University of Southern California

Viterbi School of Engineering

EE577A

VLSI System Design

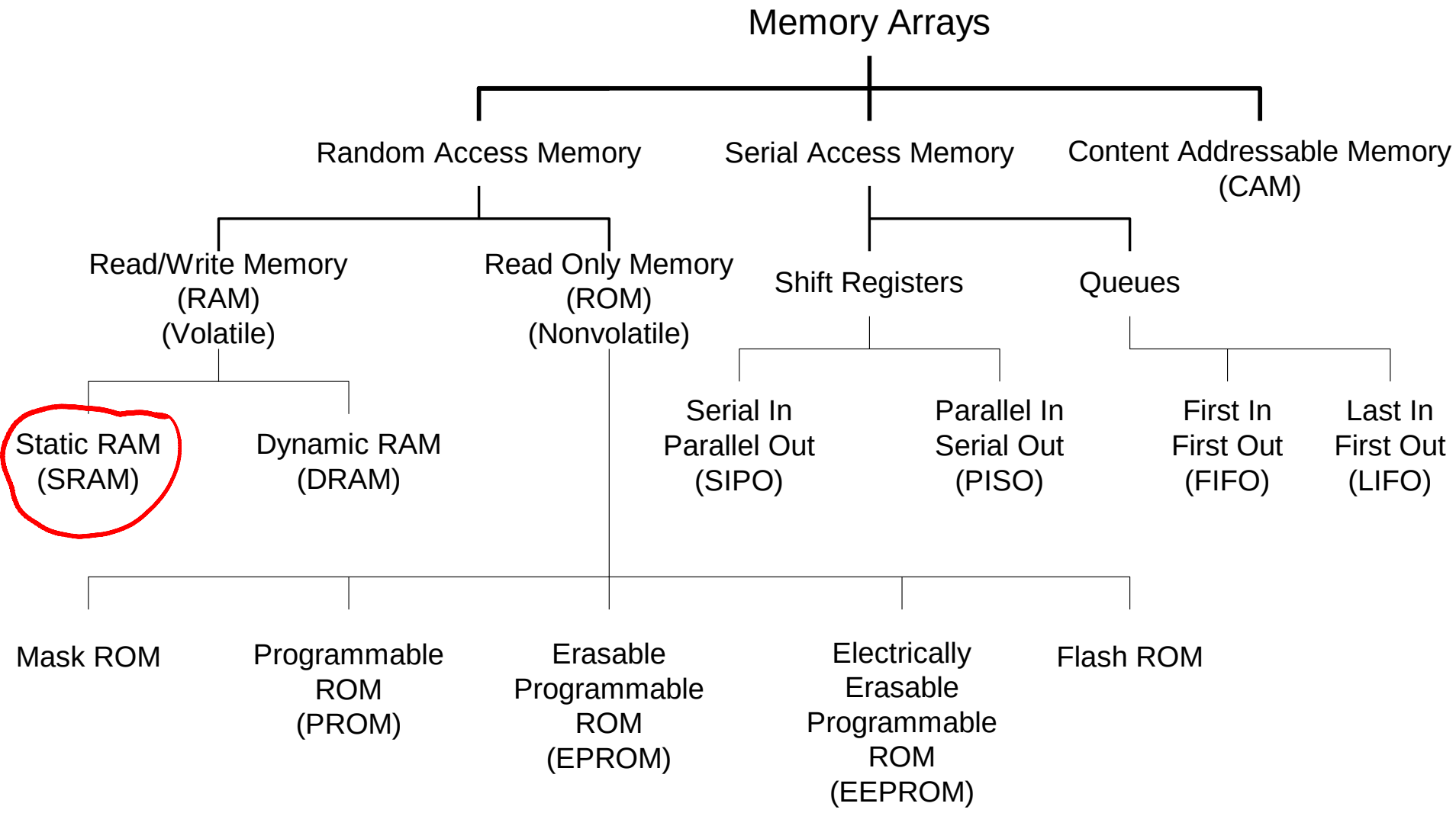
Memory Design

References: syllabus textbooks, Slides and notes from Professor Pedram, online resources

Shahin Nazarian

Spring 2013

Digital Memories Types



Review: Random Access Technology

- As the technology evolves randomness plays more important role and we want data to be randomly accessible
- We also want the access time to not be a function of the location of the memory data
- Memory can be classified into Random Access Memory (RAM) and non-RAM memories
- Random Access Memories can be further classified into ROMs and Read/Write (R/W) memories
- In RAM technology access time is the same regardless of the location of the memory data
- R/W memory is also commonly called RAM due to historical reasons
- R/W (or RAMs) have two main types of Dynamic RAMs (DRAMs) and Static RAMs (SRAMs)

Review: Not-so-Random Access Technology

- Random Access:
  - DRAM: Dynamic Random Access Memory
    - High density, low power, cheap, slow
    - Dynamic: need to be "refreshed" regularly
  - SRAM: Static Random Access Memory
    - Low density, high power, expensive, fast
    - Static: content will last "forever"(until lose power)
- "Not-so-random" Access Technology:
  - Access time varies from location to location and from time to time
  - It's random accessible, but it's not the same time
  - Examples: Disk, CDROM

Volatile vs Nonvolatile

- Volatile memory: A memory that requires power to maintain the stored data, if we switch off its power supply, all the stored information will be lost. Hence it gets the name "volatile".
  - Examples: SRAMs, DRAMs are the commonly used volatile memory
  - Note: This can be considered as a drawback of DRAM and SRAM (that the stored data are lost in the absence of the power supply)

- Nonvolatile: A memory that can hold the data even when not powered is referred to as NVM

Nonvolatile Memory

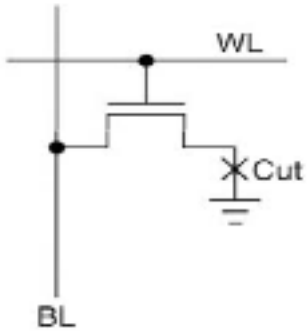
- To overcome the drawback of volatile SRAM and DRAM memories, various kinds of nonvolatile memories (nonprogrammable such as mask ROM and programmable such flash) have been developed
- Storing binary information at a particular address location can be achieved by the presence or absence of a data path from the selected row
- Recently flash memory based on the floating-gate concept has become the most popular nonvolatile memory due to its small cell size and better functionality

Nonvolatile Memory (Cont.)

- Example
  - Different types of ROMs such as Flash memories, EPROM (Electrically Programmable ROM), EEPROM (Electrically Erasable and Programmable ROM)
  - Magnetic memories such magnetic tapes and hard disks, optical discs,
  - OTP (One-Time Programmable (diodes/fuses)
  - … and even punch cards!

Review: ROM

- ROM allows only retrieval of previously stored data. No modification is permitted. ROMs are nonvolatile memories, i.e., the stored data is not lost even when the power supply is off and refresh operation is not required
- ROM is classified to Mask ROM and PROM
  - In Mask (Fuse) ROM, data is written during chip manufacturing by using a photo mask



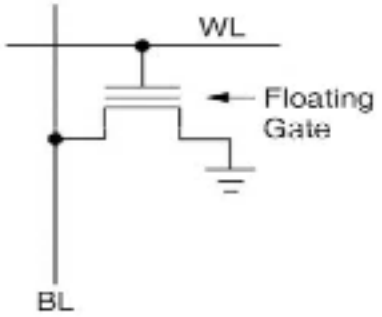
Mask (Fuse) ROM

  - In PROM the data is written electronically after the chip is fabricated

Review: ROM (Cont.)

- PROM is classified to EPROM, and EEPROM
  - Data written by blowing the fuse electrically cannot be erased and modified in Fuse ROM
  - Data in EPROM and EEPROM can be rewritten, but the number of subsequent re-writes is limited to 10^4-10^5
- In EPROM: ultraviolet rays (that can penetrate through the crystal glass on the package) are used to erase whole data in chip simultaneously. Programming is done by higher than normal voltages
- In EEPROM higher than normal electrical voltage is used to program/erase data in 8 bit units
- EEPROM drawback: slower write speed,

  in order of microseconds



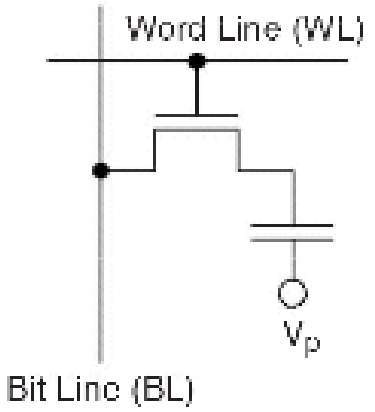
Review: ROM (Cont.)

- ROMs are generally used for permanent (look-up) memory in printers, fax, game machines, and ID cards, due to lower cost than RAM
- Ferroelectric RAM (FRAM) utilizes the hysteresis characteristics of a ferroelectric capacitor to overcome the slow write operation of other EEPROMs
- Flash ROM is similar to EEPROM and EPROM in using an array of floating gates (also referred to as cells). A single-level cell can store one bit of information, whereas a multi-level cells can store more than on bit of info by varying the number of electrons placed on the floating gate of the cell. Similarly to EEPROM, higher than normal voltages are used to program/erase the cells

DRAM

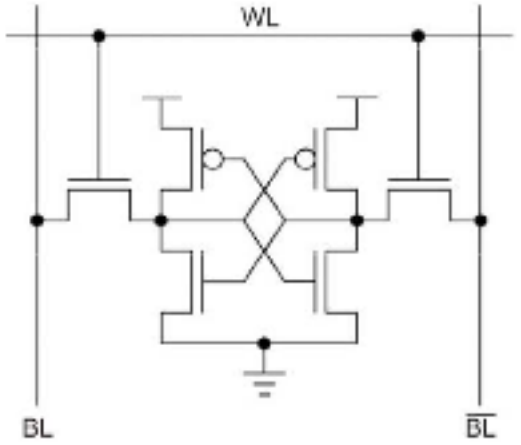
- DRAM has high density compared to SRAM, however DRAM cell info is degraded due to junction leakage current at the storage node, so cell data must be read and rewritten periodically (refresh operation). Due to low cost and high density, DRAM is widely used for the main memory in personal and mainframe computers and workstations
- Example: 1T (one-transistor) DRAM cell consists of a capacitor to store binary 1 (high voltage) or 0 (low voltage) and a transistor to access the capacitor

1T DRAM



SRAM

- SRAM consists of a latch, so the cell data is kept as long as power is turned on and refresh operation is not required
- SRAM is mainly used for the cache memory in microprocessors, main frame computers, engineering workstations and memory in hand-held devices due to high speed and low power consumption
- Example: 6T SRAM

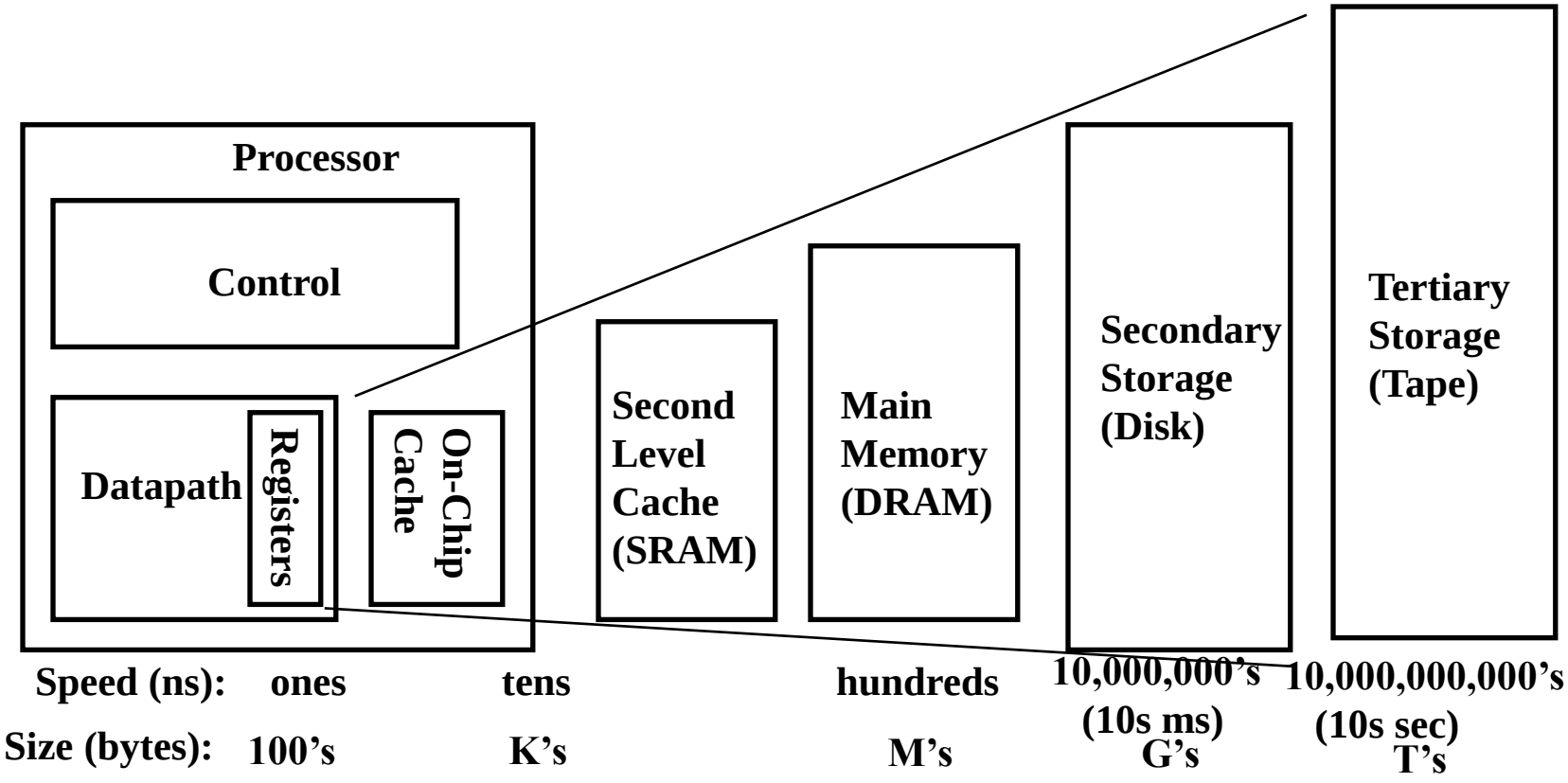


6T SRAM

Memory Design Goals

- The goal is to design memories that are larger, denser (more bits per area), faster (faster write and read operation), more reliable, consume less power, and have less design complexity
  - However some of these goals are contradictory, so compromises have to be made
  - Paradoxes of memory design
    - Denser and faster
    - Larger capacity and low power
    - Reduced complexity and high reliability

