Static Random Access Memories (SRAM)

Birdy Amrutur
amrutur@hpl.hp.com

Key features of SRAMs

Holds data statically:

As long as power is supplied to the chip, the data remains.

Data randomly accessible:

ReadData = Memory[Address]

Memory[Address] = WriteData

SRAMs Used as:

Embedded memory, e.g.:

- **First and second level caches in processors**
- **Data buffers in various DSP chips**

Standalone SRAMs:

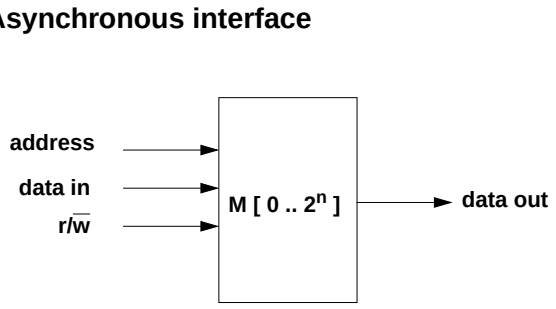
- **Caches in computer systems**
- **Main memory in low power applications**

Typical sizes:

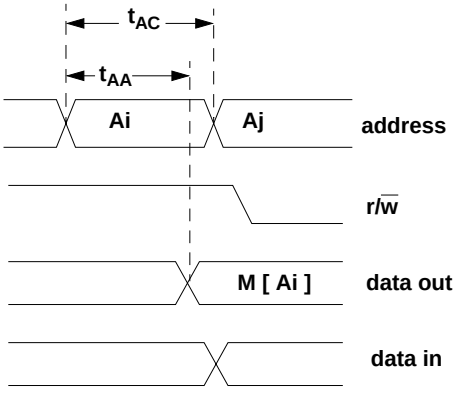
- **Embedded: upto 1Mbit**
- **Standalone: upto 16Mbit**

