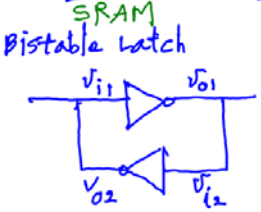
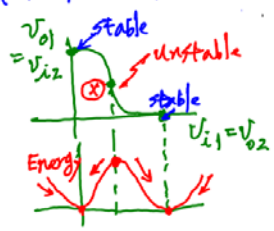
EE 222 Lecture 9 Feb. 9, 2018

SRAM

Bistable Latch





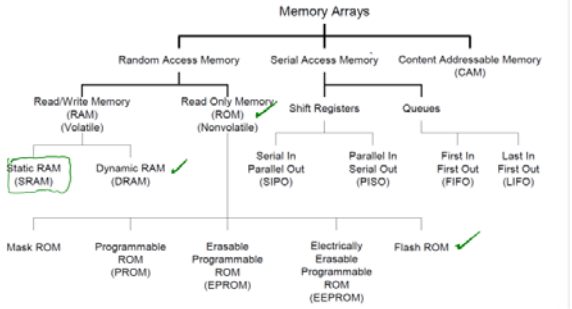
Two stable points

$(V_{i1} = V_{o2} \approx V_{DD},\ V_{o1} \approx V_{i2} \approx 0)$

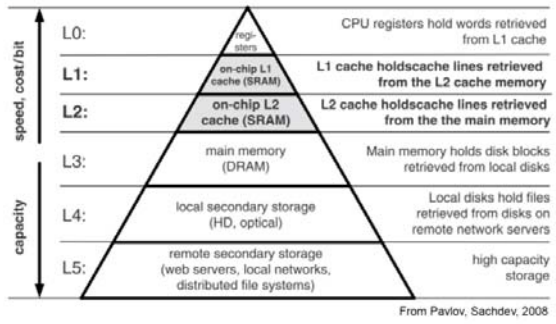
& $(V_{i1} = V_{o2} \approx 0,\ V_{o1} \approx V_{i2} \approx V_{DD})$

The Point ⊗ where $V_{i1}(\approx V_{o2}) \approx V_{o1}(\approx V_{i2})$ is unstable

Semiconductor Memory Classification

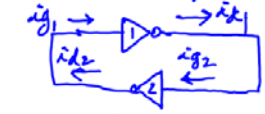


Memory Hierarchy of a Personal Computer



From Pavlov, Sachdev, 2008

Small signal analysis at ⊗.



$$\left.\begin{aligned} i_{g1} &= i_{d2} = g_m v_{g2} \\ i_{g2} &= i_{d1} = g_m v_{g1} \end{aligned}\right\} \quad (1)$$

$$v_{g1} = \frac{q_1}{C_g} \qquad v_{g2} = \frac{q_2}{C_g} \qquad \text{---------(2)}$$

Also

$$\left.\begin{aligned} i_{g1} &= C_g \frac{dv_{g1}}{dt} \\ i_{g2} &= C_g \frac{dv_{g2}}{dt} \end{aligned}\right\} \quad \text{---------(3)}$$

From (1) & (3)

$$\left.\begin{aligned} g_m v_{g2} &= C_g \frac{dv_{g1}}{dt} \\ g_m v_{g1} &= C_g \frac{dv_{g2}}{dt} \end{aligned}\right\} \quad \text{---(4)}$$

From (4)

$$g_m v_{g2} = C_g \frac{dv_{g1}}{dt} = C_g \frac{d}{dt}\left(\frac{C_g}{g_m}\frac{dv_{g2}}{dt}\right)$$

$$g_m v_{g2} = \frac{C_g^2}{g_m}\frac{d^2 v_{g2}}{dt^2}$$

$$\boxed{\frac{d^2 v_{g2}}{dt^2} - \left(\frac{g_m}{C_g}\right)^2 v_{g2} = 0}$$

let $\frac{1}{\tau} = \frac{g_m}{C_g}$, then $\frac{d^2 v_{g2}}{dt^2} - \tau^2 v_{g2} = 0$

