CMOS Digital Integrated Circuits

Chapter 10
Semiconductor Memories
Semiconductor Memory Types

Semiconductor Memories

Read/Write (R/W) Memory or Random Access Memory (RAM)
- Dynamic RAM (DRAM)
- Static RAM (SRAM)

Read-Only Memory (ROM)
- 1. Mask (Fuse) ROM
- 2. Programmable ROM (PROM)
- Erasable PROM (EPROM)
- Electrically Erasable PROM (EEPROM)
- 3. Flash Memory
- 4. Ferroelectric RAM (FRAM)
Semiconductor Memory Types (Cont.)

- **Design Issues**
  - **Area Efficiency of Memory Array:** number of stored data bits per unit area
  - **Memory Access Time:** the time required to store and/or retrieve a particular data bit.
  - **Static and Dynamic Power Consumption**

- **RAM: the stored data is volatile**
  - **DRAM**
    - A capacitor to store data, and a transistor to access the capacitor
    - Need refresh operation
    - Low cost, and high density ⇒ it is used for main memory
  - **SRAM**
    - Consists of a latch
    - Don’t need the refresh operation
    - High speed and low power consumption ⇒ it is mainly used for cache memory and memory in hand-held devices
Semiconductor Memory Types (Cont.)

- **ROM:** 1, nonvolatile memories
  2, only can access data, cannot to modify data
  3, lower cost: used for permanent memory in printers, fax, and game machines, and ID cards
  - *Mask ROM:* data are written during chip fabrication by a *photo mask*
  - *PROM:* data are written electrically after the chip is fabricated.
    » *Fuse ROM:* data cannot be erased and modified.
    » *EPROM and EEPROM:* data can be rewritten, but the number of subsequent re-write operations is limited to $10^4$-$10^5$.
      - *EPROM* uses ultraviolet rays which can penetrate through the crystal glass on package to erase whole data simultaneously.
      - *EEPROM* uses high electrical voltage to erase data in 8 bit units.
  - *Flash Memory:* similar to EEPROM
  - *FRAM:* utilizes the *hysteresis* characteristics of a capacitor to overcome the slow written operation of EEPROMs.
Random-Access Memory Array Organization

- Bit Lines($2^M$)
- Row Decoder Bits
- Word Decoder Bits
- Column Decoder Bits
- Memory Cell

Data Line Control Circuits

Column Decoder

(2^N x 2^M total)

CMOS Digital Integrated Circuits
Nonvolatile Memory

4Bit × 4Bit NOR-based ROM Array

- One word line “$R_i$” is activated by raising its voltage to $V_{DD}$
- Logic “1” is stored: Absent transistor
- Logic “0” is stored: Present transistor
- To reduce static power consumption, the pMOS can be driven by a periodic pre-charge signal.
Layout of Contact-Mask Programmable NOR ROM

- **“0” bit**: drain is connected to metal line via a metal-to-diffusion contact.
- **“1” bit**: omission the connect between drain and metal line.
- **To save silicon area**: the transistors on every two adjacent rows share a common ground line, also are routed in n-type diffusion.
Layout of Contact-Mask Programmable

4Bit × 4Bit NOR ROM

- In reality, the metal lines are laid out directly on top of diffusion columns to reduce the horizontal dimension.
Implant-Mask Programmable NOR ROM Array

Logic “0” is stored in each cell:
Present transistor

- $V_{T0}$ is implanted to activate 1 bit:
  Let $V_{T0} > V_{DD} \Rightarrow$ permanently **turn off** transistor
  $\Rightarrow$ disconnect the contact
Each diffusion-to-metal contact is shared by two adjacent transistors → need smaller area than contact-mask ROM layout
4Bit × 4Bit NAND-based ROM Array

- All word lines are kept at logic “1” level, except the selected line pulled down by “0” level.
- Logic “0” is stored: Absent transistor
  Logic “1” is stored: Present transistor

<table>
<thead>
<tr>
<th>( R_1 )</th>
<th>( R_2 )</th>
<th>( R_3 )</th>
<th>( R_4 )</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( C_3 )</th>
<th>( C_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Layout of Implant-Mask Programmable 4Bit × 4Bit NAND ROM

- No contact in the array ⇒ More compact than NOR ROM array
- Series-connected nMOS transistors exist in each column
  ⇒ The access time is slower than NOR ROM
Design of Row and Column Decoders

- Row and Column Decoders: To select a particular memory location in the array.