Memory Hierarchy of a Modern Computer System



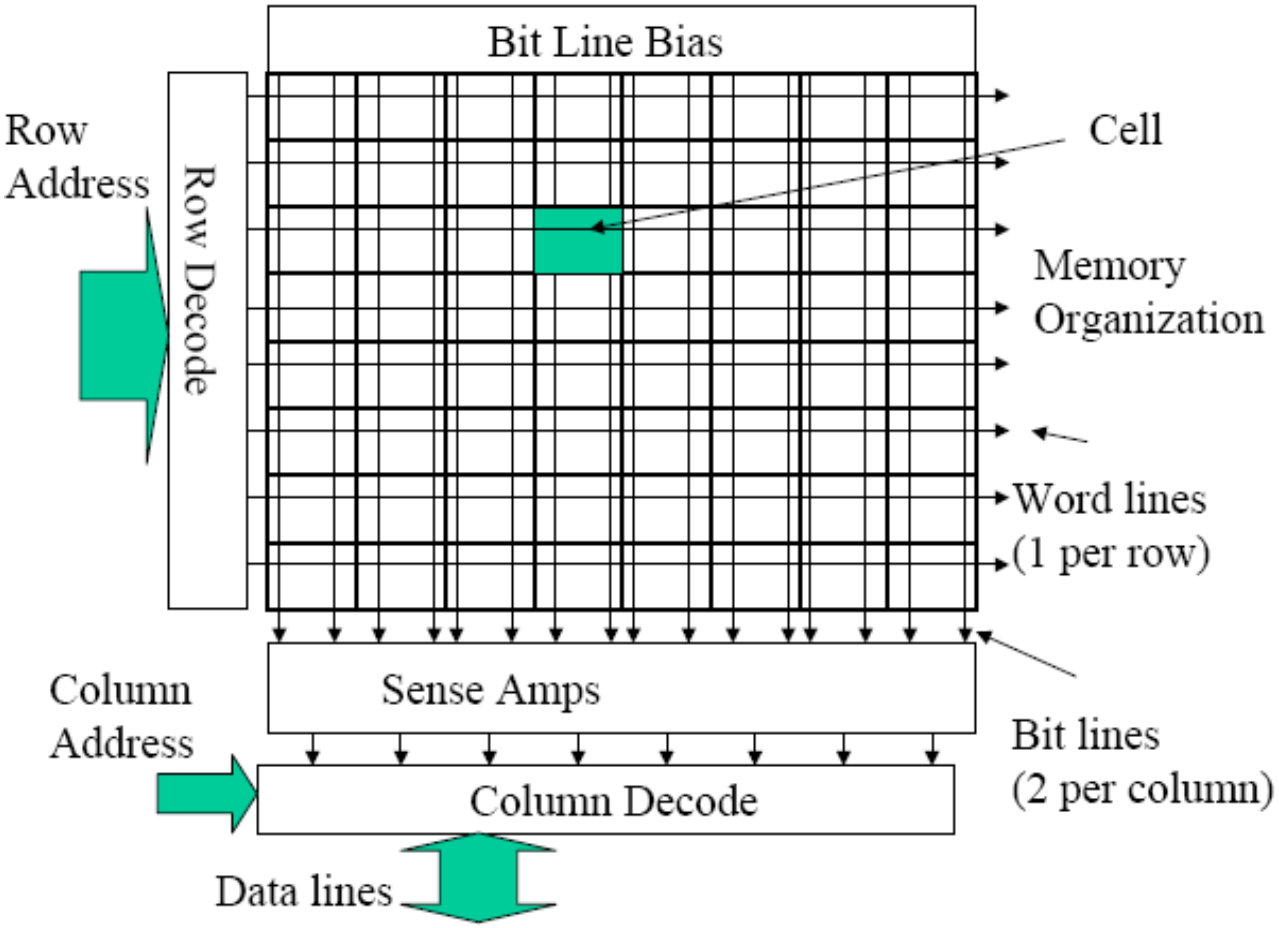
Static Read-Write Memory (SRAM)

- Historical trend
- Structural trend
- Future needs

SRAM vs. DRAM Summary

- SRAM
  - Faster because bit lines are actively driven by the D-Latch
  - Faster, simpler interface due to lack of refresh
  - Larger area for each cell which means less memory per chip
  - Used for cache memories (and also register files) memory where speed/latency is key
- DRAM
  - Slower because passive value (charge on cap.) drives bl
  - Slower due to refresh cycles
  - Small area means much greater density of cells and thus large memories
  - Used for main memories where density is key

Typical SRAM Array



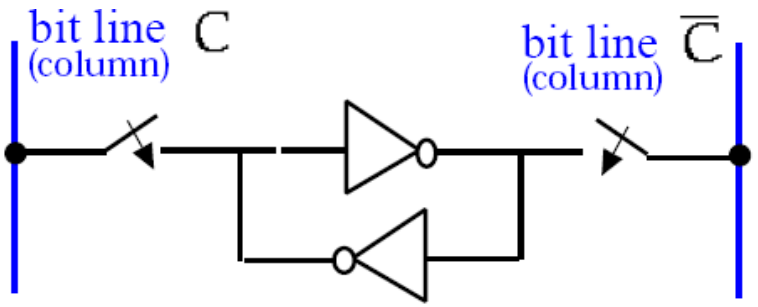
Some Issues in Determining the Memory Array Organization

- Typically we want an aspect ratio that is nearly one
- How to divide up the row, column address decoding?
  - Consider an 8K x 32 SRAM = 256 Kb = 2^{18} with 2^9 rows x 2^9 columns as an example
    - Row decoder is 9 to 512 decoder. Every 32 (2^5) columns is a 'word', and we only need to decode words. So, column decoder needs to decode 16 words, that is, we only need a 4 to 16 column decoder

Some Issues in Determining the Memory Array Organization (Cont.)

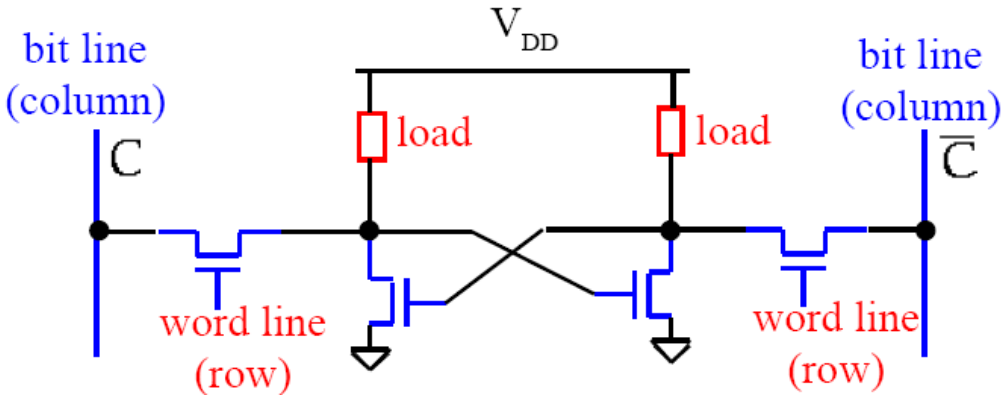
- Assertion of word line accesses all cells in a row
  - Not all bits that are read from a row may be used
  - Loading on word line is high!
- Bit lines connect all cells in a column, only one cell in a column can ever be ON at a time
- Would like to keep the bitline swing low to preserve power

SRAM Cell

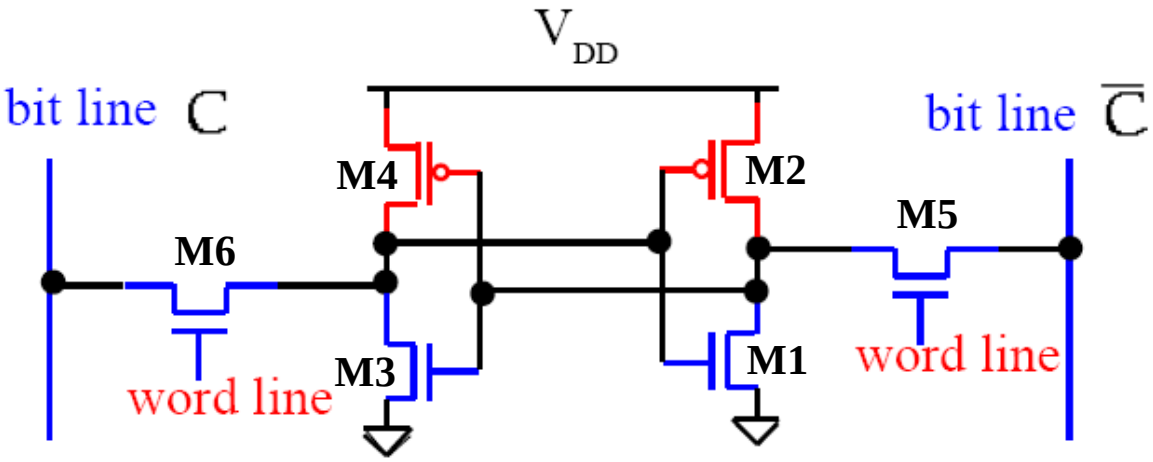


1 - BIT SRAM CELL

Complementary Column arrangement achieves more reliable operation



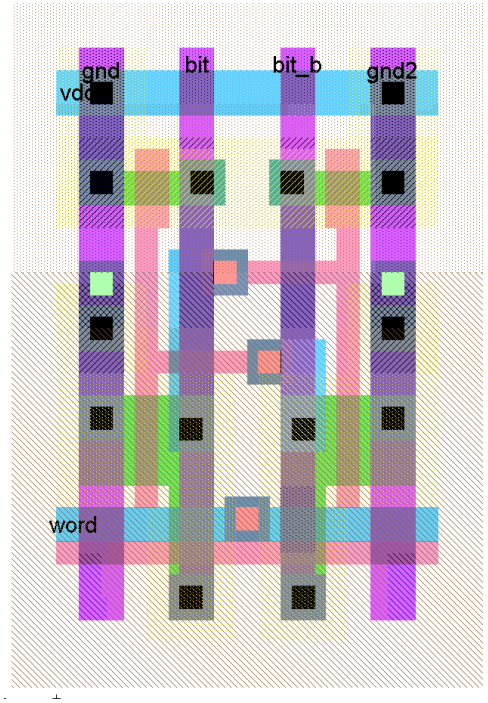
Full CMOS (6-T) SRAM Cell

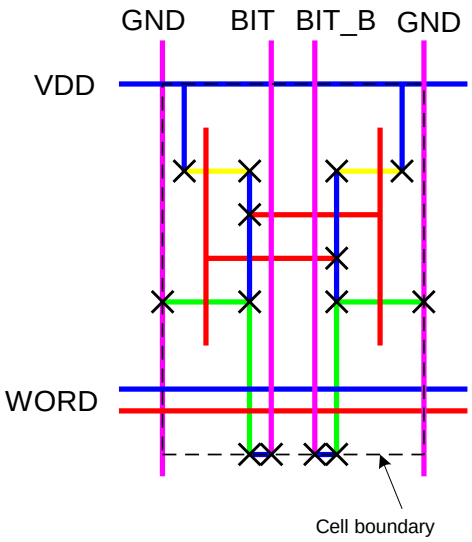


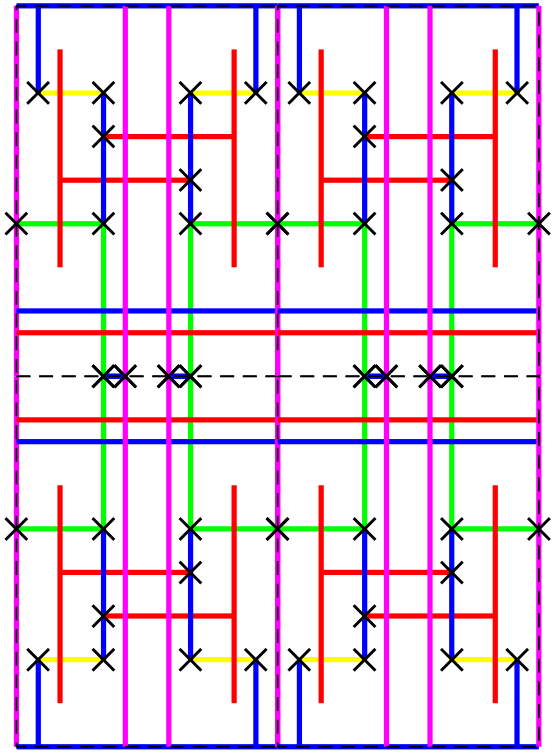
- Very low standby power consumption, large noise margin, low supply voltage
- Basic requirements for setting the (W/L) ratios:
  - Data-write operation is capable of modifying stored data in SRAM cell
  - Data-read operation does not modify stored data

SRAM Layout

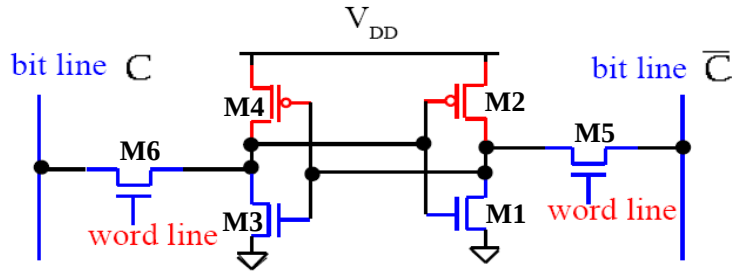
- Cell size is critical: 26 x 45 λ (even smaller in industry)
- Tile cells sharing V_{DD}, GND, bitline contacts

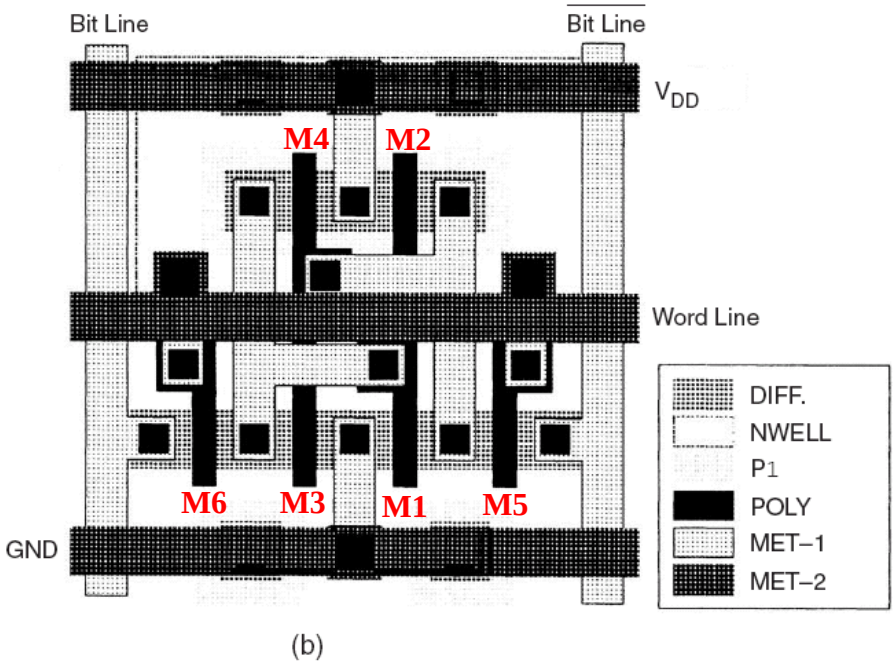




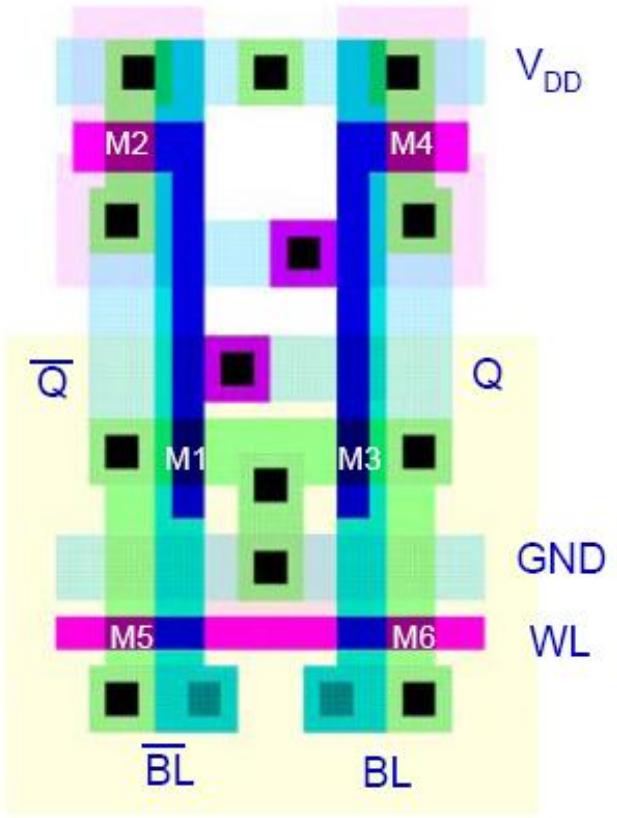


Layout of the CMOS SRAM Cell





6T SRAM cell layout



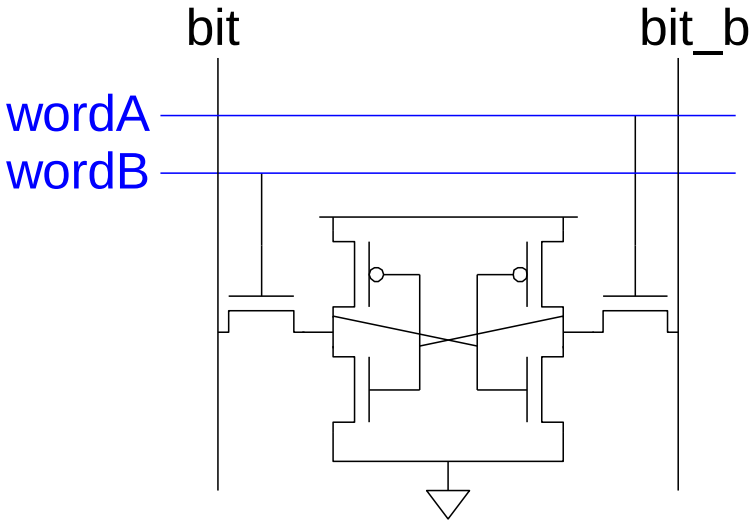
A different layout

Multiple Ports

- We have considered single-ported SRAM
  - One read or one write on each cycle
- *Multiported* SRAM are needed for register files
- Examples:
  - Multicycle MIPS must read two sources or write a result on some cycles
  - Pipelined MIPS must read two sources and write a third result each cycle
  - Superscalar MIPS must read and write many sources and results each cycle