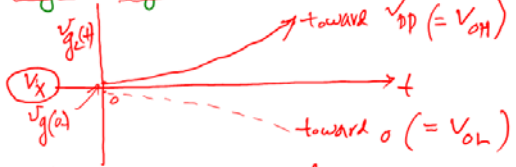
solving for v_{g2} by Laplace transformation

$$s^2 V_{g2}(s) - s v_{g2}(0-) - v'_{g2}(0-) - \left(\frac{1}{\tau}\right)^2 V_{g2}(s) = 0$$

For $v_{g2}(0-) = 0$

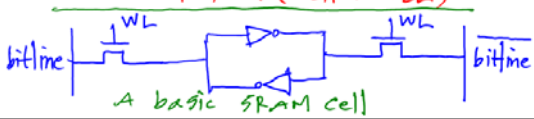
$$V_{g2}(s) = \frac{s v_{g2}(0-)}{s^2 - \left(\frac{1}{\tau}\right)^2} = \frac{1}{2} v_{g2}(0-)\left[\frac{1}{s+\frac{1}{\tau}} + \frac{1}{s-\frac{1}{\tau}}\right]$$

$$\Rightarrow v_{g2}(t) = \frac{1}{2} v_{g2}(0-)\left(e^{-t/\tau} + e^{+t/\tau}\right)$$

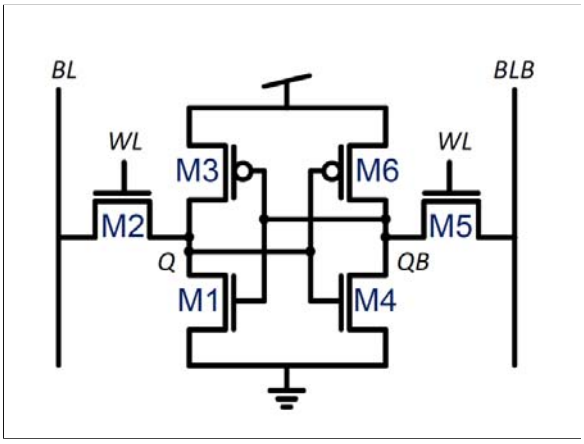


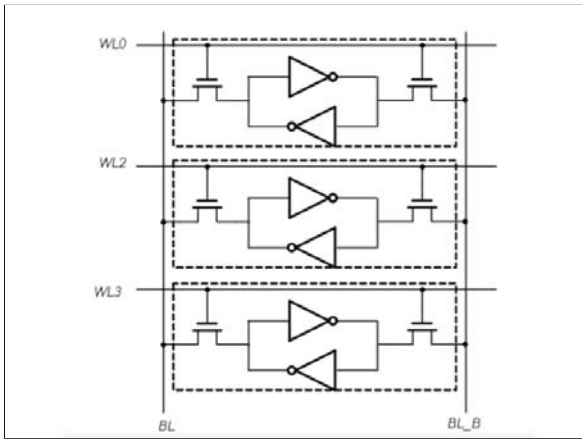
thus ⊗ is an unstable point

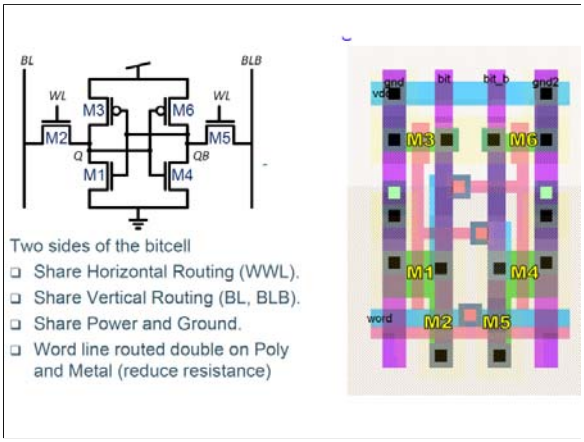
any perturbation will lead to migration to a stable Point (V_{OH} or V_{OL})

