17. Semiconductor Memories

Overview

- Introduction
- Read Only Memory (ROM)
- Nonvolatile Read/Write Memory (RWM)
- Static Random Access Memory (SRAM)
- Dynamic Random Access Memory (DRAM)
- Summary
Semiconductor Memory Classification

<table>
<thead>
<tr>
<th>Non-Volatile Memory</th>
<th>Volatile Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Only Memory (ROM)</td>
<td>Read/Write Memory (RWM)</td>
</tr>
<tr>
<td>Mask-Programmable ROM</td>
<td>EPROM</td>
</tr>
<tr>
<td>Programmable ROM</td>
<td>E²PROM</td>
</tr>
<tr>
<td>FLASH</td>
<td>SRAM</td>
</tr>
<tr>
<td></td>
<td>DRAM</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EPROM - Erasable Programmable ROM
E²PROM - Electrically Erasable Programmable ROM
SRAM - Static Random Access Memory
DRAM - Dynamic Random Access Memory
FIFO - First-In First-Out
LIFO - Last-In First-Out

Random Access Memory Array Organization

- Memory array
  - Memory storage cells
  - Address decoders

Each memory cell
- stores one bit of binary information ("0" or "1" logic)
- shares common connections with other cells: rows, columns
Read Only Memory - ROM

- Simple combinatorial Boolean network which produces a specific output for each input combination (address)
  - "1" bit stored - absence of an active transistor
  - "0" bit stored - presence of an active transistor
- Organized in arrays of $2^N$ words

- Typical applications:
  - store the microcoded instructions set of a microprocessor
  - store a portion of the operation system for PCs
  - store the fixed programs for microcontrollers (firmware)

Mask Programmable NOR ROM (1)

- Each column $C_i$ (NOR gate) corresponds to one bit of the stored word
- A word is selected by rising to "1" the corresponding wordline
- All the wordlines are "0" except the selected wordline which is "1"

<table>
<thead>
<tr>
<th>Function Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1  R2  R3  R4</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>1   0   0   0</td>
</tr>
<tr>
<td>0   1   0   0</td>
</tr>
<tr>
<td>0   0   1   0</td>
</tr>
<tr>
<td>0   0   0   1</td>
</tr>
</tbody>
</table>

NOR ROM with 4-bit words
**Mask Programmable NOR ROM (2)**

- “1” bit stored - the drain/source connection (or the gate electrode) are omitted in the final metallization step.
- “0” bit stored - the drain of the corresponding transistor is connected to the metal bit line.

Cost efficient, since few masks have to be manufactured only.

**Implant Mask Programmable NOR ROM**

Idea: deactivation of the NMOS transistors by raising their threshold voltage above the $V_{OH}$ level through channel implants.

- “1” bit stored - the corresponding transistor is turned off through channel implant.
- “0” bit stored - non-implanted (normal) transistors.

Advantage: higher density (smaller area)!
Implant Mask Programmable NAND ROM (1)

- Each column $C_i$ (NAND gate) corresponds to one bit of the stored word
- A word is selected by putting to “0" the corresponding wordline $R_i$
- All the wordlines $R_i$ are “1" except the selected wordline which is “0".

Normally on transistors: have a lower threshold voltage (channel implant)

NAND ROM with 4-bit words

Implant-Mask-Programmable NAND ROM (2)

- The structure is more compact than NOR array (no contacts)
- The access time is larger than NOR array access time (chain of nMOS)
NOR Row Address Decoder for a NOR ROM Array

- The decoder must select out one row by rising its voltage to “1” logic
- Different combinations for the address bits $A_1A_2$ select the desired row
- The NOR decoder array and the NOR ROM array are fabricated as two adjacent arrays, using the same layout strategy

### NAND Row Address Decoder for a NAND ROM Array

- The decoder has to lower the voltage level of the selected row to logic “0” while keeping all the other rows at logic “1”
- The NAND row decoder of the NAND ROM array is implemented using the same layout strategy as the memory itself
NOR Column Address Decoder for a NOR ROM Array

NOR Address decoder + $2^M$ pass transistors
- Large area!

