Memory organisation

Er. ExamRadar

Concept of Memory

We have already mentioned that digital computer works on stored programmed concept introduced by Von Neumann. We use memory to store the information, which includes both program and data.

Due to several reasons, we have different kind of memories. We use different kind of memory at different lavel.

The memory of computer is broadly categories into two categories:

Internal and external
Internal memory is used by CPU to perform task and external memory is used to store bulk information, which includes large software and data.

Memory is used to store the information in digital form. The memory hierarchy is given by:

Register
Cache Memory
Main Memory
Magnetic Disk
Removable media (Magnetic tape)

Register:

This is a part of Central Processor Unit, so they reside inside the CPU. The information from main memory is brought to CPU and keep the information in register. Due to space and cost constraints, we have got a limited number of registers in a CPU. These are basically faster devices.

Cache Memory:

Er. ExamRadar

Cache memory is a storage device placed in between CPU and main memory. These are semiconductor memories. These are basically fast memory device, faster than main memory.

We can not have a big volume of cache memory due to its higher cost and some constraints of the CPU. Due to higher cost we can not replace the whole main memory by faster memory. Generally, the most recently used information is kept in the cache memory. It is brought from the main memory and placed in the cache memory. Now a days, we get CPU with internal cache.

Main Memory:

Like cache memory, main memory is also semiconductor memory. But the main memory is relatively slower memory. We have to first bring the information (whether it is data or program), to main memory. CPU can work with the information available in main memory only.

Magnetic Disk:

This is bulk storage device. We have to deal with huge amount of data in many application. But we don't have so much semiconductor memory to keep these information in our computer. On the other hand, semiconductor memories are volatile in nature. It loses its content once we switch off the computer. For permanent storage, we use magnetic disk. The storage capacity of magnetic disk is very high.

Removable media:

For different application, we use different data. It may not be possible to keep all the information in magnetic disk. So, which ever data we are not using currently, can be kept in removable media. Magnetic tape is one kind of removable medium. CD is also a removable media, which is an optical device.

Register, cache memory and main memory are internal memory. Magnetic Disk, removable media are external memory. Internal memories are semiconductor memory. Semiconductor memories are categoried as volatile memory and non-volatile memory.

RAM: Random Access Memories are volatile in nature. As soon as the computer is switched off, the contents of memory are also lost.

ROM: Read only memories are non volatile in nature. The storage is permanent, but it is read only memory. We can not store new information in ROM.

Several types of ROM are available:

Er. ExamRadar

PROM: Programmable Read Only Memory; it can be programmed once as per user requirements.

EPROM: Erasable Programmable Read Only Memory; the contents of the memory can be erased and store new data into the memory. In this case, we have to erase whole information.

EEPROM: Electrically Erasable Programmable Read Only Memory; in this type of memory the contents of a particular location can be changed without effecting the contents of other location.

Main Memory

The main memory of a computer is semiconductor memory. The main memory unit of computer is basically consists of two kinds of memory:

RAM : Random access memory; which is volatile in nature.
ROM : Read only memory; which is non-volatile.

The permanent information are kept in ROM and the user space is basically in RAM.

The smallest unit of information is known as bit (binary digit), and in one memory cell we can store one bit of information. 8 bit together is termed as a byte.

The maximum size of main memory that can be used in any computer is determined by the addressing scheme.

A computer that generates 16-bit address is capable of addressing upto 216 which is equal to 64K memory location. Similarly, for 32 bit addresses, the total capacity will be 232 which is equal to 4G memory location.

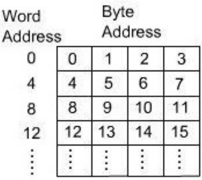
In some computer, the smallest addressable unit of information is a memory word and the machine is called word-addressable.
In some computer, individual address is assigned for each byte of information, and it is called byte-addressable computer. In this computer, one memory word contains one or more memory bytes which can be addressed individually.

A byte addressable 32-bit computer, each memory word contains 4 bytes. A possible way of address assignment is shown in figure3.1. The address of a word is always integer multiple of 4.

