Memory Organization

Microcomputer Memory

- Memory is an essential component of the microcomputer system.
- It stores binary instructions and datum for the microcomputer.
- The memory is the place where the computer holds current programs and data that are in use.
- None technology is optimal in satisfying the memory requirements for a computer system.
- Computer memory exhibits perhaps the widest range of type, technology, organization, performance and cost of any feature of a computer system.
- The memory unit that communicates directly with the CPU is called main memory.
- Devices that provide backup storage are called auxiliary memory or secondary memory.

Characteristics of memory systems

The memory system can be characterized with their Location, Capacity, Unit of transfer, Access method, Performance, Physical type, Physical characteristics, Organization.

Location

- Processor memory: The memory like registers is included within the processor and termed as processor memory.
- Internal memory: It is often termed as main memory and resides within the CPU.
- External memory: It consists of peripheral storage devices such as disk and magnetic tape that are accessible to processor via i/o controllers.

Capacity

- Word size: Capacity is expressed in terms of words or bytes.
  - The natural unit of organization
- Number of words: Common word lengths are 8, 16, 32 bits etc.
  - or Bytes

Unit of Transfer

- Internal: For internal memory, the unit of transfer is equal to the number of data lines into and out of the memory module.
- External: For external memory, they are transferred in block which is larger than a word.
- Addressable unit
  - Smallest location which can be uniquely addressed
  - Word internally
  - Cluster on Magnetic disks

Access Method

- Sequential access: In this access, it must start with beginning and read through a specific linear sequence. This means access time of data unit depends on position of records (unit of data) and previous location.
    - e.g. tape
- Direct Access: Individual blocks of records have unique address based on location. Access is accomplished by jumping (direct access) to general vicinity plus a sequential search to reach the final location.
    - e.g. disk
- Random access: The time to access a given location is independent of the sequence of prior accesses and is constant. Thus any location can be selected out randomly and directly addressed and accessed.
    - e.g. RAM
- Associative access: This is random access type of memory that enables one to make a comparison of desired bit locations within a word for a specified match, and to do this for all words simultaneously.
    - e.g. cache

Performance

- Access time: For random access memory, access time is the time it takes to perform a read or write operation i.e. time taken to address a memory plus to read / write from addressed memory location. Whereas for non-random access, it is the time needed to position read / write mechanism at desired location.
    - Time between presenting the address and getting the valid data
- Memory Cycle time: It is the total time that is required to store next memory access operation from the previous memory access operation.
- Memory cycle time = access time plus transient time (any additional time required before a second access can commence).
    - Time may be required for the memory to “recover” before next access
    - Cycle time is access + recovery
- Transfer Rate: This is the rate at which data can be transferred in and out of a memory unit.
    - Rate at which data can be moved
    - For random access, R = 1 / cycle time
    - For non-random access, Tn = Ta + N / R; where Tn – average time to read or write N bits, Ta – average access time, N – number of bits, R – Transfer rate in bits per second (bps).

Physical Types

- Semiconductor
    - RAM
- Magnetic
    - Disk & Tape
- Optical
    - CD & DVD
- Others
    - Bubble
    - Hologram

Physical Characteristics

- Decay: Information decays mean data loss.
- Volatility: Information decays when electrical power is switched off.
- Erasable: Erasable means permission to erase.
- Power consumption: how much power consumes?

Organization

- Physical arrangement of bits into words
- Not always obvious
    - e.g. interleaved

The Memory Hierarchy

- Capacity, cost and speed of different types of memory play a vital role while designing a memory system for computers.
- If the memory has larger capacity, more application will get space to run smoothly.
- It's better to have fastest memory as far as possible to achieve a greater performance. Moreover for the practical system, the cost should be reasonable.
- There is a tradeoff between these three characteristics cost, capacity and access time. One cannot achieve all these quantities in same memory module because
- If capacity increases, access time increases (slower) and due to which cost per bit decreases.
- If access time decreases (faster), capacity decreases and due to which cost per bit increases.
- The designer tries to increase capacity because cost per bit decreases and the more application program can be accommodated. But at the same time, access time increases and hence decreases the performance.