Binary selection tree decoder
- No need for NOR address decoder, but are necessary additional inverters!
- Smaller area
- Drawback - long data access time

Nonvolatile Read-Write Memories

- The architecture is similar to the ROM structure
- Array of transistors placed on a word-line/bit-line grid
- Special transistor that permits its threshold to be altered electrically
- Programming: selectively disabling or enabling some of these transistors
- Reprogramming: erasing the old threshold values and start a new programming cycle

Method of erasing:
- ultraviolet light - EPROMs
- electrically - EEPROMs
EPROM (1)

The floating gate avalanche-injection MOS (FAMOS) transistor:
- **extra polysilicon strip** is inserted between the gate and the channel - **floating gate**
- impact: double the gate oxide thickness, reduce the transconductance, increase the threshold voltage
- **threshold voltage is programmable** by the trapping electrons on the floating gate through avalanche injection

![Schematic symbol](image)

![Diagram](image)

EPROM (2)

- Electrons acquire sufficient energy to became “hot” and traverse the first oxide insulator (100nm) so that they get trapped on the floating gate
- **Electron accumulation** on the floating gate is a self-limiting process that increases the threshold voltage (~7V)
- The trapped charge can be stored for many years
- The erasure is performed by shining strong ultraviolet light on the cells through a transparent window in the package
- The UV radiation renders the oxide conductive by direct generation of electron-hole pairs
EPROM (3)

- The erasure process is slow (~min.)
- The erasure procedure is **off-system**!
- Programming takes several usecs/word
- Limited endurance - max 1000 erase/program cycles
- The cell is very simple and dense: large memories at low cost!
- Applications that do not require regular reprogramming

EEPROM

- Provide an electrical-erasure procedure
- Modified floating-gate device, floating-gate tunneling oxide (**FLOTOX**):
  - reduce the distance between floating gate and channel near the drain
  - Fowler-Nordheim tunneling mechanism (when apply 10V over the thin insulator)
- Reversible programming by reversing the applied voltage (rise and lower the threshold voltage) difficult to control the threshold voltage extra transistor required as access device
- Larger area than EPROM
- More expensive technology than EPROM
- Offers a higher versatility than EPROM
- Can support $10^5$ erase/write cycles
Flash Memories

Combines the density of the EPROM with the versatility of EEPROM structures
- Programming: avalanche hot-electron-injection
- Erasure: Fowler-Nordheim tunneling (as for EEPROM cells)
- Difference: erasure is performed in bulk for the complete (or subsection of) memory chip - reduction in flexibility!
- Extra access transistor of the EEPROM is eliminated because the global erasure process allows a careful monitoring of the device characteristics and control of the threshold voltage!
- High integration density

ETOX Flash cell - introduced by INTEL

Static Random Access Memory - SRAM (1)

- Permit the modification (writing) of stored data bits
- The stored data can be retained infinitely, without need of any refresh operation
- Data storage cell - simple latch circuit with 2 stable states
- Any voltages disturbance the latch switches from one stable point to the other stable point
- Two switches are required to access (r/w) the data
Resistive Load SRAM Cell - Operation Principle (1)

- MP1,2 pull up transistors - charge up the large column parasitic capacitances $C_C$, $C_{\overline{C}}$
- The steady-state voltage: $V_{CC} = V_{DD} - V_T \sim 3.5V$

Here we define the memory content to be located

The basic operations on SRAM cells

**RS = 1** (M3, M4 on)
- Read/Write “1”
- Read/Write “0”

**RS = 0** (M3, M4 off)
- data is being held
Resistive Load SRAM Cell - Operation Principle (2)

- **Write “1” operation** (RS = 1 - M3, M4 on)
  - $V_C$ - forced to 0 by data write circuitry, $V_2$ decreases to 0, M1 off; $V_1$ increases;
  - Final state: $V_1= 1$, $V_2= 0$