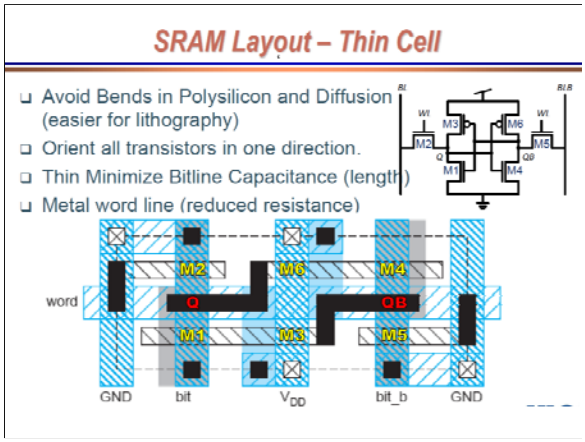


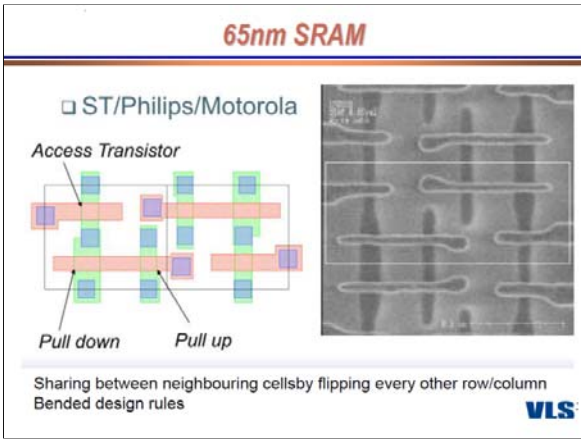
A basic SRAM cell

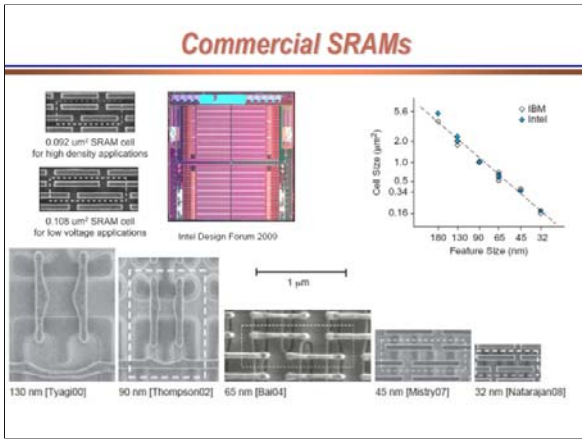

BL
BLB
WL
WL
M3
M6
M2
M5
Q
QB
M1
M4


WL0
WL2
WL3
BL
BL_B

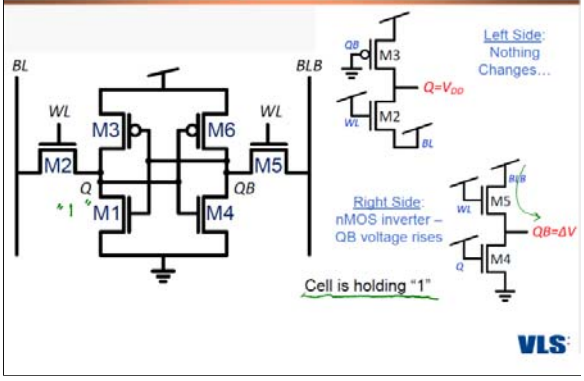

BL
BLB
WL
WL
M3
M6
M2
M5
Q
QB
M1
M4
Two sides of the bitcell
Share Horizontal Routing (WWL).
Share Vertical Routing (BL, BLB).
Share Power and Ground.
Word line routed double on Poly and Metal (reduce resistance)
gnd
bit
bit_b
gnd2
M3
M6
M1
M4
word
M2
M5


SRAM Layout – Thin Cell
Avoid Bends in Polysilicon and Diffusion (easier for lithography)
Orient all transistors in one direction.
Thin Minimize Bitline Capacitance (length)
Metal word line (reduced resistance)
word
M2
M6
M4
Q
QB
M1
M3
M5
GND
bit
V_{DD}
bit_b
GND