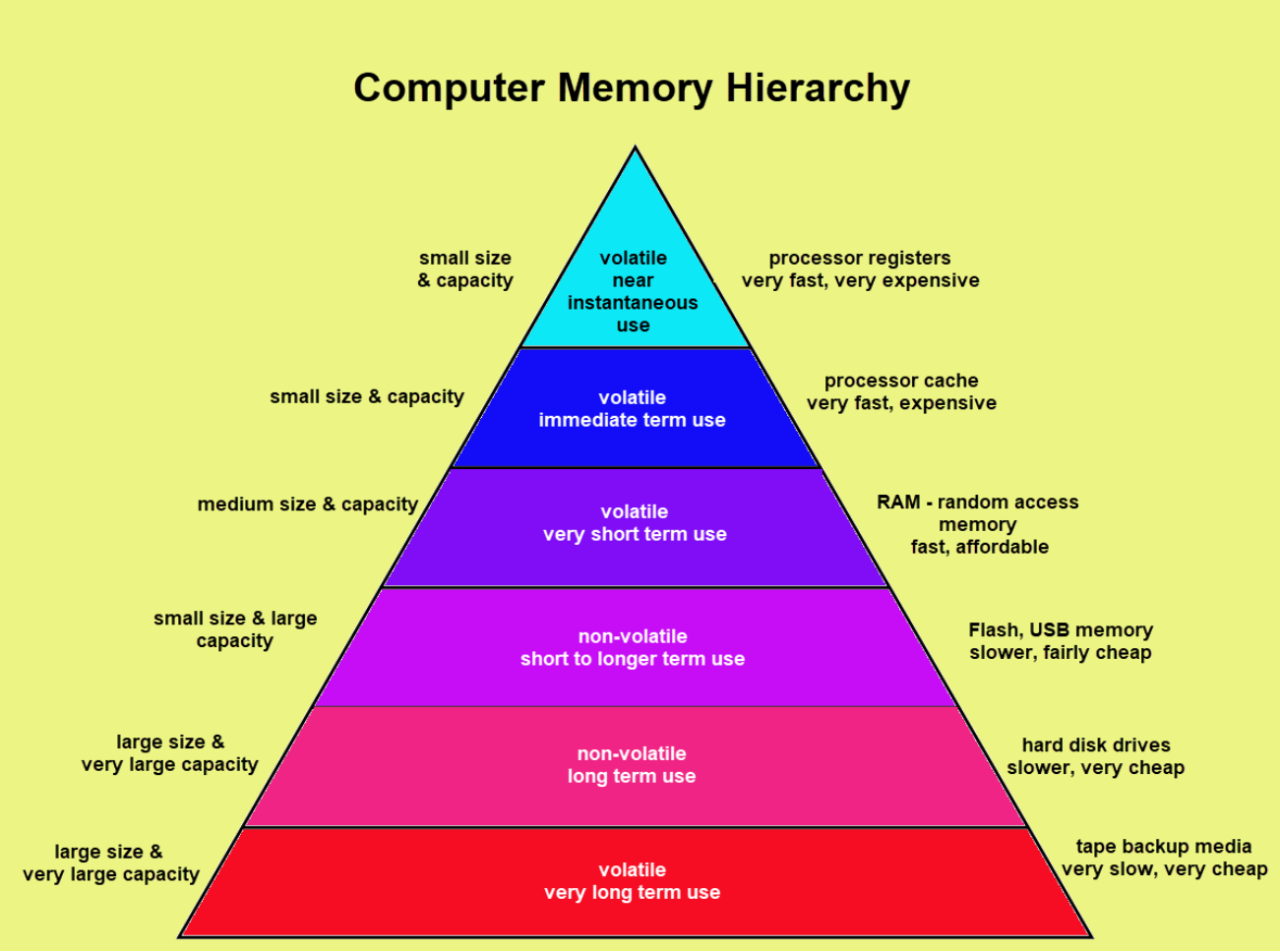
So the best idea will be to use memory hierarchy.

- Memory Hierarchy is to obtain the highest possible access speed while minimizing the total cost of the memory system.
- Not all accumulated information is needed by the CPU at the same time.
- Therefore, it is more economical to use low-cost storage devices to serve as a backup for storing the information that is not currently used by CPU
- The memory unit that directly communicate with CPU is called the *main memory*
- Devices that provide backup storage are called *auxiliary memory*
- The memory hierarchy system consists of all storage devices employed in a computer system from the slow by high-capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory
- The main memory occupies a central position by being able to communicate directly with the CPU and with auxiliary memory devices through an I/O processor
- A special very-high-speed memory called **cache** is used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate
- CPU logic is usually faster than main memory access time, with the result that processing speed is limited primarily by the speed of main memory
- The cache is used for storing segments of programs currently being executed in the CPU and temporary data frequently needed in the present calculations

- The memory hierarchy system consists of all storage devices employed in a computer system from slow but high capacity auxiliary memory to a relatively faster cache memory accessible to high speed processing logic. The figure below illustrates memory hierarchy.
- As we go down in the hierarchy
    - Cost per bit decreases
    - Capacity of memory increases
    - Access time increases
    - Frequency of access of memory by processor also decreases.

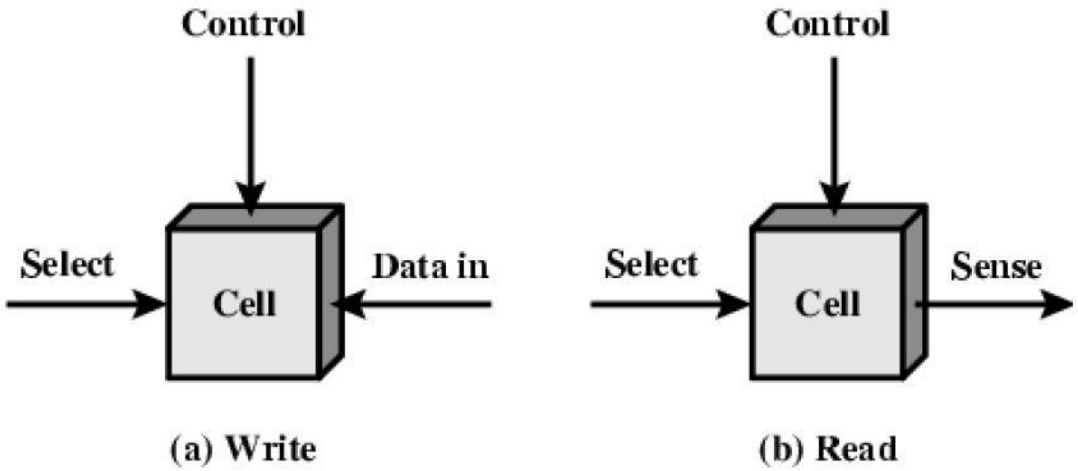
Hierarchy List

- Registers
- L1 Cache
- L2 Cache
- L3 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape



Main Memory

- The main memory is the central unit of the computer system. It is relatively large and fast memory to store programs and data during the computer operation. These memories employ semiconductor integrated circuits. The basic element of the semiconductor memory is the memory cell.
- The memory cell has three functional terminals which carries the electrical signal.
  -The select terminal: It selects the cell.
  -The data in terminal: It is used to input data as 0 or 1 and data out or sense terminal is used for the output of the cell's state.
  -The control terminal: It controls the function i.e. it indicates read and write.



