

Introduction to Digital Electronics

John Crowe and Barrie Hayes-Gill

Both Lecturers in the

Department of Electrical and Electronic Engineering

University of Nottingham



Newnes

OXFORD AMSTERDAM BOSTON LONDON NEW YORK PARIS
SAN DIEGO SAN FRANCISCO SINGAPORE SYDNEY TOKYO

10 Semiconductor memories

10.1 INTRODUCTION

In general semiconductor memories are used when fast access of data is required. Semiconductor memories are historically divided into two types. These are Read Only Memory (ROM) and Random Access Memory (RAM). The main members of each family are shown in Fig. 10.1.

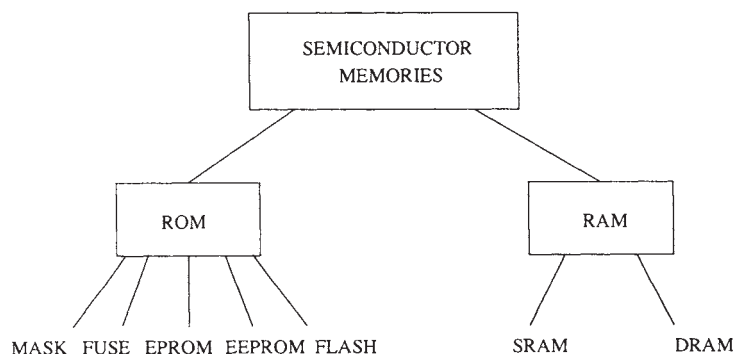


Fig. 10.1 Semiconductor memory types

Read only memory overview

ROM devices are used for storage of data that does not require modification, hence the name 'read only memory'. This definition however, has become less clear over the years and now includes devices whose data are occasionally modified. The original true ROM types are mask programmable ROM and fuse programmable ROM (or PROM). The mask programmed ROM devices are programmed at the factory during manufacture whilst the fuse programmed ROM devices are programmed by blowing small fuses and hence are sometimes called One Time Programmable ROM or OTPROM. Both mask ROM and OTPROM devices are true read only memory devices which are written only once. Other ROM devices that are, paradoxically, written more than once are: Erasable PROM (called EPROM) – these devices are programmed electrically but are fully erased with ultraviolet light; Electrically Erasable ROM (referred to as EEPROM or E²PROM) – these devices can be both programmed *and* erased electrically; FLASH memory – these devices use the same technology as EPROM but

not only are they electrically programmed, they can be erased electrically in a very short time. The great advantage of *all* these ROM devices is that they are non-volatile. This means that when the power is removed the stored data is not lost.

Random access memory overview

The RAM device family is divided into two types. These are Static RAM (SRAM) and Dynamic RAM (DRAM). The SRAM device retains its data as long as the supply is maintained. The storage element used is the transmission gate latch introduced in Chapter 9 (see Fig. 9.18(a)). On the other hand, DRAM devices retain their information as charge on MOS transistor gates. This charge can leak away and so must be periodically refreshed by the user. In both cases these devices are volatile, i.e. when the power is removed the data is lost. However, newer devices are available which muddy the water, such as non-volatile SRAM (NOVRAM) which have small batteries located within their packages. If the external supply is removed the data is retained by the on-board battery. Another relatively new device is the Pseudo Static RAM (PSRAM). This is a DRAM device with on-board refresh circuitry that partially relieves the user from refreshing the DRAM and hence from the outside it has a similar appearance to that of an SRAM device.

The term ‘random access memory’ is given to this family for historical reasons as opposed to the magnetic storage media devices, such as tape drives, which are sequential. In RAM devices any data location can be read and written in approximately equal access times hence the name ‘random access memory’.

Semiconductor memory architecture

Most semiconductor memories are organised in the general form as shown in Fig. 10.2. The chip consists of an array of cells with each cell holding a single bit of data as either a logic ‘1’ or a logic ‘0’. The memory cells are arranged in rows and columns. Each row is individually accessed by a row decoder. As seen in Chapter 4, a decoder is a device which has n inputs and 2^n outputs such that only one output line goes high when an n -input data string is applied. When a row goes high the data for that row is presented at the bottom of the array on the column lines. A second decoder, called a column decoder, is used in conjunction with the row decoder thus allowing a single cell or bit to be individually accessed. Hence by applying a row and a column address the data of a single bit can be either read (R) or written (W) via a single input/output pin under the control of a R/W pin.

In some cases the data is organised as more than one bit, e.g. an eight-bit data word called a *byte* and the chip therefore has eight data inputs and outputs. In this case eight column decoders could be used to access in parallel eight different data bits from a single row. This structure is true for both RAMs and ROMs, although mask programmable ROMs of course would only have a DATA OUT pin(s) and no R/W pin.

This chapter will focus on the structure of a single memory cell and we shall assume that the row lines (sometimes called WORD) and column lines (sometimes called BIT) are decoded such that only one row and column line is high at

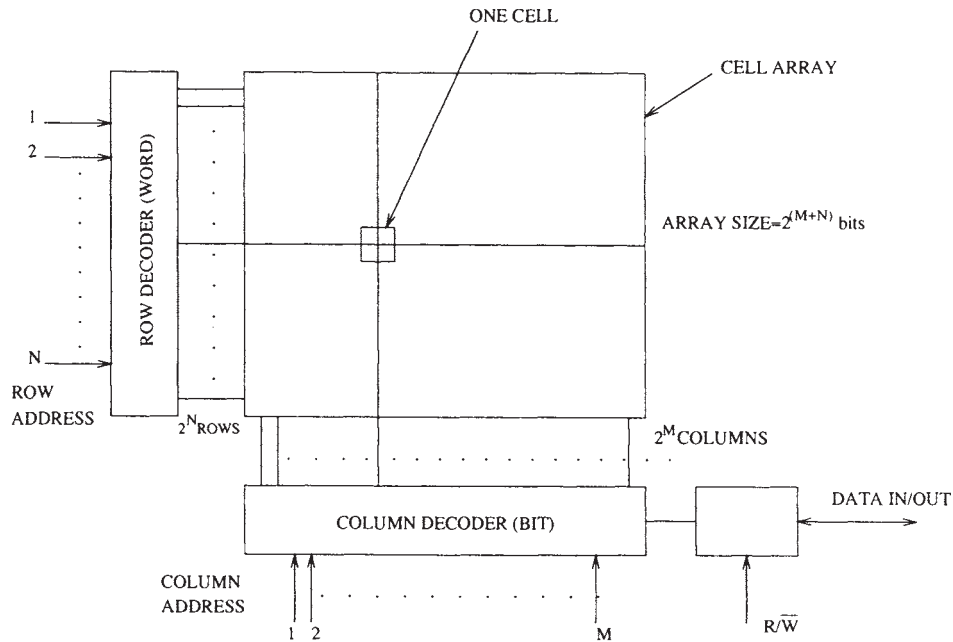


Fig. 10.2 Generic semiconductor memory architecture

any one time to select that particular cell. However, in order to visualise the memory layout, other cells may have to be included. In addition to looking at each cell we shall discuss the relative timing requirements for the input and output signals in order to read and write data.

10.2 READ ONLY MEMORY – ROM

ROMs are read only memory devices, or nowadays more strictly RMM (Read Mostly Memory). As stated, an important property of all ROMs is that they are non-volatile, i.e. when the power is interrupted the data is retained. Most semiconductor memories are implemented with MOS technology due to its high packing density and low power consumption compared to bipolar.

10.2.1 Mask programmed ROMs

Mask programmable ROMs are programmed during manufacture and hence cannot be changed once programmed. These devices are thus used for program storage where the data stored is not required to change.

A section of a programmed NMOS ROM array holding eight bits is shown in Fig. 10.3. This type of array is called a NOR type memory array. Data is stored by the presence or non-presence of an NMOS drain connection to the COLUMN lines. When a ROW is high then the NMOS transistors connected to that ROW will be turned on and the output on the column line will be '0'. Those locations

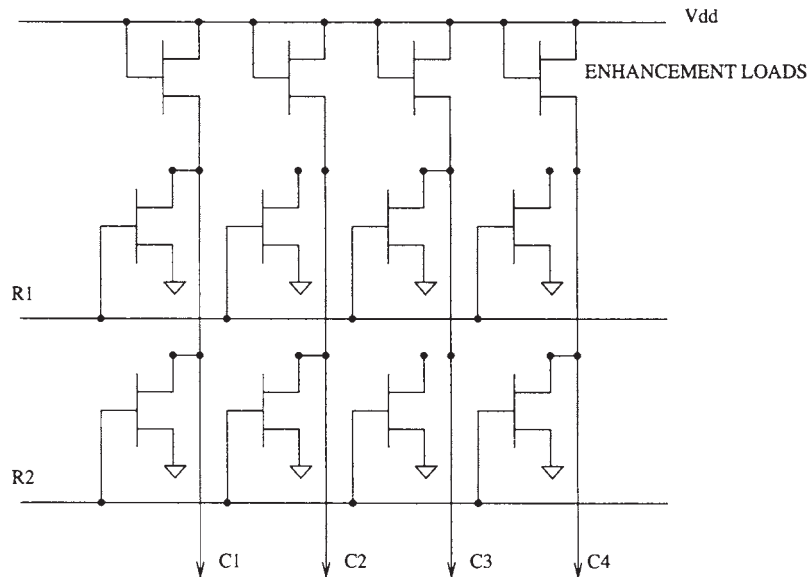


Fig. 10.3 Mask programmed ROM

with no NMOS drain connections on the column line are pulled up to '1' by the enhancement mode NMOS devices acting as a load.

