ARRAY SUBSYSTEMS

Memory elements may be divided into :

(i) Random access memory
(ii) Serial access memory
(iii) Content access memory.

RAM at chip level is classed as memory that has access time independent of the physical location of the data. Where as serial-access memories have some latency associated with reading or writing of particular data and with content addressable memories.

ROMs have write time greater than read times. But RAMs have very similar read and write times. These are divided into

* Static-load : reqs no clock.
* Synchronous : require clk' edge to enable memory operation.
* Asynchronous : asyn. RAMs recognize address changes & o/p new data after any such change.

→ Static-load & synchronous memories are easier to design & best choice for system-level building block.

→ Memory cells in RAMs further divided into :
(a) Static structures (b) Dynamic structures.

Static cells use some form of latched storage, while dynamic cells use dynamic storage of charge on a capacitor.

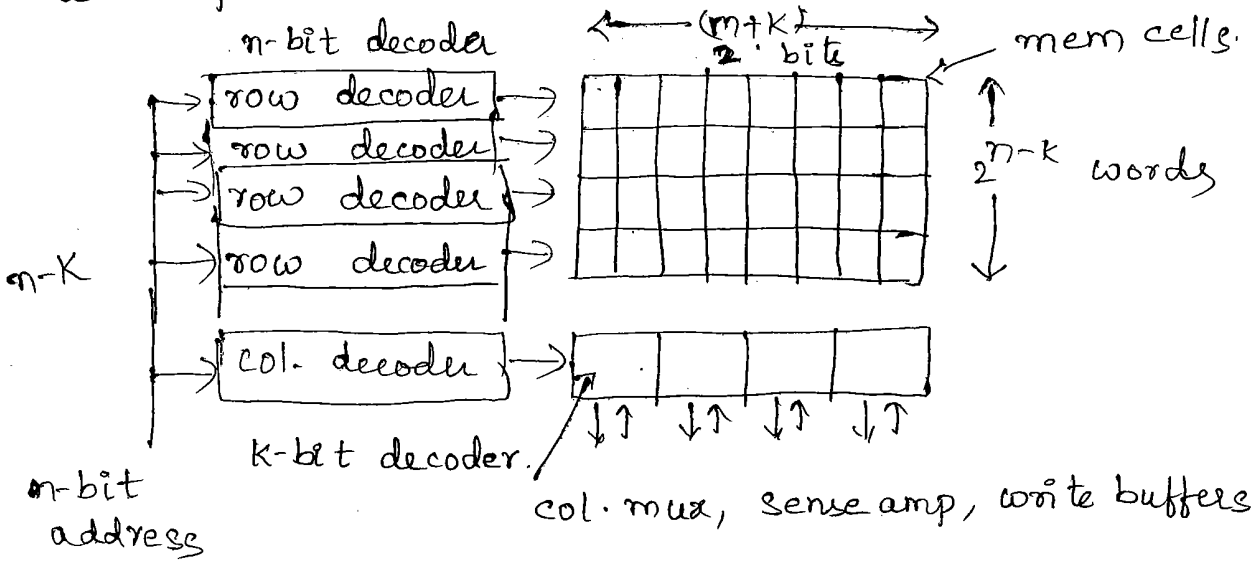


fig: Memory-chip architecture.

RAM :-

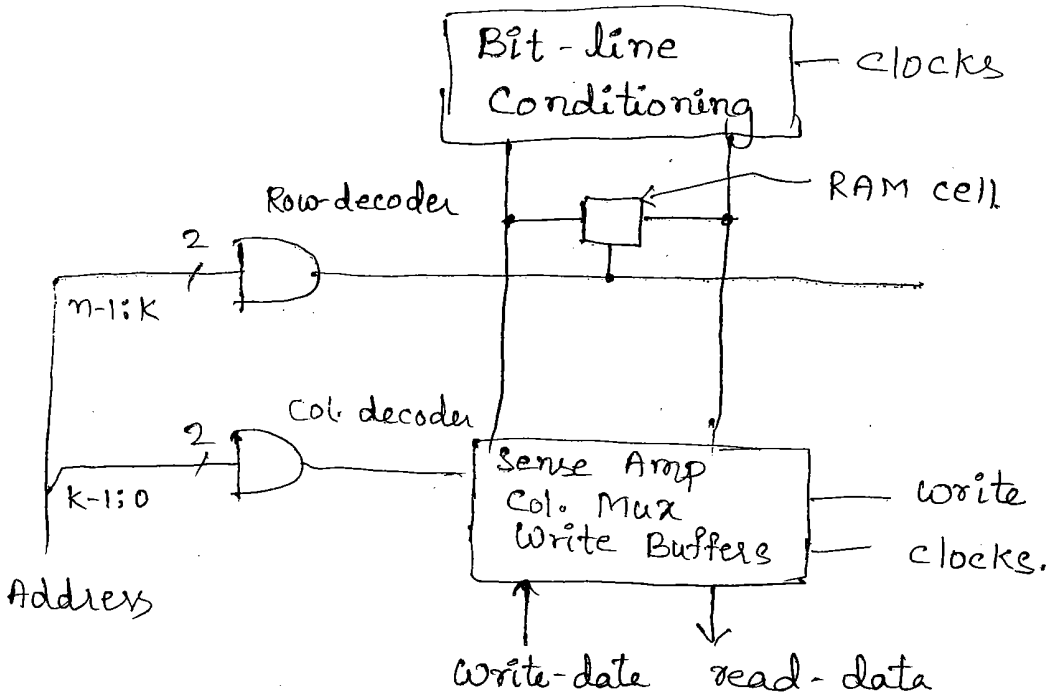
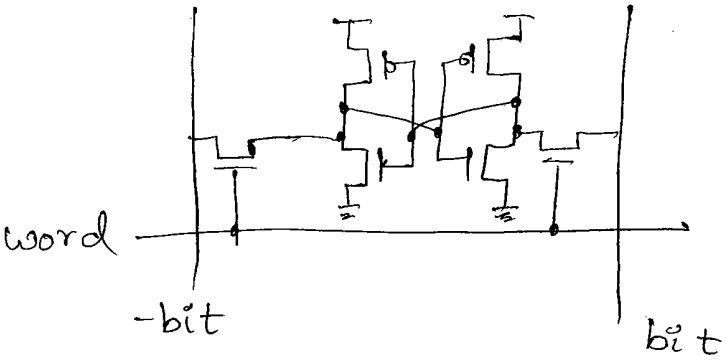


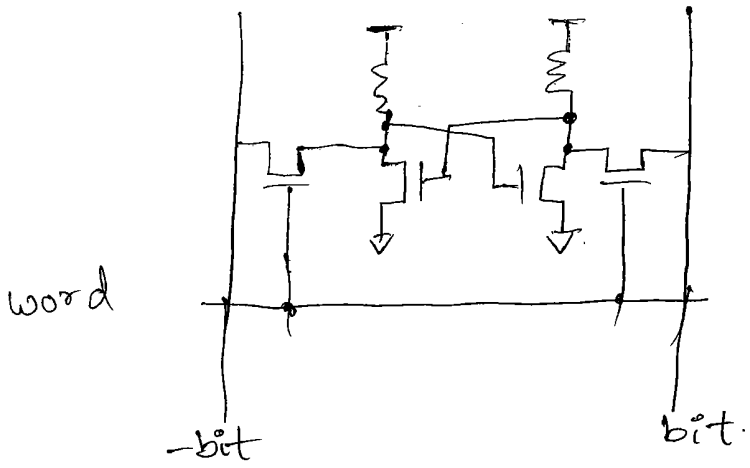
fig: Generic RAM circuit.

cross- coupled inverter ckt.