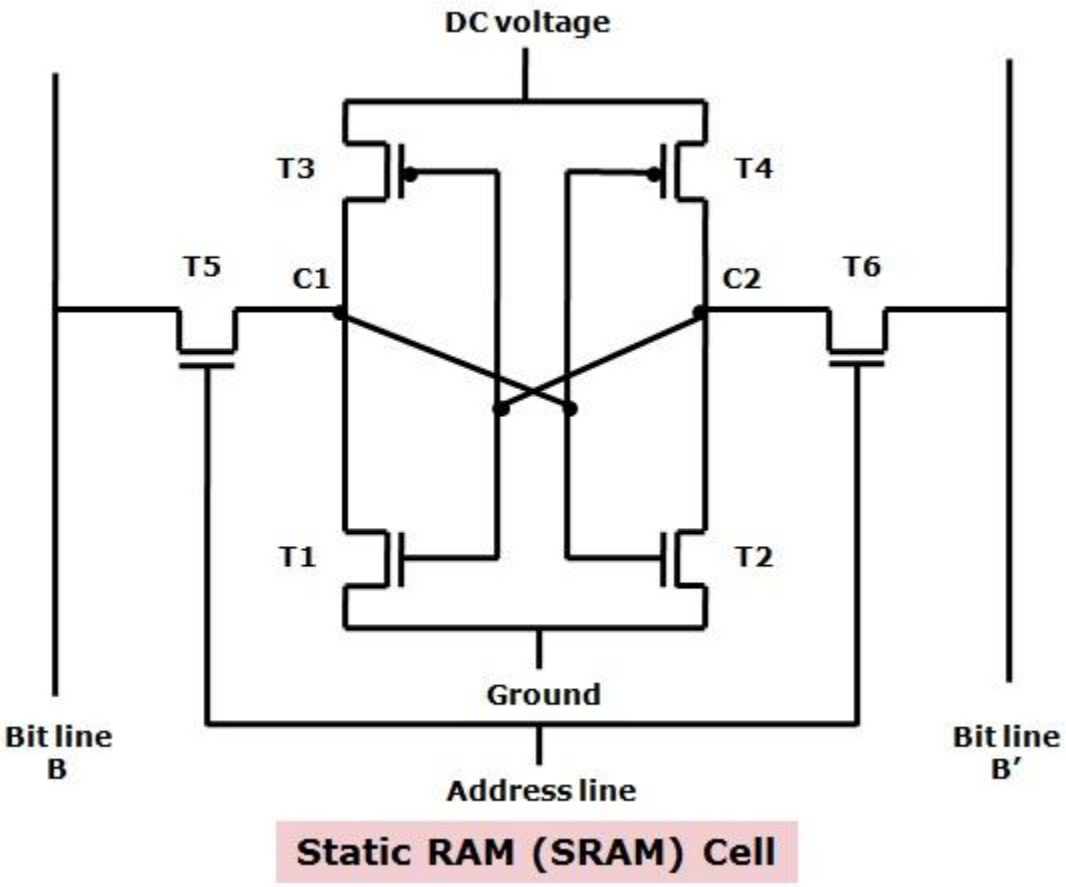
- Most of the main memory in a general purpose computer is made up of RAM integrated circuits chips, but a portion of the memory may be constructed with ROM chips.

RAM– Random Access memory

- Memory cells can be accessed for information transfer from any desired random location.
- The process of locating a word in memory is the same and requires of locating a word in memory is the same and requires an equal amount of time no matter where the cells are located physically in memory thus named 'Random access'.
- Integrated RAM are available in two possible operating modes, S*tatic and Dynamic*

Static RAM (SRAM)

- The static RAM consists of flip flop that stores binary information and this stored information remains valid as long as power is applied to the unit.

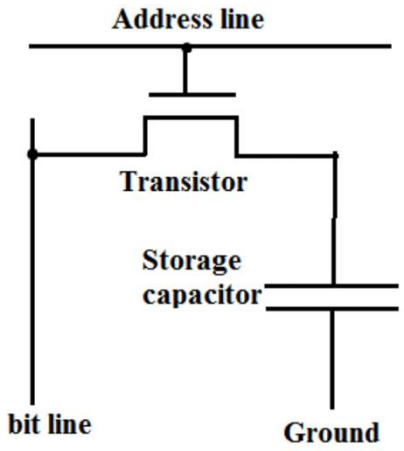


Static RAM (SRAM) Cell

- Four transistors T1, T2, T3 and t4 are cross connected in an arrangement that produces a stable logical state.
- In logic state 1, point C1 is high and point C2 is low. In this state, T1 & T4 are off and T2 & T3 are on.
- In logic state 0, point C1 is low and C2 is high. In this state, T1 & T4 are on and T2 & T3 are off.
- The address line controls the two transistors T5 & T6. When a signal is applied to this line, the two transistors are switched on allowing for read and write operation.
- For a write operation, the desired bit value is applied to line B while it's complement is applied to line B complement. This forces the four transistors T1, T2, T3 & T4 into a proper state.
- For the read operation, the bit value is read from line B.

