7

Memory

The computer can be used to solve a mathematical problem, to type a letter, to draw some figure etc., for all these applications you will have to execute some program or set of instructions.

Before executing a program the program is stored into the computers memory and from the memory, computer takes these instructions one by one and execute them.

The data required to operate on or the result after the operation are also kept in the computer's memory– before transferring them onto some other permanent storage device or to an output device.

The computer memory is divided into two types based on the whether the memory is inside the computer or it is an external storage device.

- Primary Memory
- Secondary Memory

Primary Memory

The main memory or the memory on the motherboard of the computer is called the **primary memory**.

This is also called the **on-line memory** because it is always directly available to the processor.

This primary memory can be further divided into following types

- RAM
- ROM

RAM is a type of memory which is used by the computer to store temporary values, programs etc.

- RAM is also a **Volatile Memory**.
- The **volatile memory** is a memory which loses its content when the power supply to this memory is switched off.

Any information in the volatile memory or the RAM is transferred to some permanent storage device such as floppy disk or hard disk before switching off the computer.

- **Nonvolatile memory** is a permanent memory, which does not lose its contents even when the power supply is switched off.
- A Nonvolatile memory such as ROM is used inside the computer to keep permanent information such as the "boot program" which is required each time the machine is switched on.

Secondary Memory

Secondary memory is not a memory in conventional terms, it is actually the storage media used to store the program or the data.

Floppy disk or hard disk are the example of secondary memory. This is also called **off-line memory** because it is not directly accessible to the processor.

Because of its size, it is also called **mass storage device**.

Bits & Byte

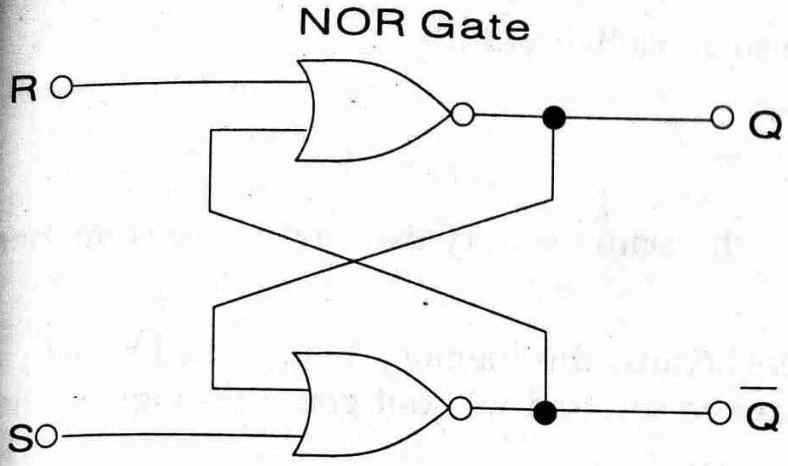
Bit or the **Binary Digit** i.e. number 0 and number 1 are the most basic information that one can store inside the computer.

A single bit can be stored inside the computer as a charged capacitor, by using any bistable multivibrator such as flip-flops, by using the transistor as a switch, or by using a relay etc.

A S-R Flip-Flop is made using the NOR gates, this circuit can be made using about 5 to 6 transistor and is used to store single bit of information.

In an S-R Flip-Flop,

- When the input S and R both are 0, the flip-flop will remain in the last set stage i.e. it will remember the last set state.
- When the input S is 1 and R is 0, the flip-flop will get set or will store a binary number 1
- When the input S is 0 and R is 1, the flip-flop will get reset or will store a



R	S	Q	$\overline{Q}$	STATUS
0	0	Q	$\overline{Q}$	MEMORY
0	1	0	1	RESET
1	0	1	0	SET
1	1			ILLEGAL

Fig. 7-1 : S-R Flip-Flop using NOR gates, used to store one bit.

- When the input S and R both are 1, it will be treated as an invalid input by the flip-flop.

A large memory can be constructed by arranging this type of many flip-flops in an array.

As a single bit alone does not provide much information, a combination of these bits are used to convey different information.

When two bits are used they can be combined in four different forms 00, 01, 10, and 11 and each combination can convey a different information.

A combination of 4 bit is called a "**nibble**" and is used to store information inside the calculators, as the 4 bit can have $2^4 = 16$ different combinations, it is enough to convey 10 different digits (0 to 9) and some additional information such as decimal point etc.

The computer can manipulate characters as well as numbers, so a combination of 8 bits (called a "**byte**") is used inside the computer to store a character.

A combination of 8 bits can convey $2^8 = 256$ different combinations, which is enough for all the alphabet, lower case, upper case, numbers, and some special text and graphic symbols used by the computer.

Different combination of 0s and 1s of these 8 bits is used to store different information inside the computer's memory. for example

- 01000001 is used to store character "A"
- 00110001 is used to store the number "1"

1 byte can store 1 character, a character can be an alphabet, number or special symbol character such as @, #, $, * etc.

- Combination of 1000 byte is called a Kilo byte (1KB)
- Combination of 1000 KB is called a Mega Byte (1 MB)
- Combination of 1000 MB is called a Giga Byte (1 GB)
- Combination of 1000 GB is called a Tera Byte (1 TB)

- Combination of 1000 TB is called a Peta Byte (1 PB)
- Combination of 1000 PB is called a Exa Byte (1 EB).

RAM

RAM or **Random Access Memory** is the main memory used inside the computer to store program, data and results.

The term Random access is used here because this memory is organized in such a way that any part of the memory can be accessed without going through all the previous parts.

- It is a temporary memory that computer use as a work area.
- A better term to describe the RAM would be **Read/Write Memory**. One can read the information stored inside the RAM as well as write or store information into it.

Two of the common RAM are

- DRAM (Dynamic RAM)
- SRAM (Static RAM)

Dynamic RAM (DRAM)

A **dynamic RAM** (DRAM) is cost wise cheaper than the **static RAM** (SRAM), and it is the main reason of using the DRAM as the main memory.

Since the DRAM utilizes capacitors to store information, the problem with the DRAM is once some data is stored into it, it can't retain the data for a long time unless the data is refreshed after some time.

Refreshing of a DRAM memory is done by rewriting the content of the memory every few millisecond. Unless the DRAM is refreshed every few millisecond it loses its content.

Static RAM (SRAM)

Unlike the DRAM, information stored inside SRAM memory remains as it is as long as the power supply is provided to the SRAM chip.

One need not refresh the information stored into the SRAM.

Because the SRAM does not require any refresh circuits, the circuitry required to interface SRAM is very simple compared to the interface circuitry for the DRAM chips.

SRAM basically use **flip-flops** to store the information.

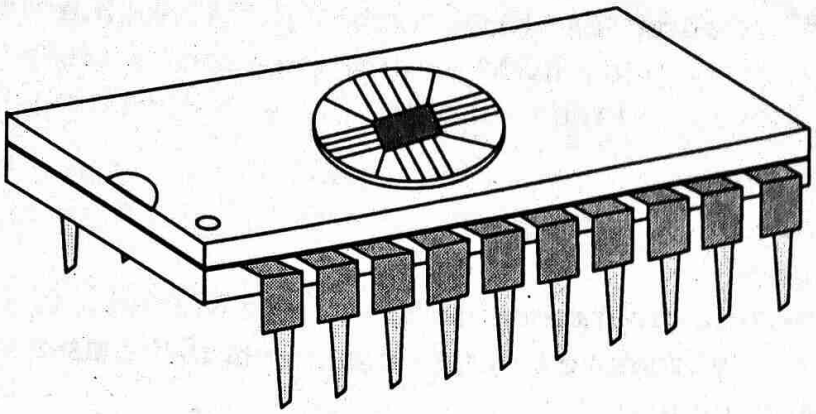


Fig. 7-2 : EPROM Memory Chip.

ROM

ROM or **Read Only Memory** is as its name suggest is a memory that can be read only, one cannot write any information into it.

One big plus point about the ROM is, it does not lose its content when the power supply is cut off, this is a **nonvolatile** type memory.

Like a RAM, a ROM is also random access memory i.e. one can directly access any part of the ROM without serially going through the complete ROM.

There are many types of ROM available in the market, some of them are

Mask ROM

A mask ROM is the basic ROM chip. In this type of chips the information is fabricated into the chip at the time of manufacturing itself.

The mask is the master pattern used during the chip manufacturing to make various circuit elements on the chip.

Number of chips to be manufactured should be very high when this type of ROM is made, to justify the high investment required in their making.

PROM

PROM or **Programmable Read Only Memory** is the ROM memory with a small difference.

At the time of manufacturing, this chip is made as a blank ROM chip and later using special **PROM programmers** the information is stored into them.

Initially when the PROM is manufactured it contains row and address connections, i.e. all the locations contain a binary 1.

Later to store any information, the connection between the row and column is broken to store a binary 0, and the connection is made to remain as it is to store a binary number 1.

Once some information is stored into the PROM by "**burning**" the information into it, it becomes equal to a ROM i.e. now the information stored into it cannot be changed or removed and removal of the power to the chip will not clear the data stored into it.

EPROM

EPROM or **Erasable Programmable Read Only Memory**, is a PROM but with an option to be able to remove or erase it contents if the user wants to change the information stored into it.

- The EPROM is easily distinguishable from other chips because of a window in the middle of this chip. The window is covered with a transparent quartz glass.
- Through this window, one can see the internal circuitry of the EPROM chip. This windows is used to erase the content of the EPROM chip.
- To erase the content stored in the EPROM, the EPROM is put under a short wave ultraviolet light source inside an EPROM erasure device.
- Once the content of the EPROM is erased it can be reprogrammed or rewritten using the **EPROM programmer**.
- After the EPROM is programmed the top window should be closed with some opaque label, because the sun rays which contains the ultra violet light can slowly delete/erase the content of the EPROM chip.
- In EPROM one single memory location cannot be erased or changed, to change the content of even a single location the entire EPROM's content should be erased and then the complete EPROM should be rewritten with the new value.

EPROMs are one of the most common ROM used in the computer field.

EEPROM

EEPROM or **Electrically Erasable Programmable Read Only Memory** (pronounced "ee ee prom" or "double e prom") is another type of EPROM.

The difference between an EPROM and EEPROM is in the way its content is erased.

- In an EPROM the content is removed by shining ultraviolet rays through the window directly to the circuits, whereas in the EEPROM the content is removed by using a higher than normal electric voltage.

For example, if the EEPROM is normally used with +5 Volt for the read operation than by using a voltage higher than +5 Volt for example +12 Volt, the content of the EEPROM could be erased.

One big advantage of the EEPROM is its content can be deleted without removing it form the circuit, whereas to erase the content of the EPROM one has to remove it from the circuit and put under the ultraviolet rays for some time.

By providing the erase voltage in the circuit itself the content of the EEPROM can be erased without removing it from the circuit.

Even though the EEPROM does not lose its content when the power supply to it is switched off and it can be erased and programmed without removing it from the circuit, it is not used in place of RAM memory in your computer.

Two main reasons behind this are

- The EEPROM can be erased and programmed for a limited number of times only.
- The second problem with the EEPROM is that to change even one bit of information stored in the EEPROM, the entire content is to be first removed and then everything is to be written back with the new values.

EAROM

EAROM or **Electrically Alterable Read only Memory** is another type of ROM which can be written and read in somewhat the same fashion as the RAM.