- **Read “1” operation** (RS = 1 - M3, M4 on)
  - M1 off; M2, M4 on; $V_C$ - pulled down , $V_C > V_C$ read as a logic “1”

- **Write “0” operation** (RS = 1 - M3, M4 on)
  - $V_C$ - forced to 0 by data write circuitry, $V_1$ goes to 0, M2 off; $V_2$ increases to 1
  - Final state: $V_1= 0$, $V_2= 1$

- **Read “0” operation** (RS = 1 - M3, M4 on)
  - M2 off; M1, M3 on; $V_C$ - pulled down, $V_C < V_C$ read as logic 0

**Full CMOS SRAM Cell**

- Low-power SRAM Cell: the static power dissipation is limited by the leakage current during a switching event
- The pMOS pull-up transistors allow the column voltage to reach full $V_{DD}$ level
- High noise immunity due to larger noise margins
- Lower power supply voltages than resistive-load SRAM cell
- Drawback: large area!
CMOS SRAM Cell Design Strategy (1)

Layout of the resistive-load SRAM cell

Layout of the CMOS SRAM cell

CMOS SRAM Cell Design Strategy (2)

(1) The **data read operation should not destroy the stored information**

Assume that a logic “0” is stored in the cell ($V_1 = 0, V_2 = 1$: M1, M6-linear; M2, M5-off)

- RS = 0: M3, M4-off;
- RS = 1: M3-saturation; M4, M1-linear

$V_C$ decreases, $V_1$ increases slowly

Condition - M2 must remain **turned off** during the data reading operation:

$$V_{1,\text{max}} \leq V_{T,2}; \quad I_{M3} = I_{M1} \Rightarrow$$

$$\left(\frac{W}{L}\right)_3 < \frac{2(V_{DD}-1.5V_{T,n})V_{T,n}}{(V_{DD}-2V_{T,n})^2}$$

Design rule:

A symmetrical rule is valid also for M2 and M4
CMOS SRAM Cell Design Strategy (3)

(2) The cell should allow modification of the stored information during the data write phase.

Consider the write “0” operation, assuming that “1” is stored in the cell ($V_1 = 1, V_2 = 0$: M1, M6-off; M2, M5-linear)

- RS = 0: M3, M4-off;
- RS = 1: M3, M4 saturation, M5-linear

In order to change the stored information: $V_1 = 0, V_2 = 1 \Rightarrow M1$ on and M2 off!

But $V_2 < V_{T1}$ (previous design condition) $\Rightarrow M1$ cannot be switched on! $\Rightarrow M2$ must be switched off $\Rightarrow V_1$ must be reduced below $V_{T2}$

$V_1 \leq V_{T,2}; I_{M3} = I_{M5} \Rightarrow$

\[
\left(\frac{W}{L}\right)_3 = \frac{\mu_n}{\mu_p}\frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2}
\]

Design rule:

A symmetrical rule is valid also for M6 and M4

---

SRAM Write Circuitry

<table>
<thead>
<tr>
<th>W</th>
<th>DATA</th>
<th>WB</th>
<th>WB</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>M1-off, M2-on, $V_C$ high, $V_{\overline{C}}$ low</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>M1-on, M2-off, $V_C$ low, $V_{\overline{C}}$ high</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>M1, M2 off, $V_C$, $V_{\overline{C}}$ high</td>
</tr>
</tbody>
</table>

Write operation is performed by forcing the voltage level of either column (bit line) to “0”
SRAM Read Circuitry

The read circuitry must detect a very small difference between the two complementary columns (sense amplifier)

\[
\frac{\partial (V_{o1} - V_{o2})}{\partial (V_c - V_{\overline{c}})} = -R \cdot g_m, \text{ where } g_m = \frac{\partial I_D}{\partial V_{GS}} = \sqrt{2k_n I_D}
\]

The gain can be increased by using
- active loads
- cascode configuration

Precharging of bit lines plays a significant role in the access time!
- The equalization of bit lines prior to each new access (between two access cycles)

Dual Port SRAM Arrays

Allows simultaneous access to the same location in the memory array (systems with multiple high speed processors).