Alternative ways of programming ROM devices during processing exist. For example omitting the source connection at locations where a logic '1' is desired is just as effective. Alternatively the threshold voltage of those transistors which are not to turn on can be increased by using a selective implantation of P-type dopants into the MOS substrate. These processing steps are performed by use of a processing tool called a *photo or electron beam mask* (for further reading on semiconductor processing the reader is referred to an alternative text in this series¹). Since these processing steps are performed near the end of the process then a large inventory of uncommitted ROM wafers with most of the processing completed are kept on the shelf. All that is required is for the customer to specify the data to be stored and hence the mask step can be implemented.

It should be noted that this mask step is very expensive and usually mask programmed ROMs are used only when large volumes are required – greater than 10 000 pieces. The delivery time for such devices is approximately 3–6 weeks and hence mistakes incur a heavy financial and time penalty.

The inputs and outputs for a typical mask programmed ROM are as follows:

- Inputs: address lines; chip enable ($\overline{CE}$); output enable ($\overline{OE}$);
- Outputs: data out;
- Power: V_{dd} and V_{ss} .

The address lines and data output lines require no explanation. The chip enable pin is quite simply used to 'wake up' the chip from its low power, stand-by mode.

¹L. Ibbotson, *Introduction to Solid State Devices*, in this series.

When 'chip enable' is low the device operates normally but when it is high the device is 'off' and in 'stand by'. Since a low turns the device on then this pin is said to be *active low* and hence it is labelled as $\overline{CE}$. This pin can be used as an extra address line for cascading two or more ROM devices when larger memory capacities are required. This is illustrated in Fig. 10.4 where an extra address line is used with an inverter to select either ROM1 or ROM2 via $\overline{CE}$. The output enable pin is used for shared data buses, with microprocessors for example, where the data output pins can be made *tri-state* by holding $\overline{OE}$ high. When $\overline{OE}$ (and $\overline{CE}$) is low then data is presented at the output pins. Since it is active low it is therefore labelled $\overline{OE}$.

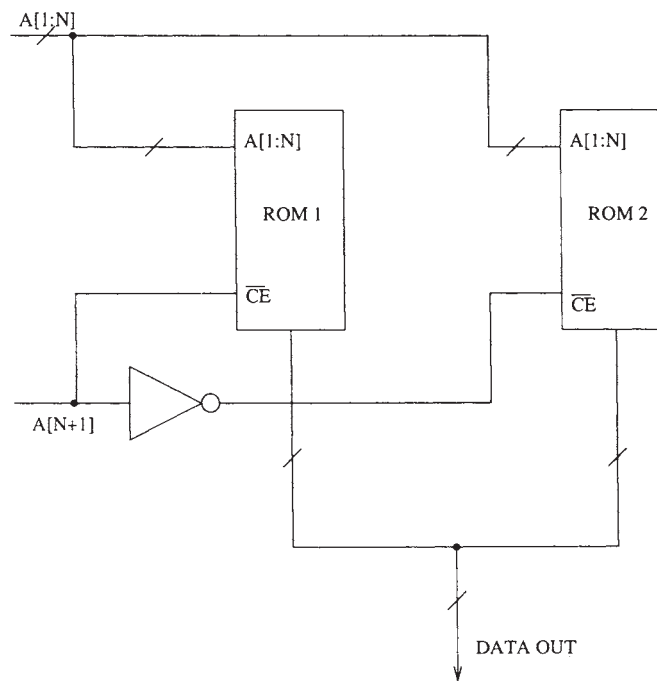


Fig. 10.4 Using $\overline{CE}$ to increase memory storage

The timing diagram for a read operation of a mask programmable ROM is shown in Fig. 10.5. Although these diagrams appear daunting at first they are in fact essential to the system designer so that the device can be correctly interfaced to other devices. This timing diagram shows that we must first set up a valid address and then bring chip enable ($\overline{CE}$) and output enable ($\overline{OE}$) low in order to read data. Three signal types 'a', 'b', and 'c' are labelled in Fig. 10.5. Type 'a' indicates that the signal can be either '1' or '0'; type 'b' is the tri-state or high impedance condition where the outputs are floating (see Chapter 9); and type 'c' refers to an unknown state that occurs whilst the system is changing states. We can see that once the address is set up (i.e. valid) and $\overline{CE}$ and $\overline{OE}$ are low then valid data can be read out of the ROM but only after a time t_{acc} called the address access time. This access time is an important figure of merit for all types of memory

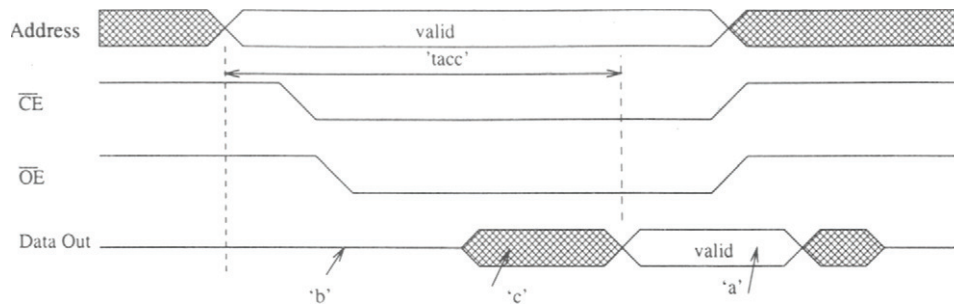


Fig. 10.5 . ROM read timing diagram

devices. For a ROM device it is typically 100 ns whilst for a hard disk on a computer it can be as large as 100 ms!

In semiconductor memories this access time is limited by the resistance and capacitance of the row and column lines which act as an RC delay line. The row and column lines are usually made from metal since this has a low resistance but unfortunately it will also have wiring capacitance as well as the gate capacitance of each of the storage transistors. Hence the larger the array the longer the row and column lines and so the larger the value of the RC component. Large memory arrays, using the same technology, therefore have longer access times.

Like all of the semiconductor memory market, single chip mask ROM packing densities have grown over the years from 256 Kbit in 1986 to 16 Mbit now.

Example 10.1

For the mask programmable ROM layout shown in Fig. 10.3 determine the data appearing on the column lines when both rows 1 and 2 are accessed.

Solution

To access a row the row decoder output (not shown) for that row must be high. When this happens all other row lines will be low. Hence to access row 1 a high must be present on the decoder output. All transistors with the drain connected to the column line will produce a 0 and those not connected will produce a 1 on the column line. Hence:

Row 1 selected: then data out = $C_1 C_2 C_3 C_4 = 0101$

Row 2 selected: then data out = $C_1 C_2 C_3 C_4 = 0010$

Example 10.2

Assume that the memory array shown in Fig. 10.3 are two rows of a larger array of size 64 by 4 bits. What would the row address be to access these two rows if they are the *last two* rows in the array?

Solution

This chip would have 64 rows with each row four bits in length. To address all 64 rows we would need a decoder which has six address inputs and 64 outputs each connected to a single row. The address 000000 will produce a '1' on the first row with a '0' on all the other rows. Hence in order to address the last two rows we will need row addresses of 111110 and 111111, respectively.

10.2.2 PROMs

A PROM, or programmable read only memory device, is programmed by blowing small fusible links which are made of nichrome or polycrystalline silicon. Since fuses are blown the result is irreversible and hence the devices are sometimes called one-time programmable ROMs (or OTPROMs). The early PROM devices were mainly of bipolar form which have a higher effective K than MOSFETs and thus can generate the 15–20 mA needed to blow a fuse without the use of large W/L ratio MOS transistors.