**Row Decoder**

- 2 address bits
-⇒ 4 word lines

<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
NOR-based Row Decoder Circuit
2 Address Bits and 4 Word Lines

<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Implementation of Row Decoder and ROM

- Can be implemented as *two adjacent* NOR planes

![Diagram](image-url)
Implementation of Row Decoder and ROM (Cont.)

• Can also be implemented as two adjacent NAND planes

\[ 2^N \text{ word lines} \]

\[ 2^M \text{ columns} \]

<table>
<thead>
<tr>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( R_1 )</th>
<th>( R_2 )</th>
<th>( R_3 )</th>
<th>( R_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

4×4 NAND ROM Array
Column Decoder (1)

NOR Address Decoder and Pass Transistors

- **Column Decoder**: To select one out of $2^M$ bits lines of the ROM array, and to route the data of the selected bit line to the data output
- **NOR-based column address decoder and pass transistors**:  
  » Only one nMOS pass transistor is turned on at a time  
  » # of transistors required: $2^M(M+1)$ ($2^M$ pass transistors, $M2^M$ decoder)
Column Decoder (2)

Binary Tree Decoder

- **Binary Tree Decoder:** A binary selection tree with consecutive stages
  - The pass transistor network is used to select one out of every two bit lines at each stage. The NOR address decoder is not needed.
  - **Advantage:** Reduce the transistor count \((2^{M+1}-2+M)\)
  - **Disadvantage:** Large number of series connected nMOS pass transistors ⇒ long data access time

Data output: Serial or Parallel

Column address bits
An Example of NOR ROM Array

- Consider the design of a 32-kbit **NOR ROM** array and the design issues related to *access time analysis*
  - # of total bits: 15 \( (2^{15}=32,768) \)
  - 7 row address bits \( (2^7 = 128 \) rows) \)
  - 8 column address bits \( (2^8 = 256 \) columns) \)
  - Layout: implant-mask
  - \( W = 2 \) µm, \( L = 1.5 \) µm
  - \( \mu_n C_{ox} = 20 \) µA/V²
  - \( C_{ox} = 3.47 \) µF/cm²
  - \( R_{\text{sheet-poly}} = 20 \) Ω/square

- \( R_{row} \) and \( C_{row} / \) unit memory cell
  - \( C_{row} = C_{ox} \cdot W \cdot L = 10.4 \) fF/bit
  - \( R_{row} = (\# \text{ of squares}) \times R_{\text{sheet-poly}} = 3 \times 20 = 60 \) Ω
An Example of NOR ROM Array (Cont.)

• The poly word line can be modeled as a RC transmission line with up to 256 transistors

  \[ t_{row} \approx 0.38 \cdot R_T \cdot C_T = 15.53 \text{ ns} \]

  \[ R_T = \sum_{\text{all cols}} R_i = 15.36 \text{ k}\Omega \]

  \[ C_T = \sum_{\text{all cols}} C_i = 2.66 \text{ pF} \]
An Example of NOR ROM Array (Cont.)

- A more accurate RC delay value: *Elmore time constant* for RC ladder circuits

\[
t_{row} = \sum_{k=1}^{256} R_{jk} C_k = 20.52 \text{ ns}
\]

- The column access time \( t_{column} \): worst case delay \( \tau_{PHL} \) associated with discharging the precharged bit line when a row is activated.
An Example of NOR ROM Array (Cont.)

