Chapter 16

Memory Circuits

- Array Architectures
  - Sensing Basics (NSA, Open Array, PSA, Refresh)
- Folded Array
  - DRAM Layout
- Chip Organization
- Peripheral Circuits (Sense Amps)
  - Kickback, Clock Feedthrough, Memory, Current Draw
  - Contention Current, Removing SA Memory
  - Reducing Kickback Noise, Increasing Input Range
- Peripheral Circuits (Row/Col Decoders)
  - Global, Local Decode; Reducing Layout
- Peripheral Circuits (Row Drivers)
- Memory Cells
  - SRAM, ROM, FG (E², FLASH)
**ARRAY ARCHITECTURE**

- **DESCRIPTION**
  - MEM CELL AT INTERSECTION R, C
  - TERMINOLOGY
    - BL – BIT LINE
    - WL – WORD LINE
    - R/W – READ/WRITE
  - ROW ADDRESSES
    - LATCHED →
    - DECODED →
    - BUFFERED → H WHEN SELECTED
    - DRIVES ENTIRE WL CAPACITANCE
    - WHAT CAUSES WL CAP?
  - COL ADDRESSES
    - DECODE WHICH BITS OF WORD R/W
    - SENSE AMP PER BIT

---

**Figure 16.1** Block diagram of random access memory.
**ARRAY ARCHITECTURE**

- **Parasitics, Loading**

  - **BL Metal Capacitance**
    - \( C_{BL} = \text{AREA} \times \text{CAP/AREA} \)
    - \( C_{BL} \approx (0.1\text{um}) \times (100\text{um}) \times (100\text{aF/um}^2) \)
    - \( C_{BL} \approx 1\text{fF} \)
    - WHERE DOES 100aF COME FROM?

  - **BL Depletion Capacitance**
    - MOSFET S/D CAP ~ 0.4fF
    - \( C_{BL} \approx 0.4\text{fF} \times 250 \approx 100\text{fF} \)
    - WHAT IS 250?

---

**Figure 16.2** The parasitic capacitance of a bitline to ground (substrate).

**Figure 16.3** How a bit line is loaded with implants (depeletion capacitance).
SENSING BASICS

- **INTRODUCTION**
  - ROW ACCESSED, WL GOES HIGH
  - CHARGE ON BL
    - EXACTLY HOW CHG APPEAR LATER
    - BL IS A CAP, VOLTAGE ON IT, CHGS
    - $\Delta V_{\text{BIT}} \approx 50\text{mV}$

- **NSA (NMOS SENSE AMP)**
  - WANT IMBALANCE IN M1/M2 DRIVE
  - SENSE IS SEEN AT BIT LINE
  - M1, M2 INV BASED LATCH
  - $V_{\text{SENSE}}$ IS ON THE DRAIN OF M1
  - PRE-CHRG M1, M2 DRN VOLT SAME
  - $\text{SENSE}_N$ GOES H
  - $\text{NLAT}$ GOES TO GROUND
  - M2 GOES TO GND $\rightarrow$ M1 GATE GND
  - LOADING NOT BALANCED M1, M2
**SENSING BASICS**

- Balance load on both sides
- Opens like a book
- AKA Butterfly

**OPEN ARRAY**

- VDD=1, $C_{\text{COL}}=1\,\text{fF}$
- BL equilibrated (AKA Prechg)
- BL shorted, Charge=VDD/2
- EQ=H, WL=L (not shown)

**NSA WITH EQUILIBRATION**

- VDD=1, $C_{\text{COL}}=1\,\text{fF}$
- BL equilibrated (AKA Prechg)
- BL shorted, Charge=VDD/2
- EQ=H, WL=L (not shown)

Figure 16.6 How the NSA is placed between two memory arrays in the so-called open memory array architecture.

Figure 16.7 An NSA with equilibration circuitry and connections to two bit lines.