A schematic for an eight-bit bipolar fuse PROM is shown in Fig. 10.6. before the fuses are blown. To blow a fuse the row is selected and the corresponding column line held low in order to program a zero. The power supply V_{pp} is then held at typically 12.5 V for approximately 50 μ s which is sufficient to generate enough power to blow the fuse. In this case the devices are supplied to the customer with a 1 in every location and the user is able to program a 0 where

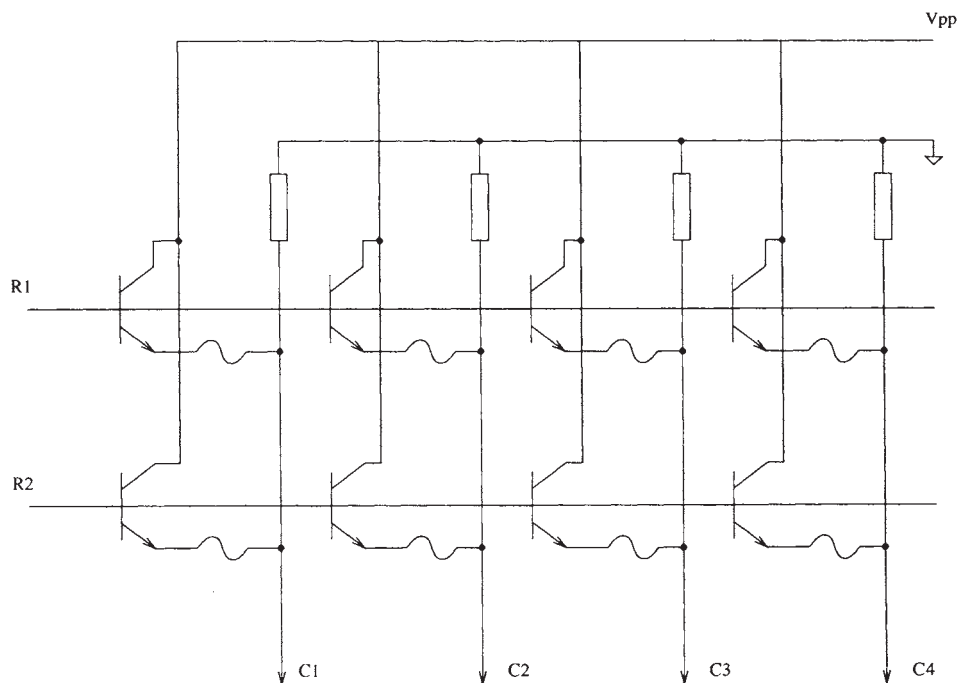


Fig. 10.6 A bipolar fuse programmable ROM – OTPROM

required by blowing the small fuses. During normal operation the power supply voltage is 5 V and so there is no danger of accidentally blowing the fuses.

The increased packing density of MOS transistors over bipolar has resulted in PROMs now being predominantly fabricated using MOS techniques. These MOS devices are in some cases quite simply MOS EPROM devices (see next section) with no transparent window hence stopping erasure by ultraviolet light. In both cases the pin-out for bipolar or MOS OTPROMs are:

- Inputs: address lines; data in; chip enable ($\overline{CE}$); output enable ($\overline{OE}$);
- Outputs: data out;
- Power: V_{dd} , V_{pp} and V_{ss} or V_{cc} , V_{pp} and V_{ee} (for bipolar).

The difference in pin-outs between mask and one-time programmable ROMs is the addition of the higher power supply voltage pin (V_{pp}) and that the data pins are bidirectional, i.e. having both input and output capability for programming purposes. The read timing diagram is the same as for mask programmable ROMs with similar address access times. However, since these devices need to be programmed by the user then a write program timing sequence is supplied in the data sheet. A simplified write or fuse program timing diagram is shown in Fig. 10.7 which actually consists of two stages: data programming and then data verification. As can be seen the address and data are set up first and V_{pp} is pulled up to 12.5 V and V_{cc} to 6.25 V. The actual programming operation occurs when $\overline{CE}$ is held low for typically 50–100 μ s (t_{pw}). To verify that the correct data was programmed into the PROM the data is read back out again by pulling $\overline{OE}$ low for a short time (approximately 200 ns). This process is repeated for every address value and hence the total time taken to program a PROM depends upon the total number of addresses and can take as long as 10 minutes in some cases. This timing diagram is fairly complex but fortunately PROM programmers are readily available that will operate automatically from PCs.

Single-chip OTPROM packing densities are currently at 4 Mbit.

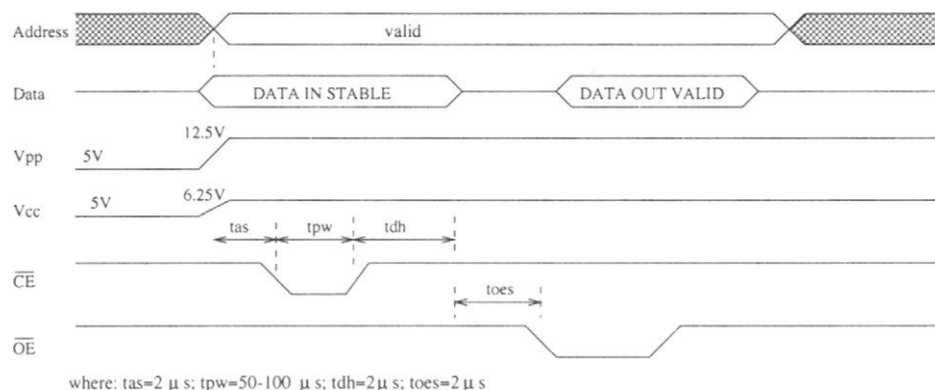


Fig. 10.7 Program timing diagram for fuse programmable ROMs

Example 10.3

A PROM of size 4Mbit, organised as 0.5 Mbit by 8 bit, is to be programmed using the $\overline{CE}$ pulse technique described above. If the $\overline{CE}$ minimum low pulse width is 100 μs then what is the maximum programming time to program all bits? Assume that all other timing parameters can be neglected.

Solution

A total of 0.5 Mbytes require to be programmed. Since each byte is programmed in 100 μs then the total approximate programming time, ignoring t_{as} , t_{dh} and t_{oes} , etc., is

$$T_{\text{total}} = 0.5 \times 10^6 \times 100 \times 10^{-6} = 50 \text{ seconds}$$

10.2.3 EPROM

The problem with the OTPROM devices is that for program development they are inefficient since only one address change will require a completely new device. Hence OTPROMs are only used when the program is settled and contains no known bugs. However, during system development several iterations are usually required, hence devices which can be reprogrammed are more useful. The erasable programmable ROM, more commonly known as an EPROM, is not only electrically programmed but can be fully erased by exposure to ultraviolet light. Hence the device can be reused over and over again until the design is completed. The code on the EPROM can then be transferred to either an OTPROM or a mask programmable ROM.

The cross-section of a single EPROM cell is shown in Fig. 10.8. It consists of two gates:

- (i) FG1 – Floating gate not connected in the array and insulated from the channel by an oxide layer of standard thickness of 0.05 μm .
- (ii) G2 – Polysilicon gate used as the normal memory transistor on the row or word line.

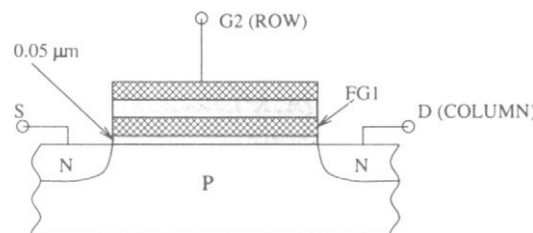


Fig. 10.8 Cross-section of a single EPROM cell

This device fits into the same NOR array as in Fig. 10.3 with inverters on the column line outputs and some additional circuitry. The floating gate (FG1) is used to store charge which thus modifies the threshold voltage (V_t) of G2.

Assume initially that FG1 has zero charge and thus when G2 (or ROW) is held high then the transistor conducts as normal and the column line is 1 (due to the column inverters). Hence unprogrammed EPROM devices have all bits at logic 1.

To write a '0' into the EPROM cell, the row and column lines are both held at a sufficiently high voltage (typically 12.5 V) to cause the drain to enter a condition called avalanche breakdown and a large current flows. The high field from D to S accelerates these electrons to high velocities and some of these electrons (called hot electrons) have sufficient energy to jump to FG1 where they are trapped. (It should be noted that these hot electrons pass over a potential barrier which is larger when looking back from FG to substrate. These hot electrons do not tunnel through the oxide. For a more detailed explanation of the MOS device the reader is referred to Hart²). Now since FG1 is totally insulated then on removing this voltage a negative charge is left on FG1 and under normal conditions it will not leak away for typically 5–10 years.

This negative charge on FG1 will attract holes to the silicon/silicon dioxide interface and so raises the threshold voltage seen by G2. Thus the row voltage at G2 will not turn this device on and so the drain will be pulled up to logic 1 and the column output will be programmed as a logic 0. This avalanche hot electron MOS technique is known as the floating gate avalanche MOS or FAMOS process.