- \( C_{\text{column}} = 128 \times (C_{\text{gd,driver}} + C_{\text{db,driver}}) \approx 1.5\text{pF} \)
  where \( C_{\text{gd,driver}} + C_{\text{db,driver}} = 0.0118 \text{ pF/word line} \)

- Since only one word line is activated at a time, the above circuit can be reduced to an inverter circuit

\[
\begin{align*}
V_{\text{DD}} & \rightarrow \left( \frac{4}{1.5} \right) \\
R_1 & \rightarrow \left( \frac{2}{1.5} \right) \\
C_{\text{column}} &
\end{align*}
\]

Remark: \( \tau_{\text{PLH}} \) is not considered because the bit line is precharged high before each row access operation

\[
t_{\text{column}} = \tau_{\text{PHL}} = \frac{C_{\text{load}}}{k_n(V_{\text{OH}} - V_{T0,n})} \left[ \frac{2V_{T0,n}}{V_{\text{OH}} - V_{T0,n}} + \ln \left( \frac{4(V_{\text{OH}} - V_{T0,n})}{V_{\text{OH}} + V_{\text{OL}}} - 1 \right) \right] = 18\text{ns}
\]

\[
t_{\text{access}} = t_{\text{row}} + t_{\text{column}} = 20.52 + 18 = 38.52 \text{ ns}
\]
Static Random Access Memory (SRAM)

- SRAM: The stored data can be retained indefinitely, without any need for a periodic refresh operation.

- Complementary Column arrangement is to achieve a more reliable SRAM operation.
Resistive-Load SRAM Cell

Basic cross-coupled 2-inverter latch with 2 stable op points for storing one-bit

SRAM cell is accessed via two bit (column) lines $C$ and its complement for reliable operation

Pass transistors to activated by a row select (RS) signal to enable read/write operators
Full CMOS and Depletion-Load SRAM Cell

Full CMOS SRAM Cell

Depletion-Load SRAM Cell
SRAM Operation Principles

- **RS=0**: The word line is not selected. $M_3$ and $M_4$ are OFF
  - One data-bit is held: The latch preserves one of its two stable states.
  - If RS=0 for all rows: $C_C$ and $C_{\overline{C}}$ are charged up to near $V_{DD}$ by pulling up of $M_{P1}$ and $M_{P2}$ (both in saturation)
    \[
    V_{\overline{C}} = V_C = V_{DD} - (V_{T0} + \gamma \sqrt{|2\phi_F|} + V_C - \sqrt{|2\phi_F|})
    \]
    - Ex: $V_C = V_{\overline{C}} = 3.5V$ for $V_{DD} = 5V$, $V_{T0}=1V$, $|2\phi_F|=0.6V$, $=0.4V^{1/2}$
SRAM Operation Principles (Cont.)

Pull-up transistor (one per column)

- **RS=1**: The word line is now selected. $M_3$ and $M_4$ are ON

Four Operations

1. **Write “1” Operation** ($V_1=V_{OL}$, $V_2=V_{OH}$ at $t=0^-$):

   $V_C \Rightarrow V_{OL}$ by the *data-write circuitry*. Therefore, $V_2 \Rightarrow V_{OL}$, then $M_1$ turns *off* $V_1 \Rightarrow V_{OH}$ and $M_2$ turns on pulling down $V_2 \Rightarrow V_{OL}$.
2. **Read “1” Operation** 

\( V_I = V_{OH}, \ V_2 = V_{OL} \) at \( t = 0^- \):

- \( V_C \) retains pre-charge level, while \( V_{\overline{C}} \Rightarrow V_{OL} \) by \( M_2 \) ON. **Data-read circuitry** detects small voltage difference \( V_C - V_{\overline{C}} > 0 \), and amplifies it as a “1” data output.
3. Write “0” Operation ($V_1=V_{OH}$, $V_2=V_{OL}$ at $t=0^*$):

$V_C \Rightarrow V_{OL}$ by the *data-write circuitry*.

Since $V_1 \Rightarrow V_{OL}$, $M_2$ turns off, therefore $V_2 \Rightarrow V_{OH}$. 
4. **Read “0” Operation** \((V_1=V_{OL}, V_2=V_{OH} \text{ at } t=\theta^-):\)

- \(V_C\) retains pre-charge level, while \(V_C \Rightarrow V_{OL}\) by \(M_1\) **ON**.
- **Data-read circuitry** detects small voltage difference \(V_C - V_{\overline{C}} < 0\), and amplifies it as a “0” data output.

