Unit 4

Addressing

4.1 Addressing Techniques

Each instruction needs data on which it has to perform the specified operation. The operand (data) may be in the accumulator, general purpose register or in some specified memory location. Therefore, there are various way to specify data. The techniques of specifying the address of the data are known as addressing modes. The important addressing modes are as follows :

(i) Direct (or absolute) addressing mode
(ii) Register addressing mode
(iii) Register indirect addressing mode
(iv) Immediate addressing mode
(v) Implicit addressing/Implied addressing mode
(vi) Indexed addressing mode
(vii) Based addressing mode
(viii) Based indexed addressing mode
(ix) Relative addressing mode
(x) Relative indexed addressing mode
(xi) Page addressing mode
(xii) Stack addressing mode

Intel 8085 use addressing modes only from (i) to (v). In addition to (i) to (v) addressing modes other addressing modes are also used by other microprocessors.

Advantage of addressing modes :-

1. To give programming versatility to the user by providing facilities like pointers to memory, counter for loop control, indexing of data and program relocation.
2. To reduce the number of bits in the addressing field of the instruction.

4.1.1 Direct (or Absolute) Addressing

In direct addressing the address of the data (operand) is specified within the instruction itself. Examples of direct addressing are :

(i) **STA 2500H** ; Store the contents of the accumulator in the memory location 2500 H.

(ii) **LDA 2500H** ; Load accumulator with the contents of the memory location 2500 H.

In example (i) above, 2500H is the memory address where data are to be stored. The memory address 2500H is given in the instruction itself. In this case it is understood that the source of data is accumulator.

(iii) IN 01 ; Read data from an input device whose address is 01.

In this instruction 01 is the address of an input device from where data are to be read. In this instruction it is implied that the data will be transferred from the input device to the accumulator.

4.1.2 Immediate Addressing

In immediate addressing the operand is given in the instruction itself. The examples of immediate addressing are :

(i) **MVI A, 06** ; Move 06 to the accumulator.

(ii) **ADI 06** ; Add 05 to the contents of the accumulator.

(iii) **LXI H, 2500H** ; Load H-L pair with 2500 H.

4.1.3 Relative Addressing

In relative addressing a signed displacement is added to the current value of the program counter to form the effective address. This mode of addressing is commonly used in branch (or jump) instructions. The displacement is a signed quantity so to allow either a forward or a backward jump from the location pointed out by the program counter. This mode of addressing is also known as *PC relative addressing*. The effective address specifies memory location in relation to the current value of the program counter.

4.1.4 Relative Indexed Addressing

In this mode of addressing the contents of the program counter and the contents of the index register are added together to form the effective address.

4.1.5 Indexed Addressing

The operand's offset is determined by adding an 8-bit or 16-bit displacement (given in the instruction) to the content of the index register.

4.1.6 Based Addressing

In this mode of addressing the operand's offset is the sum of the content of the base register and the 8-bit displacement given in the instruction.

4.1.7 Based Indexed Addressing

In based indexed addressing the contents of the base register and the contents of the index register are added together to form the effective address. The base register contains a base address and the index register contains an index

4.1.8 Paging

In paged mode of addressing the memory is divided into a number of equal length pages. The page size is 256 bytes for 8-bit microprocessors and 4KB for 16-bit microprocessors. The microprocessor contains a page register to hold page number. The instruction contains an offset. The offset indicates the address within the page with reference to the starting address of the page. The numbering of pages is done in such a way that in case of 16-bit memory address, the 8-bit page number contained in the page register indicates 8 MSBs of the operand address. The advantage of this mode of addressing is that a fever bits in this instruction are required to indicate the memory address. This result in shorter instruction and faster execution.

0 Page Addressing : Some microprocessors provide pages addressing in the limited form without employing a page register. In such a case the paged addressing is restricted only to the first 256 memory locations. The 8MSBs of the address of these 256 memory locations are zero and this part of the memory space is called 0-page (or base page). Such mode of addressing is called *0-page addressing*.

4.1.9 Register Addressing

In register addressing the operands are located in general purpose registers. In other words the contents of a register is the operand. Therefore, only the name of the registers are to be specified in the instruction. Examples of register addressing are :

(i) **MOV A, B** ; Transfer the contents of register B to register A.

The opcode of this instruction is 78H. In addition to the operation to be performed, the opcode also specifies the addresses of the registers mentioned in the instruction. The opcode 78H in binary form is 01111000. The first two bits 01 denote MOV operation, the next three bits 111 are the binary code of register A and the last three bits 000 are the binary code of register B of Intel 8085.

(ii) **ADD B** ; Add the contents of the register B to the contents of the accumulator.

In register indirect addressing the address of the operand is given indirectly. The contents of a register or a register-pair are the address of the operand. Compare this with the register addressing where the content of the register is the operand.

Examples are :

(i) **LXI H, 3400H** ; Load H-L pair with 2400H.

MOV A,M ; Move the content of the memory location, whose address is in H-L pair (i.e. 2400H), to the accumulator.

In this example MOV A, M is an example of register indirect addressing. For MOV A, M instruction the operand is in a memory location whose address is not directly given in this instruction. The address of the memory location is stored in H-L pair, which has been specified by the earlier instruction in the program, i.e. LXI H, 2400H.

(ii) **LXI H, 200 H;** Load H-L pair with 2200 H.

ADD M ; Add the contents of the memory location, whose address is in H-L pair, to the contents of the accumulator.

In this example the instruction ADD M is an example of register indirect addressing.

4.1.10 Implicit (or Implied) addressing

This mode of addressing is also called implied addressing or inherent addressing. Some instructions operate on only one operand which is in the accumulator. So address need not be specified. Such addressing is known as implicit addressing.

Example are :

(i) **RAL ;** Rotate the contents of the accumulator left through carry.

(ii) **RLC ;** Rotate the contents of the accumulator left.

(iii) **CMA ;** Take complement of the contents of the accumulator.

Many instructions use two operands. For one of the operands they use implicit addressing while for the other operand they employ any one of the other addressing modes

Before discussing other addressing modes. Certain terms are to be defined. A large memory is divided into segments. The memory address of an operand consists of two components : the starting address of the segment and an offset. The starting address of the segment is supplied by the processor. The operand is placed at an offset within the segment with reference to the starting address of the segment. The offset is determined by adding any combination of three offset address elements : displacement, base and index. The combination depends on the addressing mode of an instruction to be executed. The offset is also called **effective address**. The memory address of an operand is given by

The memory address of an operand = Starting address of the memory segment + offset

Displacement. It is an 8-bit or 16-bit immediate value given in the instruction.

Base. It is the content of the base register.

Index. It is content of the index register.

4.1.11 Stack Addressing

In this mode of addressing the address of the operand is specified by the stack pointer (SP). The length of instruction is shortest because it does not include any address of the memory location or mention and register (just like implied mode of addressing). The contents of SP are automatically incremented or decremented after each stack operation. PUSH instruction is used to save the contents of a register pair into the stack. The POP instruction is used to transfer the contents from the stack to the register pair into the stack. The POP instruction is used to transfer the contents from the stack to the register-pair. SP contains the address of the memory location of the stack from (or to) which data are to be transferred.

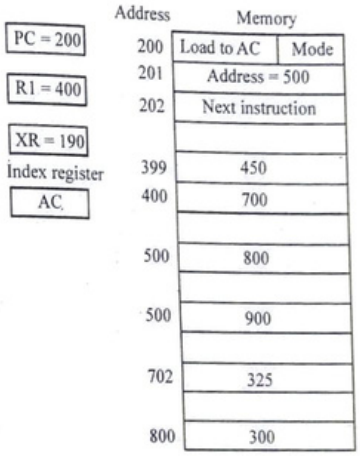
The stack addressing is employed in the following cases :

(i) When PUSH and POP instructions are used in a program by the programmer.