The p- Trs may be replaced with high value polysilicon resistors. The value of resistor has to be ∋ it prevents leakage from changing any value stored in RAM cell. Generally 100's to 1000's of MΩ.



(a) SRAM



(b) SRAM

HIGH DENSITY MEMORY ELEMENTS

→ On-chip memory is important as levels of integration increase to allow both processors and useful amts of mem to be integrated on a single chip.

→ ROM

→ Flash memory : dominant form of electrically erasable PROM memory.

→ RAM — SRAM
DRAM

→ SRAM is faster, Larger & uses more pow.

→ DRAM has smaller layout & uses less pow. DRAM cells require periodically refreshing of dynamically stored values.

→ A design that reqs high-density ROM or RAM is partitioned into several chips.

→ Medium density memory, on the order of one 'K' bytes, often be put on same chip with logic that uses it, giving faster access times, as well as greater integration.

→ The bit lines are typically precharged, so the cell discharges one of the lines.

→ The bit lines are read by ckts that sense the value on the line, amplify to speed it up & restore the signals to the proper vtg levels.

→ A write is performed by setting the bit lines to the desired values & driving that value from the bit lines into the cell.

→ Row decoders typically use NOR gates to decode the addrs, followed by chain of buffers to allow ckt to drive large capacitance of the word line.

→ ckts to implement NOR :
  → Pseudo-nMOS
  → Precharged.

→ Precharged ckts provide better performance for large memory arrays compared to Pseudo-nMOS.

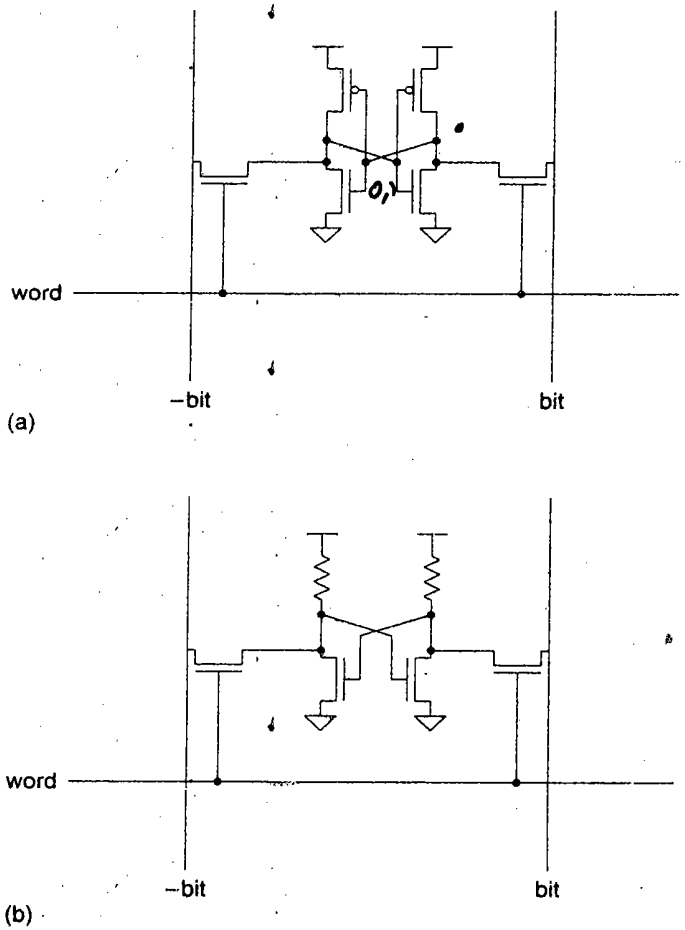


Figure 8.50 Static RAM cell circuits

merged read and write data busses may be used. A 1-transistor cell is shown in Fig. 8.52(c).[20] The memory value is again stored on a capacitor. The capacitor can be implemented as a transistor as shown in Figs. 8.52(d) and 8.52(e). Sense amplifiers sense the small change in voltage that results when

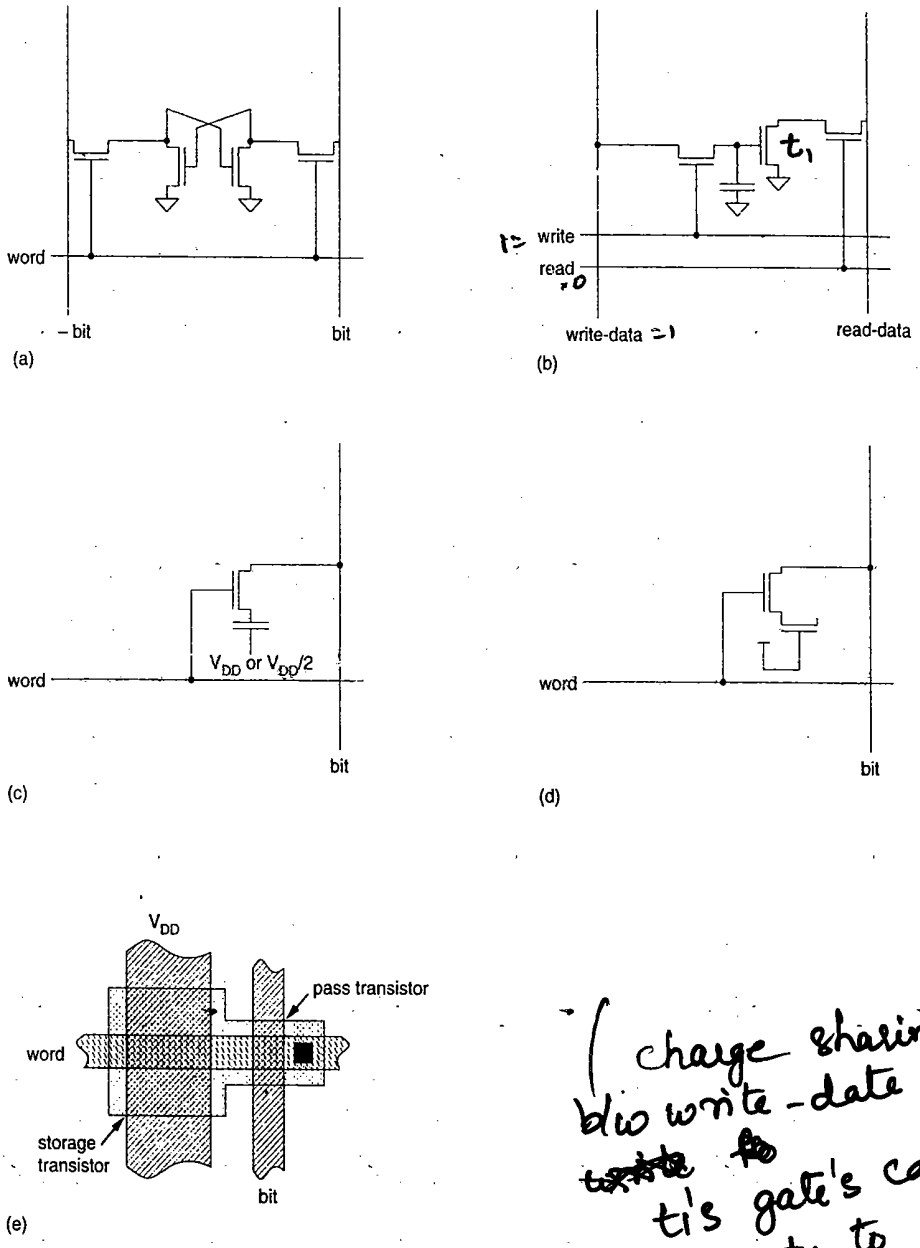
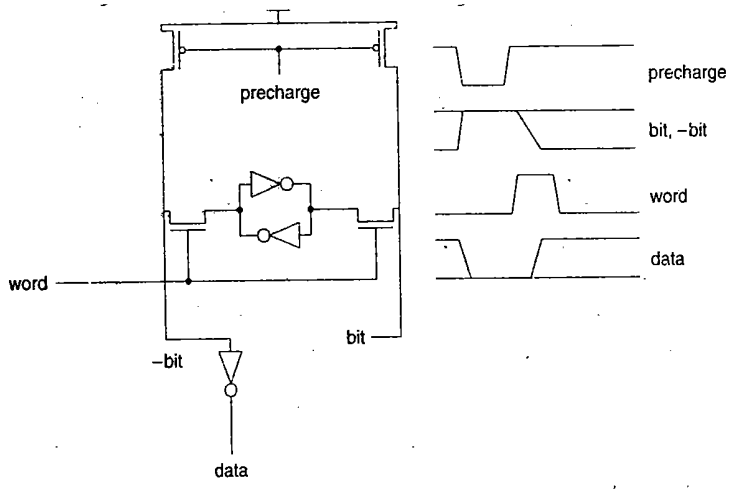
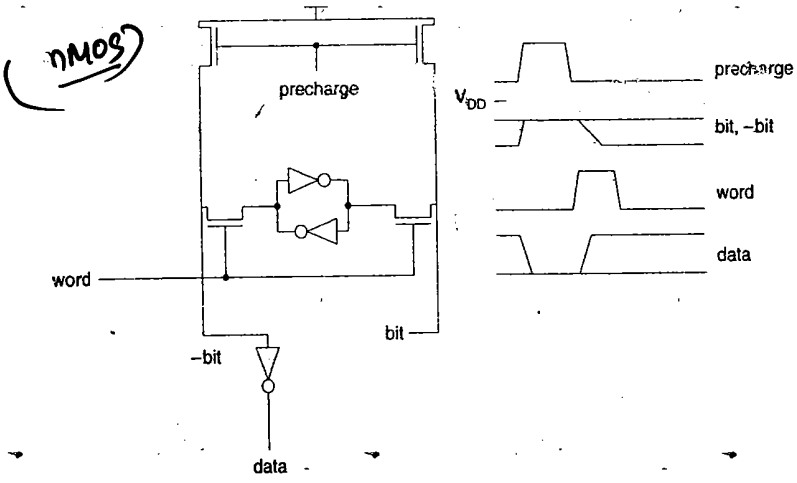