Figure 16.8 How the equilibrate circuitry operates.
Baker Ch. 16 Memory Circuits

**SENSING BASICS**

- **Vfinal**
  - **Vbit**
  - **C_{mbit}
  - **C_{col}
  - **VDD/2
  - **NLAT
  - **MPD
  - **ME1
  - **ME2
  - **ME3

**DRAM (1T1C)**

- **ONE TRANSISTOR, ONE CAPACITOR**

- **WL = VDD + VT_N (OVERCOME DROP)**
- **C_{BIT} \sim 20fF (STORED CHG V_{BIT})**
- **BEFORE ACCESS XTOR ON:**
  - \( Q_{TOT} = C_{BIT} V_{BIT} + VDD/2*C_{COL} \)
- **AFTER ACCESS XTOR ON:**
  - **CAP VOLT EQUAL, Q IS CONST**
  - **V_{FINAL} (C_{BIT} + C_{COL}) [CHG AFTER] = C_{BIT} V_{BIT} + (VDD/2) * C_{COL} [CHG BEFORE]**
  - \( V_{FINAL} = \frac{\{C_{BIT} V_{BIT} + (VDD/2) * C_{COL}\}}{\{(C_{BIT} + C_{COL})}\} \)
  - **VDD=1, C_{BIT}=20fF, C_{COL}=100fF**
  - \( \Delta V_{BIT} = V_{FINAL} - VDD/2 \sim 83mV**
  - **NECESSARY S/A RESOLUTION**

**Figure 16.9** The one-transistor, one-capacitor (1T1C) DRAM memory cell.

**Figure 16.10** The connection of the NSA to the memory arrays.
**SENSING BASICS**

- A word line in array0
- $C_{mbit}$
- Other columns in array0.

**SENSING**

- WL in REF ARRAY AT GND
- WL in SENSE ARRAY = VDD + $V_{tn}$
- Cbit = 0 (READING A ZERO)
- SENSE_N HIGH

**Figure 16.10** The connection of the NSA to the memory arrays.

**Figure 16.11** N-sense amp's operation.

**Figure 16.12** Reading out a "1" from the cell in array0.
**SENSING BASICS**

- A word line in array 0
- Parasitic bit line capacitance not shown.
- \( C_{mbit} \)
- \( \text{WL=GND} \) A word line in array 1
- \( \text{Eq} \)
- \( N\text{-sense amplifier} \)
- \( \text{ME1, ME2, ME3} \)
- \( \text{VDD/2} \)
- \( \text{M1, M2, MPD} \)
- \( \text{NLAT} \)

**PSA SENSING**

- **READING 1 GOING TO VDD**
- **DATA NEEDS TO BE REFRESHED**
  - S/A TAKES BL TO GND OR VDD
  - NOT SHOWN HERE

**Figure 16.10** The connection of the NSA to the memory arrays.

**Figure 16.13** Schematic diagram of a PMOS sense amplifier.

**‘0’ PULLED DN ‘1’ PULLED UP**

**Figure 16.14** How the PSA pulls the bit line from array 0 high.
DRAM DESIGN

- Bit line from array 0
- Bit line from array 1
- Different shapes of noise

ARRAY 1

ARRAY ARCHITECTURE

- OPEN/BUTTERFLY ARCHITECTURE
  - MOST DENSE
  - NOISE AN ISSUE
  - BL ACT DIFFERENTLY
  - SA READS DATA INCORRECTLY

- FOLDED ARRAY ARCHITECTURE
  - TAKE A1 BL AND FOLD INTO A0
  - MEM BIT EVERY OTHER ROW/COL
  - LOADING SLIGHTLY DIFFERENT

Figure 16.15 Different amplitudes of coupled noise into physically separated bit line.

Figure 16.16 The folded array is formed by taking the open array architecture (open book) topology seen in Fig. 16.6 and "closing the book," that is, folding array 1 on top of array 0. Note that the bold lines indicate the bitlines from array 1 in the newly formed array.

Figure 16.17 How a memory cell is located at every other intersection of a row line and a column line in a folded-area architecture.


### DRAM DESIGN

- **Bit line pitch**
- **Contact**
- **Capacitor**

**Figure 16.18** Two mbits sharing a contact to the bit line.

### BIT LAYOUT

- **BL CONTACT SHARED**
- **WL SILICIDED POLY**
- **DELAY DUE TO WL**
  - \( \tau_d = R \times C \)
  - \( \#COL \times \{ R_{GATE} \} \)
  - \( \#COL \times \{ C_{FETOX} + C_{PAR} \} \)
  - \( C_{PAR} \) is O/L cap, etc.
  - \( C_{FETOX} = 400aF, C_{PAR} = 100aF \)
  - \( R_{GATE} = 4 \Omega, \#COL=512 \)
  - \( \tau_d = (512)(4) \times (512)(400E-18+100E-18) \)
  - \( \tau_d = (2048 \Omega)(2.56E-13 F) \)
  - \( Q=CV \rightarrow coul=F \times V \rightarrow F=c/V \)
  - \( V=IR \rightarrow c/s \Omega \rightarrow \Omega = Vs/c \)
  - \( \tau_d = \{2048 Vs/c\} \times \{2.56E-13 c/V\} = 524ps \)
  - **50% POINT FOR Tdelay, 6X TO VDD**

