

STUDY MATERIALS ON COMPUTER ORGANIZATION

(As per the curriculum of Third semester B.Sc.
Electronics of Mahatma Gandh University)

Compiled by Sam Kollannore U..
Lecturer in Electronics
M.E.S. College, Marampally

4. THE MEMORY

Programs and data (on which they operate) are held in the main memory during execution. Execution speed of the programs depends on the speed with which instructions and data can be transferred between the CPU and the main memory.

Ideally the main memory would be;

- Fast
- Large and
- Inexpensive

But it is impossible to meet all the requirements simultaneously
} Refer topic 2.2

4.1 Basic Concepts

- The max. size of the memory that can be used in any computer is determined by the addressing scheme.
Eg. 16 bit address is capable of addressing upto $2^{16} = 64K$ memory locations
32 bit address can access $2^{32} = 4G$ memory locations etc.
- Most modern computers are byte-addressable. Two types of addressing assignments are
 - 1) Big-Endian Assignment and
 - 2) Little-Endian Assignment
- The main memory is designed to store and retrieve data in word length quantities. In fact the word length of a computer is defined as the number of bits actually stored and retrieved in one main memory access.
For eg. In a byte-addressable computer, generating 32 bit address from CPU to the main memory unit, high-order 30-bits determine which word will be accessed and the low-order 2-bits specifies which byte location is involved.
- Data transfer between the memory and the CPU takes place through the use of two CPU registers usually called MAR and MDR. If MAR is k-bits long and MDR is n-bits long, then the memory unit may contain upto 2^k addressable location and during a memory cycle n-bits of data are transferred between memory and CPU. This transfer takes place over the processor bus which has k address lines and n data lines. Bus also includes Control lines – Read, Write and Memory Function Complete (MFC) for co-ordinating data transfers. In a byte-addressable computer, another control may be added to indicate whenever only a byte rather than a full word of n bits is to be transferred.

- CPU initiates a memory operation by loading the appropriate data into MDR and MAR – then setting either the Read or Write memory control line to 1. When the required operation is completed, the memory control circuitry indicates this to the CPU by setting MFC to 1.

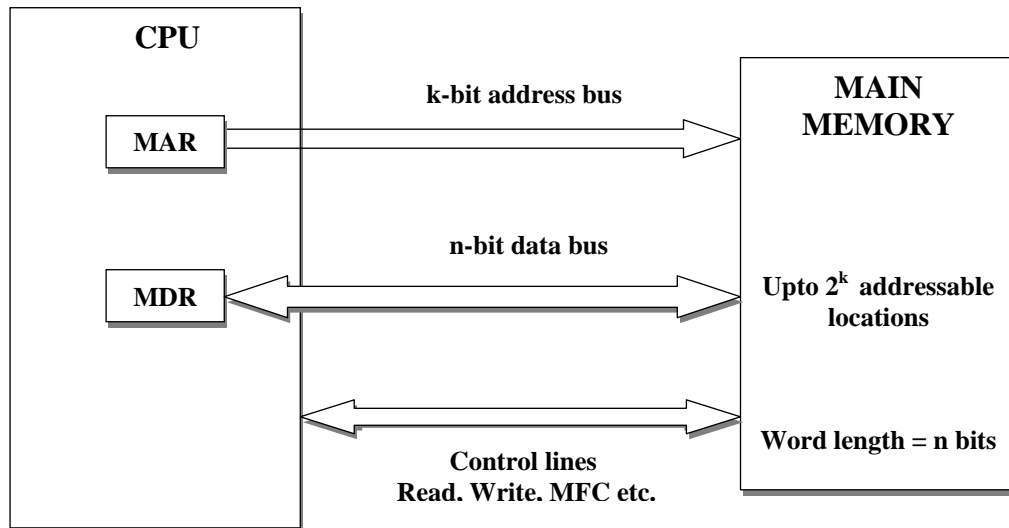


Fig 4.1

- Speed of the memory unit is measured by
 1. **Memory Access Time:-** time elapsed between the initiation of an operation and the completion of that operation (time between the Read and the MFC signals)
 2. **Memory Cycle Time:-** minimum time delay required between the initiation of two successive memory operations (time between two successive Read operation) – slightly longer than memory access time

Basic technology for implementing main memories uses semiconductor integrated circuits. The CPU of a computer can usually process instruction and data faster than they can be fetched from the main memory unit. So we have to reduce the memory access time. There are different ways to reduce the memory access time viz. using Cache memory / Memory interleaving / Virtual memory.

1. **Cache memory:-** small, fast memory that is inserted between the larger, slower main memory and the CPU – holds the currently active segments of a program and their data.
2. **Memory interleaving:-** divides the system into a number of memory modules and arranges addressing so that successive words in the address-space are placed in different modules. Since parallel access to these modules is possible, the average rate of fetching words from the main memory can be increased.
3. **Virtual memory:-** Until now we have assumed that, the addresses generated by the CPU directly specifies physical locations in the main memory. This may not be the case – Data may be stored in physical memory locations that have addresses different from those specified by the program. The memory control circuitry translates the address specified by the program into an address that can be used to access the physical memory. In this case, the address generated by the CPU is referred to as a *virtual or logical address*. The virtual address space is mapped onto the physical memory where data are actually stored. The mapping function is implemented by a special memory control circuit called Memory Management Unit. The mapping function can be changed during program

execution according to system requirements. Thus virtual memory is used to increase the apparent size of the main memory.

4.2 SEMICONDUCTOR RAM MEMORIES

- Semiconductor memories are available in a wide range of speeds – cycle time ranges from a few hundred nanoseconds (ns) to less than 10 ns.
- Uses VLSI technology – thus cost is reduced

4.2.1 Internal Organization

Memory cells are usually organized in the form of an array in which each cell is capable of storing one bit of information. Each row of cells constitutes a memory word and all cells of a row are connected to a common line referred to as word line, which is driven by the address decoder on the chip. The cells in each column are connected to a Sense/Write circuit by two bit lines. The sense/write circuits are connected to the data input/output lines of the chip. During a Read operation, these circuits sense or read the information stored in the cells selected by a word line and transmit this information to the output data lines. During a write operation, the sense/write circuits receive input information and store it in the cells of the selected word.

Figure shows an example of a very small memory chip consisting of 16 words of 8 bits each—referred to as 16×8 organization. Data i/p and data o/p of each sense/write circuit are connected to a single bidirectional data line in order to reduce the number of pins.