---

4. **Read “0” Operation** \((V_1=V_{OL}, V_2=V_{OH} \text{ at } t=\theta^-):\)

- \(V_C\) retains pre-charge level, while \(V_C \Rightarrow V_{OL}\) by \(M_1\) **ON**.
- **Data-read circuitry** detects small voltage difference \(V_C - V_{\overline{C}} < 0\), and amplifies it as a “0” data output.
SRAM Operation Principles (Cont.)

Pull-up transistor (one per column)

- **$V_{DD}$**
- **$M_{P1}$**
- **$M_{P2}$**
- **$V_C$**
- **$V_{CC}$**
- **$R$**
- **$M_1$**
- **$M_2$**
- **$M_3$**
- **$M_4$**
- **$V_{CC}$**
- **$CC$**
- **$C_C$**
- **$RS$**
- **$V_{C}$**
- **$V_{C^{-}}$**

**Pull-up Transistor (one per column)**

- **Write 1**: $V_{C} = 3.5V$
- **Read 1**: $V_{C} = 3.5V$
- **Write 0**: $V_{C} = 3.5V$
- **Read 0**: $V_{C} = 3.0V$

- **Hold**: $V_{C} = 3.0V$
- **$0V$**: $V_{C} = 0V$

CMOS Digital Integrated Circuits
Static or “Standby” Power Consumption

• **Assume**: 1 bit is stored in the cell ⇒ $M_1$ OFF, $M_2$ ON ⇒ $V_1=V_{OH}$, $V_2=V_{OL}$. *I.E. One load resistor is always conducting non-zero current.*

\[
P_{\text{standby}} = \frac{(V_{DD}-V_{OL})^2}{R}
\]

with $R = 100M\Omega$ (undoped poly), $P_{\text{standby}} \approx 0.25 \, \mu W$ per cell for $V_{DD} = 5V$
Circuit of CMOS SRAM Cell

Pull-up transistor (one per column)

(Column voltages can reach to full $V_{DD}$)

Advantages

- Very low standby power consumption
- Large noise margins than $R$-load SRAMS
- Operate at lower supply voltages than $R$-load SRAMS

Disadvantages

- Larger die area: To accommodate the n-well for pMOS transistors and polysilicon contacts. The area has been reduced by using multi-layer polysilicon and multi-layer metal processes
- CMOS more complex process
CMOS SRAM Cell Design strategy

- Two basic requirements which dictate \( W/L \) ratios
  - Data-read operation should **not destroy data** in the cell
  - **Allow modification** of stored data during data-write operation

Pull-up transistor (one per column)

\[ (Column \ vibrations \ can \ reach \ to \ full \ V_{DD}) \]

- **Read “0” operation**
  - at \( t=0^- \): \( V_1=0V, \ V_2=V_{DD} \); \( M_3, M_4 \) OFF; \( M_2, M_5 \) OFF; \( M_1, M_6 \) Linear
  - at \( t=0^+ \): \( RS = V_{DD}, M_3 \) Saturation, \( M_4 \) Linear; \( M_2, M_5 \) OFF; \( M_1, M_6 \) Linear

- **Slow discharge of large \( C_C \)**: Require \( V_1 < V_{T,2} \) ⇒ **Limits \( M_3 \) \( W/L \) wrt \( M_1 \) \( W/L \)
CMOS SRAM Cell Design strategy (Cont.)

- **Design Constraint:** \( V_{I,\text{max}} < V_{T,2} = V_{T,n} \) to keep \( M_2 \) OFF

  » \( M_3 \) saturation, \( M_1 \) linear \( \Rightarrow \)

  \[
  k_{n,3}(V_{DD}-V_1-V_{T,n})^2/2 = k_{n,1}(2(V_{DD}-V_{T,n})V_1-V_1^2)/2
  \]

  » Therefore,

\[
\frac{k_{n,3}}{k_{n,1}} = \left(\frac{W}{L}\right)_3 < \frac{2(V_{DD}-1.5V_{T,n})V_{T,n}}{(V_{DD}-2V_{T,n})^2}.
\]

Symmetry:

- \textit{Same} for \( k_{n,4}/k_{n,2} \)
- \( M_1 \) OFF for Read “1”
CMOS SRAM Cell Design strategy (Cont.)