- **PERIODIC ARRAY**
  - PITCH-DISTANCE TO COMMON
  - **F – FEATURE SIZE**
  - **F = PITCH/2**

**Figure 16.19** Layout of mbit used in an open bit line configuration.
Baker Ch. 16 Memory Circuits

**DRAM DESIGN**

- **BIT LAYOUT**
  - **FOLDED ARCHITECTURE**
    - ADDITIONAL POLY INTERCONNECT
    - CELL SIZE $8F^2$ vs. $6F^2$
    - CELL SIZE IS #1, NO QUESTION
  - **TRENCH**
    - DIFFICULT, HIGHER DENSITY, SCALES
  - **BURIED**
    - EASIER, MORE PARASITICS, NO SCALE

---

**Figure 16.20** The mbit pair used for a folded architecture.

**Figure 16.21** Folded architecture layout and cell size.

**Figure 16.23** Cross-sectional view of a buried capacitor cell.
CHIP ORGANIZATION

<table>
<thead>
<tr>
<th>RD</th>
<th>256k</th>
<th>256k</th>
<th>256k</th>
<th>256k</th>
<th>256k</th>
<th>256k</th>
<th>256k</th>
<th>256k</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
<tr>
<td>RD</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
<td>256k</td>
</tr>
<tr>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
<td>SA</td>
</tr>
</tbody>
</table>

- **Row decoder and driver circuitry**
- **SA** Sense amplifier and column decoder circuitry.

**Figure 16.25** A 16-Mbit array block.

DESCRIPTION

- **INCREASING ROWS**
  - MORE BL CAP
  - POWER INCREASES
  - \[ P_{AVG} = \#SA \times C_{COL} \times (VDD/2)^2 \times f \]
  - 512 BL, WL PAIRS DRAM LIMIT (16.20)
  - \[ 512^2 = 256kB \]
- **ARRAY OF BLOCKS**
  - 1GB DRAM = 8k * 256kB BLOCKS
**PERIPHERAL CKTS (SA)**

- **SENSE AMP DESIGN**
  - CLK=L; MS3 OFF; MS1, MS2 ON
  - IN+, IN- < Vtp, SHUTS OFF MS1,2

---

**Figure 16.26** Clocked sense amplifier.

**Figure 16.27** Simulating the operation of the circuit in Fig. 16.26.
PERIPHERAL CKTS (SA)

- Long L inverters driving sense amplifier.

CLOCK FEEDTHROUGH
- Inputs are not ideal sources
- CLK=H @ 20ns; IN+, IN- "GLITCH"
- CLK signal cap path to input

KICKBACK NOISE
- Present when latch switches
- Noise can be limiting factor
- Have to use non-ideal sources

MEMORY

CURRENT DRAW

CONTENTION CURRENT

REMOVING SA MEMORY

REDUCING KICKBACK NOISE

INCREASING INPUT RANGE
PERIPHERAL CKTS (SA)

MEMORY
- ALL NODES NEED KNOWN STATE

CURRENT DRAW
- INPUTS DRIVE M3/M4
- M3/M4 SOURCE CURRENT \( \rightarrow \text{IN}+/\text{IN}^-\)
  - \(\text{V}_{\text{sg}} \approx 0.5\text{V}, \text{M3/M4 CONDUCTING}\)
- \(\text{I}_{\text{avg}} \approx 50\text{uA}, \text{VDD}\approx 50\text{mA FOR S/A}\)
- MINIMIZE CURRENT VDD\(\rightarrow\)GND
- 16.27, 16.28 SAME SIM? NOT SURE

FLOATING NODE
DYNAMIC STORAGE

VDD CURRENT OF F16.26, F16.27

Figure 16.26 Clocked sense amplifier.