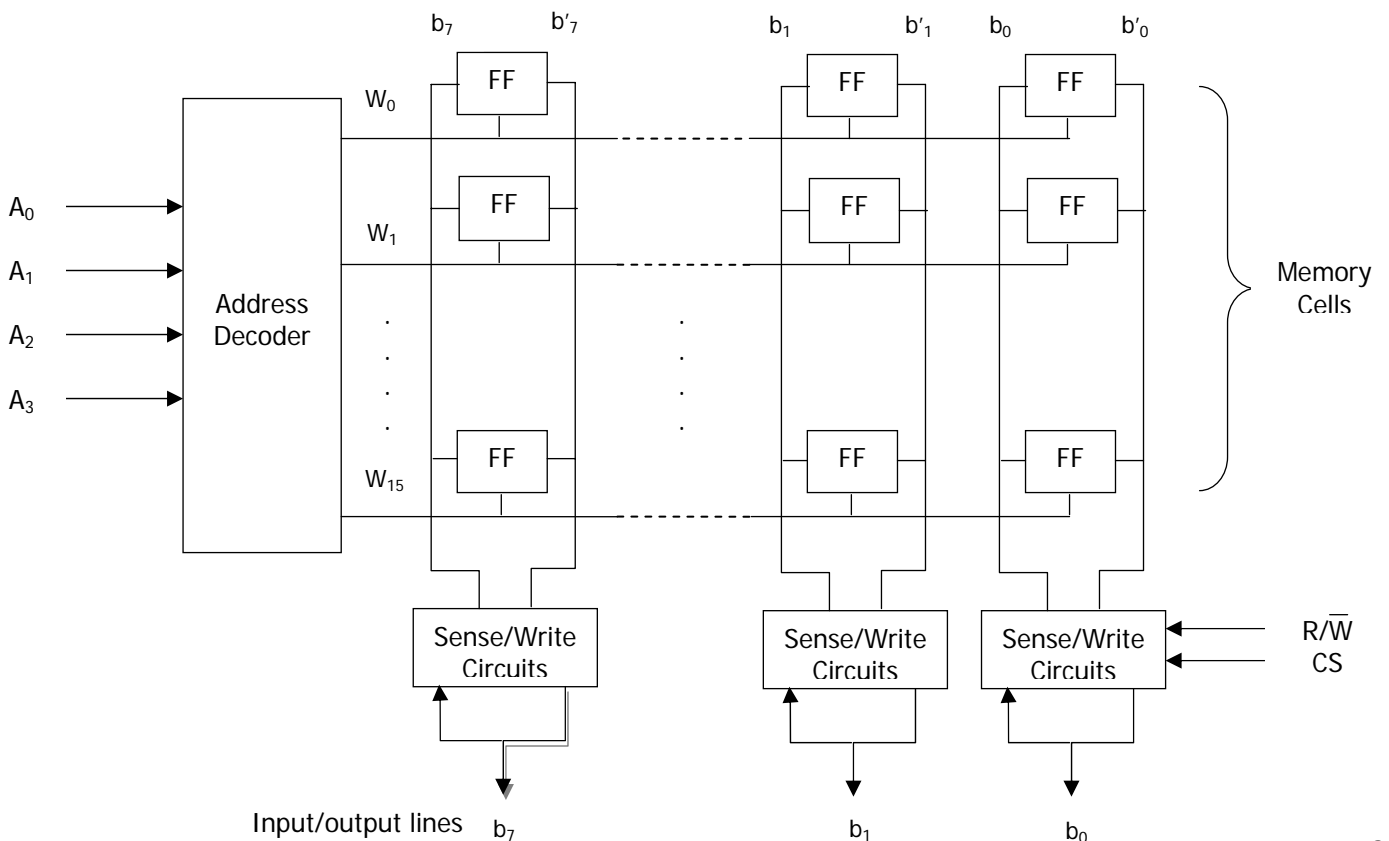


Fig 4.2

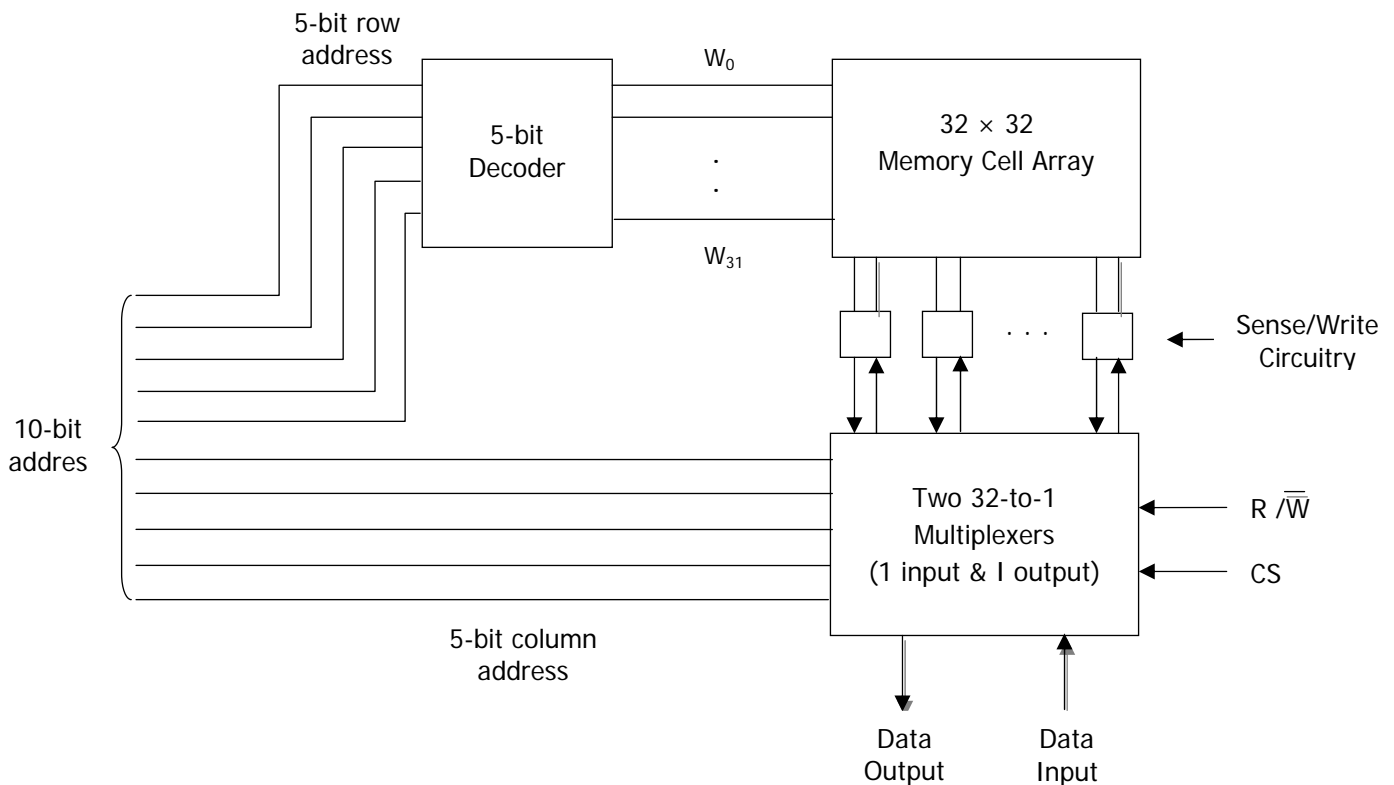


Fig 4.3

Two control lines R/\bar{W} and CS are provided in addition to the address and the data lines. R/\bar{W} input specifies the required the operation and the CS (Chip Select) input select a given chip in a multi-chip memory system. Thus stores 128 bits and requires 14 external connections. It is manufactured in the form of 16 pin chip (2 pins for power supply and ground connections).

Consider a 1K format:-

Case (i): This can be organized as a 128×8 memory chip – requiring total 19 pins.

Case (ii): Same number of cells can be organized into $1K \times 1$ format. Now we have to use only 16 pin chip (even if separate pins are provided for the data input and data output lines)

Case (iii): Using 32×32 format: Here the required 10 bit address is divided into two groups of 5 bits each to form row and column address for the cell array. Row address selects a row of 32 cells. According to column address one of the cells is connected to the external data lines by input and output multiplexers.

Note :- Larger chips have essentially the same organization but use a larger memory cell array, more address inputs and more pins. For example:- a 4M-bit chip may have a $1M \times 4$ organization with 20 address and 4 data input/output pins.

4.3 STATIC AND DYNAMIC MEMORIES

4.3.1 STATIC MEMORIES

Memories that consist of circuits that are capable of retaining their state as long as the power is applied are known as Static Memories – eg. Static RAM (SRAM).

Two inverters are cross linked to form a latch. The latch is connected to two bit lines by two transistors T1 and T2. These transistors act as switches that can be opened or closed under the control of the word line. When the word line is at the ground level, the transistors are turned off and the latch retains its state.

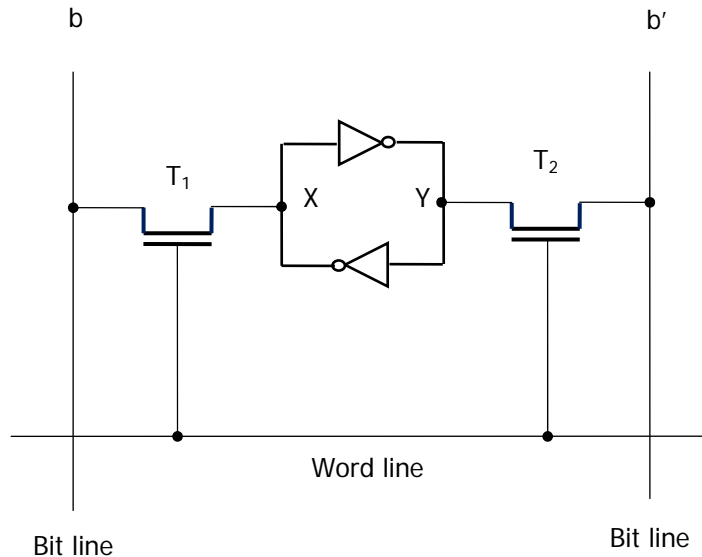


Fig 4.4 A Static RAM Cell

For example, let us assume that the cell is in state 1 if the logic value at point X is 1 and at point Y is 0. This state is maintained as long as the signal on the word line is at ground level.

Read operation:- In order to read the state of the SRAM cell, the word line is activated to close switches T1 and T2. If the cell is in state 1, the signal on bit line b is high and the signal on the bit line b' is low. If the cell is in state 0, then $b = 0$ & $b' = 1$. Sense/write circuits at the end of the bit lines monitor the state of b and b' and set the output accordingly.