The main memory is usually designed to store and retrieve data in word length quantities. The word length of a computer is generally defined by the number of bits actually stored or retrieved in one main memory access.

Consider a machine with 32 bit address bus. If the word size is 32 bit, then the high order 30 bit will specify the address of a word. If we want to access any byte of the word, then it can be specified by the lower two bit of the address bus.

Er. ExamRadar

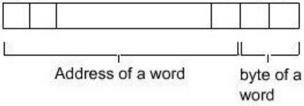


Word Address	Byte Address			
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11
12	12	13	14	15
⋮	⋮	⋮	⋮	⋮

Organization of the main memory in a 32-bit byte addressable computer

32 bit address bus/word size is 32 bit

31 30 2 1 0



Address assignment to a 4-byte word

The data transfer between main memory and the CPU takes place through two CPU registers.

MAR : Memory Address Register
MDR : Memory Data Register.
If the MAR is k-bit long, then the total addressable memory location will be 2k.

If the MDR is n-bit long, then the n bit of data is transferred in one memory cycle.

The transfer of data takes place through memory bus, which consist of address bus and data bus. In the above example, size of data bus is n-bit and size of address bus is k bit.

It also includes control lines like Read, Write and Memory Function Complete (MFC) for coordinating data transfer. In the case of byte addressable computer, another control line to be added to indicate the byte transfer instead of the whole word.
For memory operation, the CPU initiates a memory operation by loading the appropriate data i.e., address to MAR.

If it is a memory read operation, then it sets the read memory control line to 1. Then the contents of the memory location is brought to MDR and the memory control circuitry indicates this to the CPU by setting MFC to 1.

If the operation is a memory write operation, then the CPU places the data into MDR and sets the write memory control line to 1. Once the contents of MDR are stored in specified memory location, then the memory control circuitry indicates the end of operation by setting MFC to 1.

A useful measure of the speed of memory unit is the time that elapses between the initiation of an operation and the completion of the operation (for example, the time between Read and MFC). This is referred to as Memory Access Time. Another measure is memory cycle time. This is the minimum time delay between the initiation two independent memory operations (for example, two successive memory read operation). Memory cycle time is slightly larger than memory access time.

Er. ExamRadar

Binary Storage Cell:

The binary storage cell is the basic building block of a memory unit.

The binary storage cell that stores one bit of information can be modelled by an SR latch with associated gates.
1 bit Binary Cell (BC)

The binary cell sotres one bit of information in its internal latch.

Control input to binary cell

Select	Read/Write	Memory Operation
0	X	None
1	0	Write
1	1	Read

The storage part is modelled here with SR-latch, but in reality it is an electronics circuit made up of transistors.
The memory consttucted with the help of transistors is known as semiconductor memory. The semiconductor memories are termed as Random Access Memory(RAM), because it is possible to access any memory location in random.

Depending on the technology used to construct a RAM, there are two types of RAM –

SRAM: Static Random Access Memory.
DRAM: Dynamic Random Access Memory.

Dynamic Ram (DRAM):

A DRAM is made with cells that store data as charge on capacitors. The presence or absence of charge in a capacitor is interpreted as binary 1 or 0.
Because capacitors have a natural tendency to discharge due to leakage current, dynamic RAM require periodic charge refreshing to maintain data storage. The term dynamic refers to this tendency of the stored charge to leak away, even with power continuously applied.
For the write operation, a voltage signal is applied to the bit line B, a high voltage represents 1 and a low voltage represents 0. A signal is then applied to the address line, which will turn on the transistor T, allowing a charge to be transferred to the capacitor.

For the read operation, when a signal is applied to the address line, the transistor T turns on and the charge stored on the capacitor is fed out onto the bit line B and to a sense amplifier.

The sense amplifier compares the capacitor voltage to a reference value and determines if the cell contains a logic 1 or a logic 0.

The read out from the cell discharges the capacitor, widh must be restored to complete the read operation.