The only way to remove the charge from FG1 and hence erase the device is to make the silicon dioxide conductive by using light of energy greater than its energy gap. This is usually light in the ultraviolet region and the whole process takes approximately 20 minutes to erase. Since these devices are erased optically by the user they must have a transparent window in the package which adds slightly to the cost of the EPROM compared to OTPROMs. It should be noted that daylight also contains the correct wavelength for erasing EPROMs. Consequently once an EPROM has been programmed then the quartz window must be covered with an opaque label. (The MOS OTPROMs discussed in the previous section are nowadays actually made by using EPROM technology but with no transparent window present for erasing. Hence the device can only be programmed once.)

The input and output pins for an EPROM device are exactly the same as for the OTPROM including power supplies (i.e. V_{dd} , V_{pp} and V_{ss}). Consequently the read and write timing diagrams are also identical and it is not surprising that EPROM devices are programmed by the same programmer as that which programmed an OTPROM. However, since an EPROM device is reprogrammable it is necessary to erase the device completely before every write operation. This is because the write operation can only program a logic '0' and will not reprogram a logic '0' to a logic '1'. Finally the number of write/erase cycles is an important factor – this is usually called *endurance* in memory terminology. The use of hot electrons gradually damages the gate oxide and hence the number of write/erase cycles is typically only 100 for an EPROM device.

²B. Hart, *Introduction to Analogue Electronics*, in this series.

In 1986 EPROM packing densities were typically 256 Kbit. Currently this figure stands at 4 Mbit.

Example 10.4

How much current is required to charge the floating gate of a $1\text{ }\mu\text{m}$ EPROM cell to 5 V from 0 V over a write time of $50\text{ }\mu\text{s}$? Assume that the capacitance of the floating gate is $8 \times 10^{-4}\text{ pF}/\mu\text{m}^2$.

Solution

The total capacitance of the floating gate is

$$C_{fg} = 8 \times 10^{-4} \times 1 \times 1 = 0.8\text{ fF}$$

Hence the charge current, derived from $I = C dV/dt$, is

$$I_{\text{charge}} = 0.8 \times 10^{-15} \times 5 \div (50 \times 10^{-6}) = 80\text{ pA}$$

i.e. only a small fraction of the avalanche current (which is of the order of 1 mA) is required to charge the floating gate.

10.2.4 E²PROM

Although the EPROM is an extremely mature technology having been available for more than 15 years its main disadvantage, apart from requiring an ultraviolet light source to erase, is that this erase is not selective and all cells are erased at once. The electrically erasable PROM (EEPROM or E²PROM), on the other hand, is not only programmed electrically but the cells can be erased electrically. This allows the devices to be programmed whilst still in the system. Hence these devices can be used not only for programs and program upgrades but also for storing data that occasionally require updating whilst in use (for example telephone numbers on mobile telephones).

One variant of the E²PROM device employs an MNOS (Metal Nitride Oxide Semiconductor) transistor as the memory element. This MOS transistor, shown in Fig. 10.9, consists of an insulator which is composed of two layers – a silicon nitride layer of thickness $0.05\text{ }\mu\text{m}$ and a very thin silicon dioxide layer of thickness

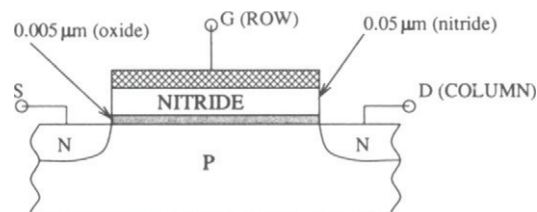


Fig. 10.9 Cross-section of a single E²PROM cell

0.005 μm . The principle of operation relies on the fact that the interface between the oxide and nitride is capable of trapping electrons. Assuming that this interface has zero charge initially then as in the EPROM a '1' on the *row* line will result in the column line reading '1' (remember that an inverter is present on the column line).

To write a '0', 12.5 V is applied to the gate with respect to the source and substrate for approximately 10 ms. The high field generated across the very thin oxide layer allows electrons to tunnel through which are caught by the electron traps at the oxide/nitride interface. The presence of this negative charge will increase the threshold voltage seen by the *row* line. When *row* goes high next time then the MNOS transistor will not conduct and the column output will indicate a logic '0'. Just as in the EPROM the charge can stay at this interface for many years.

The advantage of this tunnelling action is that it is reversible. Hence to erase, all that is required is to apply -12.5 V on the gate which repels electrons from the oxide/nitride interface. No distinct erase operation is needed; only a write cycle is required which loads in either a logic '1' or a logic '0'.

This type of device has high reliability. For example, in the event of a pinhole or a defect in the oxide then only the charge at the oxide/nitride interface above the pinhole or defect will leak away. The rest of the charge at the interface is trapped in the non-conductive layers and so the cell retains its state. The device thus has a very high endurance in that at least 10^4 write/erase operations can be carried out before the device shows signs of degradation.

An alternative to the MNOS E^2PROM cell is to use a variation of the EPROM floating gate cell. In this case the floating gate is extended over the drain and here the oxide is thinned down to 0.01 μm from 0.05 μm . The cross-section for this structure is shown in Fig. 10.10. The advantage of this structure is that the thin oxide region is limited to a small area and hence reliability problems caused by defects and pinholes in the oxide are greatly reduced due to the small area occupied by this thin oxide. The device is programmed by holding the gate at a high positive voltage with respect to the drain. Electrons tunnel through to the floating gate and thus become trapped. To erase the cell the process is reversible and thus a high positive voltage is applied to the drain with respect to the gate and electrons are withdrawn from the floating gate. Note that since in a memory array all the

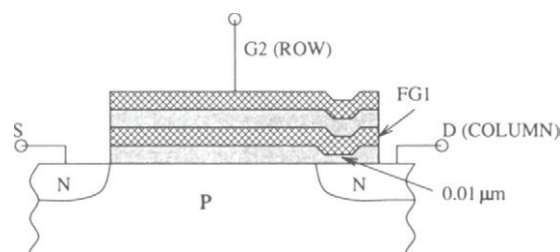


Fig. 10.10 Cross-section of a single floating gate E^2PROM cell

drains in one column are connected together, then the application of a large positive voltage to the drain during erase will also be passed to other cells on that column. Hence with the floating gate E²PROM device each memory element must have its own select transistor to individually access each transistor which reduces the packing density.

Whichever cell is used the pin-out for an E²PROM has typically the following pins:

- Inputs: address; data in; chip enable ($\overline{CE}$); output enable ($\overline{OE}$); write enable ($\overline{WE}$);
- Outputs: data out; RDY/ $\overline{BUSY}$;
- Power: V_{dd} ; V_{ss} .

The first thing to note about these pins is that no high voltage supply (V_{pp}) is necessary. This is generated automatically on chip for both a logic '0' and logic '1'. Since this device can be both written and read then a write enable pin ($\overline{WE}$) is needed to indicate to the array that data is to be written. Hence the read timing diagram for the E²PROM is the same as for the mask programmed ROM but with $\overline{WE}$ held high. Due to the E²PROM cell having a slightly larger capacitance on the row lines its access time is slightly larger than an EPROM at typically 150 ns.

A simplified write timing diagram for an E²PROM is shown in Figure. 10.11. Since the data is usually byte wide then eight bits at a time are written in parallel – this being true for most memories. A pin to indicate that the device is busy writing is provided, called RDY/ $\overline{BUSY}$. When $\overline{WE}$ goes from low to high $\overline{BUSY}$ goes low indicating that the array is busy writing. Only when $\overline{BUSY}$ goes high impedance has the data been correctly written. The complete cycle takes approximately 10ms. Another way of indicating successful data writing is by using the *data polling* method. In this case the data is latched into the chip and an inverted data value appears at the output. When the data has been correctly written this data changes to non-inverted.

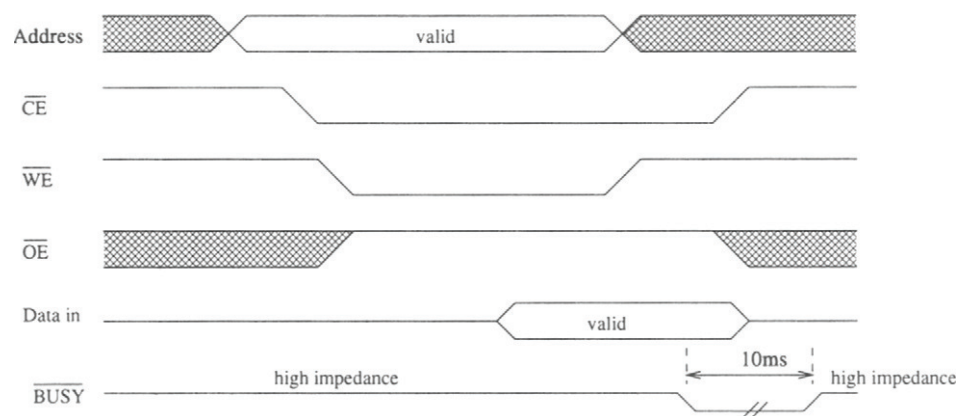


Fig. 10.11 Write timing diagram for an E²PROM device

The current state of the art of E²PROM using MNOS is approximately 1 Mbit compared to 64 Kbit in 1986. However, because the floating gate version has to have an extra transistor per cell these versions are lower capacity being typically 256 Kbit.