Write operation:- The state of the cell is set by placing the appropriate value on the bit line b and its complement on b' and then activating the word line. This forces the cell into the corresponding state. The required signals on the bit lines are generated by the sense/write circuits.

4.3.2 CMOS Cell

CMOS realization of the above cell is given below. Transistor pairs (T3, T5) and (T4, T6) form the inverters in the latch. The state of the cell is read or written as explained above.

Read operation:- In state 1, the voltage at point X is maintained high by having transistors T3 and T6 ON while T4 and T5 are OFF. Thus if T1 and T2 are turned ON (closed), bit lines b and b' will have high and low signals respectively.

Write operation:- Activate the word line; let us apply $b=1$ and $b'=0$ by the sense/write circuits connected to the bit lines. This makes the transistors T1 ON and T2 OFF; thereby making T6 & T3 ON and T5 & T4 OFF – thus maintaining $X=1$ and $Y=0$.

The power supply voltage V_{supply} is 5V in standard CMOS SRAMs or 3.3V in low-voltage versions. Note that continuous power is needed for the cell to retain its state. If the power is interrupted, the cell's contents will be lost. When the power is restored, the latch will settle into a stable state, but it may not be the state before the interruption.

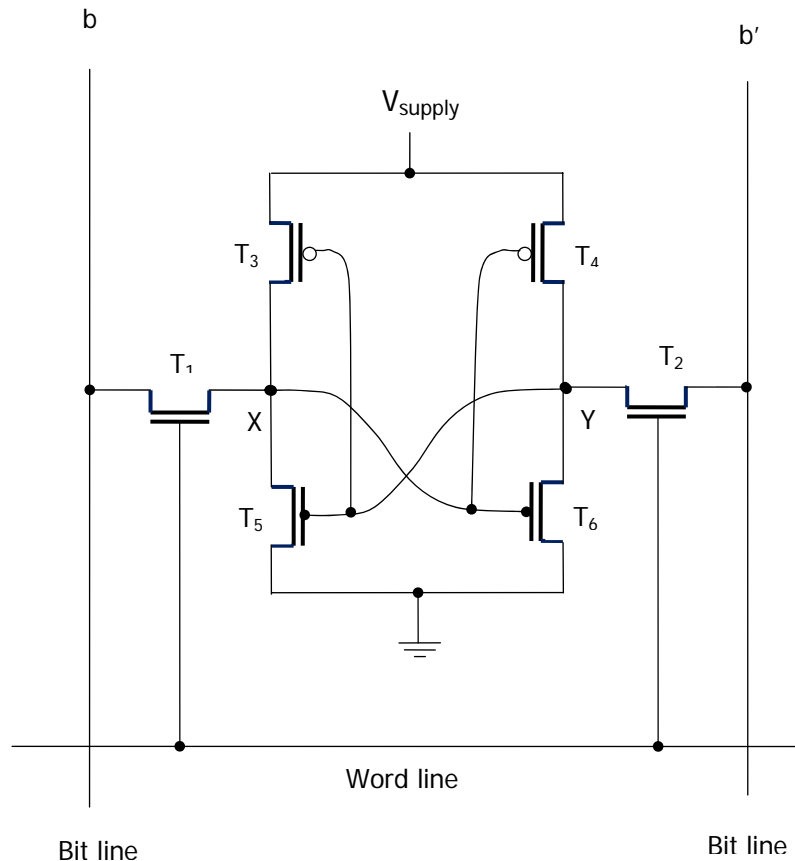


Fig 4.5 An example of a CMOS memory cell

Advantages:

1. Very low power consumption because current flows in the cell only when the cell is being accessed. Otherwise T1, T2 and one transistor in each inverters are turned off ensuring that there is no active path between V_{supply} and ground
2. Static RAMs can be accessed very quickly. Access time $< 10ns$

Drawbacks:

1. High cost because their cells require several transistors

4.3.3 DYNAMIC MEMORIES

- Less expensive RAMs
- Simpler cells are used
- Do not retain their state indefinitely – hence they are called dynamic RAMs (DRAMs)
- Information is stored in the form of a charge on a capacitor. Thus DRAM is capable of storing information for only a few milliseconds. Since each cell is usually required to store information for a much longer time, its contents must be periodically refreshed by restoring the capacitor charge to its full value

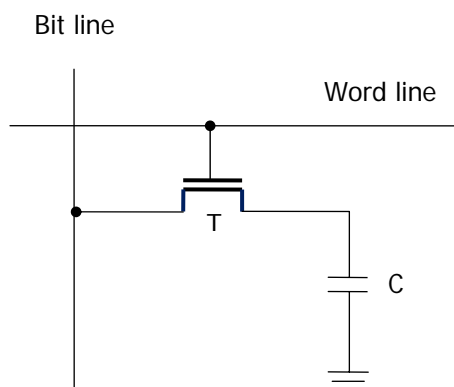


Fig 4.6

A dynamic memory cell consisting of a transistor T and a capacitor C is as shown. In order to store information in this cell, transistor T is turned ON and an appropriate voltage is applied to the bit line. This causes a known amount of charge to be stored on the capacitor. After the transistor is turned OFF, the capacitor begins to discharge because of capacitor's own leakage resistance and the fact that the transistor continues to conduct a tiny amount of current (in Pico amperes) after it is turned OFF. Hence the information stored in the cell can be retrieved correctly, only if it is read before the charge on the capacitor drops below some threshold value.

During a read operation, the bit line is placed in a high impedance state and the transistor is turned ON. A sense circuit connected to the bit line determines whether the charge on the capacitor is above or below the threshold value.

Note: The Read operation discharges the capacitor in the cell that is being accessed. In order to retain the information stored in the cell, DRAM includes a special circuitry that writes back the value that has been read. A memory cell (& all cells connected to a given word line) is therefore refreshed when every time its contents are read.

Example:- A typical 1MB DRAM chip configured as 1M×1

- Organized in the form of a 1K×1K array such that the high and low order 10 bits of the 20-bit address constitutes the row and column address of a cell respectively. To reduce the number of pins needed for external connections, the row and the column addresses are multiplexed on 10 pins. During a Read or Write operation, the row address is applied first. It is loaded into the row address latch in response to a signal pulse on the Row Address Strobe (RAS) input of the chip. Then a Read operation is initiated in which all cells on the selected row are read and refreshed.

Shortly after the row address is loaded, the column address is applied into the column address latch under the control of the Column Address Strobe (CAS) signal. The information in this latch is decoded and the appropriate sense/write circuit is selected. If the $\overline{R/W}$ control signal indicates a Read operation, then the output of the selected circuit is transferred to the data output D_o . For a Write operation, the information at the data input D_i is transferred to the selected circuit.

Refresh Circuitry:- To ensure that the contents of a DRAM are maintained, each row of cells must be accessed periodically, typically once every 2 to 16 milliseconds. A refresh circuit usually performs this function automatically. Some dynamic memory chips contain a refresh facility within the chip itself – such chips are often referred to as pseudostatic since the dynamic nature of these memory chips are invisible to the user.

