

Unit - V.

Low-Voltage Low-Power Memories.

Introduction:

Semiconductor memory arrays capable of storing large quantities of digital information are essential to all digital systems. The amount of memory required in a particular system depends on the type of application, but, in general, the number of transistors utilized for the information (data) storage function is much larger than the number of transistors used in logic operations and for other purposes. Thus, the maximum realizable data storage capacity of single-chip semiconductor memory arrays approximately doubles every two years. On-chip memory arrays have become widely used subsystems in many VLSI circuits, and commercially available single-chip read/write memory capacity has reached 64 megabits.

Memory circuits are generally classified according to the type of data storage and the type of data access. Read-Only Memory (ROM) circuits allow, as the name implies, only the retrieval of previously stored data and do not permit modifications of the stored information contents during normal operation. ROMs are non-volatile memories, i.e., the data storage function is not lost even when the power supply voltage is off. Depending on the type of data storage (data write) method, ROMs are classified as mask-programmed ROMs, Programmable ROMs (PROM), Erasable PROMs (EPROM), and Electrically Erasable PROMs (eEPPROM).

Semiconductor Memories

Read-Only Memory (ROM)

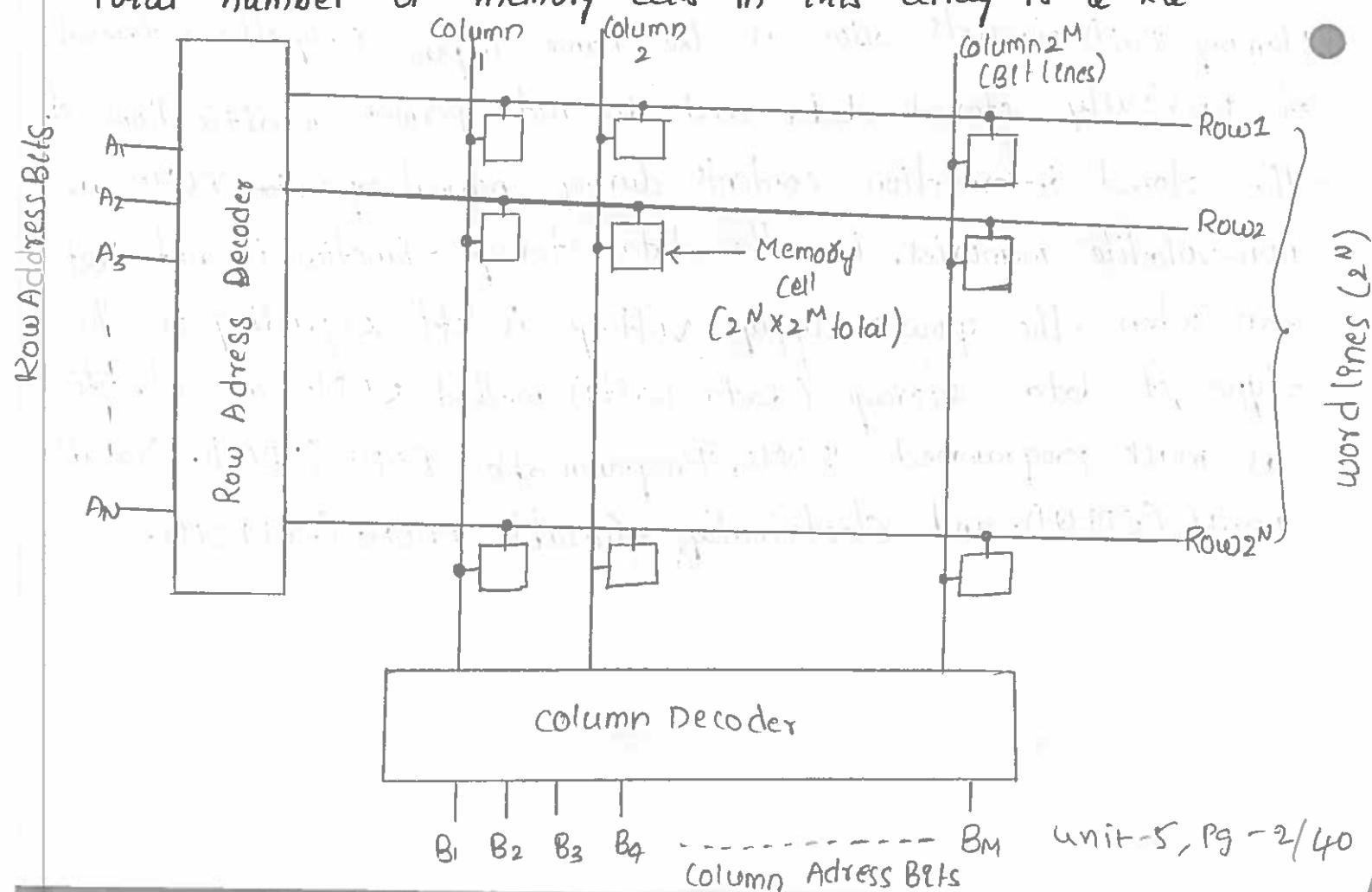
- Mask Programmed
- Programmable ROM (PROM)
- Erasable PROM (EPROM)
- Electrically EPROM (EEPROM)

Read/Write (R/W) Memory
or Random Access Memory (RAM)

Static RAM
(SRAM)

Dynamic RAM
(DRAM)

A typical memory array organization. The data storage structure, or core, consists of individual memory cells arranged in an array of horizontal rows and vertical columns. Each cell is capable of storing one bit of binary information. Also, each memory cell shares a common connection with the other cells in the same row, and another common connection with the other cells in the same column. In this structure, there are 2^N rows, also called word lines, and 2^M columns, also called bit lines. Thus, the total number of memory cells in this array is $2^M \times 2^N$.

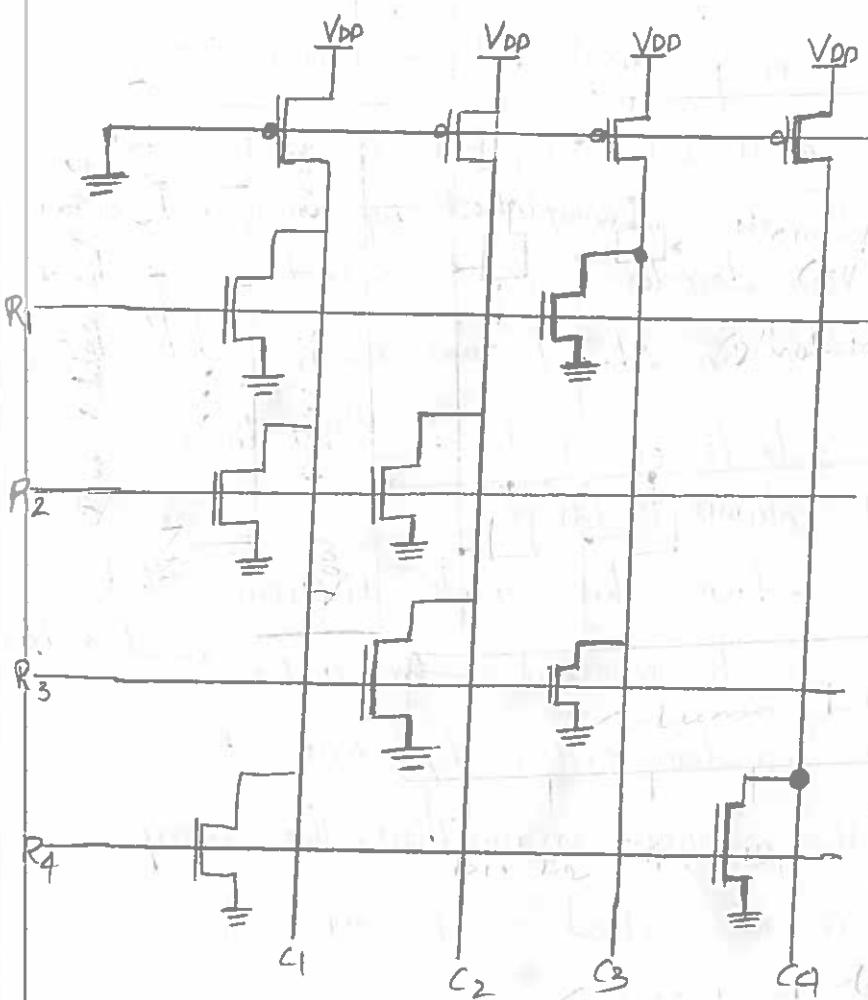


To access a particular memory cells, i.e., a particular data bit in this array, the corresponding bit line and the corresponding word line must be activated. The row and column selection operations are accomplished by row and column decoders, respectively. The row decoder circuit selects one out of 2^N word lines according to an N -bit row address, while the column decoder circuit selects one out of 2^M bit lines according to an M -bit column address.

We can see from this simple discussion that individual memory cells can be accessed for data read and/or data write operations in random order, independent of their physical locations in the memory array. Thus, the array organization examined here is called a Random Access Memory (RAM) structure.

Read-Only Memory (ROM) circuits:-

The read-only memory array can also be seen as a simple combinational Boolean network which produces a specified output value for each address. Thus, storing binary information at a particular address location can be achieved by the presence or absence of a data path from the selected row (word line) to the selected column (bit line), which is equivalent to the presence or absence of a device at that particular location. In the following, we will examine two different implementations for MOSROM arrays. Consider first the 4-bit \times 4-bit memory array shown in fig 10.3. Here, each column consists of a pseudo-nMOS NOR gate driven by some of the row signals, i.e., the word lines.



R ₁	R ₂	R ₃	R ₄	C ₁	C ₂	C ₃	C ₄
1	0	0	0	0	1	0	1
0	1	0	0	0	0	1	1
0	0	1	0	1	0	0	1
0	0	0	1	0	1	1	0

As described in the previous selection, only one word line is activated at a time by raising its voltage to V_{DD} , while all other rows are held at a low voltage level. If an active transistor exists at the cross point of a column and the selected row, the column voltage is pulled down to the logic level by that transistor. Thus, a logic "1"-bit is stored as the absence of an active transistor, while a logic "0"-bit is stored as the presence of an active transistor at the crosspoint.

In actual ROM layout, the array can be initially manufactured with nMOS transistors at every row-column intersection. The "1"-bits are then realized by omitting the drain or source connection, or the gate electrode of an corresponding nMOS transistors in the final metallization step.

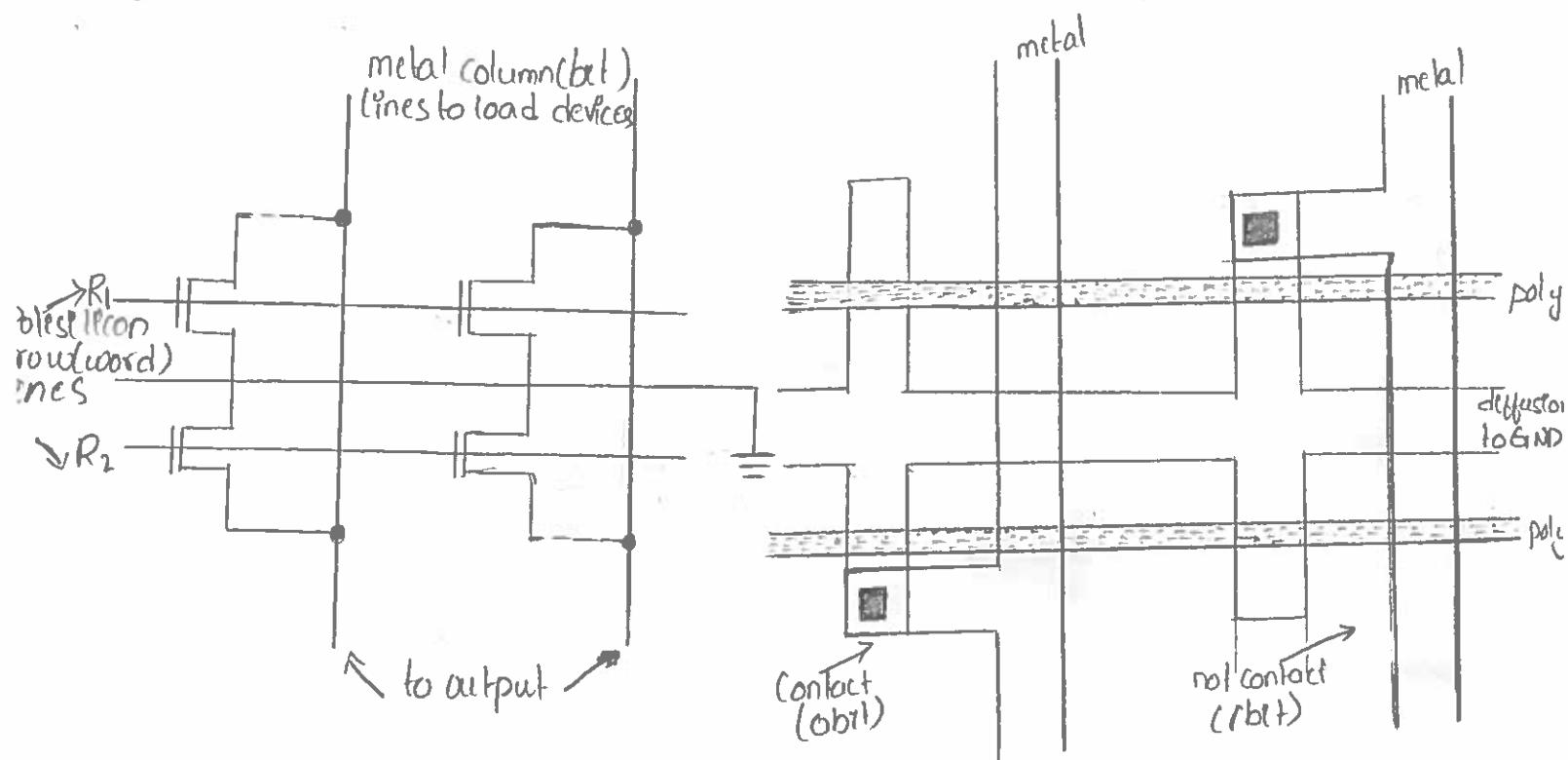


Fig. 8 shows a larger portion of the ROM array, except for the pMOS load transistors connected to the metal columns. Here, the 4-bit \times 4-bit ROM array shown in Fig. 1 is realized using the contact-mask programming methodology described above.

A different NOR ROM layout implementation is based on deactivation of the nMOS transistors by raising their threshold voltage through channel implants. Instead, the nMOS transistor corresponding to the stored "1"-bit can be deactivated, i.e., permanently turned off, by raising its threshold voltage above the V_{DD} level through a selective channel implant during fabrication.

The alternative-level layout of the 4-bit \times 4-bit ROM array example, which is based on implant-mask programming, is shown in Fig. 9. Note that in this case, each threshold voltage implant signifies a stored "1"-bit, and all other transistors correspond to stored "0"-bits. Since each diffusion-to-metal contact in this structure by two adjacent transistors.

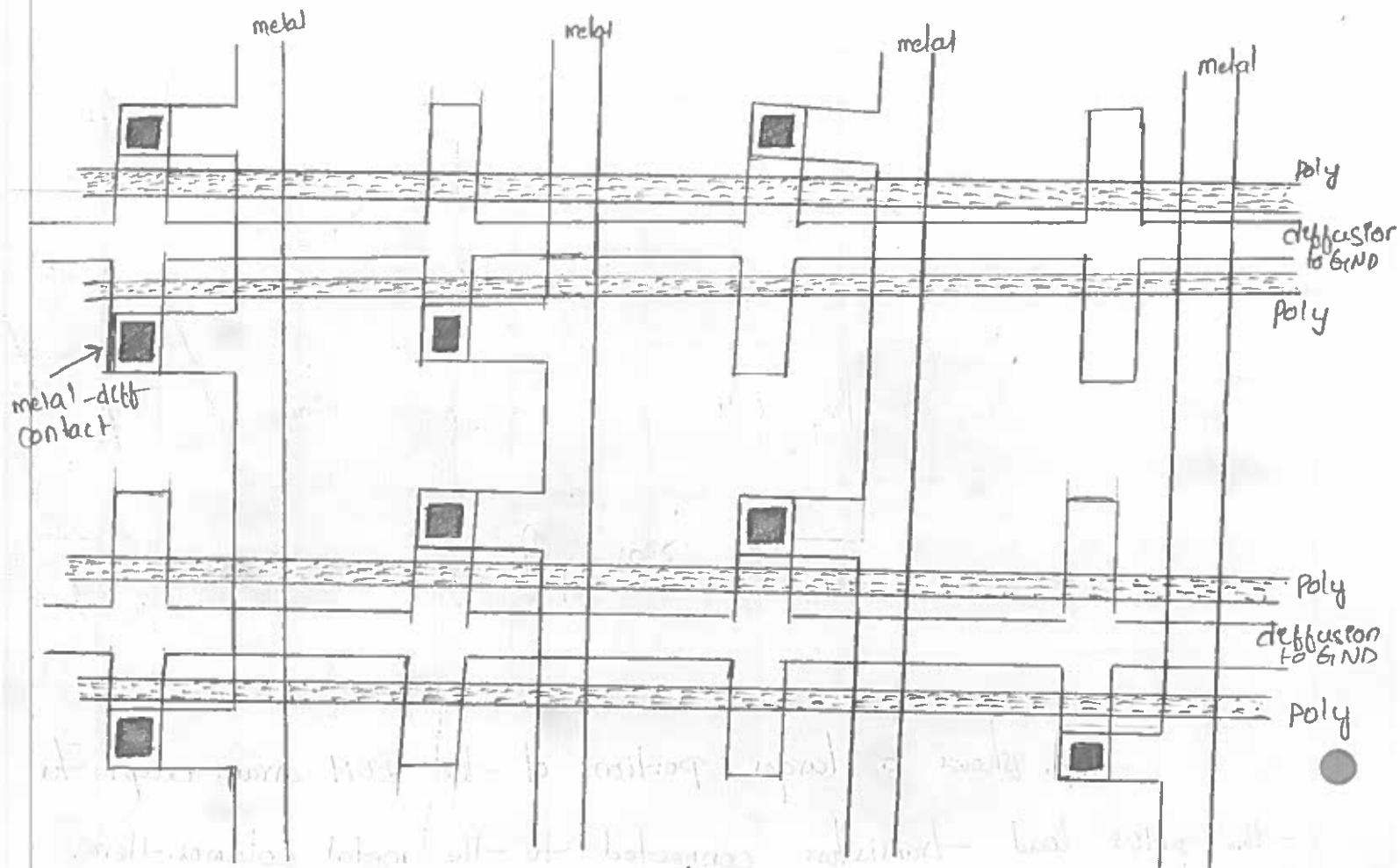


Figure 5.5 Layout of the 4-bit X 4-bit NOR ROM array Example Shown in fig. 5.3.