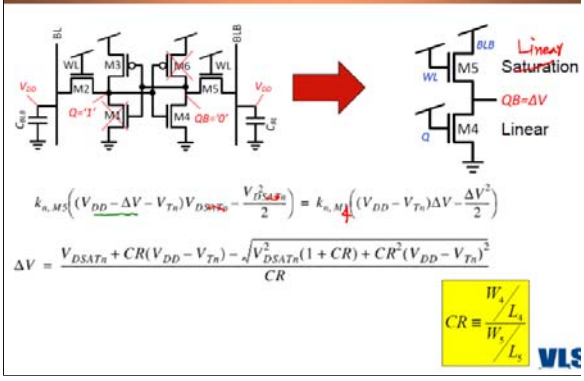

65nm SRAM
ST/Philips/Motorola
Access Transistor
Pull down
Pull up
Sharing between neighbouring cellsby flipping every other row/column
Bended design rules
VLS


Commercial SRAMs
0.092 um² SRAM cell for high density applications
0.108 um² SRAM cell for low voltage applications
Intel Design Forum 2009
IBM
Intel
Cell Size (µm²)
Feature Size (nm)
1 µm
130 nm [Tyagi00]
90 nm [Thompson02]
65 nm [Bai04]
45 nm [Mistry07]
32 nm [Natarajan08]

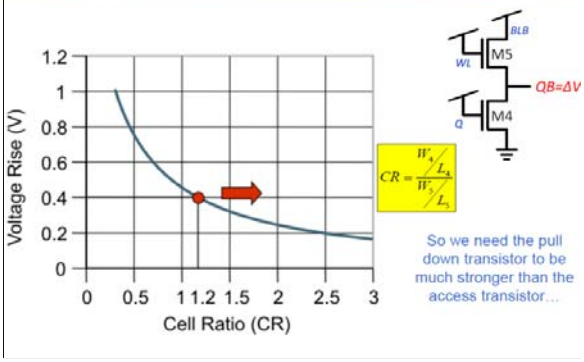
SRAM Operation: READ



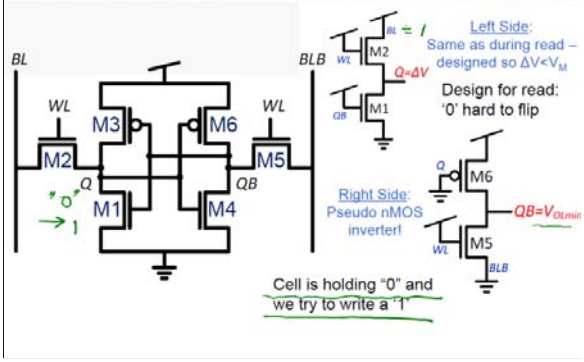
SRAM Operation - Read



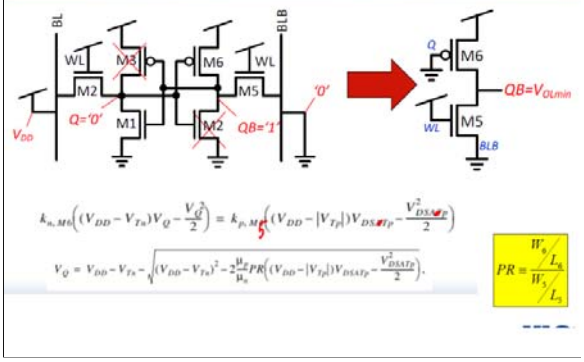
Cell Ratio (Read Constraint)