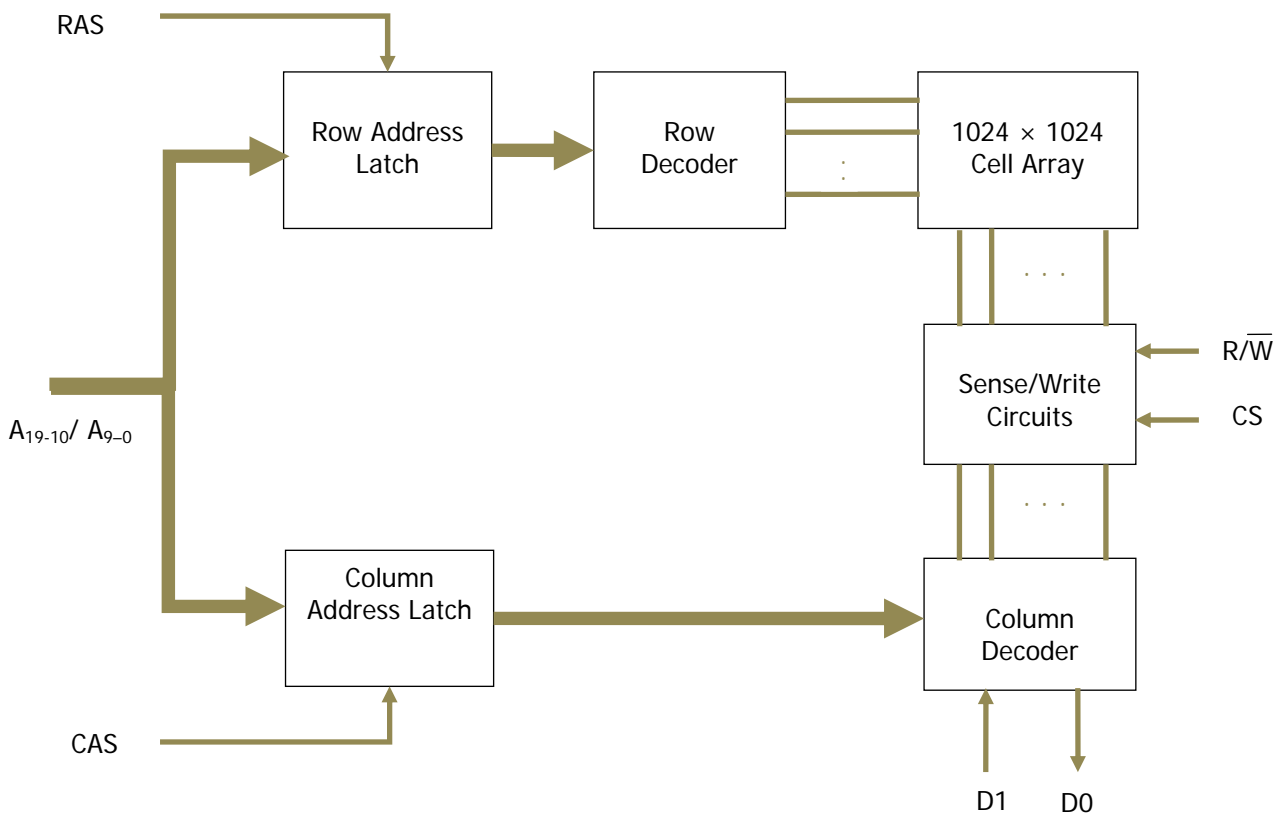


Figure 4.7 Internal organization of a 1Mx1 dynamic memory chip

Advantages:-

1. High Density & low cost Therefore widely used in the main memory units of computers
2. Wide capacity ranges: 1K to 16Mbits or larger
3. Flexibility in design: 4M chip may be organized as 4Mx1, 1Mx4, 512Kx8 or 256Kx16
4. Blocks of data within any row can be transferred by loading only the row address. This feature is used for transferring data blocks between the main memory and a cache.

4.4 Memory System Consideration

Factors affecting the choice of a RAM chip for a given application are

- i) Speed
- ii) Power dissipation
- iii) Size & Cost of the chip
- iv) Availability of block transfers (in certain situations)

Static RAMs are used where every fast operation is required. But they are costly and their size is large. DRAMs are a good choice for implementing in computer main memories – high densities are achieved with low cost.

Design of memory systems

a) Using SRAMs

Consider a small memory of 64K (65536) words of 8 bits each, organized using 16K×1 static memory chips. Each column consists of 4 chips implementing one bit position. 8 of these sets provide 64K×8 memory. Each chip has a control input CS. When this input is set to 1, it enables that chip to accept that data input or to place the data on its output line. The address bus is 16 bit wide and the higher order 2 bits are decoded to obtain the four chip select (CS) control signal. Remaining 14 bits are used to access specific bit locations. R/W inputs of all chips are tied together to provide a common Read/Write control (not shown in figure)

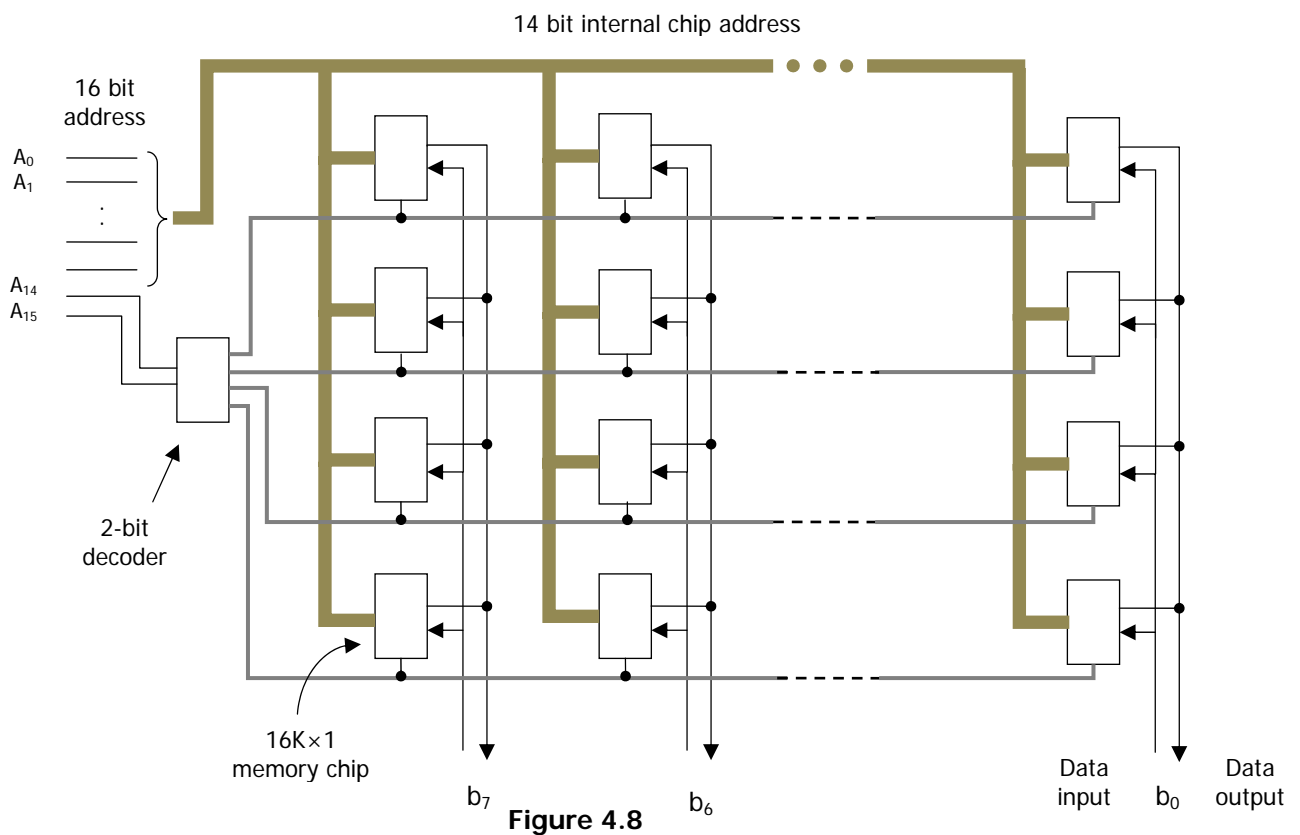


Figure 4.8

b) Using DRAMs

This organization differs in three aspects.

- 1) Row and Column parts of the address have to be multiplexed
- 2) A refresh circuit is needed.
- 3) Timing of various steps of a monitoring cycle must be carefully controlled.

Consider a dynamic memory unit- arranged in a 4×8 array format. (similar to SRAM array as shown above) The individual chips have a 1M×4 bits organization. Thus the total storage capacity is 4M words of 32 bits each

The control circuitry provides the multiplexed address and Chip Select inputs and sends the RAS and CAS signals to the memory chip array. The circuitry generates refresh cycles as needed.

The memory unit is assumed to be connected to an asynchronous memory bus that has 22 address lines (ARS_{21-0}), 32 data lines ($DATA_{31-0}$), two handshake signals (Memory Request and MFC) and a Read/Write line to indicate the type of memory cycle requested.

