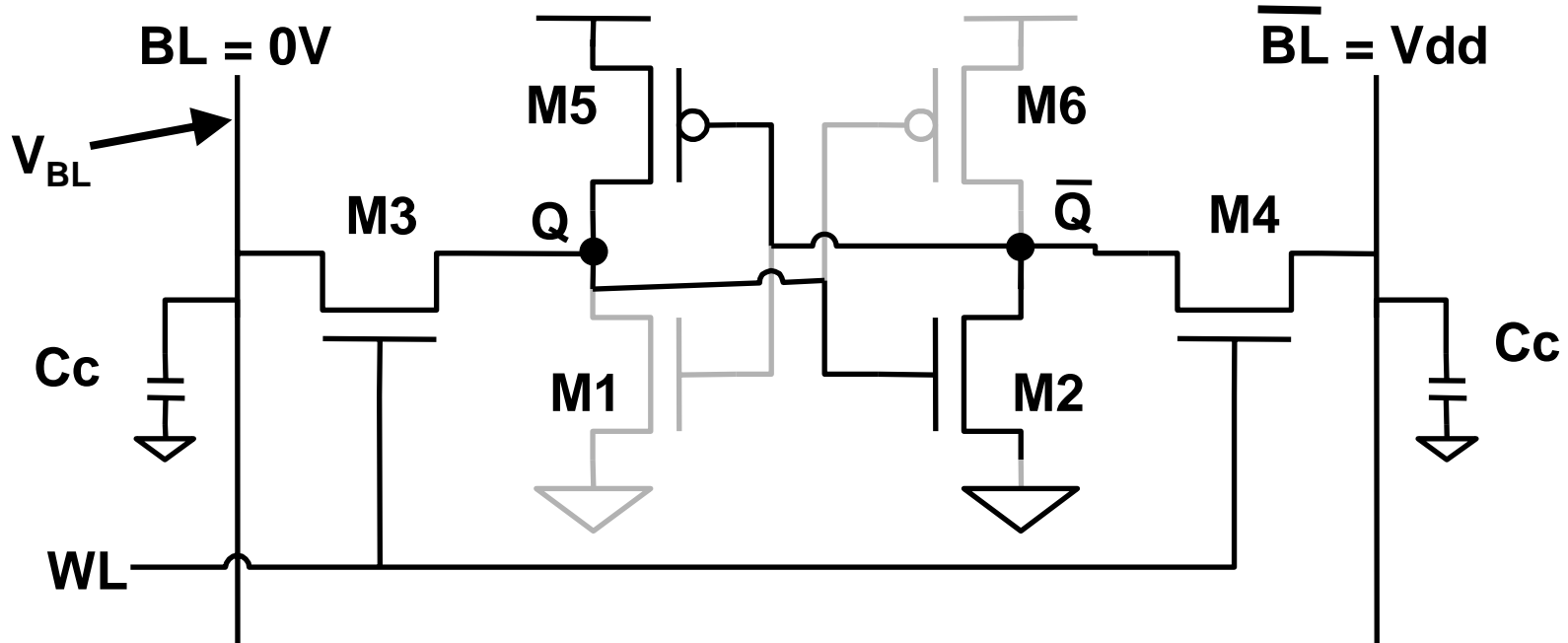
# SRAM Design: Write "0" (2<sup>nd</sup> Analysis)



- **Desired current conditions, want  $V_Q < V_{TN}$  (M3 lin., M5 sat.):**

$$\frac{k_{n,3}}{2} \left( 2(V_{DD} - V_{TN})V_{TN} - V_{TN}^2 \right) = \frac{k_{p,5}}{2} \left( 0 - V_{DD} - V_{TP} \right)^2$$