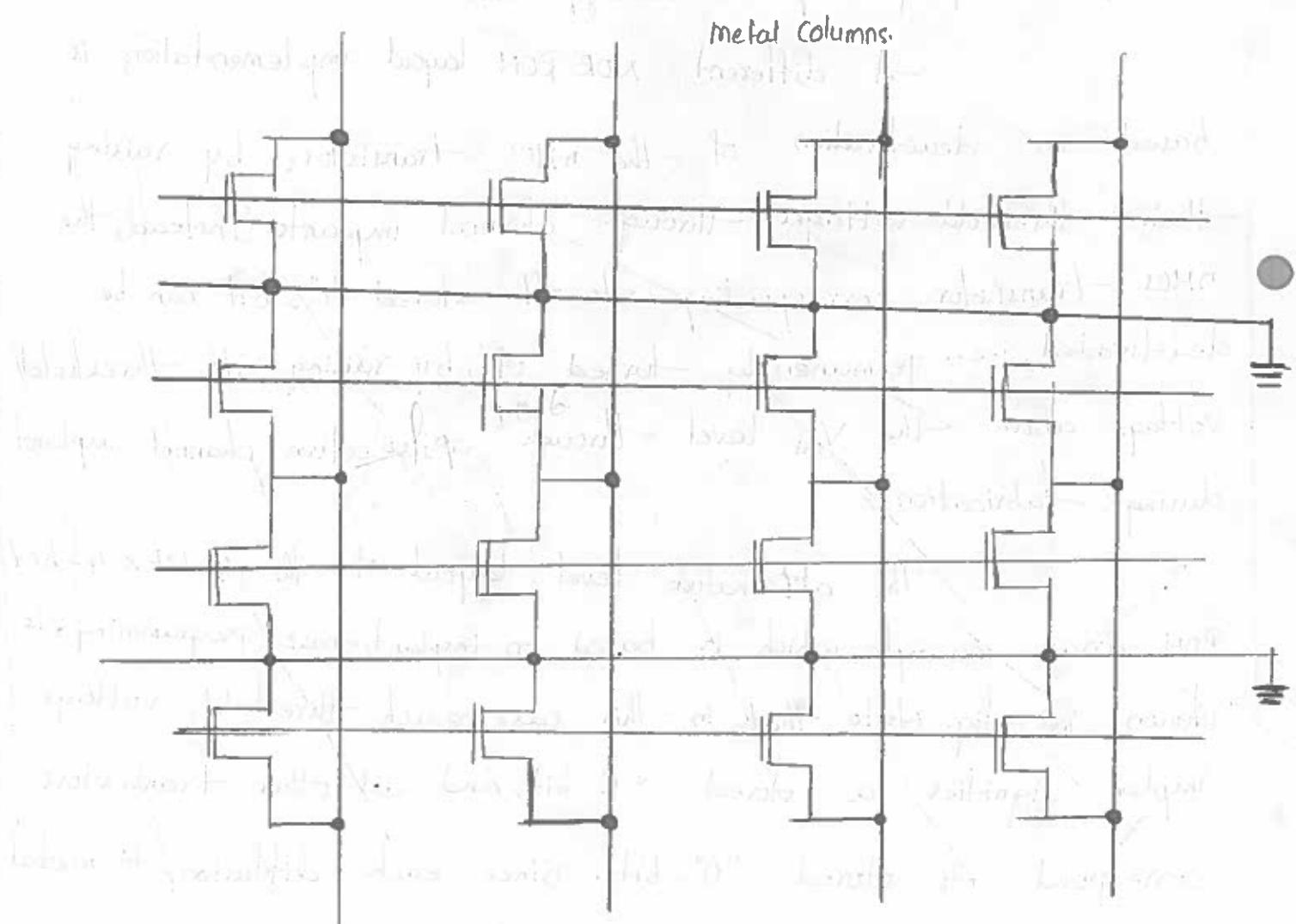


Figure 5.6 Arrangement of the nmos transistors in the implant-mask programmable NOR ROM array. Every metal-to-diffusion contact is shared by two adjacent devices.

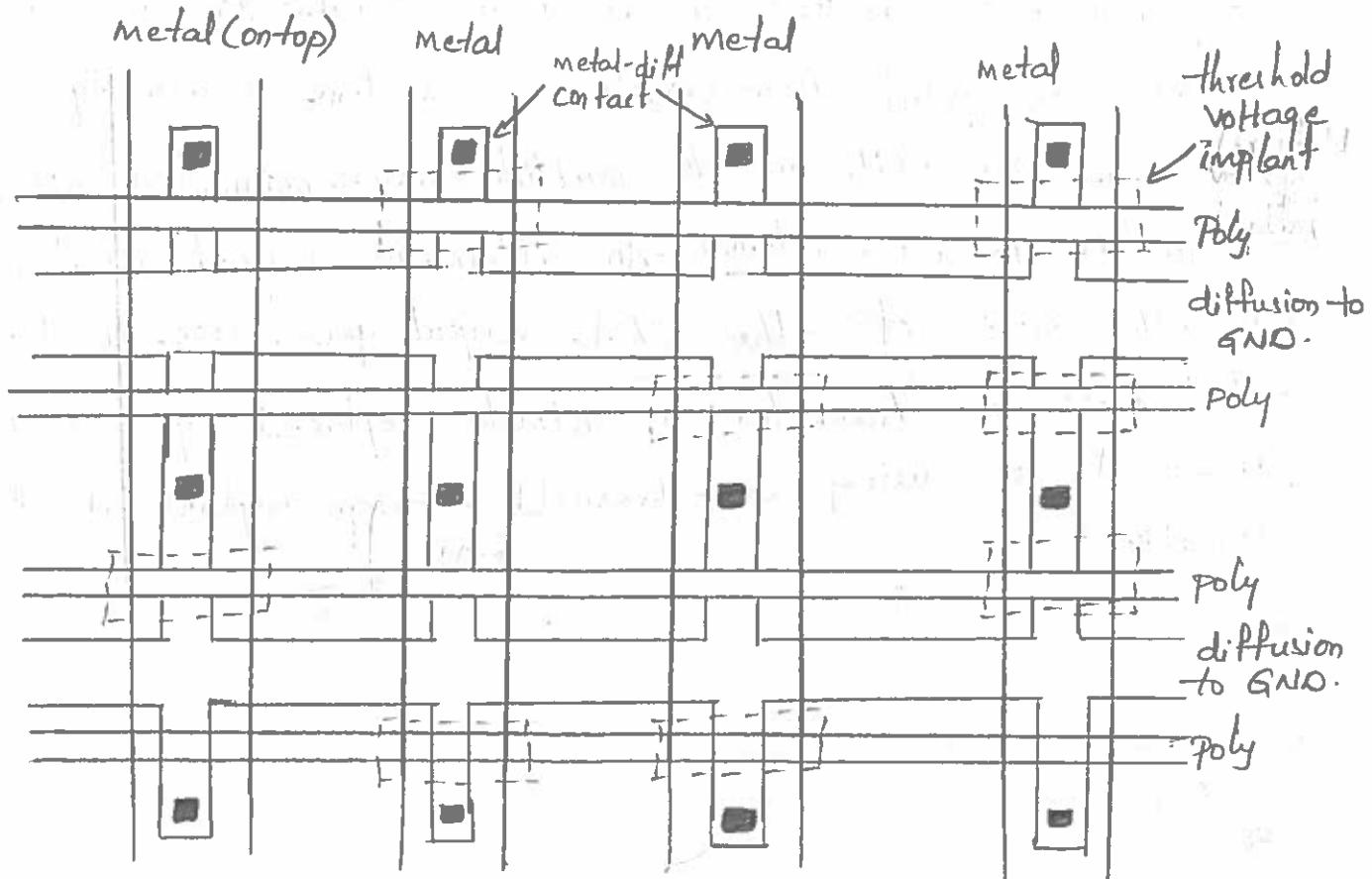


Fig 5.7: Layout of the 4-bit x 4-bit ROM array example shown in Fig 5.3. The threshold voltages of '1'-bit transistors are raised above V_{DD} through implant

Next, we will examine a significantly different ROM array design, which is also called a NAND ROM. Here, each bit line consists of a depletion-load NAND gate, driven by some of the row signals, i.e., the word lines. In normal operation, all word lines are held at the logic-high voltage level except for the selected line, which is pulled down to logic-low level. Thus, a logic "1"-bit is stored by the presence of a transistor that can be deactivated, while a logic "0"-bit is stored by a shorted or normally on transistor at the crosspoint.

The availability of this process step is also the reason why depletion-type nMOS load transistors are used instead of pMOS loads in the example shown below. Fig 8 shows the sample 4-bit x 4-bit layout of the implant-mask NAND ROM array. Here, vertical columns of n-type diffusion intersect at regular intervals with horizontal rows of

polysilicon, which results in an nMOS transistor at each intersection point. However, the access time is usually slower than the NOR ROM, due to multiple-series connected nMOS transistors in each column. An alternative layout locations, as in the case of the PLA. Layout generation. In this case, the missing transistor is simply replaced by a metal line, instead of using a threshold voltage implant at that location.

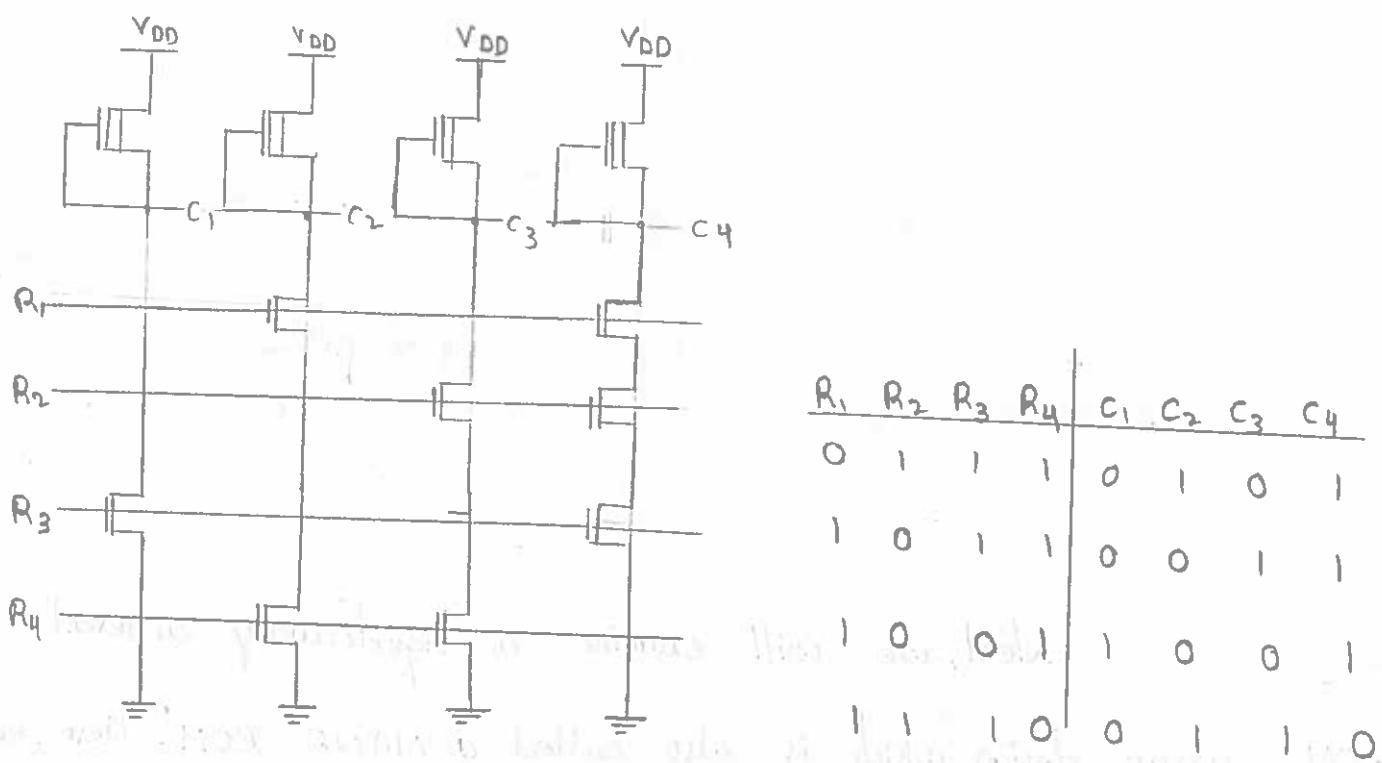


Figure 15.8 A 4-bit x 4-bit NAND-based ROM array.

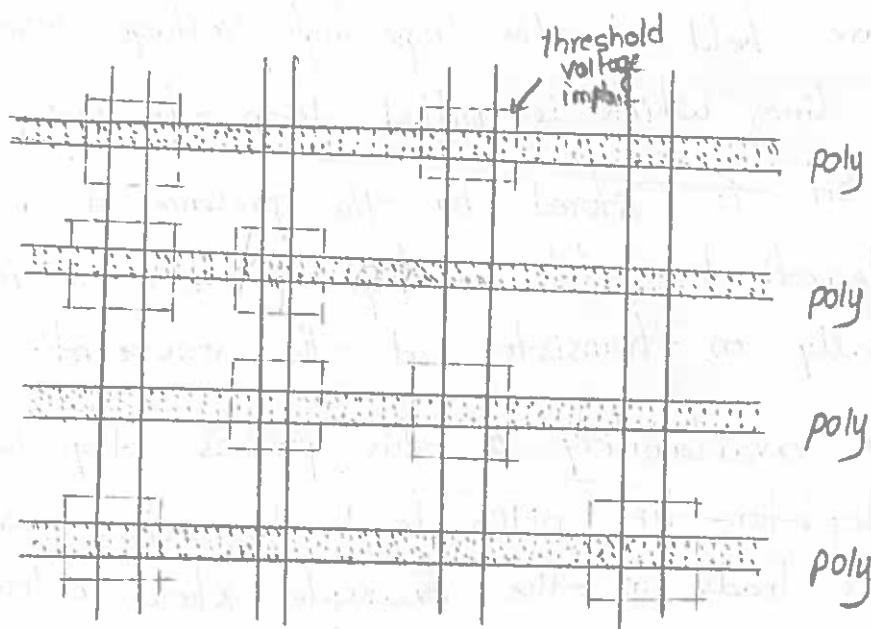
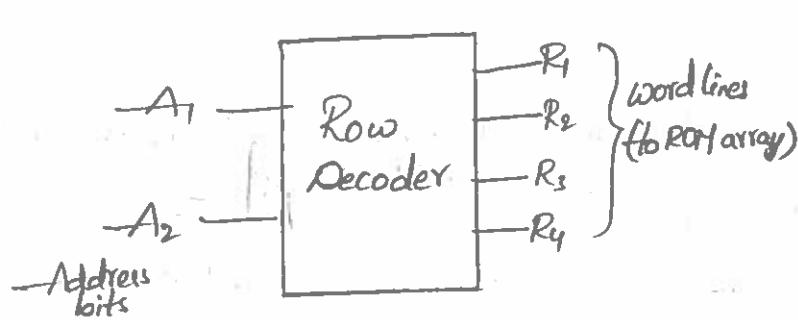


figure 5.9: Implant-mark layout of the NAND ROM array ex in fig 5.8.

The threshold voltages of "0"-bit transistors are lowered below 0V through implant

Now we will turn our attention to the circuit structures of row and column address decoders, which select a particular memory location in the array, based on the binary row and column addresses. A row decoder designed to drive a NOR ROM array must, by definition, select one of the 2^N word lines by raising its voltage to V_{DD} .



$A_1 \ A_2$	R_1	R_2	R_3	R_4
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1

Fig 5.10: Row address decoder example for 2 address bits and 4 word lines.

A most straightforward implementation of this decoder is another NOR array, consisting of 4 rows and 4 columns. Note that this NOR-based decoder array can be built just like the NOR ROM array, using the same selective programming approach.

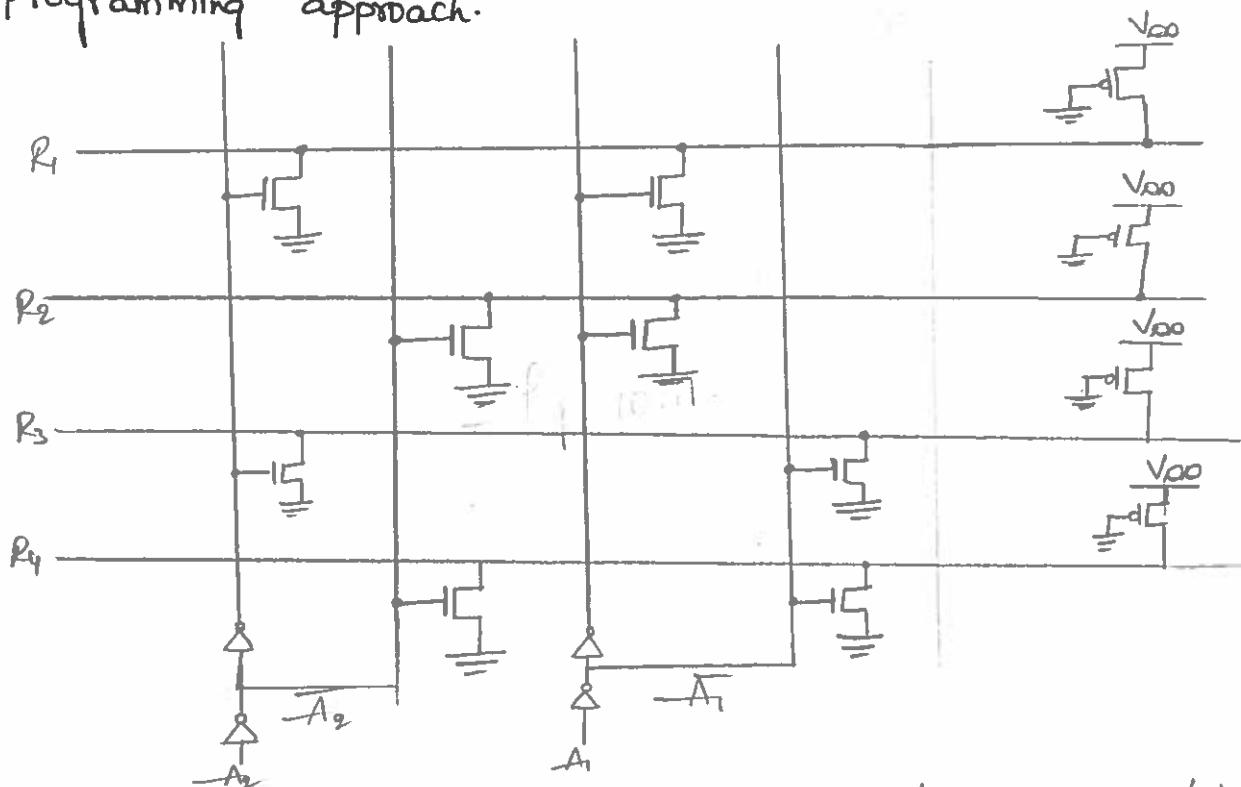


Fig 5.11: NOR-based row decoder circuit for 2 address bits and 4 word lines.