Dynamic RAM (DRAM)

- The dynamic RAM stores the binary information in the form of electrical charges and capacitor is used for this purpose.
- Since charge stored in capacitor discharges with time, capacitor must be periodically recharged and which is also called refreshing memory.



- The address line is activated when the bit value from this cell is to be read or written.
- The transistor acts as switch that is closed i.e. allowed current to flow, if voltage is applied to the address line; and opened i.e. no current to flow, if no voltage is present in the address line.

For DRAM writing

- The address line is activated which causes the transistor to conduct.
- The sense amplifier senses the content of the data bus line for this cell.
- If the bus line is low, then amplifier will ground the bit line of cell and any charge in capacitor is addressed out.
- If data bus is high, then a +5V is applied on bit line and voltage will flow through transistor and charge the capacitor.

For DRAM reading

- Address line is activated which causes the transistor to conduct.
- If there is charge stored in capacitor, then current will flow through transistor and raise the voltage in bit line. The amplifier will store the voltage and place a 1 on data out line.
- If there is no charge stored in capacitor, then no current will flow through transistor and voltage bit line will not be raised. The amplifier senses that there is no charge and places a 0 on data out line.

SRAM versus DRAM

- Both volatile
  Power needed to preserve data
- Static RAM
  - Uses flip flop to store information
  - Needs more space
  - Faster, digital device
  - Expensive, big in size
  - Don't require refreshing circuit
  - Used in cache memory
- Dynamic RAM
  - Uses capacitor to store information
  - More dense i.e. more cells can be accommodated per unit area
  - Slower, analog device
  - Less expensive, small in size
  - Needs refreshing circuit
  - Used in main memory, larger memory units

ROM– Read Only memory

- Read only memory (ROM) contains a permanent pattern of data that cannot be changed.
- A ROM is non-volatile that is no power source is required to maintain the bit values in memory.
- While it is possible to read a ROM, it is not possible to write new data into it.
- The data or program is permanently presented in main memory and never be loaded from a secondary storage device with the advantage of ROM.
- A ROM is created like any other integrated circuit chip, with the data actually wired into the chip as part of the fabrication process.
- It presents two problems
  - The data insertion step includes a relatively large fixed cost, whether one or thousands of copies of a particular ROM are fabricated.
  - There is no room for error. If one bit is wrong, the whole batch of ROM must be thrown out.

Types of ROM

- Programmable ROM (PROM)
  It is non-volatile and may be written into only once. The writing process is performed electrically and may be performed by a supplier or customer at a time later than the original chip fabrication.
- Erasable Programmable ROM (EPROM)
  It is read and written electrically. However, before a write operation, all the storage cells must be erased to the same initial state by exposure of the packaged chip to ultraviolet radiation (UV ray). Erasure is performed by shining an intense ultraviolet light through a window that is designed into the memory chip. EPROM is optically managed and more expensive than PROM, but it has the advantage of the multiple update capability.

- Electrically Erasable programmable ROM (EEPROM)
  This is a read mostly memory that can be written into at any time without erasing prior contents, only the byte or byte addresses are updated. The write operation takes considerably longer than the read operation, on the order of several hundred microseconds per byte. The EEPROM combines the advantage of non-volatility with the flexibility of being updatable in place, using ordinary bus control, addresses and data lines. EEPROM is more expensive than EPROM and also is less dense, supporting fewer bits per chip.
- Flash Memory
  Flash memory is also the semiconductor memory and because of the speed with which it can be reprogrammed, it is termed as flash. It is interpreted between EPROM and EEPROM in both cost and functionality. Like EEPROM, flash memory uses an electrical erasing technology. An entire flash memory can be erased in one or a few seconds, which is much faster than EPROM. In addition, it is possible to erase just blocks of memory rather than an entire chip. However, flash memory doesn't provide byte level erasure, a section of memory cells are erased in an action or 'flash'.

External Memory

- The devices that provide backup storage are called external memory or auxiliary memory. It includes serial access type such as magnetic tapes and random access type such as magnetic disks.

Magnetic Tape

- A magnetic tape is the strip of plastic coated with a magnetic recording medium. Data can be recorded and read as a sequence of character through read / write head. It can be stopped, started to move forward or in reverse or can be rewound. Data on tapes are structured as number of parallel tracks running length wise. Earlier tape system typically used nine tracks. This made it possible to store data one byte at a time with additional parity bit as 9th track. The recording of data in this form is referred to as parallel recording.

Magnetic Disk

- A magnetic disk is a circular plate constructed with metal or plastic coated with magnetic material often both side of disk are used and several disk stacked on one spindle which Read/write head available on each surface. All disks rotate together at high speed. Bits are stored in magnetize surface in spots along concentric circles called tracks. The tracks are commonly divided into sections called sectors. After the read/write head are positioned in specified track the system has to wait until the rotating disk reaches the specified sector under read/write head. Information transfer is very fast once the beginning of sector has been reached. Disk that are permanently attached to the unit assembly and cannot be used by occasional user are called hard disk drive with removal disk is called floppy disk.