- In an EAROM by applying a high voltage to a particular bit of the memory the content of that location can be changed
- To change the content of one location the entire content need not be erased and rewritten as it is done in the EPROM and EEPROM devices.
- Once some information is stored into the EAROM it works just like a normal ROM, it does not lose its contents when the power supply is cut off.
- The only drawback of the EAROM is the write operation is very slow compared to the write speed of RAM memory.

EAROMs are useful in areas when small amount of information is to be stored permanently, which may require changes from time to time and where the battery backup may not be available or possible.

Flash Memory

A new type of EEPROM that can be erased and reprogrammed using the normal operating voltage found inside the PC is called flash memory.

- The flash memory has the same limitation as the EEPROM, it can be erased and programmed only for a fixed number of times and it must be erased and programmed in blocks.
- The first generation of Flash ROM had the entire chip as a single block, so the entire chip had to be erased to reprogram it.
- Newer Flash ROMs have multiple, independently erasable blocks in sizes from 4K to 128K bytes.
- Once a block is erased, each cell will contain a value of zero. Standard write operations can change the cell values from zero to one but cannot change them back to zero.

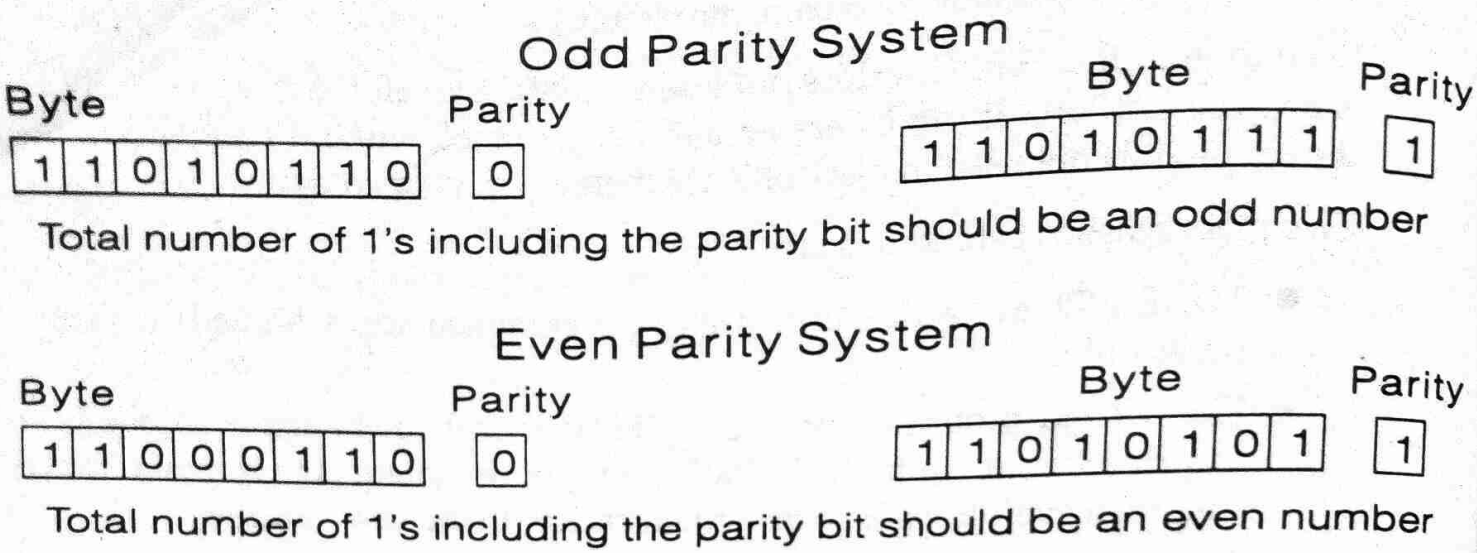


Fig. 7-3 : Odd and Even Parity.

- Once a given cell has been changed to a logical one with a write operation, it will maintain that value until the Flash ROM gets erased once again.
- The value stored in flash memory is retained even if the power to your system or the Flash ROM chip fails.

Some manufacturers use Flash memory as **disk emulators** to provide a very fast disk drive, but it requires special drivers software to make sure that the number of erase and write operation remain minimum.

Parity

What is Parity?

Parity is a method used by the **Memory Management Unit** (MMU) to find if any error has happened when writing or reading the information stored into the memory.

Parity checking involves adding one extra bit to each byte of information in order to make the total number of ones (1s) in the byte odd or even.

In this method one additional parity bit is generated by the processor and stored with each byte during the memory write process and during the read process the state of this additional bit will inform the processor if the byte information is correct or has some error.

If because of some read or write error more than one bit changes its state then this method will not be useful to find the error.

Based on the parity generation method, two types of parity error checking method is used

- Odd parity
- Even parity

In the **odd parity system**, the number of bits that are 1, should be an odd number (including the parity bit).

In the **even parity system**, the number of ones including the parity bit should be an even number such as 2, 4, 6 etc.

ECC Memory

As we add more and more memory, the chance of data corruption due to even one bit going wrong increases. A computer with parity checking memory can only detect the error, it cannot take any corrective steps.

However, an Error Correction Code (ECC) memory can automatically detect and correct memory errors using special algorithms.

To use ECC memory, motherboard must have this option. When ECC memory is installed on such systems, the motherboard will detect and configure the system to use it.

In a 168-pin DIMM, 64-bit variety is without ECC and 72-bit variety is with ECC, so if one wants ECC facility, one should buy 72-bit DIMM.

Physical Memory Organization

Physically the memory used inside the computer come in as separate DIP chips and earlier PCs were designed with sockets on their motherboard to accept these chips as memory modules.

The main memory is divided into a number of sets or banks and normally nine chips are used per bank, 8 chips for eight bit of data and one additional chip for the parity information.

With the introduction of 32 bit processors 386, 486 and Pentium etc. each bank required 36 chips– 32 chips for 32 bit of data and 4 chips for 4 bit of parity.

If you have more than one bank then the number of chips in this type of computers will become unmanageable. This has lead to the concept of **memory modules**.

The memory module is a set of number of RAM chip on a single plug-in circuit board. The memory modules are available in the following type of memory packing

- DIP (Dual Inline Package)
- SIPP (Single In-line Pin Package Modules)
- SIMM (Single In-line Memory Modules)
- DIMM (Dual In-line Memory Modules)

Out of these packing the DIMM modules are in the most common use.

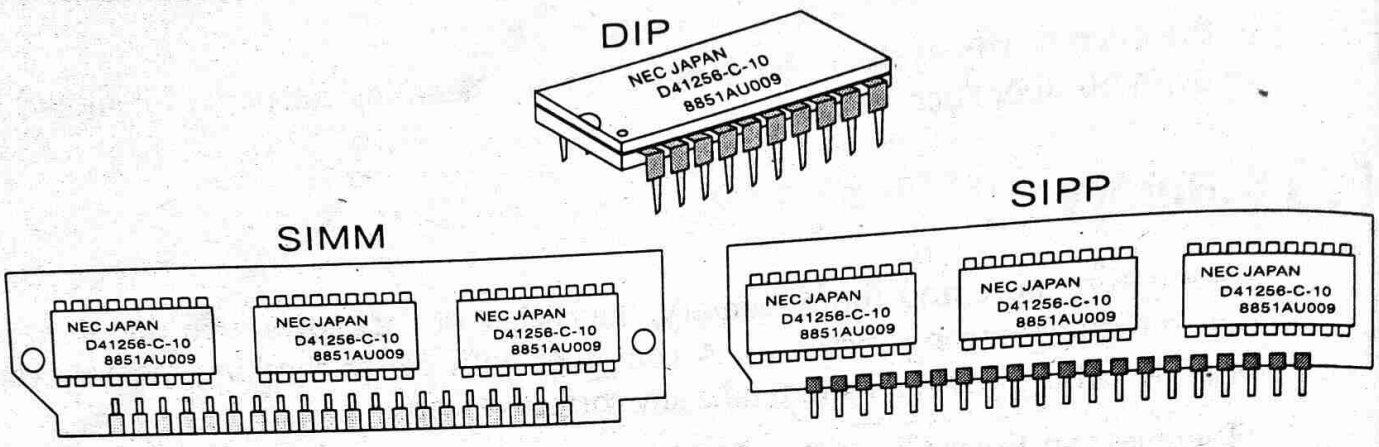


Fig. 7-4 : DIP, SIMM and SIPP modules.

DIP

DIP or **Dual Inline Package** was used to be the most common packing for the memory chips, it resembles a small flat, rectangle box with metal legs on both sides.

A typical size would be 1 inch length and 1/2 inch width and both the sides may contain 8 legs each to connect this chip to the computer circuitry, or to insert this chip into the socket on the computer motherboard.

The memory chip introduced with the first PC could store only 16Kbits of information each. Later with the AT machines 256Kilobit chips were introduced.

With the new machines such as AT-386 the trend is towards the memory modules. You may not find a new motherboard that accepts discrete memory chips in DIP packing.

When using the discrete chips in the motherboard one had be careful about the speed and capacity of the chips being used.

These chips also differ in the number of bits that one chip supplies to complete the 8 bit byte.

Some chips are available as 256K x 1 bit configuration. In this you will require total 8 chips to make a complete 256 Kilobyte memory, as each chip contributes only one bit of information.

- If the chip is 256K x 4 bit chip then you will require only 2 chips to make a complete 256Kilobyte of memory.

When using the discrete chips as the main memory, one should be very careful while inserting these chips into the socket on the motherboard.

- The most important point when inserting the chip into the socket is the orientation of the chip.

SIPP

SIPP or **Single Inline Pin Package** is almost same as the SIMM except the SIPP contains pins at the bottom to connect them into the motherboard memory socket

Most of the time these SIPPs are directly soldered onto the mother board itself rather then inserting them into the socket.

SIPP and SIMM have only one difference and i.e. their packaging

- on the SIMM memory chips are soldered on to the circuit board, and the connection to the socket is done using the edge connectors
- on the SIPP, the discrete memory chips are first soldered onto a small circuit board and then it is connected to the main board using small IC like pins
  these pins are either directly soldered onto the motherboard or installed into their socket.

Removal and insertion of the SIPP from their socket is easier than the removal and insertion of the SIMM modules.

- To remove a SIPP module there is no clips etc. to be removed, one can just pull the SIPP module from both sides using even force and remove it from the socket.
- To insert a SIPP module into it's socket, first place the module in proper orientation on the socket and slowly press both sides down using even force, this should fix the module into the socket.

SIMM

SIMM or **Single Inline Memory Module** is not a single memory chip, it is a memory module i.e. a number of memory chips soldered onto a small expansion board.

The edge connector of this expansion board is plugged into a special SIMM sockets on the motherboard.

- This design allows the memory to be added and removed from the computer without the risk of destroying it.
- Just like the discrete memory chips, the memory modules also come in different capacities.
- On a 32-bit computer such as 386, 486 etc you will have to add the 8-bit modules in a group of four modules to provide complete 32-bit or to fill a complete bank of memory.
- For these computers 32-bit memory modules are available where a single module provides complete 32-bit storage. On 32-bit machines, these memory modules can be added or removed individually.