- **Write “0” operation with “1” stored in cell:**

  - **Pull-up transistor (one per column)**
  - *(Column voltages can reach to full $V_{DD}$)*

  - **$V_C$ is set “0” by data-write circuit** *(“1” stored)*
    - at $t=0$: $V_1=V_{DD}$, $V_2=0V$; $M_3$, $M_4$ OFF; $M_2$, $M_5$ Linear; $M_1$, $M_6$ OFF
    - at $t=0$: $V_C=0V$, $V_C=V_{DD}$; $M_3$, $M_4$ saturation; $M_2$, $M_5$ Linear; $M_1$, $M_6$ OFF
    - » Write “0” $\Rightarrow V_1: V_{DD} \rightarrow 0 (<V_{2T,n})$ and $V_2: 0 \rightarrow V_{DD}(M_2 \rightarrow OFF)$
CMOS SRAM Cell Design strategy (Cont.)

- **Design constraint:** \( V_{l,\text{max}} < V_{T,2} = V_{T,n} \) to keep \( M_2 \) OFF
  
  - When \( V_I = V_{T,n} \): \( M_3 \) Linear and \( M_5 \) saturation \( \Rightarrow \)

\[
k_{p,5}(0-V_{DD}-V_{T,p})^2/2 = k_{n,3}(2(V_{DD}-V_{T,n})V_{T,n}-V_{T,n}^2)/2
\]

- \( V_I < V_{T,n} \), i.e. \( M_2 (M_1) \) forced OFF

\[
\frac{k_{p,5}}{k_{n,3}} = \frac{k_{p,6}}{k_{n,4}} < \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2} \Rightarrow
\]

By symmetry

\[
\left( \frac{W}{L} \right)_3 < \left( \frac{W}{L} \right)_6 \quad \Rightarrow \quad \frac{\mu_n}{\mu_p} \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2} \quad \Rightarrow
\]
### SRAM Write Circuit

#### Diagram Description:
- **SRAM Cell**: The core component of the SRAM, where data is stored.
- **Bit Lines (C)**: Connected to the top and bottom of the cell, modulated by the voltage state of V_C.
- **Word Line (W)**: Controls the gate of M1 and M2, determining the state of V_C.
- **Input Signals**:
  - W: Write enable.
  - DATA: Data input.
  - WB: Write back signal.

#### Operation Table (M3 on):
<table>
<thead>
<tr>
<th>W</th>
<th>DATA</th>
<th>WB</th>
<th>Operation (M3 on)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>M1 off, M2 on ⇒ V_C → low</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M1 on, M2 off ⇒ V_C → low</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>0</td>
<td>M1 off, M2 off ⇒ V_C, V_C no change</td>
</tr>
</tbody>
</table>

---

**Shared by several columns**: Indicates that the write operation is replicated across multiple columns.

---

**From Column Decoder**: Suggests that the write operation is triggered by the decoder output (W)

---

**CMOS Digital Integrated Circuits**
SRAM Read Circuit

Source coupled differential amplifier

\[ I_{D1} = \frac{k_n}{2} (V_C - V_X - V_{T1,n})^2 \]
\[ I_{D2} = \frac{k_n}{2} (V_C - V_X - V_{T2,n})^2 \]
\[ A_{sense} = \frac{1}{\partial (V_{o1} - V_{o2})} = -g_m R \]
\[ g_m = \frac{\partial I_D}{\partial V_{GS}} = \sqrt{2k_n I_D} \]

Increase R→
Use active load
Use cascode
Fast Sense Amplifier

- $V_C < V_{C\overline{C}}$: $M_1 \Rightarrow \text{OFF}$, $V_o$ decreases, $V_{ON} \Rightarrow \text{High}$
- $V_C > V_{C\overline{C}}$: $M_2 \Rightarrow \text{OFF}$, $V_o$ remains high, $V_{ON} = \text{Low}$