Figure 8.52 Dynamic RAM circuits: (a) 4-transistor; (b) 3 transistor; (c) 1 transistor with capacitor; (d) 1 transistor with transistor capacitor; (e) representative layout for (d)

the RAM circuit concentrates on pulling the bit line from high to low. Thus one method of reading a RAM cell would be to precharge the bit lines high and then enable the word-line decoder. For a given pair of bit lines, one RAM cell will attempt to pull down either the *bit* or *–bit* line depending on the stored data. The bit-line pull-up circuit may use p-channel transistors to precharge each bit line (Fig. 8.53a). In this example, the sense amplifier is an inverter that forms a single-ended sense amplifier. The sense time is roughly



(a)



(b)

FIGURE 8.53 RAM read options: (a) V_{DD} precharge; (b) $V_{DD}-V_{tn}$ precharge

the time it takes one RAM cell pull-down and access transistor to reach the inverter threshold. To optimize speed, one might set the inverter threshold above the V_{DD} midpoint, but below an adequate noise margin down from the V_{DD} rail. Alternatively, one can precharge the bit lines with n-channel transistors, which results in the bit lines being precharged to an n threshold down from V_{DD} (Fig. 8.53b). This can dramatically improve the speed of the RAM cell access. In addition, it reduces power dissipation because the bit lines do not change by the supply voltage. The key aspect of the precharged RAM read cycle is the timing relationship between the RAM addresses, the precharge pulse, and the enabling of the row decoder. If the word-line assertion precedes the end of the precharge cycle, the RAM cells on the active word-line will see both bit lines pulled high and the RAM cells may flip state. If

to amplify this bit-line change. Design margins must be valid over all process, temperature, and voltage extremes. Figure 8.55 shows the zero bit voltage ($V_{bit(0)}$) and the pull-down voltage ($V_{pulldown}$) for various ratios of pull-up beta to pull-down betas. As the pull-up becomes weaker, the $V_{bit(0)}$ voltage approaches V_{SS} and the differential voltage between a high and a low on the bit lines increases. However, as the pull-down transistors are limited in size by the desire to keep the RAM cell small, a design trade-off has to be made between speed and the differential bit voltage, which affects the noise