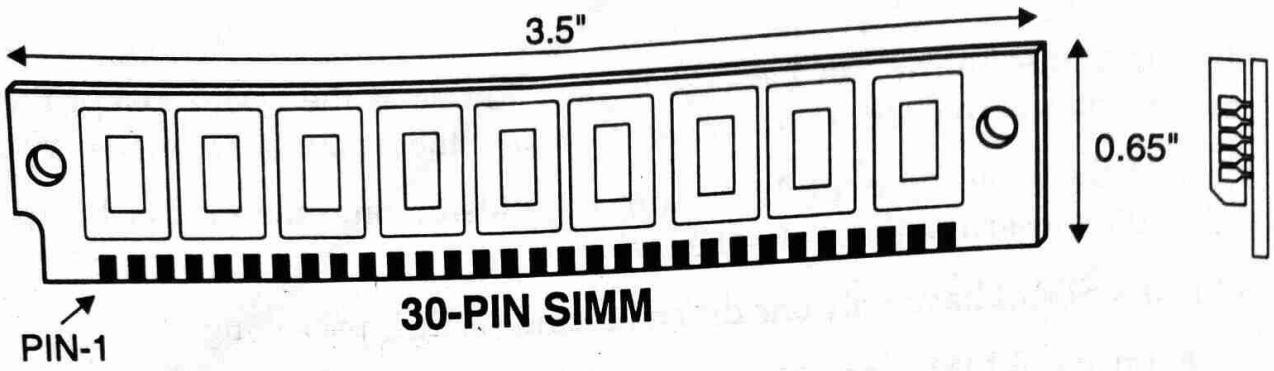


Fig. 7-5 : 30-Pin SIMM.

Other than the bit width these modules are available in different storage capacities, these are available in the capacity ranging from 256KB to 256MB.

SIMMs are available in SRAM, DRAM, VRAM etc., different technology, the chips used in the modules can be made of any of these technologies

- all the chips must be of the same technology, one cannot mix different technologies in the same module.

30-pin SIMM

30-pin SIMM is the first SIMM introduced with the PC system. The 30-pin SIMM measures about 3.5 inches wide and an inch tall.

30-pin SIMMs use one byte-wide data buses, you need multiple SIMMs to make a single memory bank with most microprocessors. PCs with 16-bit data buses (based on the 286 and 386SX microprocessors) will require two 30-pin SIMMs per bank.

PCs with 32-bit data buses (such as 386DX and 486 PCs) require four 30-pin SIMMs per memory bank. PCs with 64-bit data buses (such as the Pentium) would require eight 30-pin SIMMs per memory bank.

Using eight 30-pin SIMMs per memory bank is almost as ungainly as using discrete memory chips. This resulted in 30-pin SIMMs getting replaced with 72-pin SIMM modules.

72-pin SIMM

Compared to using individual memory chips, using 30-pin SIMMs were a big improvement, but with the introduction of new processors with wider data buses a new SIMM with more pins to accommodate wider data buses was required.

This lead IBM to introduce 72-pin SIMMs, its 72-pin could pack four byte-wide banks on a single module.

72-pin SIMM incorporates several interlocks to prevent you from plugging in the wrong style of SIMM or sliding in at improper orientation.

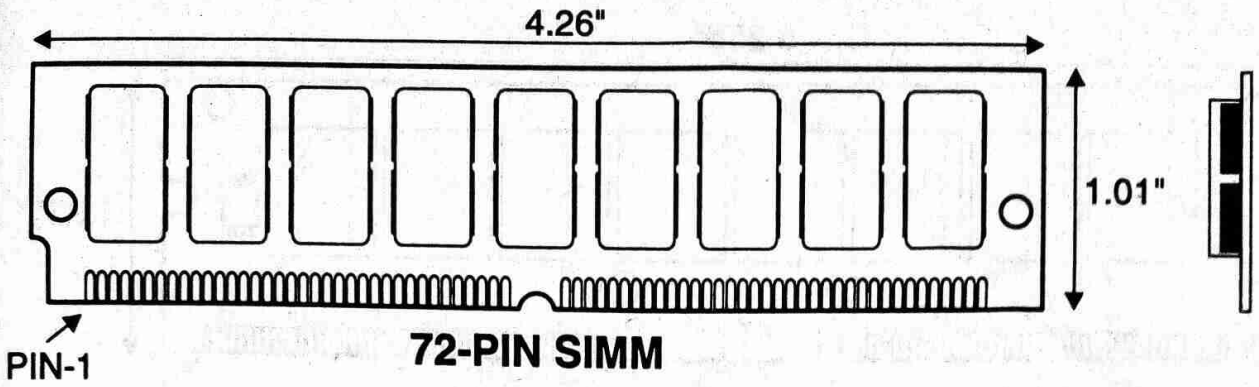


Fig. 7-6 : 72-Pin SIMM.

The notch in the center of the SIMM edge connector prevents one from accidentally sliding a 30-pin SIMM into a 72-pin socket.

Modules may have components on either one or two sides. To achieve higher capacities, 72-pin SIMMs are often double-sided. That is, they place chips on both sides of board.

DIMM (Dual Inline Memory Module)

Even 72-pin SIMMs fall short when it comes to Pentium and Pentium Pro PCs. With their 64-bit data buses, these chips would require two 72-pin SIMMs per bank, just as 486 machines required four 30-pin SIMMs.

To provide easier expansion for newer 64-bit machines, memory makers developed modules with more connections to permit wider addressing.

The resulting modules have 168 separate connections, arrayed across two sides of the module, a design that created the first Dual In-line Memory Modules.

Two rows of edge connectors, one on each side of the module, are divided into three groups with short gaps between. The first group runs from pin 1 to pin 10; the second group from pin 11 to pin 40; and the third group from pin 41 to pin 84. Pin 85 is opposite pin 1.

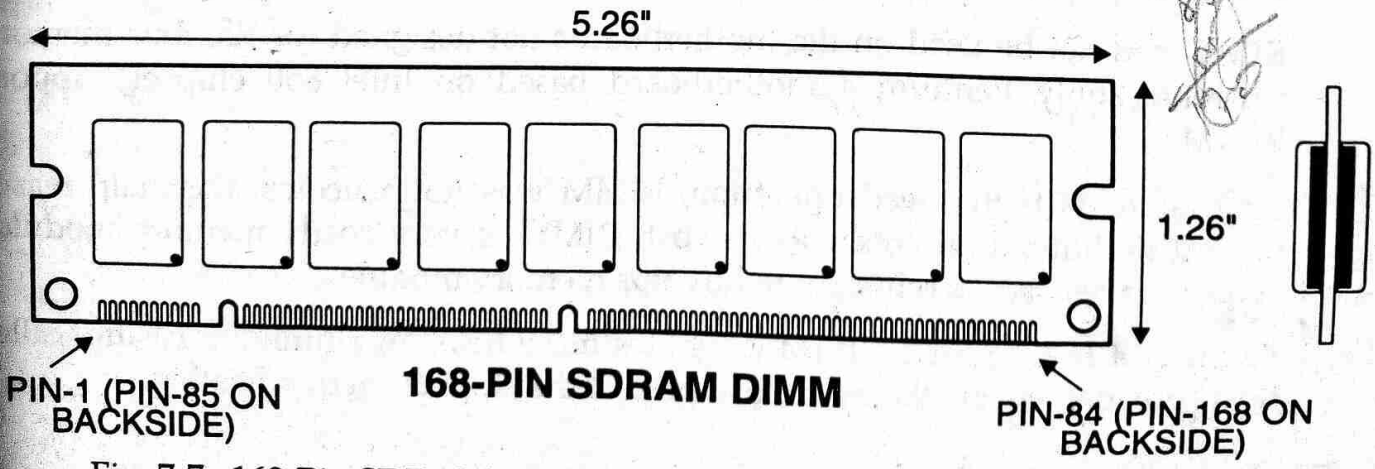


Fig. 7-7 : 168-Pin SDRAM DIMM.

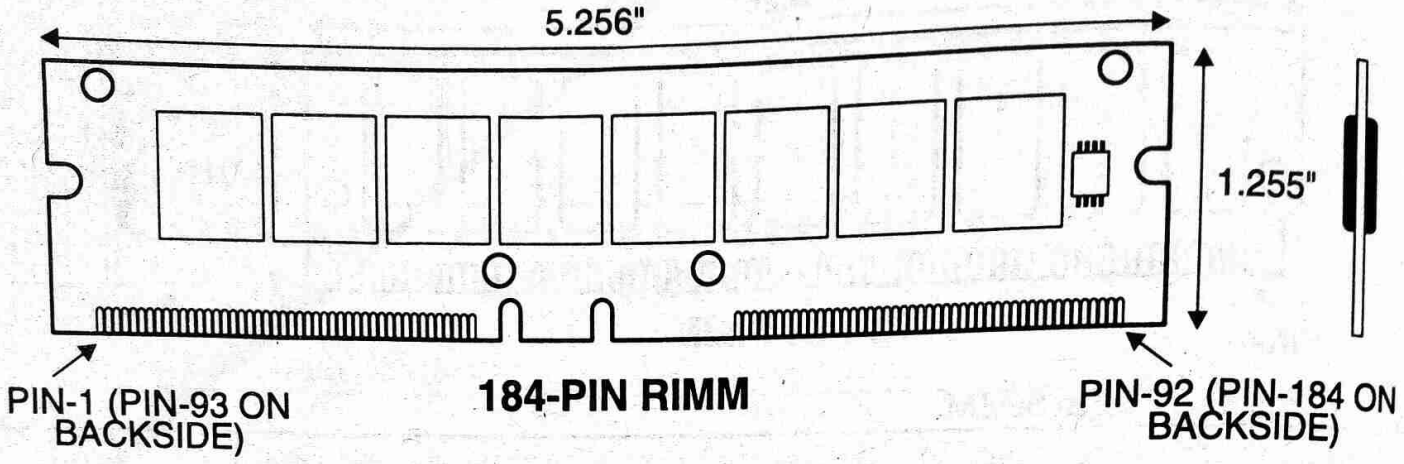


Fig. 7-8 : 184-Pin RIMM.

Notches in the edge connector of the DIMM prevent one from sliding smaller SIMMs with fewer connections into a DIMM socket.

Instead of a notch on the side of the DIMM, the asymmetrical arrangement of the notches on the DIMM prevent its unintentional improper insertion into its socket.

Some DIMMs have holes to allow you to latch the modules into their sockets, although some DIMMs lack these holes.

To accommodate the larger edge connectors and provide greater storage capacity, DIMMs are physically large, about 5.25 inches wide and typically one inch tall.

As with 72-pin SIMMs, 168-pin DIMMs include electrical provisions for telling your PC the speed rating of the module. The module connector provides eight pins for signaling this information.

RIMM (Rambus Inline Memory Module)

When Intel introduced Pentium 4 microprocessor, they wanted a very fast memory for it and there choice was RDRAM or Rambus DRAM memory.

Instead of using SIMM or DIMM, RDRAM comes in a special 184-pin RIMM, i.e. Rambus Inline Memory Module.

RIMM can not be used on the motherboards not designed for Rambus memory. Currently only Pentium 4 motherboard based on Intel 850 chipset supports RIMM.

Even after its high speed operation, RIMM was not asuccess, the main reason behind its limited success was its cost, RIMM is very costly memory modules. Which makes people reluctant to buy this memory module.

Because of faster speed, SIMM generates more heat, an aluminum casing, called heat spreader, covers the module to protect the chips from overheating.