Dual-Ported SRAM

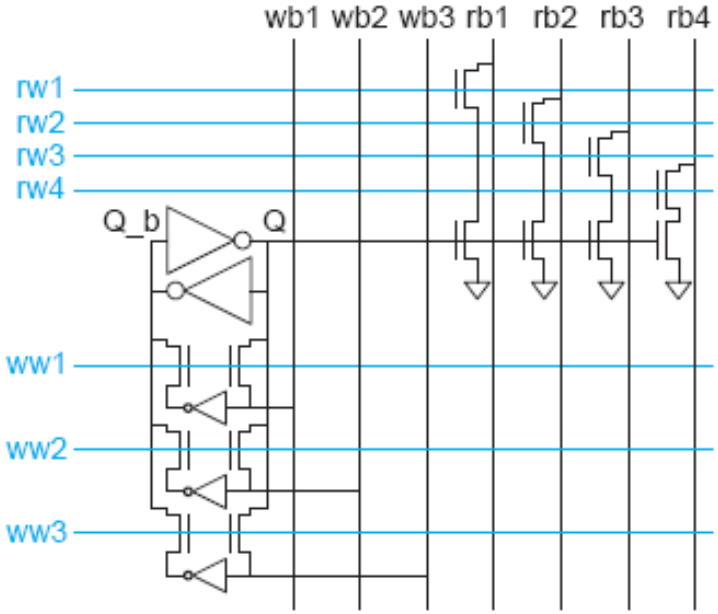
- Simple dual-ported SRAM
  - Two independent single-ended reads
  - Or one differential write



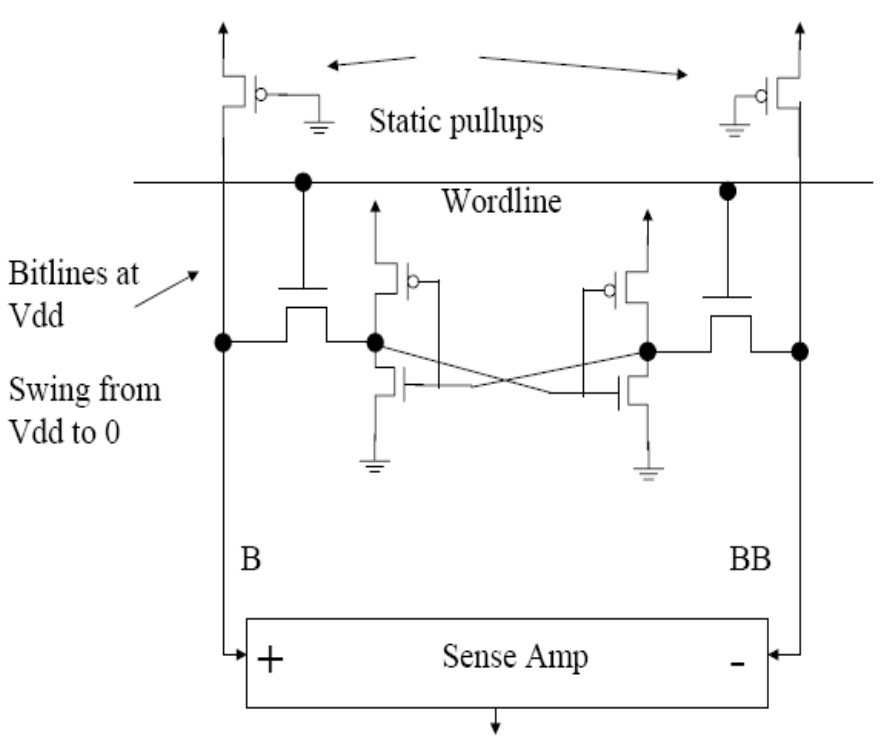
- Do two reads and one write by time multiplexing
  - Read during ph1, write during ph2

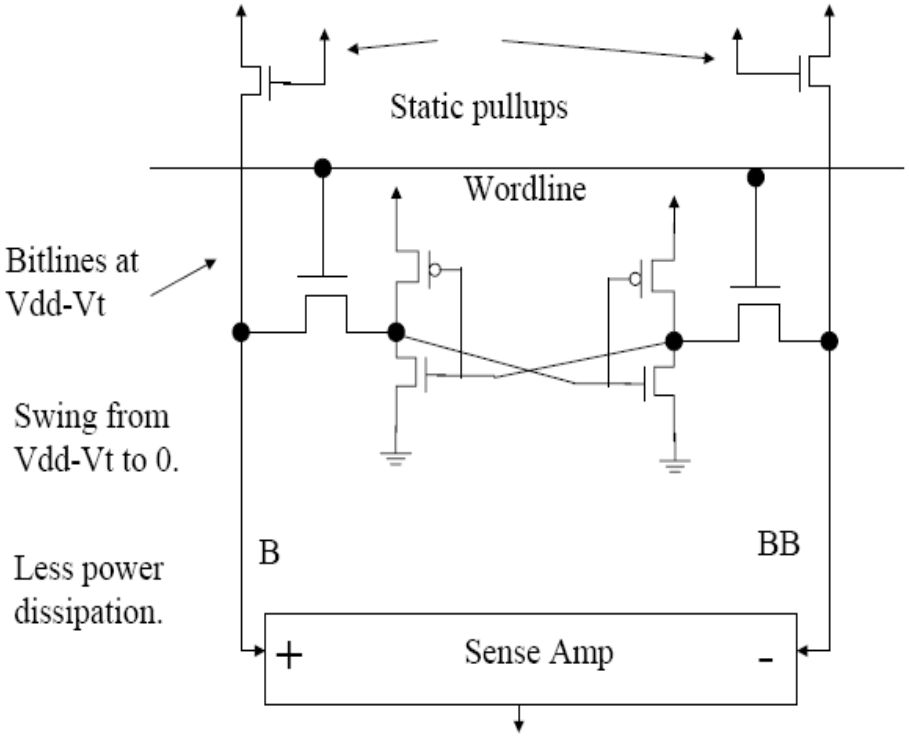
Multi-Ported SRAM

- Adding more access transistors hurts read stability
- Multiported SRAM isolates reads from state node
- Single-ended bitlines save area

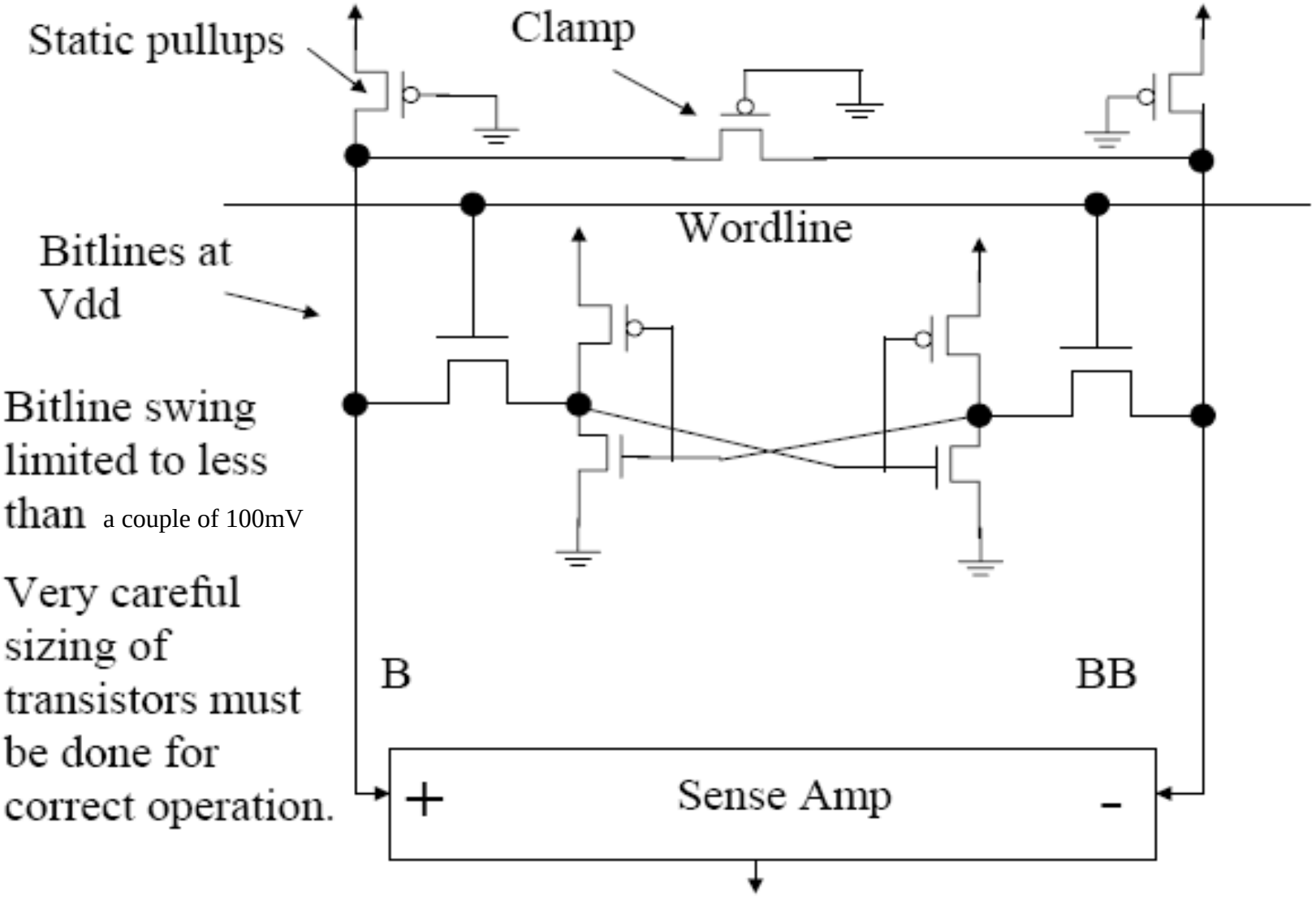


Static Bit Line Biasing

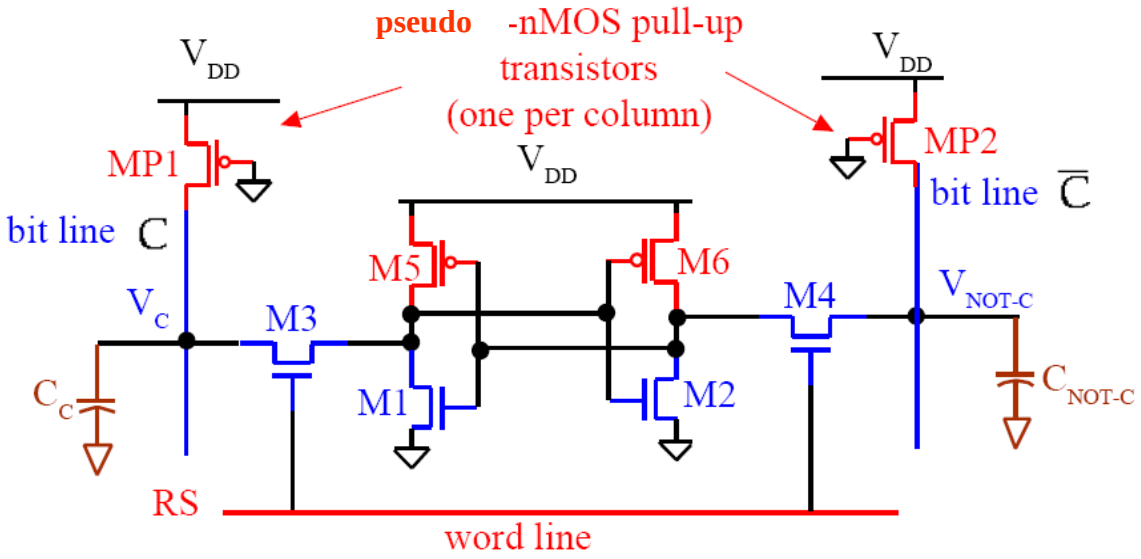




Static Bit Line Biasing with Clamps



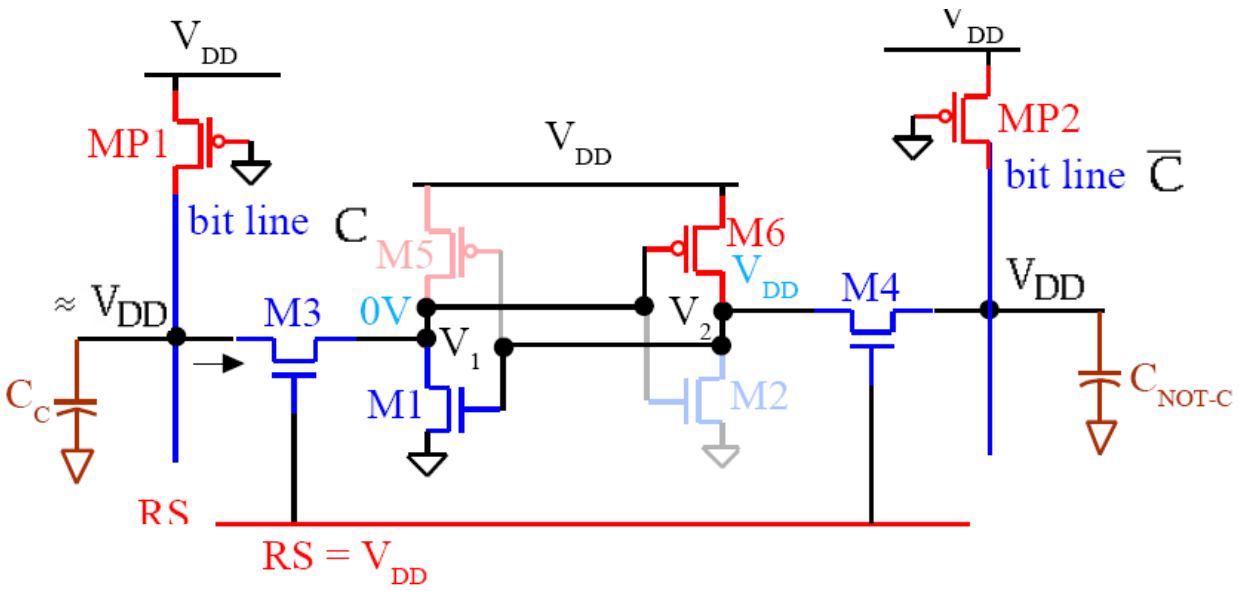
SRAM Cell w/ Static Bitline Pull-ups



- When the word line is not selected, RS=0. M3 and M4 are OFF
  - If RS = 0 for ALL rows, the bit lines capacitances C_c and C_{NOT-C} are charged-up to V_{DD} by pull-up of MP1 and MP2
  - Depending on application, MP1 and MP2 are turned OFF or are kept ON during the read operation

CMOS SRAM Cell Design Strategy

- **Consider data-read operation with "0" stored in cell**



a. @ t = 0⁻: M3, M4 OFF; M2, M5 OFF & M1, M6 LIN

b. @ t = 0: M3 SAT, M4 OFF; M2, M5 OFF & M1, M6 LIN

slow discharge of large C_c and V_1 increases

REQUIRE $V_1 < V_{T02}$ => LIMITS M3 W/L wrt M1 W/L

Data-Read Operation

- **Conservative design constraint: $V_{1,max} \leq V_{T,2}$ to keep M2 OFF during the read operation. M3 will be in saturation whereas M1 operates in the linear region:**

$$\frac{k_{n,3}}{2}\left(V_{DD}-V_1-V_{T,n}\right)^2 \leq \frac{k_{n,1}}{2}\left(2(V_{DD}-V_{T,n})V_1-V_1^2\right) \quad \text{at} \quad V_1 = V_{T,n}$$

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

A typical example:

With $V_{DD} = 2.5V, V_{T,n} = 0.4V$:

$$\frac{\left(\frac{W}{L}\right)_3}{\left(\frac{W}{L}\right)_1} \leq \frac{2(1.9)(0.4)}{(1.7)^2} \Rightarrow \left(\frac{W}{L}\right)_3 \leq 0.5\left(\frac{W}{L}\right)_1$$

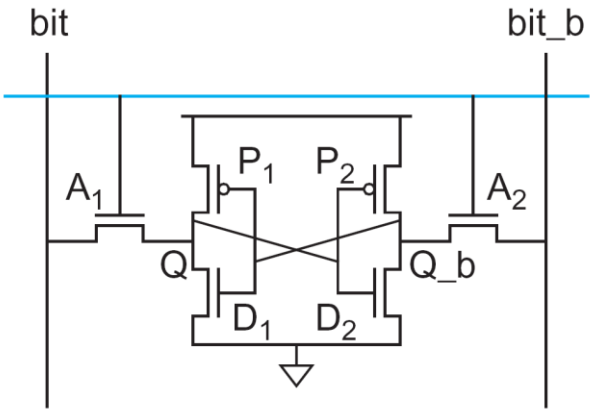
- **A symmetrical condition also dictates the aspect ratios of M2 and M4**

Read Operation (Cont.)