Figure 16.27 Simulating the operation of the circuit in Fig. 16.26.

Figure 16.29 The current that flows in the sense amplifier in Fig. 16.26.
PERIPHERAL CKTS (SA)

- **VDD**
  - **VDD**

- **IN+**
  - **MS1 Out+**

- **IN-**
  - **MS2 Out-**

- **clock**
  - 0

- **Vss**

- **M3**

- **M4**

- **M1**

- **M2**

- **MS3 OFF**

**Contention Current**

- **CLK=L, M3, M4 Conducting**
- **When CLK L→H, M1,M2 On**
- **DC Path Power to Ground**
  - Metastable Condition
  - Want Imbalance to Occur In+/-
- **VDD-Vtp < IN < VDD, Not an Issue**
  - Why is this the case?

**Removing SA Memory**

- **All Nodes to Known State**
- **CLK=L, Outputs→VDD**
- **MS1, MS2 OFF**
- **M1, M2 Drain→GND**

---

**Figure 16.26** Clocked sense amplifier.

**Figure 16.30** Removing memory in a sense amplifier.
PERIPHERAL CKTS (SA)

Figure 16.31 Creating an imbalance in a sense amplifier.

CREATING AN IMBALANCE

- TWO CASES
  - MB1/MB2 TO DRAIN OF MS1/MS2
  - MB1/MB2 TO SOURCE OF MS1/MS2
- 1) DRAIN CONNECTION
  - CLK=L, MB1/2 DRAINS TO VDD
  - IF IN+/− LARGE, PATH VDD→GND
    - INDEPENDENT OF CLK
  - BENEFIT IS LARGE GAIN
    - SMALL INPUT Δ, LARGE OUTPUT
- 2) SOURCE CONNECTION
  - CLK=L, MB1/2 DRAINS @ GND
  - NO CURRENT DRAW
  - CLK=L→H, SMALLER GAIN

WHY HERE?

1) IN+/−

2) IN+/−
PERIPHERAL CKTS (SA)

- **CREATING AN IMBALANCE**
  - **INCR W/L MB1/2 → M1/M2 VGS △**
    - **PRO:** HIGHER GAIN
    - **CON:** INCR POWER
  - **USE LONG L**
    - **PRO:** REDUCE POWER
    - **CON:** DECR GAIN, SPEED
  - **POWER, GAIN, SPEED TRADEOFF**
  - **IMBALANCE WITH DECR POWER**
    - PRIOR TO SWITCH, VDD ON M1/2
    - M1/2 Vgs, MB1/2 Vds LARGE
    - IN+/− MUST BE > Vtn
    - IN+/− ISOLATED FROM LATCH BY MB*

INCORRECTLY INCR RANGE

- **LAST EXAMPLE, INPUTS > Vtn**
- **HERE, Vtn < INPUTS < VDD-Vtp**
- **MB1-4 NEED TO BE ON FOR READ**
- **IF MB1/2 ON (STRONG 1), MB3/4 OFF**
- **MB3/4 ON WHEN M3/4 VG=VDD-Vtp**
- **LARGE IN+/− △ OK, BUT SMALL NOT**
- **IN+/− SWITCHES ON M1/2 MISMATCH**
PERIPHERAL CKTS (SA)

CORRECTLY INCRR RANGE

- SA WORKS FINE FOR IN>Vtn
- NEED SOMETHING BELOW Vtn
- MB3-8 ADDED TO WORK < Vtn
- MB3/4 FOR SINKING M1/2 CURRENT
- MB5-8 LEVEL SHIFT INPUTS < Vtn
  - IN < Vtn, MB7/8 ON, MB1/2 OFF
  - IN+/− CAUSES ΔVds IN MB7/8
  - MB5/6 HAVE DIFFERENT CURRENT
  - DIFF VDS IN MB5/6 → GATE MB3/4
  - IF IN+/− ABOVE Vtn, MB1, MB2 FIGHT IMBALANCE, BUT PWR TRADEOFF
  - PROBLEM IN MB8→6, MB7→5?
  - INCR L TO REDUCE DC CURRENT

Figure 16.34 Rail-to-rail input signal range. See also Fig. 27.16.
PERIPHERAL CKTS (SA)

- **PERIPHERAL CKTS (SA)**

![Diagram of PERIPHERAL CKTS (SA)](image)

- **REDUCE POWER CONSUMPTION**
  - SA OUTPUTS = H WHEN CLK = L
  - WANT H ON RISING EDGE OF CLK
  - NAND LATCH CHG ONLY ON CLK
  - NAND CURRENT DRAW MORE
  - HOWEVER, NO "DC" CURRENT

![Plot of current consumption](image)

~30uA
"DC"

Figure 16.32 Reducing power in a sense amplifier.