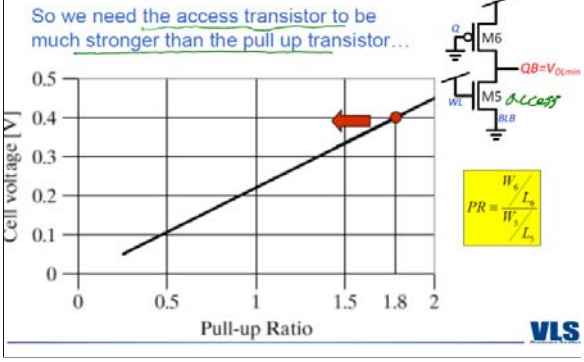
SRAM Operation: WRITE



SRAM Operation - Write

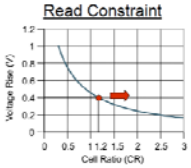


Pull Up Ratio – Write Constraint



Summary – SRAM Sizing Constraints

Read Constraint



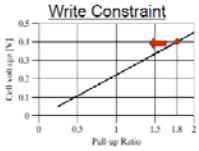
$$CR = \frac{W_1/L_1}{W_2/L_2} = \frac{W_4/L_4}{W_5/L_5} = \frac{PDN}{access}$$

Strong '0' (keep during read
Weak '1' (flip during write)

$$K_{PDN} > K_{access}$$

$$K_{PDN} > K_{access} > K_{PUN}$$

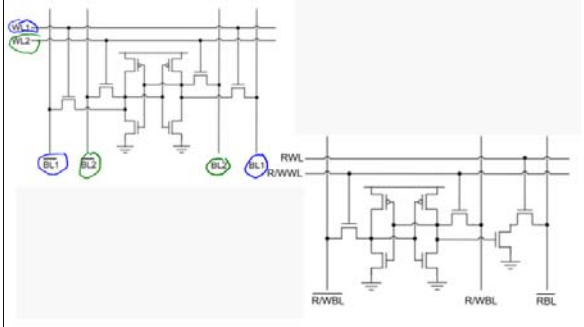
Write Constraint



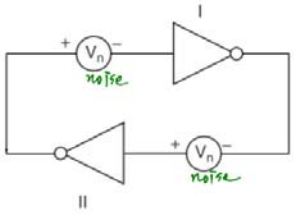
$$K_{access} > K_{PUN}$$

$$PR = \frac{W_3/L_3}{W_7/L_7} = \frac{W_6/L_6}{W_5/L_5} = \frac{PUN}{access}$$

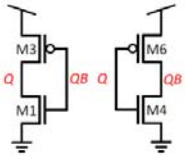
2-Port SRAM



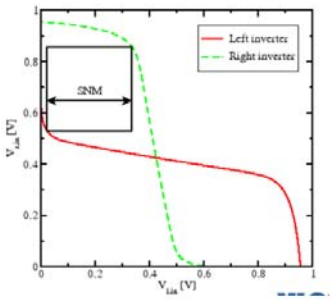
Static Noise Margin - Hold



Static Noise Margin - Hold

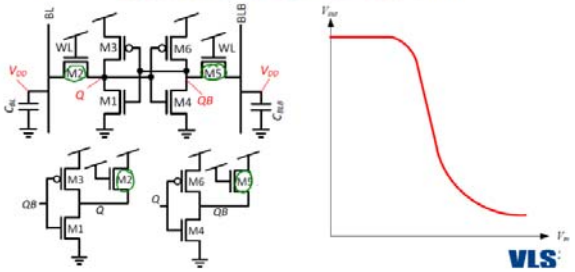


1. Plot both VTCs on the same graph
2. Find the maximum square that fits in the VTC.
3. The SNM is defined as the side of the maximum square.