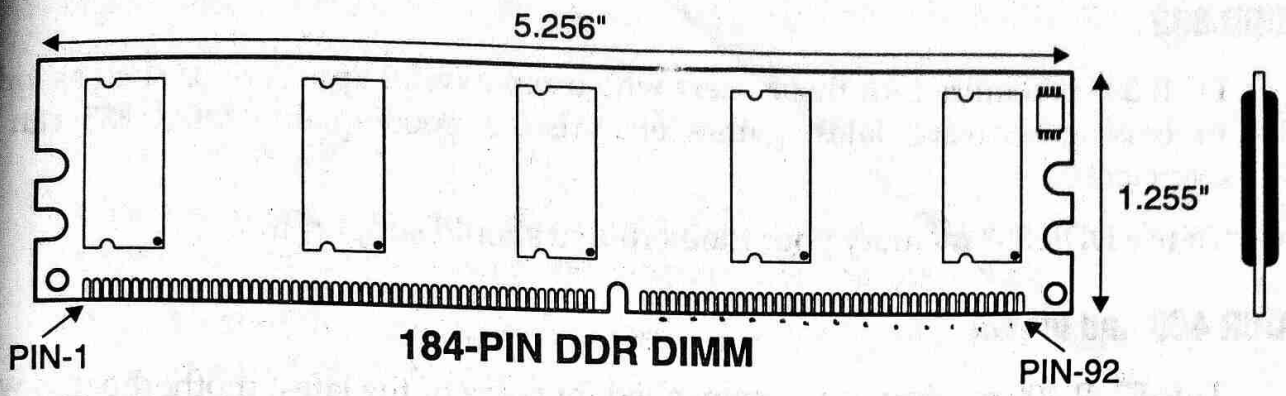


Fig. 7-9 : 184-Pin DDR DIMM.

DDR DIMM

A much cheaper Double Data Rate (DDR) SDRAM provides speed almost equal to the RDRAM. DDR SDRAM doubles the rate at which SDRAM process data.

SDRAM carries information only on the rising edge of clock signal, whereas the DDR RAM is able to read data on the rising as well as falling edge of the clock signal. This results in double performance by the DDR RAM.

DDR SDRAM is costlier than standard SDRAM but it is much cheaper than the RDRAM. This resulted in most of the Pentium 4 based system going for DDR DIMM, instead of going for costlier RIMM.

Also, the DDR RAM is widely supported by different brands of motherboards and its prices have been low and falling so it offers great value for money.

DDR DIMM use a connector with 184-pins. This connector contains a single notch near the center of the connector.

Some of the variations of the DDR RAM available in the market are

- DDR 266
- DDR 333
- DDR 400 and above
- DDR2 533

DDR 266

This is one of the cheapest DDR memories available. This is the minimum one should get for a P4 motherboard.

Even if the motherboard does not support dual channel DDR memory, one should install the maximum DDR 266 memory that one can afford.

DDR 333

DDR 333 is required for those users who use powerful applications such as video processing software, latest games etc. Also, a good quality DDR 333 can be overclocked.

To use DDR 333 memory your motherboard should support it.

DDR 400 and higher

This DDR 400 memory is recommended for some of the latest motherboards with at least P4 2.8 GHz or above or Athlon 64 processor.

With the DDR 400 memory and a good board and processor combination one can easily overclock the system. To get the best performance it should be run in the dual channel mode.

DDR2 533

This is a new DDR memory standard and boards that support this new memory standard are gradually beginning to show up in the market.

Once its price becomes competitive one should go for it, provided the motherboard supports it.

Memory Speed

The speed of a memory chips is always shown in **nano second** or **ns** which is one billionth of a second, PC memory speed vary from **60 to 100 nano second**.

- 1 ns = .000000001 second, so, The lower the nano second value the faster is the chip i.e. a 20 ns chip is faster than a 30 ns chip.

When installing a memory chip or memory module one should be very careful about the speed of the chip being used. Mixing chips of different speed may give problem and the machine may not work properly.

Using chips of higher than required speed can only waste the chip's additional speed, whereas using the chip of lower then required speed can only waste the speed of the motherboard or the CPU.

- A **wait state** is added in the memory read or write operation when the memory chip is slower than the CPU, in a wait state the CPU suspends all its work for one or more clock cycle for the memory to finish its job.

These wait states are a big block to the CPU speed, a wait state extends a normal memory cycle from 2 cycles to three cycles i.e. the total time is increased by one third and two wait state may double the total memory access time.

For example, if the motherboard is made for 100 ns memory chip and the chips on the board is of 120 ns speed then the board will work slower than its standard

speed, because the slow memory will make the CPU wait for the RAM read/write operation to finish.

If you replace the 120 ns chip with 100 ns or faster chip then the CPU will be able to work at its full capacity.

A very common symptom of mixing different speed of the memory chips is **parity error** or **computer failure**, i.e. the computer may not start at all.

- Most of the time the memory speed is displayed on the packing of the chip as a part of the chips part number
- Sometimes this may be shown separately as –12 or –7 etc. and sometimes it may be difficult to distinguish the speed value from the part number.
- 60, 70 and 80ns speed are very common so you can look for these numbers in the part number.
- The speed of a chip slower than 100ns is shown by only first two character, for example 100ns will be shown as 10 or 140ns will be shown as 14 etc.

Motherboard Memory Capacity

The memory capacity of the motherboard depends on the CPU and on the motherboard design.

If a CPU can address 4 gigabyte of memory, it does not mean that the motherboard using that CPU can address 4 gigabyte on the motherboard.

- A 486 or Pentium based computer has the capability to address 4 gigabyte of memory, but if you look at the cost of the SIMM memory module then it is unlikely that someone will put 4 gigabyte of memory in his computer.
- Nowadays most motherboards come with atleast 256MB memory on the motherboard.
- Most Pentium based motherboard have the option to add up to 1 GB of memory on the motherboard itself.

If the motherboard does not have a capacity to install as much memory you need, then you have two options

- get a new motherboard
- use memory expansion boards that plug into the expansion slots.

Some motherboards have an address limitation i.e. they do not access memory beyond a certain limit, may be beyond 16 MB or 32MB etc. on this type of motherboards adding memory beyond this limit will not have any use.

Reading Chip Number

Understanding the part number on the RAM chips is very confusing as there is no fixed or properly defined method used by all the chip manufacturer.

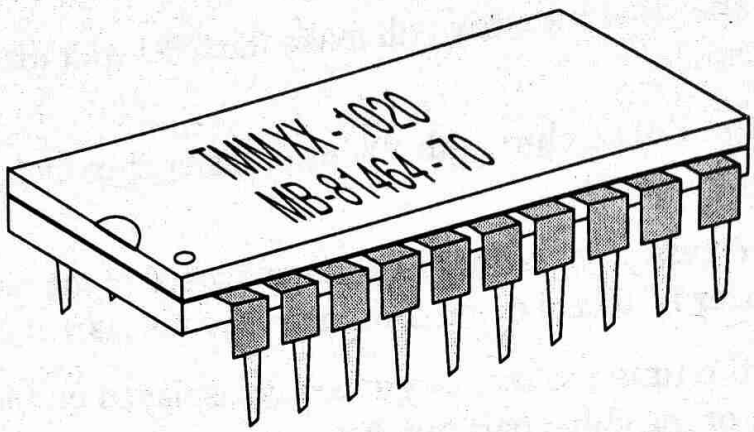


Fig. 7-10 : Reading Chip Number.

- Most of the time the speed of the chip is given separately as –10, –40 etc. The capacity of the chip in kilobyte and the number of bits that the chip supplies are also given in the chip number by most of the chip manufacturers.
- For example, a chip number MB81464–70 tells us that it is a 64K x 4 bit chip and its speed is 70 ns. Other than these values sometimes a chip may also contain the batch number or the date of manufacturing.
- The alphabets at the beginning of the code is used to identify the manufacturer. For example **Hitachi uses HM** as first two character, **Intel uses small i, Motorola uses MCM, National Semi. use NMC** etc.

As there is no fixed standard used by the chip manufacturers, one should look into data book to be sure about the speed and the capacity of a particular chip.

Logical Memory Organization

So far we have looked at the way the computer memory is physically organized, now let us see how this memory chips are accessed and used by the computer system.

Based on the motherboard or the CPU of your computer and the operating system used by you, the computer's main memory can be divided into different categories such as expanded, extended, high etc.

It is important to understand the different types of logical memories to make the computer use these memories efficiently.

Conventional Memory

The first PC and PC-XT systems used 8088/8086 processor as the main processor. These two chips had 20 bit address lines and the maximum address that these processor could access is 2^{20} Bytes or 1 MB.

- This address limit of 1 MB made the system designers to keep 640 KB of this memory as the RAM memory area.

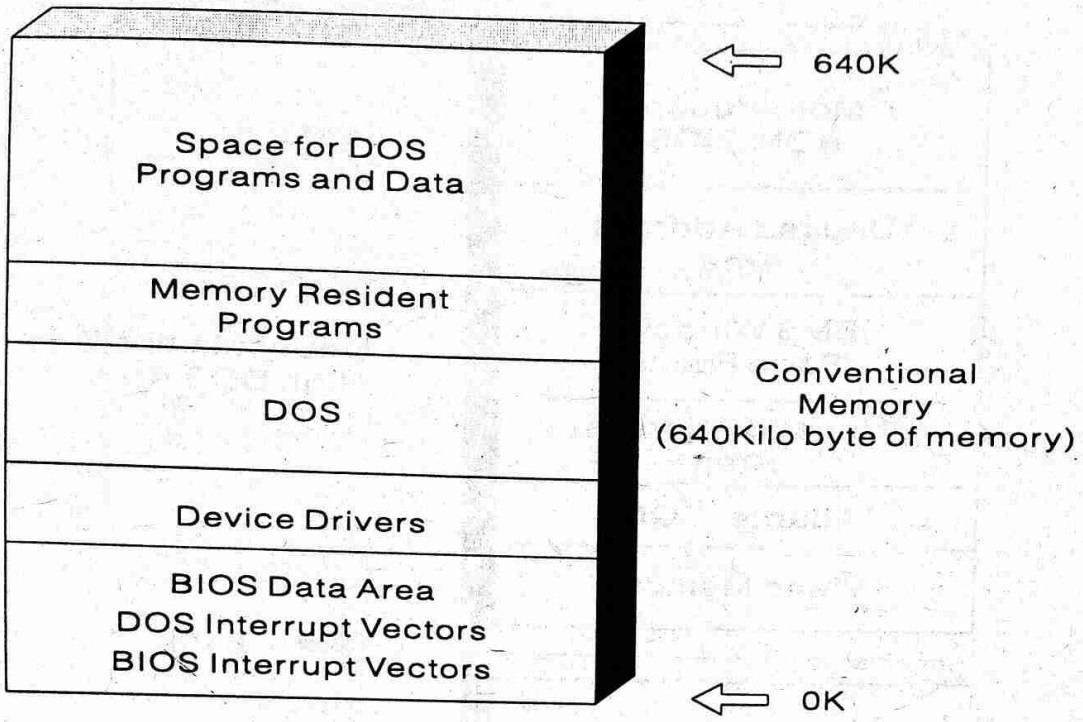


Fig. 7-11 : Conventional Memory.

- The 640KB may not sound like a big memory today but at the time of PC and XTs when most of the computer were having only 64KB of the memory 640KB was a very big memory.
- This 640KB which is used by the DOS and other programs such as WordStar, Lotus etc. is called the **Conventional Memory, DOS Memory** or **BASE Memory**.
- Even today when the addressing capability of the processor has reached gigabytes, the 640KB limitation still exists.

DOS still cannot use more than 640KB for most of its works, this limitations exists to make the old software and hardware compatible with the new generation software and hardware.

Upper Memory Area (UMA), High DOS Memory Area

The memory area between the 640 KB and the 1Mb is called the **Upper Memory Area** or the **high DOS memory area**.