![Diagram of SR latch](image)

Figure 16.35 Adding an SR latch so output only changes on rising edge of the clock.

![Plot of current consumption](image)

Figure 16.36 Regenerating the plot seen in Fig. 16.29 with the sense amplifier from Fig. 16.35. The current includes the switching current from the NAND gates.
PERIPHERAL CKTS (SA)

Long L inverters driving sense amplifier.

ADD NAND LATCH

REDUCING KICKBACK NOISE

50mV

5mV

Figure 16.34 Rail-to-rail input signal range. See also Fig. 27.16.

Figure 16.37 Kickback noise.

Figure 16.27 Simulating the operation of the circuit in Fig. 16.26.

Figure 16.38 The operation of the circuit in Fig. 16.35.
PERI CKTS (ROW/COL)

1Gbits x 1
M=N=15

64 SUBARRAYS

ROW/COL DECODE

- N x M ARRAY ADDRESSES
  - 1 ROW HIGH, (2^N – 1) LOW
  - WORD COL HIGH, (2^M – WORD) LOW
  - WL = VDD + Vtn
  - BL = SA BIAS

- ADD PINS CAN BE MULTIPLEXED
  - STROBE IN RADD, CADD VS. TIME

- 1 Gbit CONFIGURATIONS
  - 1G WORDS x 1 (2^30 ADDRESSES)
  - 256M WORDS x 4
  - 16M WORDS x 16
  - 8M WORDS x 32
  - ADDR SPLIT INTO GLOBAL, LOCAL

- GLOBAL DEC EXAMPLE 1G x 1
  - 2^15 = 32,768 ROW LINES
  - 2^15 = 32,768 COL LINES
  - SUB ARRAY 256kb → 512 WL, 512 BL
  - → 4,096 ARRAYS x 256k BITS/ARRAY
  - 1 RADDR H → 32k COLS AVAILABLE
    - ROW AKA PAGE
    - FLIP PAGES BUT MORE PWR

Figure 16.39 Detailed block diagram of a RAM.
PERI CKTS (ROW/COL)

- **ROW/COL DECODE**
  - LOCAL DECODE
    - PRO: LESS POWER
    - CON: MORE ROUTING, AREA
  - COMPROMISE, GLOBAL+LOCAL
    - GLOBAL CHOOSES SUBARRAY
      - 64 ARRAYS $\rightarrow$ 6 ADDR LINES
    - LOCAL WITHIN SUBARRAY
      - $15 - 6 = 9$ ADDR SUBARRAY
      - $2^9 = 512$ ROW/COL $\rightarrow$ 256kb
    - WHAT IS THE PENALTY?
      - COL DEC PASS XTOR DATA I/O
      - NEED TO BE PITCH MATCHED

---

### ADDR PHYSICAL BITS

<table>
<thead>
<tr>
<th>ADDR</th>
<th>PHYSICAL BITS</th>
<th>1bit</th>
<th>2bit</th>
<th>4bit</th>
<th>8bit</th>
<th>16bit</th>
<th>32bit</th>
<th>64bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>10</td>
<td>1,024</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2,048</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4,096</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>8,192</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>16,384</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>32,768</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>65,536</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>131,072</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>262,144</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>524,288</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1,048,576</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>2,097,152</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>4,194,304</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>8,388,609</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>16,777,216</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>33,554,432</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>67,108,864</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>134,217,728</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>268,435,456</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>536,870,912</td>
<td>1k</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1,073,741,824</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>2,147,483,648</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>4,294,967,296</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>8,589,934,592</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>17,179,869,184</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>34,359,738,368</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>68,719,476,736</td>
<td>1G</td>
<td>64</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PERI CKTS (ROW/COL)

Decoder output

To row driver

Reduced switching point voltage

VDD

Long L

Figure 16.43 A tree decoder used to reduce layout area.