The system level view of SRAMs

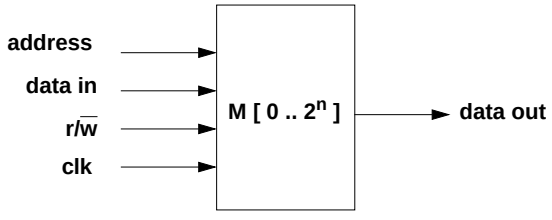
Asynchronous interface



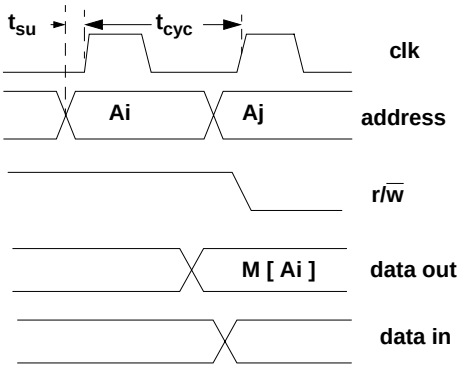
Used for stand alone SRAM chips



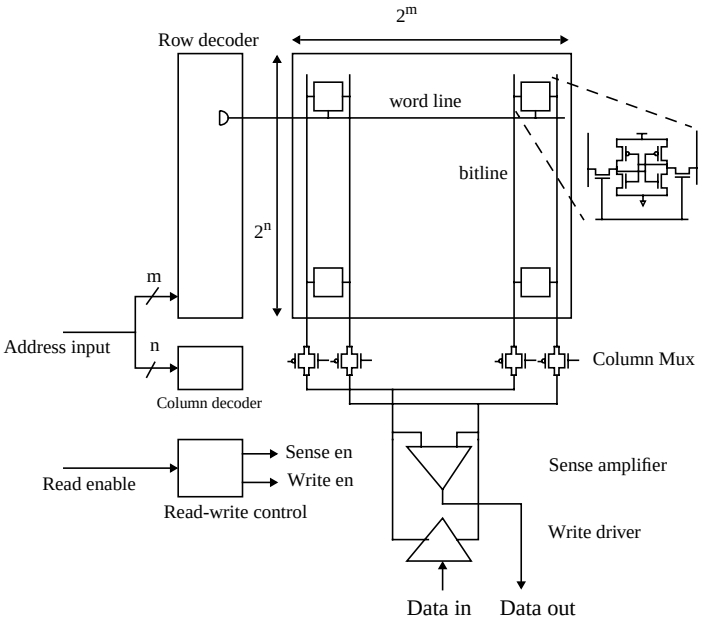
Synchronous interface



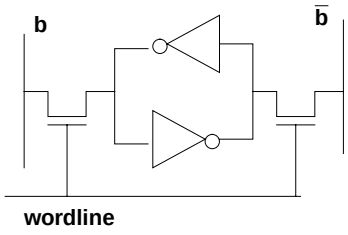
Used for embedded and standalone SRAMs



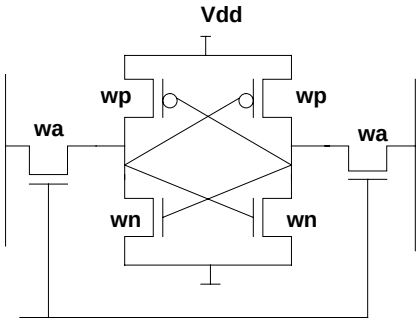
SRAM Architecture



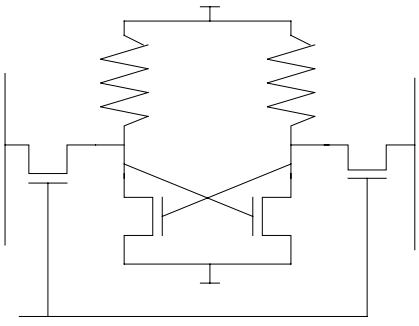
CMOS SRAM cell



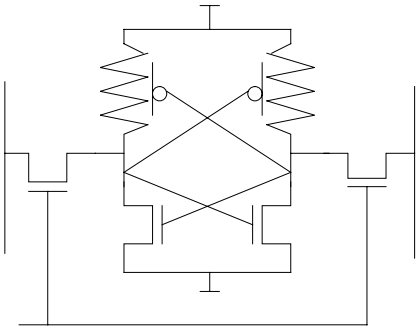
a) Static cell