# SRAM Design: Write "0" (2<sup>nd</sup> Analysis)



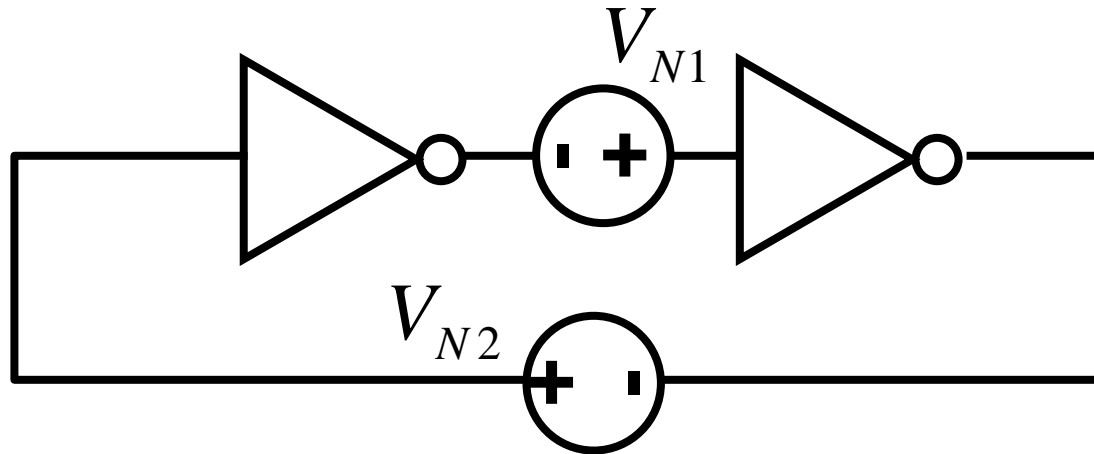
- Required sizing (make M5 relatively weaker than 1<sup>st</sup> case):

$$\frac{\left(\frac{W}{L}\right)_5}{\left(\frac{W}{L}\right)_3} < \frac{k'_n}{k'_p} \frac{2(V_{DD} - 1.5V_{TN})V_{TN}}{(V_{DD} + V_{TP})^2}$$

# SRAM Static Noise Margins

---

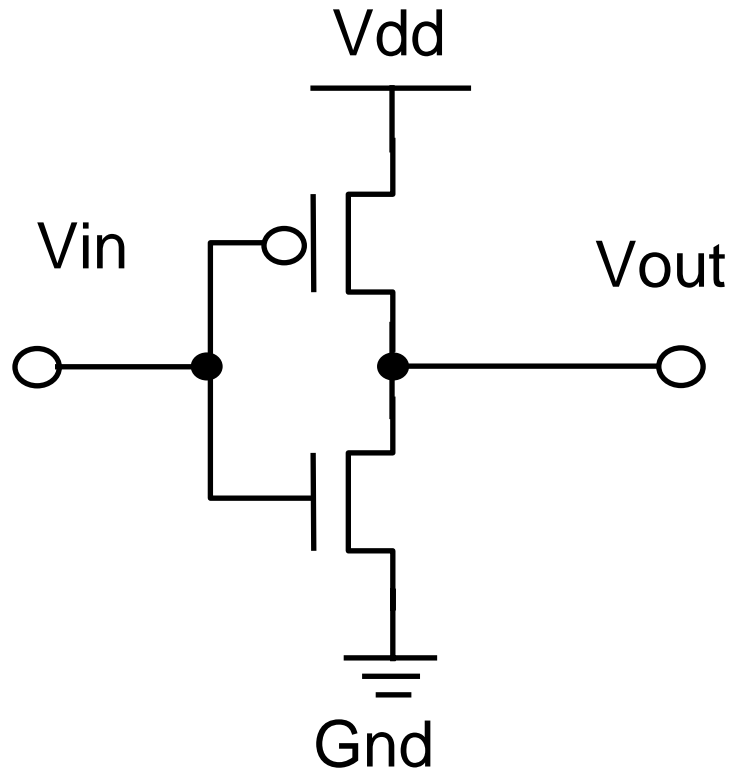
- **SRAM cell stability and writability quantitatively specified by static noise margin (SNM)**
  - Dependent on mode of operation (Hold, Read, or Write)