Optical Disk

- The huge commercial success of CD enabled the development of low cost optical disk storage technology that has revolutionized computer data storage. The disk is form from resin such as polycarbonate. Digitally recorded information is imprinted as series of microscopic pits on the surface of poly carbonate. This is done with the finely focused high intensity leaser. The pitted surface is then coated with reflecting surface usually aluminum or gold. The shiny surface is protected against dust and scratches by the top coat of acrylic. Information is retrieved from CD by low power laser. The intensity of reflected light of laser changes as it encounters a pit. Specifically if the laser beam falls on pit which has somewhat rough surface the light scatters

and low intensity is reflected back to the surface. The areas between pits are called lands. A land is a smooth surface which reflects back at higher intensity. The change between pits and land is detected by photo sensor and converted into digital signal. The sensor tests the surface at regular interval.

DVD-Technology

- Multi-layer
- Very high capacity (4.7G per layer)
- Full length movie on single disk
- Using MPEG compression
- Finally standardized (honest!)
- Movies carry regional coding
- Players only play correct region films

DVD-Writable

- Loads of trouble with standards
- First generation DVD drives may not read first generation DVD-W disks
- First generation DVD drives may not read CD-RW disks

RAID (Redundant Arrays of Independent Disks)

RAID, or "Redundant Arrays of Independent Disks" is a technique which makes use of a combination of multiple disks instead of using a single disk for increased performance, data redundancy or both. The term was coined by David Patterson, Garth A. Gibson, and Randy Katz at the University of California, Berkeley in 1987.

Why data redundancy?

Data redundancy, although taking up extra space, adds to disk reliability. This means, in case of disk failure, if the same data is also backed up onto another disk, we can retrieve the data and go on with the operation. On the other hand, if the data is spread across just multiple disks without the RAID technique, the loss of a single disk can affect the entire data.

Key evaluation points for a RAID System

- **Reliability:** How many disk faults can the system tolerate?
- **Availability:** What fraction of the total session time is a system in uptime mode, i.e. how available is the system for actual use?
- **Performance:** How good is the response time? How high is the throughput (rate of processing work)? Note that performance contains a lot of parameters and not just the two.
- **Capacity:** Given a set of N disks each with B blocks, how much useful capacity is available to the user?

RAID is very transparent to the underlying system. This means, to the host system, it appears as a single big disk presenting itself as a linear array of blocks. This allows older technologies to be replaced by RAID without making too many changes in the existing code.

Different RAID levels

RAID-0 (Striping)

- Blocks are “striped” across disks.

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

In the figure, blocks “0,1,2,3” form a stripe.

- Instead of placing just one block into a disk at a time, we can work with two (or more) blocks placed into a disk before moving on to the next one.

Disk 0	Disk 1	Disk 2	Disk 3
0	3	4	6
1	3	5	7
8	10	12	14
9	11	13	15

Evaluation:

- Reliability: 0
  There is no duplication of data. Hence, a block once lost cannot be recovered.
- Capacity: N*B
  The entire space is being used to store data. Since there is no duplication, N disks each having B blocks are fully utilized.

RAID-1 (Mirroring)

- More than one copy of each block is stored in a separate disk. Thus, every block has two (or more) copies, lying on different disks.

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

The above figure shows a RAID-1 system with mirroring level 2.

- RAID 0 was unable to tolerate any disk failure. But RAID 1 is capable of reliability.

Evaluation:
Assume a RAID system with mirroring level 2.

- Reliability: 1 to N/2
  1 disk failure can be handled for certain, because blocks of that disk would have duplicates on some other disk. If we are lucky enough and disks 0 and 2 fail, then again this can be handled as the blocks of these disks have duplicates on disks 1 and 3. So, in the best case, N/2 disk failures can be handled.
- Capacity: N*B/2
  Only half the space is being used to store data. The other half is just a mirror to the already stored data.

RAID-4 (Block-Level Striping with Dedicated Parity)

- Instead of duplicating data, this adopts a parity-based approach.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

In the figure, we can observe one column (disk) dedicated to parity.

- Parity is calculated using a simple XOR function. If the data bits are 0,0,0,1 the parity bit is XOR(0,0,0,1) = 1. If the data bits are 0,1,1,0 the parity bit is XOR(0,1,1,0) = 0. A simple approach is that even number of ones results in parity 0, and an odd number of ones results in parity 1.

C1	C2	C3	C4	Parity
0	0	0	1	1
0	1	1	0	0

Assume that in the above figure, C3 is lost due to some disk failure. Then, we can recompute the data bit stored in C3 by looking at the values of all the other columns and the parity bit. This allows us to recover lost data.

Evaluation:

- Reliability: 1
  RAID-4 allows recovery of at most 1 disk failure (because of the way parity works). If more than one disk fails, there is no way to recover the data.
- Capacity: (N-1)*B
  One disk in the system is reserved for storing the parity. Hence, (N-1) disks are made available for data storage, each disk having B blocks.

RAID-5 (Block-Level Striping with Distributed Parity)

- This is a slight modification of the RAID-4 system where the only difference is that the parity rotates among the drives.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

  In the figure, we can notice how the parity bit “rotates”.