Example 10.5

How many pins would an E²PROM chip have if it has a capacity of 256 Kbit organised as 32 Kbits by 8 bit?

Solution

This chip has 32 Kbytes of storage space. To address this storage space we need 15 address lines. Hence the total pin-out would be: 15 Address lines; eight data bits; $\overline{CE}$, $\overline{OE}$, $\overline{WE}$, $\overline{BUSY}$, V_{dd} and V_{ss} , i.e. 29 pins in total.

10.2.5 FLASH E²PROM

Although the MNOS E²PROM device has a relatively high capacity the fabrication of the nitride layer and the very thin oxide layer (0.005 μm) is more expensive to manufacture than the floating gate EPROM transistor. The alternative E²PROM utilising a thinned down oxide transistor over the drain also requires extra processing steps, has an extra select transistor and it too is expensive per bit. To take advantage of the higher packing density of EPROM arrays and its lower manufacturing costs recent processing improvements have allowed a complete thinning down of the oxide under the whole length of the floating gate. The process of charge storage could now be reversed by simply reversing the applied potential. Hence the need for ultraviolet light to erase the device is no longer necessary. This type of memory element is called a Flash E²PROM cell and is shown in Fig. 10.12.

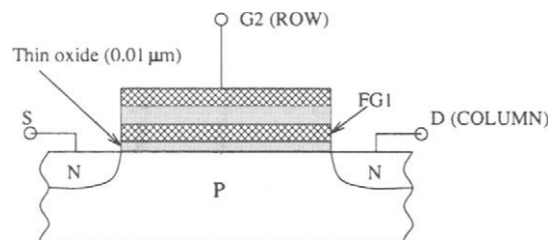


Fig. 10.12 Cross-section for a single flash E²PROM cell

A single flash cell is basically the same as the EPROM cell (i.e. having a floating gate) but the oxide under the floating gate is reduced in thickness from 0.05 μm to 0.01 μm . As a reminder, to program the floating gate EPROM a large positive voltage is applied to the drain and gate with respect to the source and the device

enters breakdown. This creates hot electrons of which some pass to the floating gate therefore programming the cell. Now, to program the flash device 5 V is applied to the drain and a large positive voltage (12 V) is applied to the gate with respect to the source. Since for the flash device the oxide under the floating gate is much thinner then electrons are able to tunnel through this gate oxide using a mechanism called Fowler–Nordheim tunnelling. Unlike the EPROM this process is reversible and to erase the stored charge all that is required is to apply +12 V to the source with the gate grounded (drain floating) thus returning the electrons on FG1 to the source. Since the drain is never held at a high positive voltage then the use of a select transistor to isolate other cells as in the alternative E²PROM is unnecessary. The problem with this structure is that a much larger area of thin oxide exists and hence the device is much more susceptible to damage. In addition if a cell is erased that is already erased then the MOS device will over time acquire a negative threshold voltage and will not program properly. Hence in order to ensure that this problem does not occur *all* cells must be preprogrammed before erasure can start. In this way all cells will be erased by the same amount and the problem of over erasure will not occur. Since all cells are erased at the same time the device is called *flash*. Note that before erasure a preprogram time must be allowed for of approximately the product of the number of bytes and the data write time. For 1 Mbyte arrays the total pre-write and erase time for the complete chip is typically 2–3 seconds. It should also be noted that before *programming* can commence the memory array must be erased (i.e. all cells at logic ‘1’) since the action of programming only writes a logic ‘0’. The diagrams in Fig. 10.13(a), (b) and (c) illustrate these write, erase and read modes on a small 2 × 2 array.

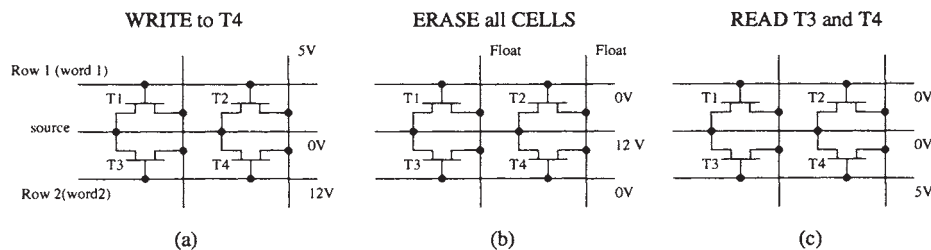


Fig. 10.13 Write, erase and read programming for a flash E²PROM array

The input and output pins for a flash E²PROM are as follows:

- Inputs: address; data in; chip enable ($\overline{CE}$); output enable ($\overline{OE}$); write enable ($\overline{WE}$)
- Outputs: RDY/ $\overline{BUSY}$; data out;
- Power: V_{dd} ; V_{ss} ; V_{pp} .

However, on some of the more recent flash devices the V_{pp} pin is removed as in the E²PROM and is generated on chip.

The timing diagrams for a flash read are the same as in E²PROM but the timing diagrams for write and erase are much more complicated than E²PROM and

require 2–3 write cycles to load in not only data but also commands indicating whether the operation is to be a write, an erase, a data verify, etc. The timing diagrams for these devices will not be included here so as not to detract your interest from this exciting new product! For these details you are advised to consult the manufacturers' data sheets although it is expected that as with the trend of all memory products the complexity of writing and erasing will ease as more and more circuitry is included within the chip. The flash memory device is a relatively new entry into the memory market and is starting to provide strong competition to hard disk drives especially in mobile computers where the lack of moving parts is a great boost for reliability.

Example 10.6

A 1 Mbit flash memory, organised as 128 K by 8 bits, has a byte write time of 10 μ s, a flash erase time of 10 ms and a verify time of 6 μ s. Calculate the total length of time for a complete memory write and then a complete erase.

Solution

To write a byte of data takes 10 μ s but each byte should be verified and so the total write time per byte is 16 μ s. Hence the total write time for the array is: $128 \times 10^3 \times 16 \times 10^{-6} = 2.048$ seconds.

To completely erase we must first prewrite all bytes with a logic 0, then erase and finally perform an erase verify. The total erase time is thus:

$$\begin{aligned} T_{\text{erase}} &= 2.048 + 10 \text{ ms} + 128 \times 10^3 \times 6 \mu\text{s} \\ &= 2.048 + 0.01 + 0.768 \\ &= 2.826 \text{ seconds} \end{aligned}$$

10.3 RANDOM ACCESS MEMORY – RAM

All the programmable ROM devices we have looked at have read times of the order of 100 ns but byte write times from 10 μ s to 10 ms. This is acceptable when holding programs or storing data that change fairly infrequently but in cases where fast write times are needed such as in computers during mathematical calculations then these devices are inappropriate. For such applications RAM devices are more suitable. RAM or random access memories (more appropriately called read write memories) can be both written to or read from in a very fast time of typically 100 ns. They are classed as either static or dynamic.

Static RAMs retain their data indefinitely unless the power to the circuit is interrupted. Dynamic RAMs require that the data stored in each cell be refreshed periodically to retain the stored information. Again the data is lost if the power is interrupted.

10.3.1 Static RAMs

The basic schematic structure of a single static RAM cell is shown in Fig. 10.14. This cell consists of two inverters connected back to back in what is called a flip-flop arrangement. Two switches, S1 and S2, are controlled by the *row* line so that data may be read from or written to this cell. The output of one inverter reinforces the output of the other and hence the state of the circuit is locked. By closing switches S1 and S2 the cell can be accessed and data can be read. Alternatively the cell can again be accessed and by forcing *column* (and $\overline{\text{column}}$) to the required voltage, data can be written accordingly.