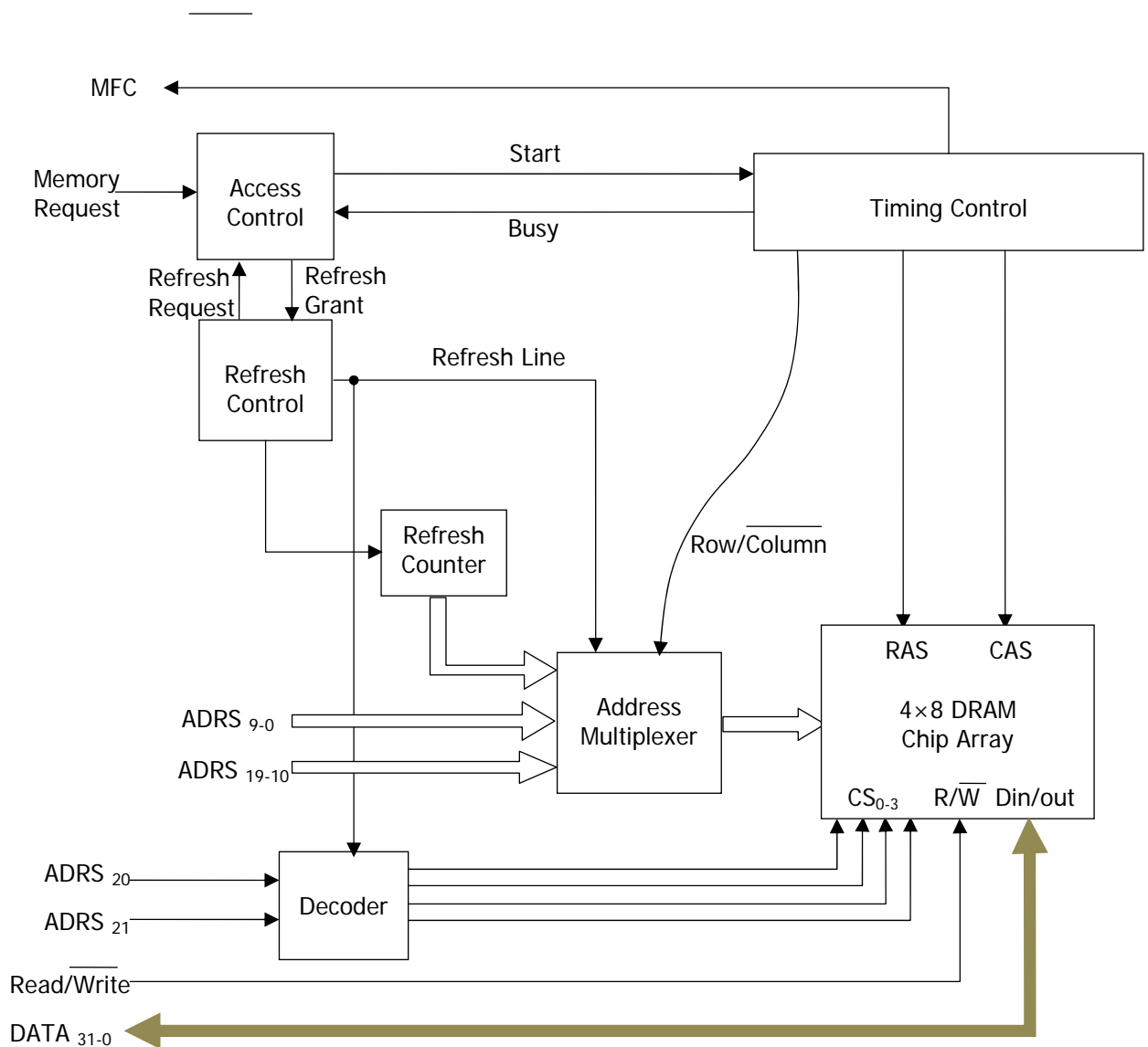


Fig 4.9 Block diagram of a $4M \times 32$ memory unit using $1M \times 4$ DRAM chips

The cycle begins when the CPU activates the address, the Read/Write and the Memory Request lines. The Access Control block recognizes the request and it sets the Start signal to 1. The Timing Control block responds immediately by activating the Busy signal, to prevent the Access control block from accepting new requests before the cycle ends. The Timing Control block then loads the row and column addresses into the memory chips by activating the RAS and CAS lines. During this time, it uses the Row/ Column line to select first the row address, $ADRS_{19-10}$, followed by the column address $ADRS_{9-0}$. The Decoder decodes the address as per the input lines. It also decodes the two most significant bits of the address, $ADRS_{21-20}$ and activates one of the chip select lines CS_{3-0} . After obtaining the row and column parts of the address, the selected memory chips place the contents of the requested bit cells on their data outputs. This information is transferred to the data lines of the

memory bus via appropriate drivers. The timing control block then activates the MFC lines indicating the requested data are available on the memory bus. At the end of the memory cycle, the Busy signal is deactivated and the access unit becomes free to accept new requests.

Refresh operation: The Refresh Control block periodically generates Refresh requests causing the Access control block to start a memory cycle in the normal way. The access control block indicates to the refresh control block that it may proceed with a refresh operation by activating the Refresh Grant line. The access control block checks the Memory access requests and Refresh requests. If these requests arrive simultaneously, Refresh requests are given priority to ensure that no information is lost. As soon as the refresh control block receives the refresh grant signal, it activates the refresh line. This causes the Address Multiplexer to select the row address from the Refresh Counter instead of the external address lines. (Refresh counter keeps track of the address of the row to be refreshed) This row address is loaded into the row address latches of all memory chips when the RAS signal is activated.

[Note: During this time, the R/W line of the memory bus may indicate a write operation. We must ensure that this does not cause new information to be loaded into some of the cells that are being refreshed. We can prevent this in several ways. One way is to have the decoder block deactivate all CS lines to prevent the memory chips from responding to the R/W line.]

The remaining of the refresh cycle is then the same as a normal cycle. At the end, the Refresh Control block increments the Refresh counter in preparation for the next refresh cycle so that each row in the memory array is refreshed every 15 μ S. The main purpose of the refresh circuit is to maintain the integrity of the stored information.

4.5 READ ONLY MEMORIES

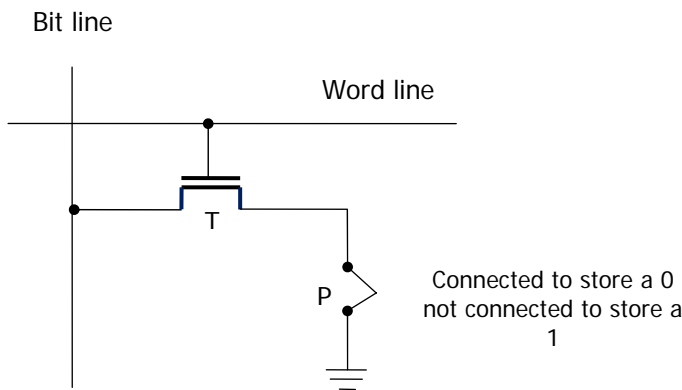


Fig 4.10 A ROM cell

A logic 0 is stored in the cell if the transistor is connected to ground at point P. Otherwise a 1 is stored. The bit line is connected through a resistor to the power supply.

Read: To read the state of the cell, the word line is activated. Thus the transistor switch is closed. The voltage on the bit line drop to near zero if there is a connection between the transistor and ground. If there is no connection to ground, the bit line remains at the high voltage, indicating a 1. A sense circuit at the end of the bit line generates the proper output value.

Write: Data are written into a ROM when it is manufactured

ROMs are highly economical for storing fixed programs and data when high volumes of ROMs are produced. But the cost of preparing the masks needed for storing a particular information pattern in ROMs makes them very expensive when only a small number are required.

PROM

Some ROM designs allow the data to be loaded by the user – they are called Programmable ROM. Programmability is achieved by inserting a fuse at point P. Before it is programmed, the memory contains all 0s. The user can insert 1s at the required locations by burning out the fuses at these locations using high-current pulses. The process is irreversible. PROMs are faster and less expensive because they can be programmed directly by the user.

EPROM

Erasable programmable ROMs allow the stored data to be erased and new data to be loaded. They are capable of retaining stored information for a long time. Therefore they can be used in place of ROMs. Memory changes and updates can be easily done.

Structure of an EPROM is similar to the ROM cell. The connection to ground is made at point P and a special transistor is used which has the ability to function either as a normal transistor or as a disabled transistor that is always turned OFF. This transistor can be programmed to behave as a permanently open switch, by injecting into it a charge that becomes trapped inside.

But the advantage is that their contents can be erased and reprogrammed. Erasing can be done by dissipating the charges trapped in the transistor of memory cells – by exposing the chip to ultraviolet light. For this reason, EPROM chips are mounted in packages that have transparent windows.

Drawbacks: Chip must be physically removed from the circuit for reprogramming and its entire contents are erased by the ultraviolet light.

EEPROM

- They can be programmed and erased electrically.
- Do not have to be removed for erasing.
- Possible to erase the cell contents selectively.
- **Drawback:** Different voltages are needed for erasing, writing and reading the stored data.

4.6 Speed, Size and Cost of memories

Ideal main memory would be fast, large and inexpensive. A very fast memory can be achieved by using SRAM chips. But these chips are expensive (because their basic cells have six transistors) and bulky. Alternative is to use DRAM chips (because they use simpler basic cells and are less expensive) But such memories are very slower. Although DRAMs allow main memories in the range of tens of megabytes, they are not sufficient compared to the demands of large programs with much data. This is solved by using secondary storage (mainly magnetic disks) to implement large memory spaces. However they are much slower than the main memory unit.

Anyway all of the above said types of memory units are employed effectively in a computer as shown below.