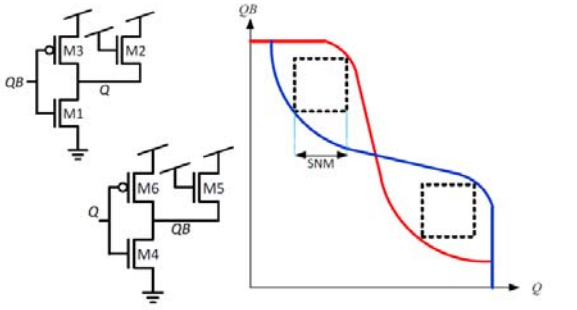


Static Noise Margin - Read

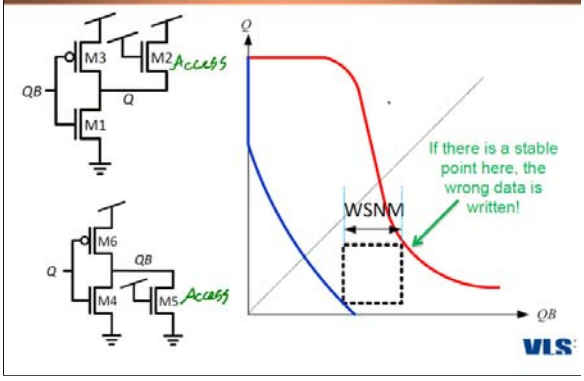
- What happens during Read?
  - » We can't ignore the access transistors anymore…



Static Noise Margin - Read

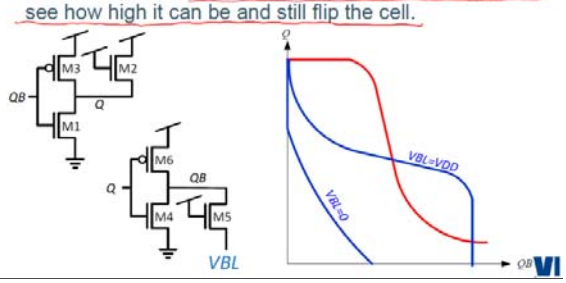


Static Noise Margin - Write



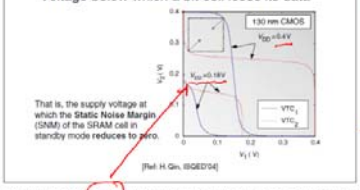
Alternative Write SNM Definition

- Write SNM depends on the cell's separatrix, therefore alternative definitions have been proposed.
- For example, add a DC Voltage (*VBL*) to the 0 bitline and see how high it can be and still flip the cell.



Limits to V_{DD} Scaling: DRV

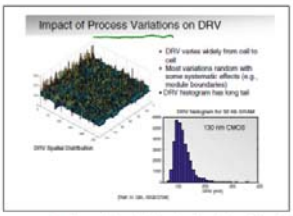
Data Retention Voltage (DRV):
Voltage below which a bit-cell loses its data



Slide 9.7
Given the effectiveness of voltage reduction in lowering the standby power of an SRAM memory, the ultimate question now is how much the supply voltage can safely be reduced. We define the minimum supply voltage for which an SRAM bit-cell (or an SRAM array) retains its data as the *Data Retention Voltage (DRV)*.

The butterfly plots shown on this slide illustrate how the noise margins of a 6T cell (with its access transistors turned off) collapse as the supply voltage is reduced. Due to the asymmetrical nature of a typical cell (caused by the dimensioning of the cell transistors as well as by variations), the SNM of the cell is determined by the upper lobe of the butterfly plot. Once the supply voltage reaches 180 mV, the SNM drops to zero and the stored value is lost. The cell becomes monostable at that point. In a purely symmetrical cell, the supply voltage could be lowered substantially more before the data is lost.

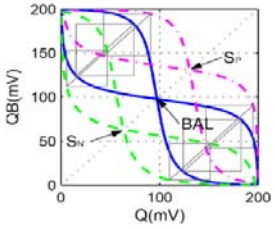
Impact of Process Variations on DRV



Slide 9.11
Given the high sensitivity of the DRV to the relative strengths of transistors, it should be no surprise that process variations have a major impact on the minimal operational voltage of an SRAM cell. Local variations in channel length and threshold voltages are the most important cause of DRV degradation. This is best demonstrated with some experimental results. This plot shows a 3-D rendition of the DRV of a 130nm 32Kb SRAM memory, with the x- and y-axis indicating the position of the cell in the array, and the z-axis denoting the value of the DRV. Local transistor variations seem to cause the largest DRV changes. Especially threshold variations play a major role.