Due to the discharge of the capacitor during read operation, the read operation of DRAM is termed as destructive read out.

Static RAM (SRAM):

Er. ExamRadar

In an SRAM, binary values are stored using traditional flip-flop constructed with the help of transistors. A static RAM will hold its data as long as power is supplied to it.

SRAM Versus DRAM :

Both static and dynamic RAMs are volatile, that is, it will retain the information as long as power supply is applied.

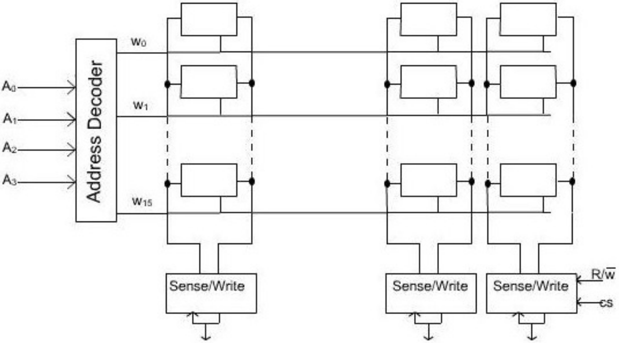
A dynamic memory cell is simpler and smaller than a static memory cell. Thus a DRAM is more dense,
i.e., packing density is high(more cell per unit area). DRAM is less expensive than corresponding SRAM.

DRAM requires the supporting refresh circuitry. For larger memories, the fixed cost of the refresh circuitry is more than compensated for by the less cost of DRAM cells

SRAM cells are generally faster than the DRAM cells. Therefore, to construct faster memory modules(like cache memory) SRAM is used.

Internal Organization of Memory Chips

A memory cell is capable of storing 1-bit of information. A number of memory cells are organized in the form of a matrix to form the memory chip.



b_7 b_1 b_0

16 memory location w_0, w_1,...................w_{15}

8 bits in each location b_0, b_1, ...b_7

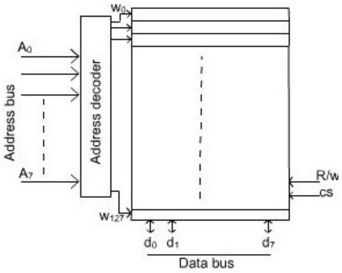
Figure : 16 X 8 Memory Organization

Each row of cells constitutes a memory word, and all cell of a row are connected to a common line which is referred as word line. An address decoder is used to drive the word line. At a particular instant, one word line is enabled depending on the address present in the address bus. The cells in each column are connected by two lines. These are known as bit lines. These bit lines are connected to data input line and data output line through a Sense/Write circuit. During a Read operation, the Sense/Write circuit sense, or read the information stored in the cells selected by a word line and transmit this information to the output data line. During a write operation, the sense/write circuit receive information and store it in the cells of the selected word.

A memory chip consisting of 16 words of 8 bits each, usually referred to as 16 x 8 organization. The data input and data output line of each Sense/Write circuit are connected to a single bidirectional data line in order to reduce the pin required. For 16 words, we need an address bus of size 4. In addition to address and data lines, two control lines, and CS, are provided. The line is to used to specify the required operation about read or write. The CS (Chip Select) line is required to select a given chip in a multi chip memory system.

Consider a slightly larger memory unit that has 1K (1024) memory cells...
128 x 8 memory chips:

If it is organised as a 128 x 8 memory chips, then it has got 128 memory words of size 8 bits. So the size of data bus is 8 bits and the size of address bus is 7 bits (2^7=128). The storage organization of 128 x 8 memory chip is shown in the figure 3.6.



1024 x 1 memory chips:

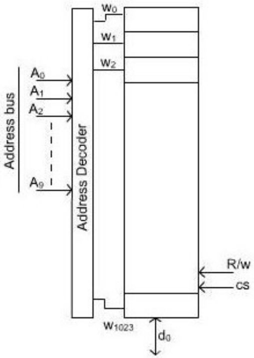
Er. ExamRadar

If it is organized as a 1024 x 1 memory chips, then it has got 1024 memory words of size 1 bit only.

