UNIT 4 Memory Systems

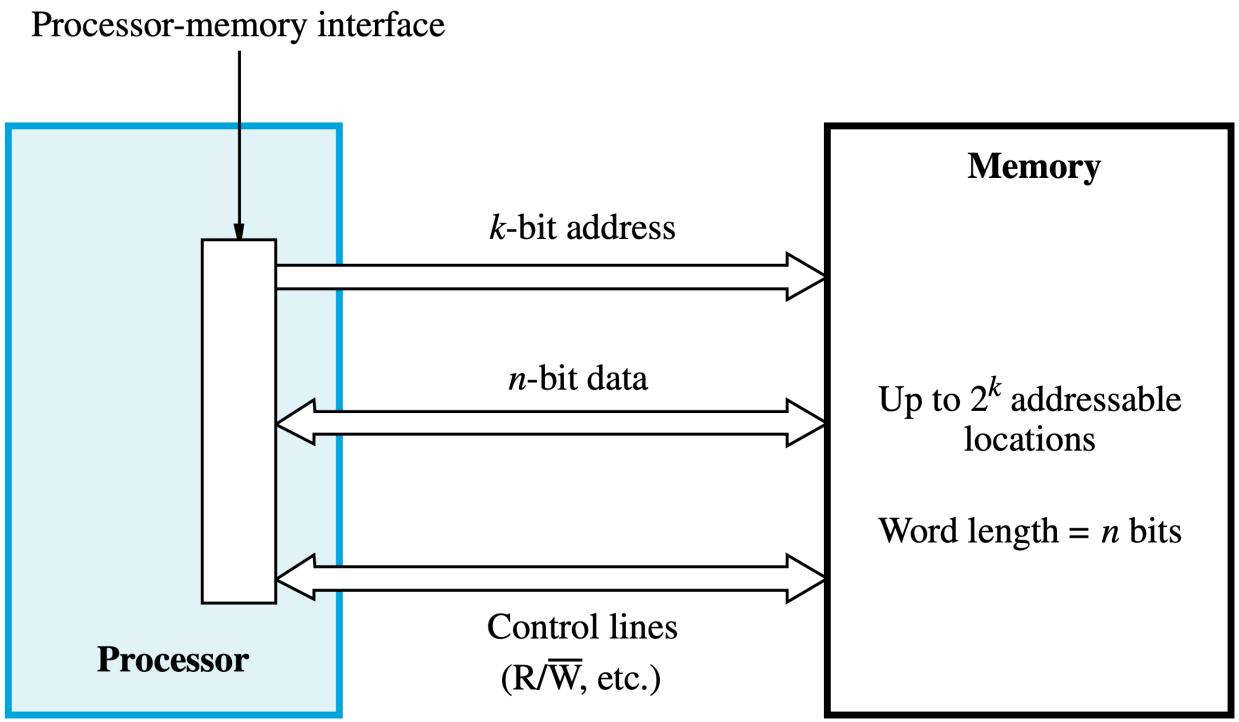


Figure 8.1 Connection of the memory to the processor.

Memory access time: A useful measure of the speed of memory units is the time that elapses between the initiation of an operation to transfer a word of data and the completion of that operation. This is referred to as the *memory access time*.

Memory cycle time: *it* is the minimum time delay required between the initiation of two successive memory operations.

Note: The cycle time is usually slightly longer than the access time, depending on the implementation details of the memory unit.

Random-access memory (RAM): if the access time to any location is the same, independent of the location's address. This distinguishes such memory units from serial, or partly serial, access storage devices such as magnetic and optical disks. Access time of the latter devices depends on the address or position of the data.

Cache Memory: The processor of a computer can usually process instructions and data faster than they can be fetched from the main memory. Hence, the memory access time is the bottleneck in the system. One way to reduce the memory access time is to use a ***cache memory***.

Virtual Memory is another important concept related to memory organization. With this technique, only the active portions of a program are stored in the main memory, and the remainder is stored on the much larger secondary storage device.