(ii) When CALL instruction is used to call a subroutine. Before a program jumps from the main program to a sub-routine the contents of the PC are saved. When program goes back form the subroutine to the main program the contents of PC are restored. For this RET instruction is used at the end of a subroutine. This uses stack addressing.

(iii) When interrupt occurs the contents of important registers are saved into the stack. For this stack addressing is used.

Numerical Examples



Addressing Mode	E.A (Effective Address)	Content of AC
1. Direct	500	800
2. Immediate	201	500
3. Indirect	800	300
4. Relative	702	325
5. Indexed	600	900
6. Register	—	400
7. Register indirect	400	700

4.2 Registers

Iordaing arithmetical or logical operations, an where data may be kept temporarily. stach memory locations duilt in the ALU are called its registers.

"Registers are electronic devices that tempararily hold values in the form of 1 and 0."

Registers are temporary storage unit of a computer that keep both data and instruction in a binary form. If you happen to use elevator/lift in tall multistorey building, the floor number is indicated by a digital number. The floor number harges as the lift goes up or down, the value of the floor number is stored temporily in a register and shown. As the floor changes this value will also change. A similar role is played by registers some examples of registers.

4.2.1 Index registers

An index register in a computer's CPU is a processor register used for modifying open and addresses dyring the run of a program typically for doing vector/array operations.

The content of an index register is added to an immediate address to form the "effective" address of the actual data. special instructions are typically provided to test the index register and if the test fails, in crements the index registers by an immediate constant and branches typically to the start of the loop. Some instruction sets allow more than one index register to be used in that case additional instruction field specify which index register to use. While normally processors that allow an instruction to specify multiple index registers add the content together, I_{BM} had a line of computers in which the contents were ordered together.

In early computers without any form of indirect addressing, array operations had to be performed by modifying the instruction address which required seural additional program steps and used up more computer memory, a scarce rescurce in computer installations of the early era.

Index register, commonly known as a B-line in early British computers, were first used in British mancherter mark 1 computer in 1949. In general index registers become a standard part of computers during the technology is second generation raughly 1954–1966 most machiner in the IBM 700.

4.2.2 General Purpose Registers

The general purpose registers may be used for temporarily storing data. These registers may be combined to form register pairs in order to handle larger size data. General purpose registers are also known as programmable registers as they may be programmed by the user with the help of instructions.

4.2.3 Special Purpose registers

It hold program state, then usually include the program counter, also called the instruction pointer and the status register the pogram counter and status register might be combined in a program status word (PSW) register. The a forementioned stack pointer is sometimes also

included in this group. Embeded microprocessor can also have register corresponding to specialized hardware elements.

4.2.4 Overflow

In computer processors, the overflow flag is usually a single bit in a system status register used to indicate when an arithmetic overflow has occured in an operation, indicating that the signed two's complement result would not fit in the number of bits used for the operation. Some architecture may be configured to automatically generate an exception on an operation resulting in overflow.

4.2.5 Carry

In computer architecture the carry flag is a single bit in a system status (flag) rgister used to indicate when an arithmetic carry or borrow has been generated out of the most significant ALU bit position. The cary flag enables number larger than a single ALU width to be added/ subtracted by carrying (adding) a binary digit from a partial addition/substraction to the least significant bit position of a more significant word. It is also used to extent bit shifts and rotate in a similar manner on many processor for substactive operations, two convention are employed as most machine set the carry flag on borrow while some machine (such as the 6502 and the PIC) instead reset the carry flag on borrow.

4.2.6 Shift register

In digital circuits, a shift register is a cascade of flipflops sharing the same clock, in which the output of each flip flop is connected to the data input of the next flip flop in thechain resulting in a circuit that shifts by one position the "bit array" stored in it, shifting in the data present at its input and shifting out the last bit in the array at each transition of the clock input.

Generally, a shift register may be multidimensional, such that its 'data in' and stage outputs are themselves bit array ; this is implemented simply by running several shift registers of the same bit length in parallel.

Shift registers can have both parallel and serial input and outputs. These are often configured as serial-in, parallel out (SIPO) or "Parallel in-serial out) (PISO).These are also types that have both serial and parallel inptu and types with serial and parallel output. There are also bidirectional shift registers which alow shifting in both direction L → R or R → L.

4.2.7 Memory Buffer Register

The Memory Buffer Register (MBR) is the register in the central processor that stores the data being transferred to and from the immediate access store. It acts as a buffer allowing the central processor and memory units to act independently without being affected by minor differences in operation. A data item will be copied to the MBR ready for use at the next clock pulse when it can be either used by the central processor unit or stored in main memory. This register holds

the contents of the memory which are to be transferred from memory to other components or vice versa. A word to be stored must be transferred to the MBR from where it goes to the specific memory location and the arithmetic data to be processed in the ALU first goes to MBR and then to accumulated register and then it is processed in the ALU. Memory Buffer Register is used both in programming and hardware.

In other words buffer is temporary holding area for data; unlike register it stores more than one piece of data. It is normally used between the CPU and the input/output devices. You may have noticed, you send some data for printing; your computer is busy in printing the data. But your computer is busy in printing the data. The reason is, there is a buffer and now printer is printing the data from that buffer and now printer is printing the data from that buffer. The advantage is, it clears the computer for your working. There is a keyboard for your working. There is a keyboard buffer in the keyboard to store few keys. The size of the buffer depends on the nature work of the I/O device.

4.2.8 Accumulator

This register is generally referred to as register AC. It is used for storing data and for doing arithmetic and logic operations. The result of an arithmetic or logical operation is automatically stored in this register.

4.2.9 Stack Pointer

Stack may be defined as a set of memory locations and the stack pointer may be defined as the indicator to these memory locations. stack memory locations are used by a microprocessor for storing data temporarily for execution of a program.

For the basic computer we shall consider the following registers :

(a) One register for holding data called data register-DR.

(b) One register for storing instruction called instruction Register- IR.

(c) A register for holding the address of memory word called address register-AR.

(d) A register for holding temporary data generated during processing. This register is named as temporary register (TR).

(e) A processor register (Accumulator-AC) is required for doing operations on data. This processor register holds data on which addition, substraction, multiplication, shift and logical operations are to be carried out.

(f) A register that will act as a counter and will hold the address of next instruction. Such a register is named as program Counter-PC.

(g) Register for inputting and outputting data. Input register (INPR) will hold data obtained from user through input devices like keyboard. Output register (OUTR) will hold data that need to be sent to output devices like monitor, printer, etc.

Thus, we will need all these registers to hold data temporarily as well as we will need memory unit and control unit. All these are shown in figure 4.1

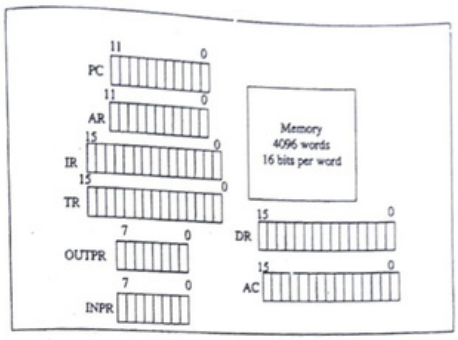


Fig. 1 : Registers and Memory units

☞ Memory Word is the name given to a set of binary cells in memory. A word is treated as an entity and moves in and out of memory as single unit. The PC and AR are of size 12 bits each (0–11). The registers IR, TR, DR and AC are of size 16 (0 to 15). The two other registers namely OUTER and INPR are of 8 bits i.e. (0–7) each.