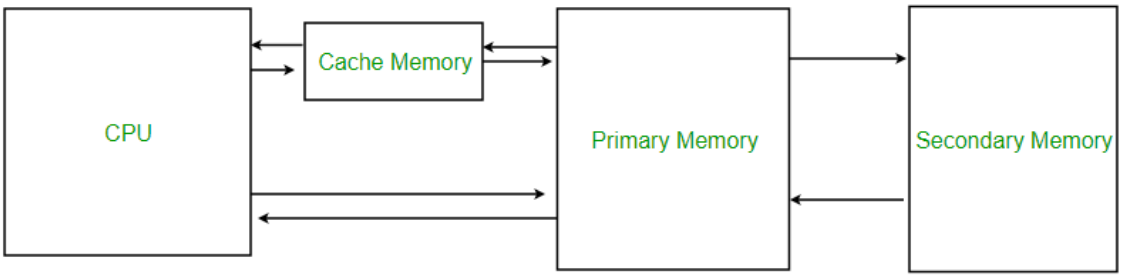
- This was introduced to make the random write performance better.

Evaluation:

- Reliability: 1
  RAID-5 allows recovery of at most 1 disk failure (because of the way parity works). If more than one disk fails, there is no way to recover the data. This is identical to RAID-4.
- Capacity: (N-1)*B
  Overall, space equivalent to one disk is utilized in storing the parity. Hence, (N-1) disks are made available for data storage, each disk having B blocks.

Cache Memory in Computer Organization

Cache Memory is a special very high-speed memory. It is used to speed up and synchronizing with high-speed CPU. Cache memory is costlier than main memory or disk memory but economical than CPU



registers. Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU. It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
Cache memory is used to reduce the average time to access data from the Main memory. The cache is a smaller and faster memory which stores copies of the data from frequently used main memory locations. There are various different independent caches in a CPU, which store instructions and data.

Levels of memory:

- **Level 1 or Register**
  It is a type of memory in which data is stored and accepted that are immediately stored in CPU. Most commonly used register is accumulator, Program counter, address register etc.
- **Level 2 or Cache memory**
  It is the fastest memory which has faster access time where data is temporarily stored for faster access.
- **Level 3 or Main Memory**
  It is memory on which computer works currently. It is small in size and once power is off data no longer stays in this memory.
- **Level 4 or Secondary Memory**
  It is external memory which is not as fast as main memory but data stays permanently in this memory.

Cache Performance
When the processor needs to read or write a location in main memory, it first checks for a corresponding entry in the cache.

- If the processor finds that the memory location is in the cache, a **cache hit** has occurred and data is read from cache
- If the processor **does not** find the memory location in the cache, a **cache miss** has occurred. For a cache miss, the cache allocates a new entry and copies in data from main memory, then the request is fulfilled from the contents of the cache.

The performance of cache memory is frequently measured in terms of a quantity called **Hit ratio.**

```
Hit ratio = hit / (hit + miss) =  no. of hits/total accesses
```

We can improve Cache performance using higher cache block size, higher associativity, reduce miss rate, reduce miss penalty, and reduce the time to hit in the cache.

Cache Mapping:
There are three different types of mapping used for the purpose of cache memory which are as follows: Direct mapping, Associative mapping, and Set-Associative mapping. These are explained below.

1. **Direct Mapping –**
   The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. In Direct mapping, assign each memory block to a specific line in the cache. If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field. The cache is used to store the tag field whereas the rest is stored in the main memory. Direct mapping`s performance is directly proportional to the Hit ratio.
   ```
   i = j modulo m
   ```