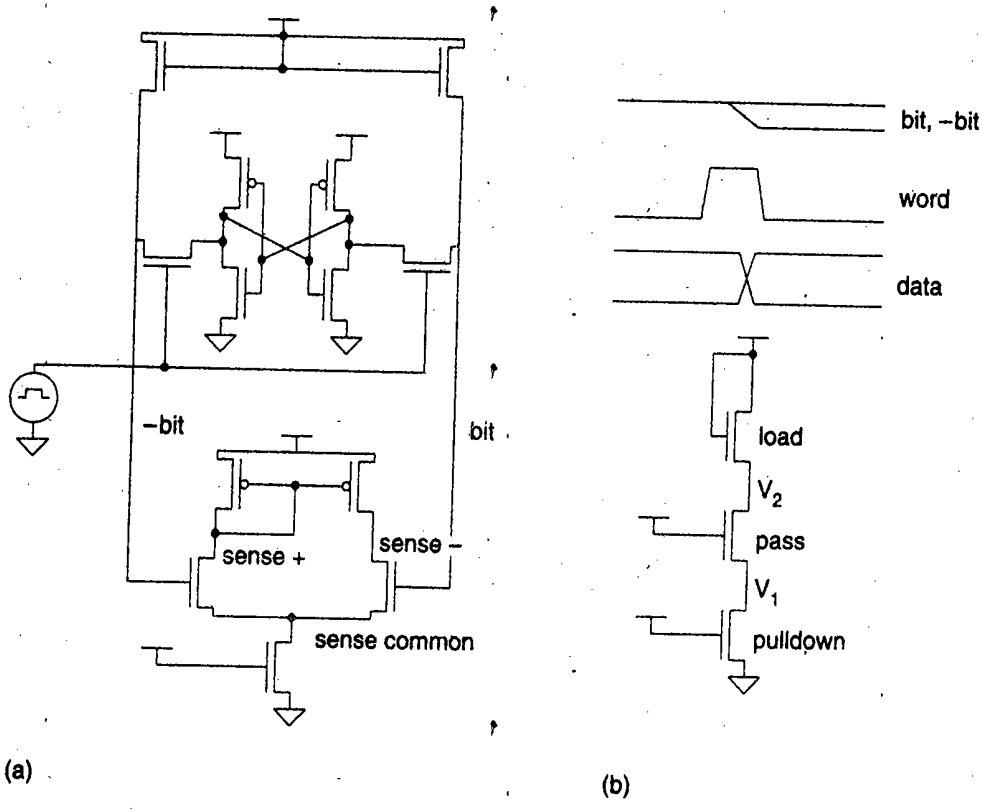


Figure 8.54 RAM read operation model

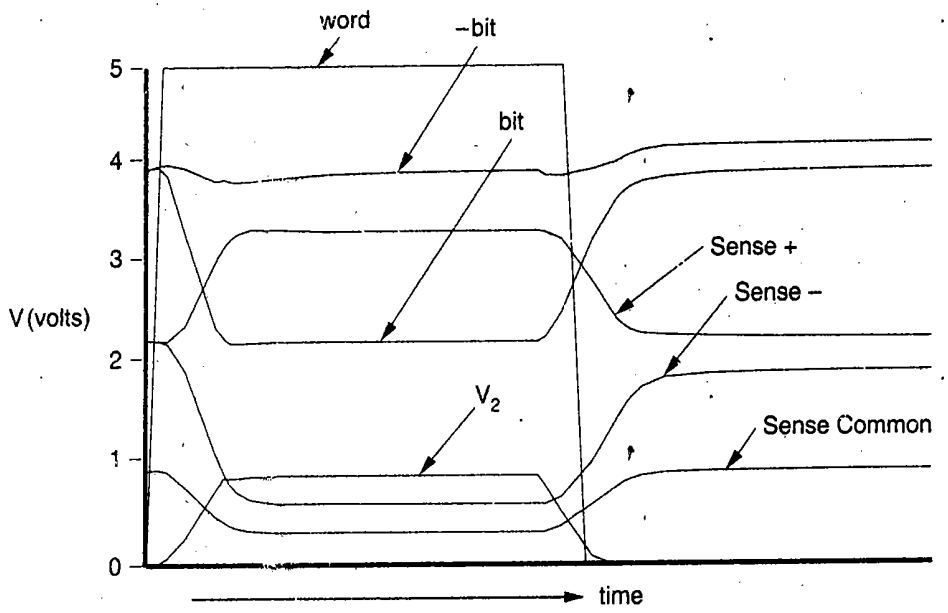
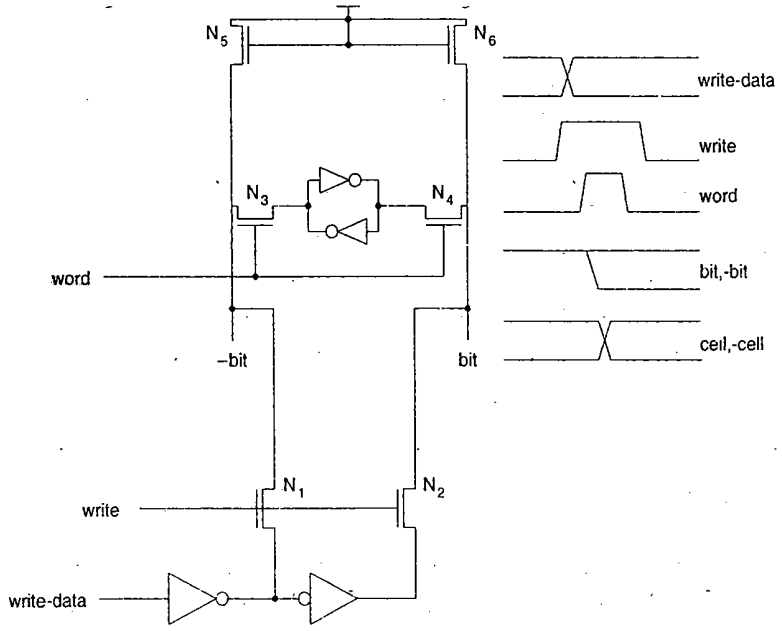


Figure 8.56 Static RAM–read waveforms

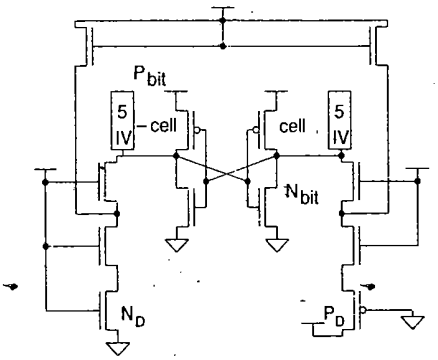
Current mode sensing may also be used.[29,30,31] In this technique, the current change in the bit lines is detected using special circuits. The theory is that by using low-impedance circuits, the RC delay inherent in driving the bit lines may be decreased.

8.3.1.1.2 Static RAM—write

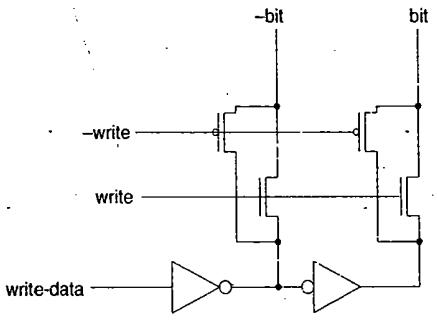
The objective of the RAM write operation is to apply voltages to the RAM



(a)



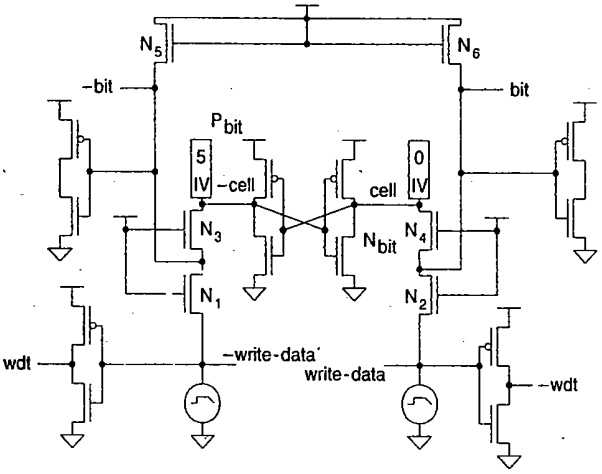
(b)



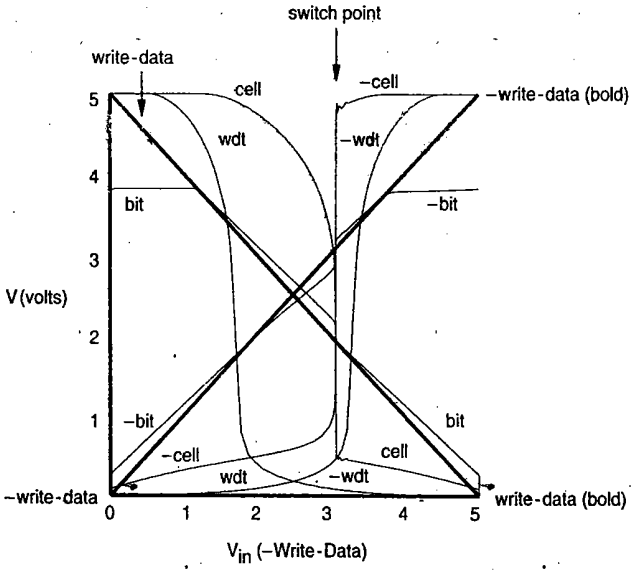
(c)

FIGURE 8.57 Static RAM–write circuits: (a) n-channel pass transistors; (b) circuit model during write; (c) complementary transmission gate version

write enable trs N_1, N_2 are enabled to allow data & comp to move to bit lines.