b) 6T CMOS cell

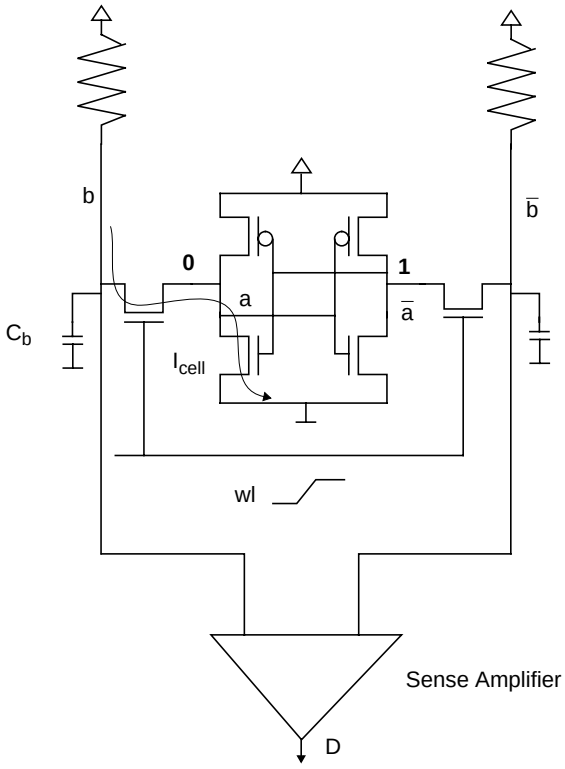


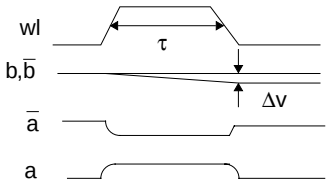
c) 4T poly-R cell



d) 6T poly-PMOS cell

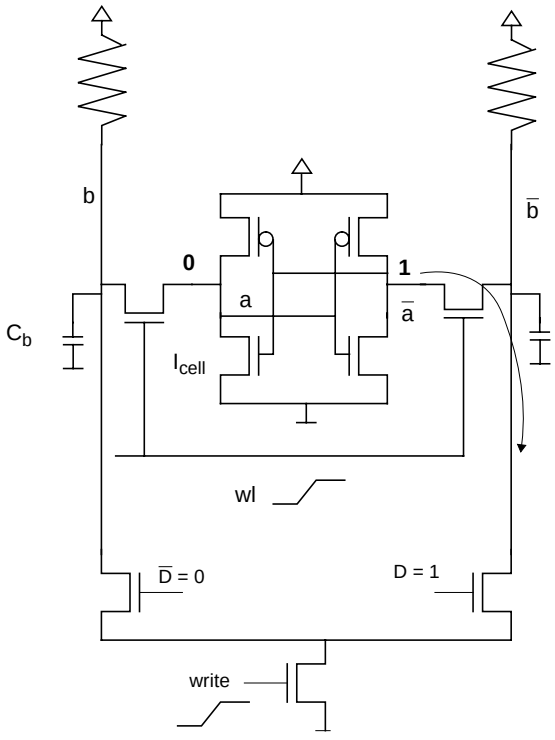
Reading a cell

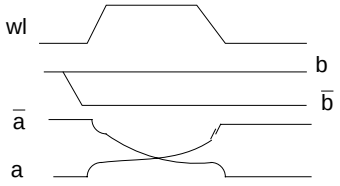




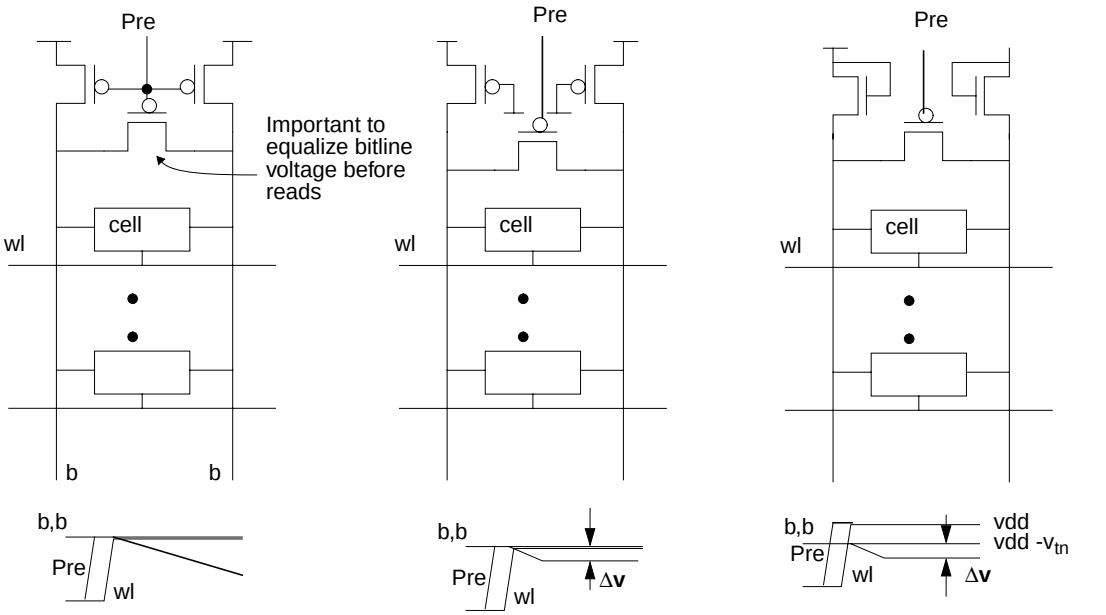
$$\Delta v = \frac{I_{cell} * \tau}{C_b}$$

Writing a cell





Bitline precharge and load



Precharged to Vdd .
Precharge shut off during reads and writes

Use with latch style sense amps

Bitline voltage clamped during reads.