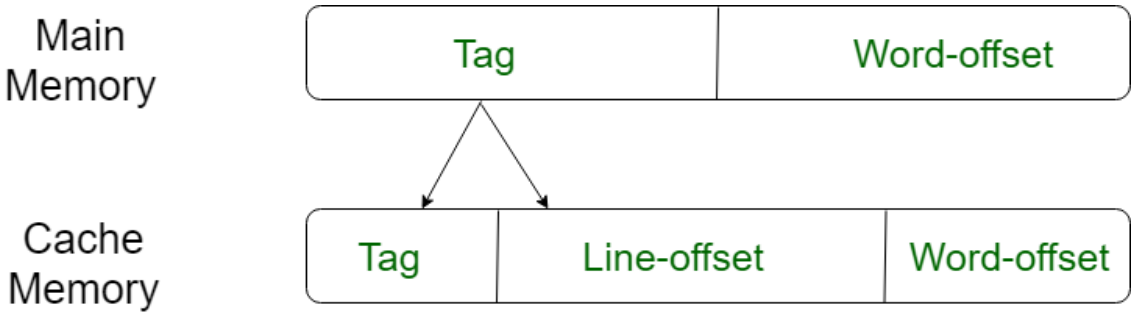
where

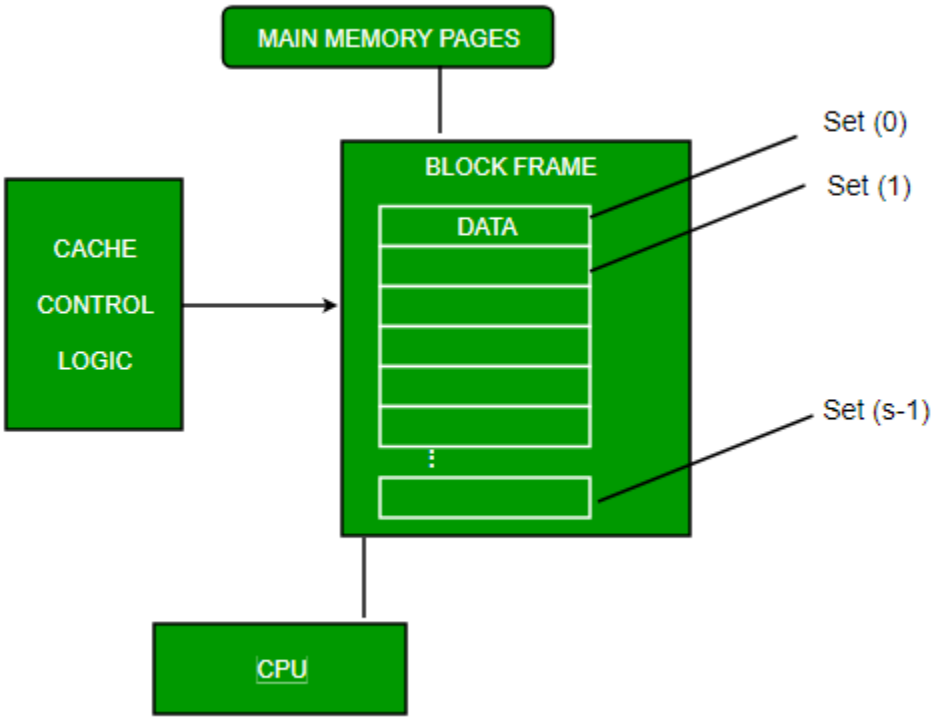
i=cache line number

j= main memory block number

m=number of lines in the cache

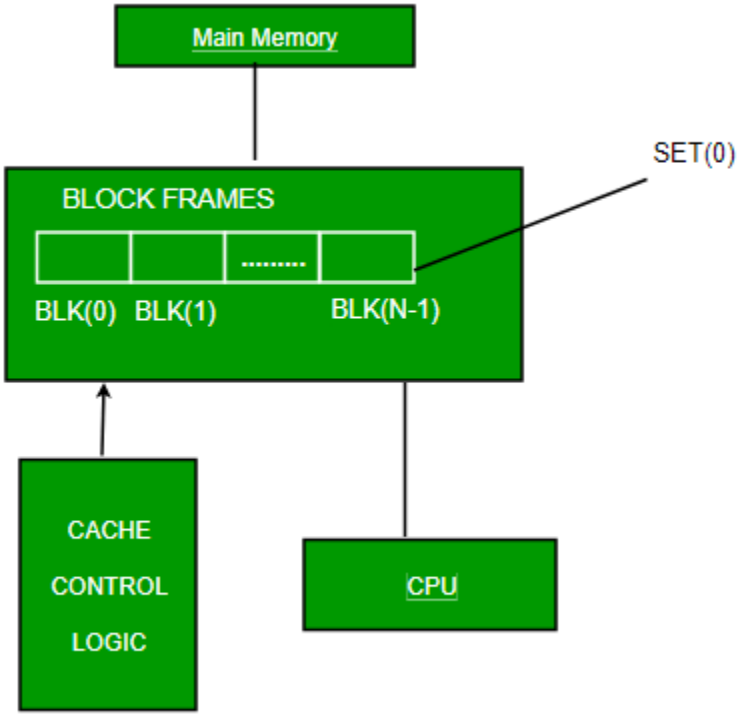
For purposes of cache access, each main memory address can be viewed as consisting of three fields. The least significant w bits identify a unique word or byte within a block of main memory. In most contemporary machines, the address is at the byte level. The remaining s bits specify one of the 2^s blocks of main memory. The cache logic interprets these s bits as a tag of s-r bits (most significant portion) and a line field of r bits. This latter field identifies one of the $m=2^r$ lines of the cache.





2. **Full Associative Mapping**
   In this type of mapping, the associative memory is used to store content and addresses of the memory word. Any block can go into any line of the cache. This means that the word id bits are

used to identify which word in the block is needed, but the tag becomes all of the remaining bits. This enables the placement of any word at any place in the cache memory. It is considered to be the fastest and the most flexible mapping form.