4.2.10 Floating Point Representation

The floating point representation of a number has two parts. The first part represent a signed, fixed point number called the mantissa. The second part designates the position of the decimal (or binary) point and is called the exponent. The fixed point mantissa may be a fractio or an integer. For example the decimal number +6132.789 is represented in floating with a fraction and an exponent as follows :

Fraction	Exponent
+ 0.6132789	+ 04

The value of the exponent indicates that the actual position of the decimal point is four position to the right of the indicated decimal point in the fraction. This representation is equivalent to the scientific notation $+ 0.6132789 \times 10^{+4}$

Floating point is always intepreted to represent a number in the following form :

$$m \times r^e$$

Only the mantissa m and the exponent e ae physically represented in the register (including their signs). The radix r and the radix point position of the mantissa are always assumed. The circuits that manipulate the floating point numbers in registers conform with these two assumptions in order to provide the correct computational results.

A floating point binay number is represented in a similar manner except that it uses base 2 for the exponent (for examle, the binary number + 1001.11 is represented with a 8 bit fraction

and 6 bit exponent as follows :

Fraction	Exponent
01001110	000100

The fraction has a 0 in the leftmost position to denote positive. The binary point of the fraction follows the sign bit but is not shown in the registes. The exponent has the equivalent binary number +4. The floating point numbe is equivalent to

$$m \times 2^{e} = +(.1001110)_2 \times 2^{+4}$$

A floating point number is said to be normalized if the most significant digit of the mantissa is non zero. For example the decimal number 350 is normalized but 00035 is not. Regardless of where the position of the radix point is assumed to be in the mantissa, the numbe is normalized only if its left most digit in non zero. For example the 8-bit binary number 00011010 is not normalized because of the three leding 0's. The number can be normalized by shifting it three position to the left and discarding the leading 0's to between 1010000. The three shift multiply the number by $2^3 = 8$. To keep the same value for the floating point number, the exponent must be substracted by 3. Normalized numbers provide the maximum possible precision for the floating point number. A zero can not be normalized because it does not have a non zero digit. It is usuually represented in floating point by all 0's in the mantissa and exponent.

Arithmetic operations with floating point numbers are more complicated thatn arithmetic oprations with fixed point numbers and their execution taken longe and require more complex hardware. However floating point representation is must for scientific computations because of the sealing problem involved with fixed point computation. Many computes and all electronic calculators have the built in capability of performing floating point arithmetic operation computes that do not have hardware for floating point computations have a set of subroutines to help the user program scientific problems with floating point numbers.

4.2.11 Status Information Register

The status register also called flag register, holds 1-bit flag to indicate certain conditions that arise during arithmetic and logical operations. The important conditions shown by flag or status registers are :

Carry – indicates whether there is overflow or not.

Zero – indicates whether the result is zero or nonzero.

Sign – indicates whether the result is plus or minus.

Parity – indicates whether the result contains odd number of is or even number of 1s.

4.2.12 Buffer Resisters

A memory buffer register act as buffer register allowing the processo and memory units to act independently without being affected by minor difference in operation. A data item will be copied to the MBR eady for uset at the next clock cycle when it can be either used by the processor of reading and writing or stoed in main memory after being written.

The register hold the content of the memory which are to be transferred from memory to other components ov vice versa. A word to be stored must be transferred to the MBR from where it goes to the specific memory location and the arithmetic data to be processed in the ALU first goes to MBR and then to accumulator register and then it is processed in the ALU.

4.2.13 Program Counter

This is a register which deals with the order for excution of instructions. This acts like a pointer which indicates the subsequent memory location where instruction is stored. After one instruction is executed, the program counter gets incremented by one to indicate the location of the next instruction in the serial order.

4.2.14 Word Size and Register Size

We shall now consider the basic computer to be a 16 bit computer. It means that each word of this computer contains 16 bits of data. If there are 4096 words in the memory unit, then we will need 12 bits to address or locate each of these words. This is so because $2^{12} = 4096$. Since DR, IR, AC and TR hold data, these registers should also be of 16 bits because each word consists of 16 bits.

AR and PC store addresses of memory words. Therefore, AR and PC registers are of 12 bits each because we need to store 4096 addresses and this is possible with a 12 bit register, as $2^{12} = 4096$. Thus, 12 bits would from 4096 different combinations.

INPR and OUTER (Input and Output register) are taken to be of 8 bits each. They are supposed to transfer 8 bits at a time to the memory or to other registers or to output devices.

☞ If we have 16 bit word, we wqould move it in two parts of 8 bits each. This is because, we consider that the transfer of bits is done using one byte (or 8 bits) at a time through the bus.

The bus connects different parts in a mother board to move 8 bits at a time. The rest of the 8 bits would follow. Thus, the 16 bits are put together as one word.

Table 4.1 gives the summary of different register and their functions as well as the number of bits they store temporarily.

Table 4.1 : Register and their contents for basic computer

Register symbol	Register name	Number of bits	Contents
DR	Data Register	16	Memory operand
IR	Instruction Register	16	Instruction code
AR	Address Register	12	Address of memory word
TR	Temporary Register	16	Temporary data
AC	Accumulator	16	Data to be processed
PC	Program Counter	12	Address of instruction
INPR	Input Register	8	Entered character
OUTER	Output Register	8	Displayed character
MBR	Memory Buffer Register	16	To act as buffer for main memory

4.3 Main Memory

Main memory is a set of many register connected together. These registers are called storage registers. The registers discussed in the earlier section are called temporary storage units. The difference between these two types of registers lies in the nature of circuits associated with them. The temporary storage registers have combinational circuits, because many arithmentc logical as well as data transfer operations are performed on them. The storage registers found in memory have circuits that facilitate just reading and writing of data from and to memory.

Figure 4.2 shows a block diagram of memory organisation of the basic computer. The memory depicted in this figure contains 4096 words of 16 bits each. These words are given.

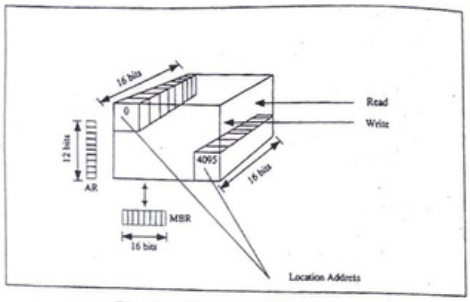


Fig. 2 : Memory organisation

4.3.1 Main Memory

Our human brain contains a large number of memories or instructions we received from many sources such as office, school, work, life experiences, etc. When information is received from one or more of our senses, we react based our memories. The computers memory functions much the same as your own brain. The CPU (Central processing unit) along with this internal memory makes up the computer's "brain."

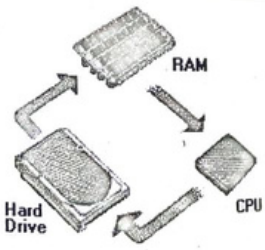


Fig. 3 : Component of computer

Memory is a term that is used to represent storage that has "almost" instantaneous access by the CPU or other processor. Every computer will have some type of memory. However, the memory we speak of the most often is Random

Access Memory (RAM). RAM is the primary memory used when we "boot" (start up) a computer. During boot-up the PC's operating system and device drivers are loaded into RAM.

RAM (random access memory) has many functions, but its primary function is to store programs and drivers for the various system components. It is probably the most important type of memory in your PC. Each time we start up the computer, memory is filled with instructions to give it in ability to do work. You control which instructions will go into the computer's brain by loading or starting software programs. Once a program is in memory, the computer will evaluate inputs from many devices and react accordingly. The types of instructions that can be placed in memory are:

- How to print a word or document to a printer.
- How to calculate the solution to a math problem.
- How to send information over a telephone line to another computer.

Each time you start another program you empty the computer's brain and install a different set of instructions, thus giving the computer a different ability. The most typical input is commands you type on the keyboard. When you enter any input command from your keyboard, the CPU (microprocessor) processes the command and calls for data (a program or data file) to be copied from a storage device (hard disk, floppy, CD-ROM, etc.) into memory, where it is stored temporarily until the CPU needs it. The memory then provides the data to the CPU more quickly than from the storage device. This data may be a program or actual data such as a letter you previously typed. "RAM" is what you are referring to when you say "my system has 8, 32 MB of memory".