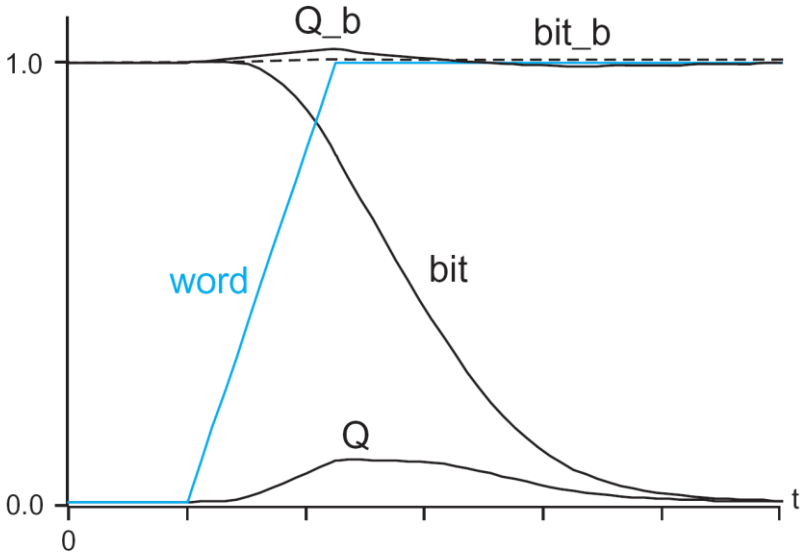
$V_1 = Q$

$V_2 = Q\_b$

RS ≡ word



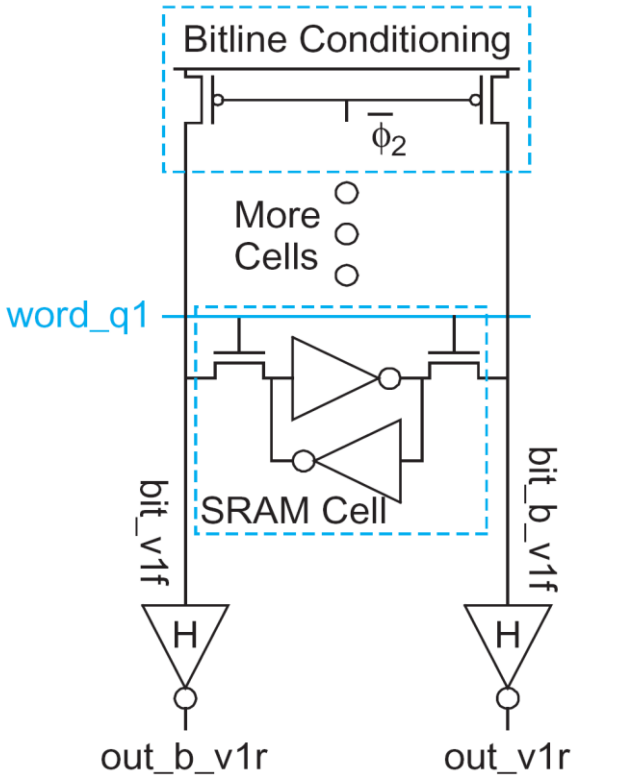
(a)

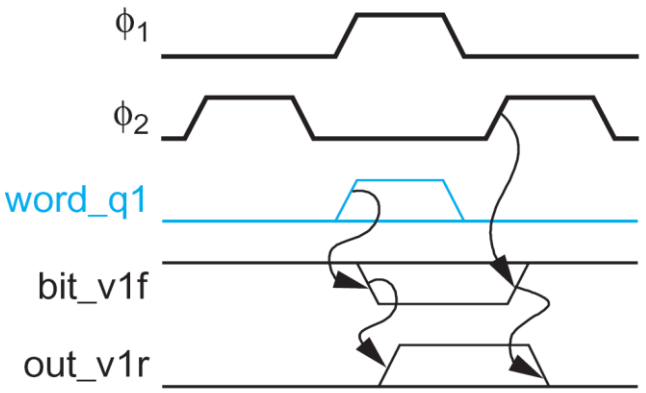


(b)

SRAM Column Read

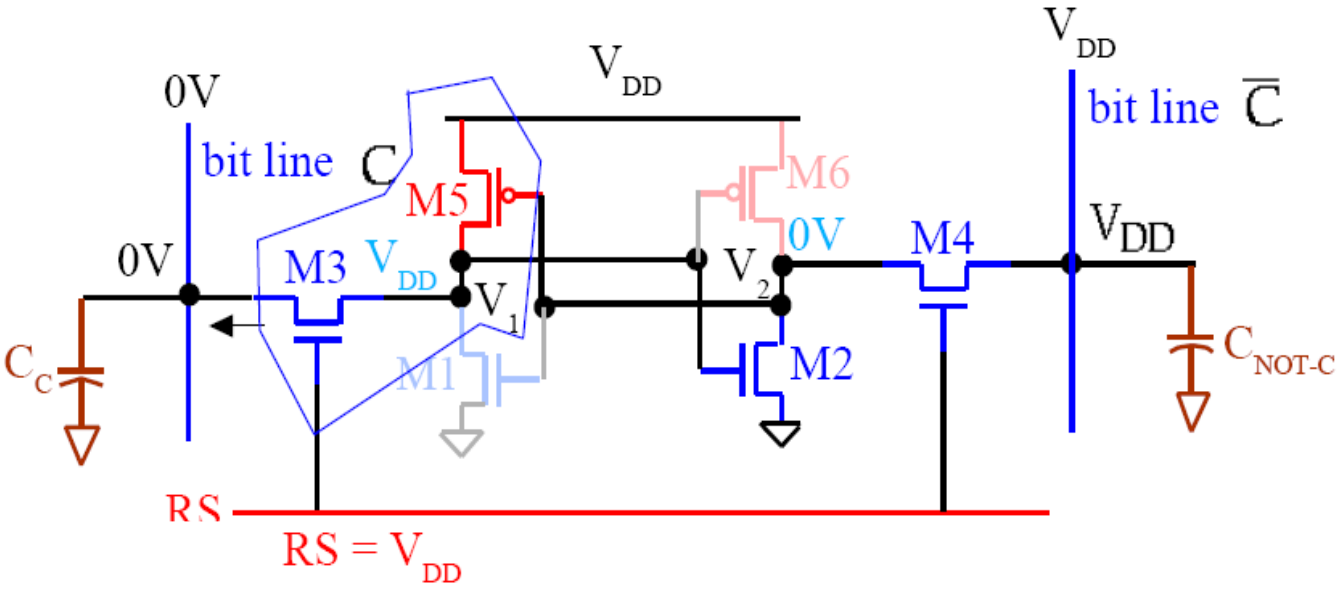
- Large signal sensing





Data-Write Operation

- **Consider the write "0" operation assuming a logic "1" is already stored in the SRAM cell**



V_C IS SET "0" BY DATA-WRITE CIRCUIT

a. @ t = 0⁻: M3, M4 OFF; M2, M5 LIN & M1, M6 OFF ("1" stored)

b. @ t = 0: M3 SAT, M4 SAT; M2, M5 LIN & M1, M6 OFF

WRITE "0" => V_1: V_{DD} → 0 ($< V_{T0n}$) AND V_2: 0 → V_{DD} (M2 → OFF)

Data-Write Operation (Cont.)

- Design constraint: $V_{1,max} \leq V_{T,2}$ so M2 turns OFF when $V_1 = V_{T,2}$. M3 is in linear region whereas M5 operates in saturation:

$$\frac{k_{n,3}}{2}\left(2(V_{DD}-V_{T,n})V_1 - V_1^2\right) \geq \frac{k_{p,5}}{2}\left(0 - V_{DD} - V_{T,p}\right)^2 \quad \text{at} \quad V_1 = V_{T,n}$$

$$\Rightarrow \frac{k_{n,3}}{k_{p,5}} \geq \frac{(V_{DD}+V_{T,p})^2}{2(V_{DD}-1.5V_{T,n})V_{T,n}}$$

$$\frac{\left(\frac{W}{L}\right)_3}{\left(\frac{W}{L}\right)_5} \geq \frac{\mu_p}{\mu_n} \cdot \frac{(V_{DD}+V_{T,p})^2}{2(V_{DD}-1.5V_{T,n})V_{T,n}}$$

A typical example:

With $V_{DD} = 2.5V, V_{T,n} = -V_{T,p} = 0.4V, \frac{\mu_n}{\mu_p} = 2.25$

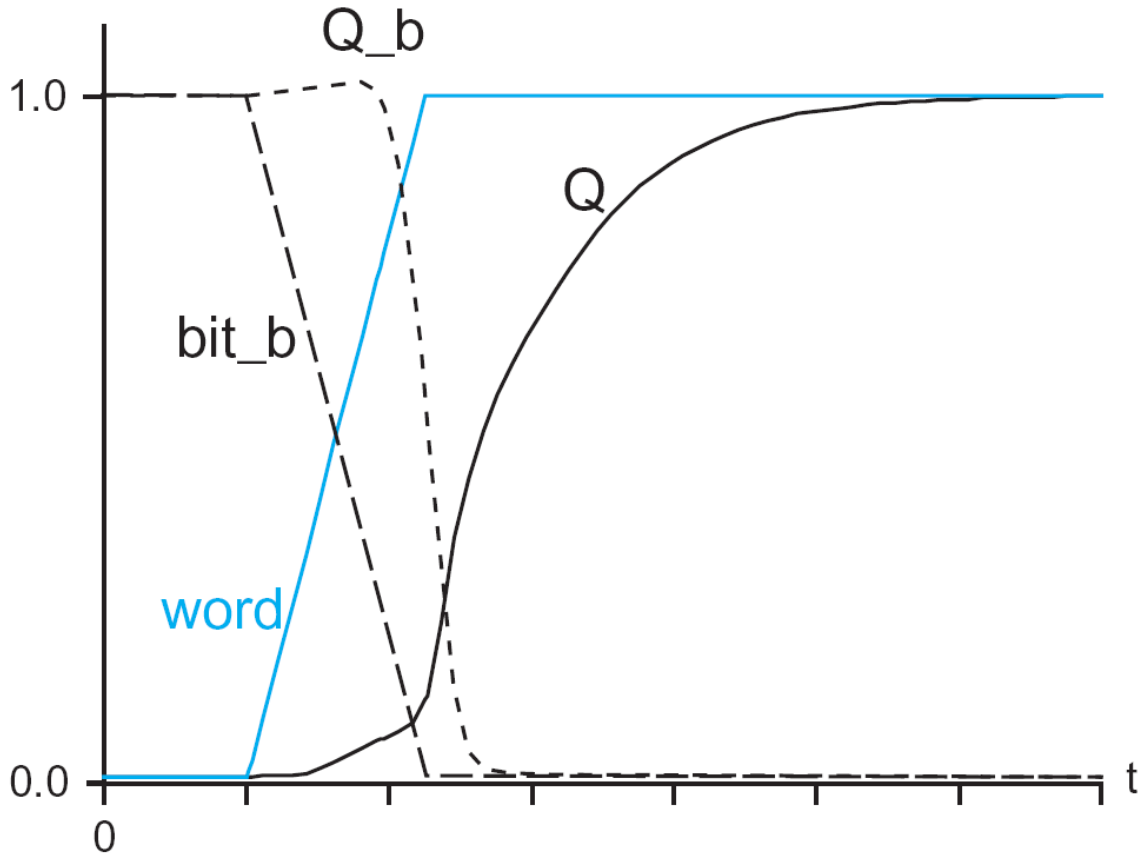
$$\frac{\left(\frac{W}{L}\right)_3}{\left(\frac{W}{L}\right)_5} \geq \frac{1}{2.25} \cdot \frac{(2.1)^2}{2(1.9)0.4} \Rightarrow \left(\frac{W}{L}\right)_3 \geq 1.3\left(\frac{W}{L}\right)_5$$

- A symmetrical condition also dictates the aspect ratios of M6 and M4

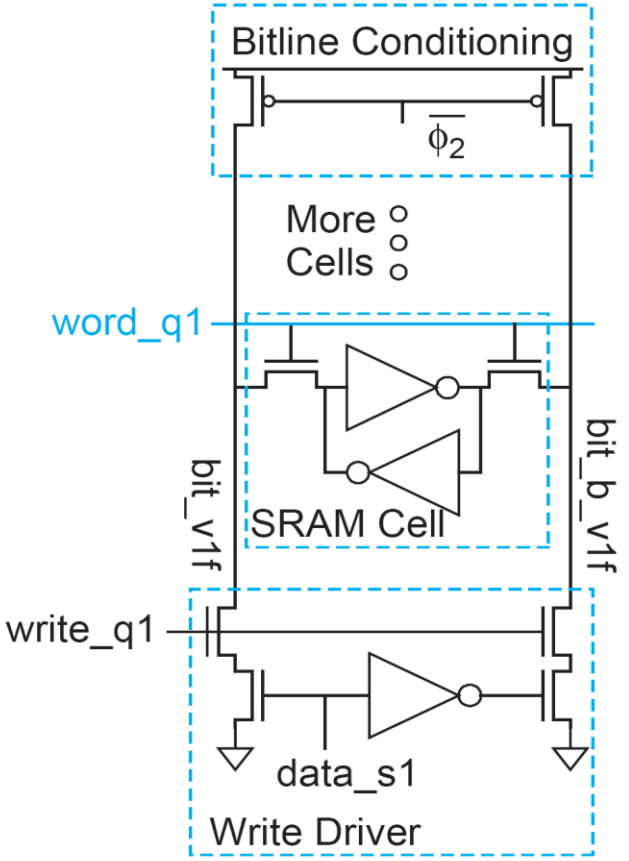
Write Operation (Cont.)