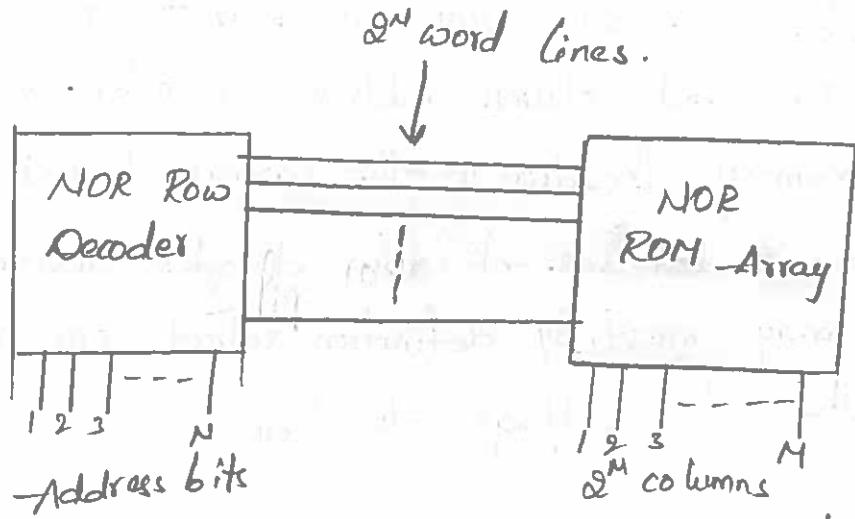


fig 5.12: Implementation of the row decoder circuit and the ROM array as two adjacent NOR planes.

A row decoder designed to drive a NAND ROM, on the other hand, must lower the voltage level of the selected row - to logic "0" while keeping all other rows at a logic-high level. This function can be implemented by using an N -input NAND gate for each of the row outputs. The truth table of a simple address decoder - for four rows and the double NAND-array implementation of the decoder and the ROM are shown in fig.

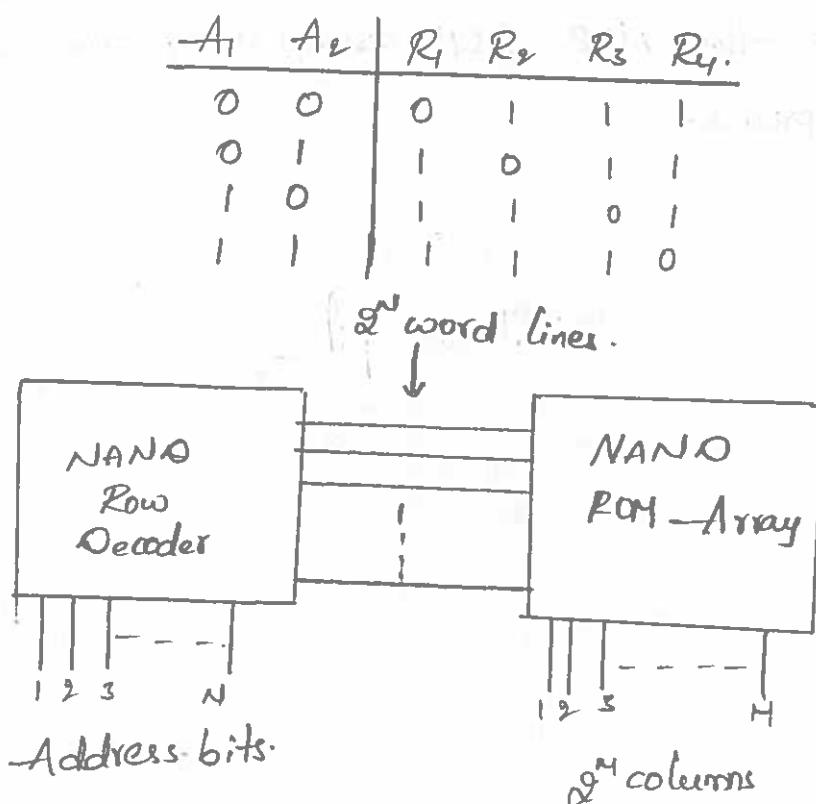
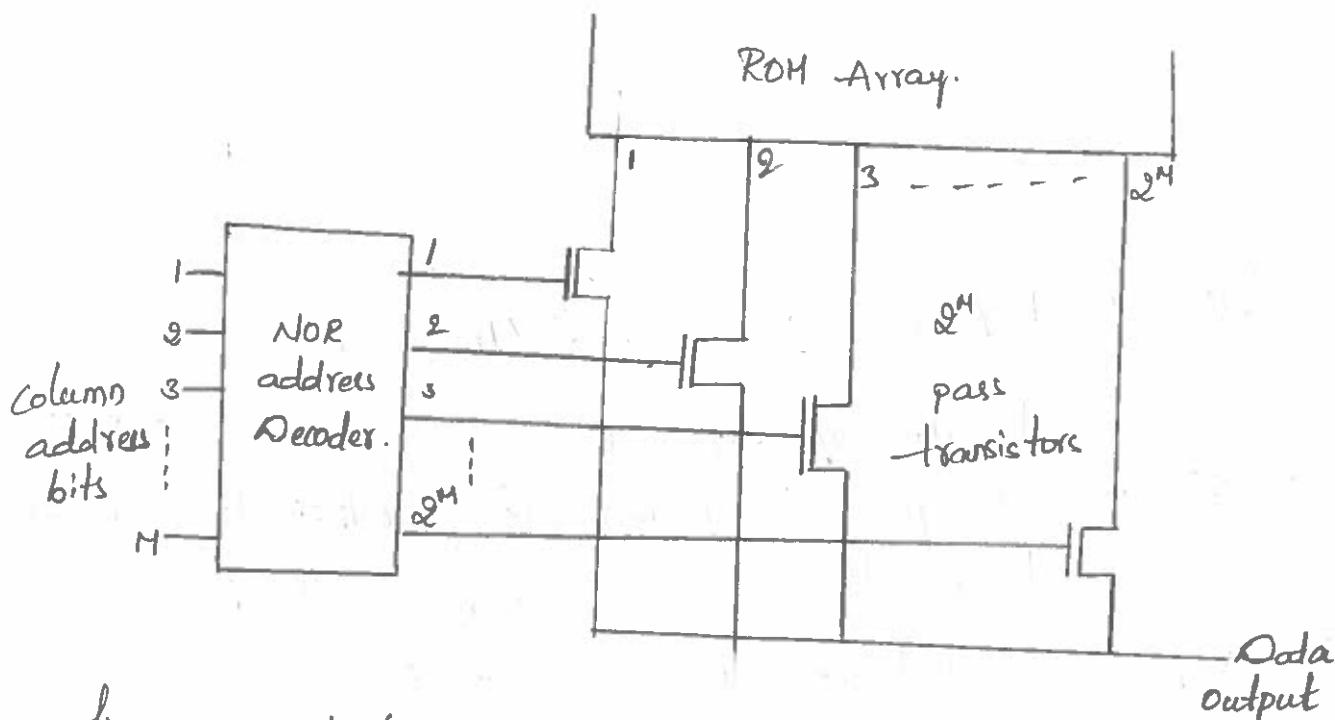


fig 5.13: Truth table of a row decoder for the NAND ROM array, and implementation of the row decoder circuit and the ROM array as two adjacent NAND planes.

(6)

The column decoder circuitry is designed to select one out of 2^M bit lines of the ROM array according to an M-bit column address, and to route the data content of the selected line to the data output. A straightforward but costly approach would be to connect an nMOS pass transistor to each bit-line output, and to selectively drive one out of 2^M pass transistors by using a NOR-based column address decoder, as shown in fig. Similarly, a number of columns can be chosen at a time, and the selected columns can be chosen at a time, and the selected columns can be routed to a parallel data output port.

Note that the number of transistors required for this column decoder implementation is $2^M(M+1)$, i.e., 2^M pass transistors for each bit line and $M 2^M$ transistors for the decoder circuit. This number can quickly become excessive for M , i.e., for a large number of bit lines.



→ fig 5.14: Bit-line (column) decoder arrangement using a NOR address decoder and nMOS pass transistors for every bit line.

~

An alternative design of the column decoder circuit is to build a binary Selection tree consisting of consecutive stages as shown in fig. 10.15 In this case, the pass transistor network is used to select one out of every two bit lines at each stage (level), whereas the column address bits.

One drawback of the decoder tree approach is that the number of series-connected nMOS pass transistors in the data path is equal to the number of columns address bits, M.

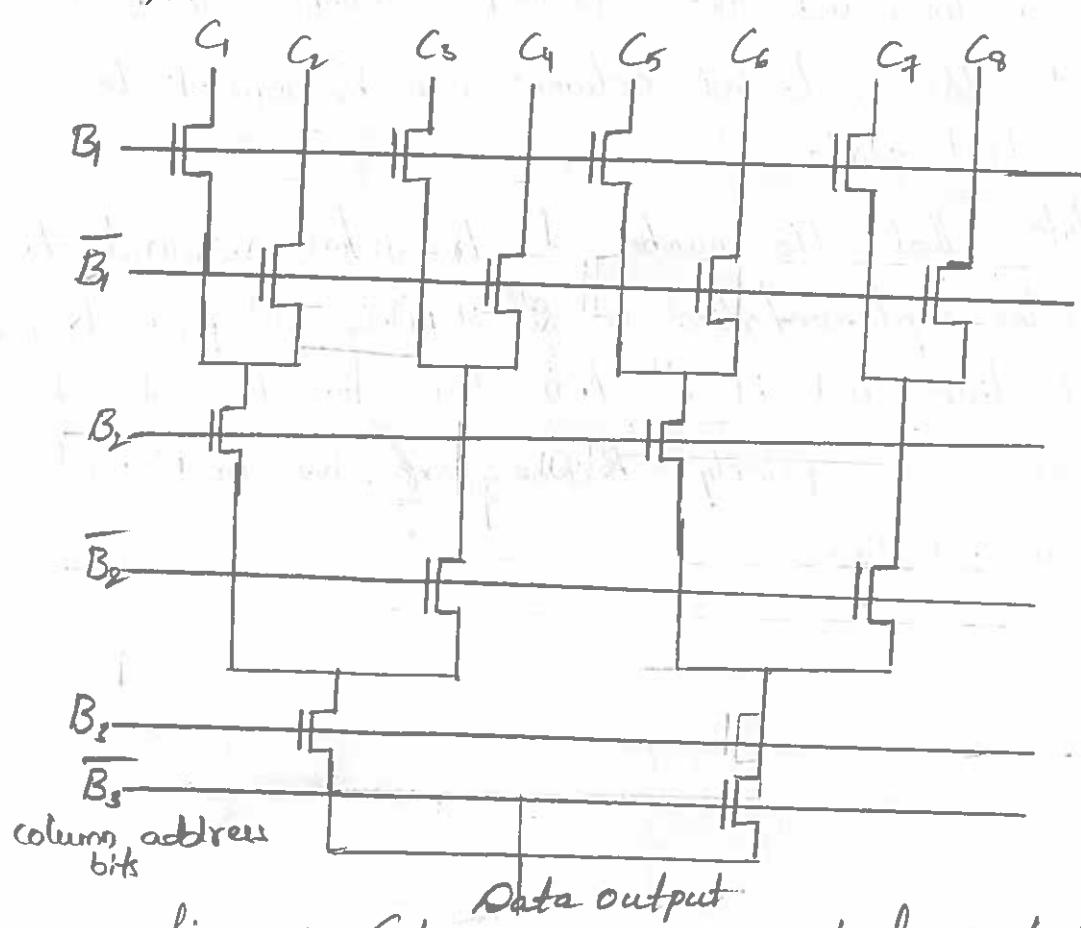


fig 5.15: Column decoder circuit for 8 bit lines.

Ramatic Read-Write Memory (SRAM) Circuits:-

As already explained in question, read-write (R/W) memory circuits are designed to permit the modification of data bits to be stored in the memory array, as well as their retrieval (reading) on demand. The memory circuit is used to be static if the stored data can be retained indefinitely, without any need of a periodic refresh operation. We will examine the circuit structure and the operation of simple SRAM cells, as well as the peripheral circuits designed to read and write the data.

The data storage cell, i.e., the 1-bit memory cell in static RAM arrays, invariably consists of a simple latch circuit with two stable operating points. Depending on the preserved state of the two-inverter latch circuit, the data being held in the memory cell will be interpreted either as a logic "0" or as a logic "1". To access the data controlled by the corresponding word line, i.e., the row address selection signal. This can be likened to turning the car steering wheel with both left and right hands in complementary directions.

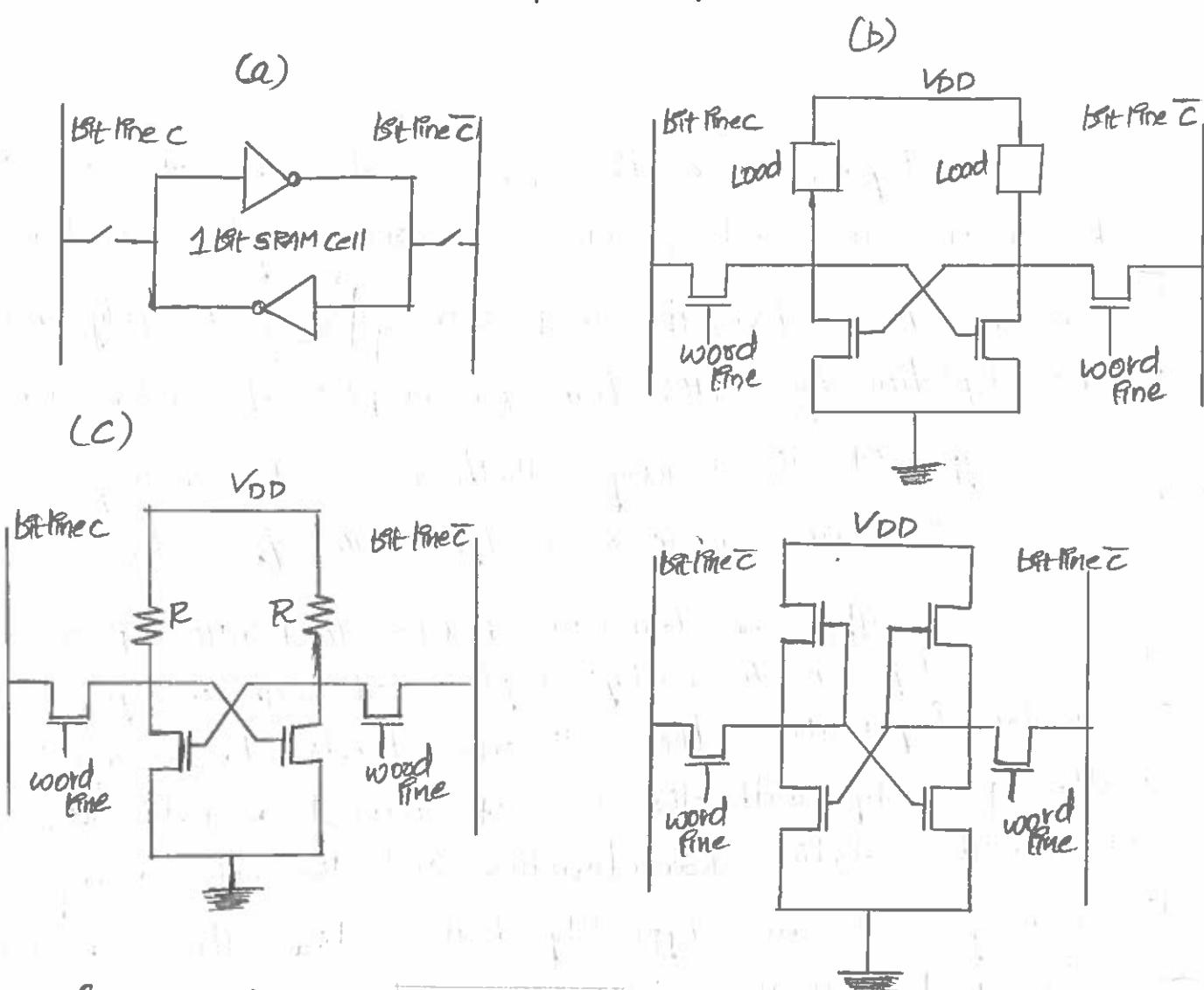


fig: 5.21

- Symbolic representation of 2 Inverter latch ckt with access switches.
- Generic circuit topology of the MOS static RAM cell.
- Resistive-load SRAM cell.
- Depletion-load nMOS SRAM cell.

(e)

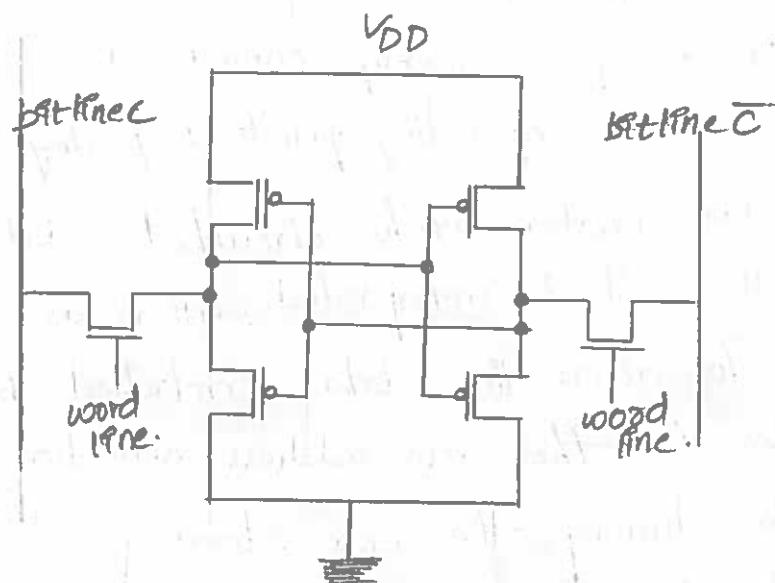


fig:-5.21

e) Full CMOS SRAM cell.

figure shows the generic structure of the MOS static RAM cell, consisting of two cross-coupled inverters and two access transistors. The load devices may be polysilicon resistors, depletion-type nMOS transistors, or pMOS transistors, depending on the type of the memory cell. The pass gates acting as data access switches are enhancement-type nMOS transistors.

shown in fig. The six-transistor depletion-load nMOS SRAM cell can be easily implemented with one polysilicon and one metal layer, and the cell size tends to be relatively small, especially with the use of buried metal-diffusion contacts. The static characteristics and the noise margins of this memory cell are typically better than those of the resistive-load cell. The static power consumption of the depletion-load SRAM cell, however, makes it an unsuitable candidate for high-density SRAM arrays.

