Computer Architecture and Organization

**UNIT - III**

**MEMORY SUBSYSTEM**

**STORAGE TECHNOLOGIES**

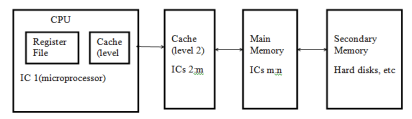
**Q. Draw conceptual organization of a multilevel memory system in a computer. Describe all information storage groups in brief.**

Every computer contains several types of devices to store the instructions and data required for its operation. These storage devices plus the algorithms implemented by hardware and/or software needed to manage the stored information from the memory system of the computer.

**Memory Access Characteristics**

A CPU should have rapid, uninterrupted access to the external memories where its programs and the data they process are stored so that the CPU can operate at or near its maximum speed. Unfortunately, memories that operate at speed comparable to processor speeds are expensive and generally only very small systems can afford to employ a single memory using just one type of technology.

**Memory types:** The information storage components or a computer can be placed in four groups as shown in figure 3.1.

**Fig. 3.1: Conceptual Organization of a multilevel memory system in a computer. CPU Registers:** These high speed registers in the CPU serve as the working memory for temporary storage of instructions and the data as it is processed. A capacity of data words is typical of a register file and each register can be accessed, that is, read from or written into, within a single clock cycle (a few nanoseconds).

**Main (primary) memory:** It stores programs and data that are in active use. Storage locations in main memory are addressed directly by the CPU’s load and store instructions. While an IC technology similar to that of a CPU register file is used, access

M.Sc. Semester – II (Computer Science) 1

Computer Architecture and Organization

is slower because of main memory’s large capacity and the fact that it is physically separated from the CPU. Main memory capacity is typically between 1 and 210 megabytes, where a megabyte also denoted by 1 MB, is 220 bytes, and 210 MB = 230 bytes is referred to as a gigabyte (1 GB).

**Secondary Memory:** This memory type is much larger in capacity but also much slower than main memory. Secondary memory stores system programs, large data files, and the like that are not continually required by the CPU. It also acts as an overflow memory when the capacity of the main memory is exceeded. Information in secondary storage is considered to be online but is accessed indirectly via input/output programs that transfer information between main and secondary memory. Representation technologies for secondary memory are magnetic hard disks and CD-ROMs, both of which have relatively slow electromechanical access mechanisms. Storage capacities of many gigabytes are common, while access times are measured in milliseconds.

**Cache:** Most computers now have another level of IC memory known as cache memory, which is positioned logically between the CPU registers and main memory, but with an access time of one to three cycles. The cache is much faster than main memory because some or all of it can reside on the same IC as the CPU. Caches are essential components of high-performance computers that aim to make CPI<=1.

The goal of every memory system is to provide adequate storage capacity with an acceptable level of performance and cost. This goal can achieve by employing several memory types with different cost/performance ratios that are organized to provide a high average performance at a low average cost per bit.

**Performance and Cost:** The computer architect can choose from a bewildering variety of memory devices that employ various electronic, magnetic and optical technologies and offer many cost/performance. All memories are based on just a few physical phenomena and organizational principles.

The most meaningful measure of the cost or a memory device is the purchase price to the user of a complete unit. The price should not include only the cost of the information storage medium itself but also the cost of the peripheral equipment (access circuitry) needed to operate the memory. Let C be the price in dollars of a complete memory system with S bits of storage capacity. So, the cost c of the memory is as follows

M.Sc. Semester – II (Computer Science) 2

Computer Architecture and Organization

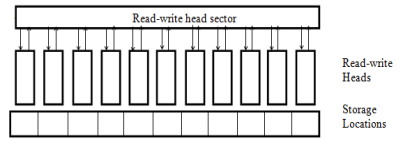
c = (C/S) dollars/bit

The performance of an individual memory device is primarily determined by the rate at which information can be read from or written into the memory. A basic performance measure is the average time to read a fixed amount of information, for instance, one word, from the memory. This parameter is called the read access time or simply the access time, of the memory and is denoted by ����. The write access time is defined similarly, but not always equal to the read access time. The time depends on the physical nature of the storage medium and on the access mechanism used. It is calculated from the time the memory receives a read request to the time at which the requested information becomes available at the memory’s output terminals.

Clearly, low cost and short access time are desirable memory characteristics. But they also tend to the incompatible. Memory units with fast access are expensive, while low cost memories are slow.

**Access Modes:** A fundamental characteristics of a memory is the order or sequence in which information can be accessed. If storage location can be accessed in any order and access time is independent of the location being accessed, the memory is known as random access memory (RAM). IC(semiconductor) memories are generally of this type. Memories whose storage location can be accessed only in a certain predetermined sequence are called serial access memories. Magnetic disks and tapes as well as optical memories like CD-ROM employ serial-access methods.

Each storage location in a RAM can be accessed independently of the other locations. There is a separate access mechanism, or read write “head” for every location as given in figure 3.2.

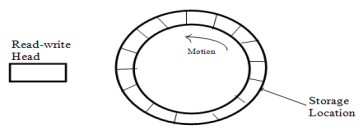
**Fig. 3.2: Conceptual model of a random access memory.**

M.Sc. Semester – II (Computer Science) 3

Computer Architecture and Organization

In serial memories, the access mechanism is shared by the storage locations and must be assigned to different locations at different times by moving the stored information, the read-write head or both. Many serial-access memories operate by continually moving the storage locations around a closed path or track, as given in figure 3.3. a particular location can be accessed only when it passes the fixed read-write head. Hence the time to access a particular location depends on its position relative to the read

write head when the memory receives an access request.