$V_1 = Q$

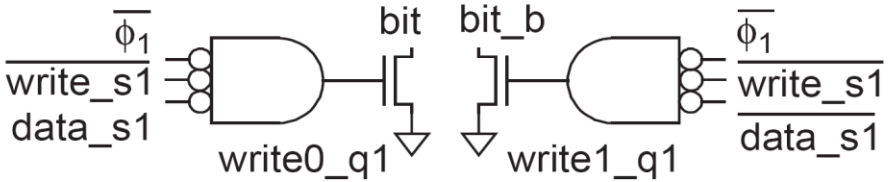
$V_2 = Q\_b$



SRAM Column Write



(a)

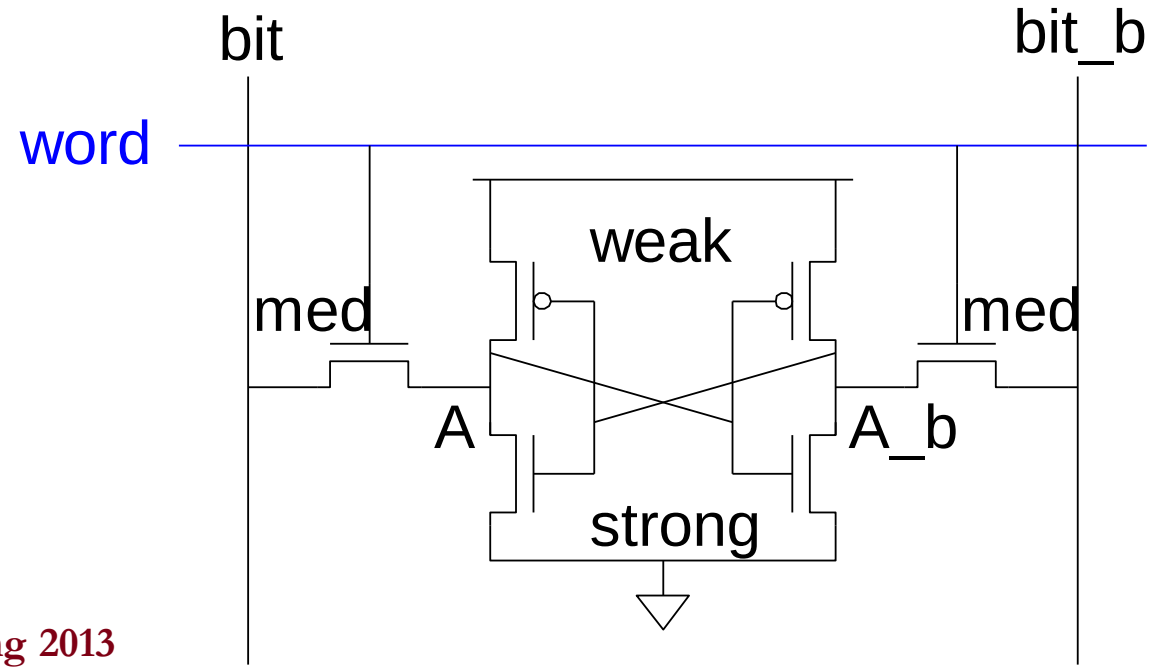


(b)

SRAM Sizing Summary

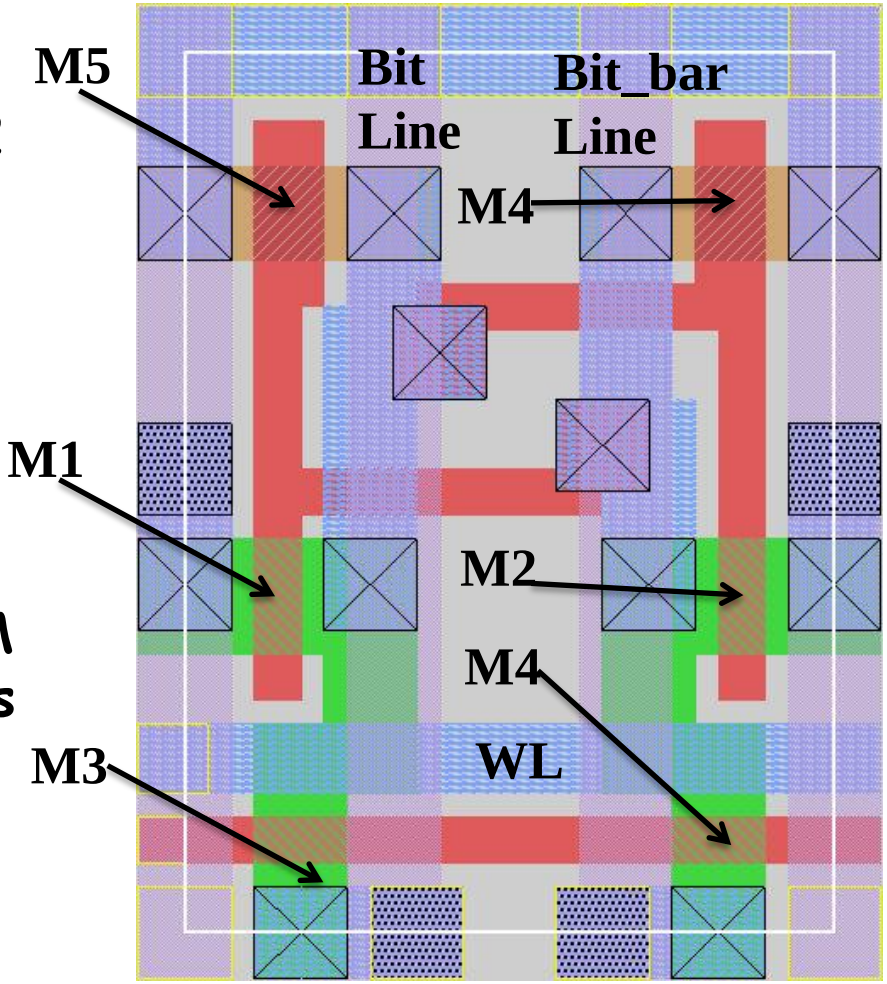
- Bls are high during read, they should not overpower the inverters during read, therefore nMOS transistors should be strong to pull them down

- However during write, the bls need to overpower, so we make pMOS transistors weak



Typical SRAM Transistor Sizes

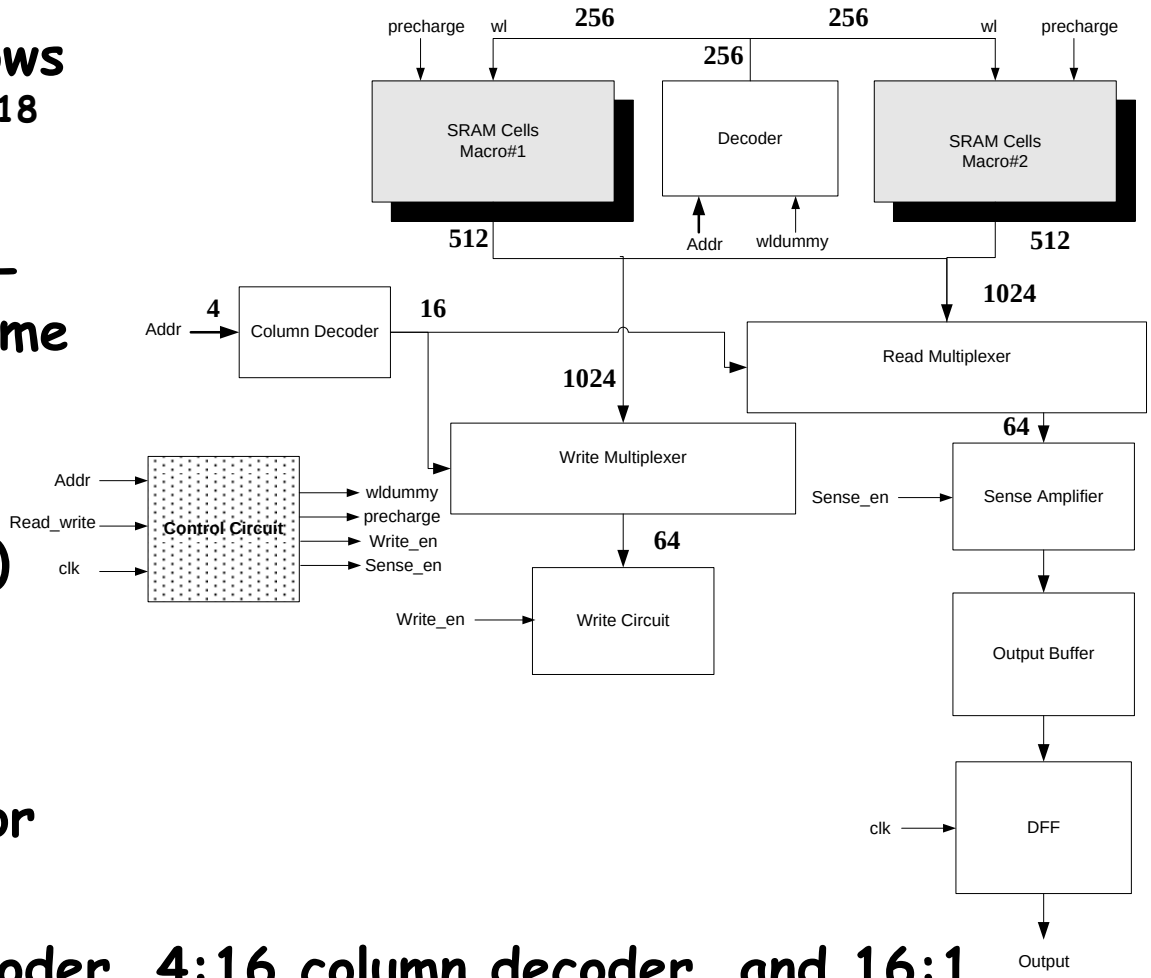
- Example I: Transistors may be sized as follows:
  - nMOS pulldowns:M1,M2: 6:2
  - pMOS pullups: M5, M6: 4:3
  - Access xtors: M3, M4: 4:2
- All boundaries are shared
  - Reduces the Write delay
- Example II: One may also use equal-size transistors in the SRAM cell (e.g., 4:2 for all) however this should be carefully checked, as this sizing is not conservative may not work for all scenarios
- Example III: pulldowns: 8:2, pullups: 3:3, access xtors: 4:2



Yet a different layout

Example Design of a 256Kbit SRAM Array

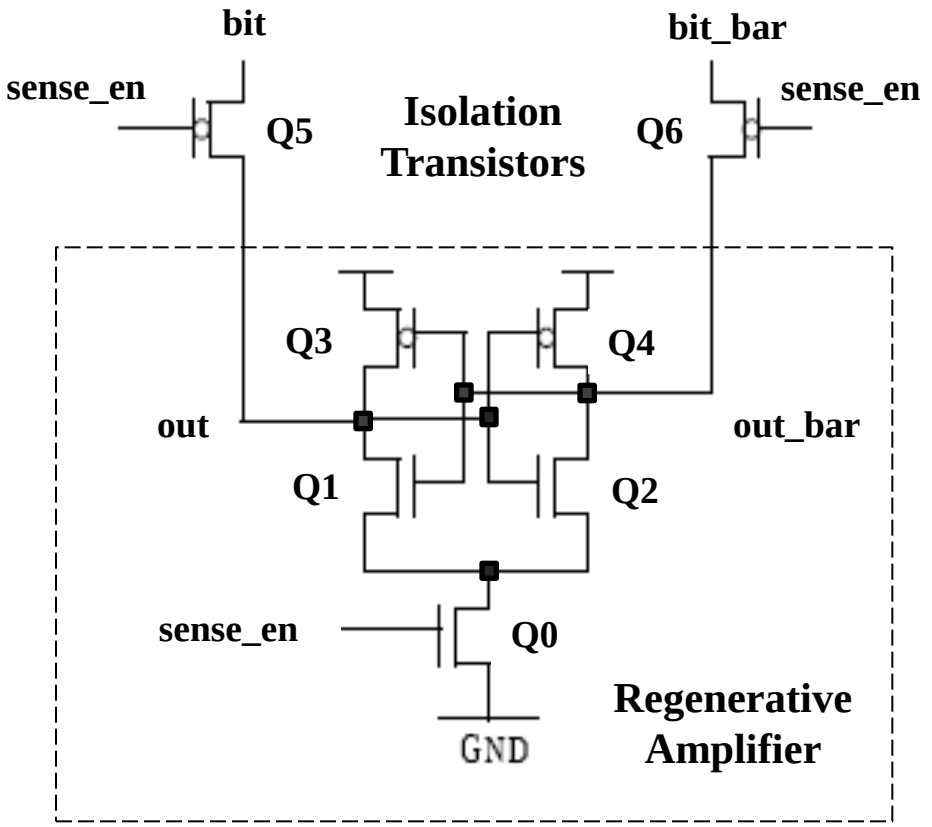
- 2 macro-blocks (aka 2 banks), each with 256 rows and 512 columns (total 2^{18} bits)
- Want to access a double-word (2^6=64 bits) at a time
- Need 12 address lines $A_0,\ldots,A_{11}$
  - 4 LSB bits ($A_0,\ldots,A_3$) are used for column addressing while the other 8 MSB bits ($A_4\ldots A_{11}$) are used for row addressing
  - Need 8:256 row decoder, 4:16 column decoder, and 16:1 Read and Write Multiplexers
  - Use 64 sense amplifiers



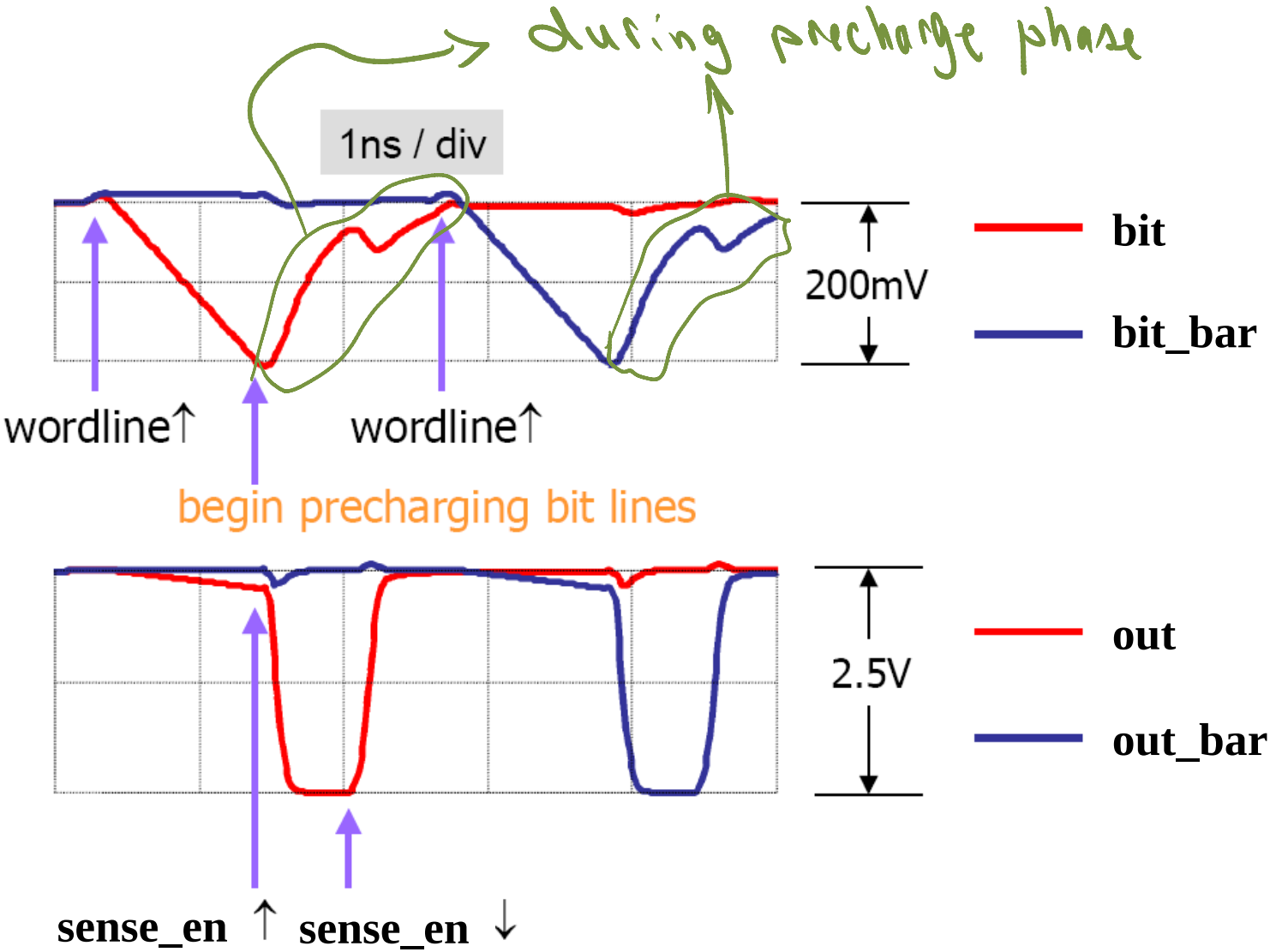
Sense Amplifier

- The bit line capacitance is significant for large arrays
  - If each cell contributes 2fF, with 256 cells per column, we get 512fF plus wire cap. Pull-down resistance is about 15K. The propagation delay will be 5.3ns (with $\Delta V = V_{DD}$) ?
- We cannot easily change R, C, or V_{DD}, but can change ΔV !
  - It is possible to reliably sense ΔV's as small as 50mV
  - With margin for noise, most SRAMs sense bit-line swings between 100~300mV
- For writes, we still need to drive the bit line to full-swing
  - Only one driver needs to be this big

Use SPICE sweep function to optimize transistor sizes (typically, Q0, Q5, Q6 are minimum-size transistors)