(a)



(b)

Figure 8.58 Static RAM–write waveforms and circuit model

8.3.1.1.3 Row decoders

The simplest row decoder is an AND gate. Figure 8.59 shows two straightforward implementations. The first in Fig. 8.59(a) is a static complementary NAND gate followed by an inverter. This structure is useful for up to 5–6 inputs or more if speed is not critical. The NAND transistors are usually made minimum size to reduce the load on the buffered address lines because there are 2^{n-k} $(N_{load} + P_{load})$'s on each address line. The second implementation, shown in Fig. 8.59(b), uses a pseudo-nMOS NOR gate buffered with two inverters. The NOR gate transistors can be made minimum size, and the inverters can be scaled appropriately to drive the word line. Large fan-in AND gates can also be constructed from smaller NAND and NOR gates, as shown in Fig. 8.59(c). Figure 8.60 shows two possible layout styles (in sym-

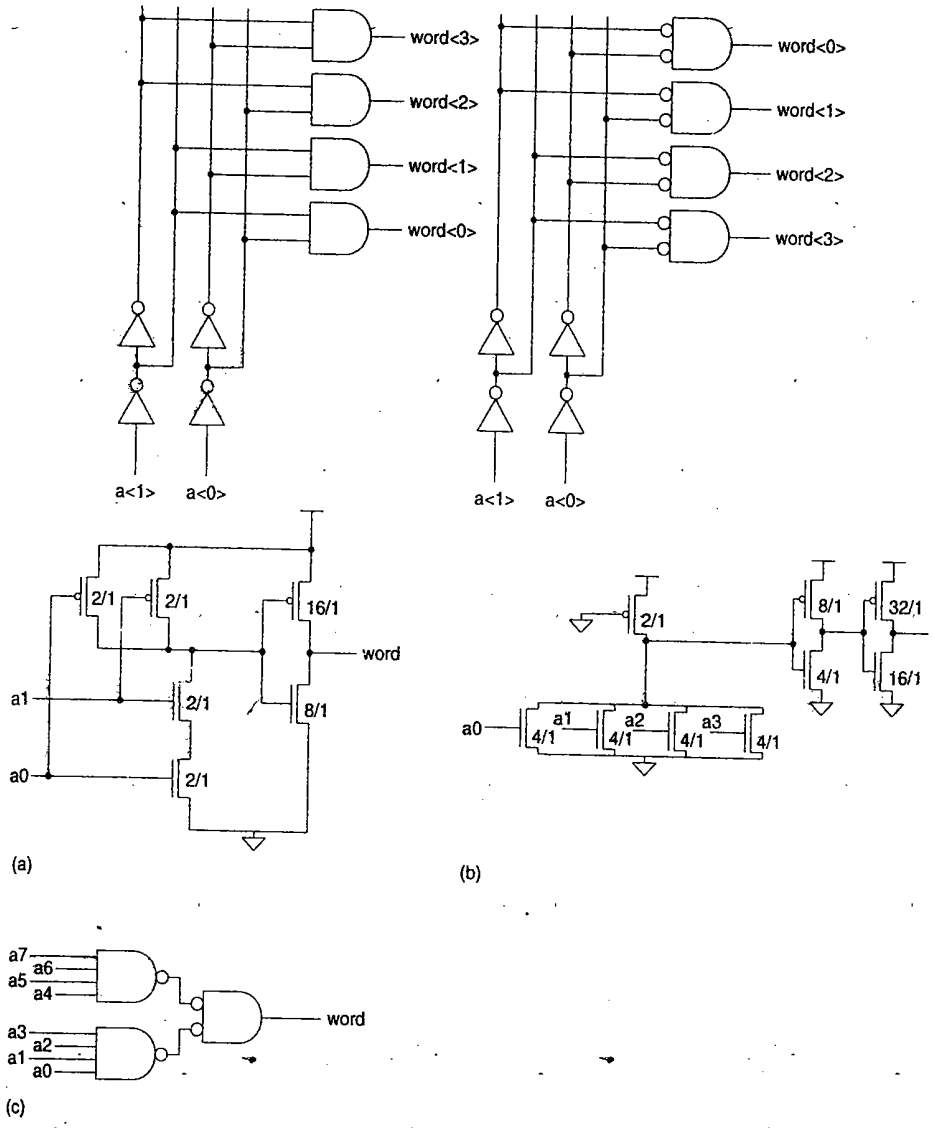


FIGURE 8.59 Row-decoder circuits: (a) complementary AND gate; (b) pseudo-nMOS gate; (c) cascaded NAND, NOR gates

bolic form) for the row decoders. One passes the address lines over the decode gates, while the other uses a more standard cell style. Choice would depend on the size of the decoder in relation to the size of the RAM cell. Often, speed requirements or size restrict the use of single-level decoding, such as that shown in Fig. 8.59. The alternative is a predecoding scheme, which is illustrated in Fig. 8.61(a). Here the $(n\text{-}k)$ row address lines are split into a p-bit predecode field and a q-bit direct decode field. The q-bit decode field requires a gate per word line, so q is chosen to suit the pitch of the RAM cell. The p-bit predecode field generates 2^p predecode lines (4 in this example), each of which is fed vertically to 2^{n-k}-row decode gates (8 in this example). Figure 8.61(b) shows a possible implementation of a predecode scheme, where the predecode gate is a NAND gate and the word-decode gate is a NOR gate. An additional input ($-clk$) has been included in the NOR gate