**Fig. 3.3: Conceptual model of a sequential access memory.**

**Memory Retention:** The method of writing information into a memory can be permanent or temporary. Memories whose contents cannot be altered online are ROM. A Rom is a non-erasable storage device. ROMs are widely used to stored control programs such as micro-programs. Compact Disk (CD) ROMs are a class of non-erasable secondary memory devices developed in the 1980s that employ an optical (laser) read

write mechanism. A standard (12 cm diameter) CD-ROM has a capacity of about 600 MB and is used to store large program and data files. Semiconductor ROMs whose contents can be changed off-line and with some difficulty are called PROMs. Programmable CDs are referred to as CD-recordable (CD-R) disks.

Memories in which reading or writing can be done are called RAM to differentiate them from ROMs. All memories used for temporary storage purposes are read-write memories.

**Permanence of Storage:** In some memory devices, The stored information is lost over a period of time unless corrective action is taken. Three characteristics of memory that destroy information are;

i) Destructive readout (DRO)

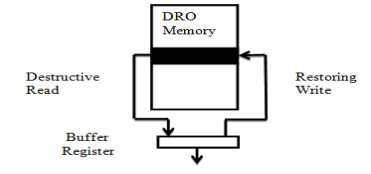
M.Sc. Semester – II (Computer Science) 4

Computer Architecture and Organization

ii) Dynamic storage

iii) Volatility

In some memories, the method reading memory destroys the stored information; this phenomenon is called destructive readout (DRO). Memories in which reading does not affect the stored data have a non-destructive readout (NDRO). In DRO memories, each read operation must be followed by a write operation that restores the memory’s original state. This restoration is carried out automatically using a buffer register as shown in figure 3.4. The read transfer the word at the addressed (shaded) location to the buffer register where it is available to external devices. The content of the buffer are automatically written back into the original location.

**Fig. 3.4: Memory Restoration in a Destructive Readout (DRO) memory**

Certain memory devices have the property that a stored 1 tends to become a 0, or vice-versa, due to some physical decay process. For example, in some IC memories an electric charge is a capacitor represents a stored 1; the absence of a stored charge represents a 0. A stored charge tends to leak away, causing a loss of information unless the charge is restored by a process called refreshing. Memories that require periodic refreshing are called dynamic storage.

Another physical process that can destroy the content of the memory is the removal or failure of its power supply. A memory is volatile, if the loss of power destroys the stored information. Following table shows the characteristics for some important memory technologies.

M.Sc. Semester – II (Computer Science) 5

Computer Architecture and Organization

**Access time, Cycle time and Data transfer rate:-** Access time ���� as the time between the receipt of a read request signal by a memory and the delivery of the requested information to its output terminals. Some DRO and dynamic memories cannot initiate a new access until a restore or refresh operation has been carried out. Therefore, the minimum time that must elapse between the start of two consecutive access operations cannot be greater than����. This elapsed time is called the cycle time ���� of the memory.

The maximum amount of information that can be transfer to or from the memory per unit time is the data transfer rate or bandwidth ���� and is measured in bits or words per second. If w is the number of bits that can be transfer simultaneously to or from the memory, that ���� = w/���� bits per second.

**Characteristics of some common memory technologies:**

| **Technology** | **Primary**  **Storage**  **Medium** | **Access**  **Mode** | **Operations** | **Performance** | **Typical**  **Access**  **Time** ���� |
| --- | --- | --- | --- | --- | --- |
| Bipolar semi conductor | Electronic | Random | Read/Write | Volatile | 10 ns |
| Metal oxide semiconductor | Electronic | Random | Read/Write | Volatile | 50 ns |
| Magnetic  (hard) Disk | Magnetic | Semi  random | Read/Write | Non volatile | 10 ms |
| CD-ROM | Optical | Semi  random | Read Only | Non volatile | 100 ms |
| Magnetic Tape | Magnetic | Serial | Read/Write | Non volatile | 1 s |

**MEMORY ARRAY ORGANIZATION**

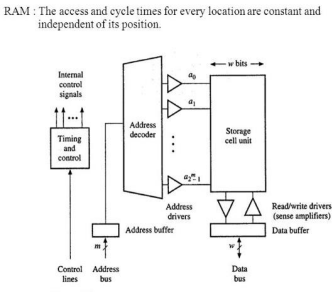
In RAMs, each storage location can be accessed independently with fixed access and cycle times that are independent of the position of the accessed location to be accessed is transferred via the address bus to the Ram’s address buffer. The address is then processed by the address decoder, which select the required location in the storage cell unit. A control line indicates the type of access to be performed. If a read operation (load0 is requested, the contents of the address location are transferred into the storage cell unit to the data buffer and from there to the data bus. If a write (store0 is requested, the word to be stored is transferred to the data bus to the selected location in the storage unit. It is not desirable or necessary to permit simultaneous reading and writing, the input and output data buses are often combined into a single, bidirectional data bus.

M.Sc. Semester – II (Computer Science) 6

Computer Architecture and Organization

The storage unit is made up of many identical 1-bit memory cells and their interconnections. The actual number of lines connected to the cell and their functions depend on the memory technology and the addressing scheme to use. Each cell is connected to a set of data, address and control signals. One physical line can perform several logical functions; for example, it can serve as both address and data line in each line connected to the storage cell unit, a driver is finding which acts as either an amplifier or a transducer of physical signals. Thus figure 3.5 shows several sets of drivers for the address and data lines. The drivers, decoders and control circuit from the access circuitry of the RAM and can have a significant impact on the total size and cost of the memory.