- Eliminates wait states for the processes during data read operation
- Problems can occur if:
  - two processors attempt to write data simultaneously onto the same cell
  - one processor attempts to read while other writes data onto the same cell
- Solution: contention arbitration logic
Dynamic Random Access Memories - DRAM (1)

SRAM drawbacks
- large area: 4-6 transistors/bit + 4 lines connections
- static power dissipation (exception CMOS SRAM)

Need for high density RAM arrays → DRAM

DRAM
- binary data is stored as charge in a capacitor
- requires periodic refreshing of the stored data
- no static power dissipation

4-transistor DRAM cell
- one of the earliest DRAM cells
- derived from 6 transistor SRAM cell
- two storage nodes (parasitic capacitances)
- large area

Dynamic Random Access Memories - DRAM (2)

3-transistor DRAM cell
- 1 transistor - storage device
- 2 transistors for r/w access (switches)
- 2 r/w control lines
- 2 I/O lines

1-transistor DRAM cell
- 1 transistor for r/w access
- 1 explicit capacitor - information storage
- 1 r/w control line
- 1 I/O line
Three-Transistor DRAM Cell (1)

MP1, MP2 pull-up (precharge) transistors
M2 storage transistor (on or off depending on the charge stored in C1)
M1, 3 access switches
C2, 3 >> C1

Two phase non overlapping clock scheme
CLK1 - precharge events
CLK2 - r/w events (CLK1 - low)

Three-Transistors DRAM Cell (2)

• Every r/w operation is preceded by a precharge cycle - C2, 3 are charged up
• Refresh operation (row): data are read, inverted and written back into the same cell location every 2-4 ms
Three-Transistors DRAM Cell (3)

**WRITE 1 operation:**

- Precharge: C2, C3 charged up to 1 logic level
- \( \text{DATA} = 0, \text{MD off}; \text{WS} = 1, \text{M1 on} \Rightarrow \text{the charge on C2 is shared with C1} \)
- After write operation: WS = 0, M1 off; Since C1 is charged up to 1: M2 on

**READ 1 operation:**

- Precharge: C2, C3 charged up to 1 logic level
- RS = 1, M3 on, M2 on
- C3 discharges through M2, M3 and the falling column voltage is interpreted as a stored 1

Three Transistors DRAM Cell (4)

**Write 0 operation:**

- Precharge: C2, 3
- \( \text{DATA} = 1, \text{MD on}; \text{WS} = 1, \text{M1 on} \Rightarrow \text{C2, C1 pulled to 0 through M1 and MD;} \)
- After write operation WS = 0, M1 off; C2 is discharged to 0, M2 off

**READ 0 operation:**

- Precharge: C2, 3
- RS = 1, M3 on; M2 off
- C3 does not discharge - the 1 logic level is interpreted as a stored 0

C1 is discharged by the leakage currents of M1 - data must be periodically read, inverted and written back!
One-Transistor DRAM Cell (1)

Charge sharing between C2 and C1 has a key role in the r/w operations

- **Data WRITE:**
  - "1" - D = 1, R/W = 1 M1-on; C1 charge up to 1 level
  - "0" - D = 0, R/W = 0 M1-on; C1 discharge to 0 level

- **Data READ (destructive operation):**
  - Precharge C2
  - R/W = 1 M1-on; charge sharing between C1 and C2
  - Data refresh operation is required!