There are many empty memory location in this area that is not used by the display memory or the ROMs, but these memory areas cannot be used by a PC as there is no physical memory assigned to this area.

On machines with memory mapping capability, i.e. 386 or higher processor with proper software driver or 286 processor with proper support circuitry on the motherboard and driver software, one can map these empty locations to some real memory area and use them for storing small driver or memory resident programs.

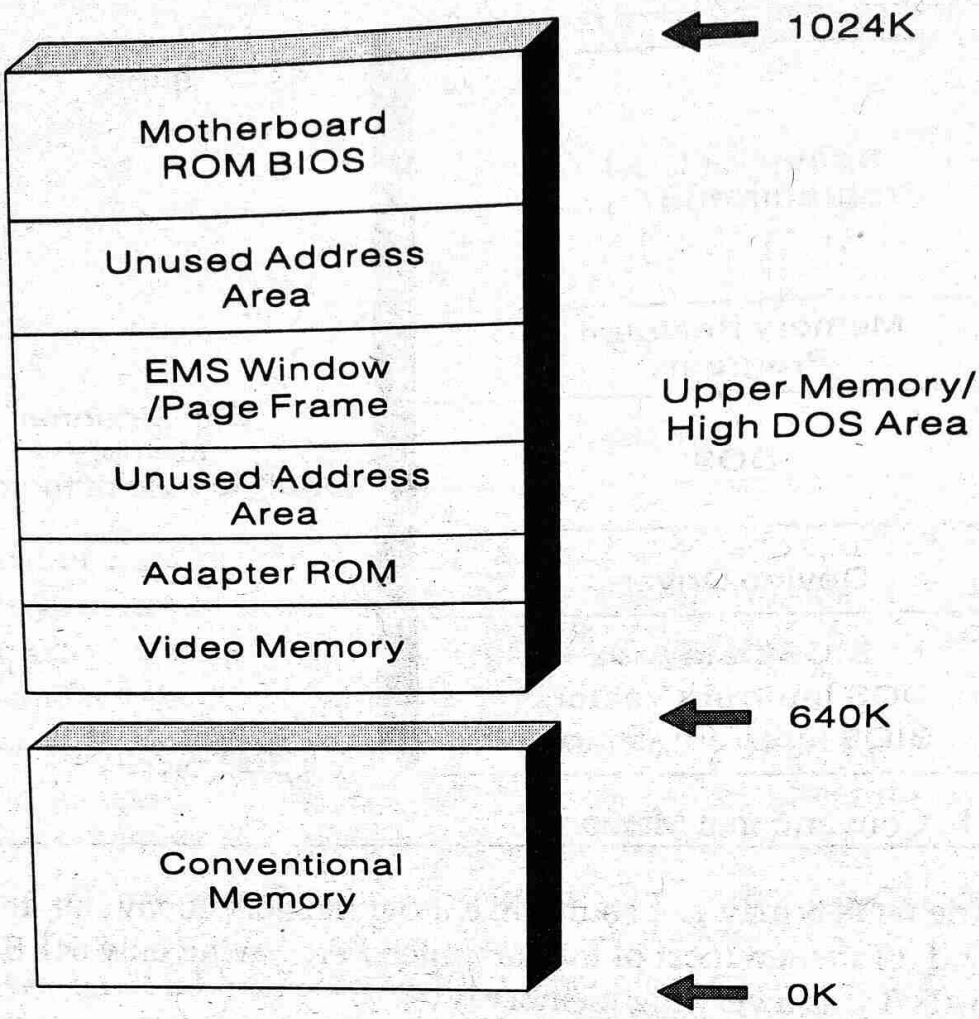


Fig. 7-12 : Upper Memory Area.

Big programs cannot be stored here because these memory may not be available continuously, it is available in small chunks of different sized memory.

Eventhough this memory can be used by user programs to store some data, a programmer does not use this memory area, because the memory area available in one computer may not be available in a second computer if the second computer is using different hardware/software configuration.

Newer operating systems such as **Windows** or **OS/2,** because of their strict memory management rules can use these empty areas in the upper memory location to store different information.

If your processor is a 386 or higher processor then using some memory manager such as **EMM386.EXE** you can relocate the drivers (mouse driver, network driver etc.) and memory resident programs from the **conventional memory** (0 - 640K) to the empty spaces in the **Upper Memory Area.**

This frees the conventional memory for some other purpose.

The **EMM386.EXE** is one of the most common memory manager that comes with the DOS and the Window software.

Some third party drivers are also available in the market for this purpose.

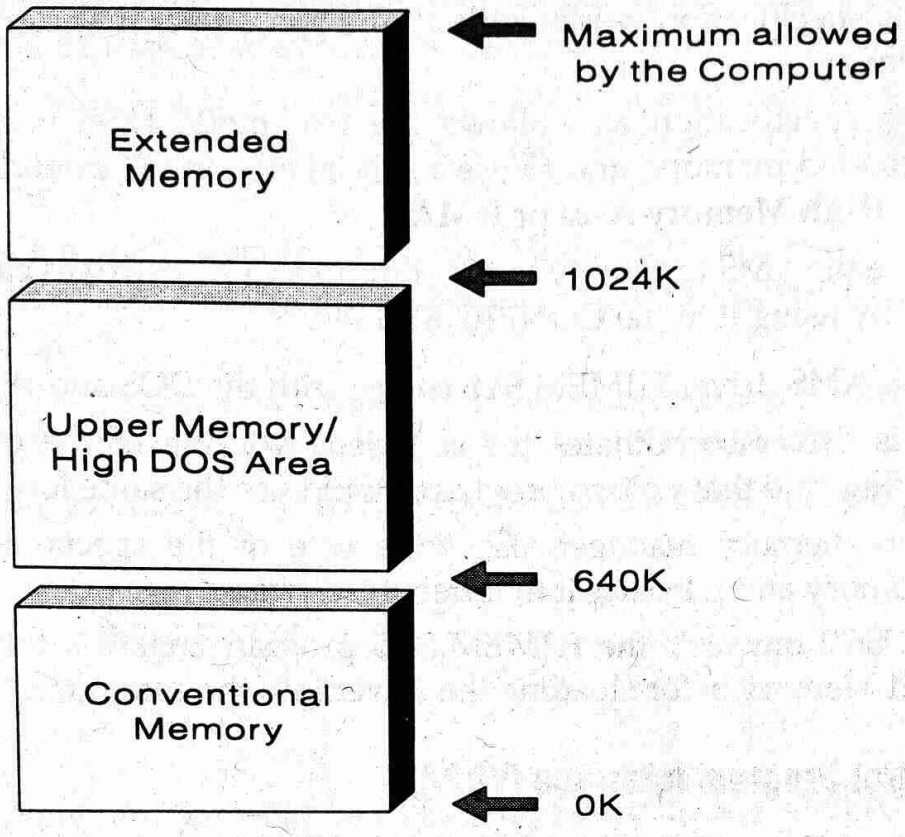


Fig. 7-13 : Extended Memory.

Extended (XMS) Memory

Extended Memory is the memory beyond the 1 MB limit, any memory available after the 1 MB memory is called extended memory.

The 8088/8086 cannot address memory beyond 1 MB, the extended memory is available in the 286 and later processor based computers only.

Based on the addressing capacity of the 286 processor it can have memory up to 16 MB and the 386 processor can have up to 4 gigabyte of extended memory.

The extended memory is not much useful for the DOS users because the DOS does not know how to use this memory, for a DOS user this memory can be used only as **disk cache** or **RAM disk**.

For the Windows and the OS/2 user this memory is very useful as these operating systems can use this extended memory and any program specifically written for these environment can use the extended memory available in the computer system.

Window and OS/2 operating systems allow multiple DOS program to run in the extended memory, each program working in its own 640K memory area.

Extended Memory Specification (XMS)

Extended Memory Specification or **XMS** is a standard developed jointly by the Microsoft, Intel, Lotus and the AST research in the 1987.

- This specification works with all the processors that can address extended memory.
- This specification also allows the real mode DOS programs to use the extended memory, and to use a special area in the extended memory called the **High Memory Area** or **HMA**.

To provide the XMS capability to the computer, a XMS drives called **HIMEM.SYS** is loaded by using it in the **CONFIG.SYS** file.

- This XMS driver HIMEM.SYS comes with the DOS and Window.
- This driver coordinates the complete working of the extended memory, taking care that no two program should use the same location.
- This memory manager also takes care of the special **HMA** area of the memory and allocates it to different software.

From DOS6.2 onwards the **HIMEM.SYS** program makes a thorough test of the Extended Memory before loading the driver into the computer.

Virtual Control Program Interface (VCPI)

The **Virtual Control Program Interface** or **VCPI** is another extended memory manager specification developed by the Phar lap software (developer of DOS extender) and the Quarterdeck systems (developer of Q-EMM-386, an extended memory manager).

This specification was mainly developed to make the DOS programs in the virtual 86 mode to work without any conflicts. This specification uses software interrupt **67h** for the virtual 86 programs to communicate with each other.

Any application specifically written for the VCPI includes a memory manager and the first VCPI program loaded into the memory acts as a memory manager and takes the responsibility of linking the interrupt with all the VCPI programs loaded following it.

DOS Protected Mode Interface (DPMI)

DOS Protected Mode Interface or **DPMI** is the latest extended memory manager standard introduced by the Microsoft in 1990.

This standard first introduced with the Windows 3.0 in the market. This specification includes all the functions of an extended memory manager as well as of a DOS extender.

High Memory Area

HMA or **High Memory Area** is a 64 KB of memory at the beginning of the extended memory.

- The HMA starts at the 1 megabyte limit i.e. at the location 1024 KB and goes up to 1088 KB of memory.

- Because it is not contiguous with the address range of lower memory, it cannot be used as extra memory by ordinary DOS applications.
- From version 5.0 onwards the DOS can use this memory as a part of the conventional memory, i.e. some part of the DOS can be stored at this location.
- This HMA is different from the **High DOS memory** area or the **upper memory** area, which is the memory location between the 640KB and the 1024KB.
- **HIMEM.SYS** or some other XMS memory manager must be loaded using the **CONFIG.SYS** file before loading the DOS into HMA.

The commands in the CONFIG.SYS file to load the DOS in HMA is

```
DEVICE=HIMEM.SYS
DOS=HIGH
```

This two command lines can be placed anywhere in the **CONFIG.SYS** file.

- The use of HMA is controlled and coordinated by the HIMEM.SYS extended memory manager.
- This memory is used be the **HIMEM.SYS** program by making the 21st address line i.e. the line A20 active during real mode using a program instruction.
- This makes one segment of additional memory accessible by 286 and better microprocessors in real mode.
- This allows the use of first 64KB of the memory above the 1024 KB memory.
- This area can be used by the device drivers and memory resident (**TSR - Terminate and Stay Resident**) programs.

From DOS 5.0 onwards some part of the DOS, about 40 KB can be moved into this area.

However, in this area you can store only one program driver or TSR and the program size should be less than 65,520 bytes.

Expanded Memory (EMS) (LIM)

What is EMS?

EMS or **Expanded Memory Specification**, is a specification which defines a method to access system memory above 1MB of RAM on primarily PC-XT and AT compatible computers.

This memory is accessed via a 16KB **window** within the first 1MB or memory. It is accessed in 16KB **pages**.