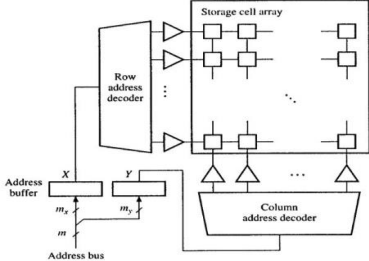
A RAM’s storage cells are physically arranged into regular arrays to reduce the cost of the connections between the cells and the access circuitry. The memory address is partitioned, into d components so that the address ���� of cell ���� becomes a d-dimensional vector (����,1, ����,2, …, ����,��) = ����. Each of the d parts of the address word goes to a separate address decoder and a separate set of address drivers. A cell is selected by simultaneously activating all d of its address lines. A memory unit with this kind of addressing is said to be d-dimensional. Thus the basic RAM of above figure is one-dimensional (1-D).

**Fig. 3.5: One-Dimensional (1-D) Random Access Memory Unit**

M.Sc. Semester – II (Computer Science) 7

Computer Architecture and Organization

The most common RAM organization is the two-dimensional (2-D) or row column scheme shown in fig 3.6. Where the data and control circuits are omitted there the m-bit address word in divided into two parts, X and Y, consisting of mx and my bits, respectively. The cells are arranged in a rectangular array of Nx ≤ 2mx, rows and Ny ≤ 2my columns. So the total number of cell is N=Nx Ny. A cell is selected by the coincidence of signals applied to its X and Y address lines. The 2-D organization requires much less access circuitry than a 1-D organization for the same storage capacity. For example if Nx= Ny=√N the number of address drivers needed is 2 √N, whereas the 1-D RAM of fig 3.4 has N=Nx Ny address drivers. Instead of a single one out of N addresses decoders. In addition the 2-D organization is a good match for the inherently two dimensional layout structures allowed by VLSI technology.



**Fig. 3.6: Two dimensional (2-D) random-access memory addressing scheme**

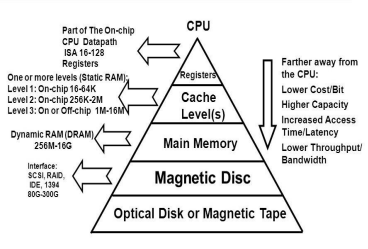
The memory unit is an essential component in every digital computer; it is needed for storing programs and data. A very small computer with a limited application may be able to fulfill its intended task without the need of additional storage capacity. Most general purpose computers would run efficiently if they equipped with additional storage beyond the capacity of main memory.

M.Sc. Semester – II (Computer Science) 8

Computer Architecture and Organization

Memory unit that communicates directly with the CPU is called main memory. Devices that provide backup storage are called auxiliary memory. The most common auxiliary memory devices used in computer are magnetic tapes and disks. They are used for storing system program, large data files and other backup information. Only program and data currently needed by the processor reside in main memory. All other information is stored in auxiliary memory and transferred to main memory when needed.

The total memory capacity of a system can be visualized as being hierarchy of components. The memory hierarchy system consists of a storage devices employed in a computer system from the slow but high capacity auxiliary memory to relatively faster main memory, to an even smaller and faster cache memory, accessible to the high speed processing logic. Memory hierarchy structure is shown in figure 3.7. At the top are the CPU register, which can be accessed at full CPU speed. Next comes the cache memory, which is currently on the order of 32KB to a few megabytes. Main memory is next, with sizes currently ranging from 16MB for entry-level systems to tens of gigabytes at the high end. After that come magnetic disks the current work for permanent storage finally magnetic tape and optical disks for archival storage.

**Fig. 3.7: A five level memory hierarchy**

As we move down the hierarchy, three key parameters increase. First the access time gets bigger. CPU registers can be accessed in a few nanoseconds (10-9) .Cache

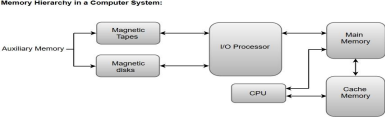
M.Sc. Semester – II (Computer Science) 9

Computer Architecture and Organization

memories take a small multiple of CPU registers. Main memory accesses are typically a few tens of nanoseconds. Disk access time is at least 10 msec (10-6) and tape or optical disk access can be measured in seconds if the media have to be fetched and inserted into a drive.

Second, the storage capacity increase as we go downwards. CPU registers are good for 128 bytes, caches for a few megabytes, main memories for tens to thousands of megabytes magnetic disks for a few gigabytes to tens of gigabytes. Tapes and optical disks are usually kept off-line, so their capacity is limited only by the owner’s budget.

Third, the number of bits gets per dollar spent increases down the hierarchy. Although the actual prices change rapidly main memory is measured in dollar/gigabytes, magnetic disk storage in pennies/megabytes and magnetic tape in dollar/gigabyte or less.

All these memory units are interconnected as shown in following figure 3.8 **Fig. 3.8: Memory hierarchy**

While the I/O processor manages data transfer between auxiliary memory and main memory, the cache organization is concerned with transfer of information between main memory and CPU, thus each is involved with a different level in memory hierarchy system. The reason for having two or three levels of memory hierarchy is economy. As the storage capacity of the memory increases, the cost per bit for storing binary information decreases and access time of memory becomes higher. The auxiliary memory has a large storage capacity, is capacity is relatively inexpensive but has low access speed compared to main memory.

**INTERLEAVING**

Main memories are comprised of a series of semiconductor memory chips. A number of these chips like caches from a bank. Multiple memory banks can be connected together to form an interleaved (or parallel) memory system. Since each bank can service

M.Sc. Semester – II (Computer Science) 10

Computer Architecture and Organization

a request an interleaved memory system with K banks can service K requests simultaneously increasing the peak bandwidth of the memory system to K times the band width of a single bank. In most interleaved memory system the number of banks is a power of two that is K=2k. An n-bit memory word address is broken into two parts: a k

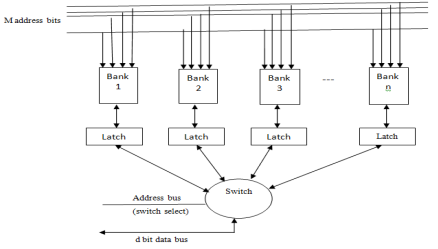
bit bank number and m-bit address of a word within a bank. Though the k bits used to select a bank number could be any k bits of the n-bit word address typical interleaved memory system use the low order k address bits to select the bank number; the higher order m=n-k bits of the word address are used to access a word in selected bank. An interleaved memory system which uses the low order k bits to select the bank is referred to as a low order or a standard interleaved memory.