Types of Memory

There are two kinds of computer memory: Primary and Secondary memory. Primary memory is accessible directly by the processing unit. RAM is an example of primary memory. As soon as the computer is switched off the contents of the primary memory is lost. The primary memory as you know in the computer is in the form of IC's (Integrated Circuits). These circuits are called Random Access Memory (RAM). Each of RAM's locations stores one byte of information. (One byte is equal to 8 bits). A bit is an acronym for binary digit, which stands for one binary piece of information. This can be either 0 or 1. You will know more about RAM later. The Primary or internal storage section is made up of several small storage locations (ICs) called cells. Each of these cells can store a fixed number of bits called word length.

You can store and retrieve data much faster with primary memory compared to secondary memory. Secondary memory such as floppy disks, magnetic disk, etc., is located outside the computer. Primary memory is more expensive than secondary memory. Because of this the size of primary memory is less than that of secondary memory. We will discuss about secondary memory later on.

Some of the terms you will hear concerning memory is: RAM, DRAM, SRAM, EDO RAM, SDRAM, ROM, PROM, EPROM, EEPROM, FLASH, internal and external cache, etc. In this chapter we will cover the three basic types of memory found in a computer:

2.4 RAM - Random Access Memory

2.5 ROM - Read Only Memory.

2.6 Cache - A fast storage type of memory.

Before we confuse you more, let us give you a table which will list these types and more, and the features of each and every:

Memory Type	Features and usefulness
ROM	Read only Memory used for BIOS chips, CMOS chips, and Special function Chips.
RAM	Random access memory - FPM RAM, EDO & BEDO RAM, Synchronous DRAM (SDRAM)
DRAM	Dynamic RAM-Actual memory chips on SIMM boards or Motherboard (Main Memory)-DRAM is actually FPM, EDO RAM, or SDRAM on a chip.
SRAM	Static RAM, used as External (L2) CACHE. L2 SRAM is on chips. L2 On-board cache is in the CPU chip. It comes in 3 basic types – Async SRAM, Sync SRAM, and PB SRAM (Pipelined Burst RAM - the fastest).
FPM	Fast page mode DRAM-Some modern computers today use Fast Page Mode DRAM. The difference between FPM DRAM and regular DRAM is in the way the memory is accessed by the controller. When data needed is in the same page or row that the previous data was found, The controller only has to indicate the next column location to access the data. By not having to generate a complete address the memory is accessed a little faster.
EDO RAM	Extended Data Out RAM Memory used on Pentium or later type motherboards. EDO RAM is not designed for 486 or earlier motherboards. EDO RAM is on 72-pin SIMMs. EDO RAM comes in plain EDO and Burst EDO (BEDO RAM) versions. EDO and BEDO RAM are ok in systems with bus speeds up to 66MHz.
SDRAM	Synchronous DRAM (Pentium w/MMX has SDRAM as main memory). SDRAM memory is on 168-pin DIMM chips. SDRAM comes in several types with speeds from 10, 15, 20, and 25 nanoseconds.
FLASH MEMORY	Normally, it is memory on a card. The size of a PCMCIA card.
Cache Memory	Normally, it is memory on a card. The size of a PCMCIA card.

So this is the categorization of the memory. Now we will study each and every memory one by one.

4.3.2 Random Access Memory (RAM)

The primary storage is referred to as random access memory (RAM) because it is possible to randomly select and use any location of the memory directly store and retrieve data. It takes

same time to any address of the memory as the first address. It is also called read/write memory because information can be read from a RAM chip and can also be written into it. The storage of data and an instruction inside the primary storage is temporary. It disappears from RAM as soon as the power to the computer is switched off. The memories, which loose their content on failure of power supply, are known as volatile memories .So now we can say that RAM is volatile memory. The simple figure of RAM is shown in Fig 2 and another view of RAM is shown in figure 3.

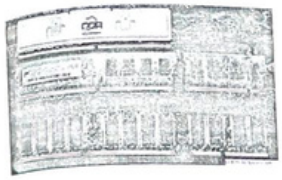


Fig. 4 : Simple view of RAM

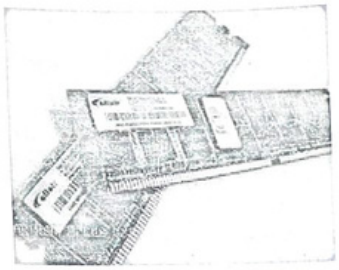


Fig. 5 : Constructive view of RAM

So this is the formal description of the Random access memory (RAM). There are various types of the RAM. These are as follows:

- Dynamic RAM (DRAM)
- Static RAM (SRAM)
- Synchronous DRAM (SDRAM)
- Fast page mode (FPM)
- Error correcting code (ECC)
- Extended data output (EDO)
- Burst extended data output (BEDO)

So these are all the types of the RAM. Now we can explain one by one as in following manner:

Static RAM (SRAM) :

Static RAM is also a storage device. Static RAM does not need to be periodically refreshed. Because SRAM uses bistable latching circuitry to store each bit. SRAM exhibits data remanence but is still volatile in the conventional sense that data is eventually lost when the memory is not powered. Random access means that locations in the memory can be written to or read from in any order, regardless of the memory location that was last accessed.

Each bit in an SRAM is stored on four transistors that form two cross-coupled inverters. This

storage cell has two stable states which are used to denote 0 and 1. Two additional access transistors serve to control the access to a storage cell during read and write operations. A typical SRAM uses six MOSFETs to store each memory bit. In addition to such 6T SRAM, other kinds of SRAM chips use 8T, 10T, or more transistors per bit. This is sometimes used to implement more than one (read and/or write) port, which may be useful in certain types of video memory implemented with multi ported SRAM circuitry.

There are many operations of the SRAM some of these are as follows:

- Standby operation
- Reading operation
- Writing operation

So SRAM cell has three different states it can be in: standby where the circuit is idle, reading when the data has been requested and writing when updating the contents.

Types of SRAM

There are many types of the SRAM. Some of these are as follows:

- Non Volatile SRAM
- Asynchronous SRAM
- By transistor type
- By feature

By function

Now we can explain one by one as in following manner:

Non Volatile SRAM : Non-volatile SRAM has standard SRAM functionality, but they save the data when the power supply is lost, ensuring preservation of critical information. Non volatile SRAM are used in a wide range of situations -- networking, aerospace, and medical, among many others -- where the preservation of data is critical and where batteries are impractical.

Asynchronous SRAM : Asynchronous SRAM are available from 4 Kb to 32 Mb. The fast access time of asynchronous SRAM makes it appropriate for networking equipment. They are used in various applications like switches and routers, IP-Phones, IC-Testers, DSLAM Cards, to Automotive Electronics.

By transistor type

- Bipolar junction transistor(BJT) - very fast but consumes a lot of power
- MOSFET (used in CMOS) - low power and very common today

By function

- **Asynchronous** - independent of clock frequency; data in and data out are controlled by address transition
- **Synchronous** - all timings are initiated by the clock edge(s). Address, data in and other control signals are associated with the clock signals

By feature

- **ZBT (zero bus turnaround)** - the turnaround is the number of clock cycles it takes to change access to the SRAM from write to read and vice versa. The turnaround for ZBT SRAMs or the latency between read and writes cycle is zero.
- **Sync Burst** (sync Burst SRAM or synchronous-burst SRAM) - features synchronous burst write access to the SRAM to increase write operation to the SRAM.
- **DDR SRAM** - Synchronous, single read/write port, double data rate IO
- **Quad Data Rate SRAM** - Synchronous, separate read & write ports, double data rate IO

There are many characteristics and features of the SRAM. SRAM is more expensive, but faster and significantly less power hungry than DRAM. It is therefore used where either bandwidth or low power, or both, are principal considerations. SRAM is also easier to control (interface to) and generally more truly random access than modern types of DRAM. Due to a more complex internal structure, SRAM (Static RAM) is less dense than DRAM (Dynamic RAM) and is therefore not used for high-capacity, low-cost applications such as the main memory in personal computers.