The full CMOS SRAM cell shown in fig achieves the lowest static power dissipation among the various circuit configurations presented here. In addition, the CMOS cell offers superior noise margins and switching speed as well. The comparative advantages and disadvantages of the

(R)

CMOS static RAM cell will be investigated in depth later in this section.

SRAM Operation Principles:

Fig shows a typical four-transistor resistive-load SRAM cell widely used in high-density memory arrays, consisting of a pair of cross-coupled inverters. The two stable operating points of this basic latch circuit are used to stored a one-bit piece of information; hence, this pair of cross-coupled inverters make up the central component of the SRAM cell. To perform read and write operations, we use two nMOS pass-transistors, both of which are driven by the row select signal, RS.

When the word line (RS) is not selected, i.e., when the voltage level of line RS is equal to logic "0", the pass transistors M3 and M4 are turned off. The simple latch circuit consisting of two cross-connected inverters preserves one of its two stable operating points; hence, data is being held. At this point, consider the two columns, C and \bar{C} .

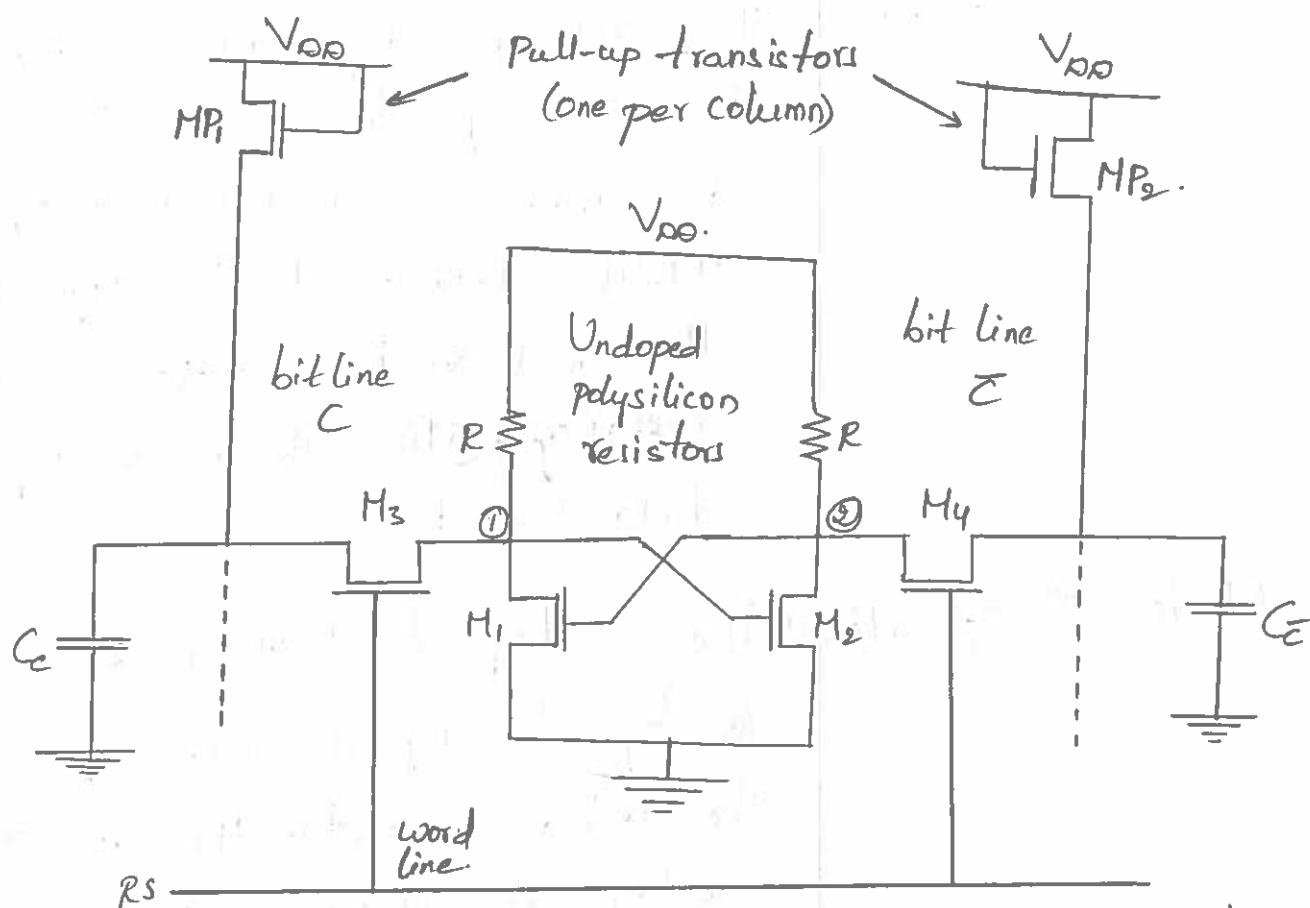


Fig 5.22: Basic structure of the resistive-load SRAM cell, shown with the column pull-up transistors.

If all word lines in the SRAM array are inactive, the relatively large column capacitances are charged-up by the column pull-up transistors, M_{P1} and M_{P2}. Since both transistors operate in saturation, the steady-state voltage level V_C for both columns is determined by the following relationship:

$$V_{DD} - V_C = V_{T0} + \gamma (\sqrt{|2Q_f| + V_C} - \sqrt{|2Q_f|}).$$

Assuming $V_{DD} = 5V$, $V_{T0} = 1V$, $|2Q_f| = 0.6V$, and $\gamma = 0.4V^{1/2}$, this voltage level is found to be approximately equal to 3.5V. Note that the voltage levels of the two complementary bit lines (columns) are equal during this phase.

- a) Write "1" operation: The voltage level of column \bar{C} is forced to logic-low by the data-write circuitry. The driver transistor M₁ turns off. The voltage V_1 attains a logic-high level, while V_2 goes low.
 - b) Read "0" operation: The voltage of column C retains its precharge level while the voltage of column \bar{C} is pulled down by M₂ and M₄. The data-read circuitry detects the small voltage difference ($V_C > V_{\bar{C}}$) and amplifies it as a logic "1" data output.
 - c) Write "0" operation: The voltage level of column C is forced to logic-low by the data-write circuitry. The driver transistor M₂ turns off. The voltage V_2 attains a logic-high level, while V_1 goes low.
- Unit-5, Pg - 16/40

d) Read "0" operation: The voltage of column \bar{C} retains its precharge level while the voltage of column C is pulled down by M_1 and M_3 . The data-read circuitry detects the small voltage difference ($V_C < V_{\bar{C}}$) and amplifies it a logic "0" data output.

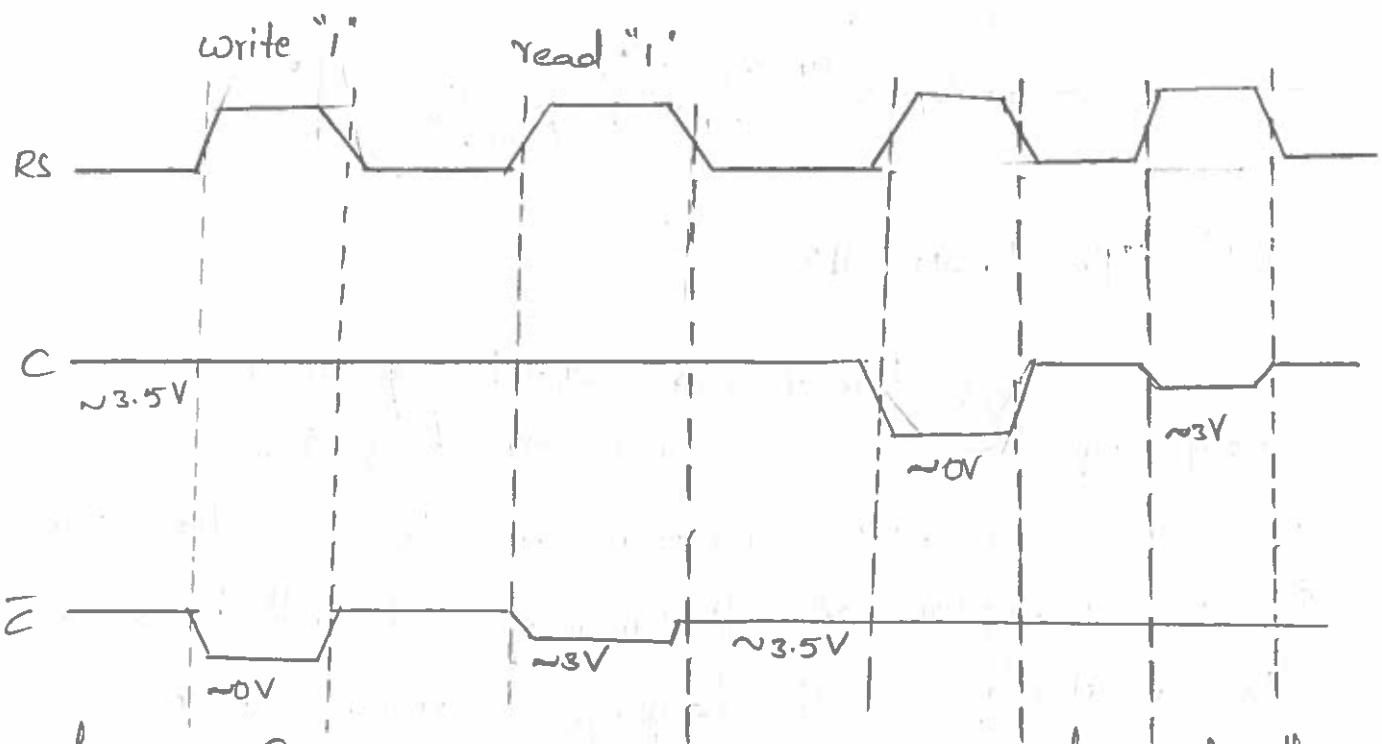


Fig 5.23: Typical row and column voltage waveforms of the resistive-load SRAM shown in fig 5.22 during read and write.

Power Consumption:

To estimate the standby power consumption of the static read-write memory cell, assume that a "1"-bit is stored in the cell. This means that the driver transistor M_1 is turned off, while the driver transistor M_2 is conducting, resulting in $V_i = V_{OH}$ and $V_o = V_{OL}$. In this circuit, one of the load transistors will always conduct a non-zero current and, consequently, consume steady-state power. The amount of the standby power consumption is ultimately determined by the value of the load resistor.

Large resistance values and a smaller cell size can be achieved by using lightly-doped or undoped

Polysilicon — for the load resistors, which has a typical sheet resistivity of $10\text{M}\Omega$ per square or higher. The added process complexity for implementing resistive-load SRAM cells using undoped poly resistors is usually worth the advantage of low-power operation, which is manifested by a standby power dissipation of the resistive SRAM cell shown in Fig becomes.

$$P_{\text{stand-by}} = V_{\text{DD}} \cdot \frac{\mu_n C_{\text{ox}}}{2} \cdot \left(\frac{W}{L} \right)_{\text{load}} |V_{\text{i,load}}|^2$$

Full CMOS SRAM cell:

A low-power SRAM cell may be designed simply by using cross-coupled CMOS inverters instead of the resistive-load nMOS inverters. In this case, the stand-by power consumption of the memory cell will be limited to the relatively small leakage currents of both CMOS inverters. The possible drawback of using CMOS SRAM cells, on the other hand, is that the cell area tends to increase in order to accommodate the n-well for the PMOS transistors and the polysilicon contacts.

Other advantages of CMOS SRAM cells include high-density noise immunity due to larger noise margins, and the ability to operate at lower power supply voltages than, for example, the resistive-load SRAM cells. The major disadvantages of CMOS memories historically were larger cell size, the added complexity of the CMOS memories historically and the tendency to exhibit "latch-up" phenomena. It compares typical layouts of the four-transistor resistive-load SRAM cell and the six-transistor full CMOS SRAM cell.

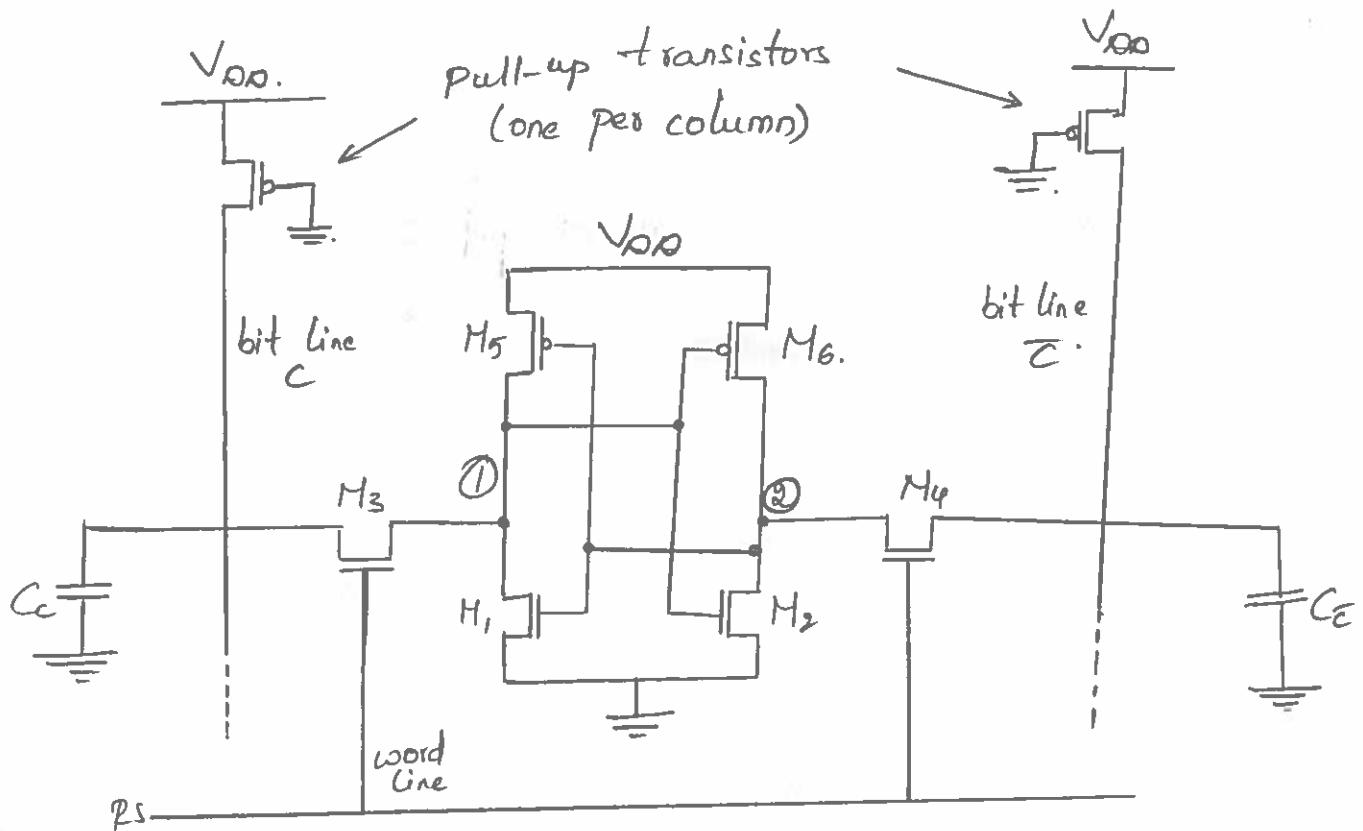


Fig 5.24: Circuit topology of the CMOS SRAM cell.

CMOS SRAM Cell Design Strategy:

To determine the (W/L) ratios of the transistors in a typical CMOS SRAM cell as shown in fig, a number of design criteria must be taken into consideration. The two basic requirements which dictate the (W/L) ratios are: (a) the data-read operation should not destroy the stored information in the SRAM cell, and (b) the cell should allow modification of the stored information during the data-write phase. Here, the transistors M₂ and M₅ are turned off, while the transistors M₁ and M₆ operate in the linear mode. Thus, the internal node voltages V_t=0 and V_g=V_{DD} before the cell access transistors M₃ and M₄ are turned on. The active transistors at the beginning of the data-read operation are highlighted in fig.

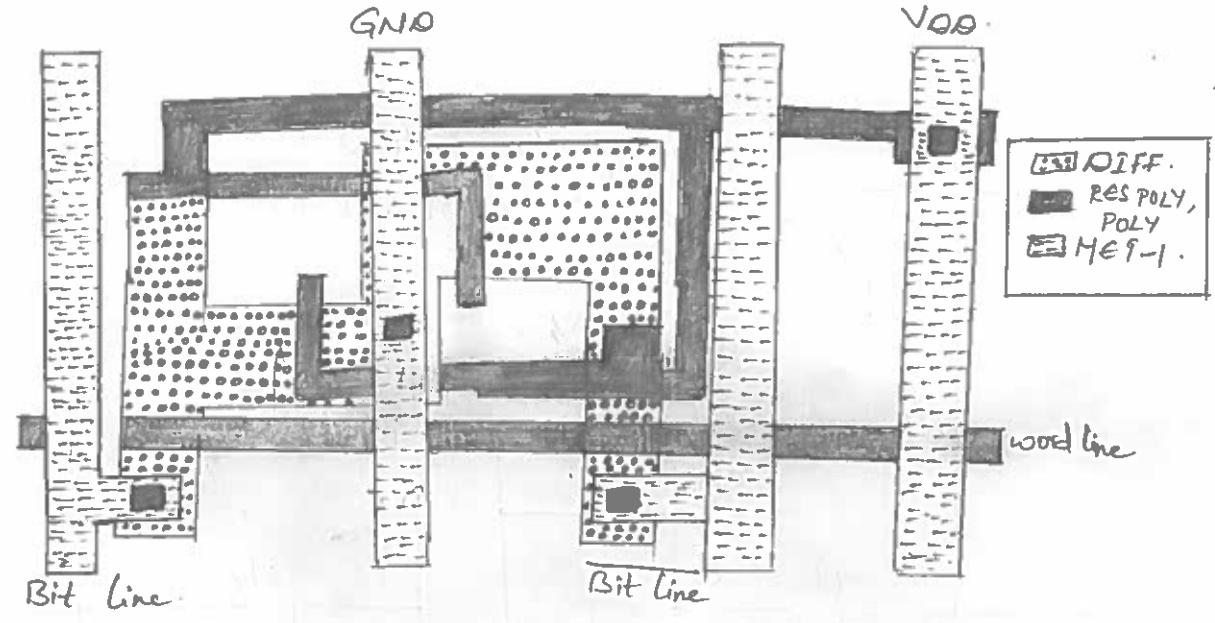


fig 10.25. layout of the resistive-load SRAM cell.