There are two ways of connecting multiple memory banks: simple interleaving and complex interleaving. Sometimes simple interleaving is also referred to as interleaving and complex interleaving as banking.

Figure 3.9 shows the structure of a simple interleaving memory system m address bits are simultaneously supplied to every memory bank. All banks are also connected to the same read/ write control line. For a read operation the banks start the read operation the banks start the read operation and deposit the data in their latches. Data can then be read from the latches one by one by appropriately setting the switch. Meanwhile the banks could access again to carry out another read or write operation. For a write operation the latches are loaded, one by one. when all the latches have been written , their contents can be written into the memory banks by supplying m bits or address( they will be written into the same word in each of the different banks) . In a simple interleaved memory all banks are cycled at the same time each bank starts and completes its individual operations at the same as every other bank: a new memory cycle can start (for all banks) once the previous cycle is complete.

M.Sc. Semester – II (Computer Science) 11

Computer Architecture and Organization



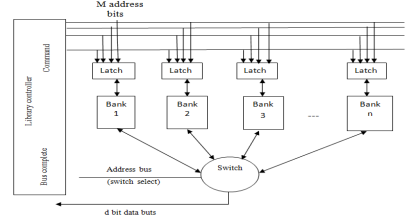
**Fig. 3.9: Structure of a simple Interleaved Memory System**

One use of a simple interleaved memory system is to black up a cache memory to do so, the memory usable to read block of contiguous word (a cache block) and supply them to the cache, if the low order k bits of the address are used to select the bank number then consecutive word of the block reside in different banks they can all be read in parallel and supplied to the cache one by one . If some other address bits are used for bank selection, then multiple words from the block might fall in the same memory bank, requiring multiple accesses to the same bank to latch the block.

Figure 3.10 shows the structure of a complex interleaved memory system in such a system each bank is set up to operate on its own, independent of the other banks operation in the example bank1 could carry out a read operation on a particular memory address, while bank 2 carries out a write operation on a completely related memory address. Complex interleaving is accomplished by providing an address latch and a read/ write command line for each bank. The memory controller handles the overall operation of the interleaved memory. The processing unit submits the memory request to the memory controller, which determines the bank that need to be accessed. The controller then determines if the bank is busy (by monitoring a busy line for each bank). The controller holds the request if the bank busy submitting it later when the bank is available

M.Sc. Semester – II (Computer Science) 12

Computer Architecture and Organization

to accept the request. When the bank responds to a read request the switch is set by the controller to accept the request from the bank and forward it to the processing unit. A typical use of a complex interleaved memory system is in a vector processor. In a vector processor the processing units operate on a vector for example apportion of a row or a column of a matrix. If consecutive elements of a vector are present in different memory bank then the memory system can sustain a bandwidth of one element per clock cycle. By arranging the data suitably in memory and using standard interleaving (for example strong the matrix in row major odder will place consecutive element in consecutive memory bank) the vector can be accessed at the rate of one element per clock cycle as long as the number of banks is greater than the bank busy time. **Figure 3.10- structure a complex interleaved system**

Memory system that are built for current machines vary widely the price and purpose of the machine being the main determinant of the memory system design. The actual memory chips which are the components of the memory system are generally commonly part built by a number of manufacturers. The major commodity DRAM manufactures. Include (but are certainly not limited to). Hitachi, Fujitsu, LC, Selicon, NEC OK!, Samsung, Texas Instruments, and Toshiba.

The low end of the price/performance spectrum is the personal computer presently simplified by Intel Pentium system. Three of the manufacturers of Pentium compatible chip sets (which include the memory controllers) are Intel, OPTI and VLSI

M.Sc. Semester – II (Computer Science) 13

Computer Architecture and Organization

Technologies. Their controllers provide for memory system that are simply interleaved all with minimum bank depths of 256 KB and maximum system sizes of 192 MB , 128MB and 1GB respectively.

Both higher end personal computers and workstations tend to have more main memory than the lower end system although they usually have similar upper limits. Two examples of such system are workstations built with the DEC Alpha-21164 and servers built with the Intel Pentium Pro. The Alpha system using the 21171 chip set are limited to 128 MB of main memory using 16 MB DRAMs, although they will be expandable to 512 MB when 64 MB DRAs are available. Their memory system is eight ways simply interleaved providing 128 bits per DRAM access. The Pentium Pro system support slightly different features. The 82450KX and 82450GX chip sets include memory controller that allow reads to bypass writes (Performing writes when the memory banks are Idle). These controllers can also buffer eight outstanding requests simultaneously. The 84250KX controller permits one or two-way interleaving, and up to 256 MB of memory when 16 MB DRAMs are used. The 82450GX chip set is more aggressive, allowing up to four separate (complex-interleaved) memory controllers, each of which can be up to four-way interleaved and have up to 1 GB of memory.

**CACHE MEMORY (65-67 Tanenbaum)**

The cache memory is the fastest memory placed between processor and main memory (primary memory). It is a semiconductor memory consists of static RAM. Its access time is 10 nanoseconds and that of main memory is 50 nanoseconds. Due to which it is very faster than main memory. The capacity of cache memory is about used by the CPU, due to which it helps CPU to access the data more quickly which improves the performance of the CPU as shown in following figure.