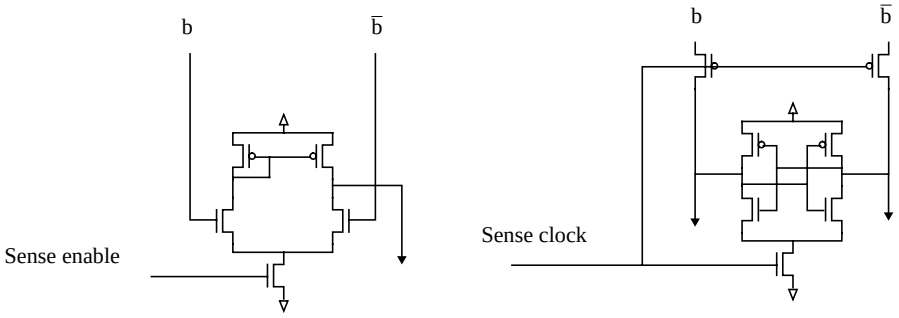
Use with current sense amps

Precharge to an Nmos threshold below supply

Useful with current mirror sense amps
(Cant operate with low supply)

Sense amplifiers

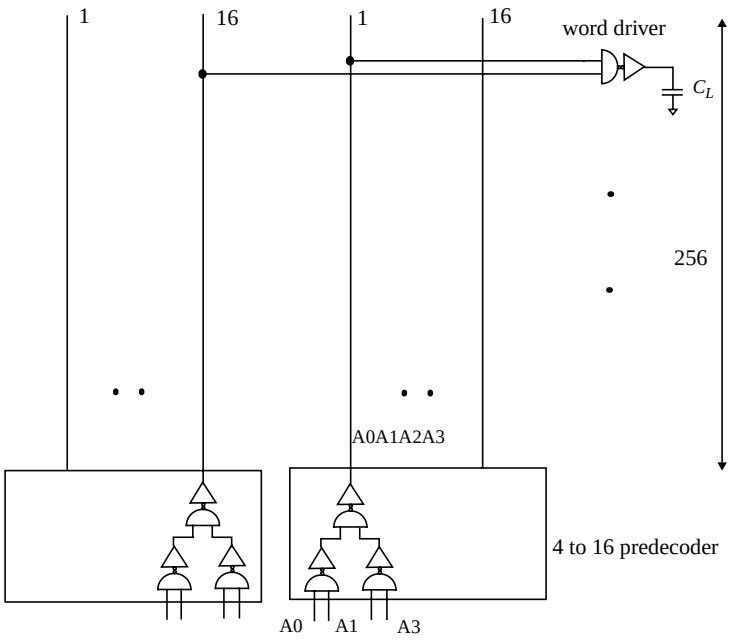
Need to amplify input bitline swing of ~100mV to full digital levels.



Current mirror amplifier

Latch type amplifier

Decoders



Logically an n-input AND function.

Enables the “Random Access” portion of RAMs.

Row decoder in the critical path of the SRAM access.

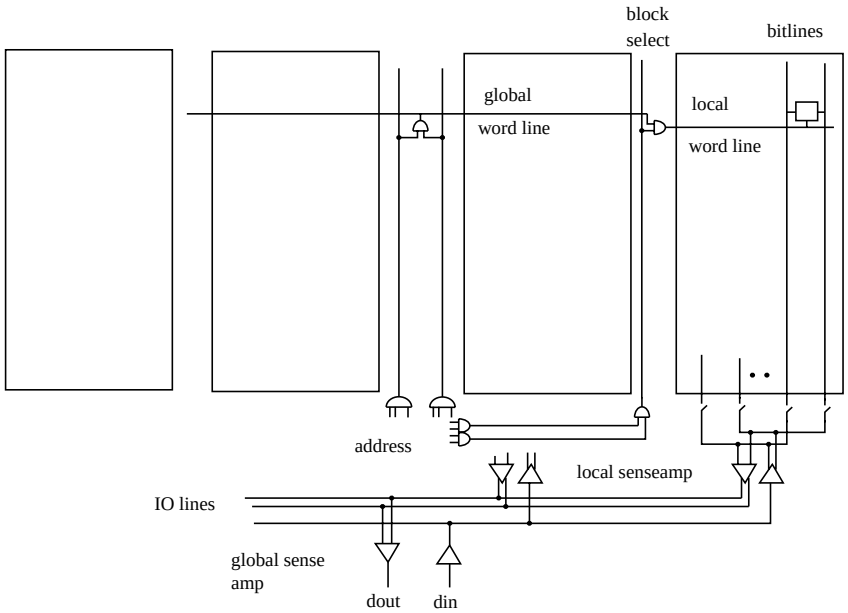
Could contribute upto 40% of the delay and power

Large decoders implemented hierarchically.