Therefore, the size of data bus is 1 bit and the size of address bus is 10 bits (2^10=1024).

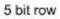
A particular memory location is identified by the contents of memory address bus. A decoder is used to decode the memory address. There are two ways of decoding of a memory address depending upon the organization of the memory module.

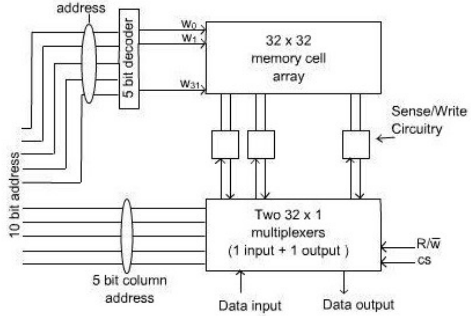
In one case, each memory word is organized in a row. In this case whole memory address bus is used together to decode the address of the specified location. The memory organization of 1024 x 1 memory chip is shown in the figure below



In second case, several memory words are organized in one row. In this case, address bus is divided into two gropus.

One group is used to form the row address and the second group is used to form the column address. Consider the memory organization of 1024 x 1 memory chip. The required 10-bit address is divided into two groups of 5 bits each to form the row and column address of the cell array. A row address selects a row of 32 cells, all of which are accessed in parallel. However, according to the column address, only one of these cells is connected to the external data line via the input output multiplexers. The arrangement for row address and column address decoders is shown in the figure below

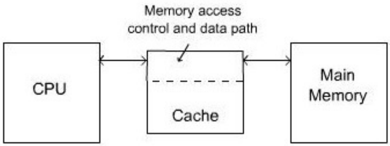




Cache Memory

Er. ExamRadar

Analysis of large number of programs has shown that a number of instructions are executed repeatedly. This may be in the form of a simple loops, nested loops, or a few procedures that repeatedly call each other. It is observed that many instructions in each of a few localized areas of the program are repeatedly executed, while the remainder of the program is accessed relatively less. This phenomenon is referred to as locality of reference.



Now, if it can be arranged to have the active segments of a program in a fast memory, then the tolal execution time can be significantly reduced. It is the fact that CPU is a faster device and memory is a relatively slower device. Memory access is the main bottleneck for the performance efficiency. If a faster memory device can be inserted between main memory and CPU, the efficiency can be increased. The faster memory that is inserted between CPU and Main Memory is termed as Cache memory. To make this arrangement effective, the cache must be considerably faster than the main memory, and typically it is 5 to 10 time faster than the main memory. This approach is more economical than the use of fast memory device to implement the entire main memory. This is also a feasible due to the locality of reference that is present in most of the program, which reduces the frequent data transfer between main memory and cache memory.

Operation of Cache Memory

The memory control circuitry is designed to take advantage of the property of locality of reference. Some assumptions are made while designing the memory control circuitry:

The CPU does not need to know explicitly about the existence of the cache.

The CPU simply makes Read and Write request. The nature of these two operations are same whether cache is present or not.

The address generated by the CPU always refer to location of main memory.

The memory access control circuitry determines whether or not the requested word currently exists in the cache.

When a Read request is received from the CPU, the contents of a block of memory words containing the location specified are transferred into the cache. When any of the locations in this block is referenced by the program, its contents are read directly from the cache.

Consider the case where the addressed word is not in the cache and the operation is a read. First the block of words is brought to the cache and then the requested word is forwarded to the CPU. But it can be forwarded to the CPU as soon as it is available to the cache, instaead of the whole block to be loaded in the cache. This is called load through, and there is some scope to save time while using load through policy.

The cache memory can store a number of such blocks at any given time.

The correspondence between the Main Memory Blocks and those in the cache is specified by means of a mapping function.

When the cache is full and a memory word is referenced that is not in the cache, a decision must be made as to which block should be removed from the cache to create space to bring the new block to the cache that contains the referenced word. Replacement algorithms are used to make the proper selection of block that must be replaced by the new one.