CPU

Processor

Cache Memory

Primary Memory RAM

Secondary Memory (HDD, etc)

**Fig. 3.11 Cache memory organization**

M.Sc. Semester – II (Computer Science) 14

Computer Architecture and Organization

Suppose that there is no cache memory then the speed of main memory is very less as compared to that of CPU then the CPU has to wait for the memory until performance is reduced.

In early period, the cache memory was to be implemented out of processor on main board just like RAM. But now a day, it is fabricated inside the CPU itself to decrease the CPU wait time.

A cache memory also requires the cache controllers which are available in the form of IC. When the CPU sends an address of instruction code or data, the cache memory or not if the instruction code or data is available in cache memory then these cache controller, enables cache memory and the required data or instruction are passed over to CPU for further process. And if not then cache controller enables the main memory controllers and sends that address code to it. The main memory then outputs the address code or data on data bus to the CPU as well as to cache memory.

Another common arrangement of the CPU, cache and main memory is illustrated in figure 3.12

Main Memory CPU

Cache

Bus **Fig. 3.12 Logical connection between cache and main memory**

Cache design is an increasingly important for high-performance CPUs. One issue is cache size. The bigger the cache, the better it performs, but also the more it costs. A second issue is the size of cache line. A 16 KB cache can be divided up into 1K lines of 16 buts, 2K lines of 8 bytes and other combination. A third issue is organization of cache because cache can be placed anywhere. A fourth issue is whether instructions and data are kept in the same cache or different ones, If instruction and data are kept in same cache the it is known as unified cache. It is a simpler design and automatically balances instruction fetches against data fetches. But now a day, a new concept of cache known as split cache is used. In split cache, instructions are kept in one cache and data in the other.

M.Sc. Semester – II (Computer Science) 15

Computer Architecture and Organization

The memory cache works as follows;

1. The CPU asks for instruction or data stored in address “a”.

2. Since the contents from address “a” are not inside the memory cache, the CPU has to fetch it directly from RAM.

3. The cache controller loads a line (typically 64 bytes) starting a address “a” into the memory cache. This is more data than the CPU requested, so if the program continues to run sequentially (i.e. asks for address “a+) the next instruction or data the CPU will ask will be already loaded in the memory cache.

4. A circuit called pre-fetcher loads more data located area this line, i.e. states loading the contents from address on into the cache. For example, Pentium 4 CPUs have 256-byts pre-fetcher, so it loads the next 256 bytes after the line already loaded into the cache. If program always run sequentially, the CPU would never need to fetch data directly from the RAM memory; except to load the very first instruction because the instruction and data required by the CPU would always be inside the memory cache before the CPU would ask for them.

However, programs do not run always sequentially. From time to time they jump from one memory position to another. The main challenge of the cache controller is trying to use what address the CPU will jump loading the content of this address into the memory cache before the CPU asks for it in order to avoid the CPU having to go to the RAM. This is called branch predicting and all modern CPUs have this feature.

**VIRTUAL MEMORY**

**Virtual Memory**: The real or actual memory. (Physical memory) is the actual main memory provided in the system. It is directly controlled by the CPU. The address of the location of the physical memory is called physical address.

The technique which allows a program to use main memory more than what a computer has is knows as **virtual memory techniques**. It gives the physically available in computer. Here whole content of program is not brought in main memory only a content of program or addresses or data which is to he currently executed by CPU is brought to main memory and other parts is in secondary memory. When the execution of data available in main memory is over, is is brought back to secondary memory and further contents are transferred to main memory.

M.Sc. Semester – II (Computer Science) 16

Computer Architecture and Organization

Thus the virtual memory in simple words is a part of secondary memory (hard disk) used as a main memory to improve the overall performance of machine. This virtual memory is implemented by the operating system itself and user needs no worry about it. The diagrammatic presentation is give in figure 3.13

Processor Cache Main

Virtual Memory

Memory Secondary

Memory

**Fig. 3.13: Virtual memory**

The term virtual memory is applied when the main and secondary memories appear to a user program like a single, large and directly addressable memory. Traditionally, there are three reasons for using virtual memory.

• To free user program from the need to carry out storage allocation and to permit efficient sharing of the available memory space different users.

• To make programs independent of the configuration and capacity of the physical memory present for their execution; for example, to allow searches overflow into secondary memory when he capacity of main memory is exceeded.

• To achieve the very low access time and cost per bit that is possible with a memory hierarchy.

Virtual memory is a HDD space that uses some portion of it as the memory. It is used to store application data and instruction that is currently not needed to be process by the system.

During the program loading process, the system will copy the application data and its instruction from the HDD into the main memory (system memory). Therefore the system can use its resources such as CPU to process and execute it. Once the system memory gets filled up, the system will start moving some of the data and instruction that don’t need to process anymore into the Virtual Memory until those data and instruction need to process again. So the system can call the next application data and instruction and copy it into the main memory in order for the system to process the rest and load the

M.Sc. Semester – II (Computer Science) 17

Computer Architecture and Organization

program. When the data and instruction that is in the Virtual Memory needs to process again, the system will first check the main memory for its space. If there is space, it will simply swap those into the main memory. If there is not any space left for the main memory, the system will first check the main memory and move any data and instruction that don’t need to be process into the Virtual Memory. And then swap the data and instruction that need to be process by the system from the Virtual Memory into the main memory.