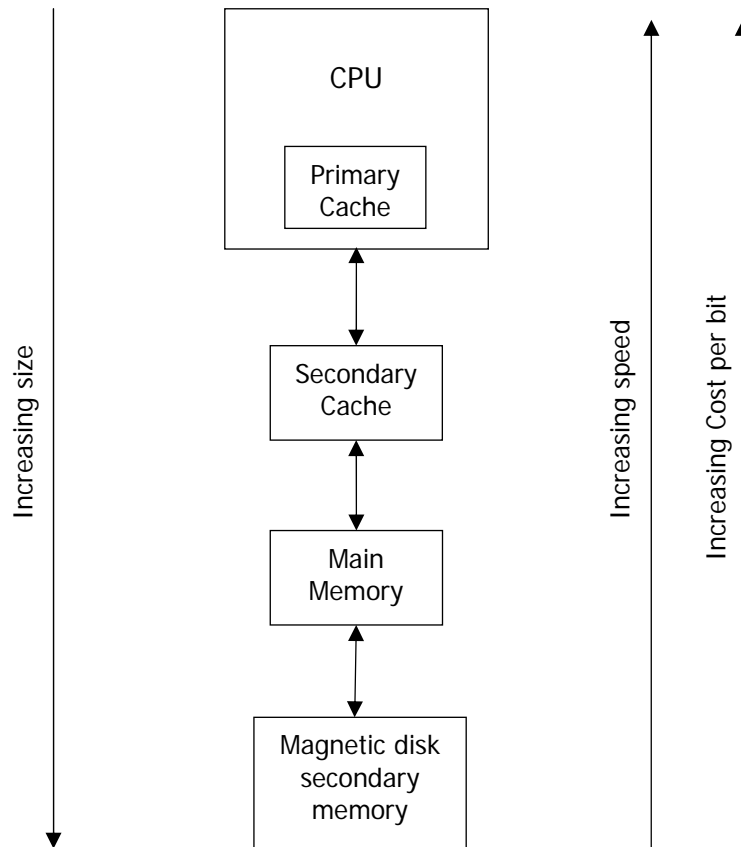


Fig 4.11 Memory hierarchy

- Two types of cache memory – Primary Cache & Secondary Cache
- Primary cache is located on the CPU chip – small
- Secondary cache – placed between the primary cache and the main memory – large
- It is possible not to have a cache on the processor chip at all.
- Also it is possible to have two levels of cache on the processor chip.

4.7 CACHE MEMORIES

To increase the speed of a processor, the time for accessing the information is to be reduced. For this a cache is used. The effectiveness of the cache is based on the property of programs called Locality of Reference.

Analysis of the programs show that most of their execution time is spent on routines in which many instructions in localized areas of the programs are executed repeatedly (These instructions may constitute simple loop, nested loop etc.) and the remainder of the program is accessed not so frequently. This is referred to as locality of reference. There are two concepts behind the locality of reference

- 1) Temporal
- 2) Spatial

Temporal means that a recently executed instruction is likely to be executed again very soon. A spatial aspect means that the instructions very close to a recently executed instruction (cache block/cache

line) are also to be executed soon. If the active segments of such a program or data can be placed in a cache, the execution time can be reduced. Consider the simple arrangement

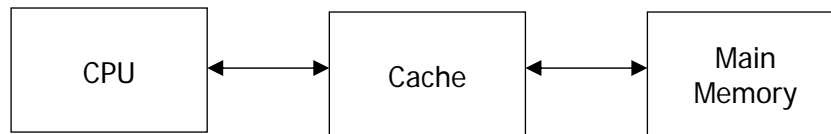


Fig 4.12

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 one word at a time. Later, when the program references any of the locations in this block, the desired contents are read directly from the cache. The number of blocks that can be stored in the cache memory is small compared to the total number of blocks in the memory. The correspondence between the main memory blocks and the blocks in the cache is specified by a **Mapping Function**. When the cache is full and a memory word (instruction/data) that is not in the cache is referenced, the cache control hardware must decide which block should be removed to create space for the new block that contains the referenced word. The collection of rules for making this decision constitutes the **Replacement Algorithm**.

The CPU does not need to know about the existence of the cache. It simply issues Read or Write requests using the addresses that refer to locations in the main memory. The cache control circuitry determines whether the requested word currently exists in the cache. If it is present, Read or Write operation is performed. – i.e. Read hit or Write hit is said to have occurred.

For a Write operation, the system can proceed in two ways

- 1) **Write-through protocol** – Cache location and the main memory location are updated simultaneously (simple but needs unnecessary write operation)
- 2) **Write-back / Copy-back protocol** – Update only the cache location and mark it as updated with an associated flag bit called as **dirty or modified bit**. The main memory word is updated later when the updated word is removed from the cache to make space for a new block.

During a Read operation, when the addressed word is not in the cache, a read-miss occurs. The block of words that contains the requested word is copied from the main memory to cache and then the requested particular word is forwarded to the CPU. Alternatively, this word may be sent to the CPU as soon as it is read from the main memory. This approach, called **Load-through** (or early restart) reduces the CPU's waiting period at the expense of more complex circuitry.

During a Write operation, if the addressed word is not in the cache, a write-miss occurs. Then if the Write-through protocol is used, the information is written directly into the main memory.

In the case of Write-back protocol, the block containing the addressed word is first brought into the cache and then the desired word in the cache is overwritten with the new information.

4.8 MAPPING FUNCTION

Consider a cache consisting of 128 blocks of 16 words each. i.e. total 2048 words. Assume that the main memory consists of 4096 blocks of 16 words each. i.e. total 64K words so that it is addressed by 16-bit address.

4.8.1 Direct Mapping

Simplest way to determine cache locations in which to store memory blocks is the direct mapping technique. Here block j of the main memory maps on to $block\ j\ modulo\ 128$ of the cache - which means that main memory blocks 0, 128, 256, ... are stored in cache block 0, main memory blocks 1, 129, 257, . . are stored in cache block 1 and so on.

Placement of a block in the cache is determined from the memory address. The memory address can be divided into three fields as shown.