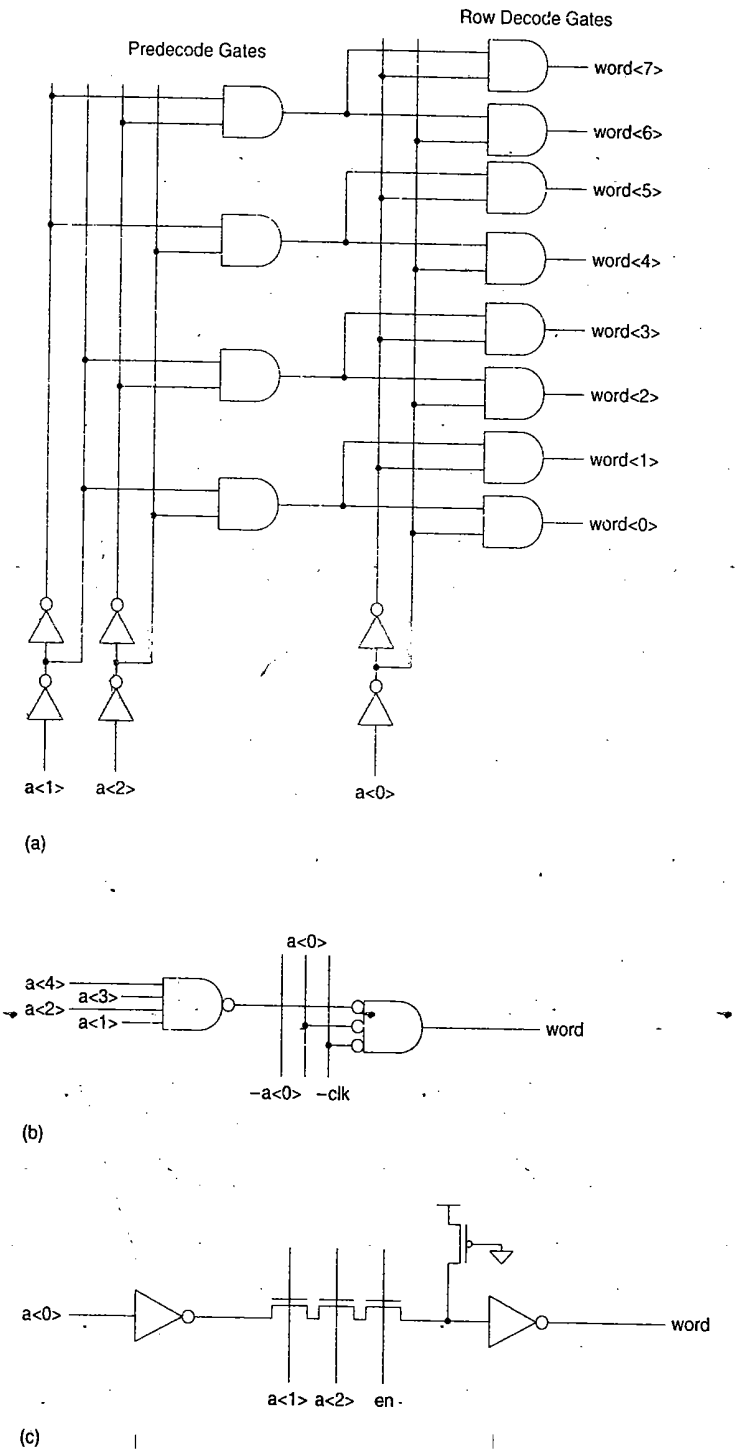


FIGURE 8.61 Predecode circuits: (a) basic approach; (b) actual implementation; (c) pseudo-nMOS example

via pass gates enabled by the column-address lines. The address decoding is in essence distributed. Decoders for *bit* and *–bit* lines are shown, although one of these may be omitted for single-ended read operations. The read (and, usually of lesser importance, write) operations are somewhat delayed by the series-transmission gates. However, in comparison with gate delays these

COL DECODER.

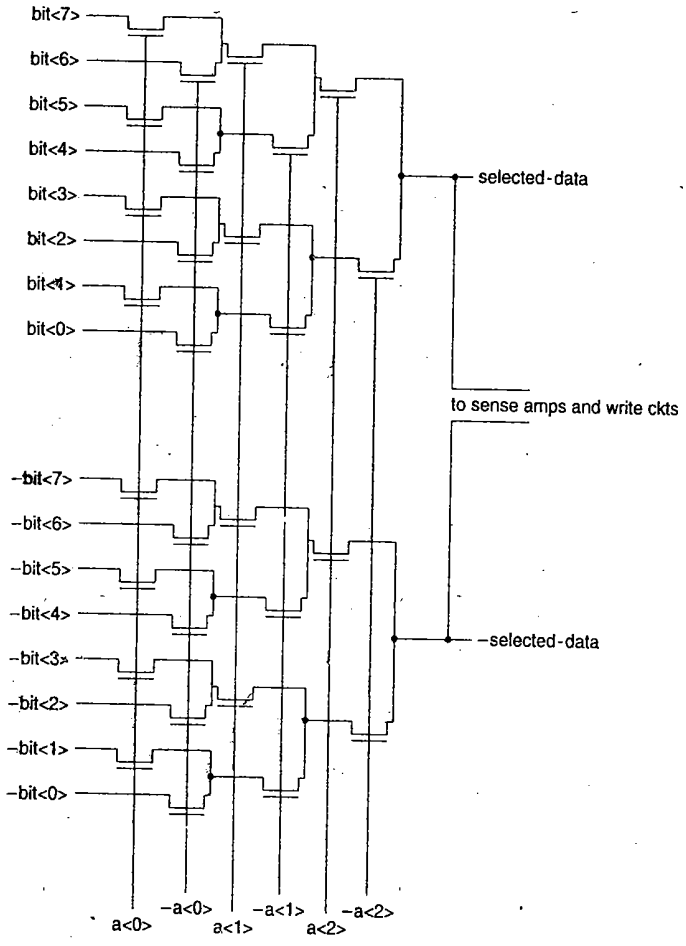


FIGURE 8.63 Tree-style column decoder

If the delay of the series-pass gates was troublesome, the decoder shown in Fig. 8.64 could be used. Here a NAND decoder is employed on a bit-by-bit basis to enable complementary transmission gates (single transistors may be used where possible) onto a common pair of data lines. These are then routed to a sense amplifier and write circuitry.

8.3.1.1.5 Sense amplifiers

Many sense amplifiers have been invented to provide faster sensing, smaller layouts, and lower power-dissipation sensing.[33] The simple inverter sense amplifier provides for low power sensing at the expense of speed. The differential sense amplifier can consume a significant amount of DC power (Fig. 8.54). Alternatively, one can employ clocked sense amplifiers similar to the SSDL gate shown in Fig. 5.40.

8.3.1.1.6 RAM timing budget

The critical path in a static RAM read cycle includes the clock to address delay time, the row address driver time, row decode time, bit-line sense time, and the setup time to any data register. The column decode is usually not in

Selecting 2^k out of 2^m bits of accessed row.

CONTENT ADDRESSABLE MEMORY

The CAM portion examines a data word and compares this data with internally stored data. If any data word internally matches the I/p data word, the CAM signals that there is a match. These match signals can be passed as word lines to RAM to enable a specific data word to be o/p. This structure may be used as translation look-aside buffer in the virtual memory look up in a microprocessor.

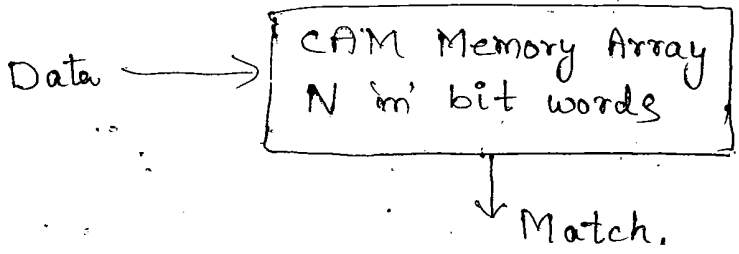


fig: Basic CAM

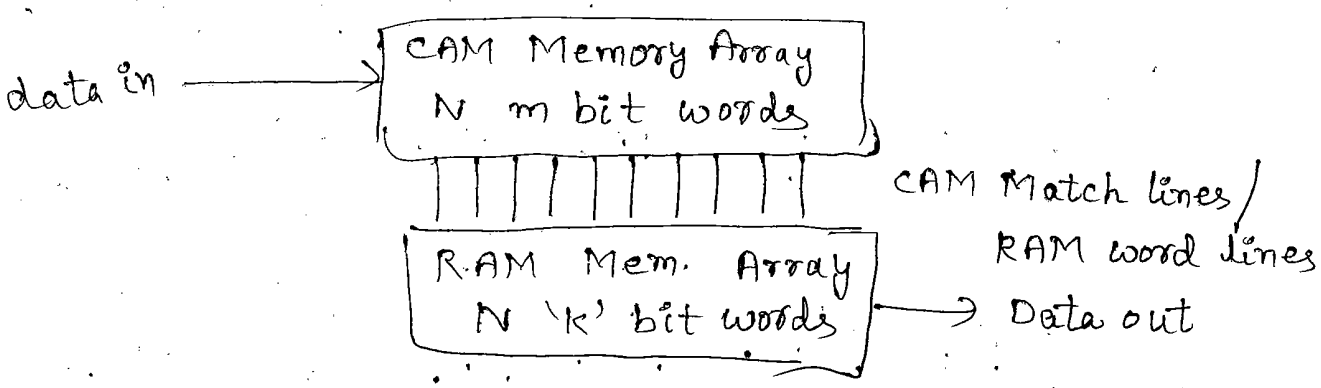


fig: applicn as translation lookaside buffer.