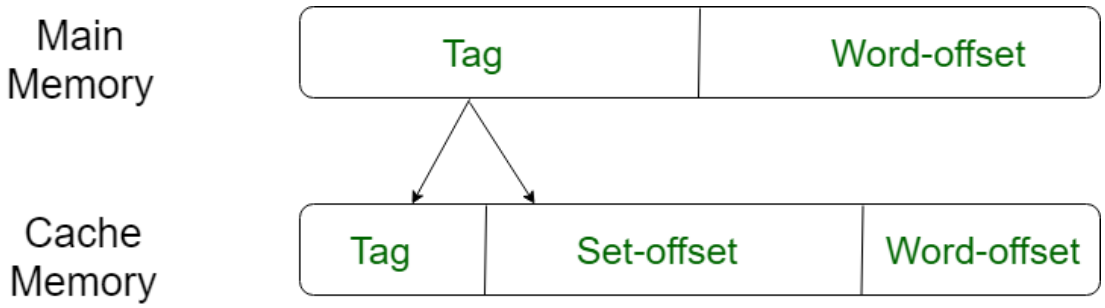


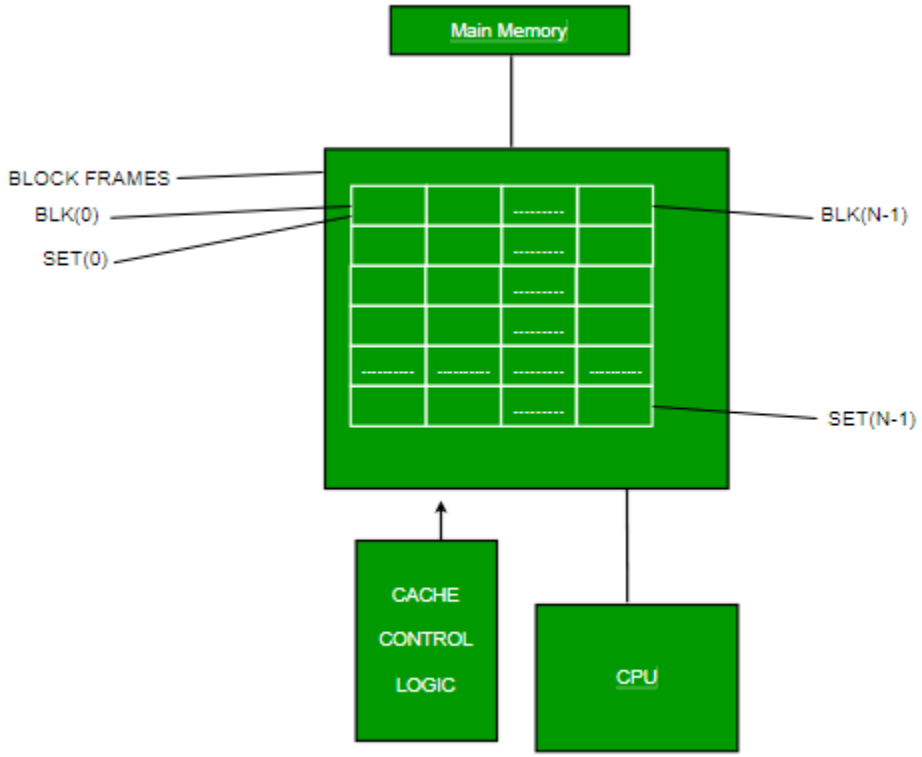
3. **Set-associative Mapping**
   This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are removed. Set associative addresses the problem of possible thrashing in the direct mapping method. It does this by saying that instead of having exactly one line that a block can map to in the cache, we will group a few lines together creating a ***set***. Then a block in memory can map to any one of the lines of a specific set. Set-associative mapping allows that each word that is present in the cache can have two or more words in the main memory for the same index address. Set associative cache mapping combines the best of direct and associative cache mapping techniques.
   In this case, the cache consists of a number of sets, each of which consists of a number of lines. The relationships are

```
m = v * k

i= j mod v


where

i=cache set number

j=main memory block number

v=number of sets
```

m=number of lines in the cache number of sets

k=number of lines in each set





Application of Cache Memory –

1. Usually, the cache memory can store a reasonable number of blocks at any given time, but this number is small compared to the total number of blocks in the main memory.
2. The correspondence between the main memory blocks and those in the cache is specified by a mapping function.

Replacement algorithm

- When all lines are occupied, bringing in a new block requires that an existing line be overwritten.

Direct mapping

- No choice possible with direct mapping
- Each block only maps to one line
- Replace that line

Associative and Set Associative mapping

- Algorithms must be implemented in hardware for speed
- Least Recently used (LRU)
    - replace that block in the set which has been in cache longest with no reference to it
    - Implementation: with 2-way set associative, have a USE bit for each line in a set. When a block is read into cache, use the line whose USE bit is set to 0, then set its USE bit to one and the other line's USE bit to 0.
    - Probably the most effective method
- First in first out (FIFO)
    - replace that block in the set which has been in the cache longest
    - Implementation: use a round-robin or circular buffer technique (keep up with which slot's "turn" is next
- Least-frequently-used (LFU)
    - replace that block in the set which has experienced the fewest references or hits
    - Implementation: associate a counter with each slot and increment when used
- Random
    - replace a random block in the set
    - Interesting because it is only slightly inferior to algorithms based on usage

Write policy

- When a line is to be replaced, must update the original copy of the line in main memory if any addressable unit in the line has been changed
- If a block has been altered in cache, it is necessary to write it back out to main memory before replacing it with another block (writes are about 15% of memory references)
- Must not overwrite a cache block unless main memory is up to date
- I/O modules may be able to read/write directly to memory
- Multiple CPU's may be attached to the same bus, each with their own cache

Write Through

- All write operations are made to main memory as well as to cache, so main memory is always valid
- Other CPU's monitor traffic to main memory to update their caches when needed
- This generates substantial memory traffic and may create a bottleneck
- Anytime a word in cache is changed, it is also changed in main memory
- Both copies always agree
- Generates lots of memory writes to main memory
- Multiple CPUs can monitor main memory traffic to keep local (to CPU) cache up to date
- Lots of traffic
- Slows down writes
- Remember bogus write through caches!

Write back

- When an update occurs, an UPDATE bit associated with that slot is set, so when the block is replaced it is written back first
- During a write, only change the contents of the cache
- Update main memory only when the cache line is to be replaced
- Causes “cache coherency” problems -- different values for the contents of an address are in the cache and the main memory
- Complex circuitry to avoid this problem
- Accesses by I/O modules must occur through the cache
- Multiple caches still can become invalidated, unless some cache coherency system is used. Such systems include:
    - Bus Watching with Write Through - other caches monitor memory writes by other caches (using write through) and invalidates their own cache line if a match
    - Hardware Transparency - additional hardware links multiple caches so that writes to one cache are made to the others
    - Non-cacheable Memory - only a portion of main memory is shared by more than one processor, and it is non-cacheable