An 8 to 256 decoder

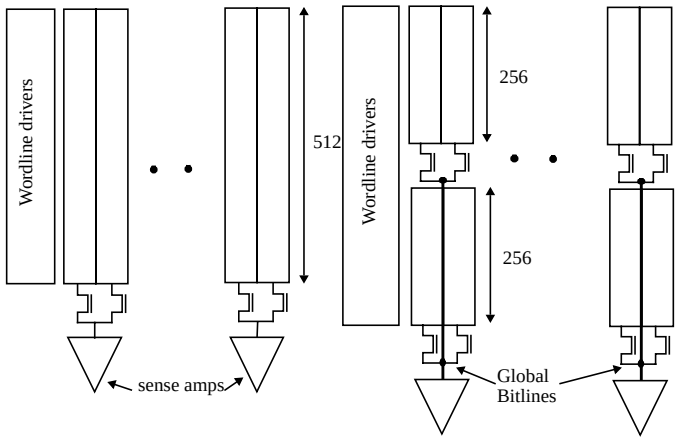
Sram Partitioning

Divided word line Architecture



Use higher level metal for global word lines

Bitline partitioning



Single Level Mux

Two Level Mux

Use higher level metal for global bitlines

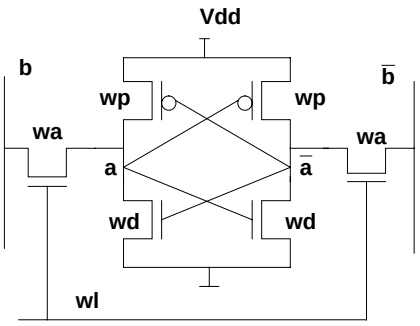
Partioning summary

Partioning involves a trade off between area, power and speed

For high speed designs, use short blocks(e.g 64 rows x 128 columns)
Keep local bitline heights small

For low power designs use tall narrow blocks (e.g 256 rows x 64 columns)
Keep the number of columns same as the access width to minimize wasted power

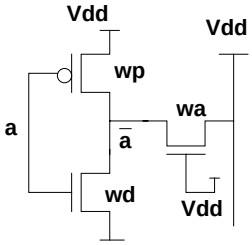
CMOS SRAM cell design-1

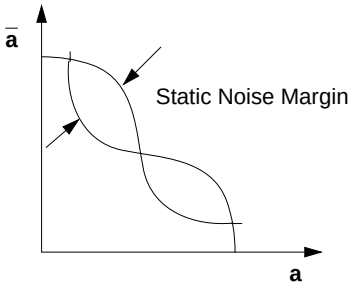


Problem: Find wa, wd, wp such that
1) minimize cell area
2) obtain good read and write cell margins
3) good soft error immunity
4) good cell read current

in that order

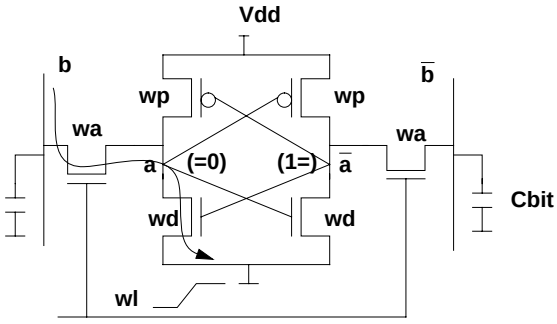
Conflicting goals!