- **Hypothetical noise sources added to inverter inputs**
- **SNM corresponds to largest noise disturbances which won't disrupt cell operation**
- **SNM can be determined graphically by *butterfly plot***

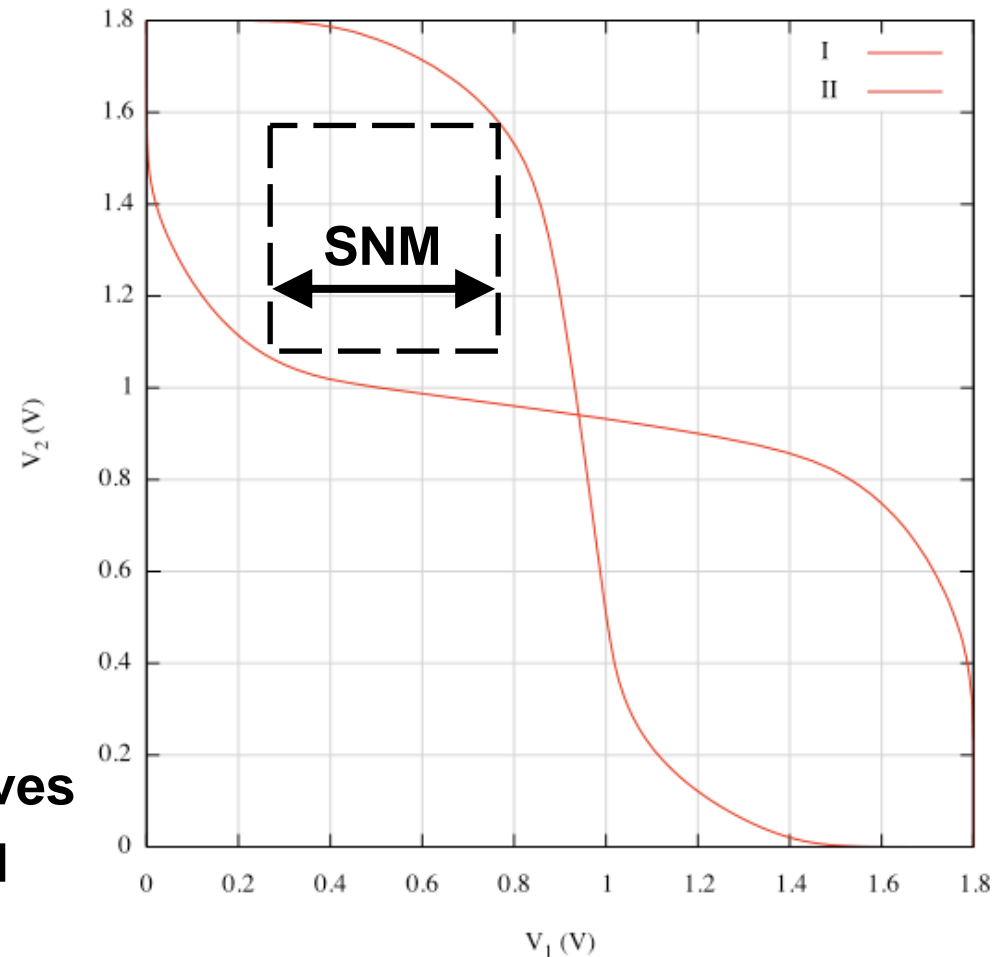
# Hold Static Noise Margin

## Hold SNM Half Circuit



## Hold SNM Butterfly Plot

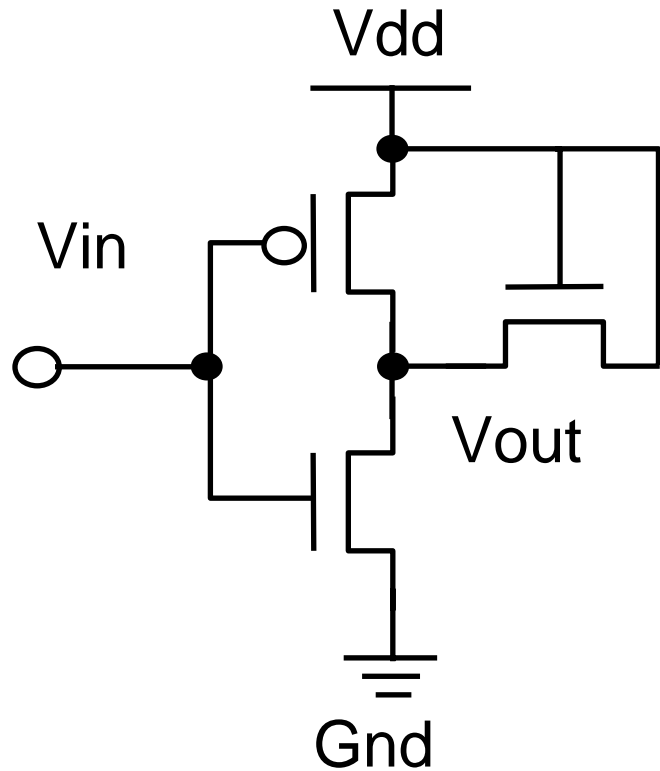
Hold Static Noise Margin



- Plot two mirrored  $V_{in}/V_{out}$  curves
- SNM = side of largest inscribed square