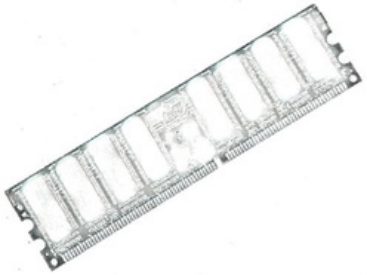


Fig. 6 : Static RAM (SRAM)

There are many advantages of the Static RAM.SRAM is also used in personal computers, workstations, routers and peripheral equipment: internal CPU caches and external burst mode SRAM caches, hard disk buffers, router buffers, etc. LCD screens and printers also normally employ static RAM to hold the image displayed (or to be printed). Small SRAM buffers are also found in CDROM and CDRW drives; usually 256 kB or more are used to buffer track data, which is transferred in blocks instead of as single values. The same applies to cable modems and similar equipment connected to computers

DYNAMIC RAM (DRAM) :

Dynamic random access memory (DRAM) is a type of random access memory that stores each bit of data in a separate capacitor within an integrated circuit. Since real capacitors leak charge, the information eventually fades unless the capacitor charge is refreshed periodically.

Because of this refresh requirement, it is a dynamic memory as opposed to SRAM and other static memory.

The first DRAM with multiplexed row and column address lines was the Mostek MK4096 (4096x1) designed by Robert Proebsting and discovered in 1973. This addressing scheme, a radical advance, allowed it to fit into packages with fewer pins, a cost advantage that would grow with every jump in memory size. The MK4096 also proved to be very robust design in customer applications. At the 16K density the cost advantage increased, and the Mostek MK4116 16K DRAM achieved greater than 70% worldwide DRAM market share.

Principle of operation of DRAM read, for simple 4 by 4 arrays And operation of DRAM writes, for simple 4 by 4 arrays. The advantage of DRAM is its structural simplicity: only one transistor and a capacitor are required per bit, compared to four transistors in SRAM. This allows DRAM to reach very high density. Unlike Flash memory, it is volatile memory (cf. non-volatile memory), since it loses its data when the power supply is removed.

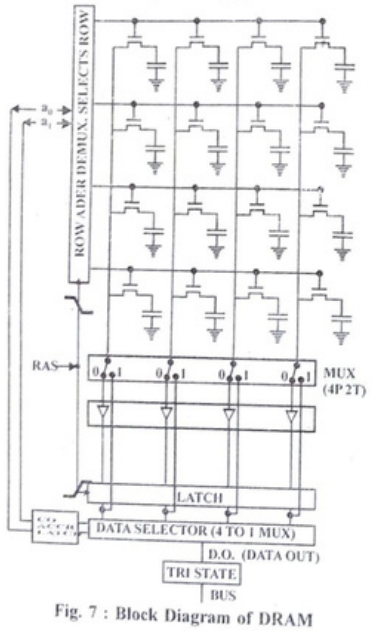


Fig. 7 : Block Diagram of DRAM

A dynamic RAM chip contains:

- A transistor (function like a on/off switch)
- A capacitor which used to storage an electric charge

DRAM is usually arranged in a square array of one capacitor and transistor per cell. The illustrations to show a simple example with only 4 by 4 cells (modern DRAM can be thousands of cells in length/width).The long lines connecting each row are known as word lines. Each column is actually composed of two bit lines, each one connected to every other storage cell in the column. (The illustration to the right does not include this important detail.) They are generally known as the + and ? bit lines. A sense amplifier is essentially a pair of cross-connected inverters between the bit lines. That is, the first inverter is connected from the + bit line to the ? bit line, and the second is connected from the ? bit to the + bit line. This is an example of positive feedback, and the arrangement is only stable with one bit line high and one bit line low.

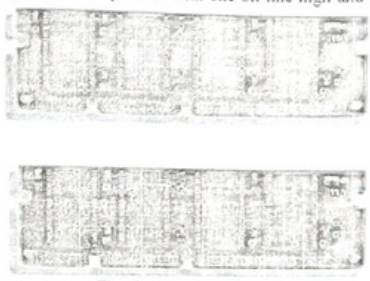


Fig. 8 : Dynamic RAM

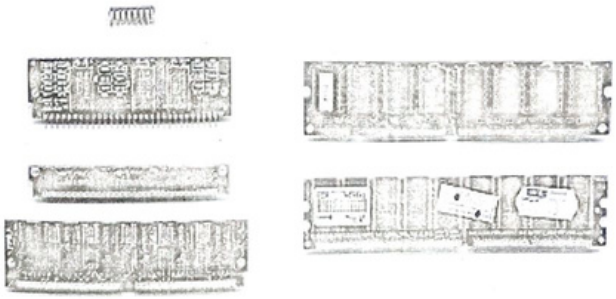


Fig. 9 : Packages of DRAM

Dynamic random access memory is produced as integrated circuits (ICs) bonded and mounted into plastic packages with metal pins for connection to control signals and buses. This is also known as volatile storage. Dynamic RAM is typically used on the primary storage section. For economic reasons, the large (main) memories found in personal computers, workstations, and non-handheld game-consoles (such as Playstation and Xbox) normally consist of dynamic RAM (DRAM). Other parts of the computer, such as cache memories and data buffers in hard disks, normally use static RAM (SRAM).

Synchronous DRAM (SDRAM)

Synchronous dynamic random access memory (SDRAM) is dynamic random access memory (DRAM) that has a synchronous interface. Traditionally, dynamic random access memory (DRAM) has an asynchronous interface which means that it responds as quickly as possible to changes in control inputs. SDRAM has a synchronous interface, meaning that it waits for a clock signal before responding to control inputs and is therefore synchronized with the computer's system bus. The clock is used to drive an internal finite state machine (FSM) that pipelines incoming instructions. This allows the chip to have a more complex pattern of operation than asynchronous DRAM which does not have a synchronized interface.

Pipelining means that the chip can accept a new instruction before it has finished processing the previous one. In a pipelined write, the write command can be immediately followed by another instruction without waiting for the data to be written to the memory array. In a pipelined read, the requested data appears after a fixed number of clock pulses after the read instruction, cycles during which additional instructions can be sent. (This delay is called the latency and is an important parameter to consider when purchasing SDRAM for a computer.)

Today, virtually all SDRAM is manufactured in compliance with standards established by JEDEC, an electronics industry association that adopts open standards to facilitate interoperability of electronic components. JEDEC formally adopted its first SDRAM standard in 1993 and subsequently adopted other SDRAM standards.

There are many type of control signals of SDRAM like as CKE (Clock Enable), CS (Chip Select), DQM (Data Mask), RAS (Row Address Strobe), CAS (Column Address Strobe) ,WE (Write enable).

There are many type of SDRAM exists like as:

- DDR
- SDRAM
- DDR2 SDRAM
- DDR3 SDRAM
- DDR4 SDRAM

SDRAM is also available in registered varieties, for systems that require greater scalability such as servers and workstations. Today, the world's largest manufacturers of SDRAM include: Samsung Electronics, Micron Technology, Qimonda (formerly Infineon Technologies) and Hynix.

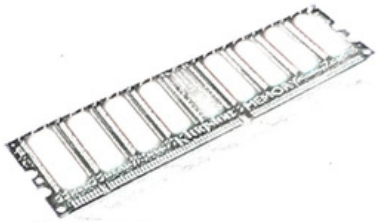


Fig. 10 : Static Dynamic RAM (SDRAM)

Fast Page Mode (FPM)

Fast page mode DRAM is also called FPM DRAM, Page mode DRAM, Fast page mode memory, or Page mode memory. In page mode, a row of the DRAM can be kept "open" by holding /RAS (Row Address Strobe) low while performing multiple reads or writes with separate pulses of /CAS (Column Address Strobe). So that successive reads or writes within the row do not suffer the delay of precharge and accessing the row. This increases the performance of the system when reading or writing bursts of data.