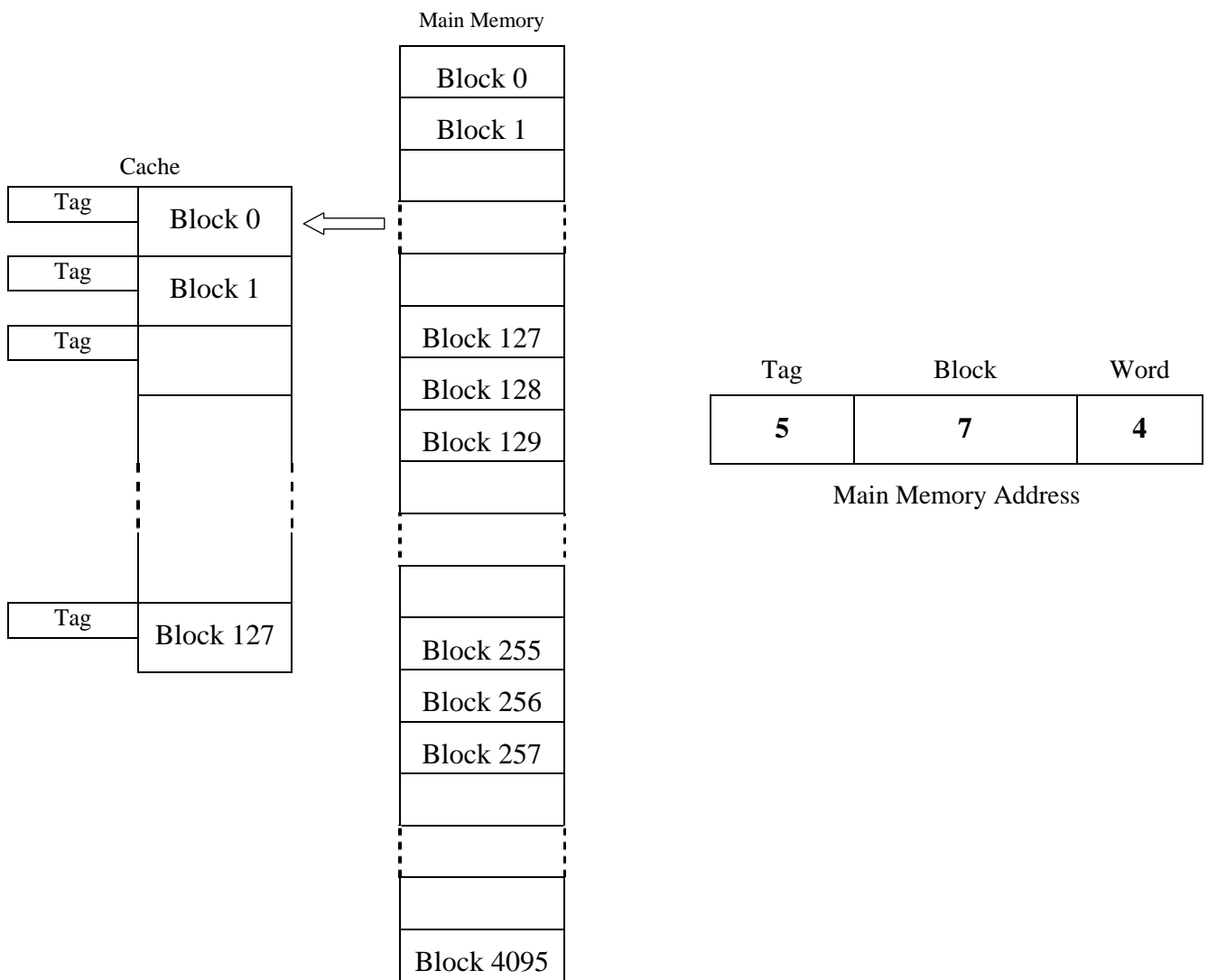


Fig 4.13

- Low order 4 bits select one of 16 words in a block
- Block field is given by 7 bits and determines the cache position in which this block must be stored
- Higher order 5 bits of the memory address are stored in 5 bit tag associated with its location in the cache. They identify which of the 32 blocks that are mapped into this cache position are currently

reside in the cache (because maximum 32 blocks of main memory are assigned to cache block of cache)

- During execution the 7 bit cache block field of each address generated by the CPU points to a particular block location in the cache. The higher order 5 bits of the address are compared with the tag bits associated with that cache location. If they match, then the desired word is in that block of the cache. If there is no match, then the block containing the required word must first read from the main memory and loaded into the cache.
- Easy to implement – but not very flexible.

4.8.2 Associative Mapping

- Much more flexible mapping method.
- Here a main memory block can be placed into any cache block position. There fore 12 tag bits are required to identify a memory block.
- Tag bits of an address received from the CPU are compared to the tag bits of each block of the cache to see if the desired block is present. This is called associative mapping.
- Space in the cache is used more effectively
- Cost of an associative cache is higher than the cost of a direct mapped cache, because of the need to search all 128 tag pattern to determine whether a given block is in the cache.

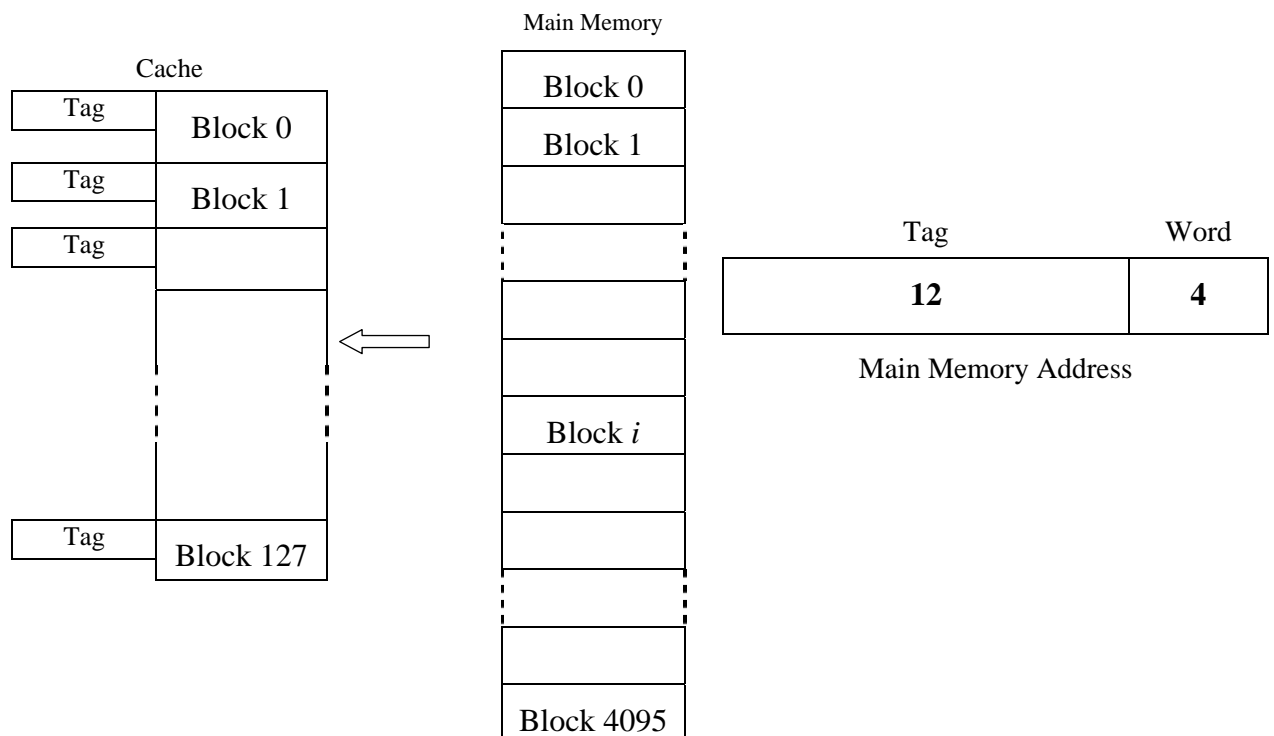


Fig 4.15

4.8.3 Set Associative Mapping

It is a combination of the direct mapping and associative mapping techniques. Here blocks of the cache are grouped into sets and allows a block of the main memory to reside in any of a specific set. Hence few choices for block placement are possible which were not available in direct mapping (i.e. contention problem is reduced). At the same time the cost for hardware of the control circuitry can be reduced.