When a write request is received from the CPU, there are two ways that the system can proceed. In the first case, the cache location and the main memory location are updated simultaneously. This is called the store through method or write through method.

The alternative is to update the cache location only. During replacement time, the cache block will be written back to the main memory. This method is called write back method. If there is no new write operation in the cache block, it is not required to write back the cache block in the main memory. This information can be kept with the help of an associated bit. This bit it set while there is a write operation in the cache block. During replacement, it checks this bit, if it is set, then write back the cache block in main memory otherwise not. This bit is known as dirty bit. If the bit gets dirty (set to one), writting to main memory is required.

The write through method is simpler, but it results in unnecessary write operations in the main memory when a given cache word is updated a number of times during its cache residency period.
During a write operation, if the address word is not in the cache, the information is written directly into the main memory. A write operation normally refers to the location of data areas and the property of locality of reference is not as pronounced in accessing data when write operation is involved. Therefore, it is not advantageous to bring the data block to the cache when there a write operation, and the addressed word is not present in cache.

Mapping Functions

Er. ExamRadar

The mapping functions are used to map a particular block of main memory to a particular block of cache. This mapping function is used to transfer the block from main memory to cache memory. Three different mapping functions are available:

Direct mapping:

A particular block of main memory can be brought to a particular block of cache memory. So, it is not flexible.

Associative mapping:

In this mapping function, any block of Main memory can potentially reside in any cache block position. This is much more flexible mapping method.

Block-set-associative mapping:

In this method, blocks of cache are grouped into sets, and the mapping allows a block of main memory to reside in any block of a specific set. From the flexibility point of view, it is in between to the other two methods.

All these three mapping methods are explained with the help of an example.

Consider a cache of 4096 (4K) words with a block size of 32 words. Therefore, the cache is organized as 128 blocks. For 4K words, required address lines are 12 bits. To select one of the block out of 128 blocks, we need 7 bits of address lines and to select one word out of 32 words, we need 5 bits of address lines. So the total 12 bits of address is divided for two groups, lower 5 bits are used to select a word within a block, and higher 7 bits of address are used to select any block of cache memory.

Let us consider a main memory system consisting 64K words. The size of address bus is 16 bits. Since the block size of cache is 32 words, so the main memory is also organized as block size of 32 words. Therefore, the total number of blocks in main memory is 2048 (2K x 32 words = 64K words). To identify any one block of 2K blocks, we need 11 address lines. Out of 16 address lines of main memory, lower 5 bits are used to select a word within a block and higher 11 bits are used to select a block out of 2048 blocks.

Number of blocks in cache memory is 128 and number of blocks in main memory is 2048, so at any instant of time only 128 blocks out of 2048 blocks can reside in cache menory. Therefore, we need mapping function to put a particular block of main memory into appropriate block of cache memory.

Direct Mapping Technique:

Er. ExamRadar

The simplest way of associating main memory blocks with cache block is the direct mapping technique. In this technique, block k of main memory maps into block k modulo m of the cache, where m is the total number of blocks in cache. In this example, the value of m is 128. In direct mapping technique, one particular block of main memory can be transfered to a particular block of cache which is derived by the modulo function.

Since more than one main memory block is mapped onto a given cache block position, contention may arise for that position. This situation may occurs even when the cache is not full. Contention is resolved by allowing the new block to overwrite the currently resident block. So the replacement algorithm is trivial.

The detail operation of direct mapping technique is as follows:

The main memory address is divided into three fields. The field size depends on the memory capacity and the block size of cache. In this example, the lower 5 bits of address is used to identify a word within a block. Next 7 bits are used to select a block out of 128 blocks (which is the capacity of the cache). The remaining 4 bits are used as a TAG to identify the proper block of main memory that is mapped to cache.

When a new block is first brought into the cache, the high order 4 bits of the main memory address are stored in four TAG bits associated with its location in the cache. When the CPU generates a memory request, the 7-bit block address determines the corresponding cache block. The TAG field of that block is compared to the TAG field of the address. If they match, the desired word specified by the low-order 5 bits of the address is in that block of the cache.