Having too low of Virtual Memory size or large Virtual Memory size (meaning the size that is above double of the system memory) is not a good idea. If you can see the Virtual Memory too low, then the OS will keep issuing an error message that states either not enough memory or Virtual to low. This is because some portion of the system memory is used to store the OS Kernel, and it requires to be remaining in the main memory all the time. Therefore the system needs to have a space to store the not currently needed process data and instruction when the main memory gets filled up. If you set the Virtual Memory size too large to support the intensive application, it is also not a good idea. Because it will create the performance lagging and even it will take the HDD free space. The system needs to transfer the application data and instruction back and forth between the Virtual Memory and the System Memory. Therefore that is not a good idea. The ideal size for the Virtual Memory is the default size of Virtual Memory and it should not be exceed the value of the triple size of system memory.

To determine how much virtual memory you need, since the user's system contains the different amount of RAM, itis based on the system. By default, the OS will set the appropriate size for Virtual Memory. The default and appropriate size of Virtual Memory is:

<Amount\_Of\_System\_Memory>\* 1.5 =

<Default\_Appropriate\_Size\_Of\_Virtual\_Memory>

For example, if your system contains 256 MB of RAM, you should set 284 MB for Virtual memory.

256 MB of RAM (Main Memory) \* 1.5 = 384 MB for Virtual Memory

M.Sc. Semester – II (Computer Science) 18

Computer Architecture and Organization

If you would like to determine how much the Virtual Memory is for your system and/or would like to configure and add more virtual memory, follow the procedure that is shown below. The following procedure is based on windows XP Professional. 1. Go to right-click My Computer and choose Properties.

2. In the System Properties dialog box, go to Advanced tab.

3. Click Settings button that is from the Performance frame.

4. Once the Performance Options shows up on the screen, go to advanced tab. 5. Under the Advance tab, click the Change button from the Virtual Memory frame to access to the Virtual Memory setting.

Then the Virtual Memory dialog box appears on the screen. In there, you are able to check how much the Virtual Memory you set. If you would like to modify size of Virtual Memory, follow the procedure that is shown below.

1. In there, select the drive letter that is used to install the Operating System. 2. Choose the option that says, "Custom Size:"

Once we choose that option, the setting for Initial Send Maximum Size become available for you to set Size (MB) means the actual size of Virtual Memory and Maximum Size (MB) means the maximum size of Virtual Memory that is allowed to use.

Let's say if your system contains 512 MB of RAM, then the ideal setting for the Virtual Memory is as follows

Initial Size (MB); 768

Maximum Size (MB): 1500

Once you are happy with that Virtual Memory size, click the Set button from Paging file size for selected drive to apply the setting for the Virtual Memory size. Then click the OK button to apply the setting.

That's where you can manage and configure for the size of Virtual Memory. **VIRTUAL MEMORY IMPLEMENTATION**

There are two popular methods in virtual memory implementation.

1. Paging

2. Segmentation

In paging, the system software divides the program into pages. This is not known to the programmer. All the pages in a program are of same size. Generally, user program

M.Sc. Semester – II (Computer Science) 19

Computer Architecture and Organization

is split into multiple pages of equal size. At a given time, only some pages are in the main memory. The remaining pages of different programs are stored on a hard disk which serves as the virtual memory. Thus paging is a fixed size block of program. It is also known as demand paging.

In segmentation, OS organizes machine program into different segments. It is a logical division of the program into meaningful modules. Different segments are not of same size. It is a variable-size block of program. It is also known as demand segmentation.

**Principle of Operation**

The purpose of virtual memory is to enlarge the address space, the set of addresses a program can utilize. For example, virtual memory might contain twice as many addresses as main memory all of virtual memory, therefore, would not be able to fit in main memory all at once. Nevertheless, the computer could execute such a program by copying into the main memory those portions of the program needed at any given point doing execution.

To facilitate copying virtual memory into real memory the operating system divides virtual memory into pages each of which contain a fixed number of addresses. Each page is stored on a disk until it is needed. When the page is needed the operating system copies it from disk to main memory, translating the virtual addresses into real addresses.

Addresses generated by programs are virtual addresses. The actual memories have physical addresses. A piece of hardware called a memory management unit 24 (MMU) translates virtual address physical addresses at run-time. The process of translating virtual addresses to physical addresses is called mapping..The copying of virtual pages from two main memory is known as paging or swapping.

**Virtual Memory Management:** This section provides the description of how the Virtual memory manager provides virtual memory. It explains how the logical and physical address spaces are mapped to one another and when it is required to use the services provided by the Virtual Memory Manager.

Before going cells into the details of the management of the virtual memory, let us see the function of the virtual memory manager. It is responsible to:

M.Sc. Semester – II (Computer Science) 20

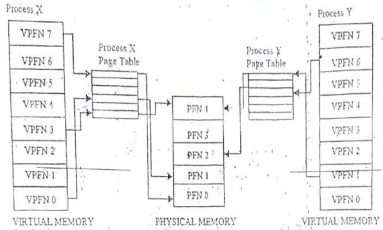
Computer Architecture and Organization

• Make port of the logical address space resident in physical RAM.

• Make portion of the logical address space immovable in physical RAM. • Map logical to physical addresses.

• Defer execution of application defined interrupt code until a safe time. As the processor executes a program it reads an instruction from memory and decodes it. In decoding the instruction it may need to fetch or store the contents of a location of operands in the memory. The processor then executes the instruction and moves onto the next instruction in the program. In this way the processor is always accessing memory either to fetch instructions or to fetch and store data. In a virtual memory system all of these addresses are virtual addresses and not physical addresses. These virtual addresses are converted into physical addresses by the processor based on information held in a set of tables maintained by the operating system.