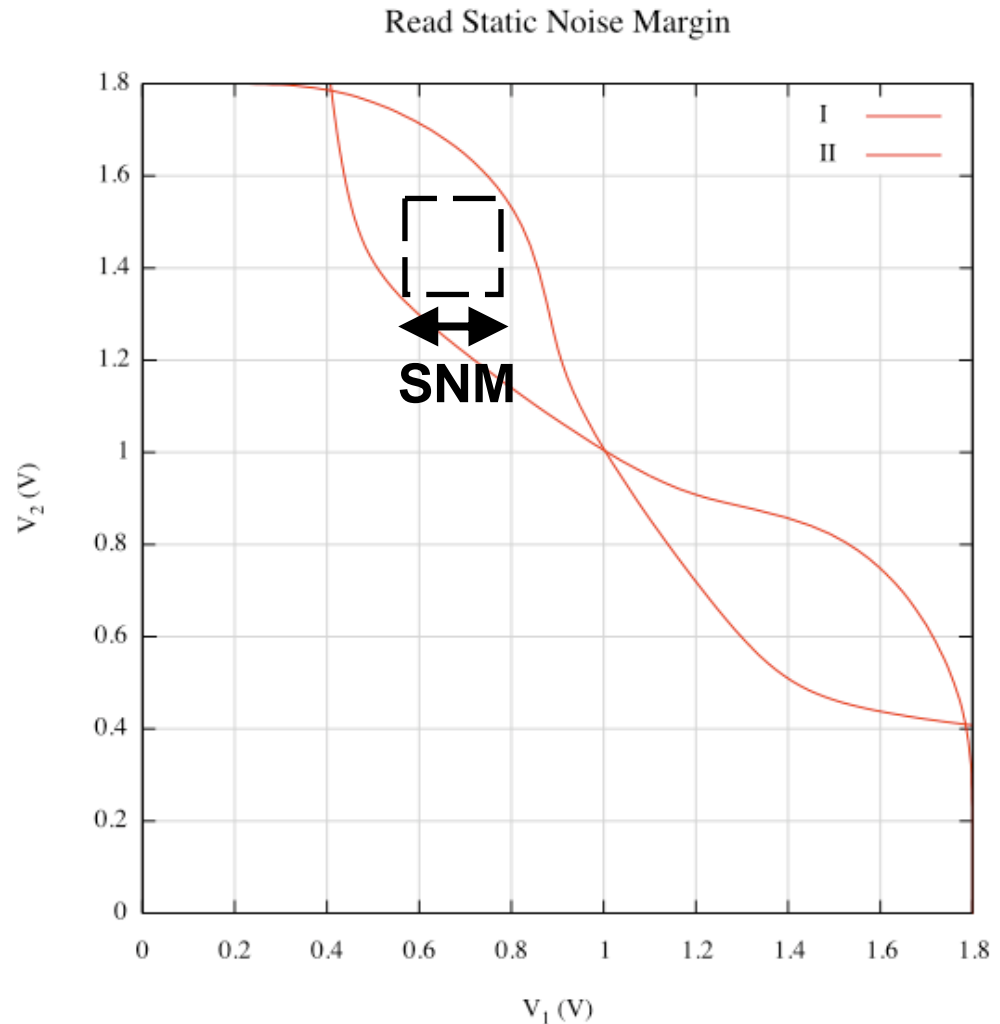
# Read Static Noise Margin

## Read SNM Half Circuit



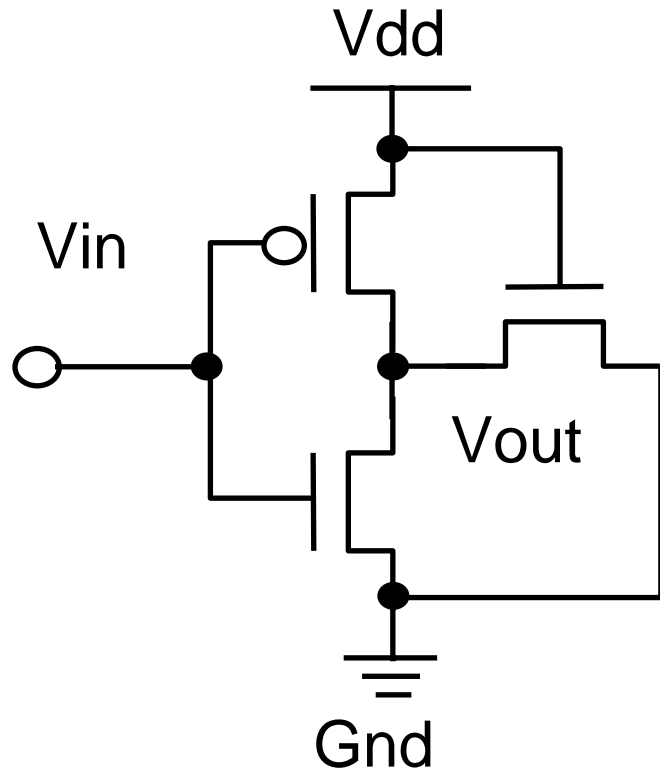
- **Diode-connected access NMOS simulates precharged bitline**