DECODER LAYOUT

- PASS XTOR $\rightarrow$ REDUCED AREA
  - FLOATING NODES
  - LONG L XTOR PULLS LOW
- PE CIRCUIT $\rightarrow$ REDUCED AREA
  - LONG L KEEPER, NO DYN NODE

Figure 16.44 3-bit PE decoder.
PERI CKTS (ROW DRIVERS)

DESCRIPTION

- SIMPLE INVERTER NOT USEFUL
  - INPUT = 0V, OUTPUT = 1.5V, OK
  - INPUT = 1V, OUTPUT = ?
    - BOTH NMOS AND PMOS ARE ON
- USE FEEDBACK
  - DEC=0, M1 OFF, M2 ON
    - CAUSES M3 ON → M4 OFF
  - DEC=1, M1 ON, M2 OFF
    - CAUSES M4 ON → M3 OFF
  - CONTENTION DURING SWITCH
    - CLOCK THE VDDP AFTER

Figure 16.45 Problems with using an inverter for a row driver.

Figure 16.46 A CMOS word line driver.

Figure 16.47 Using a clocked VDDP to reduce contention current in a row line driver.
Memory Cells

- **Memory Cells**

- **SRAM**
  - STATIC
    - POWER ON, RETAINS MEMORY
  - CROSS COUPLED LATCH
    - DRAIN GOES TO GATE
  - SIZING ACCESS TRANSISTOR
    - WEAK $\rightarrow$ WRITE, BIT WON'T FLIP
    - STRONG $\rightarrow$ LAYOUT LARGE
  - $\beta$ RATIO
    - RATIO OF PASS TO LATCH CURRENT

*Figure 16.48 The six-transistor SRAM memory cell.*
Memory Cells
Memory Cells

- Bit Line bar
- GROUND
- NLATCH
- NPASS
- Bit Line
- POWER
- PLATCH
- Cross Coupled Node
- WL (ct not shown)
Memory Cells

ROM

- SIMPLE, SOMEWHAT DENSE, CHEAP
- STORE INSTRUCTIONS, LOOKUPS
- INITIALLY UNPROGRAMMED IN FAB
  - DRAIN TO COLUMN LINE BROKEN
    - “ON-THE-FLY”: LASER, HOT \( e \)
    - PROGRAM: VIA MASK, $$ IF DONE > 1$$

Figure 16.50 A ROM memory array.

Figure 16.51 (a) n-channel MOSFET at the intersection of every column and row line and (b) eliminating the connection between the drain and column line to program the ROM.
Memory Cells

- **FG (E², FLASH)**
  - TWO DIFFERENT LAYERS OF POLY
    - DUAL-POLY TECHNOLOGY
    - VERY EXPENSIVE
    - CANNOT BE DONE IN CMOS
    - CONTROL GATE, FLOATING GATE
  - OXIDE THICKNESSES DIFFERENT
    - P1 TO CHANNEL THIN
    - P2 TO P1 THICKER
  - PROGRAMMED VS. ERASED VS. UV
    - STATE DEPENDS ON PRESENCE OF e-
    - "IDEAL" UV=EE, NEVER HAPPENS
    - ARRAY DISTRIBUTION, SLOW/FAST BITS
  - SHIFT OCCURS IN VT:
    - \( V_{\text{thn,EE}} = -V_{\text{ms}} - 2V_{\text{fp}} + 2Q_{b}/C_{\text{ox}} \)
    - \( V_{\text{thn,PG}} = -V_{\text{ms}} - 2V_{\text{fp}} + 2\{ Q_{b} + Q_{\text{poly}} \}/C_{\text{ox}} \)

UV STATE

- **Figure 16.52** A floating gate MOSFET, its symbol and layout.