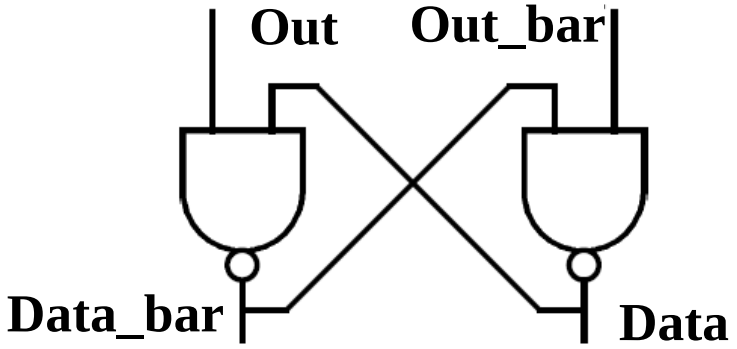
Sense Amplifier Waveforms



Comments about the Sense Amp Design

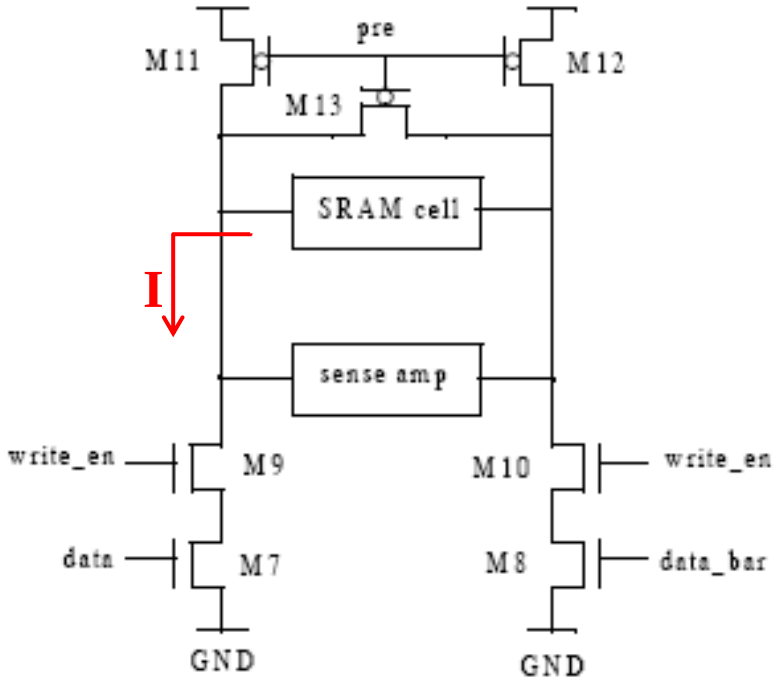
- Isolation transistors must be pMOS
  - Bit lines are within 0.2V of V_{DD} (not enough to turn an nMOS transistor ON)
- Load on outputs of regenerative amplifier must be equal
- Need to precharge the sense amp before opening the isolation transistors to avoid discharging the bit lines

- Both outputs go high during precharge
  - Usually follow the regenerative amplifier by a cross-coupled NAND latch
- Requires 3 timing phases
  - Typically self-timed



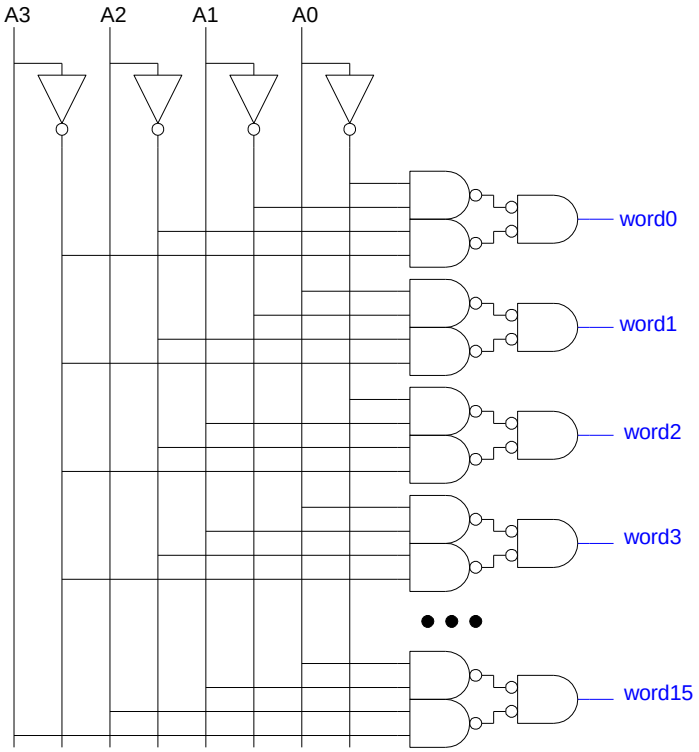
Precharge and Write Circuitry

- Recall that M3 and M5 denote the nMOS access transistor and the pMOS pullup transistor inside the SRAM cell (on the bit line side)
- For successful write operation, R3+R9+R7 should be < $\frac{1}{2}$R5
- Let R* denote resistance of 2:2 nMOS transistor, and $\mu_n/\mu_p=2$
- If M3=4:2 and M5=4:3, then $\frac{1}{2}$R5 = $\frac{3}{4}$R* and $R_3 = \frac{1}{2}R^*$; therefore, R9+R7 should be $\frac{1}{4}$R*
- M9 and M7 should be 16:2 each



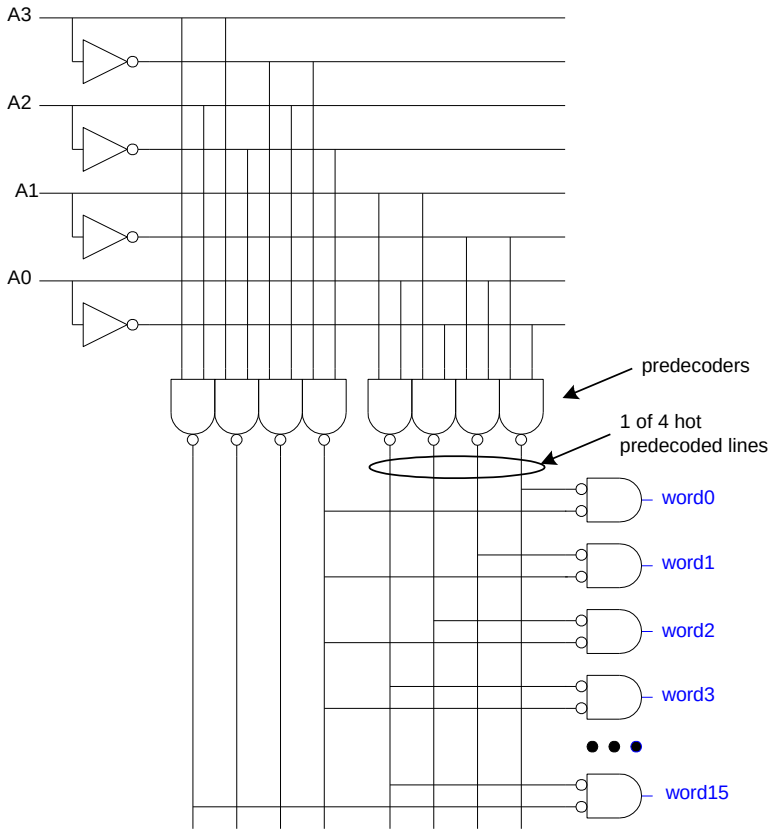
Large Decoders

- For n > 4, NAND gates become slow
  - Break large gates into multiple smaller gates

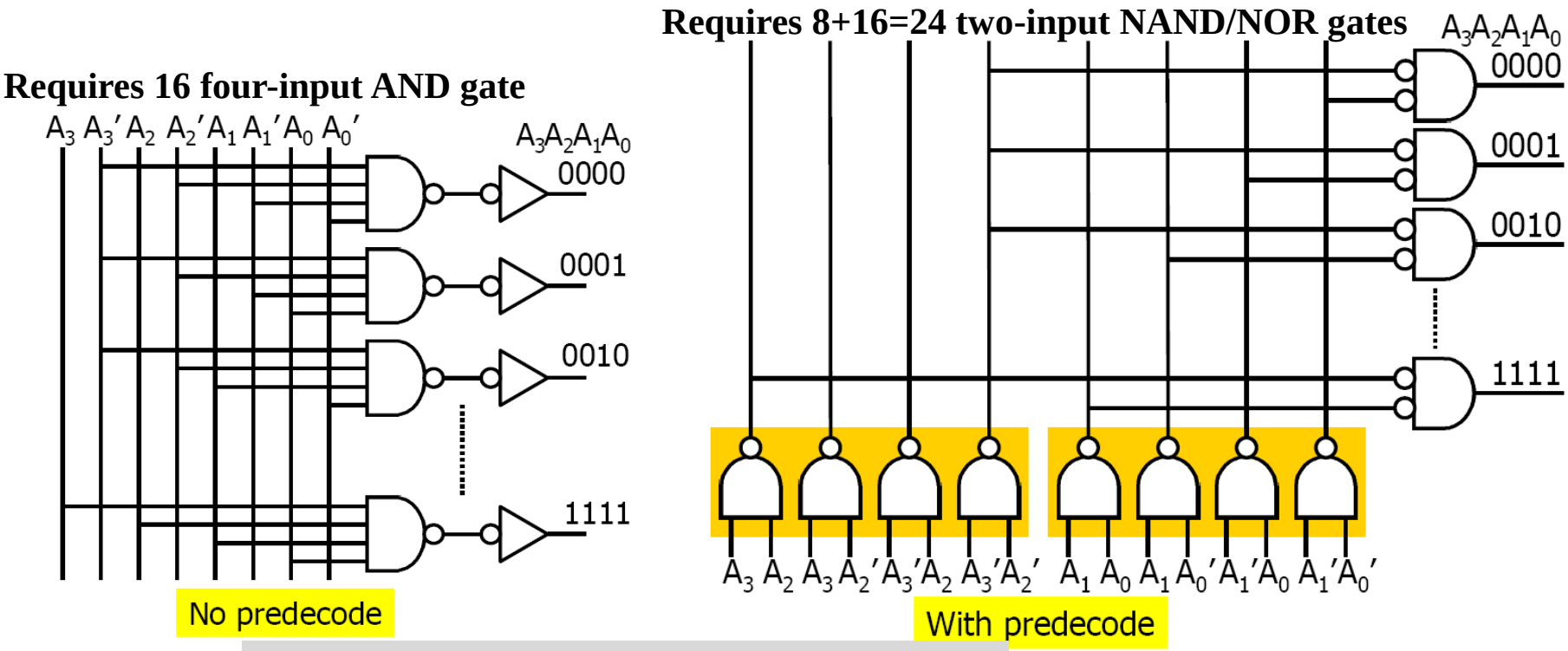


Predecoding

- Many of these gates are redundant
  - Factor out common gates into predecoder
  - This saves area
  - Optional for now: Same path effort (as later covered using logical effort)



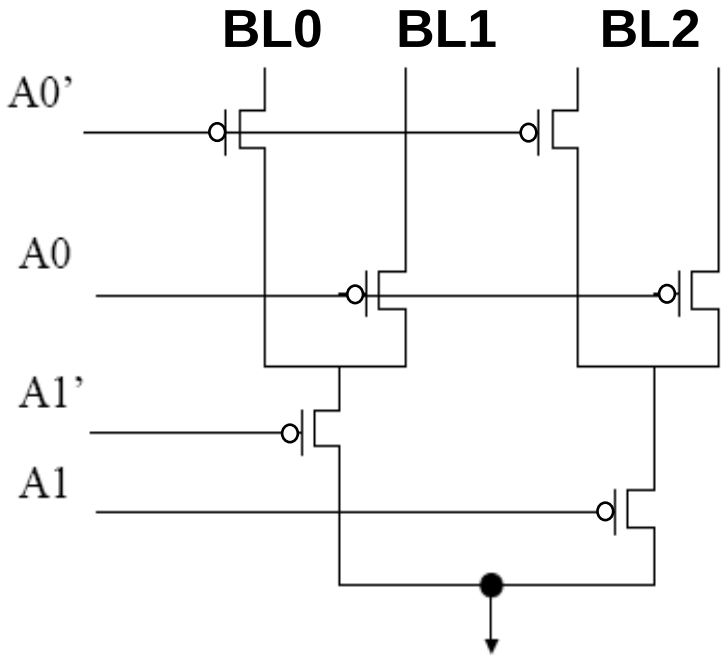
Row Decoder



Two implementations of a 4:16 decoder

- Another example of pre-decoding addresses – decode in octal addresses
  - One-level decoding of 9-bit address (A8 A7 … A0) requires 512 nine-input AND gates
  - Predecode (A2 A1 A0), (A5 A4 A3), and (A8 A7 A6) by using $3*2^3=24$ three-input NAND gates, followed by $8^3=512$ three-input NOR gates

4-to-1 Tree Multiplexer for Read Circuitry

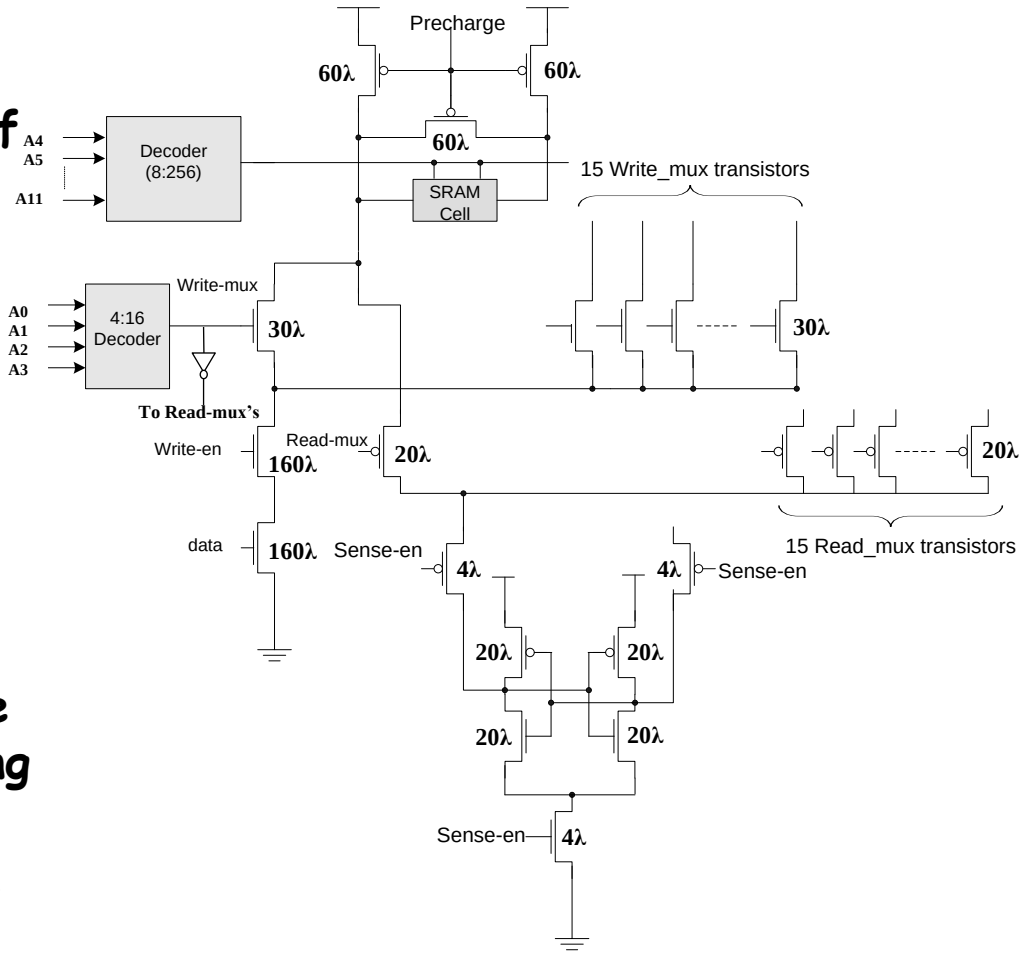


Are able to use **pmos** only pass transistors because of limited swing.