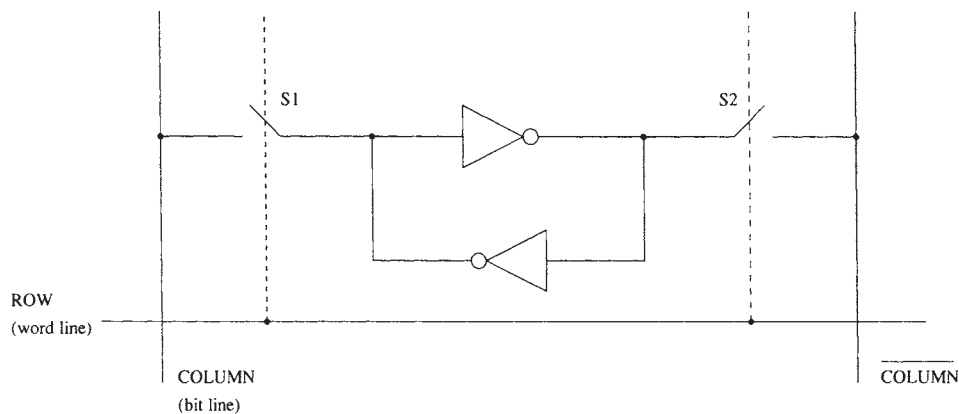


Fig. 10.14 Schematic for a single SRAM cell

SRAMs can be implemented with MOS or bipolar transistors. However, as mentioned in Chapter 9 the bipolar device is fast but consumes more space than the MOS cell and hence is only used in specialist applications. As shown in Chapter 9 BiCMOS is becoming popular when driving large capacitances which is especially important on the row lines for memory devices. Nevertheless, the MOS route is the most economical and hence is still the most popular technology for SRAMs.

The MOS transistor circuit diagram for an SRAM cell is shown in Fig. 10.15. Transistors T5 and T6 act as access transistors to the cell and implement the switches S1 and S2 from Fig. 10.14. Transistor pairs T1/T3 and T2/T4 are NMOS inverters (as described in Chapter 9) and are arranged in cross-coupled form as in Fig. 10.14. Access to the cell is achieved by a high on the *row* line which allows the state of the cell to be read out from the *column* lines. If T3 is off and T4 on then *column* will be high and $\overline{\text{column}}$ low and the cell is said to be in the logic '1' state. To write a '0' in the cell the *row* line is held high and *column* is forced low and $\overline{\text{column}}$ high. Thus T4 turns off and T3 turns on. When *row* is held low again the access transistors T5 and T6 turn off, but due to the cross-coupled nature of the two inverters the data is retained in the cell. However, if the power is removed the data will be lost and hence these devices are called

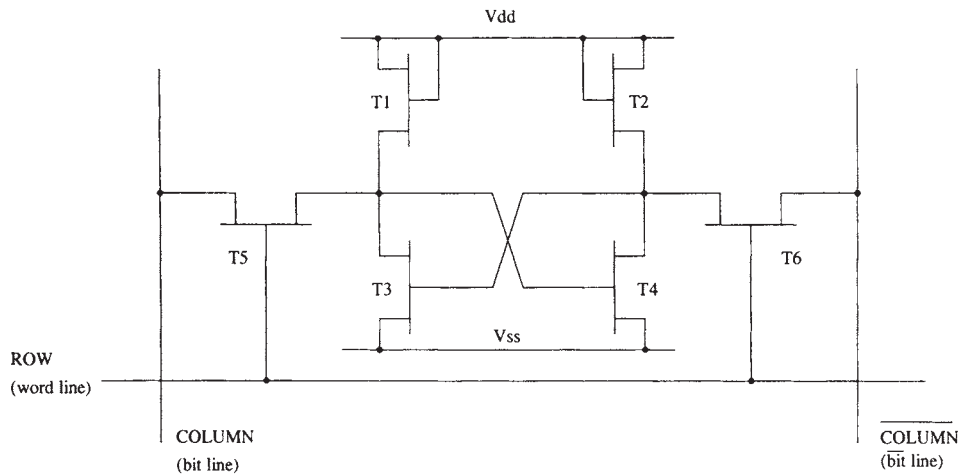


Fig. 10.15 MOS implementation of a single SRAM cell

volatile. It is possible to obtain pin compatible, directly replaceable, non-volatile SRAM (called NOVSRAM) which contains small batteries integrated into the package which cut-in when the main power is lost. However, although these batteries last for two years it is usually not possible to replace them since they are totally encapsulated.

It should be noted that using NMOS transistors as the load can result in unnecessary power consumption since a current will always flow between V_{dd} and V_{ss} when one of the driver transistors (T3 or T4) is on. Having small W/L ratios for the loads will keep this consumption low although two other techniques are used, namely: CMOS logic used for the inverters, or the load is replaced with a high-value polysilicon resistor. A cross-section of one-half of the flip-flop arrangement using this latter technique is shown in Fig. 10.16. This cross-section utilises two layers of polysilicon, one for the gate and the other for the load resistor. The resistor is positioned on top of the transistor thus saving space.

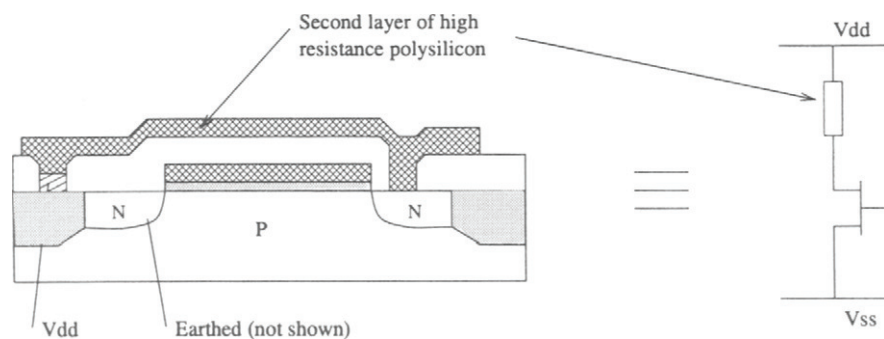


Fig. 10.16 Cross-section for one-half of an MOS SRAM cell

Typical input and output pins for an SRAM are:

- Input: address; data in; $\overline{CE}$; $\overline{OE}$; $\overline{WE}$;
- Output: data out;
- Power: V_{dd} ; V_{ss} .

The $\overline{CE}$ pin is often referred to as the Chip Select pin or $\overline{CS}$. The SRAM read timing diagram is the same as in Fig. 10.5 for the mask ROM but with $\overline{WE}$ held high. The write timing diagram is the same as in Fig. 10.11 for an E²PROM, but without the $\overline{BUSY}$ pin. Data is written into the SRAM as soon as $\overline{WE}$ goes high from its low state.

Low-power stand-by modes are possible with most SRAM devices, reducing the power consumption when not in use by a factor of 1000. This is achieved quite simply by holding $\overline{CE}$ high. In some devices the power can be reduced further by lowering V_{dd} to 2 V without corrupting the stored data. Remember though that removing the power altogether will result in a complete loss of data.

Example 10.7

A single cell of an SRAM uses 20 M Ω load resistors. If the chip operates at 5 V and contains 64 Kbits then what is its power consumption? Assume that the voltage across an MOS transistor when on is 0 V.

Solution

When one of the inverter outputs is low then current flows through the load resistor from V_{dd} to V_{ss} . The power consumed for one cell is thus

$$V_{dd}^2/20 \times 10^6 = 1.25 \mu\text{W}$$

Hence for a 64Kbit SRAM the total power consumed is 80 mW.

Notice that if the power supply is reduced to 2 V then the total power consumed reduces to 12.8 mW. If CMOS is used this static power consumption is reduced but with the added disadvantage of a lower packing density due to more transistors per bit.

10.3.2 Dynamic RAMs

Each SRAM cell either uses four transistors and two resistors or six transistors for CMOS logic. Hence each cell consumes a relatively large area of silicon compared to other semiconductor memory options. Using 1 μm technology a single SRAM cell in CMOS covers approximately 10 $\mu\text{m} \times 10 \mu\text{m}$ and thus a large multibit chip quickly becomes too large to manufacture.

The dynamic RAM or DRAM uses fewer transistors per cell than the SRAM. Hence more bits can be achieved per mm² albeit at the expense of more complex peripheral circuitry to refresh the data. All DRAMs use MOS technology and

data is stored as charge on a capacitor. The highest density DRAM is obtained with the one transistor cell. There are many variations but the most common form is shown in Fig. 10.17. It consists of a single access transistor and a storage capacitor, C_s .

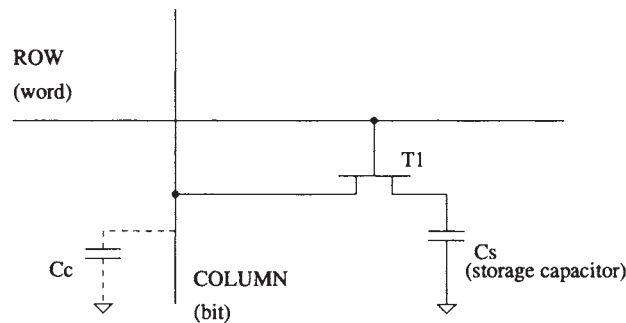


Fig. 10.17 A one-transistor dynamic RAM cell

To read the cell, *row* is held high and hence T1 is turned ON. The voltage stored on C_s is transferred to the *column* line and sensed by a sense amplifier (not shown).