If there is no match, the required word must be accessed from the main memory, that is, the contents of that block of the cache is replaced by the new block that is specified by the new address generated by the CPU and correspondingly the TAG bit will also be changed by the high order 4 bits of the address. The whole arrangement for direct mapping technique is shown in the figure below.

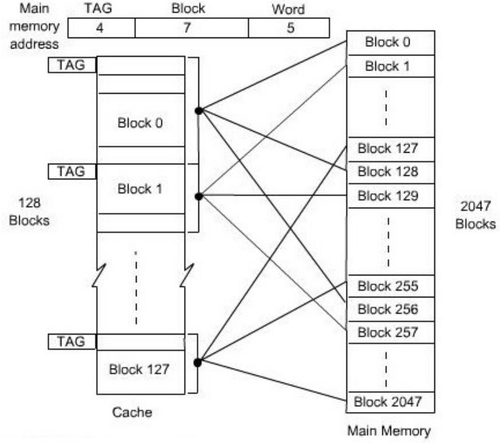


Figure : Direct-mapping cache

Er. ExamRadar

Associated Mapping Technique:

In the associative mapping technique, a main memory block can potentially reside in any cache block position. In this case, the main memory address is divided into two groups, low-order bits identifies the location of a word within a block and high-order bits identifies the block. In the example here, 11 bits are required to identify a main memory block when it is resident in the cache , high-order 11 bits are used as TAG bits and low-order 5 bits are used to identify a word within a block. The TAG bits of an address received from the CPU must be compared to the TAG bits of each block of the cache to see if the desired block is present.

In the associative mapping, any block of main memory can go to any block of cache, so it has got the complete flexibility and we have to use proper replacement policy to replace a block from cache if the currently accessed block of main memory is not present in cache. It might not be practical to use this complete flexibility of associative mapping technique due to searching overhead, because the TAG field of main memory address has to be compared with the TAG field of all the cache block. In this example, there are 128 blocks in cache and the size of TAG is 11 bits. The whole arrangement of Associative Mapping Technique is shown in the figure below.

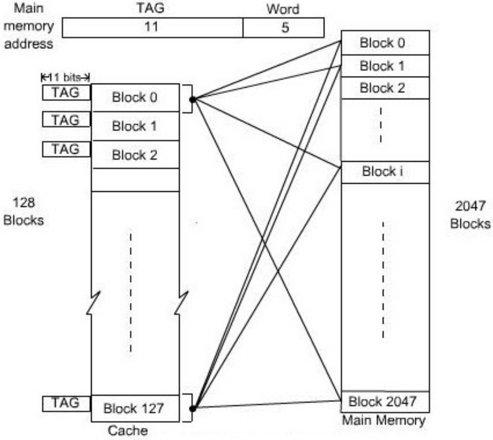


Figure : Associated Mapping Cacke

Block-Set-Associative Mapping Technique:

This mapping technique is intermediate to the previous two techniques. Blocks of the cache are grouped into sets, and the mapping allows a block of main memory to reside in any block of a specific set. Therefore, the flexibity of associative mapping is reduced from full freedom to a set of specific blocks. This also reduces the searching overhead, because the search is restricted to number of sets, instead of number of blocks. Also the contention problem of the direct mapping is eased by having a few choices for block replacement.

Consider the same cache memory and main memory organization of the previous example. Organize the cache with 4 blocks in each set. The TAG field of associative mapping technique is divided into two groups, one is termed as SET bit and the second one is termed as TAG bit. Each set contains 4 blocks, total number of set is 32. The main memory address is grouped into three parts: low-order 5 bits are used to identifies a word within a block. Since there are total 32 sets present, next 5 bits are used to identify the set. High-order 6 bits are used as TAG bits.