Fig. 11 : FPM RAM

Static column is a variant of page mode in which the column address does not require to be strobe in, but rather, the address inputs may be changed with /CAS held low, and the data output will be updated accordingly a few nanoseconds later.

Nibble mode is another variant of page monde in which four sequential locations within the row can be accessed with four consecutive pulses of /CAS. The difference from normal page mode is that the address inputs are not used for the second through fourth /CAS edges; they are generated internally starting with the address supplied for the first /CAS edge.

Simple fig. of the FPM RAM is shown in figure 9.

ERROR CORRECTING CODE (ECC)

ECC RAM is a special error correcting code RAM type. This type of RAM is specially used for servers.There is some concern that as DRAM density increases further, and thus the components on DRAM chips get smaller, while at the same time operating voltages continue to fall, DRAM chips will be affected by such radiation more frequently - since lower energy particles will be able to change a memory cell's state. On the other hand, smaller cells make smaller targets, and moves to technologies such as SOI may make individual cells less susceptible and so counteract, or even reverse this trend.

This problem can be mitigated by using DRAM (Dynamic RAM) modules that include extra memory bits and memory controllers that exploit these bits. These extra bits are used to record parity or to use an error-correcting code (ECC). Parity allows the detection of a single-bit error (actually, any odd number of wrong bits). The common error correcting code, Hamming code, allows a single-bit error to be corrected and (in the usual configuration, with an extra parity bit) double-bit errors to be detected.

Error detection and correction in computer systems seems to go in and out of fashion. Wider memory buses make parity and especially ECC more affordable. Many current microprocessor memory controllers, including almost all AMD 64-bit offerings, support ECC, but many motherboards and in particular those using low-end chipsets do not.An ECC-capable memory controller as used in many modern PCs can typically detect and correct errors of a single bit per 64-bit "word" (the unit of bus transfer), and detect errors of two bits per 64-bit word. Some systems also 'scrub' the errors, by writing the corrected version back to memory. The BIOS in some computers, and operating systems such as Linux, allow counting of detected and corrected memory errors, in part to help identify failing memory modules before the problem becomes catastrophic.

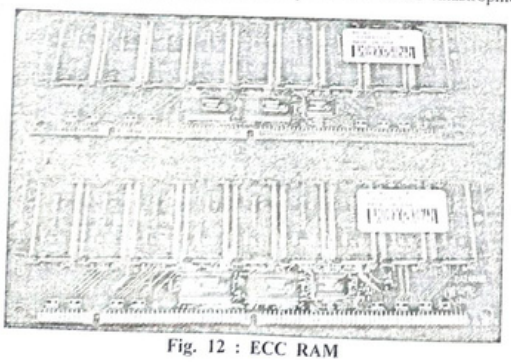


Fig. 12 : ECC RAM

Error detection and correction depends on an expectation of the kinds of errors that occur. Implicitly, we have assumed that the failure of each bit in a word of memory is independent and hence that two simultaneous errors are improbable. This used to be the case when memory chips were one bit wide (typical in the first half of the 1980s). Now many bits are in the same chipIn most computers used for serious scientific or financial computing and as servers ECC is the rule rather than the exception, as can be seen by examining manufacturers' specifications.

EXTENDED DATA OUTPUT (EDO)

The most common type of asynchronous DRAM used is called extended data out or EDO memory; sometimes it is also called hyper page mode DRAM. EDO DRAM is similar to Fast Page Mode DRAM with the additional feature that a new access cycle can be started while keeping the data output of the previous cycle active. It is slightly faster than FPM memory due to another evolutionary tweak in how the memory access works. In simplified terms, EDO memory has had its timing circuits modified so one access to the memory can begin before the last one has finished. It is therefore slightly faster than FPM memory, giving a performance boost of around 3-5% over FPM in most systems. EDO memory has been hyped up a great deal, but in real world performance it offers a minimal speed increase over FPM memory.

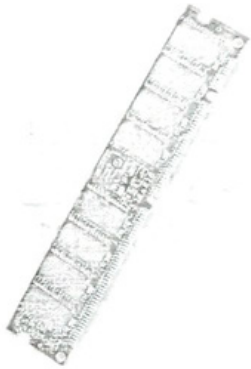


Fig. 13 : EDO RAM

EDO memory costs the same amount to manufacture as FPM, and due to its prominence in the market now is actually usually cheaper than FPM despite being newer and faster. It was originally more expensive but the reduced demand for slower FPM memory now makes FPM harder to find in most cases. Until recently, EDO was the standard for fifth- and sixth-generation systems. It still is found in later model Pentium-class PCs, but SDRAM has now replaced it as the technology of choice for sixth-generation systems. EDO memory is still not usually suitable for high-speed (75 MHz and higher) memory buses, since it is really not that different than FPM overall. EDO typically allows burst system timings as fast as 5-2-2-2 at 66 MHz, when using an optimized chipset. It will run on faster buses but the memory timing may need to be reduced.Single-cycle EDO DRAM became very popular on video cards towards the end of the 1990s. It was very low cost, yet nearly as efficient for performance as the far more costly video RAM (VRAM).

Extended Data out DRAM comes in 70ns, 60ns and 50ns speeds. 60ns is the slowest that should be used in a 66MHz bus speed system (i.e. Pentium 100MHz and above) and the Triton HX and VX chipsets can also take advantage of the 50ns version.EDO memory requires support from the system chipset. Invented in 1994, most new Pentium systems, as well as some of the latest PCI-based 486 motherboards will support EDO. Older systems will not work properly with EDO; some are "EDO tolerant", meaning that they will work with EDO but will run it at as if it were FPM memory (slower). Others will not work at all with EDO memory.

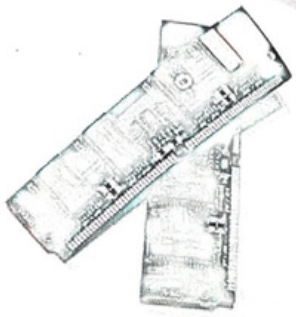


Fig. 14 : A pair of 32 MB EDO DRAM modules

BURST EXTENDED DATA OUTPUT (BEDO)

Burst EDO DRAM is an evolutionary improvement in EDO DRAM that contains a pipeline stage and a 2-bit burst counter. BEDO also added a pipelined stage allowing page-access cycle to be divided into two components. During a memory-read operation, the first component accessed the data from the memory array to the output stage. The second component drove the data bus from this latch at the appropriate logic level. Since the data is already in the output buffer, quicker access time is achieved (up to 50% for large blocks of data) than with traditional EDO.With the conventional DRAMs such as FPM and EDO, the initiator accesses DRAM through a memory controller. The controller must wait for the data to become ready before sending it to the initiator. BEDO eliminates the wait-states thus improving system performance by up to 100% over FPM DRAM and up to 50% over standard EDO DRAM, achieving system timings of 5-1-1-1 when used with a supporting chipset.

Despite the fact that BEDO arguably provides more improvement over EDO than EDO does over FPM the standard has lacked chipset support and has consequently never really caught on, losing out to Synchronous DRAM. Although BEDO DRAM showed additional optimization over EDO, by the time it was available the market had made a significant investment towards synchronous DRAM, or SDRAM. Even though BEDO RAM was superior to SDRAM in some ways, the latter technology gained significant traction and quickly displaced BEDO.

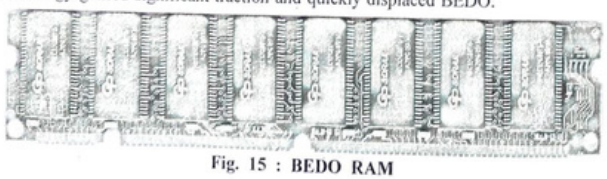


Fig. 15 : BEDO RAM

So this was the complete discussion of the primary memory or random access memory (RAM). Now we will discuss the read only memory (ROM).