One-Transistor DRAM cell (2)

One Transistor DRAM cell with trench capacitor (cross-section)
Data Read Example (1)

- 256 cells per column DRAM
- The storage array is split in half
- A cross-coupled dynamic latch is used to restored the signal levels
- The dummy cell has a capacitance equal to half of the storage capacitance value

Three stages read-refresh operation

Data Read Example (2)

- Precharge devices are turned on, $C_D$ and $C_S$ are charged up to “1” level
- The dummy nodes X and Y are pulled to “0” level
- During this phase all other signals are inactive
Data Read Example (3)

- One of the 256 word lines is raised to “1” (cell R_{128} is selected)
- The corresponding dummy cell on the other side is also selected (right)
- Charge sharing between the selected cell and C_D (depending on the value stored by cell “0” or “1”) and between dummy cell and C_D
- Voltage level is detected through the charge sharing

Data Read Example (4)

- Performed during the active phase of the CS (column-select signal)
- The slight voltage difference between the two half-column is amplified and the latch forces the two half-columns into opposite states
- The voltage level on the accessed cell is restored
### DRAM Architectures

<table>
<thead>
<tr>
<th>Name</th>
<th>Feature</th>
<th>Die size increase</th>
<th>Frequency (system level)</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM</td>
<td>Fast page mode</td>
<td>-</td>
<td>25MHz</td>
<td>Main memory</td>
</tr>
<tr>
<td>VRAM</td>
<td>DRAM+SAM</td>
<td>50%</td>
<td>40MHz</td>
<td>Video display buffer</td>
</tr>
<tr>
<td>EDO</td>
<td>DRAM with modified CAS</td>
<td>0%</td>
<td>40-50MHz</td>
<td>Main memory, low-end graphic memory</td>
</tr>
<tr>
<td>SDRAM</td>
<td>Sync.DRAM+Register (Latch)</td>
<td>0-10%</td>
<td>60-150MHz</td>
<td>Main memory in workstations, high end PCs, middle range graphic memory</td>
</tr>
<tr>
<td>SGRAM</td>
<td>SDRAM+Block write+ WPB</td>
<td>10%</td>
<td>60-150MHz 3Gb/s</td>
<td>High-end memory</td>
</tr>
<tr>
<td>CDRAM</td>
<td>Sync.DRAM+SRAM+ DTB</td>
<td>7-10%</td>
<td>66MHz</td>
<td>Low-end PC</td>
</tr>
<tr>
<td>RDRAM</td>
<td>Sync.DRAM+Raambus I/O</td>
<td>12-15%</td>
<td>250MHz</td>
<td>High-end PC, graphic memory</td>
</tr>
<tr>
<td>3D-RAM</td>
<td>Sync.DRAM+SRAM+ SAM+ALU</td>
<td>?</td>
<td>400Mb/s ext, 1.6Gb/s int</td>
<td>High-end graphic memory</td>
</tr>
<tr>
<td>EDRAM</td>
<td>DRAM + SRAM</td>
<td>?</td>
<td>?</td>
<td>Low-end PC</td>
</tr>
<tr>
<td>SVRAM</td>
<td>Sync.DRAM+SAM</td>
<td>50%</td>
<td>100MHz</td>
<td>High-end graphic memory</td>
</tr>
<tr>
<td>WRAM</td>
<td>VRAM with localized SAM</td>
<td>&lt;40%</td>
<td>66MHz</td>
<td>Middle to high-end graphic memory</td>
</tr>
</tbody>
</table>

### Summary

- The memory architecture has a major impact on the ease of use of the memory, its reliability and yield, its performance and power consumption;
- Memories are organized as arrays of cells; an individual cell is addressed by a column and row address;
- The memory cells should be designed so that a maximum signal is obtained in a minimum area; the cell design is dominated by technological considerations and most of the improvement in density results from scaling and advanced manufacturing processes;
- We have discussed cells for read-only memories (NOR and NAND ROM), nonvolatile memories (EPROM, EEPROM and FLASH) and read-write memories (SRAM and DRAM);
- The peripheral circuitry is very important to operate the memory in a reliable way and with reasonable performance; decoders, sense amplifiers and I/O buffers are an integral part of every memory design;