The 5-bit set field of the address determines which set of the cache might contain the desired block. This is similar to direct mapping technique, in case of direct mapping, it looks for block, but in case of block-set-associative mapping, it looks for set. The TAG field of the address must then be compared with the TAGs of the four blocks of that set. If a match occurs, then the block is present in the cache; otherwise the block containing the addressed word must be brought to the cache. This block will potentially come to the cooresponding set only. Since, there are four blocks in the set, we have to choose appropriately which block to be replaced if all the blocks are occupied. Since the search is restricted to four block only, so the searching complexity is reduced. The whole arrangement of block-set-associative mapping technique is shown in the figure below.

It is clear that if we increase the number of blocks per set, then the number of bits in SET field is reduced. Due to the increase of blocks per set, complexity of search is also increased. The extreme condition of 128 blocks per set requires no set bits and corrsponds to the fully associative mapping technique with 11 TAG bits. The other extreme of one block per set is the direct mapping method.

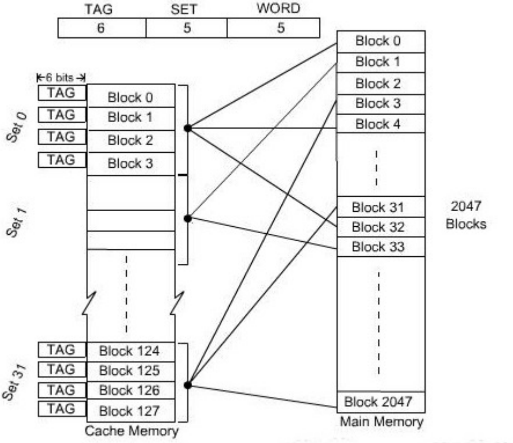


Figure : Block-set Associated mapping Cache with 4 blocks per set

Replacement Algorithms

Er. ExamRadar

When a new block must be brought into the cache and all the positions that it may occupy are full, a decision must be made as to which of the old blocks is to be overwritten. In general, a policy is required to keep the block in cache when they are likely to be referenced in near future. However, it is not easy to determine directly which of the block in the cache are about to be referenced. The property of locality of reference gives some clue to design good replacement policy.

Least Recently Used (LRU) Replacement policy:

Since program usually stay in localized areas for reasonable periods of time, it can be assumed that there is a high probability that blocks which have been referenced recently will also be referenced in the near future. Therefore, when a block is to be overwritten, it is a good decision to overwrite the one that has gone for longest time without being referenced. This is defined as the least recently used (LRU) block. Keeping track of LRU block must be done as computation proceeds.

Consider a specific example of a four-block set. It is required to track the LRU block of this four-block set. A 2-bit counter may be used for each block.

When a hit occurs, that is, when a read request is received for a word that is in the cache, the counter of the block that is referenced is set to 0. All counters which values originally lower than the referenced one are incremented by 1 and all other counters remain unchanged.

When a miss occurs, that is, when a read request is received for a word and the word is not present in the cache, we have to bring the block to cache.

There are two possibilities in case of a miss:

1. If the set is not full, the counter associated with the new block loaded from the main memory is set to 0, and the values of all other counters are incremented by 1.
2. If the set is full and a miss occurs, the block with the counter value 3 is removed , and the new block is put in its palce. The counter value is set to zero. The other three block counters are incremented by 1.

It is easy to verify that the counter values of occupied blocks are always distinct. Also it is trivial that highest counter value indicates least recently used block.

First In First Out (FIFO) replacement policy:

Er. ExamRadar

A reasonable rule may be to remove the oldest from a full set when a new block must be brought in. While using this technique, no updation is required when a hit occurs. When a miss occurs and the set is not full, the new block is put into an empty block and the counter values of the occupied block will be increment by one. When a miss occurs and the set is full, the block with highest counter value is replaced by new block and counter is set to 0, counter value of all other blocks of that set is incremented by 1. The overhead of the policy is less, since no updation is required during hit.

Random replacement policy:

The simplest algorithm is to choose the block to be overwritten at random. Interestingly enough, this simple algorithm has been found to be very effective in practice.