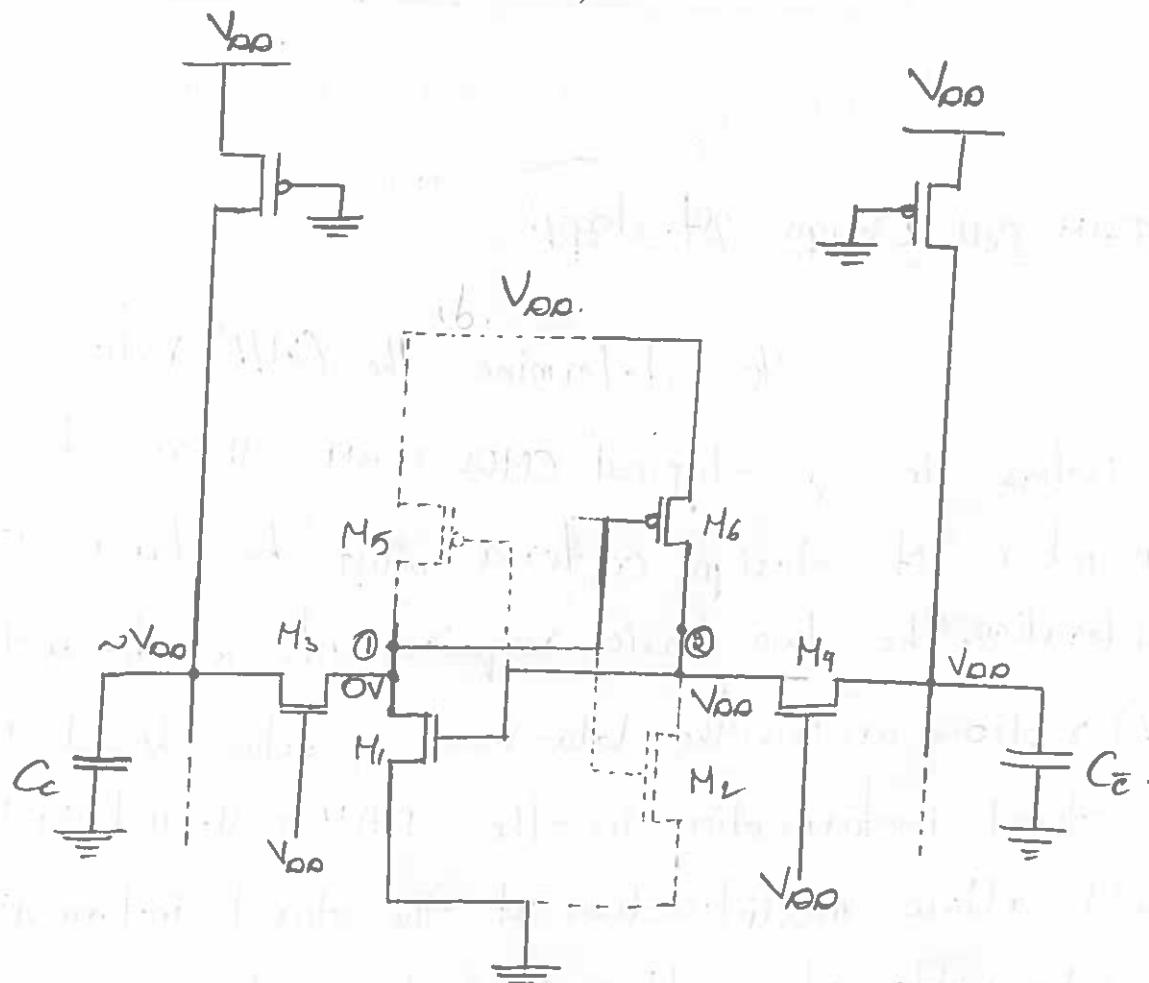


fig 5.26 Voltage levels in the SRAM cell at the beginning of the "read" operation.

(11)

While M_1 and M_3 are slowly discharging the column capacitance, the node voltage V_1 will increase from its initial value of 0V. Especially if the (W/L) ratio of the access-transistor M_3 is large compared to the (W/L) ratio of M_1 , the node voltage V_1 may exceed the threshold voltage of M_2 during this process, forcing an unintended change of the stored state. The key design issue for the data-read operation is then to guarantee that the voltage V_1 does not exceed the threshold voltage of M_2 , so that the transistor M_2 remains turned off during the read phase, i.e.,

$$V_{1,\max} \leq V_{T,2}$$

We can assume that after the access transistors are turned on, the column voltage V_c remains approximately equal to V_{DD} . Hence, M_3 operates in saturation while M_1 operates in the linear region.

$$\frac{\alpha_{n,3}}{\alpha_{n,1}} (V_{DD} - V_1 - V_{T,n})^2 = \frac{\alpha_{n,1}}{2} (2(V_{DD} - V_{T,n})V_1 - V_1^2).$$

Combining this eq" with (5.3) results in:

$$\frac{\alpha_{n,3}}{\alpha_{n,1}} = \frac{(W/L)_3}{(W/L)_1} < \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} - 2V_{T,n})^2}.$$

The upper limit of the aspect ratio found above is actually more conservative, since a portion of the drain current of M_3 will also be used to charge-up the parasitic node capacitor of node (1).

Now, consider the write "0" operation, assuming the logic "1" is stored in the SRAM cell initially. Fig shows the voltage levels in the CMOS SRAM cell at the beginning of the data-write operation. The transistors M_1 and M_6 are turned off, while the transistors M_2 and M_5 operate in their linear mode. Thus,

the internal node voltages are $V_1 = V_{DD}$ and $V_2 = 0V$ before the cell access transistors M_3 and M_4 are turned on.

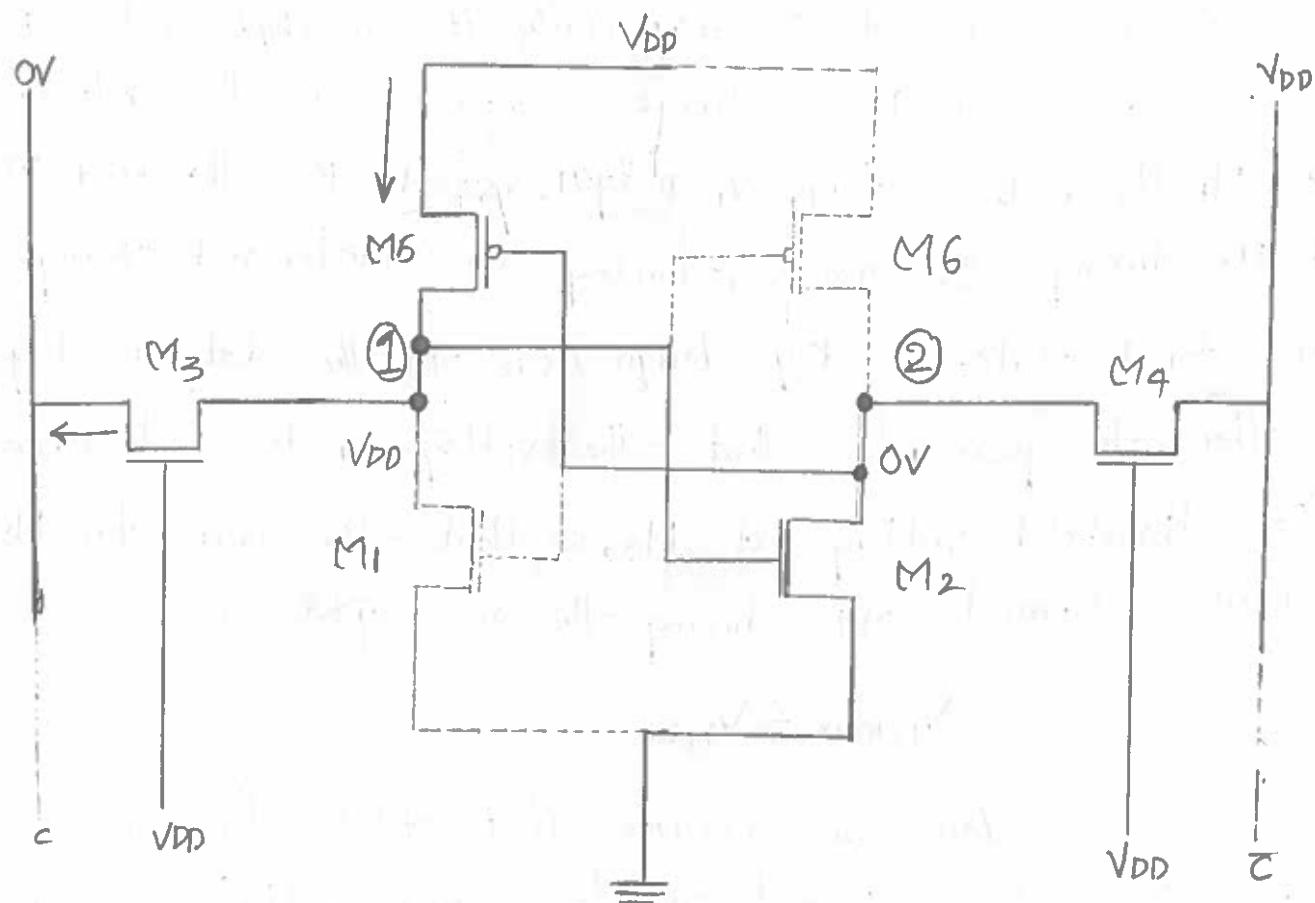


Figure 5.27 voltage Levels in the SRAM cell at the beginning of the "write" operation.

The column voltage V_c is forced to logic "0" level by the data-write circuitry; thus, we may assume that V_c is approximately equal to $0V$. Once the pass transistors M_3 and M_4 are turned on by the row selection circuitry, we expect that the node voltage V_2 remains below the threshold voltage of M_1 , since M_2 and M_4 are designed to according condition. Consequently, the voltage level at node(2) would not be sufficient to turn on M_1 . To change the stored information, i.e., to force V_1 to $0V$ and V_2 to V_{DD} , the node voltage V_1 must be reduced below the threshold voltage of M_2 , so that M_2 turns off first. When $V_1 = V_{T,n}$, the transistor M_5 operates in the linear region while M_5 operates in saturation.

$$\frac{k_{P,5}}{2} (0 - V_{DD} - V_{T,P})^2 = \frac{k_{n,3}}{2} (2(V_{DD} - V_{T,n})V_{T,n} - V_{T,n}^2).$$

Rearranging this condition results in:

$$\frac{K_{P,5}}{K_{n,3}} \leq \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2}$$

$$\frac{(W/L)_5}{(W/L)_3} \leq \frac{\mu_n}{\mu_p} \cdot \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,n})^2}$$

SRAM Write Circuit:

A "write" operation is performed by forcing the voltage level of either column to a logic-low level. To accomplish this task, a low-resistance, conducting path must be provided from each column to the ground, which can be selectively activated by the data-write signals. A simplified view of the SRAM "write" circuitry designed for this operation is shown in fig. The column pull-down transistors, on the other hand, are driven by two pseudo-complementary control signals, WB and \overline{WB} . The "write-enable" signal, W and the data to be written (DATA) are used to generate the control signals, as shown in fig.

The nMOS pull-down transistors M_1 and M_2 , as well as the column selection transistor M_3 must have sufficiently large (W/L) ratios so that the column voltages can be forced to almost 0V level during a "write" operation. Assuming that one column is activated, i.e., selected by the column address decoder, at any given time.

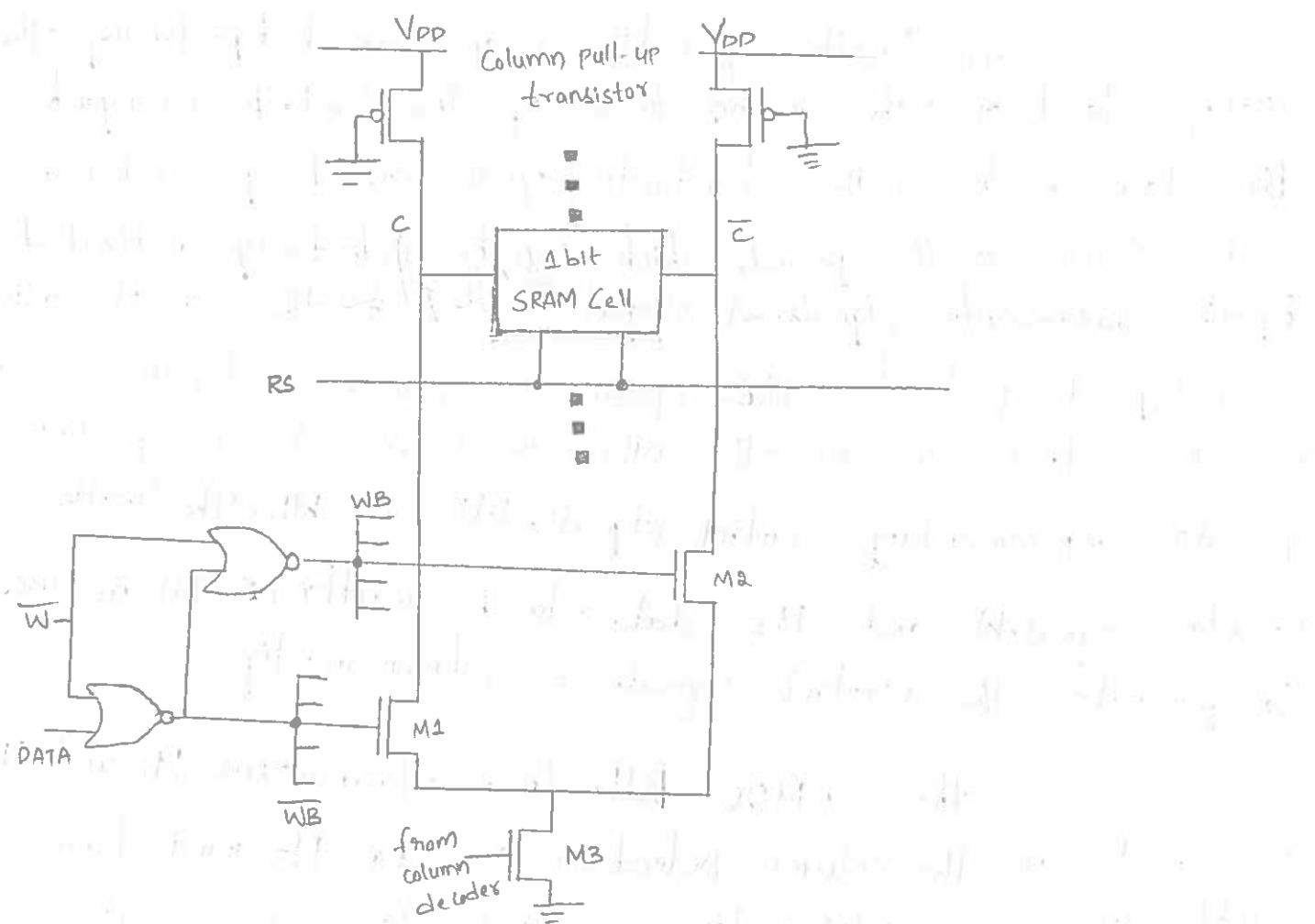
SRAM Read Circuitry:

During the "data-read" operation in the SRAM array, the voltage level on either one of the columns drops slightly after the pass transistors are turned on by the row address decoder circuit. In order to reduce the read

access time, the "read" circuitry must detect a very small voltage difference between the two complementary columns, and amplify this difference to produce a valid logic output level. A simple source-coupled differential amplifier can be used for this task, as shown in fig. Here, the drain currents of the two complementary nMOS transistors are:

$$I_{D,1} = \frac{k_n}{2} (V_c - V_x - V_{T,1})^2.$$

$$I_{D,2} = \frac{k_n}{2} (V_c - V_x - V_{T,2})^2.$$



\bar{W}	DATA	WB	\bar{WB}	Operation
0	1	1	0	M1 is off, M2 is on $\rightarrow V_c$ low
0	0	0	1	M1 is on, M2 is off $\rightarrow V_c$ low
1	X	0	0	M1 and M2 are off \rightarrow both columns remain high

Figure 5.28:- Data write circuitry associated with one column-pair in an SRAM array.

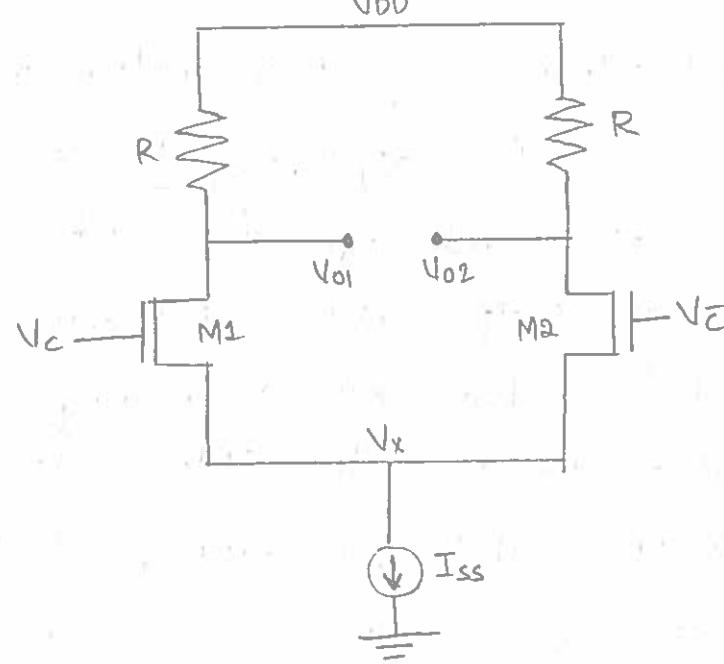


Fig 5.29: Simple Source Coupled differential amplifier circuit for "read" operation

Small-signal analysis of the circuit yields the differential gain of this circuit as

$$\frac{\delta(V_{o1} - V_{o2})}{\delta(V_c - V_x)} = -R \cdot g_m.$$

where

$$g_m = \frac{\delta I_D}{\delta V_{GS}} = \sqrt{2k_b I_D}.$$

The differential gain of the amplifier can be increased significantly by using active loads instead of resistors and by using cascode configuration, i.e., an intermediary common-gate stage between the common source transistors and the load transistors. Although the "data read" circuitry described above is capable of detecting small voltage differences between the two complementary bit lines, other types of efficient sense amplifiers are also implemented with CMOS technology, as will be examined in the following.

The architecture of the output "read" circuitry is driven primarily by the constant demand for high access speed and high integration density. We must recognize first

that the full CMOS SRAM cell has a natural speed advantage which is derived from using both active pull-up and active pull-down devices in the latch circuits. The CMOS rise and fall times are short and symmetrical, whereas the nMOS latch circuit has a short output nMOS SRAM cell and the resistive-to-nMOS SRAM cell, hence, have slower average switching speed compared to that for the full CMOS cell. Apart from the type of the SRAM cell, the precharging of bit lines also plays a significant role in the access time.

The access time penalty associated with this procedure can be substantially reduced by the equalization of bit lines prior to each new access. Equalization can be done when the memory array is deselected, i.e., between two access cycles.