## Read SNM Butterfly Plot



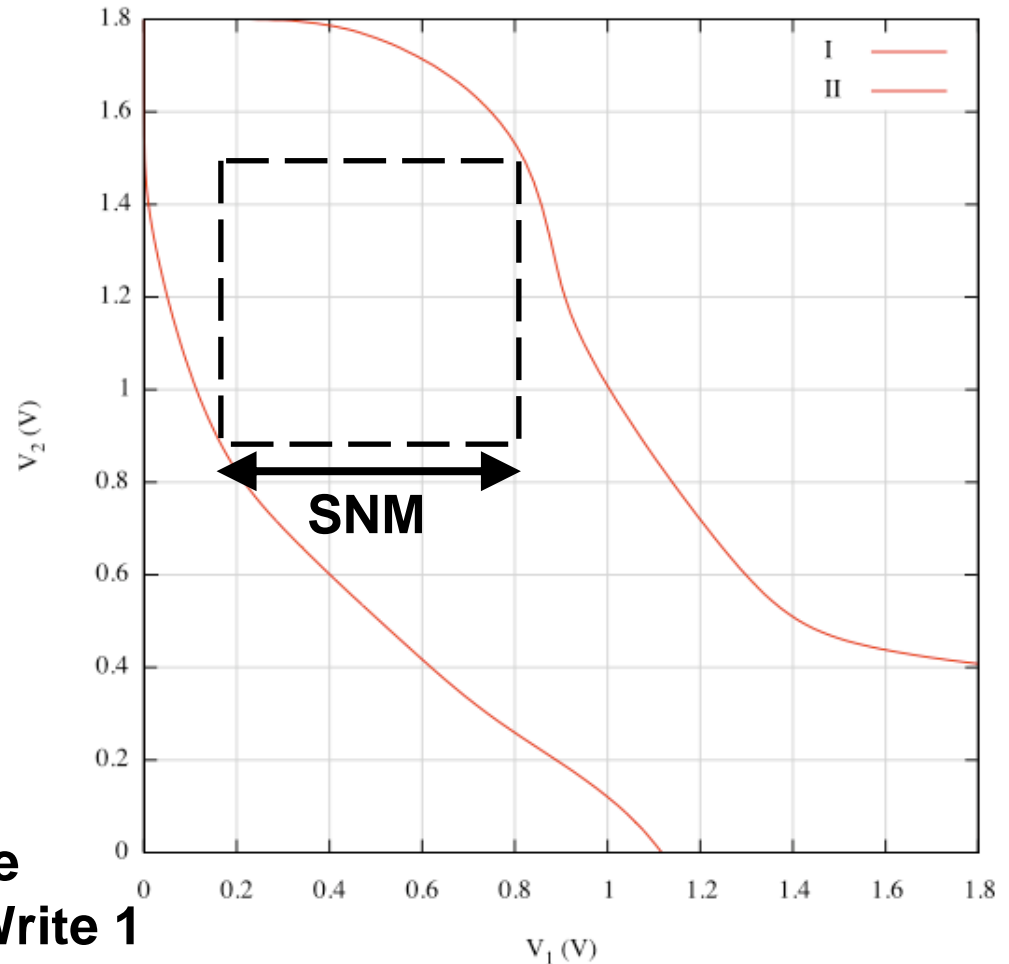
# Write Static Noise Margin

## Write 0 SNM Half Circuit



## Write SNM Butterfly Plot

Write Static Noise Margin



- Always-on ground-connected access NMOS simulates bitline driven low, use Read Ckt for Write 1