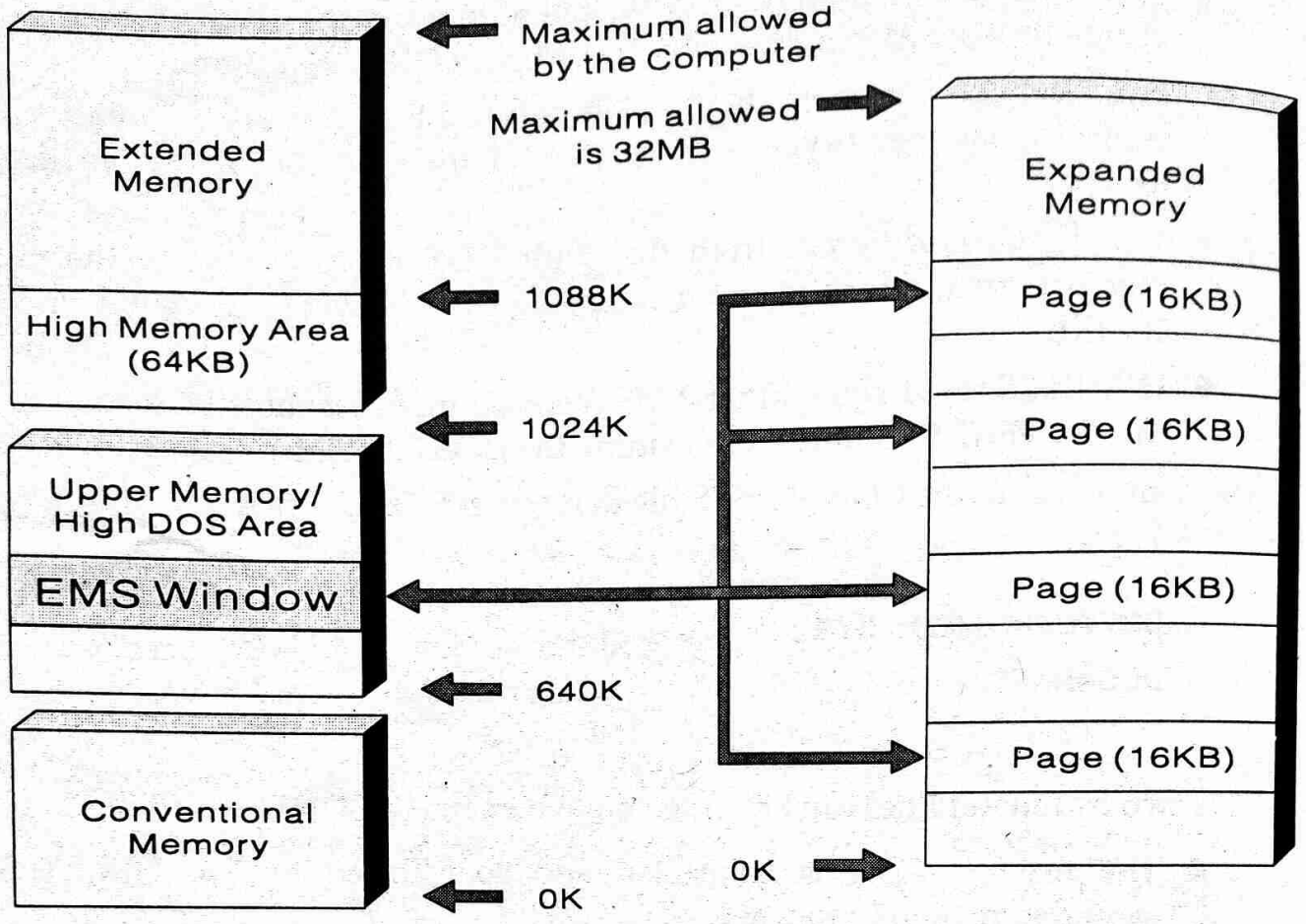


Fig. Expanded Memory.

The **Expanded Memory** or the **EMS** may sound similar to the **Extended Memory**, but it is a completely different type of memory specification.

This type of memory is mostly limited to the 286 based computers.

On a 386 and later machines because of the speed and memory management capabilities the available extended memory can be used as an expanded memory using proper driver software.

Compared to the 1 megabyte address limit of the 8088/8086 processors the 80286 processor could address 16MB of memory.

But, the DOS was not capable of addressing memory beyond the 640 KB of RAM.

To solve this problem Lotus and Intel has developed a method called **Expanded Memory Specification** or **EMS**. Later on the Microsoft also joined with the group and this specification was called **LIM EMS** version 3.2 (lotus, Intel, Microsoft, Expanded Memory Specification).

This specification is called with many different names such as **EMS Memory, LIM Memory, Expanded Memory** etc.

- The EMS is a completely different concept from the XMS. XMS memory is the part of the main memory that continues beyond the 1024K or 1MB.

- The EMS is not part of the main memory, it is a separate memory installed into the system which can be accessed in a fixed sized pages using a method called "**bank switching**".
- In this method a small window in the main memory is used to view the content of the EMS. This window is located in the memory location between 640KB and 1024KB i.e. in the upper memory area.
- The EMS memory is arranged in the blocks of 16 KB each, to access this memory, 1 block of the EMS is copied into the window in the main memory and after the processing it is copied back to the EMS memory.

The original EMS had many limitations such as,

- the EMS window is in the upper memory area and as the DOS cannot use this area for program code, program code could not be stored in the EMS, reading and writing in 16KB pages used to take a lot of time.

The AST's **Enhanced expanded memory standard** (EEMS) and IBM's **Expanded Memory Adapter** (XMA) tried to solve some of the EMS problems but they were not compatible with each other.

The LIM EMS version 4.0 solves most of the problems associated with previous versions of EMS standard.

A20 Line

The twenty first address line in 286 and later processors has a special use for the memory management.

This address line i.e. A20 can be made active in the **real mode** of the processor and the DOS will get additional 64KB of memory as a part of the **conventional memory**.

Most of the motherboard control this line using the keyboard controller (generally **Intel 8042**). This 64KB of upper memory area is managed by the DOS program **HIMEM.SYS**.

Speed Improvements

The overall speed or the throughput of the computer can be increased by increasing the speed of the RAM or by adding some new concepts such as Disk caching, RAM disk, Shadow memory etc.

Let us see what these are

Cache Memory

With each new model of the processor, the processor's speed is on the increase, but the speed of the memory chip has not increased much compared to the processor speed.

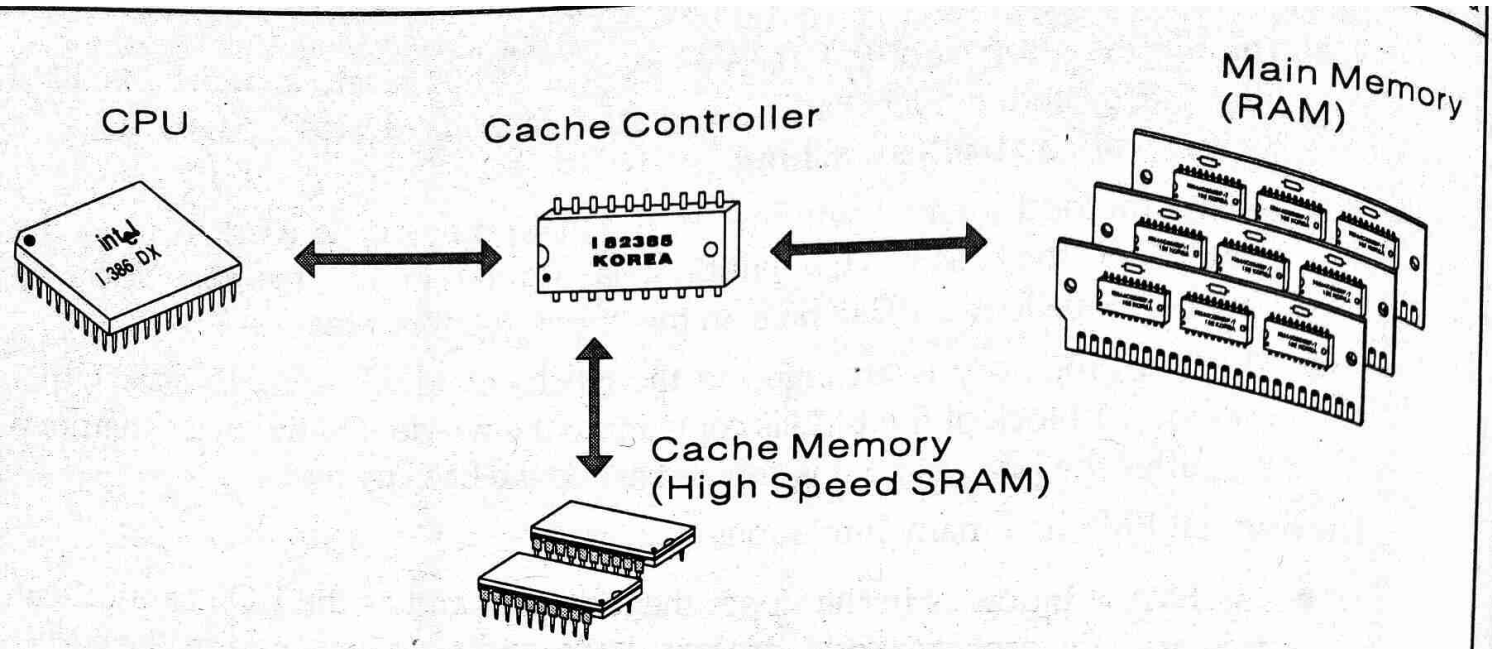


Fig. 7-15 : Cache Memory.

- High speed RAM chips are available in the market but they are very costly to be used as the main memory

This makes the CPU to wait for the result from memory. After any memory read or write command is given. Once the slow memory finishes the read or write operation then only the processor can continue with the next job.

- Cache (pronounced "**cash**") memory is a very small amount of very high speed memory used in between the main memory (RAM) and the processor.

The information frequently required by the processor is kept in the cache memory by a **cache controller**.

This cache controller always tries to make sure that the data required by the processor in the next memory access is available in the cache memory.

This improves the speed of the computer very much because if the required data is in the cache memory it is made available to the CPU without any wait state.

- The high speed memory used for the cache purpose is very costly that is the reason it is not used as the main memory of the computer.
- Using a proper size cache and a good logic to guess the next data required by the CPU, the cache can contain the required data 99 percent of the time.

As the CPU often access the data in a sequential order, when the memory is read by the CPU, the cache controller reads the data around the area read by the CPU and stores it in the cache memory.

This improves the chance of the data being available in the cache at the next memory read request by the CPU.

The performance of the cache depends on the

- **speed and size** of the cache memory and

- the logic used to read the data around the memory area read by the CPU. The better the logic used to guess the next required data, the better will be the chance of data being available in the cache at the next memory access.

When the required data is available in the cache it is called a **hit** and if the required data is not available in the cache memory it is called a **miss**.

- The logic used to guess the next required data by the processor plays a big role in the efficiency of the cache system.

Most of the new processors have the cache memory added in the CPU chip itself to speed up the process even further. Even on these CPUs with built in internal cache, you can add external cache.

This external cache is often called **secondary** or **second level (L2)** cache and could be of 128KB or 512KB size.

Newly introduced P6 processor from the Intel has this L2 cache on the processor itself. P6 has option of 256KB or 512KB secondary (L2) cache on the processor.

This cache uses a separate cache bus, different from the memory bus, to read/write the cache memory.

Logically the cache is implemented using the following methods.