Dual-Port Static RAM Arrays:

In some cases, the memory array may have to be accessed simultaneously by multiple processors or by one processor and another peripheral device. This could result in a timing conflict called "contention", which can be resolved only by having one of the processors wait until the SRAM is free. The added wait state, however, significantly reduces the advantages of the high-speed processor. The dual-port RAM architecture is implemented in the systems in which a main memory array must serve multiple high-speed processors and peripheral devices with minimum delay.

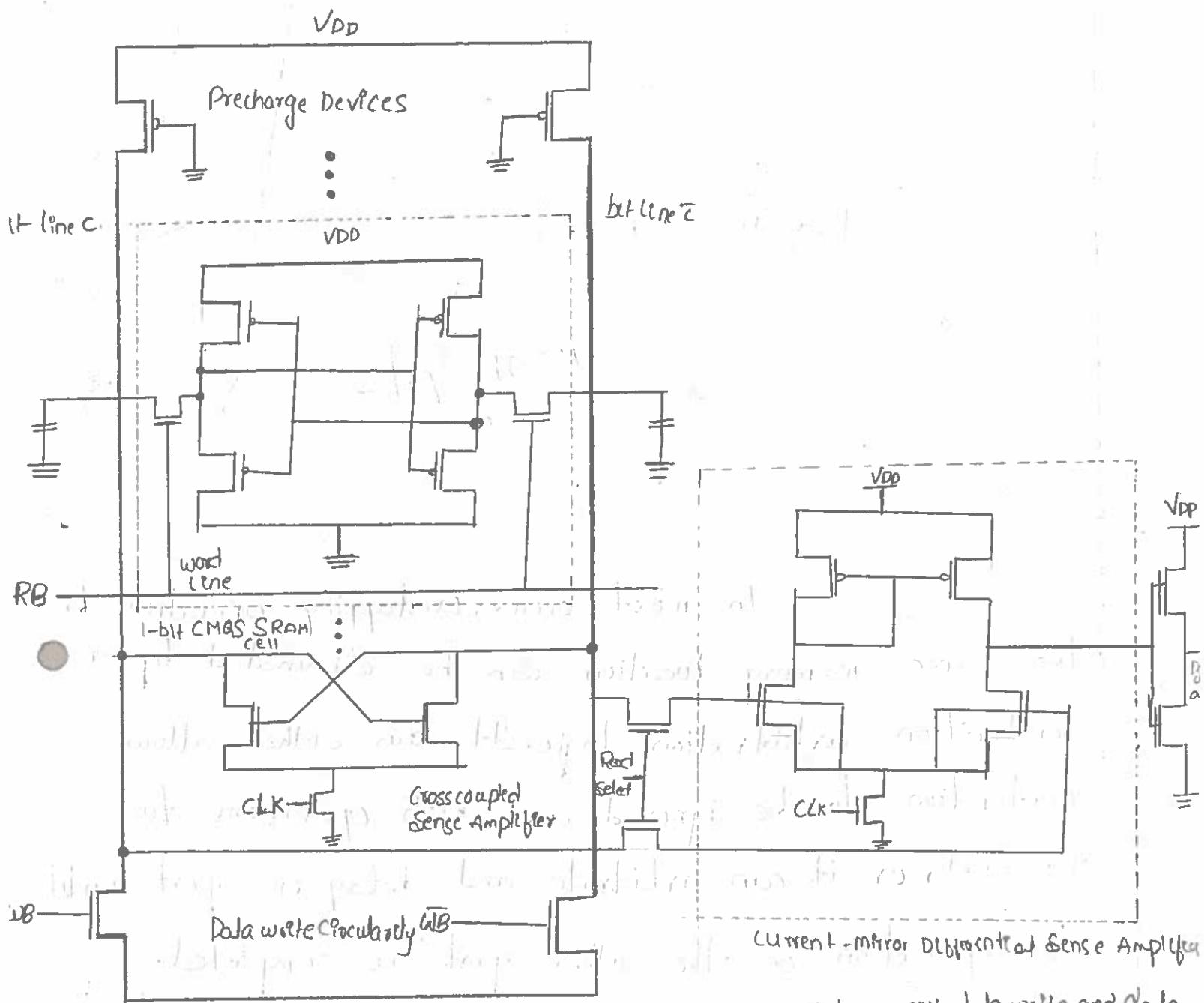


Figure: 15.34 Complete circuit diagram of CMOS static RAM column with data write and data read capability. The ideal dual-port SRAM allows simultaneous access to the same location in the memory array, by using two independent sets of bit lines and associated access switches for each memory cell. The circuit structure of a typical CMOS dual-port SRAM cell is shown in fig. Here, "word line" is used to access one set of complement-ary bit lines, while "word line \bar{C} " allows access to the other set of bit lines. The capability of simultaneous access eliminates wait states for the processors during "data read Operations".

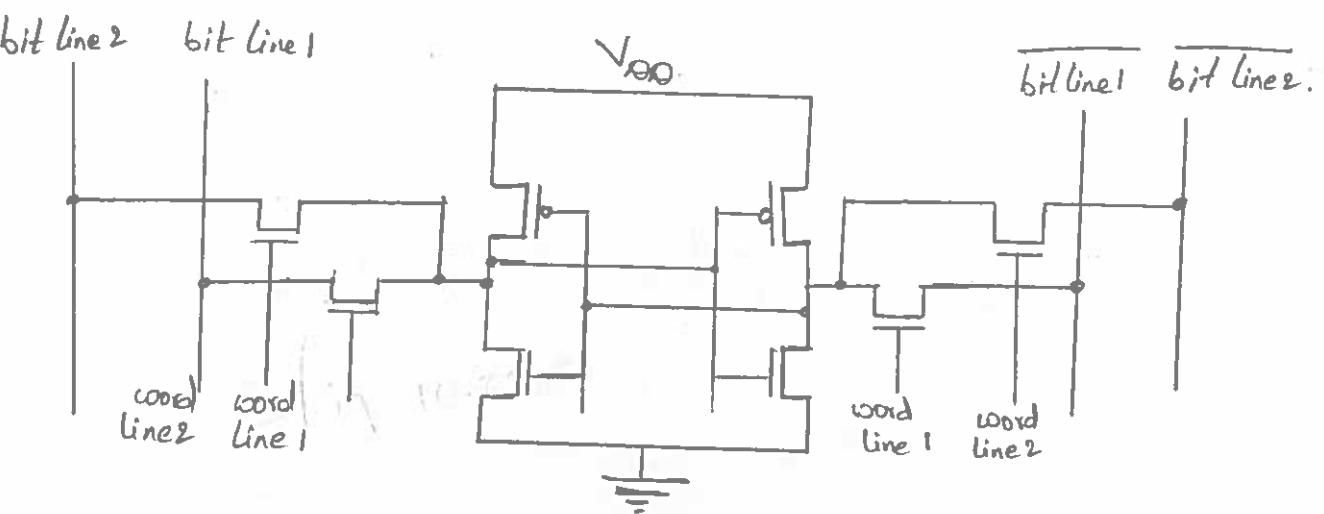


fig 5.35: Circuit diagram of the CMOS dual-port SRAM cell.

In most cases, overlapping operations to the same memory location can be eliminated by a contention arbitration logic. It can either allow contention to be ignored and both operations to proceed, or it can arbitrate and delay one port until the operation on the other port is completed.

Dynamic Read-Write Memory (DRAM) Circuits

All of the static RAM cells examined in the previous section consist of a two-inverter latch circuit, which is accessed for "read" and "write" operations via two pass-transistors. Consequently, the SRAM cells require four to six transistors per bit, and four to five lines connecting to each cell, including the power and ground connections. To satisfy these requirements, a substantial (situation) silicon area must be reserved for each memory cell. In addition, most SRAM cells have non-negligible standby (static) power dissipation, with the exception of the full CMOS SRAM cell.

As the trend for high-density RAM arrays forces the memory cell size to shrink, alternative data storage concepts must be considered to accommodate these demands. In a dynamic RAM cell, binary data is stored simply as charge in a capacitor. Note that the data stored as charge in a capacitor cannot be retained indefinitely, because the leakage currents eventually remove or modify the stored charge.

- The use of a capacitor as the primary storage device generally enables the DRAM cell to be realized on a much smaller silicon area compared to the typical SRAM cell. The DRAM cell must have access devices, or switches, which can be activated externally for "read" & "write" operations. Also, no static power is dissipated for storing charge on the capacitance. Consequently, dynamic RAM arrays can achieve higher integration densities than SRAM arrays.

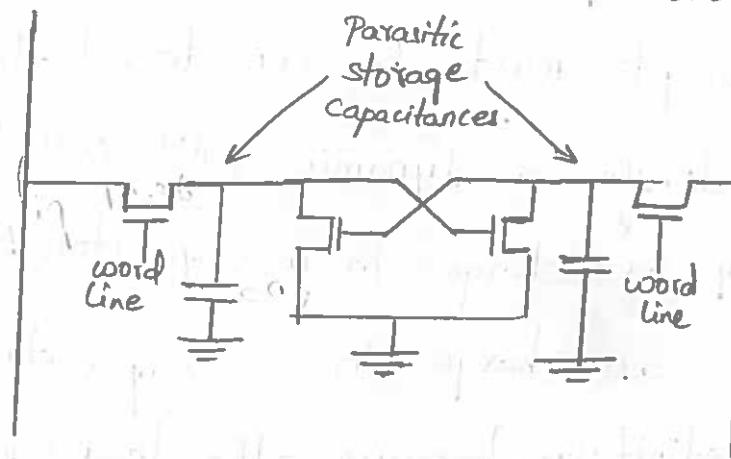
The hardware overhead of the refresh circuitry, however, does not overshadow the area advantage gained by the small cell size.

Figure shows some of the steps in the historical evolution of the DRAM cell. The four-transistor cell shown in fig. is the simplest and one of the earliest dynamic memory cells. This cell is derived from the six-transistor static RAM cell by removing the load devices. The cell has in fact two storage nodes,

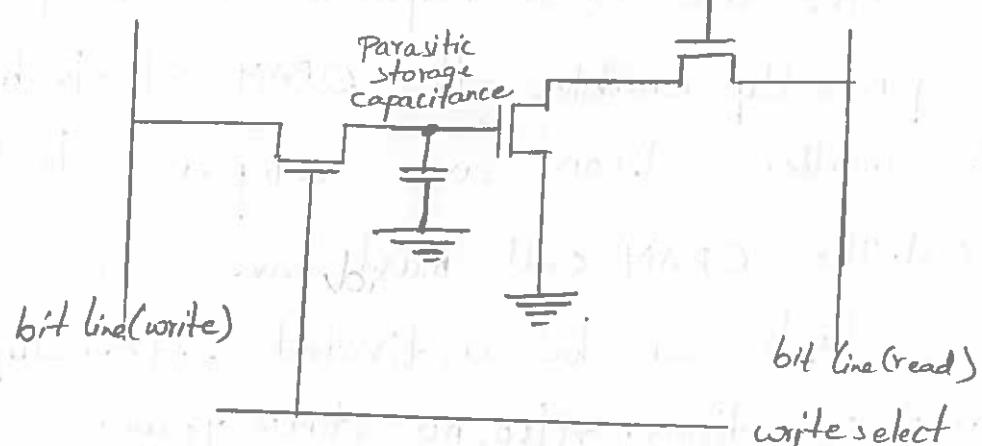
i.e., parasitic oxide and diffusion capacitances of the nodes indicated in the circuit diagram.

bit line C

bit line C.



read select



b,

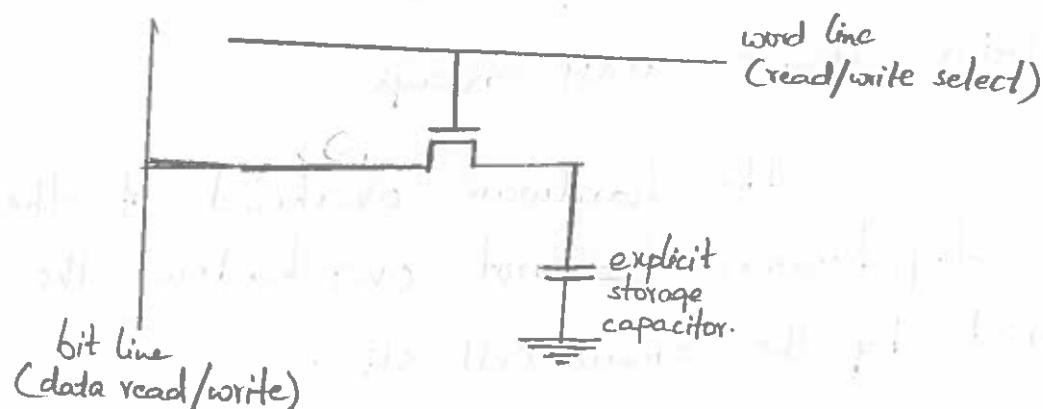


fig 5.36: Various configurations of the dynamic RAM cell.

a, Four-transistor DRAM cell with two storage nodes.

b, Three-transistor DRAM cell with two bit lines and two word lines.

c, One-transistor DRAM cell with one bit line and one word line.

The three-transistor DRAM cell shown in fig. was the first widely used dynamic memory cell. It utilizes a single transistor as the storage device, and one transistor each for the "read" and "write" access switches. The cell has two control and two I/O lines.

The one-transistor DRAM cell shown in fig. has become the industry-standard dynamic RAM cell in high-density DRAM arrays. With only one transistor and one capacitor, it has the lowest component count and, hence, the smallest silicon area of all the dynamic memory cells. The cell has one read-write control line (word line) and one I/O line (bit line). Once the selected transistor is turned on, the charge stored in the capacitor can be detected and/or modified through the bit line.

Three-Transistor DRAM Cell

The circuit diagram of a typical three-transistor dynamic RAM cell is shown in fig. as well as the column pull-up transistors and the column read/write circuitry. Here, the binary information is stored in the form of charge in the parasitic node capacitance C_p .

The operation of the three-transistor DRAM cell and its peripheral circuitry is based on a two-phase non-overlapping clock scheme. The precharge events are driven by ϕ_1 , whereas the "read" and "write" events are driven by ϕ_2 . With typical enhancement-type nMOS pull-up transistors ($V_{TO} \approx 1.0V$) and a power supply voltage of 5V,

-the voltage level of both columns after the precharge is approximately equal to 3.5V.

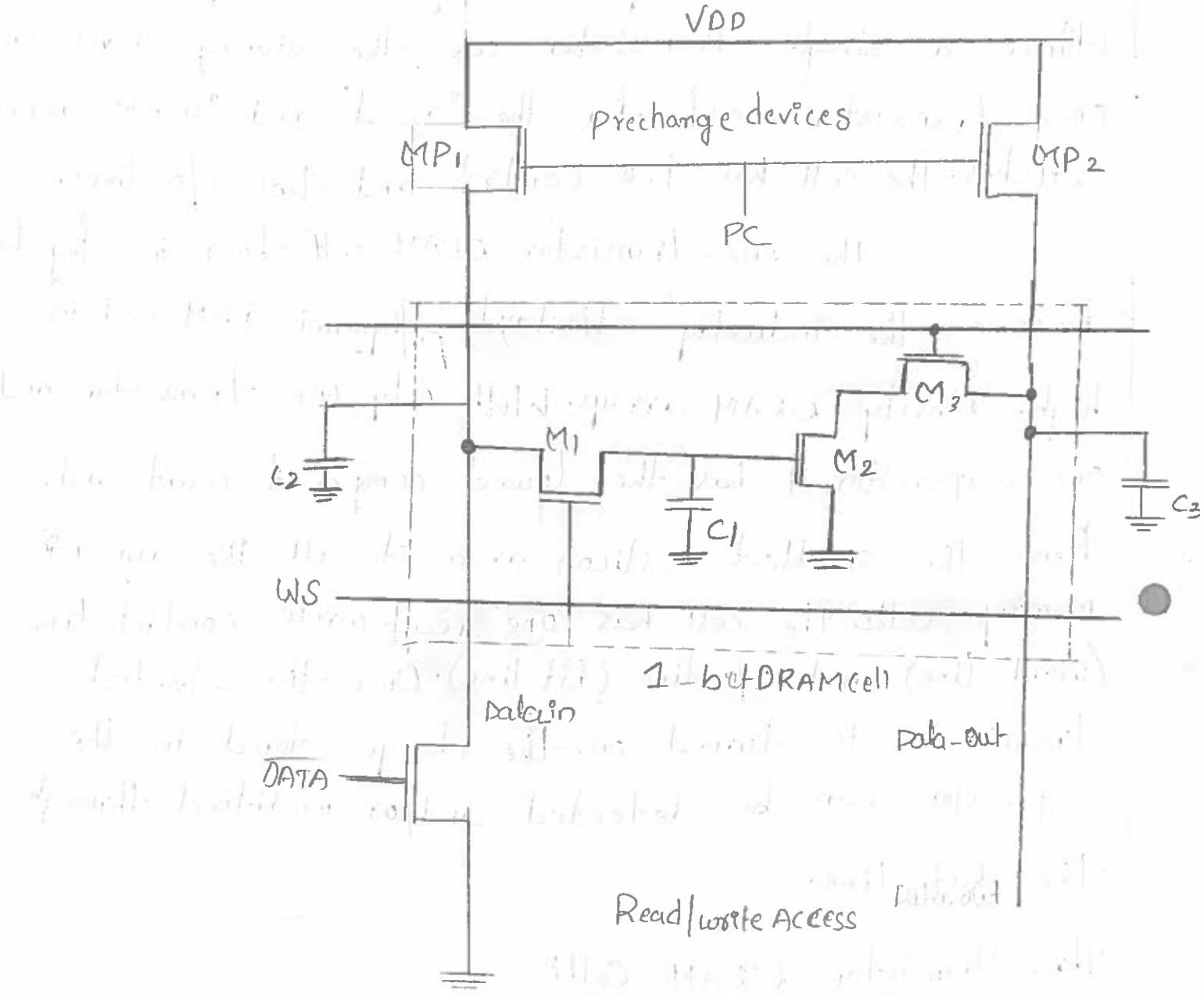


Figure 10.37 Three-transistor DRAM cell with the pull-up and read/write circuitry.

The typical voltage waveforms associated with the 3-T DRAM cell during a sequence of four consecutive operations: write "1", read "1", write "0", and read "0". The four precharge cycles shown in fig are numbered 1, 3, 5, and 7, respectively. Fig illustrates the transient currents charging up the two columns (I_{Din} and I_{Dout}) during a precharge cycle.