SNM for Variability

- Modern process technologies suffer from parameter uncertainties (i.e., transistor parameters can vary over a large rage within a single chip)

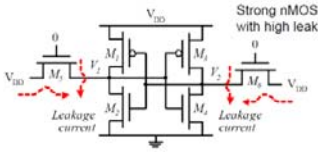


Stronger PMOS or NMOS (S_P,S_N) SNM even for typical cell

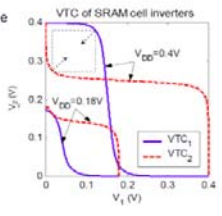
[Ref: J. Ryan, GLSVLSI'07]

SNM for Variability

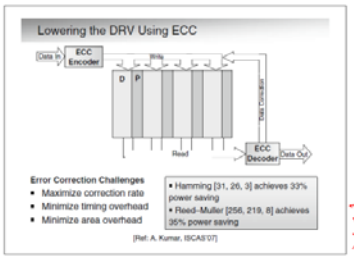
- At low voltages, even 'off' transistors play a role, especially when variability causes large leakage
  - » Degradation of Ion/Ioff ratio



Strong nMOS with high leakage



Leakage and within die variability limit minimum operating voltage (e.g., data retention voltage)

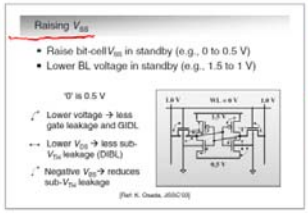


Slide 9.15

ECCs have been used in memories for a long time. Already in the 1970s, ECC had been proposed as a means to improve the yield of DRAMs. Similarly, error correction is extensively used in Flash memories to extend the number of write cycles. As indicated in the previous slides, another use of ECC is to enable "better-than-worst case", and lower the supply voltage during standby more aggressively.

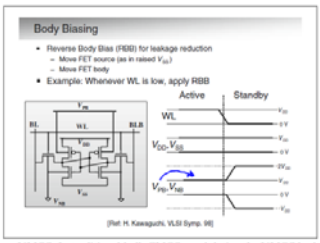
The basic concept behind error detection and correction is to add some redundancy to the information stored. For instance, in a Hamming (31, 26) code, five extra parity bits are added to the original 26 data bits, which allows for the correction of one erroneous bit (or the detection of two simultaneous errors). The incurred overhead in terms of extra storage is approximately 20%. Encoder and decoder units are needed as well, further adding to the area overhead. The leakage current reduction resulting from the ECC should be carefully weighed against the active and static power of the extra cells and components.

Yet, when all is considered, ECC yields substantial savings in standby power. Up to 33% in leakage power reduction can be obtained with Hamming codes. Reed–Muller codes perform even a bit better, but this comes at the cost of a more complex encoder/decoder and increased latency.



Slide 9.19

All standby power reduction techniques discussed so far are based on lowering the V_{DD}. An alternative approach is to raise the ground node of the bit-cells, V_{SS}. This approach decreases V_{DS} across a number of transistors, which lowers sub-threshold conduction (due to DIBL) as well as the GIDL effect. Furthermore, for bulk NMOS devices, the higher V_{SS} causes a negative V_{BS} that increases



Slide 9.20

Another option is to intentionally apply reverse body biasing (RBB) to the transistors in the cell during standby mode. Again, an increase in threshold voltage translates into an exponential decrease in sub-threshold drain–source leakage current, which makes it a powerful tool for lowering standby currents.

To induce RBB, you can either raise the source voltage (as in raised-V_{SS} approach of Slide 9.19) or lower the body voltage for an NMOS. In traditional bulk CMOS, modulating the NMOS body node means driving the full capacitance of the P-type substrate. Transitioning in and out of standby mode hence comes with a substantial power overhead. Changing the body voltage of the PMOS is relatively easier because of the smaller-granularity control offered by the N-well. Many bulk technologies now offer a triple-well option that allows for the placement of NMOS transistors in a P-well nested inside an N-well. This option makes adjustable RBB for standby mode more attractive, but the energy involved in changing the voltage of the wells must still be considered.