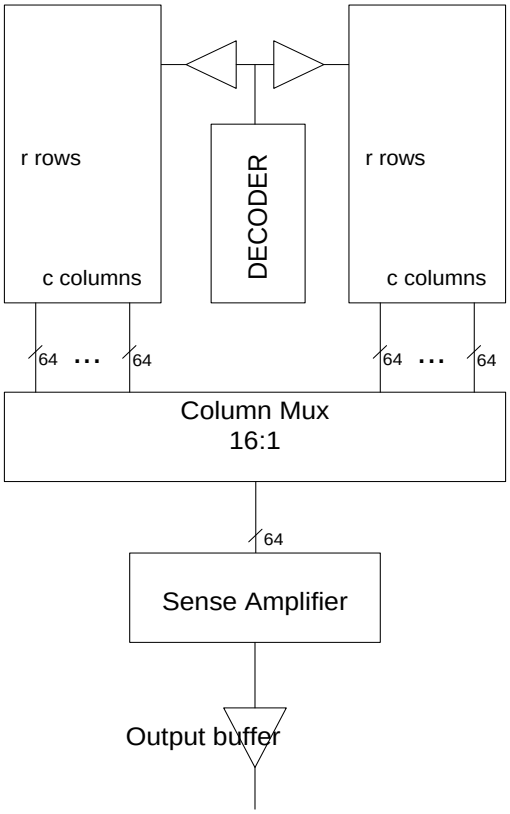
Number of pass transistors in series is a concern, but limited swing helps speed.

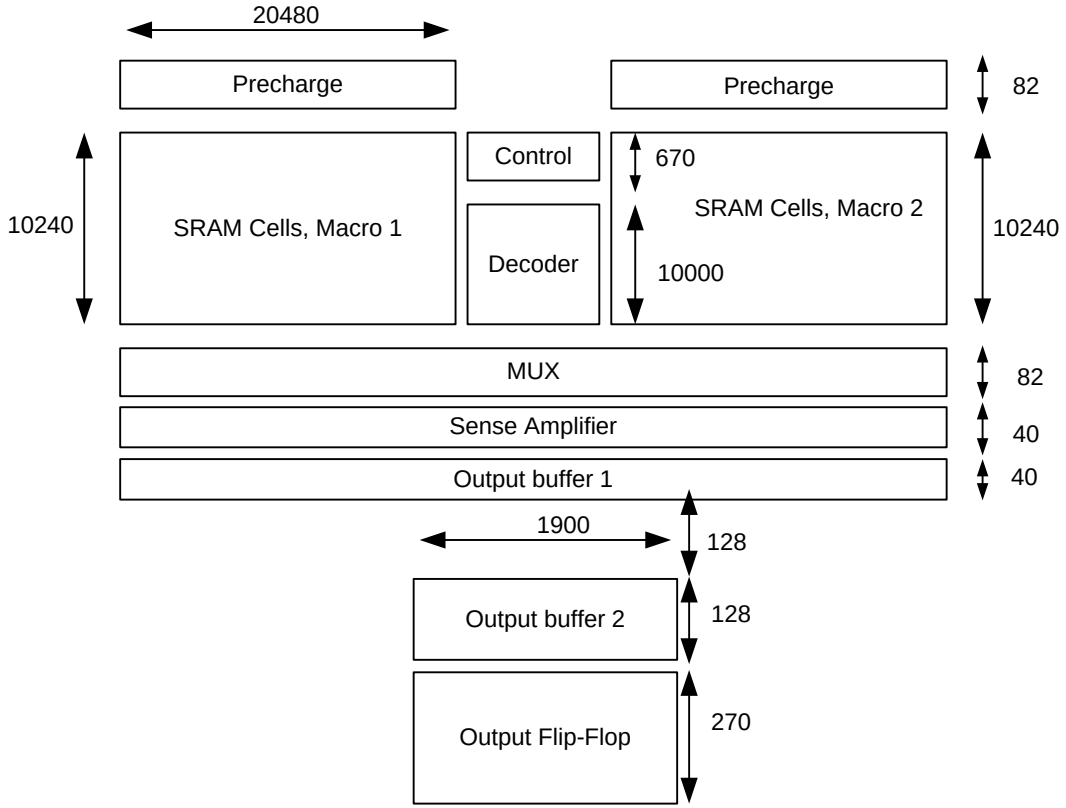
Column Mux

- We have 16 read_mux and 16 write_mux transistors in parallel
  - During read operation, one of the read MUX's is selected, according to the values of $A_0,\ldots,A_3$, and that column enables the sense amplifier and the corresponding value of SRAM cell will be read at the output of sense
  - During the write operation, the desired SRAM cell is selected and the data will be written into the corresponding SRAM cell
- Need to replicate the drawing for the bit_bar side
- Need a total of 64 similar structures which makes 16:1 64-bit wide column MUX



SRAM Array Floor plan





All units are in λ

Example Read Delay Calculation for an SRAM Array

- Consider a *256×512* SRAM core. Bit lines are pre-charged to V_{DD} = *2.5V* before each read operation. A read operation is complete when the bit line has discharged by *0.25V*. A memory cell can provide *0.25mA* of pull-down current to discharge the bit line. Assume the word line resistance is *2Ω* per memory cell, the word line capacitance is *20fF* per memory cell, while the bit line capacitance is *12fF* per cell. Ignore the bit-line resistance and read-mux transistor. Calculate the worst-case read delay for this SRAM. Assume row decoding takes *3ns* while sense amplifier and output buffer take *1ns*.
- Solution: Each word line drives 512 SRAM cells; The RC delay for driving the furthest cell is:

$$t_{row} = 0.69 \times \sum_{k=1}^{N=512} \left(\sum_{j=1}^{k} R_j \right) \cdot C_k = 0.69 \times \left(\frac{N(N+1)}{2} \right) R_{cell} \cdot C_{cell} = 0.69 \times 256 \times 513 \times 20f \times 2 = 3.7ns$$

- The time needed to discharge the bit or bit_bar line by *250mV* is:

$$t_{col} = \frac{C_{col} \Delta V}{I_{dis}} = \frac{256 \times 12f \times 0.25}{0.25m} = 3.1ns$$

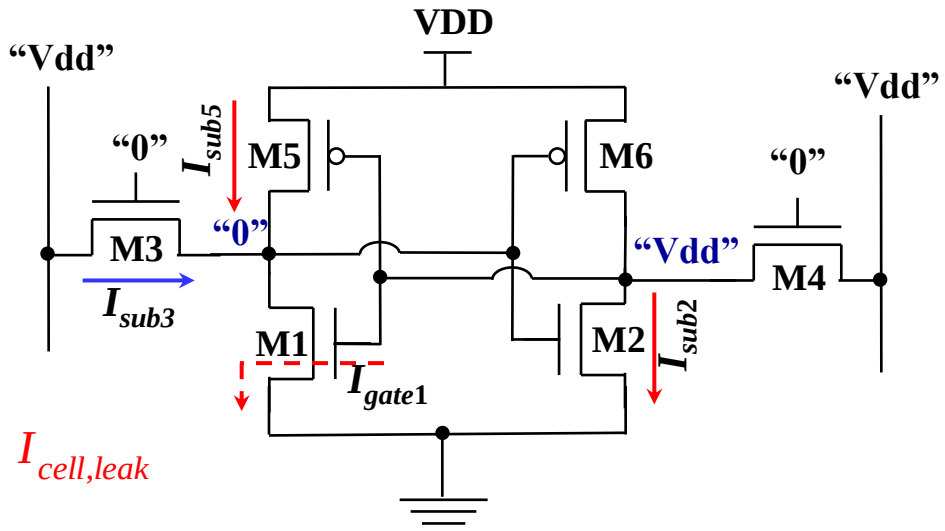
$$t_{access} = t_{dec} + t_{row} + t_{col} + t_{sen+buf} = 10.8ns$$

SRAM Scaling Challenges

- For cell stability, separate power rails for cell array vs. word line driver may be needed (bad for leakage)
- Reduced read and write margins as we scale voltages
- Increased transistor leakage (high-k gate dielectric)
- Introduction of various power management modes:
  - Reduced V_{DD}
  - Raised V_{SS}
- Soft error immunity
- Low standby power

Leakage Currents in the SRAM Cell

$$I_{leak} = I_{gate1} + I_{sub3} + I_{sub2} + I_{sub5}$$
$$\gg I_{sub3} + (I_{sub2} + I_{sub5}) = I_{bitline,leak} + I_{cell,leak}$$



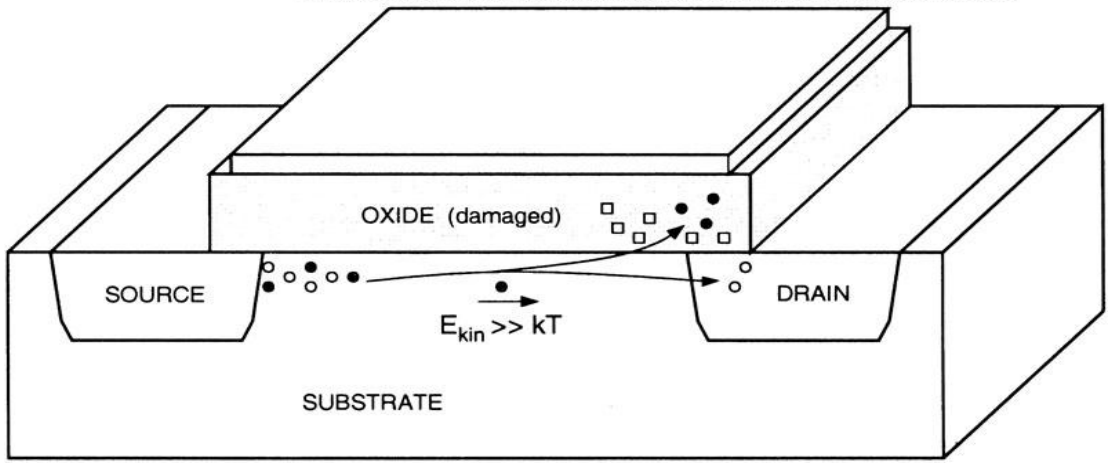
- Note that I_{leak} is dominated by the drain-source leakage in 90nm CMOS technology (i.e., we may ignore gate leakage and other leakage mechanisms which are small compared to the sub-threshold conduction currents)

Bitcell Stability Failures

- All transistors should be small to achieve high layout density, however for read stability and writability sizing should be carefully done considering instability effects, process, temperature, and voltage variations
- Read Access Failure
  - The WL activation period is too short for a pre-specified ΔV to develop between bit line and bit_bar line in order to trigger the sense amplifier correctly during read
    - This may occur due to increase in V_t for the pass-gate or pull-down transistors
      - Negative bias temperature instability
      - Hot carrier effects

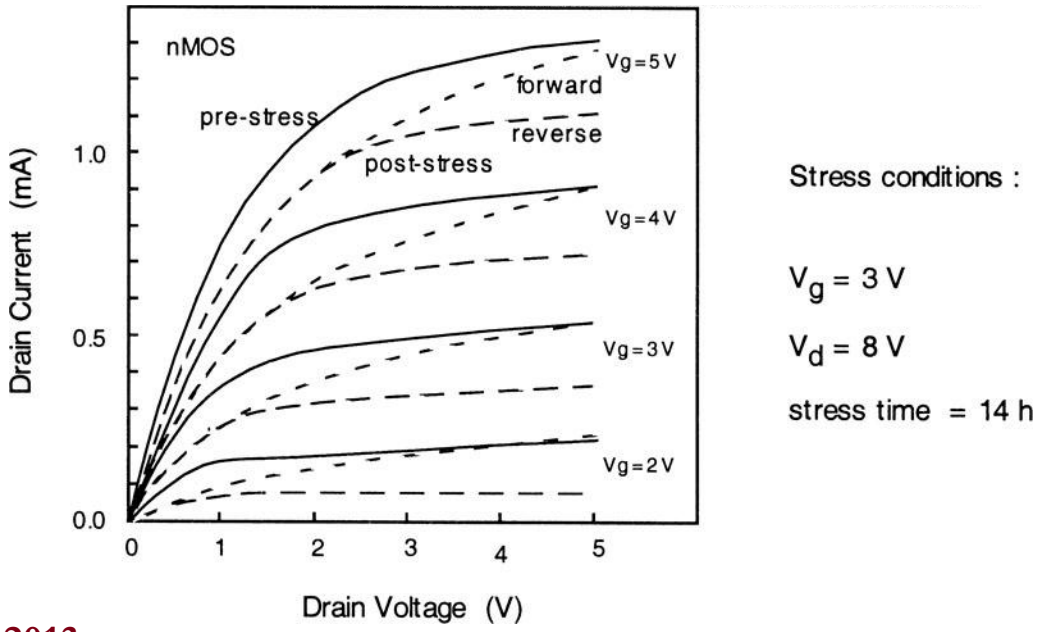
Optional (From EE477L) - Hot Carrier Effect

- Lower dimensions and higher substrate doping in small geometry technologies would result in large electric fields under gate which in turn give rise to electrons and holes with kinetic energies significantly higher than silicon band gap
- These electrons and holes may be injected into the gate oxide and can cause permanent changes in the oxide interface charge distribution. This is called hot carrier effect, or sometimes hot electron effect because this injection happens more often for electrons due to smaller barrier height of electrons as compared to holes



Optional (From EE477L) - Hot Carrier Injection Into the Gate Oxide

- A sizeable increase in the threshold voltage of the affected transistors and a corresponding decrease in their drain current driving capability are undesirable results of such hot carrier injections in the gate oxide
- The hot carrier effect is exacerbated as the technology moves toward smaller device dimensions and higher clock frequencies



Bitcell Stability Failures (Cont.)

- **Read Stability Failure**
  - Cell may flip due to increase in the "0" storage node above the trip voltage of the other inverter during a read
    - To quantify the bitcell's robustness against this failure, SNM is the most commonly used metric
    - Notice that read stability failure can occur anytime the WL is enabled even if the bitcell is not accessed for read or write operations
    - SNM related failures are the limiter for V_{DD} scaling especially after accounting for device degradation due to *hot electron effects* and *negative bias temperature instability*

Negative Bias Temperature Instability

- NBTI is a key reliability issue in MOSFETs and manifests as an increase in the threshold voltage and consequent decrease in drain current and transconductance and switching speed reduction
- The degradation exhibits logarithmic dependence on time

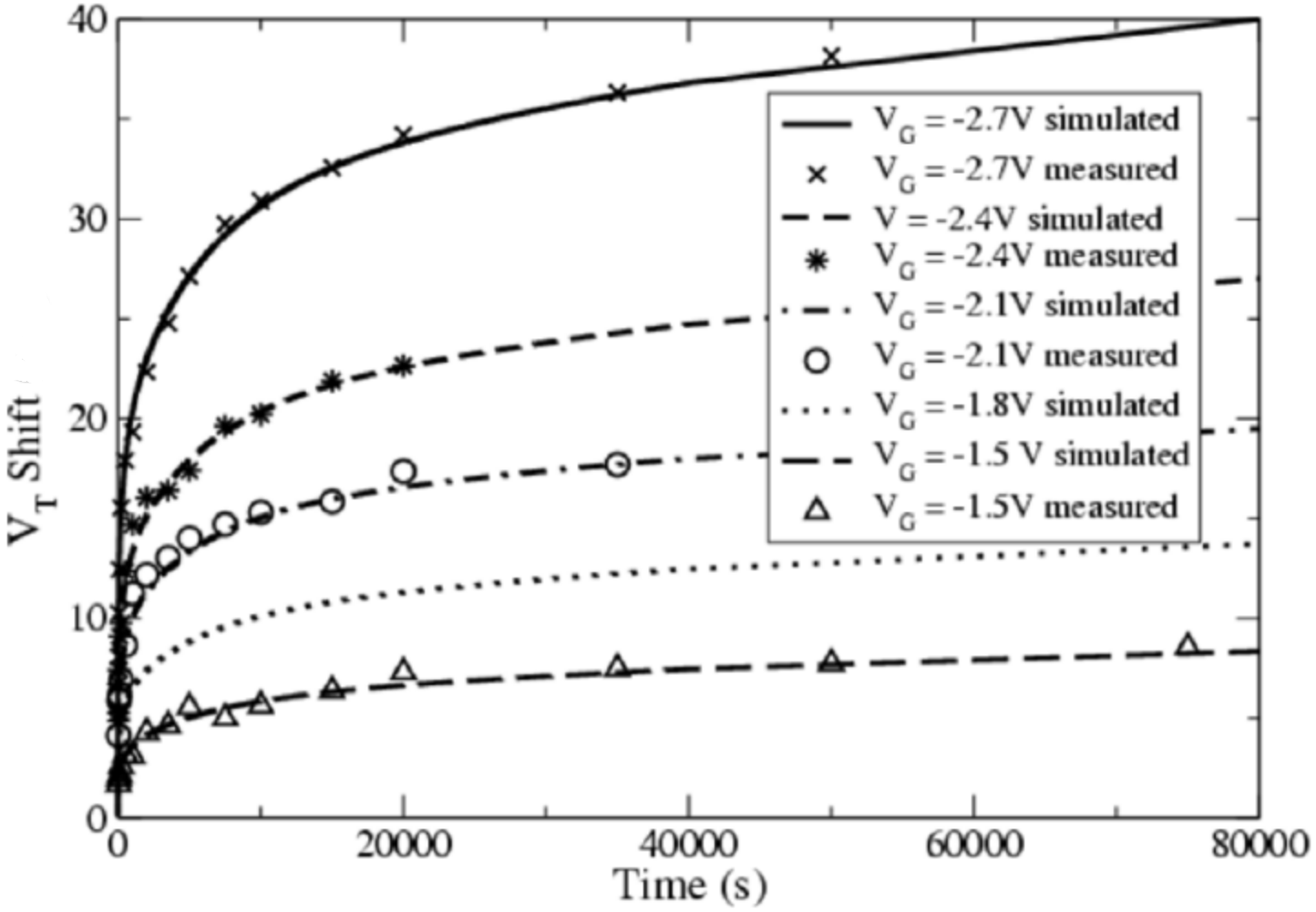
- At the atomic level, NBTI is caused by an electric field dependent disassociation of Si-H bonds at the Si-SiO2 interface
- The freed hydrogen diffuses into the oxide, resulting in interface traps that increases the threshold voltage

NBTI (Cont.)