A CMOS CAM cell consists of normal static RAM cell with transistors N_1 and N_2, which form XOR gate, & N_3 which is distributed NOR pull-down. Writes are used to store the match data in the cells, whereas reads are used for testing

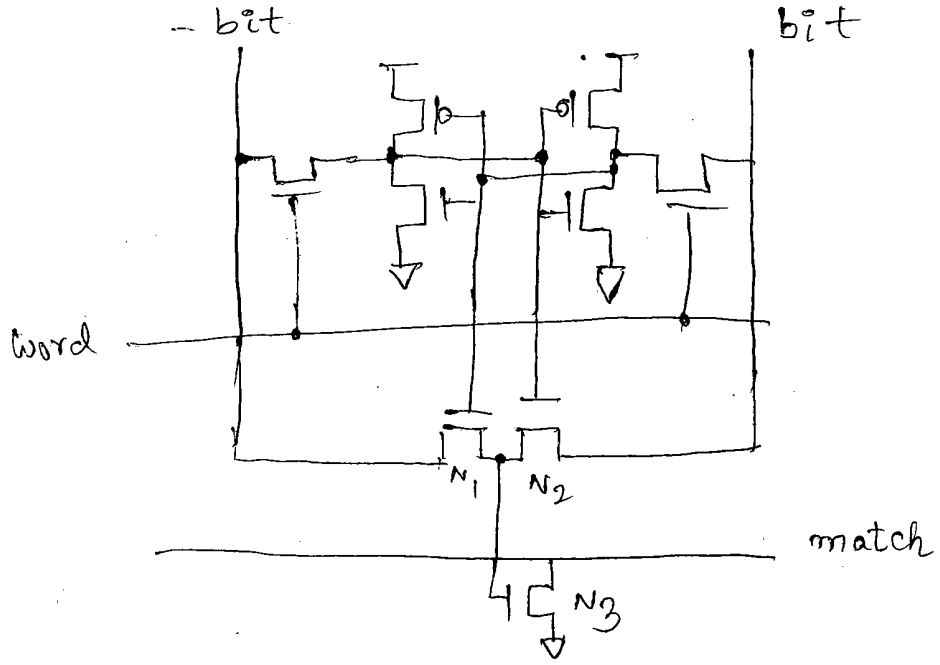


fig: CMOS CAM cell

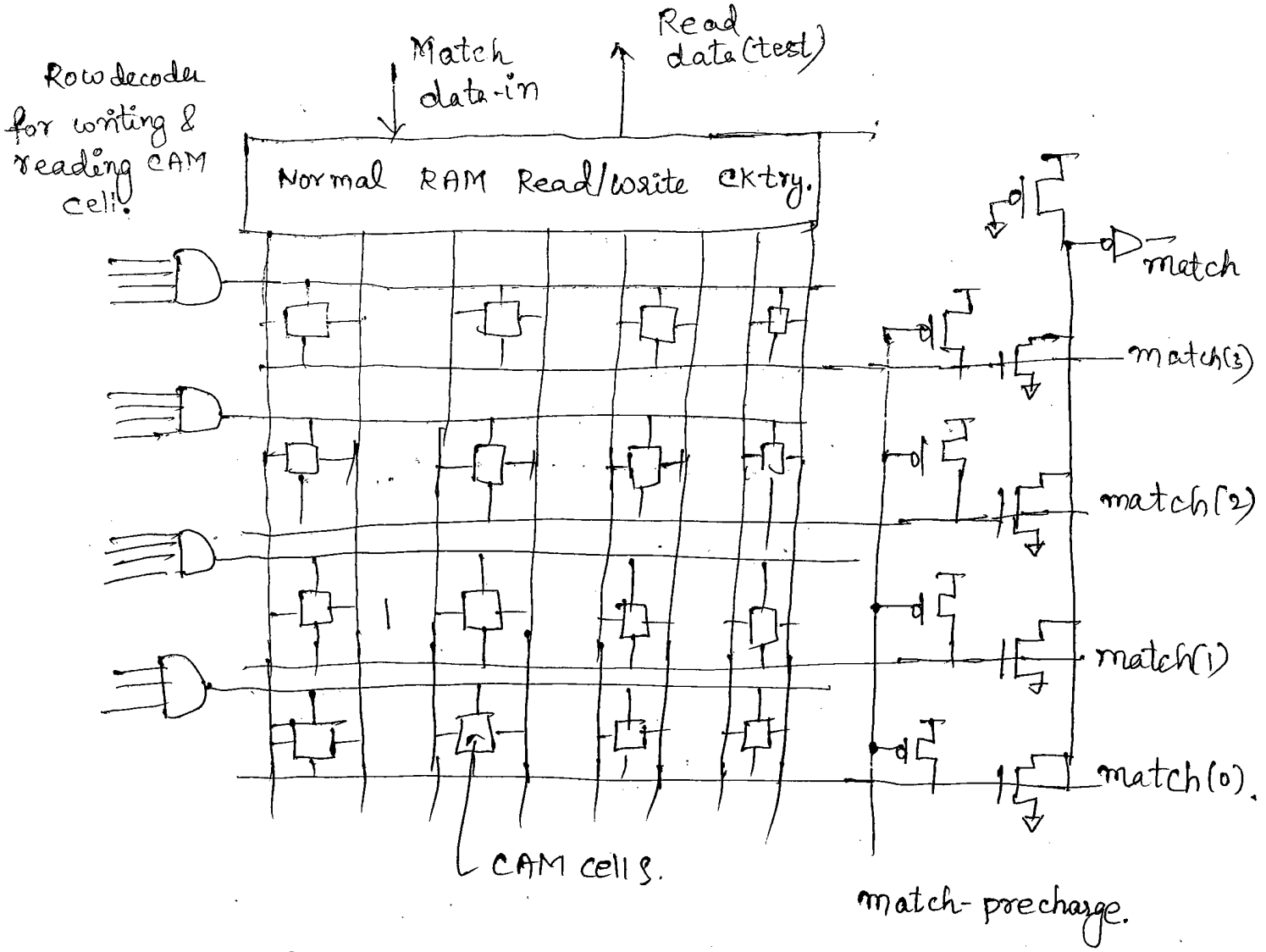


fig: array circuit.

A match operation [illegible] to be matched on the bit lines but not asserting word line. A '1' appears on gate of N3 if the data in the cell is not equal to data on bit lines. The drain of N3 Trs of cells in the same row are commoned. These form distributed NOR gate. Each match line (match<3:0>) remains high if the data in the row matches the data placed on the bit lines. These lines may be used to assert the word lines on a RAM.