CMOS SRAM cell design-2

Read Stability



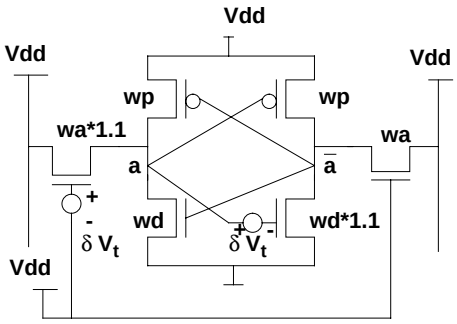
Usually Cbit is very big compared to internal node capacitances

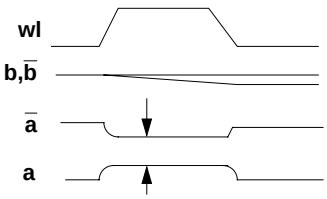
Cell will flip if the “0” node bounces high enough to cause the “1” node to discharge

Simulate for worst case scenario with threshold and size mismatches in the cell which aids in flipping

To obtain good stability :

$$\beta = \frac{wd * (Vdd - Vtn)^2}{wa * (Vdd - Vtn)^2} = \frac{wd}{wa} > 2.5$$

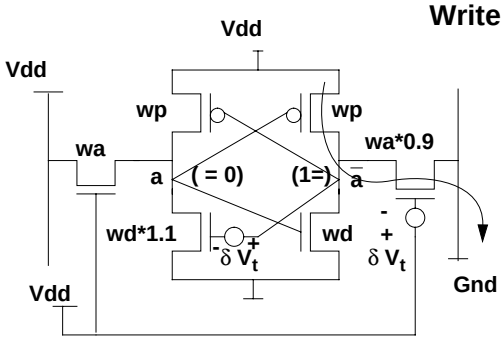




Dynamic read noise margin

CMOS SRAM cell design-3

Write Stability

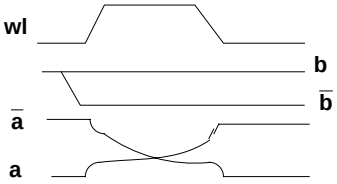


Cell is written by discharging the “1” node low. This causes the “0” node to charge up.

Need to ensure that

$wa * \mu n*(Vdd-Vtn)2 > wp*\mu p*(Vdd-Vtp)2$

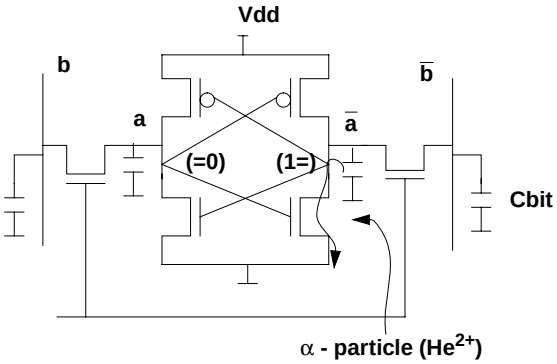
Hence use minimum sized pMOS load devices.



Dynamic write noise margin

CMOS SRAM cell design-4

Soft Error Immunity



α–particle can create electron-hole pairs in the channel of a “off” transistor.

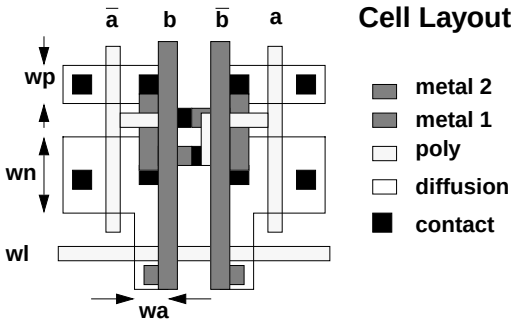
This causes the “off” transistor to leak away some of the stored charge.

Two solutions to the problem:

Increase stored charge (> 20fC) - use larger transistors (more capacitance), bigger voltages

Use system level solution like redundancy bits for error detection and correction

Cell Layout



Simple layout in MOSIS design rule. Cell area is 66.5 μm2 (1064λ2) in 0.5μm CMOS

Advanced processes with tighter rules + clever non-manhattan (non-right angle) layouts reduce cell area significantly (typically ~650λ2)

Poly-R and poly-PMOS cells have 2-3 times smaller area as the load devices can be laid out on top of the nMOS transistors