- Direct Mapped
- Full Associative
- Set Associative

Direct Mapped

This is the easiest method to implement, in this method the cache is divided into small units called lines and each lines are identified by an index.

Then the main memory is divided into blocks of the size of cache memory. The line of the cache corresponds to the location within the memory block, each line can be drawn from different memory blocks, but from the location corresponding to the location of the line in the cache memory.

The block of the memory from where the line is drawn is identified by putting a label.

Whenever the CPU requests for a memory operation the cache controller checks the label at the particular index value to find out if the required memory location is in the cache or not.

Full Associative

This is an opposite implementation than the direct mapped cache memory.

- In this implementation each line of the cache memory can correlate or associate with any memory location of the main memory.

- In this implementation whenever the processor requires access to some memory location, the cache controller must scan through the address associated with every line of the complete cache memory to check if the requested memory is in the cache or not.

Set Associative

This is a mix of the both the above methods of cache implementation. In this method the cache is implemented as many small direct mapped cache areas.

If the cache is divided into four areas then it is called four-ways set-associative cache.

The cache can be divided into more number of ways but four-way implementation is the most common because it gives maximum performance for the minimum complexity.

If the number of areas are increased more than four then the complexity of the implementation compared with the performance gain will not be worth.

This implementation solves the problem of accessing the same line from different blocks of memory.

Whenever a memory access is requested by the processor, the cache controller searches the label at the particular index value in all the areas to find out if the required memory location is in the cache or not.

Increasing the number of areas also increases the searching time required by the cache controller to find if the requested memory location is in the cache or not, which will finally slow down the cache, reducing the benefit of dividing the cache into areas.

Internal Cache

The **internal cache** is internal to the processor i.e. it is a part of the same chip as the processor. This cache is also known as **primary cache** or **L1 cache**.

The 486 processor has 8KB of primary cache.

As the primary cache is a part of the processor itself, it has direct connection with the processor, this improves the access time of the primary cache.

This cache is sometimes divided into different types based on the function, such as **data cache** and **instruction cache** as used in the 586 and P6 processors.

This further increases the performance of the cache system.

The main limitation of the L1 cache is its size, as this cache is made as a part of the processor– internal to it.

The most common size of this type of cache memory is 8KB.

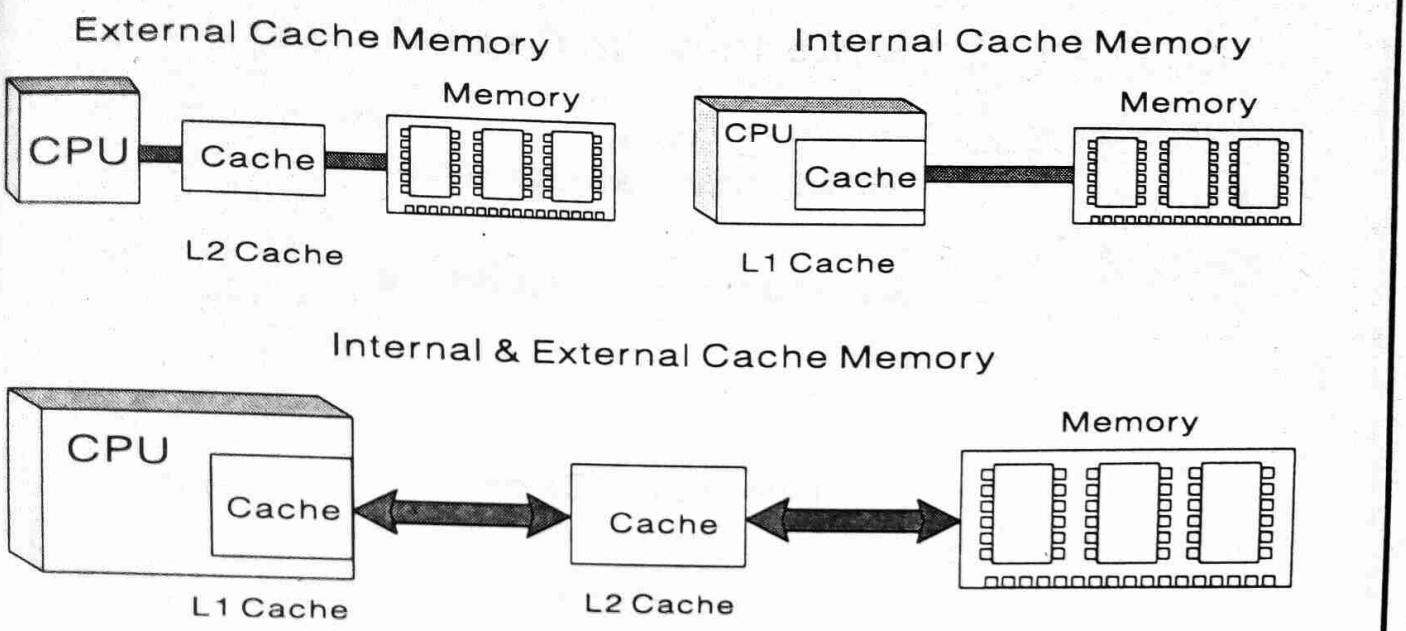


Fig. 7-16 : Internal and External Cache Memory.

External Cache

External cache is a separate high speed memory in between the processor and the main memory which is controlled by a **cache controller**.

This cache is also known as the **secondary cache, second level cache** or the **L2 cache**.

This cache is called the external cache because it is external to the main processor but the recent P6 processor from the Intel has put this cache also within the processor inside the same packing.

When this cache is external to the processor this does not have the speed benefit of the primary cache memory, but when it is external, it has another advantage that the size of the L2 cache can be changed by the user.

The most common size of the L2 cache memory is 128KB, because this size gives good enough improvement in the performance at a reasonable cost.

Write Through

This cache caches only the read operation, it does not cache the write operation.

The write through type of cache memory makes the data go through the cache to the main memory.

Once the data is written to the main memory then only the processor can continue with next operation.

This slows down the overall performance because for each write operation the processor has to go through the normal wait state.

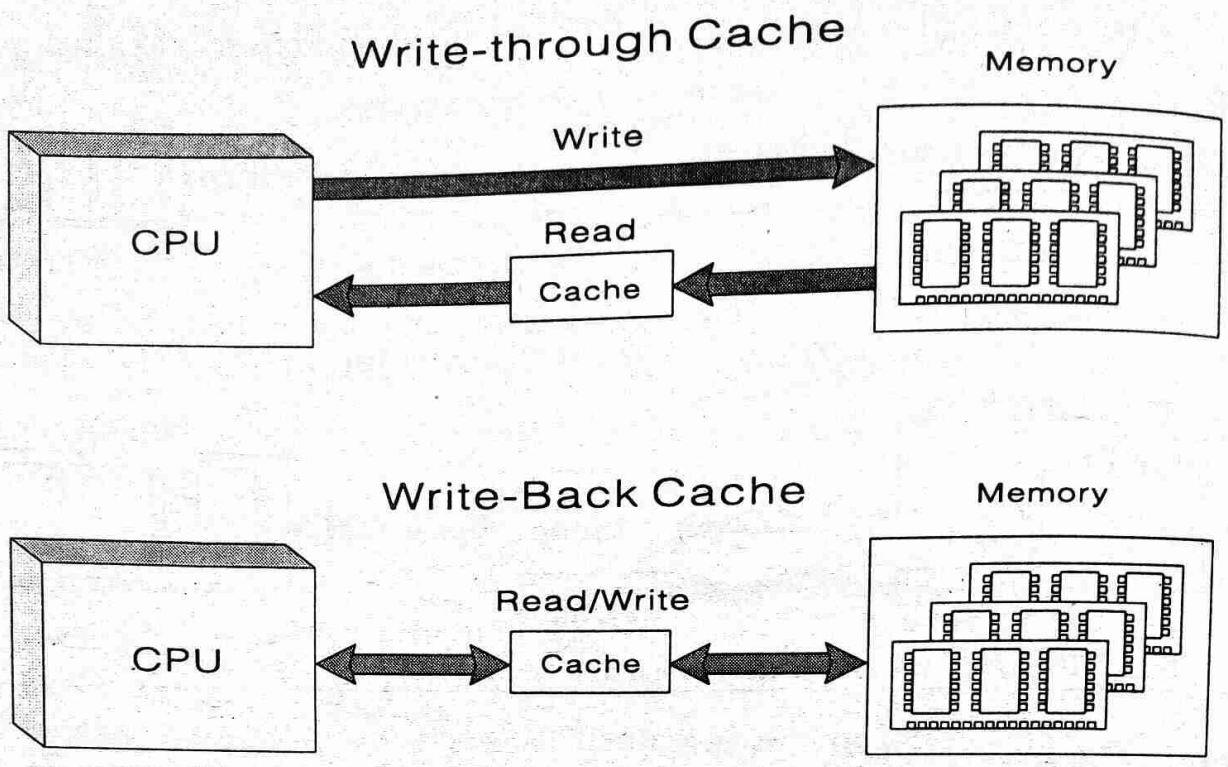


Fig. 7-17 : Write-through and Write-back Cache.

Write Back

This type of cache, caches both the read and the write operation.

When some data is written to this type of cache, the cache will accept the data and inform the processor that the data is written.

Afterwards this data is written back to the main memory by the cache controller, whenever it finds some free time.

If not properly implemented then this type of cache may have a problem of data mismatch between the main memory and the cache memory.

To solve this problem the cache controller is made to constantly monitor the address and the data bus and check for any changes being done to the memory. This is called "**snooping**" by the cache controller.

This caching method provides the maximum speed improvement.

Shadow Memory

What is Shadow Memory?

In 80386 and later processor copying the content of the slow BIOS ROM into faster system RAM is known as shadow memory. It is also known as shadow RAM or shadow ROM.

By copying the ROMs contents into empty area of the RAM and by redirecting any ROM read request to the RAM the read operation can be made faster.

- The RAM memory is about three to five times faster than the ROM memory.

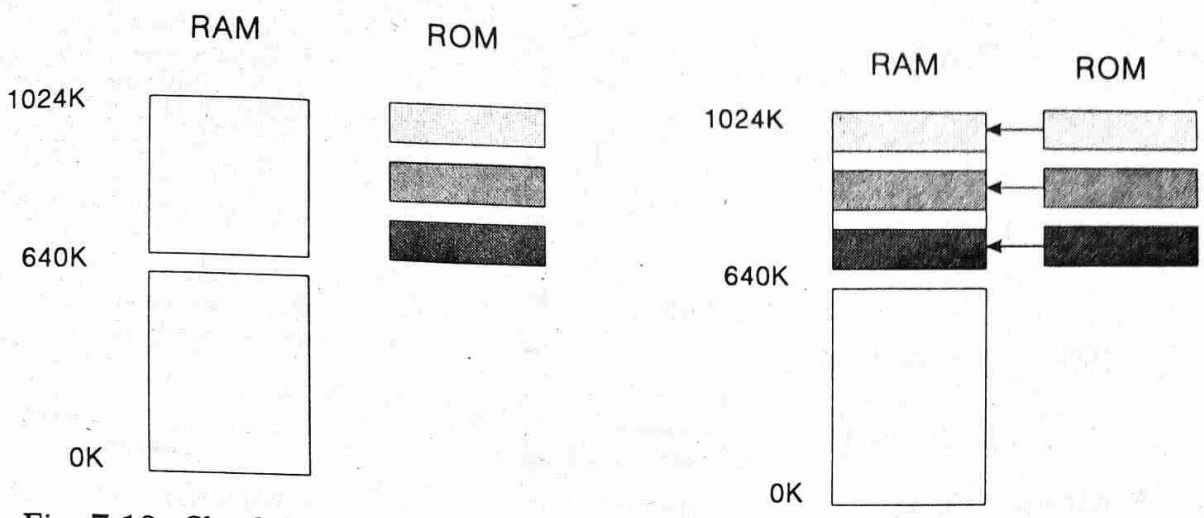


Fig. 7-18 : Shadow Memory.