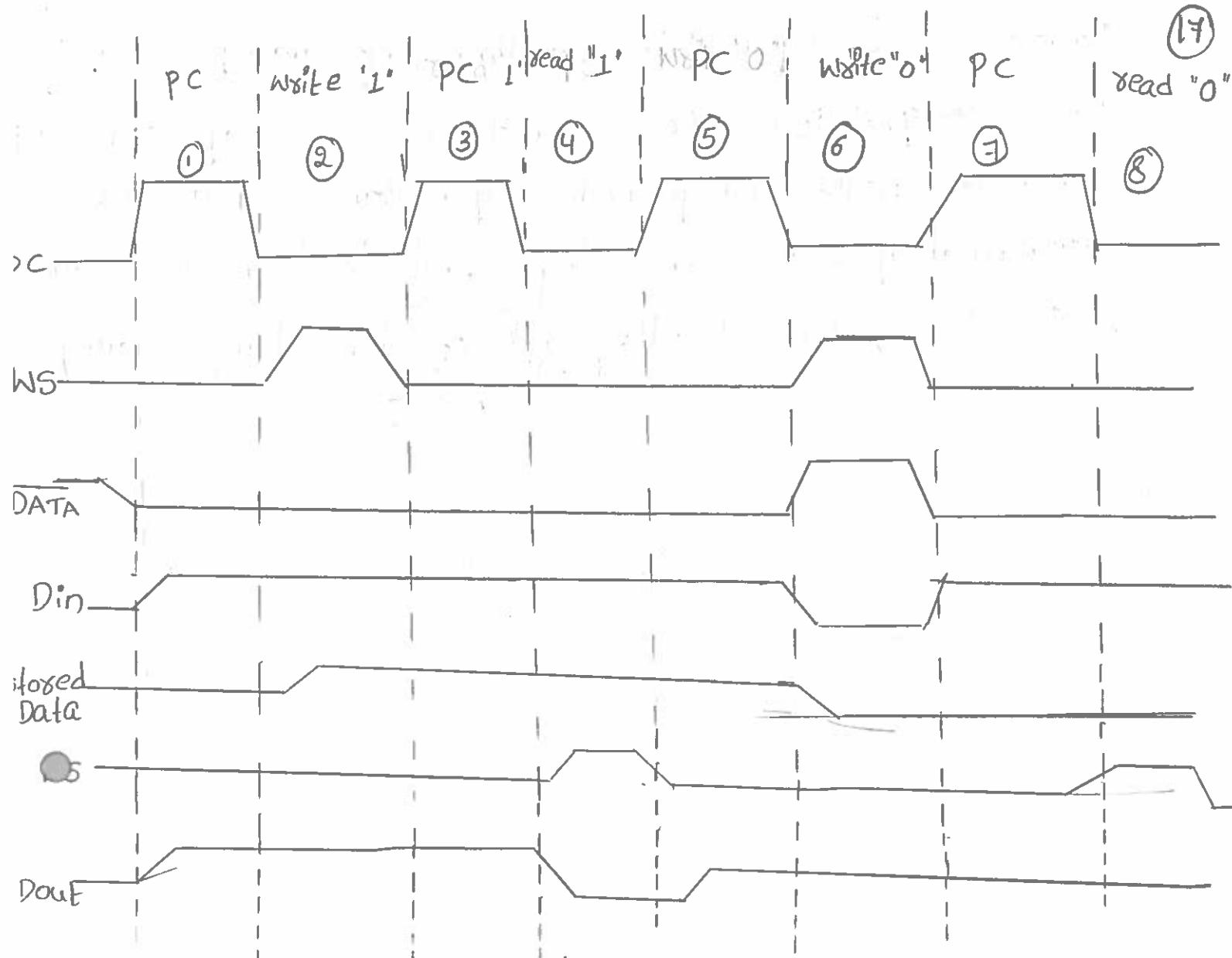


fig 5.38: Typical voltage waveforms associated with the 3-T DRAM cell during four consecutive operations: write "1", read "1", write "0", and read "0".

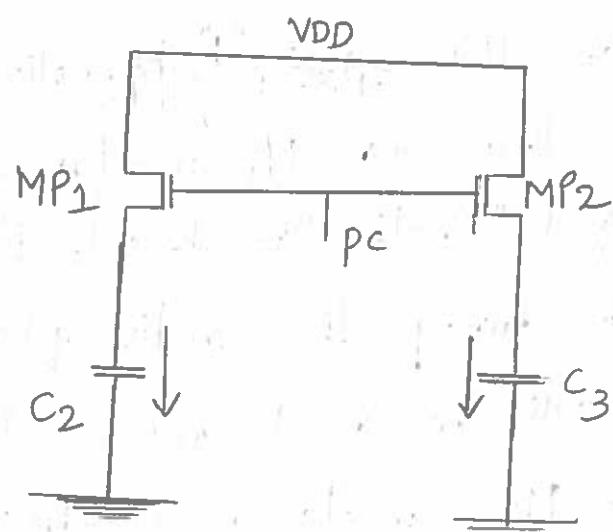


figure 5.39 Column capacitance C₂ and C₃ are being charged-up through MP₁ and MP₂ during the precharge cycle. For the write "1" operation, the inverse data input is at the logic-low level, because the data to be written onto the DRAM cell is logic "1". Consequently, the "data write" transistor MD is turned off, and the voltage level on column Din remains high. Now, the unit - 5, pg - 33/4.

"write select" signal WS is pulled high during the active phase of ϕ_2 . Since the capacitance C_2 is very large compared to C_1 , the storage node capacitance C_1 attains approximately the same logic-high level as the column capacitance C_2 at the end of the charge-sharing process.

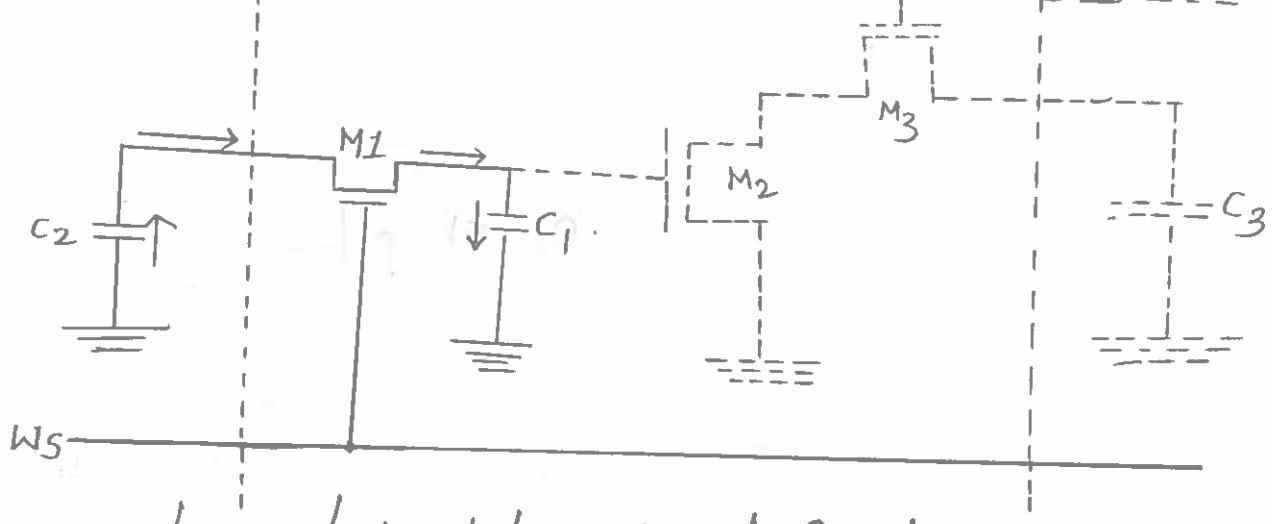


figure 5.40: Charge sharing between C_2 and C_1 during the write "1" sequence

After the write "1" operation is completed, the write access transistor M_1 is turned off. In order to read this stored "1", the "read-select" signal RS must be pulled high during the active phase of ϕ_2 , following a precharge cycle. As the read access transistor M_3 turns on, M_2 and M_3 create a conducting path between the "data read" column capacitance C_3 and the ground. The active portion of the DRAM cell during the read "1" cycle are shown in fig. Note that the 3-T DRAM cell may be read repeatedly in this fashion without disturbing the charge stored in C_1 .

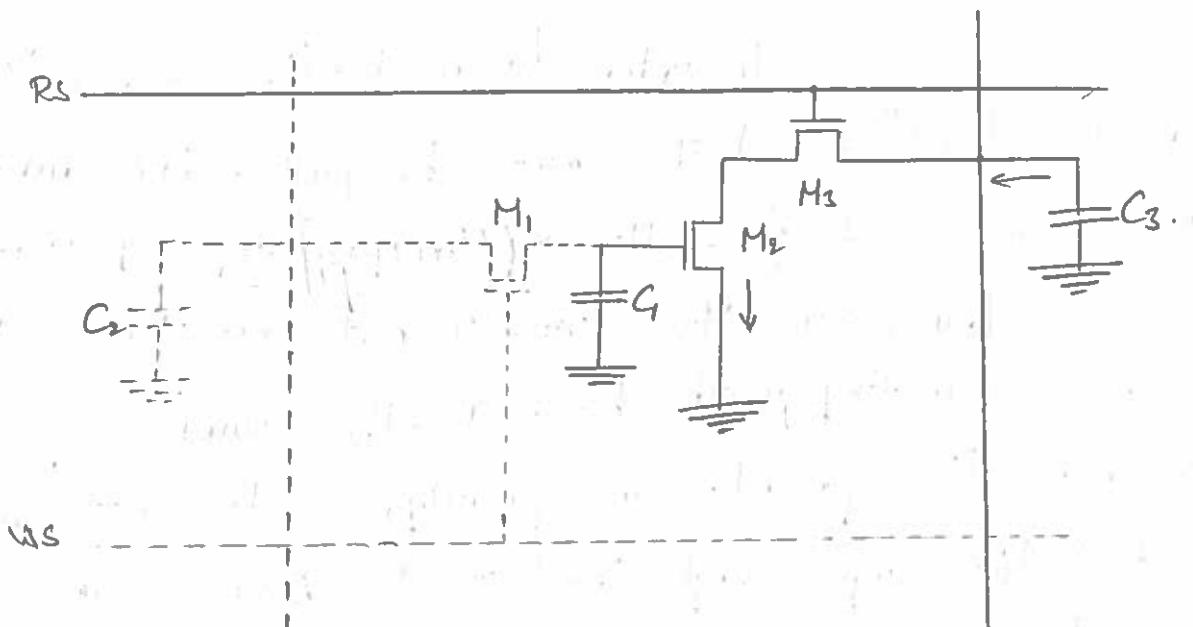


Fig 5.41: The column capacitance C_3 is discharged through the transistors M_2 and M_3 during the read "1" operation.

Write "0" operation, the inverse data input is at the logic-high level, because the data to be written onto the ORAM cell is a logic "0". Consequently the data write transistor is turned on, and the voltage level on column D_{in} is pulled to logic "0". Now, the "write select" signal WS is pulled high during the active phase of ϕ_2 . As a result, the write access transistor M_1 is turned on. The storage capacitance C_1 contains a very low charge, and the transistor M_2 is turned off.

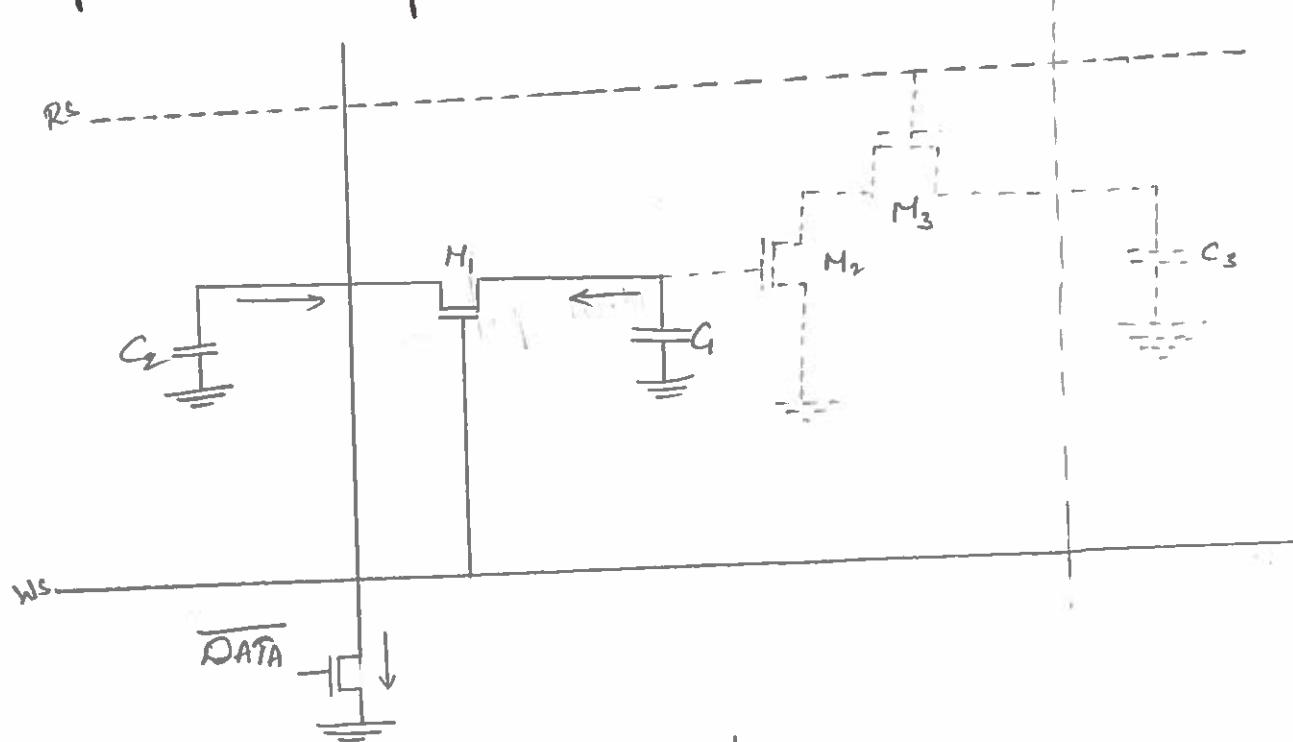


Fig 5.42: Both C_1 and C_2 are discharged via M_1 , and the data write transistor during the write "0" sequence.

In order to read this stored "0", the "read select" signal RS must be pulled high during the active phase of ϕ_2 , following a precharge cycle. The read access transistor M_3 turns on, but since M_2 is off, there is no conducting path between the column capacitance C_3 and the ground. Consequently, C_3 does not discharge, and the logic-high level on the Dout column is interpreted by the data read circuitry as a stored "0" bit.

Also, the use of periodic precharge cycles instead of static pull-up further reduces the dynamic power dissipation. The additional peripheral circuitry required for scheduling the non-overlapping control signals and the refresh cycles does not significantly overshadow these advantages of the low-power dynamic memory.

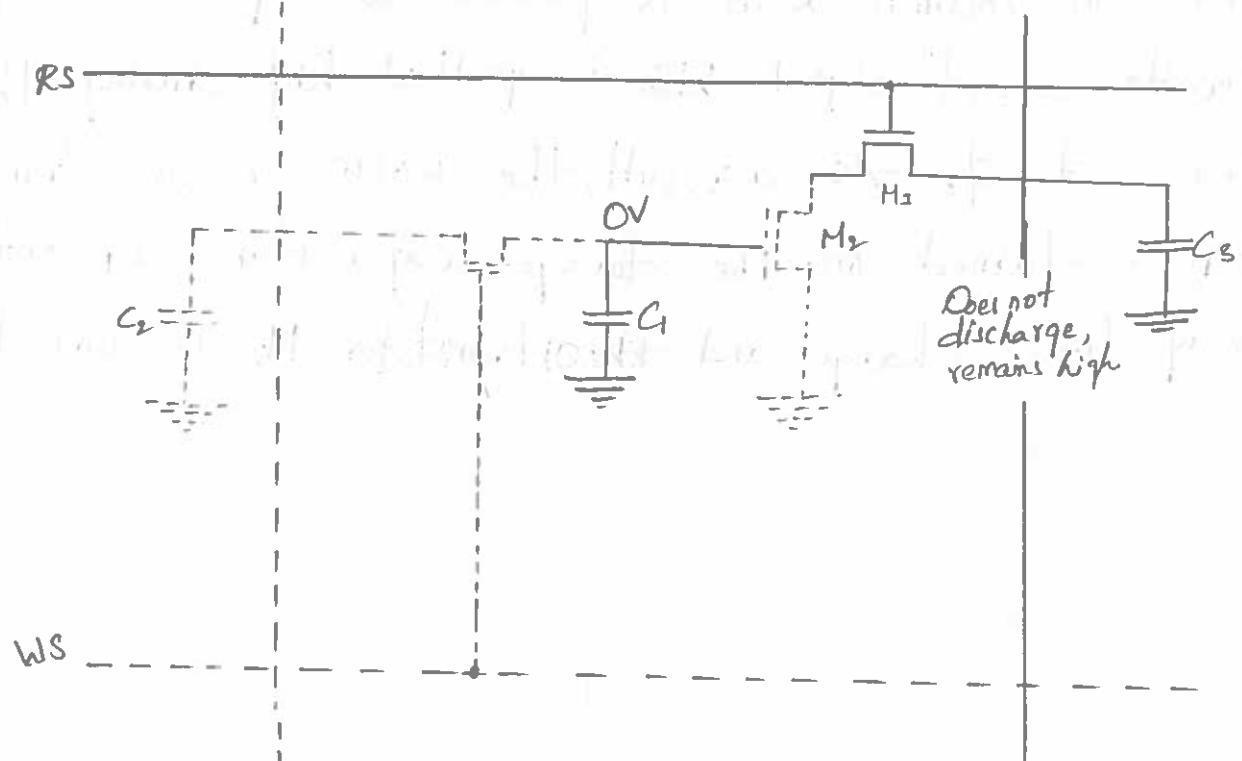


fig 5.43: The column capacitance C_3 cannot discharge during the read "0" cycle

One-Transistor DRAM Cell

The circuit diagram of the one-transistor (1-T) DRAM cell consisting of one explicit storage capacitor and one access transistor is shown in fig. Here, C_1 represents the storage capacitor which typically has a value of 30 to 100 fF. Charge sharing between this large capacitance and the very small storage capacitance plays a very important role in the operation of the 1-T DRAM cell.

Charge sharing between C_1 and C_2 occurs and, depending on the amount of stored charge on C_1 , the column voltage either increases or decreases slightly. Note that charge sharing inevitably destroys the stored charge on C_1 . Hence, we also have to refresh data every time we perform a "data read" operation.