- This disassociation is most prevalent for PMOSFETs under negative bias (Vgs = $-V_{DD}$)
- When the stress is removed (Vgs = 0), the diffusions reverses and some of the hydrogen can rebond with the Si, removing the interface traps. This reversal is called the recovery effect
- The very same mechanism also affects nMOS transistors when biased in the accumulation regime, i.e. with a negative bias applied to the gate too

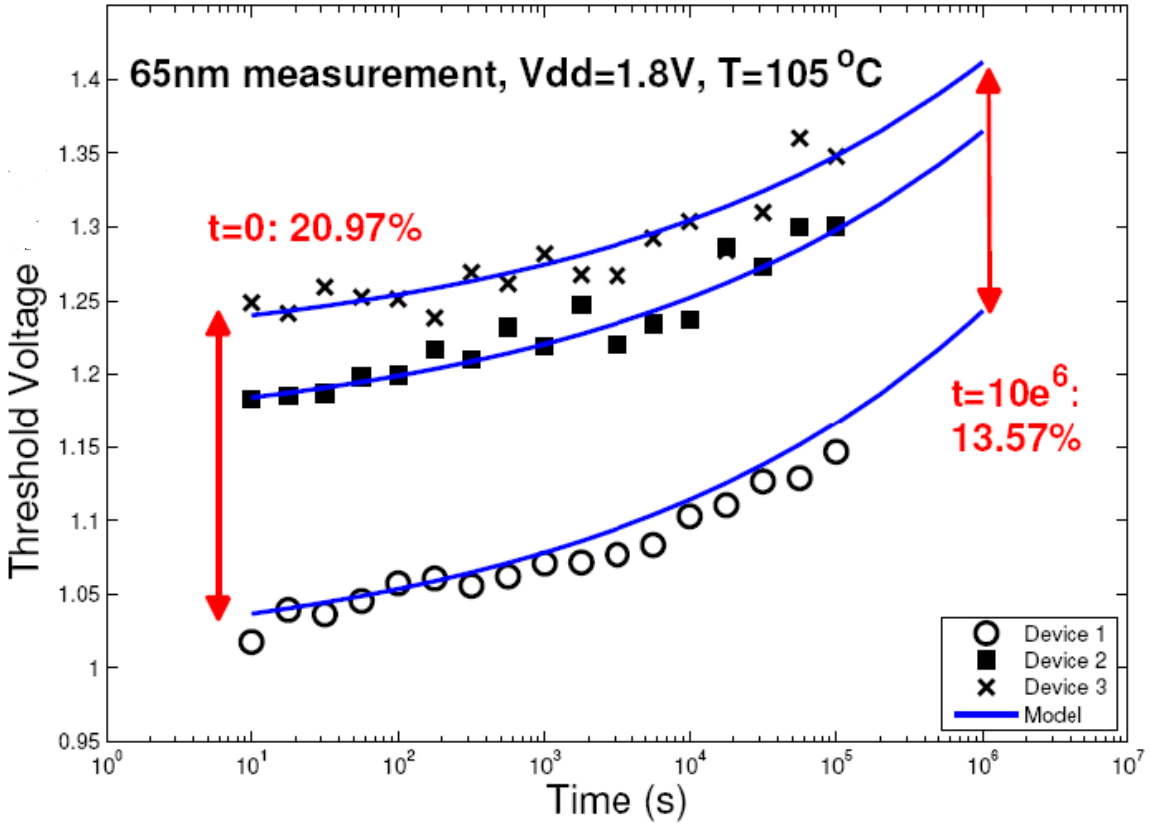
NBTI (Cont.)

- ΔV_t increases exponentially with increasing $|V_{gs}|$



NBTI (Cont.)

- NBTI and process variation on V_t



ΔVth process ≡ ΔVth NBTI

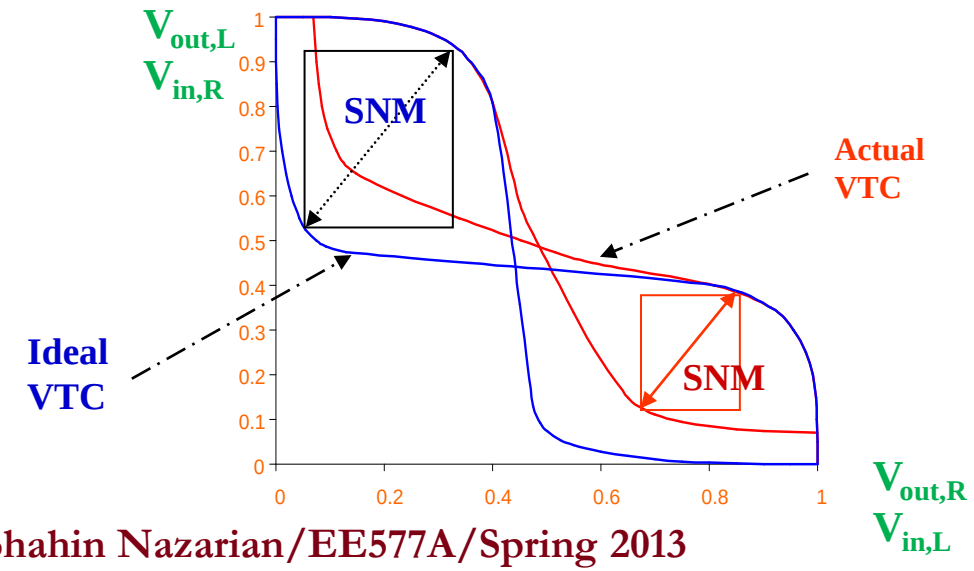
ΔVth process + ΔVth NBTI

=> Overall process variation reduced

Source : W. Ping et al. DAC 2007

Read Stability Failure in the SRAM Cell

- A read stability (a.k.a. "hold") failure occurs when stored data flips during the memory standby mode (while WL is enabled)
  - A cell's VTC is composed of the two inverters' VTCs that enclose two regions
  - The cell's hold stability is characterized by the static noise margin (SNM), which is measured by the diagonal length of the largest square fitted in the enclosed region (the derivation is omitted)



The SNM butterfly curves must be analyzed for different process corners, FS: fast NMOS, slow PMOS and SF: slow NMOS, fast PMOS and different temperatures

Bitcell Stability Failures (Cont.)

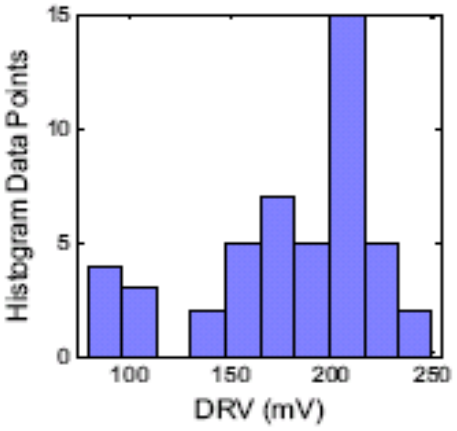
- Write Stability (or write-ability) Failure
  - The internal "1" storage node may not be reduced below the trip point of the other inverter during the WL activation period
  - One way to quantify a cell's write stability is to use write trip voltage or *write margin* (WM), which is the maximum bit line voltage at which the bitcell flips state, assuming that bit line is pulled to GND by the line driver

Bitcell Stability Failures (Cont.)

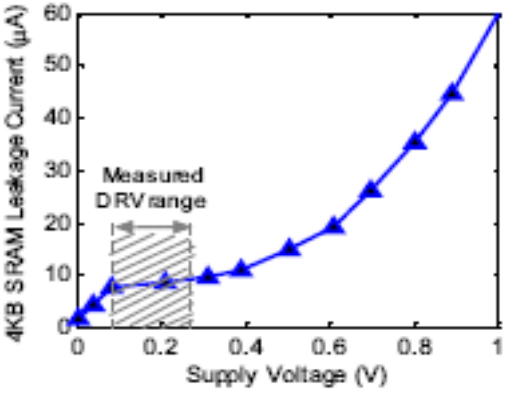
- Data Retention Failure
  - When V_{DD} is reduced to the Data Retention Voltage, all six transistors in the SRAM cell operate in subthreshold region, hence, they show strong sensitivity to variations
  - PMOS transistor must provide enough current to compensate for leakage in the NMOS pull-down and access transistors
  - Due to L and V_T variations, data retention current may not be sufficient to compensate the leakage current

Minimum Voltage Needed to Preserve Data

- The Data Retention Voltage (DRV) is defined as the minimum V_{DD} under which the data in a SRAM cell is still preserved
  - When V_{DD} is reduced to DRV, all transistors are in the sub-threshold region, thus SRAM data retention strongly depends on the sub-V_T current conduction behavior (i.e., leakage)
  - Cell leakage is greatly reduced at DRV
  - This provides a highly effective leakage suppression scheme for standby mode
    - Maximum leakage saving and minimum design overhead

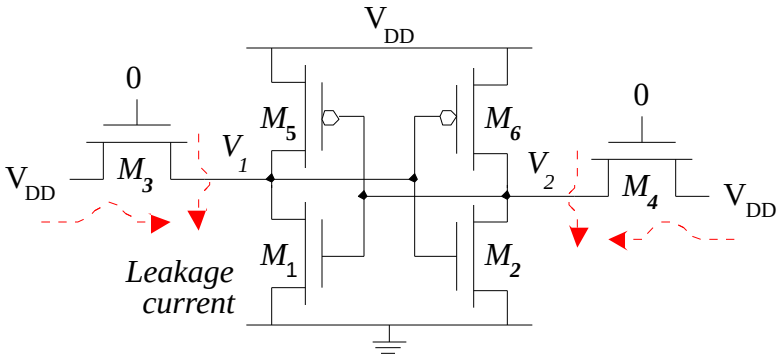


Distribution of DRV in a 0.13u CMOS with 3σ variations in VT and L



Measured SRAM leakage current

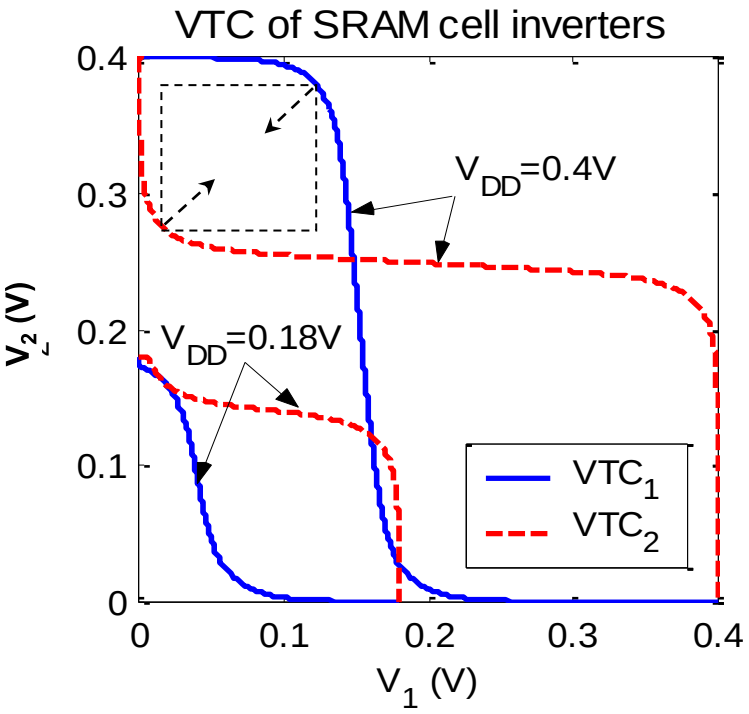
DRV of SRAM (Cont.)



DRV Condition:

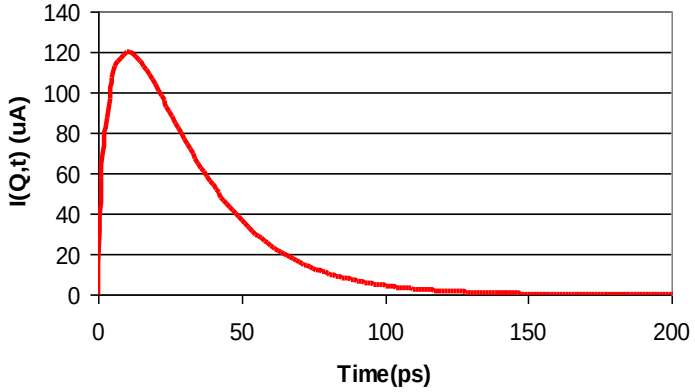
$$\left.\frac{\partial V_1}{\partial V_2}\right|_{\text{Left inverter}} = \left.\frac{\partial V_1}{\partial V_2}\right|_{\text{Rightinverter}}, \text{ when } V_{DD} = \text{DRV}$$

- When V_{DD} scales down to DRV, the Voltage Transfer Curves (VTC) of the internal inverters degrade to such a level that Static Noise Margin (SNM) of the SRAM cell is reduced to zero
- The temperature coefficient of DRV is 0.169mV/°C, which implies an increase of 12.3mV in DRV when temperature rises from 27°C to 100°C



Soft Error Rate for the SRAM Cell

- A high-energy alpha particle or an atmospheric Neutron striking a capacitive node
  - Deposits charge leading to a time-varying current injection at the node



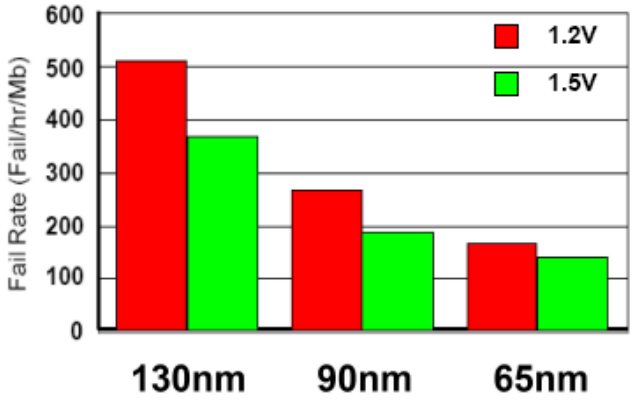
- In case of atmospheric Neutrons:

$$I(Q,t) = \frac{2Q}{\sqrt{p}T_s}\sqrt{\frac{t}{T_s}}\exp(\frac{-t}{T_s})$$

- If collected charge Q_s exceeds some critical charge level Q_{crit}, it will upset bit value and cause a soft error
- *Soft Error Rate* (SER) in SRAM:
- Q_{crit} is 10fC in a 65nm CMOS process

$$SER \;;\quad \exp(\frac{-Q_{crit}}{Q_s})$$

How to Mitigate the SER Fail Rate



Good News: SER per bit value tend to decrease with scaling

- To mitigate soft errors, several radiation-hardening techniques can be implemented
  - Process technology changes (e.g., SOI technology)
  - Circuit design (e.g., adding capacitor, using larger transistors, memory words interleaving)
  - Architecture (e.g., parity, error correction codes)