To make this translation easier, virtual and physical memory are divided into small blocks called **pages**. These pages are all of the same size. Each of these pages is given a unique number; the page frame number (PFN) as shown in the Figure 3.14

**Fig. 3.14: Model of virtual to physical address mapping.**

M.Sc. Semester – II (Computer Science) 21

Computer Architecture and Organization

In this paged model, a virtual address is composed of two parts, an offset and a virtual page frame number. `If the age size 4 Kbytes, bits 11:0 of the Virtual address contain the offset and bits 12 and above are the virtual page frame number. Each time the processor encounters a virtual address it must extract the offset and the virtual page frame number. The processor must translate the virtual page frame number into a physical one and then access the location at the correct offset into that physical page. To do this the processor uses **page tables.**

The Figure 3.14 shows the virtual address spaces of two processes, process X and process Y, each with their own page tables. These page table map each processes virtual pages into physical pages in memory. This shows that process X virtual page frame number virtual is mapped into memory in physical frame number 1 and that process Y’s virtual page frame number 1 is mapped into physical page frame number 4. Each entry in the theoretical page table contains the following information :

• **Valid flag:** this indicates if this page table entry is valid.

• **PFN:** The physical page frame number that this entry is describing. • **Access control information:** this describes how the page may be used. Can it be written to? Does it contain executable code?

The page table is accessed using the virtual page frame number as an offset. Virtual page frame 5 would be the 6th element of the table (0 is the element) To translate a virtual address into a physical one, the processor must first work out the virtual addresses page frame number and the offset within that virtual page. By making the page size a power of 2 this can be easily done by masking and shifting, Looking again at the Figure1 and assuming a page size of 0x2000 bytes (which is decimal 8192) and an address of 0x2194 in process Y's virtual address space then the processor would translate that address into offset 0x194 into virtual page frame number 1. The processor use the virtual page frame number as an index into the processes page table to retrieve its page table entry. If the page table entry at that offset is valid, the processor takes the physical page frame number from the entry. If the entry is invalid, the process has accessed a non-existent area of its virtual memory. In this case, the processor cannot resolve the address and must pass control to the operating system so that it can fix things up.

M.Sc. Semester – II (Computer Science) 22

Computer Architecture and Organization

**DEMAND PAGING**

In a multiprogramming system, memory, is divided into a number of fixed-size or variable-sized partitions or regions that are allocated to running processes. For example: a process needs m words of memory may run in a partition of n words where n >m. The variable size partition scheme may result in a situation where memory is not contiguous, but fragmented into many scattered Blocks. We distinguish between internal fragmentation and external fragmentation. The difference (n-m) is called internal fragmentation, memory that is internal to the partition but is not being used. If a partition is unused and available, but too small to be used by any waiting process, then it is accounted for external fragmentation. These memory fragments cannot be used.

In order to solve problem, we can either compact the memory making large free memory blocks, or implement paging scheme which allows a programs memory to be non-contiguous, thus permitting program to be allocated to physical memory.

Physical memory is divided to fixed size blocks called frames. Logical memory is also divided into blocks of same, fixed size called pages. When a program is to be executed, its pages are loaded into any available memory frames from the disk. The disk also divided into fixed sized blocks that are the same size as frames sized as the memory frames.

A very important aspect of paging is the clear separation between the user’s view of memory and the actual physical memory. Normally, a user believes that memory is one continuous space containing only his/her program. In fact, the logical memory is scattered through the physical memory that also contains other programs. Thus, the use can work correctly with this own view of memory because of the address translation or address mapping. The address mapping, which is by the operating system and transparent to users, translates logical memory addresses into physical addresses.

Because the operating system is managing the memory, it must be sure about the nature physical memory, for example: which frames are available, which are allocated; how many total frames there are, and so on. All these parameters are kept in data structure called frame table that has one entry for each physical frame of memory indicating whether it is free or allocated, and if allocated, to which page of which process.

M.Sc. Semester – II (Computer Science) 23

Computer Architecture and Organization

As there is much less physical memory than virtual memory the operating -system must be careful that it does not use the physical memory inefficiently. One way to save physical memory is to load only virtual pages that are currently being used by the executing program: For example: a database-program may be run to query a database. In this case not the entire-database needs to be loaded into memory, just those data records that are being examined: Also, if the database query is a search query then it is not necessary to load the code from the database that deal with adding new records. This technique of only loading virtual page memory as they are accessed is known as demand paging.

When a process attempts to access a virtual address that is not currently in memory the CPU cannot find a page table entry for the virtual page referenced. For example, in Figure 1 is no entry in Process; X's page table for virtual PFN 2 and so if Process X attempts to read from an address within virtual PFN 2 the CPU cannot translate the address into a physical one. At this point the CPU cannot cope and needs the operating system to fix things up. It notifies the operating system that a page fault has occurred and the operating system makes the process wait whilst it fixes things up.

The CPU must bring the appropriate page into memory from a while until the page has been fetched. If there are other processes that could run then the operating system will select one them to run. The fetched page is written into a free physical page frame and an entry for the virtual PEN is added to the processes page table. The process is then restarted at the point where the memory fault occurred. This time the virtual memory access is made the CPU can make the address translation and so the process continues to run. This is known as demand paging and occurs when the system is busy but also when an image is first loaded into memory. This mechanism means that a process can execute an image that only partially resides in physical memory at any one time.

**Page Replacement Policies-** Basic to the implementation of virtual memory is the concept of demand paging. This means that the operating system and not the programmer, controls the swapping of pages in and out of main memory as the active processes require them. When a process needs a non-resident page, the operating system

M.Sc. Semester – II (Computer Science) 24

Computer Architecture and Organization

must decide which resident page is to be replaced by the requested page. The part of the virtual memory which this decision is called the replacement policy. There are many approaches to the problem of deciding which page is to replace but the object is the same for all the policies that select the page that will not be referenced again for the longest time. A few page replacement policies are described below.