This slide shows an RBB scheme that raises and lowers the PMOS and NMOS bulk voltages, respectively, whenever a row is not accessed. The advantage of this approach is that it operates at a low level of granularity (row-level), in contrast to all techniques discussed previously, which work on a per-block level. In general, at most a single row of a memory module is accessed at any given time. The penalty is an increase in read and write access times.

Sense Amplifiers

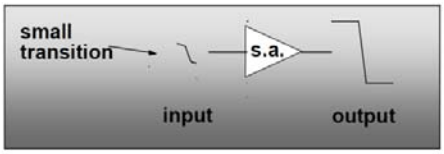
$$t_p = \frac{C \cdot \Delta V}{I_{av}}$$

make ΔV as small as possible

large

small

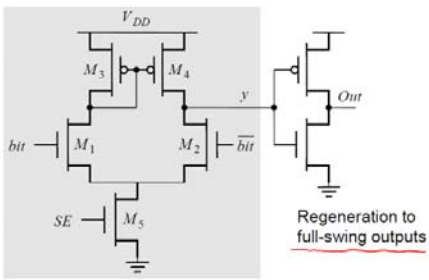
Idea: Use Sense Amplifer



Differential Sense Amplifier

Basic differential amplifier: Current source with two differential transistors

Disable when not needed to save power

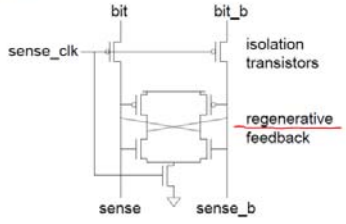


Regeneration to full-swing outputs

Directly applicable to SRAMs

Clocked Sense Amplifier

- Clocked sense amp saves power
- Requires sense_clk after enough bitline swing
- Isolation transistors cut off large bitline capacitance



Summary and Perspectives

- SRAM standby power is leakage-dominated
- Voltage knobs are effective to lower power
- Adaptive schemes must account for variation to allow outlying cells to function
- Combined schemes are most promising
  - e.g., Voltage scaling and ECC
- Important to assess overhead!
  - Need for exploration and optimization framework, in the style we have defined for logic

Slide 9.25

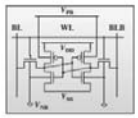
In summary, SRAM leakage power is a dominant component of the overall standby power consumption in many SoCs and general-purpose processing devices. For components that operate at low duty cycles, it is often THE most important source of power consumption. In this chapter, we have established that the most effective knobs in lowering leakage power are the various voltages that drive the bit-cells. However, these voltages must be manipulated carefully so that data preservation is not endangered.

As with active operation, the large number of small transistors in an embedded SRAM means that the far tails of power and functionality distributions drive the design. This means that any worst-case or adaptive schemes must account for the outliers on the distributions to preserve proper SRAM functionality. The most promising schemes for leakage reduction combine several different voltage-scaling approaches (selected from the set of V_{TH}, V_{DD}, V_{SS}, and well and periphery voltages) along with architectural changes (e.g., ECC). In all of these approaches, the overhead requires careful attention to ensure that the overall leakage savings are worth the extra cost in area, performance, or overhead power.

All this having been said, one cannot escape the notion that some more dramatic steps may be needed to improve the long-term perspectives of on-chip memory. Non-volatile memory structures that are compatible with logic processes and that do not require high voltages present a promising venue. Their non-volatile nature effectively eliminates the standby power concern. However, their write and (sometimes) read access times are substantially longer than what can be obtained with SRAMs. It is worth keeping an eye on the multitude of cell structures that are currently trying to make their way out of the research labs.

Combining Raised V_{SS} and RBB

Supply	Active (V)	Standby (V)
V_{DD}	1.0	1.78
V_{DD}	1.0	1.6
V_{SS}	0.0	0.65
V_{NW}	0.0	0.6



28X savings in standby power reported

[Ref: I. Clark, TVLSI'04]

Slide 9.22

Similarly we can combine the raised-V_{SS} approach with RBB. During standby, the raised-V_{SS} node reduces the effective supply voltage of the cell, while providing RBB for the NMOS transistors. A raised N-well voltage provides RBB to the PMOS devices. The advantage of this approach is that a triple-well technology is not required.