4.3.2 READ ONLY MEMORY (ROM)

Read-only memory (ROM) is a class of storage media used in computers and other electronic devices. Because data stored in ROM cannot be modified (at least not very quickly or easily), it is mainly used to distribute firmware (software that is very closely tied to specific hardware, and unlikely to require frequent updates).

In its strictest sense, ROM refers only to mask ROM (the oldest type of solid state ROM), which is fabricated with the required data permanently stored in it, and thus can never be modified. However, more modern types such as EPROM and flash EEPROM can be erased and re-programmed multiple times. They are still described as "read-only memory"(ROM) because the reprogramming process is generally infrequent, comparatively slow, and often does not permit random access writes to individual memory locations.

The simplest type of solid state ROM is as old as semiconductor technology itself. Combinational logic gates can be joined manually to map n-bit address input onto arbitrary values of m-bit data output. With the invention of the integrated circuit (IC's) came mask ROM. Mask ROM consists of a grid of word lines (the address input) and bit lines (the data output), selectively joined together with transistor switches, and can represent an arbitrary look-up table with a regular physical layout and predictable propagation delay.

Every stored-program computer requires some form of ((non-volatile) or erasable) storage to store the initial program that runs when the computer is powered on or otherwise begins execution (a process known as bootstrapping, often abbreviated to "booting" or "booting up"). Likewise, every non-trivial computer requires some form of mutable memory to record changes in its state as it executes.

Forms of read-only memory were employed as non-volatile storage for programs in most early stored-program computers, such as ENIAC after 1948 (until then it was not a stored-program computer as every program had to be manually wired into the machine, which could take days to weeks). Read-only memory was simpler to implement since it required only a mechanism to read stored values, and not to change them in-place, and thus could be implemented with very crude electromechanical devices (see historical examples below). With the advent of integrated circuits (IC's) in the 1960s, both ROM and its mutable counterpart static RAM (SRAM) were implemented as arrays of transistors in silicon chips; however, a ROM memory cell could be implemented using fewer transistors than an SRAM memory cell, since the latter requires a latch to retain its contents, while a ROM cell might consist of the absence (logical 0) or presence (logical 1) of a single transistor connecting a bit line to a word line Consequently, ROM could be implemented at a lower cost-per-bit than RAM for many years.

There are many example of the ROM. Some of these are as follows:

- Diode matrix ROM, used in small amounts in many computers in the 1960s as well as electronic desk calculators and keyboard encoders for terminals..

- Resistor, capacitor, or transformer matrix ROM, used in many computers until the 1970s.
- Core rope, a form of transformer matrix ROM technology used where size and/or weight were critical. This was used in NASA/MIT's Apollo Spacecraft Computers, DEC's PDP-8 computers, and other places. This type of ROM was programmed by hand by weaving "word line wires" inside or outside of ferrite transformer cores.
- Various mechanical devices used in early computing equipment. A machined metal plate served as ROM in the dot matrix printers .
- In the PC's the DVD ROM also is example of this type of memory.

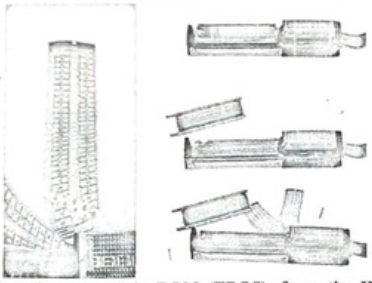


Fig. 16 : Transformer matrix ROM (TROS), from the IBM System

Simple diagram of the DVD ROM is shown below:

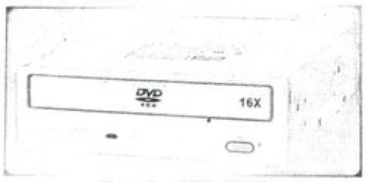


Fig. 17 : DVD ROM

While both ROM and RAM are storage devices and can be access randomly, but they differ in that data can be written into RAM while ROM does not permit the user to write into it. There are various type of ROM.

Types of ROM

2.6.1 Programmable read only memory (PROM)

2.6.2 Erasable Programmable read only memory(EPROM)

2.6.3 Electrically Erasable Programmable read only memory(EEPROM)

So these are the three types of the memory PROM, EPROM, EEPROM. Now we will study one by one.

Programmable read only memory (PROM)

A programmable read-only memory (PROM) or field programmable read-only memory (FPROM) is a form of digital memory where the setting of each bit is locked by a fuse or antifuse. Such PROMs are used to store programs permanently. The key difference from a strict ROM is that the programming is applied after the device is constructed. They are frequently seen in video game consoles, or such products as electronic dictionaries, where PROMs for different languages can be substituted. PROM was invented in 1956 by Wen Tsing Chow, working for the Arma Division of the American Bosch Arma Corporation in Garden City, New York.

There are many advantage of the PROM. Such as:

- It is very reliable to use
- It stores data permanently
- Very less price
- Built using integrated circuits, rather than discrete components
- Fast reading of neno seconds.

A PROM is shown below in figure 16.

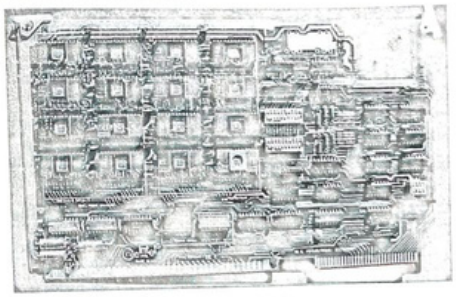


Fig. 18 : PROM

Erasable Programmable read only memory(EPROM)

An EPROM, or Erasable Programmable Read-Only Memory, is a type of memory chip that retains its data when its power supply is switched off. In other words, it is non-volatile. It is an array of floating-gate transistors individually programmed by an electronic device that supplies higher voltages than those normally used in digital circuits. Once programmed, an EPROM can be erased only by exposing it to strong ultraviolet light. That UV light usually has a wavelength of 253.7nm (for optimum erasure time) and belongs to the UVC range of UV light.

A programmed EPROM retains its data for about ten to twenty years and can be read an

unlimited number of times. The erasing window must be kept covered with a foil label to prevent accidental erasure by sunlight. Old PC BIOS chips were often EPROMs, and the erasing window was often covered with a label containing the BIOS publisher's name, the BIOS revision, and a copyright notice.

The EPROM was invented by Israeli engineer Dov Frohman in 1971.EPROMs come in several sizes both in physical packaging as well and storage capacity. While parts of the same type number from different manufacturers are compatible as long as they're only being read, there are subtle differences in the programming process. Fig. of EPROM is shown in figure 19.

Fig. 19. EPROM

Electrically Erasable Programmable read only memory(EEPROM)

EEPROM (also written E2PROM and pronounced "e-e-prom," "double-e prom") stands for Electrically Erasable Programmable Read-Only Memory and is a type of non-volatile memory used in computers and other electronic devices to store small amounts of data that must be saved when power is removed, e.g. is calibration tables or device configuration. When larger amounts of static data are to be stored (such as in USB flash drives) a specific type of EEPROM such as flash memory is more economical than traditional EEPROM devices.

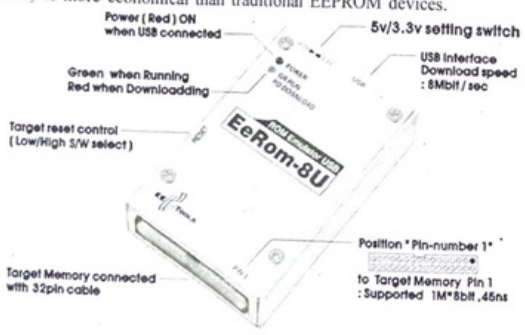


Fig. 20. EEPROM

The difference between EPROM and EEPROM lies in the way that the memory programs and erases. EEPROM can be programmed and erased electrically using field electron emission

(more commonly known in the industry as "Fowler-Nordheim tunneling").