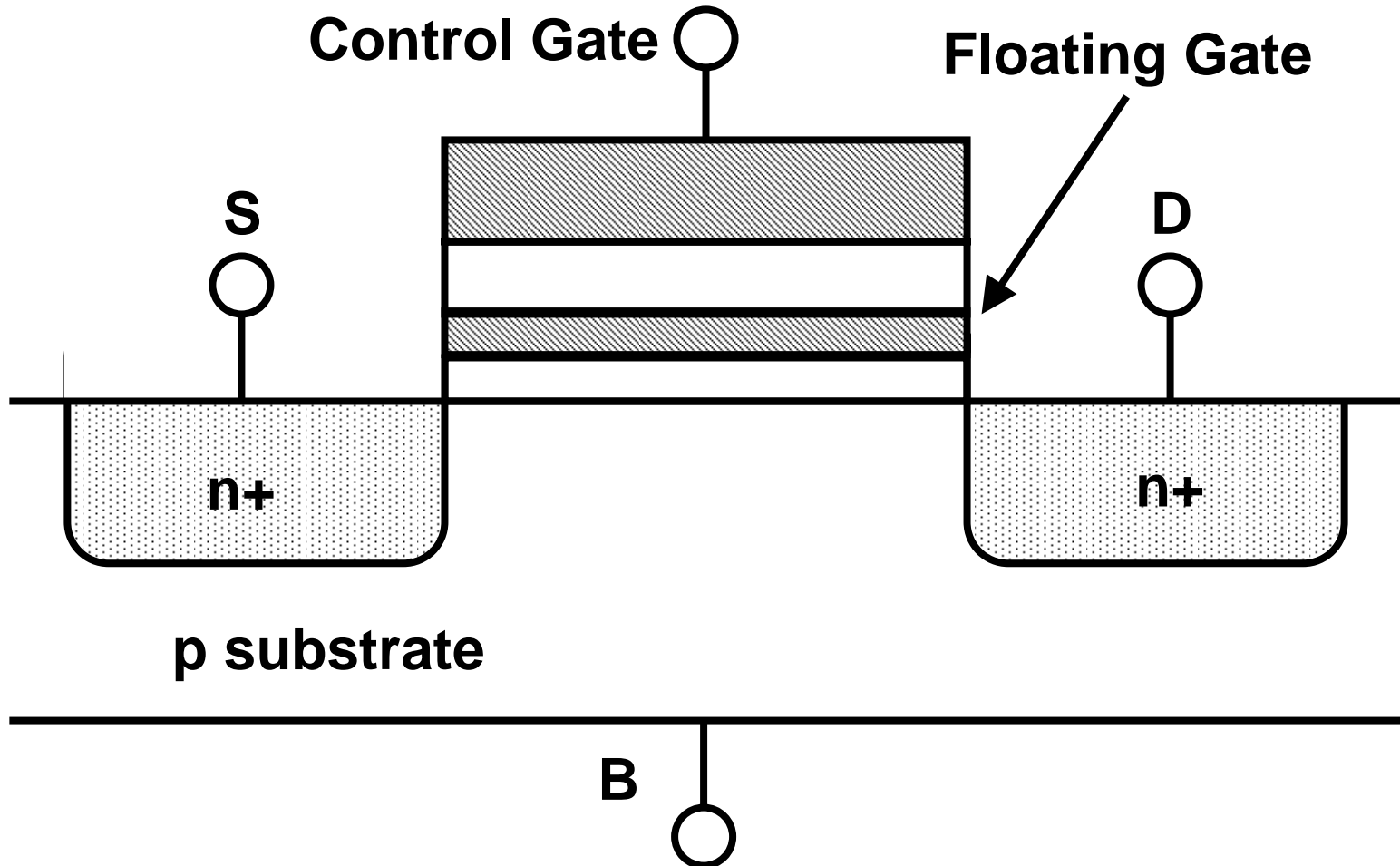
# Memory Peripherals

---

- **Memory core (memory cells) largely determined by technological considerations**
  - Emphasizes reduced area, sacrifices speed, reliability
  - Peripheral circuits can recover some of the lost performance
- **Address Decoders**
  - Row Decoders: one-hot decoding for word lines
  - Column Decoders:  $2^L$ -to-1 multiplexers for bit lines
- **I/O Buffers and Drivers**
- **Sense Amplifiers**
- **Memory Timing and Control**

# Flash Memory Transistor With Floating Gate

---



- **Threshold voltage of device adjusted by placing charge on floating gate**

# Flash Memory Operation

---

- **Two threshold voltages correspond to two states (1 bit)**
  - Bitline is precharged high before a read
  - Low  $V_T$  state, when wordline (control gate) is high the bitline is discharged and a “0” is read
  - High  $V_T$  state, the wordline (control gate) can’t go high enough to turn on transistor, bitline stays high and a “1” is read
- **Writing a “1”:** electrons are accelerated by a high field until they accumulate on the floating gate, raising  $V_T$
- **Writing a “0”:** electrons driven off floating gate by a reverse gate-source bias through Fowler-Nordheim tunneling, lowering  $V_T$

# **Next Topics: Low Power Circuits and Wires**

---

- **Low power design principles and circuit techniques**
  - Voltage scaling, activity factor reduction, clock gating, leakage reduction
- **On-chip resistance and capacitance**
  - Delay estimation
  - Buffering and repeater insertion