$A_{sense} = -g_{m2}(r_{o2}||r_{o5})$
Two-Stage differential Current-Mirror Amplifier Sense Circuit
Typical Dynamic Response for One and Two Stage Sense Amplifier Circuits

![Graph showing voltage response over time for Output-1 and Output-2 stages, with VC indicating the voltage change over time.](image-url)
Cross-Coupled nMOS Sense Amplifier

- **Assume:** $M_3$ OFF, $V_C$ and $V_{\overline{C}}$ are initially precharged to $V_{DD}$
- **Access:** $V_C$ drops slightly less than $V_{\overline{C}}$
- $M_3 \Rightarrow$ ON and $V_C < V_{\overline{C}}$: $M_1$ ON first, pulling $V_C$ lower
  - $M_2$ turns OFF, $C_C$ discharge via $M_1$ and $M_3$

Enhances differential voltage $V_C - V_{\overline{C}}$

Does not generate output logic level
Dynamic Read-Write Memory (DRAM) Circuits

- **SRAM**: 4~6 transistors per bit
  - 4~5 lines connecting as charge on capacitor
- **DRAM**: Data bit is stored as charge on capacitor
  - Reduced die area
  - Require periodic refresh

Four-Transistor DRAM Cell

\[ M_1 \quad M_3 \quad M_2 \quad M_4 \]

parasitic storage capacitances
Three-Transistor DRAM Cell

parasitic storage capacitances
One-Transistor DRAM Cell

- **Industry standard** for high density dram arrays
- **Smallest** component count and silicon area per bit
- Separate or “explicit” capacitor (dual poly) per cell
Operation of Three-Transistor DRAM Cell

- The binary information is stored as the charge in $C_1$
- **Storage transistor** $M_2$ is on or off depending on the charge in $C_3$
- **Pass transistors** $M_1$ and $M_3$: access switches
- Two separate bit lines for “data read” and “data write”
The operation is based on a two-phase non-overlapping clock scheme

- The precharge events are driven by $\phi_1$, and the “read” and “write” operations are driven by $\phi_2$.
- Every “read” and “write” operation is preceded by a precharge cycle, which is initiated with $PC$ going high.
• Write “1” OP: \( DATA = 0 \), \( WS = 1 \); \( RS = 0 \)
  
  » \( C_2, C_1 \) share charge due to \( M_1 \) ON

  » Since \( C_2 \gg C_1 \), the storage node \( C_1 \) attains approximately the same logic level.
• Read “1” OP: \( \overline{DATA} = 0, WS = 0; RS = 1 \)
  \( M_2, M_3 \text{ ON } \Rightarrow C_3, C_1 \) discharges through \( M_2 \) and \( M_3 \), and the falling column voltage is interpreted by the “data read” circuitry as a stored logic “1”.

Operation of Three-Transistor DRAM Cell (Cont.)
Operation of Three-Transistor DRAM Cell (Cont.)

- **Write “0” OP:** \( \overline{DATA} = 1, WS = 1; RS = 0 \)

  \[ M_2, M_3 \text{ ON } \Rightarrow C_2 \text{ and } C_1 \text{ discharge to 0 through } M_1 \text{ and data\_in nMOS.} \]
• Read “0” OP: $\overline{\text{DATA}} = 1$, $WS = 0$; $RS = 1$

  » $C_3$ does not discharge due to $M_2$ OFF, and the logic-high level on the $Data\_out$ column is interpreted by the data read circuitry as a stored “0” bit.
Operation of One-Transistor DRAM Cell

- **Write “1” OP:** $BL = 1$, $WL = 1$ ($M_1$ ON) $\Rightarrow C_1$ charges to “1”
- **Write “0” OP:** $BL = 0$, $WL = 1$ ($M_1$ ON) $\Rightarrow C_1$ discharges to “0”
- **Read OP:** destroys stored charge on $C_1$ $\Rightarrow$ destructive refresh is needed after every data read op.