- The BIOS and other ROMs on the computer give slow response whenever they are read by the processor.
- This is called **Shadow Memory** because the RAM memory contains exact copy or the shadow of the ROM.

The shadow memory provides an improvement in the computer speed by providing fast response to the ROM request of any computer operation.

- Software that write directly into the video, bypassing the BIOS ROM, or any other software that works directly with the hardware bypassing the BIOS ROM will not gain any speed improvement by using the shadow memory.

The shadow ROM option is provided in the **BIOS Setup** option.

Upgrading / Adding Memory

Upgrading of the memory can be done by installing faster and higher capacity memory. To install new memory you have to decide about the type, capacity and the speed of the memory you want to install in the motherboard.

When installing new memory in your computer the steps that you should follow is explained next.

SIMM

SIMM is easier to upgrade compared to the discrete memory chips. SIMM can be removed and installed any number of times without any difficulty.

Removing SIMM

- To remove a SIMM switch off the computer and remove the power cord.
- Remove the cover of the computer and locate the SIMMs.

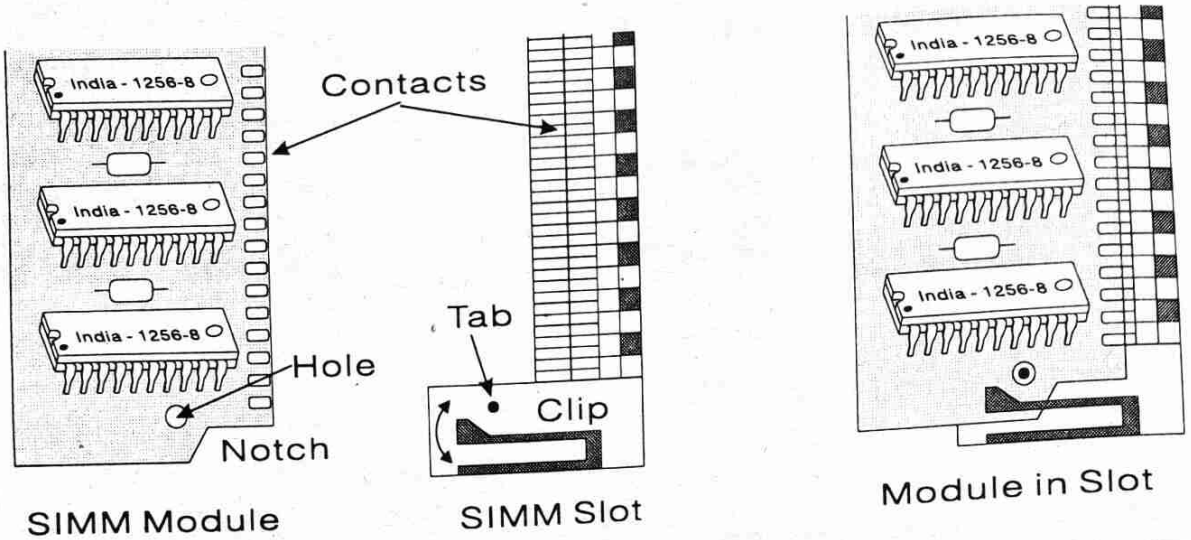


Fig. 7-19 : SIMM Socket.

- The SIMM is hold in the place using clips on both the sides of the SIMM module, To remove the SIMM you need to push these clips slightly out.
- Once the module becomes free from the clip, push it out of the tab onto which the hole of the module is connected.
- Once the module comes out of the hole you can lift it out.
- Once the SIMM is removed from the board put it in some safe place.
- When handling the SIMM do not touch the edge connectors, the oil from the hand can make the connectors bad.

Adding SIMM

- Locate the empty socket where you want to install the SIMM, now position the SIMM at some angle on the empty SIMM socket.
- Make sure that the orientation of the SIMM is correct and is edge connector is placed properly on the socket.
- Most of the SIMM module contain a notch on one side so that they cannot be inserted in wrong orientation.
- Once the SIMM is placed properly into the slot gently push it into straight position, which should lock it in using two clips on both the sides
- Never force the SIMM module into the slot, if excessive force is required then you are doing something wrong.
- Two tabs on the socket should go into the two holes on the SIMM module.
- Repeat above steps for all the SIMMs that you want to connect.
- Once all the SIMMs are placed into the socket double-check all the SIMM modules that they are placed properly into the socket.
- Check the unit by switching it on before closing the main cover.
- Most of the BIOS automatically detect the correct memory size.

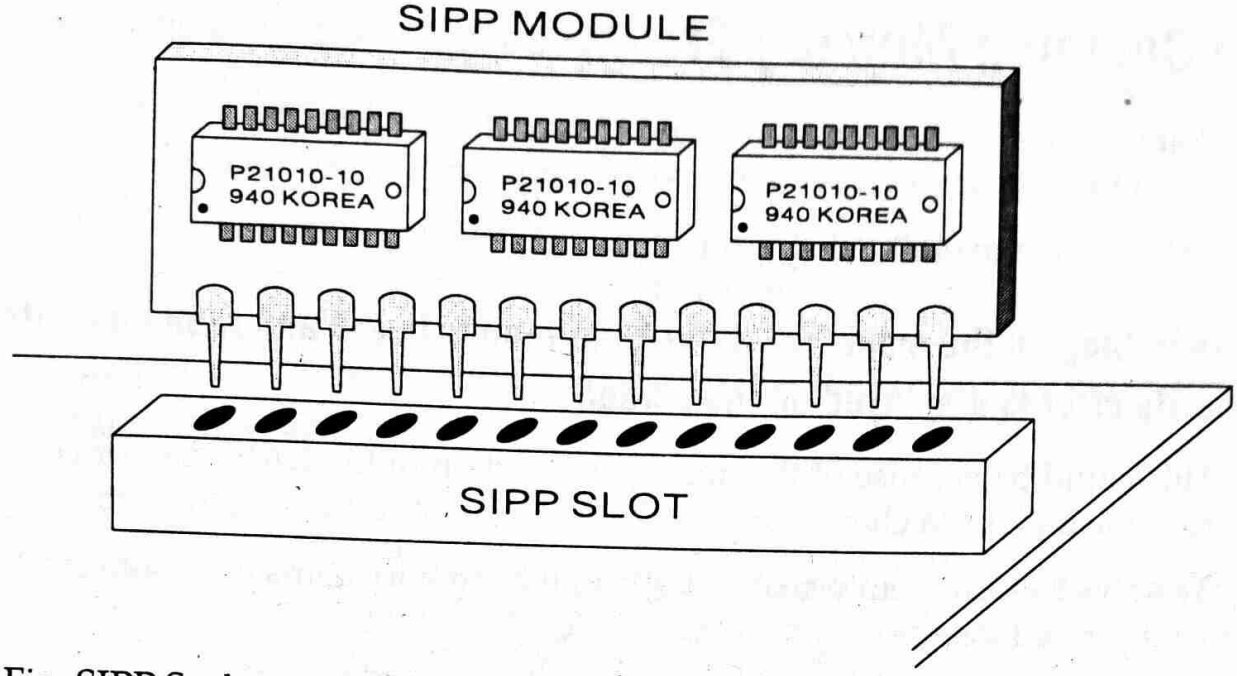


Fig. SIPP Socket.

SIPP

Adding a SIPP is just like adding a SIMM module to the computer, all the steps are same except the step required for the actual placing of the module.

- The SIPP module is installed into the special SIPP socket just as you insert a memory chip to the socket.
- Place the SIPP module with all its pin into the socket and then firmly push the module down into the socket.
- Make sure that no pin should bent or come out of the socket and that the orientation of the SIPP is proper.

As the SIPP contains pins just like the ordinary ICs, its pins may break with frequent insert and removal.

DIMM

Before installing a DIMM, please make sure to disconnect power supply.

- To unlock the DIMM slot, you need to push chip holder tabs at both ends slightly out, away from the slot.
- Align DIMM on the slot such that the notch on the DIMM matches the break on the slot.
- The DIMM only fits in one correct orientation. It will cause permanent damage to the motherboard and the DIMM if you force the DIMM into the slot at incorrect orientation.
- Firmly insert the DIMM into the slot until the retaining clips at both ends fully snap back in place and the DIMM is properly seated.

Some Common Memory Errors

You install some new memory, and the machine is switched on. On switching on computer stops with some error beep of error message on screen

You may get the following error

On switching on the machine, three beep tone is heard and then computer stops, or parity error is displayed on the screen.

This could be because of the improper insertion of the RAM chip or could be due to some bad RAM chip.

To solve this problem visually check all the memory chips you have installed for a bent pin or loose pin.

This error is not usually found in the SIMM or SIPP modules.

If the problem chip is not found by visual inspection then one by one remove and reinsert each chip and check if the problem has disappeared.

If the problem is due to some bad RAM chip then check each chip by replacing it with a new working chip and check for a defective chip.

When the computer performs a normal boot procedure it does a **Power On Self Test** (**POST**) procedure, it checks if the memory is working properly.

Wrong Memory Size

If the machine shows wrong memory size error on switching on, it could be due to

- On XT machine due to wrong setup switch setting
- On AT machine it could be due to wrong CMOS setup information.

To solve this problem on XT machine correct the **setup switch** and then switch on the machine.

On AT type machine run the **setup program** and change the RAM size to correct value and reboot the computer.

The memory card installed is not being recognized by the computer

The reason behind this could be improper installation of the card

- the card may not be seated properly in the slot
- the switch setting on the card may be wrong
- some memory chip on the card may be not working properly i.e. it may be defective or may not be seated properly in the socket.

Check all the above conditions and then rectify this.

Memory on the expansion card is partially recognized.

This again could be due to

- wrong switch setting on the motherboard or the memory card

Check all the switch and jumper setting make sure all the bank are properly activated and the chips are properly installed into them.

After installing some expanded memory card (EMS) the memory manager for the EMS is not getting loaded.

This error could be due to

- the memory card may not be EMS capable or the driver being used may not be compatible to the card installed. check and make the changes required.
- This could also be due to the driver not being used in the **CONFIG.SYS** file or the file being in some other directory than the directory specified in the **CONFIG.SYS** file.
  Check the **CONFIG.SYS** files and make the necessary changes. As the **AUTOEXEC.BAT** file is executed only after the complete **CONFIG.SYS** commands are executed putting path setting for the driver in the **AUTOEXEC.BAT** file will not be of any use.

Parity Error

One very common reason behind this error is a bad memory chip, or a chip not being seated properly into its socket.

- Check the chips to make sure that they are seated properly, and if nothing is found then replace the chips one by one to find the chip with the problem.

This error could also be due to insufficient capacity of the power supply.

- In this situation the computer may work fine until a disk access is required, when a disk access occurs the power supply to the RAM chips may get reduced and the parity error may happen.
- A power supply with a bad power to the computer or some device near the computer which is generating some electrical noise can also sometimes generate the parity error.