**First In First Out (FIFO):** The First In First Out (FIFO) replacement policy chooses the page that has been in the memory the longest to be the one replaced. Normally, as the number of page frames increases, the number of page faults should decrease. **Second Chance (SC)-** The Second Chance (SC) policy is a slight modification of FIFO in order to avoid the problem of replacing a heavily used page. In this policy, a reference b R is used to keep track of pages that have been recently referenced. This bit is set to 1 each time the page is referenced. Periodically, all the reference bits are set to 0 by the operating system to distinguish pages that have not been referenced recently from those that have been. Using this bit, the operating system can determine whether old pages are still being used (i.e., R = 1). If so, the page is moved to the end of the list of pages, and its load time is updated as though it had just arrived in memory. Then the search continues. Thus heavily accessed pages are given a "second chance."

**Least Recently Used (LRU)-** The Least Recently Used-(LRU) replacement policy chooses to replace the page which has not been referenced for the longest time. This policy assumes the recent past will approximate the immediate future. Theoperating system keeps track of when each page was referenced by recording the time of reference or by maintaining a stack of references.

**Optimal Algorithm (OPT) -** Optimal algorithm is defined as replace the page that will not be used for the longest period of time, It is optimal in the performance but not feasible to implement because we cannot predict future time.

**DEMAND SEGMENTATION**

Programs generally divide up their memory by function. Some memory holds instructions, some static, data, some dynamically allocated data, some execution frames. All of these memory types have different protection growth and sharing requirements. In

M.Sc. Semester – II (Computer Science) 25

Computer Architecture and Organization

the monolithic memory allocation of classic Virtual Memory system this model isn’t well supported.

Segmentation addresses this by providing multiple sharable, protectable, grow able address spaces that processes can access.

In pure segmentation architecture, segments are allocated like variable partitions, although the memory management hardware is involved in decoding addresses. Pure segmentation addresses replace the page identifier in the virtual address with a segment identifier and find the proper segment (not page) to which to apply the offset.

The segment table is managed like the page table, except that segments explicitly allow sharing. Protections reside in the segment descriptors; and can use keying or explicit access control lists to apply them.

Of course, the segment name space must be carefully managed and thus OS must provide a method of doing this. The file system cannot come to the rescue here a process can ask for a file to be mapped into a segment and have the OS return the segment register to use. This is known as memory mapping files. It is slightly different from memory mapping devices, because a one file system abstraction (a segment) is providing an interface to another (a file). Memory mapped files may be reflected into the file system or not and may be shared or not at the process’s discretion.

The biggest problem with segmentation is the same as with variable sized real memory allocation: managing variable sized partitions can be very inefficient, especially when the segments are large compared to physical memory. External fragmentation can easily result in expensive compaction when a large segment is loaded and swapping large segments (even when compaction is not required) can be costly.

**MULTILEVEL MEMORY**

A computer's memory units form a hierarchy of different memory types in which each member is in some sense subordinate to the next highest member of the hierarchy. This organization is to achieve a good trade-off between cost, storage capacity and performance for the memory system.

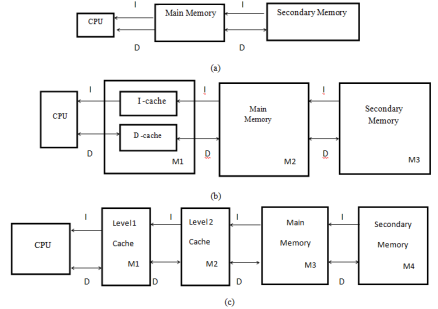
**General Characteristics:** Consider a general n-level system of n memory types (M₁, M2, Mn). Figure 3.15 shows some examples with n = 2, 3, and 4. Typicaltechnologies used in these hierarchies are semiconductor SRAMs for cache memory, semiconductor

M.Sc. Semester – II (Computer Science) 26

Computer Architecture and Organization

DRAMs for main memory and magnetic-disk for secondary memory. The two-level hierarchy of figure 3.15(a) is typical of early computers. Figure 3.15(b) adds a cache of a type called a split cache, since it has separate areas for storing Instructions (the l-cache) and data (the U-cache). The third example 3.15(e) has two cache levels, both of the non

split or unified type.

**Figure 3.15: Common Memory Hierarchies with (a) two, (b) three and (c) four level** The following relations normally hold between adjacent memory levels M1, andM1+1 in a memory hierarchy.

Cost per bit Cl+1 >Cl

Access time tAl<tAl+ 1

Storage capacity Sl< Sl+1

The differences in cost, access time and capacity between M1 and Ml+1 can be several orders of magnitude. Considerable system resources are devoted to shielding the CPU from these differences, so it almost always sees a very large and inexpensive memory space and rarely see an access time greater than that of Ml the first (highest) level of the memory, hierarchy.

M.Sc. Semester – II (Computer Science) 27

Computer Architecture and Organization

The CPU and other processors can communicate directly with My only; M₁ can communicate with M2, and so on. Consequently, for the CPU to read information held in some memory level M1 requires a sequence of data transfers of the form

Mi-1:=Mi; Mi-2:= Mi-1;Mi-3:= Mi-2; ……. M1 := M2; CPU:=M1.

During program execution the CPU produces a steady stream of memory addresses. At any time these addresses are distributed in some fashion throughout the memory hierarchy. If an address is generated that is currently assigned only to Mi where i ≠1, the address must be reassigned to M1, the level of the memory hierarchy that the CPU can access directly. This relocation of address involves the transfer of data between level M. and M1 - relatively slow process. For memory hierarchy to work efficiently, the addresses generated by the CPU should be found in M1 as often as possible. This approach requires that future addresses be to some extend predictable so that information can be transferred to M1 before it is actually referenced by CPU. If the desired data cannot be found in M1 then the program originating the memory request must be suspended until an appropriate reallocation of storage is made.

---------------------------------------------------------------------------------------------------------------------

M.Sc. Semester – II (Computer Science) 28