SERIAL ACCESS MEMORY

Serial access memories (shift regs) are of use in signal-processing applications for storage and delaying signals. It can be simulated by a RAM & this provides smallest implementation bcoz CMOS static RAM is very area efficient. SAM is surrounded by counters. Some times a dedicated shift register memory may be appropriate from density, speed or floorplanning viewpt.

fig => Blocks of byte shift regs are delayed by 32, 16, 8, 4, 2 & 1 clk cycles & muxes control the pass-around of delay blocks to yield appropriate delay amount. Each mem cell is a [illegible]

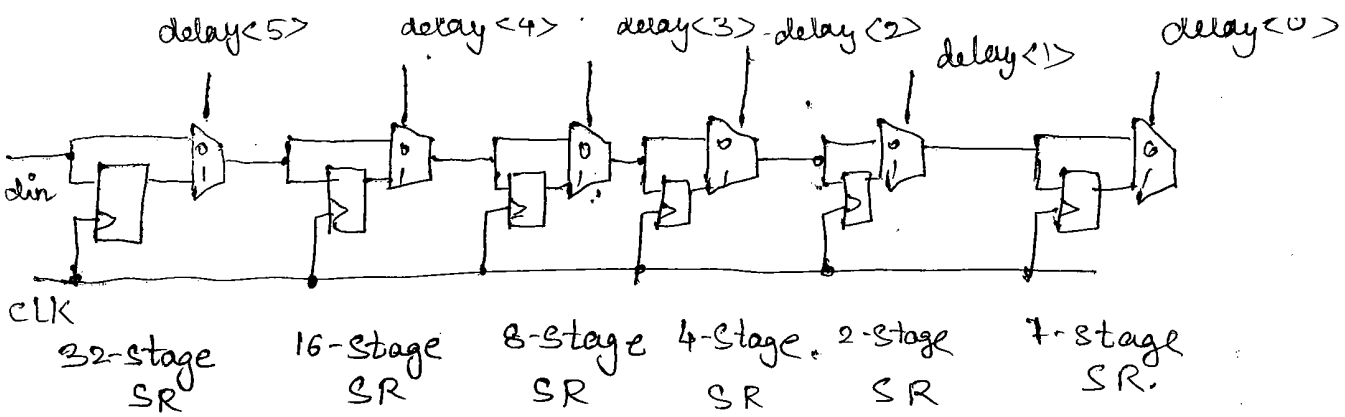


fig: Tapped delay line architecture.

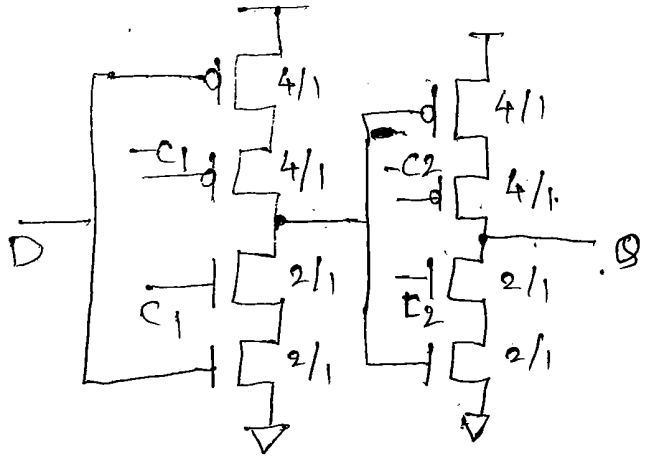


fig: ckt (mem cell).

ROM (READ ONLY MEMORY)

→ It can be implemented with only one Tr per bit of storage. It is a static memory structure in that the state is retained indefinitely without even power. It is generally implemented as a NOR array.

→ Can use NAND array for ultra small ROMs but will be slow.

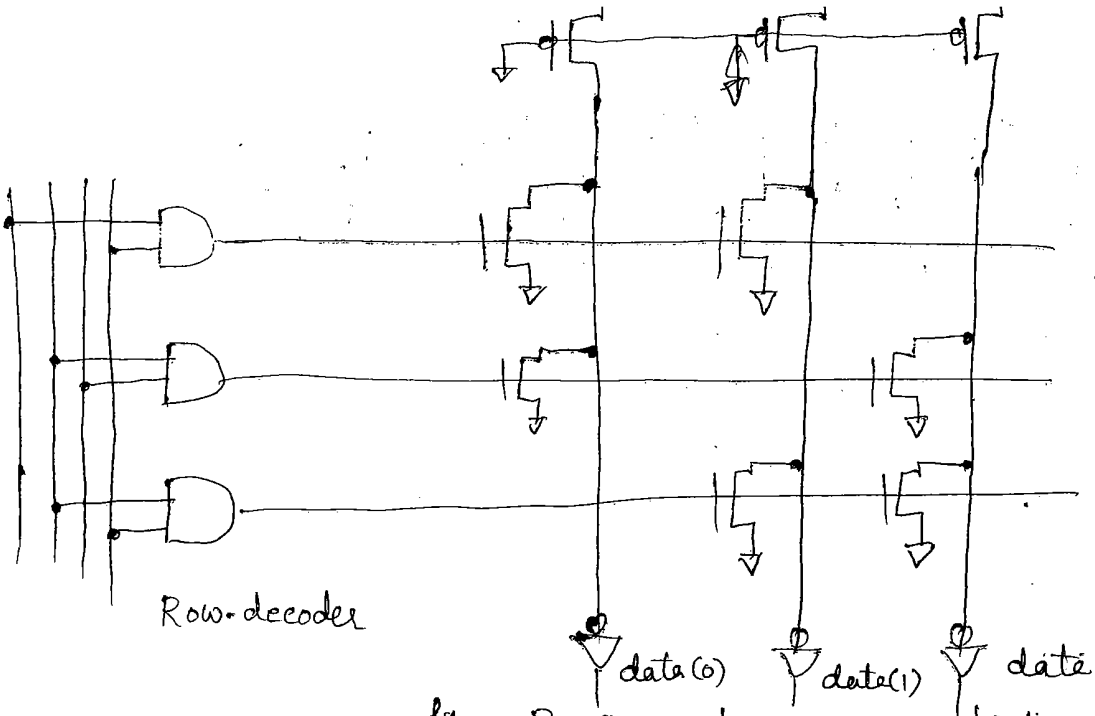


fig: Basic ROM architecture.

Domino logic ROM can slow down bit-line transition for large ROMs. ∴ use dynamic ROM.

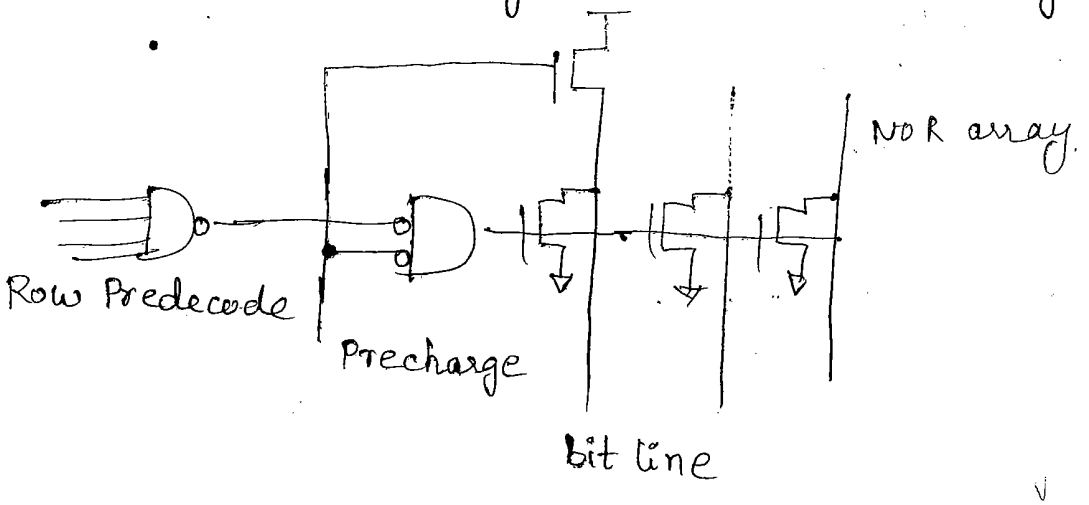


fig: dynamic ROM cktry.

Here word lines are forced low while bit lines are being precharged. This ensures that DC current does not flow. After bit-line pull-ups have been turned off, the word-line drivers are asserted and one word line is active. This reqs careful design of timing chain of sequence of events.