USUALLY THICKER OXIDE

- **Figure 16.54** Oxide capacitance estimation for calculating threshold voltage.
Memory Cells

- **READ**
  - UV
  - 0V
  - 3V
  - 3V
  - FLT
  - e-
  - e-
  - e-
  - 5V
  - 9V

- **PROGRAM**
  - FN PROGRAM
  - 0V
  - FLT
  - e-
  - e-
  - e-
  - 5.4V
  - 0V

- **ERASE**
  - SOURCE FN
  - 0V
  - FLT
  - e-
  - e-
  - -5.4V

<table>
<thead>
<tr>
<th>CG</th>
<th>S</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>L</td>
<td>G</td>
</tr>
<tr>
<td>FN EE</td>
<td>HN</td>
<td>F</td>
</tr>
<tr>
<td>S EE</td>
<td>HN</td>
<td>G</td>
</tr>
<tr>
<td>FN PG</td>
<td>HP</td>
<td>F</td>
</tr>
<tr>
<td>CH PG</td>
<td>M-H</td>
<td>G</td>
</tr>
</tbody>
</table>

**Erased state**

**Programmed state**

**Gate-source voltage**

**Drain current**

\[ I_{erased} \]

\[ I_{prog} \]

**Figure 16.64** Expanded view showing erased and programmed IV curves.
Memory Cells

Flash – NAND vs. NOR

- ARCHITECTURE
  - NAND IS SERIAL
  - NOR IS PARALLEL

- DENSITY
  - NAND DENSER
  - NOR LESS DENSE

- READING
  - NAND SLOW
  - NOR FAST

- PROGRAMMING
  - NAND FAST
  - NOR SLOW

- ERASE
  - NAND FAST
  - NOR SLOW

- CURRENT CONSUMPTION (EITHER)
  - CHE – LARGE
  - FNT – SMALL

Source: Toshiba
Memory Cells

NOR – Random Access

- WL0
- WL1
- WLn

BL0 - BL1 - BLn

Flash – NAND vs. NOR

NAND – Serial Access

- SELECT TOP
- SELECT BOTTOM
- SL per block

- RA0
- RA1
- RA2
- RA3

SOURCE: INTEGRATED CIRCUIT ENGINEERING CORP

BAKER
Memory Cells

Flash – NAND PROGRAMMING
- RA0 TO VERY HIGH POTENTIAL (VHV)
- RA1-3 TO HIGH POTENTIAL (VSEL)
- BL AT GND, RA0 GETS FNT
- NEXT COL DOES NOT GET FNT
  - NEED TO MAINTAIN > 7MV/cm

Flash – NAND READING
- SELECT XTOR AT 5V
- UNSELECTED WL AT 5V
- SELECTED WL AT 0V
- TOTAL CURRENT
  - IRA0 + IRA1 + IRA2 + IRA3
  - COMPARE TO KNOWN CURRENT
  - SERIAL READ
Memory Cells

**RELATIVE VOLTAGES (DEPENDS ON GOX)**

Table 16.1 Summarizing NAND Flash cell operation

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Erase</th>
<th>Program</th>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit line</td>
<td>Floating</td>
<td>0 V</td>
<td>High or low</td>
</tr>
<tr>
<td>select_top</td>
<td>0 V</td>
<td>20 V</td>
<td>5 V</td>
</tr>
<tr>
<td>RA0</td>
<td>0 V</td>
<td>20 V</td>
<td>0 V</td>
</tr>
<tr>
<td>RA1</td>
<td>0 V</td>
<td>5 V</td>
<td>5 V</td>
</tr>
<tr>
<td>RA2</td>
<td>0 V</td>
<td>5 V</td>
<td>5 V</td>
</tr>
<tr>
<td>RA3</td>
<td>0 V</td>
<td>5 V</td>
<td>5 V</td>
</tr>
<tr>
<td>select_bot</td>
<td>0 V</td>
<td>0 V (so the source of the cell floats)</td>
<td>5 V</td>
</tr>
<tr>
<td>n+ source</td>
<td>Floating</td>
<td>0 V</td>
<td>0 V</td>
</tr>
<tr>
<td>p+ well tie-down</td>
<td>20 V</td>
<td>0 V</td>
<td>0 V</td>
</tr>
</tbody>
</table>

**Comments**
- Erases entire array since well and word lines are common (Flash). The p-well at 20V will forward bias the n+ source and drain regions. This requires the bit line and n+ source float external to the array.
- Programming the RA0 cell. Bit lines of the cells not to be programmed, but on RA0, are driven to 7V to avoid FN. Unused word lines driven to 5V.
- Reading the contents of RA0. An average current is put into the bit line.

Figure 16.65 How gate current changes with gate voltage source grounded.

Figure 16.66 How gate current changes with gate voltage and drain is at VDD.

Figure 16.67 Erasing. Since electrons are being removed from the gate (they are tunneling through the gate oxide), we know the gate current will flow out of the device.