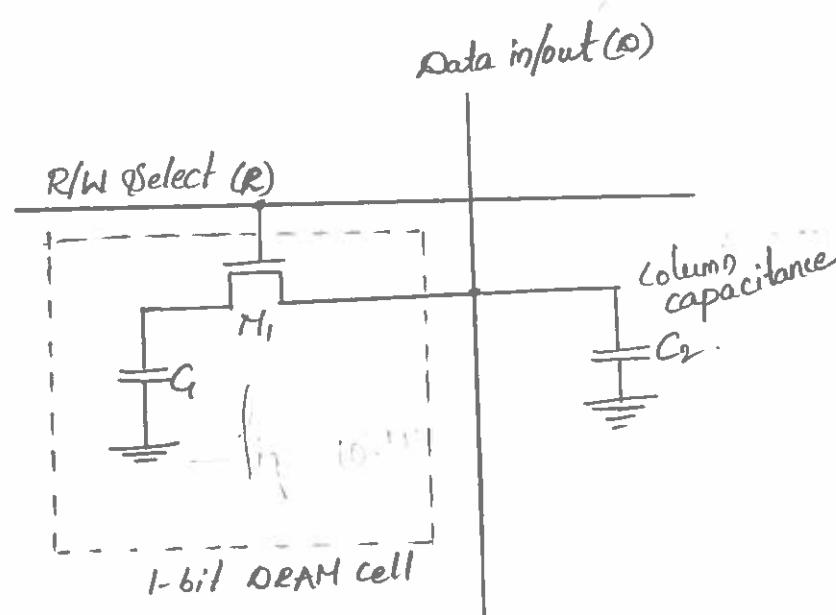


fig 5.44: Typical one-transistor (1-T) DRAM cell with its access lines.

The "read-refresh" operation occurs in three stages. First, the precharge devices are turned on during the active phase of PC. Both column capacitances C_D and C_{D2} are charged up to the same logic-high level, where the dummy nodes X and Y are pulled to logic-low level. The devices involved in the precharge operation are highlighted in fig. Note that during this phase, all other signals are in active.

fig: 10.45 (a)

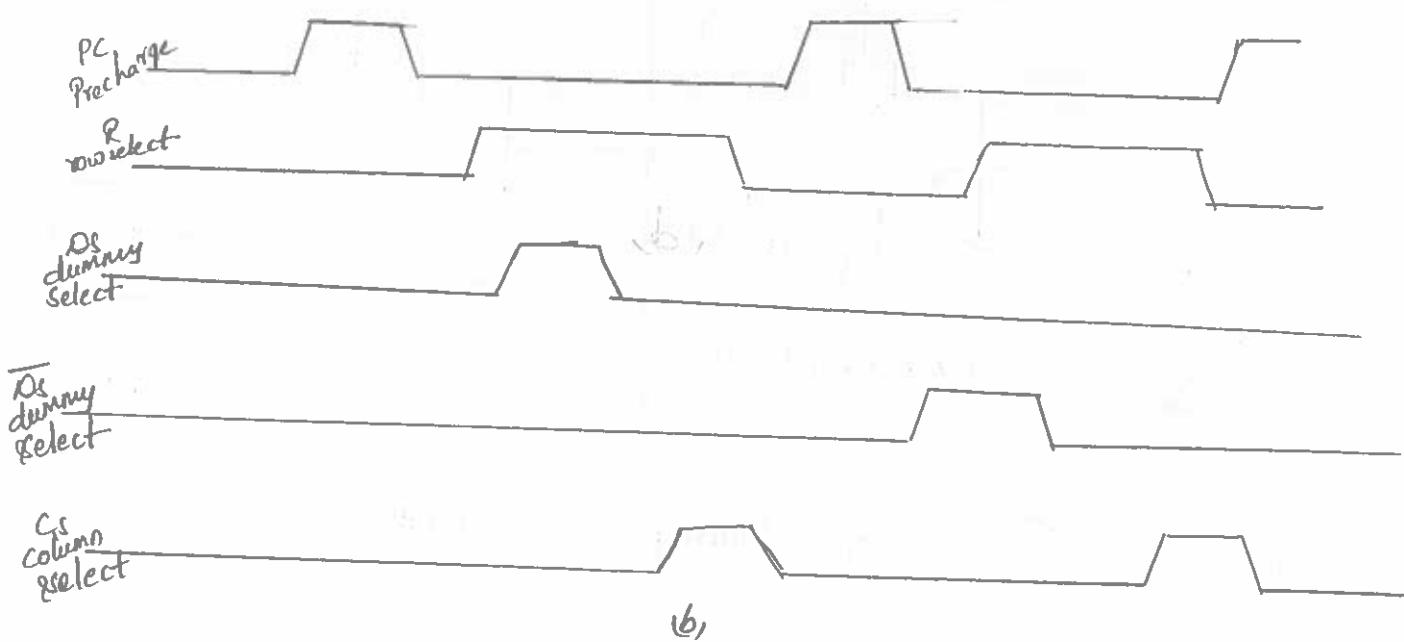
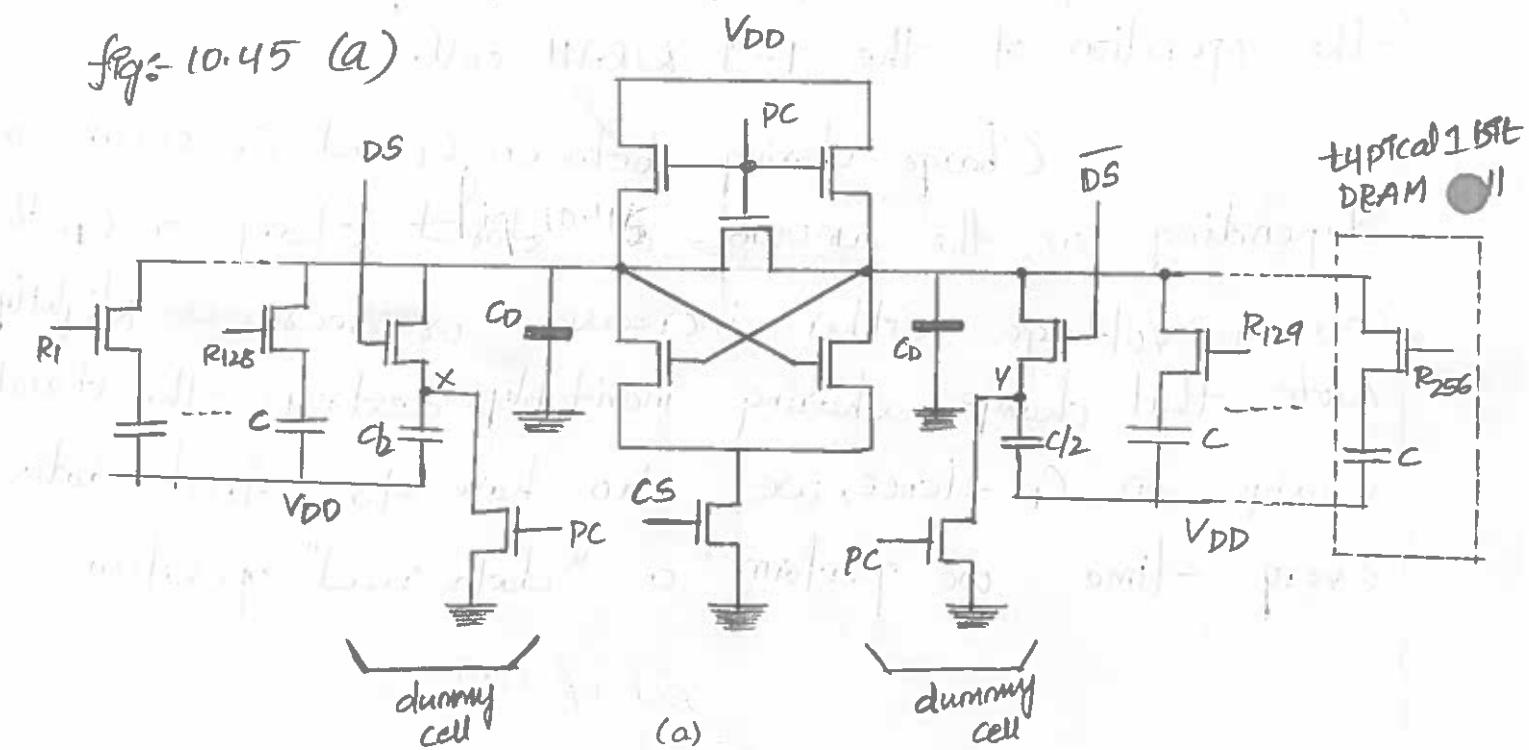


fig 5.45 (a) Data read-refresh circuit ex for 256 1-BIT DRAM cells per column
b) Typical control signal waveforms for two consecutive data read operations, performed on alternate sides (half-columns) of the array.

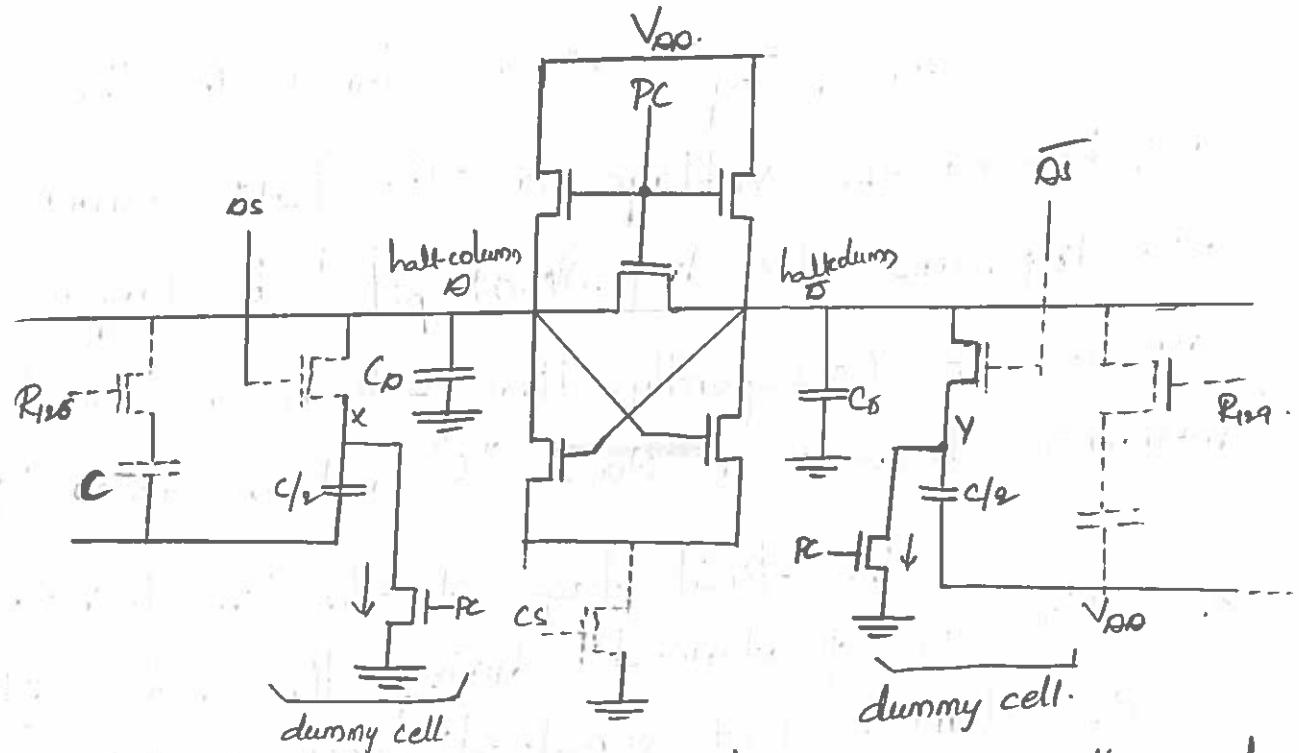


fig 5.46: The half-columns are being charged-up during the precharge phase

Next, one of the 256 word lines is raised to logic "1" during the row selection phase. At the same time, the dummy cell on the other side is also selected by raising either D_s or \bar{D}_s . This situation is depicted in fig, where only the selected DRAM cell (left) and the corresponding dummy cell (right) are highlighted.

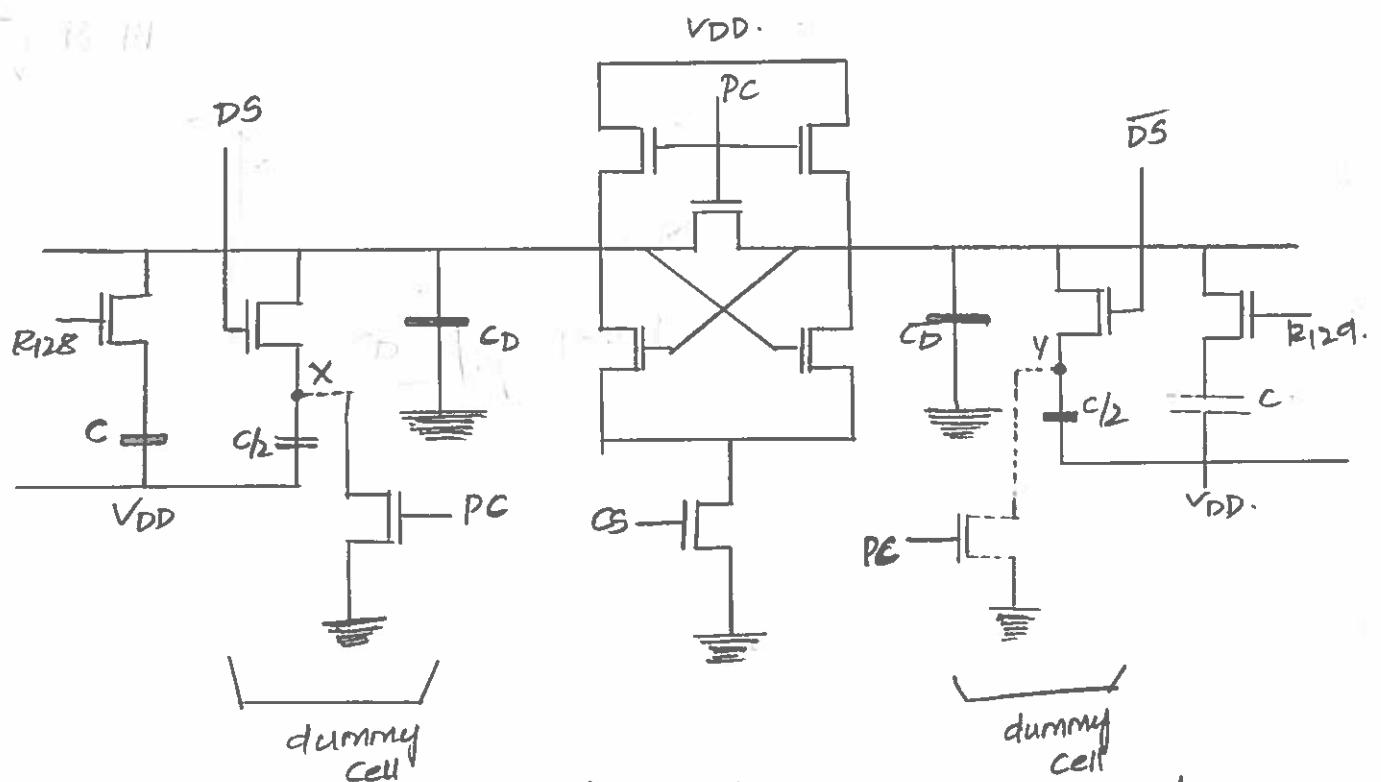


fig 5.47: Two complementary column voltages are determined through charge sharing.

If a logic '0' is stored in the selected cell, however, the voltage on the half-column D will also drop, and the drop in V_D will be larger than the drop in $V_{\bar{D}}$. Consequently, there will be a detectable difference between a stored "0" and a stored "1".

The final stage of the "read-refresh" operation is performed during the active phase of CS, the column-select signal. As soon as the cross-coupled latch is activated, the slight voltage difference between the two half-columns is amplified, and the latch forces the two half-columns into opposite states. Thus, the stored data on the selected DRAM cell is refreshed while it is being read by the "read-refresh" circuitry.

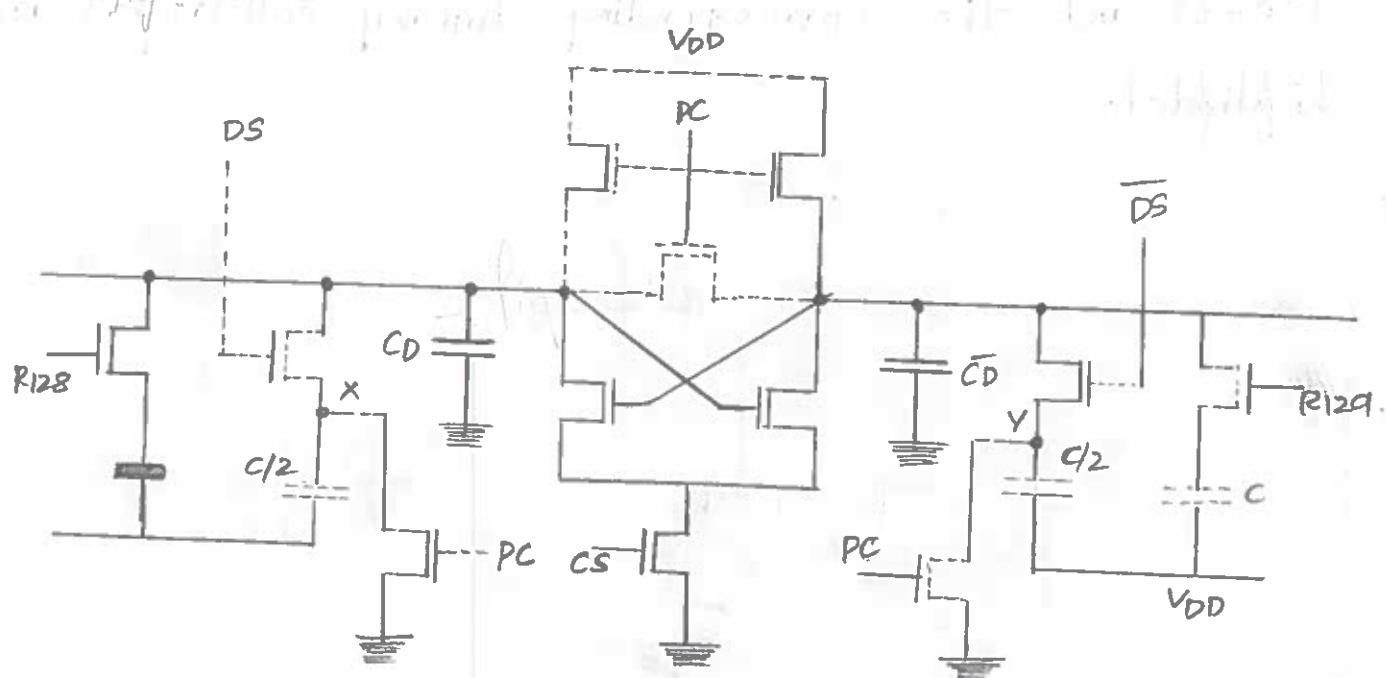


Fig 5.48: The cross-coupled latch circuit is used for detection of the voltage difference between the half-columns and for restoring the voltage level on the accessed cell.