EPROMs can't be erased electrically, and are programmed via hot carrier injection onto the floating gate. Erase is via an ultraviolet light source, although in practice many EPROMs are encapsulated in plastic that is opaque to UV light, and are "one-time programmable".

Simple fig. of EEPROM is shown in figure 18.

So this is the discussion of the read only memory (ROM). Now we will study about the virtual and cache memory.

Flash Memory

Flash memory is a non-volatile memory that can be electrically erased and reprogrammed. It is a technology that is primarily used in memory cards and USB flash drives for general storage and transfer of data between computers and other digital products. It is a specific type of EEPROM (Electrically Erasable Programmable Read-Only Memory) that is erased and programmed

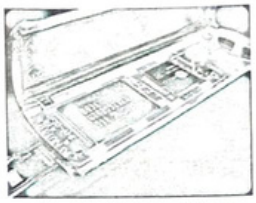


Fig. 21. USB Flesh Drive

in large blocks; in early flash the entire chip had to be erased at once. Flash memory costs far less than byte-programmable EEPROM and therefore has become the dominant technology wherever a significant amount of non-volatile, solid state storage is needed. Example applications include PDAs (personal digital assistants), laptop computers, digital audio players, digital cameras and mobile phones. It has also gained popularity in the game console market, where it is often used instead of EEPROMs or battery-powered SRAM for game save data.

Since flash memory is non-volatile, no power is needed to maintain the information stored in the chip. In addition, flash memory offers fast read access times (although not as fast as volatile DRAM memory used for main memory in PCs) and better kinetic shock resistance than hard disks. These characteristics explain the popularity of flash memory in portable devices. Another feature of flash memory is that when packaged in a "memory card," it is enormously durable, being able to withstand intense pressure, extremes of temperature, and even immersion in water. Although technically a type of EEPROM, the term "EEPROM" is generally used to refer specifically to non-flash EEPROM which is

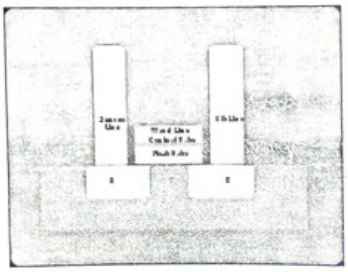


Fig. 22. A flash memory cell

erasable in small blocks, typically bytes. Because erase cycles are slow, the large block sizes used in flash memory erasing give it a significant speed advantage over old-style EEPROM when writing large amounts of data.

Flash memory stores information in an array of memory cells made from floating-gate transistors. In traditional single-level cell (SLC) devices, each cell stores only one bit of information. Some newer flash memory, known as multi-level cell (MLC) devices, can store more than one bit per cell by choosing between multiple levels of electrical charge to apply to the floating gates of its cells.

4.3.3 Cache Memory

A cache, in computer terms, is a place to store information that's faster than the place where the information is usually stored. Cache memory is fast memory that is used to hold the most recently accessed data in slower main memory. The idea is that frequently accessed data will stay in cache, which allows the CPU to access it more quickly, which means it doesn't have to wait for the data to arrive. In reference to our processor, the Cache Memory is the Processor's internal quick-hand storage that it uses for things that it's currently processing at that given time. In other words, a cache is a temporary storage area where frequently accessed data can be stored for rapid access. Once the data is stored in the cache, it can be used in the future by accessing the cached copy rather than re-fetching or recomputing the original data.

A cache is a block of memory for temporary storage of data likely to be used again. The CPU and hard drive frequently use a cache, as do web browsers and web servers. A cache is made up of a pool of entries. Each entry has a tag, which specifies the identity of the datum in the backing store of which the entry is a copy. When the cache client (a CPU, web browser, and operating system) wishes to access a datum presumably in the backing store, it first checks the cache. If an entry can be found with a tag matching that of the desired datum, the datum in the entry is used instead. This situation is known as a cache hit. So, for example, a web browser program might check its local cache on disk to see if it has a local copy of the contents of a web page at a particular URL. In this example, the URL is the tag, and the contents of the web page are the datum. The percentage of accesses that result in cache hits is known as the hit rate or hit ratio of the cache. The alternative situation, when the cache is consulted and found not to contain a datum with the desired tag, is known as a cache miss. The previously uncached datum fetched from the backing store during miss handling is usually copied into the cache, ready for the next access.

4.3.4 Virtual Memory

Virtual memory is a computer system technique which gives an application program the impression that it has contiguous working memory (an address space), while in fact it may be physically fragmented and may even overflow on to disk storage. Systems that use this technique make programming of large applications easier and use real physical memory (e.g. RAM) more efficiently than those without virtual memory. Virtual memory differs significantly from memory virtualization in that virtual memory allows resources to be virtualized as memory for a specific

system, as opposed to a large pool of memory being virtualized as smaller pools for many different systems.

Virtual memory was developed in approximately 1959-1962, at the University of Manchester for the Atlas Computer, completed in 1962.Note that "virtual memory" is more than just "using disk space to extend physical memory size" - that is merely the extension of the memory hierarchy to include hard disk. Extending memory to disk is a normal consequence of using virtual memory techniques, but could be done by other means such as overlays or swapping programs and their data completely out to disk while they are inactive. The definition of "virtual memory" is based on redefining the address space with a contiguous virtual memory addresses to "trick" programs into thinking they are using large blocks of contiguous addresses.

All modern general-purpose computer operating systems use virtual memory techniques for ordinary applications, such as word processors, spreadsheets, multimedia players, accounting, etc. Older operating systems, such as DOS and Microsoft Windows of the 1980s, or those for the mainframes of the 1960s, generally had no virtual memory functionality - notable exceptions being the Atlas, B5000 and Apple Computer's Lisa.

EXERCISES

Very Short Answer Type Questions (2 Marks each)

1. Address bus is known as **(Raj. B.C.A. 2012)**
2. Flash memory is ..... **(Raj. B.C.A. 2012)**
3. Main memory is made up of **(Raj. B.C.A. 2012)**
4. Cache memory is ..... **(Raj. B.C.A.2011)**
5. EPROM stands for .....

Short Answer Type Questions (4 Marks each)

1. What do you mean by addressing modes ? **(Raj. B.C.A. 2013)**
2. Short notes on
   EPROM, Virtual Memory, Registers. **(Raj. B.C.A. 2013)**
3. Explain cache memory and SRAM/DRAM.
4. Explain indirect and indexed addressing mode in detail. **(Raj. B.C.A. 2012)**
5. Explain in brief
   (a) Address bus
   (b) Main memory
   (c) Cache memory

Long Answer Type Questions (12 Marks each)

1. What do you mean by addressing modes ? Describe the following addressing modes ?

(i) Immediate mode
(ii) Relative address mode
(iii) Register direct and indirect mode

2. Write short notes on : **(Raj. B.C.A. 2013)**
   (i) Programmable Read only memory
   (ii) S-RAM and D-RAM
   (iii) Registers.
3. Explain virtual memory and cache memory. **(Raj. B.C.A. 2006)**
4. Explain all addressing modes. What is memory buffer registes.
5. What is virtual memory ? Write advantages of virtual memory ?
6. What is cache memory ? Why is it called high speed memory ? Explain cache-hit and cache mis ? **(Raj. B.C.A. 2010)**
7. What do you mean by addressing techniques ? Explain indirect and indexed addressing technique in detail. **(Raj. B.C.A. 2008)**
8. List and describe different registers for the basic computer. Also describe the common bus system for a basic computer. **(Raj. B.C.A. 2007)**
9. Short notes on :
   (a) Memory Buffer Registers
   (b) S-RAM and D-RAM
   (c) RAM and Main memory.

■■■