To write to the cell the *row* is again held high turning on T1. If a logic '1' is to be stored then the *column* line is held high and C_s charges up to a logic '1'. When *row* is held low T1 is then turned off thus holding the charge on C_s . The charge stored on C_s can, however, leak away through T1 due to its small leakage current when the transistor is off. Consequently the data must be periodically refreshed which thus requires extra circuitry on the chip. In fact the operation of reading the cell also results in the data being lost. This is because the storage capacitor is designed to be deliberately small for compactness. However, the *column* line feeds many cells and hence the capacitance of this line is very high (shown dotted as C_c). Each time the transistor T1 is turned on the charge on C_s is distributed between these two capacitors and hence its voltage will drop (see Example 10.9 below). Consequently each time the cell is read it must be refreshed. This is usually performed automatically after every read with on-chip refresh circuitry. However, cells that lose their charge through leakage currents via the off transistor must be periodically refreshed by the user.

Typical input and output pins for a DRAM are:

- Inputs: address; data in; $\overline{RAS}$; $\overline{CAS}$; $\overline{OE}$; $\overline{WE}$;
- Outputs: data out;
- Power: V_{dd} ; V_{ss} .

As can be seen two new pins, $\overline{CAS}$ and $\overline{RAS}$, are present with the DRAM and are called 'column address select' and 'row address select', respectively. Since DRAM devices have a large capacity then the number of address lines would be large. In order to reduce the number of address pins these are multiplexed into row and column addresses via the two select lines $\overline{RAS}$ and $\overline{CAS}$. Hence a 1 M by

16 bit DRAM would require 20 address lines but with address multiplexing this can be reduced to 10 address lines plus $\overline{CAS}$ and $\overline{RAS}$. This results in a smaller and hence lower cost chip. The problem with this for the user is that the timing diagrams are more complicated. For example a read timing diagram is shown in Fig. 10.18. Although not indicated, $\overline{WE}$ is held high and $\overline{OE}$ is held low throughout. Notice that the row address is latched into the chip on the falling edge of $\overline{RAS}$ and then the column address is latched on the falling edge of $\overline{CAS}$. After a short time, t_{cac} , called 'access time from $\overline{CAS}$ ' the data becomes valid on *Data out*. It remains valid until $\overline{CAS}$ goes high and t_{off} seconds later *Data out* goes high impedance. Typical values of t_{cac} and t_{off} are 20 ns.

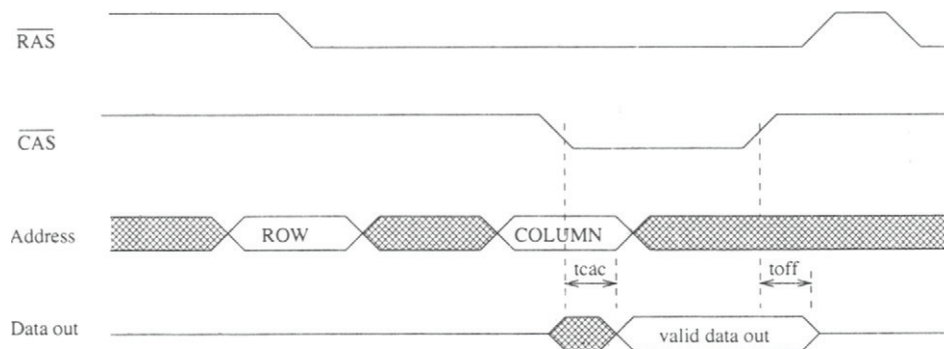


Fig. 10.18 DRAM read timing diagram

To write data into a DRAM two methods are available which provide the user with design flexibility. These are early write and late write. Both these modes are illustrated in Fig. 10.19. In both cases $\overline{RAS}$, $\overline{CAS}$ and the address lines are set up

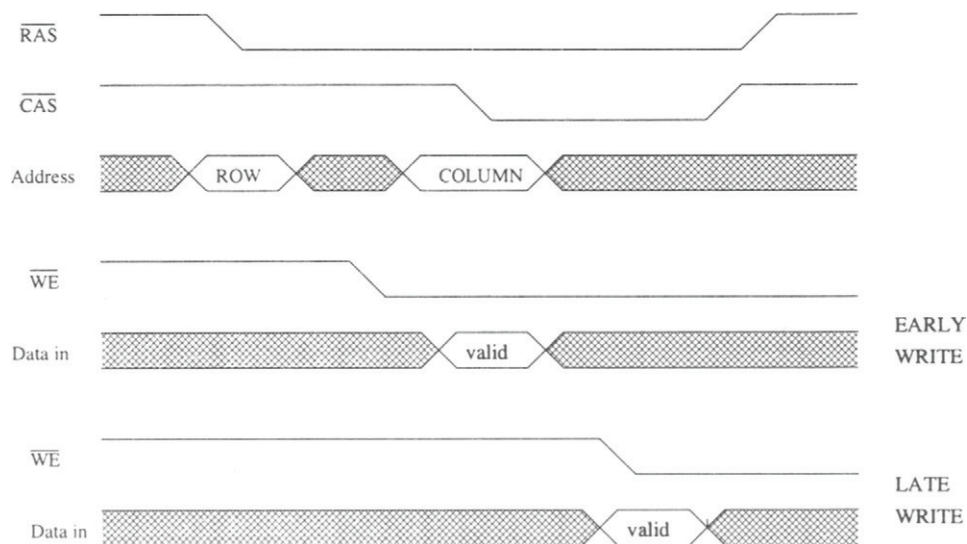


Fig. 10.19 Two techniques for writing data into a DRAM

as before, however, in early write the $\overline{WE}$ line is taken low before $\overline{CAS}$. When $\overline{CAS}$ falls the data is written into the array. For late write the $\overline{WE}$ is taken low after $\overline{CAS}$ at which point data is written into the array.

As stated the charge on the storage capacitor can leak away within a few milliseconds, hence the whole chip must be periodically refreshed. Several methods exist for the user to refresh the data. For example, every time a read operation is performed on an address then the data at that address is refreshed. Another method is called ' $\overline{RAS}$ only refresh' and the timing diagram for this mode is shown in Fig. 10.20. It consists of holding $\overline{CAS}$ high and each time $\overline{RAS}$ goes low the data on the *whole* row that is being addressed will be refreshed. For other refresh modes the reader is referred to the manufacturer's data book.

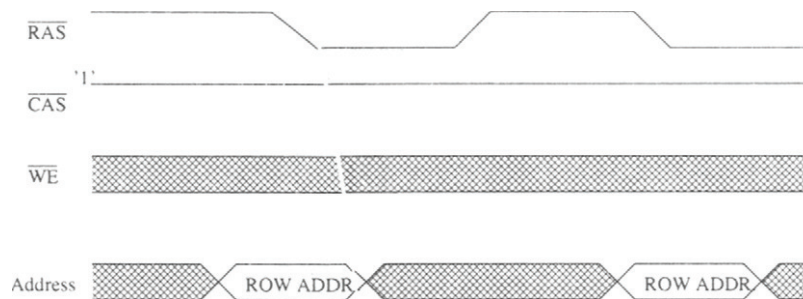


Fig. 10.20 DRAM RAS-only refresh

As can be seen the timing diagrams for DRAMs are more complex than SRAMs and hence some semiconductor manufacturers have added extra circuitry on board to relieve the system designer from complicated refreshing and address multiplexing. These devices are called Pseudo Static RAMs or PSRAM and are pin compatible with SRAMs (plus a $\overline{B\overline{U}\overline{S}\overline{Y}}$ pin) with only minor timing limitations required to allow for the chip to refresh itself.

Typical DRAM sizes are currently at 16 Mbit. The yield of such highly packed, very large, integrated circuits, needless to say, is low. However, for the cost to be kept competitive the yield must be kept high. The yield of DRAMS is dramatically increased by incorporating redundant (or spare) rows and columns of bits which can be exchanged for faulty ones.

Example 10.8

How often must a DRAM cell be refreshed if $C_s = 0.01$ pF, the leakage current of the off transistor is 10 pA and the voltage across the capacitor is 2 V for a logic '1' and 0 V for a logic '0'.

Solution

When C_s is fully charged then 2 V appears across it. This voltage gradually falls as

charge leaks away through the off transistor and must be refreshed before it changes to a logic '0'. The switching point is half-way between the two logic levels i.e. at 1 V. Hence using the equation for the discharge of a capacitor we can work out the time it will take for the voltage to fall from 2 V to 1 V, i.e. $I = C\delta V/\delta t$.

Rearranging to find δt :

$$\delta t = C\delta V/I = 0.01 \times 10^{-12} \times 1/10 \text{ pA} = 1 \text{ ms}$$

i.e. the data must be refreshed every 1 ms.