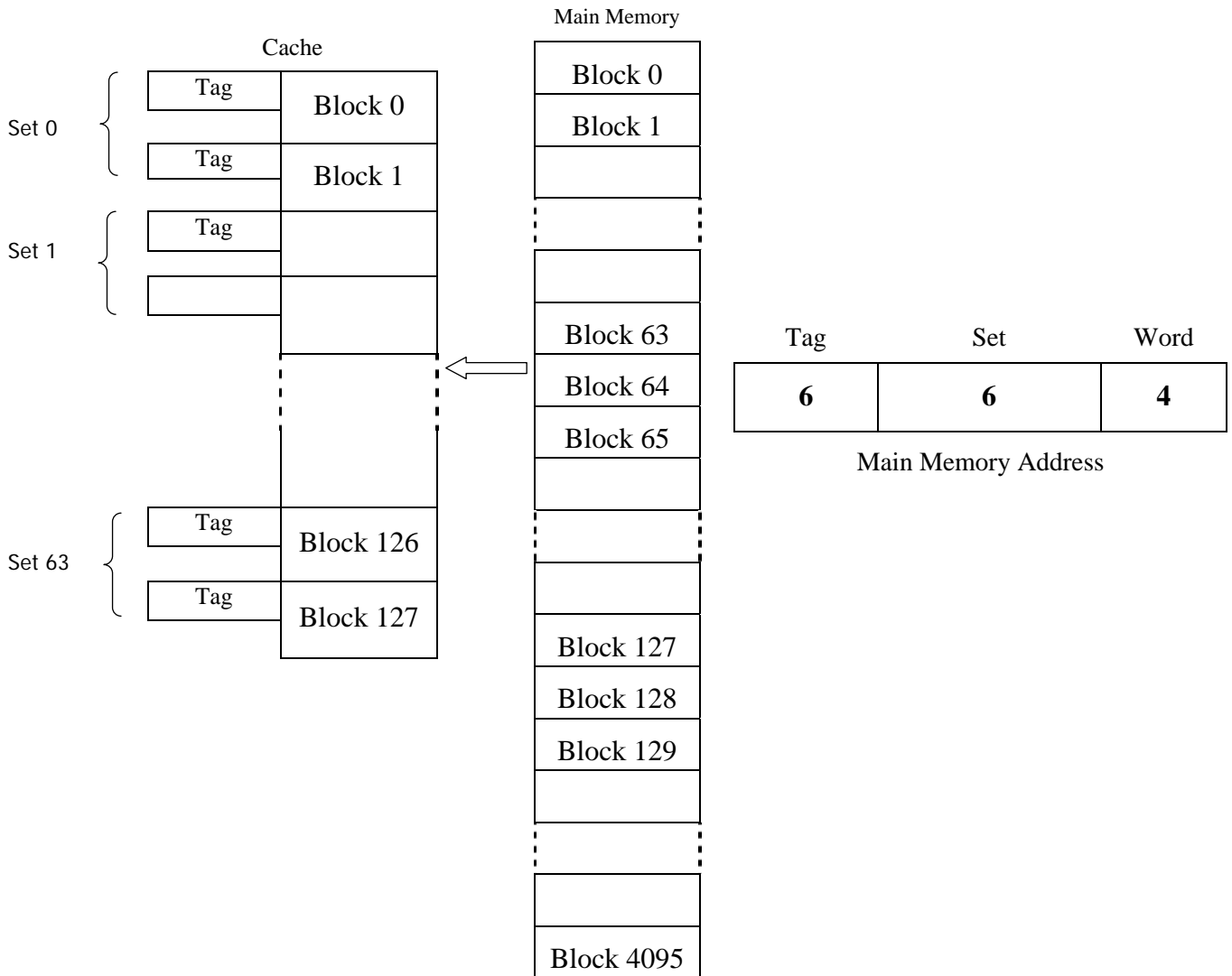


Fig 4.16 Set-Associative mapped cache with two blocks per set

- Consider the above example of Set associative mapping technique for a cache with 2 blocks per set. Therefore there are 63 sets. Main memory blocks 0, 64, 128,...4032 map into cache set 0. They can occupy either of the two block positions within the sets. 6-bit Set field of the address determines which set of the cache contains the desired block. The Tag field (6-bits) of the address must then be compared to the tags of the two blocks of the selected set to check whether the desired block is present.

Note:

- The number of blocks per set can be changed / selected to suit the requirements of a particular computer. 4 blocks per set can be accommodated by a 5-bit Set field. 8 blocks per set by a 4-bit Set field and so on.
- 12 blocks per set implies that there are no set bits i.e. fully associative technique.
- 1 block per set implies the direct mapping.
- One more control bit called Valid bit is provided for each block. – to indicate whether the block contains valid data.

4.9 REPLACEMENT ALGORITHM

In a direct mapped cache, the position of each block is predetermined. Hence no replacement algorithm is required. But in associative and set-associative mapping, replacement algorithm is required.

When a new block is to be transferred to the cache and the cache is full, the controller must decide which of the old blocks to overwrite. This decision is important because it will decide the system performance. The objective is that to keep blocks in the cache which is likely to be referenced in the near future. The property of locality of reference in programs gives a clue to this strategy. Generally a block which has not referenced for the longest time is overwritten in the cache. This block is called Least Recently Used (LRU) block and the technique is called **LRU Replacement Algorithm**.

To use the LRU algorithm, the cache controller must keep track to all blocks as the program proceeds. For a 4 block set in a set-associative cache, a 2-bit counter can be used for each block. When a hit occurs, the counter of the block that is referenced is set to 0. Counters with values originally lower than the referenced one are incremented by one and all others remain unchanged. When a miss occurs and the set is not full, the counter associated with the new block loaded from the main memory is set to 0 and values of all other counters are incremented by one. When a miss occurs and the set is full, the block with the counter value 3 is removed, the new block is put in its place and its counter is set to 0. The other 3 block counters are incremented by one.

- Used extensively
- Performs well in many cases but it can lead to poor performance in other cases
- Several other replacement algorithms are used. One rule is to remove the oldest block from the full set, when a new block is to be transferred – it is not much effective.
- The simplest algorithm is to randomly choose the block to be overwritten – very effective in practice.

4.10 MEMORY INTERLEAVING

Performance of a system depends on the speed of the system. Speed can be achieved by implementing parallelism in the organizations of slower units- thereby the data can be accessed from slower units at a speed which is equal to the speed at which the data is transferred from the faster units. An effective way to introduces parallelism is to use an interleaved memory organization.

Here the main memory is structured as a collection of separate modules - each module with its own Address Buffer Register (ABR) and Data Buffer Register (DBR). Memory access operations can be done in more than one module at the same time. Thus the total rate of transmission of words to and from the main memory system can be increased.

The way in which the individual addresses are distributed over the modules is critical for the number of modules that take part in the process. There are two methods.

I method:- The memory address generated by the CPU is decided as follows - Higher order k -bits name one of n modules and low order m -bits name a particular word in that module. When consecutive locations are accessed (as happens when a block of data is transferred to a cache) only one module is involved. At the same time, devices with DMA (Direct Memory Access) ability may access information in other memory modules.

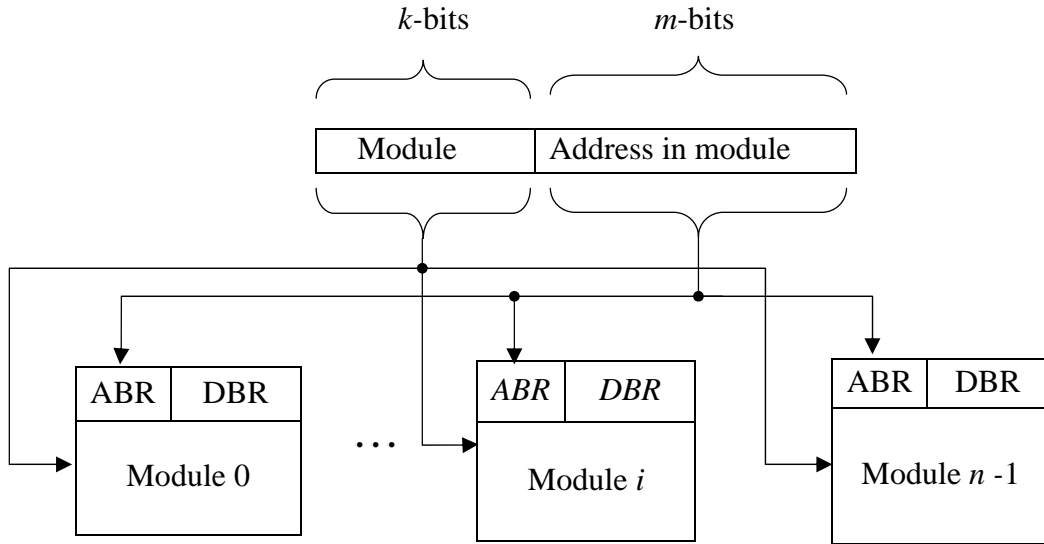


Fig 4.17 Consecutive words in a module

II Method – More effective – called *Memory Interleaving*

The low order k -bits of the memory address select a module and the high order m -bits name a location within that module. In this way consecutive locations are placed in successive modules. Thus when a system generates a request for accessing consecutive locations, several modules can be kept busy at one time. This results in faster access to a block of data and higher average utilization of the memory system. To implement the interleaved memory structure, there must be 2^k modules; otherwise there will be gaps of non-existent locations in the memory address space.

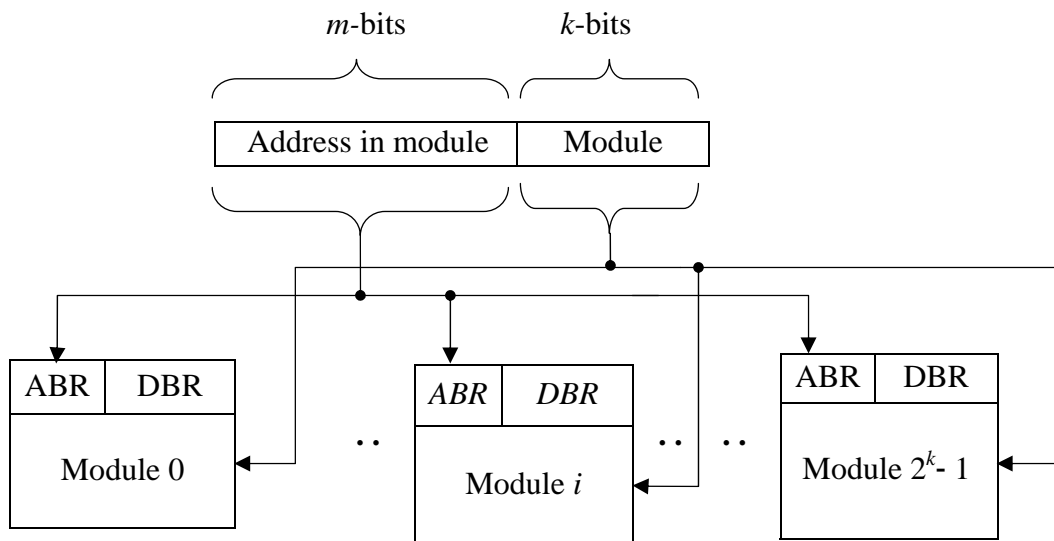


Fig 4.18 Consecutive words in consecutive modules