Example 10.9

Consider the DRAM cell shown in Fig. 10.17. The capacitance of the column line (C_c) is twenty times that of the storage capacitor (C_s) and the voltage across each capacitor is V_c and V_s respectively. If a read signal is applied then what will be the change in voltage on the column line?

Solution

Before a read occurs the charges on the two capacitances are $Q_c = C_c V_c$ and $Q_s = C_s V_s$. When a read occurs the charge is divided amongst these two capacitors which are now connected in parallel. Hence the new charge on the column line is the sum of these two charges:

$$Q_{\text{newc}} = C_c V_c + C_s V_s$$

These two capacitors can be treated in parallel and hence

$$Q_{\text{newc}} = (C_c + C_s) V_{\text{newc}}$$

Equating these two expressions and substituting $20C_s = C_c$ we obtain

$$20C_s V_c + C_s V_s = (20C_s + C_s) V_{\text{newc}}$$

Thus $V_{\text{newc}} = (20V_c + V_s)/21$.

The change in voltage on the column line is δV_c and hence $V_{\text{newc}} = V_c + \delta V_c$. Substituting for V_{newc} into the above equation:

$$V_c + \delta V_c = (20V_c + V_s)/21$$

Rearranging:

$$\delta V_c = (20V_c + V_s)/21 - V_c = (V_s - V_c)/21$$

Hence if V_s is 2 V and V_c is 1 V then the column line voltage will only change by 47.6 mV. This is for a logic '1' stored and a similar value change (but negative) for a logic '0' stored. Consequently the sense amplifiers on the column line must be able to detect this small change in voltage. Needless to say one of the key components of a DRAM is a sensitive, noise-free, high-quality, sense amplifier.

10.4 MEMORY MODULES

In order to increase the memory capacity and to take advantage of the highly reliable semiconductor memories several semiconductor memory chips are incorporated onto a small PCB or substrate called 'memory modules'. Two main types of memory modules are in use today: SIMM and PCMCIA. The SIMM or single in-line memory module devices are those used in most PC based computers and consist of several DRAM chips on one board having either 30 or 72 pins arranged in a single line. The PCMCIA memory modules are actually memory cards which look like a typical bankers card. PCMCIA cards (standing for personal computer memory card international association) have 68 pins arranged in two socketed rows of 34 pins each. Most laptops and some PCs nowadays contain a PCMCIA slot. The card consists of an array of memory chips which could either be: flash, battery backed SRAM, E²PROM or ROM. Three types exist: Type 1 (3.3 mm in thickness); Type 2 (5 mm in thickness); and Type 3 (10.5 mm in thickness). However, only Types 1 and 2 are used for memory whilst Types 2 and 3 are used for hard disks and fax cards. PCMCIA cards that use SRAM have a small lithium disk battery such that data is retained when the card is removed from the computer.

10.5 SELECTING THE APPROPRIATE MEMORY

To conclude this chapter let us try and briefly summarise these memory options as an aid to providing a selection guide. The first requirement is usually capacity – what is the maximum number of bits that can be obtained with a single chip? As you can imagine this figure is continually increasing as semiconductor processing advances. The second consideration is the write time (the read times for all types of semiconductor memory are all very similar of the order of 10 ns–100 ns). Fig. 10.21(a) and (b) shows histograms of the current capacity and the *write* time of each of the memory options discussed in this chapter. As expected these capacities have steadily increased over the years due to improvements in semiconductor processing. It can be seen that the highest capacity ROM is the mask programmable device but this is programmed once at the factory for a high cost. The DRAM is the largest read/write chip available with a very high speed write time. However, it is volatile and requires careful timing considerations. The SRAM has a medium capacity, is easy to interface to other devices, but is expensive per bit and is also volatile. The flash is a new market which is catching up with DRAMs and has the advantage of being both non-volatile and low cost but currently also requires careful timing. It is expected that these devices will become easier to use as this technology advances. EPROMs are very low cost and are ideal for program development but are totally inappropriate where in-circuit modifications are required. This area is covered by E²PROM devices but write times of 10 ms/byte make its use limited. The 10 μ s byte write times for flash make this the product to watch for the future.

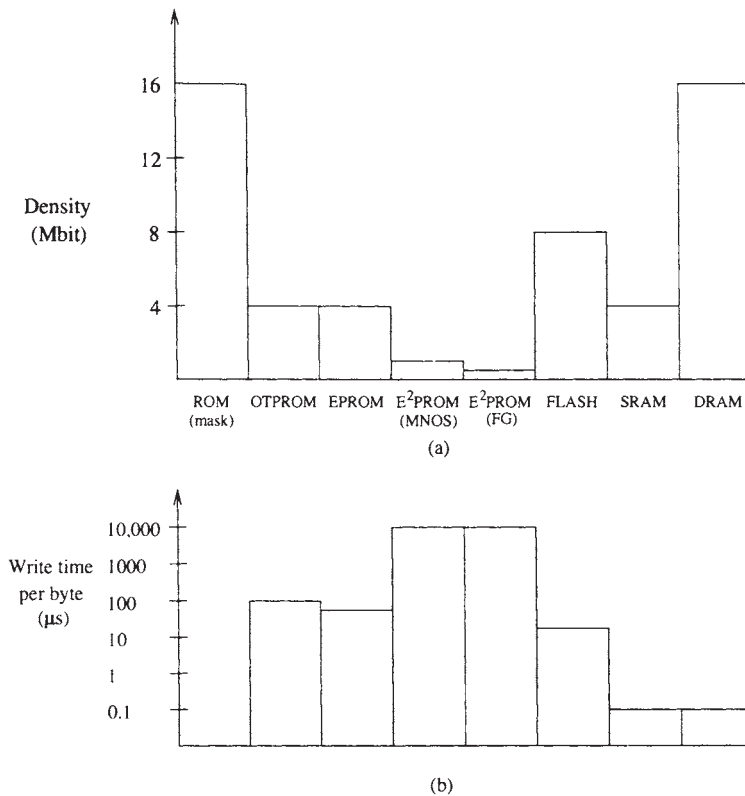


Fig. 10.21 Capacity and write time comparisons for semiconductor memories

10.6 SELF-ASSESSMENT

10.1 Define the terms ROM, RAM, SRAM, DRAM, EPROM, RWM, PSRAM, NOVRAM, OTPROM, E²PROM.

10.2 The power is removed from the devices in Question 10.1. Which devices will retain their data?

10.3 What does non-volatile mean?

10.4 Define access time.

10.5 Which of the memory types uses the most transistors per bit and which uses the least?

10.6 Which has the longest write time: a hard disk or an SRAM?

10.7 Place in increasing write time order the memory options described in this chapter.

10.8 Which type of memory device is used as the main semiconductor memory in PC computers?

- 10.9 What is a SIMM?
- 10.10 What is a PCMCIA card?
- 10.11 What voltage levels should the pins $\overline{CE}$, $\overline{OE}$, $\overline{WE}$ be in order to read data from a DRAM?
- 10.12 Why do you have to refresh a DRAM memory device?
- 10.13 What power supply voltages are needed for EPROM, flash, SRAM, E²PROM?
- 10.14 What happens when $\overline{CE}$ is held high for an SRAM device?
- 10.15 Which of these devices must be erased before they can be written: EPROM; E²PROM; SRAM; flash?

10.7 PROBLEMS

- 10.1 A semiconductor memory chip is specified as 2 K by 8:
 - (a) How many words (exactly) can be stored on this chip?
 - (b) What is the word size?
 - (c) How many total bits can this chip store?
- 10.2 (a) How many pins would a 4 M by 8 bit mask ROM have?
 (b) Repeat (a) for a multiplexed address DRAM.
- 10.3 Illustrate with a sketch how to combine 256 K by 4 bit SRAM chips (having a single $\overline{CE}$ pin and bidirectional I/O pins) to produce a:
 - (a) 256 K by 8 memory;
 - (b) 512 K by 4 memory (use only 19 address lines);
 - (c) 1024 K by 4 memory (use only 20 address lines).
- 10.4 A DRAM has a column line capacitance of 3.8 pF. What value of C_s is required such that at least 250 mV change on the column line occurs when reading a logic zero (0 V) or a logic one (5 V)? Assume that the column line is precharged to 2.5 V before reading.
- 10.5 A flash memory device organised as 256 K by 8 bits has a total write time of 4.2 seconds and a total erase time of 5.8 seconds. If the flash erase time is 9.5 ms then what are the values of byte write time and byte verify time?
- 10.6 A 5 V NMOS SRAM cell is to have a stand-by power consumption of 0.1 mW. What value of load resistor is required? Assume the voltage across the 'on' MOS transistor is 0 V.
- 10.7 A 5V, 512 K by 8 bit CMOS SRAM is to be used in a three hour battery powered application. If the leakage current of an off transistor is 0.5 nA then what is the total static power dissipation and the total ampere hours required. Comment on the validity of your answer.