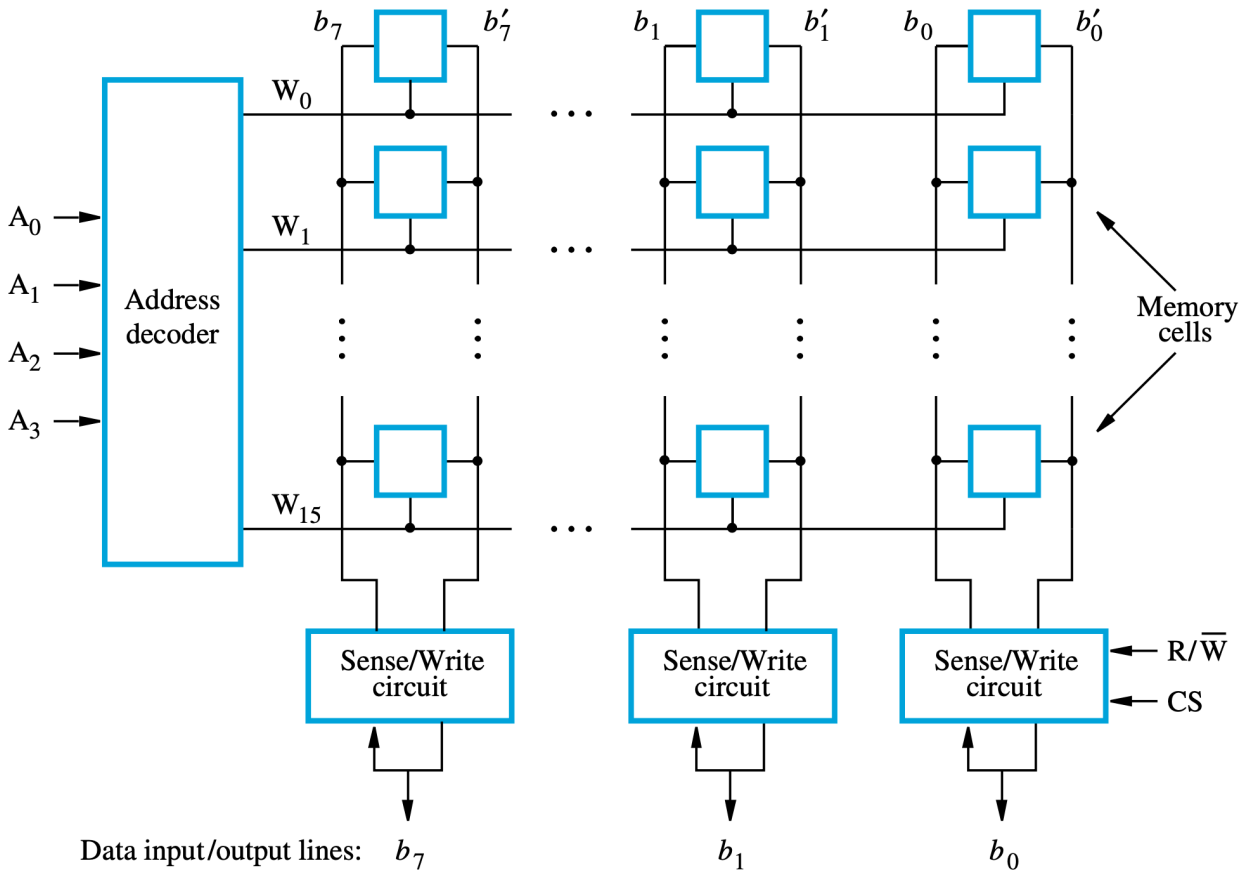


Figure 8.2 Organization of bit cells in a memory chip.

Static Memories

Memories that consist of circuits capable of retaining their state as long as power is applied are known as *static memories*.

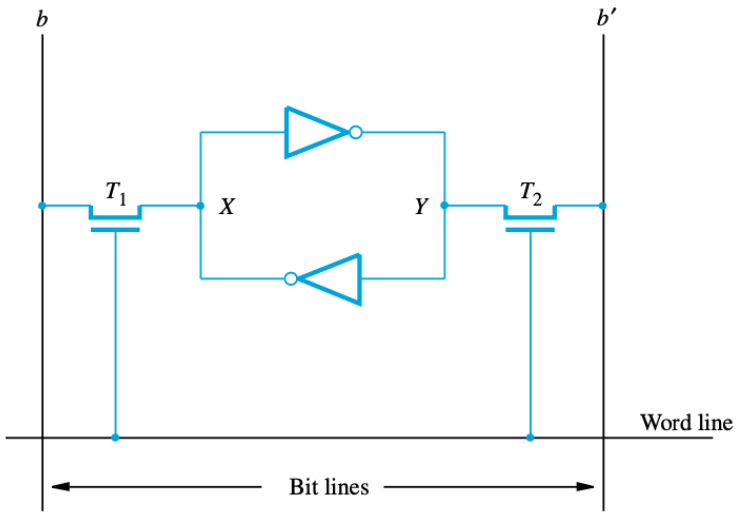


Figure 8.4 A static RAM cell.

- Two inverters are cross-connected to form a latch.
- The latch is connected to two bit lines by transistors *T*1 and *T*2.
- These transistors act as switches that can be opened or closed under control of the word line.
- When the word line is at ground level, the transistors are turned off and the latch retains its state.

For example, 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. Assume that this state represents the value 1.

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 bit line b' is low. The opposite is true if the cell is in state 0. Thus, b and b' are always complements of each other. The Sense/Write circuit at the end of the two bit lines monitors their state and sets the corresponding output accordingly.

Write Operation

During a Write operation, the Sense/Write circuit drives bit lines b and b', instead of sensing their state. It places the appropriate value on bit line b and its complement on b' and activates the word line. This forces the cell into the corresponding state, which the cell retains when the word line is deactivated.

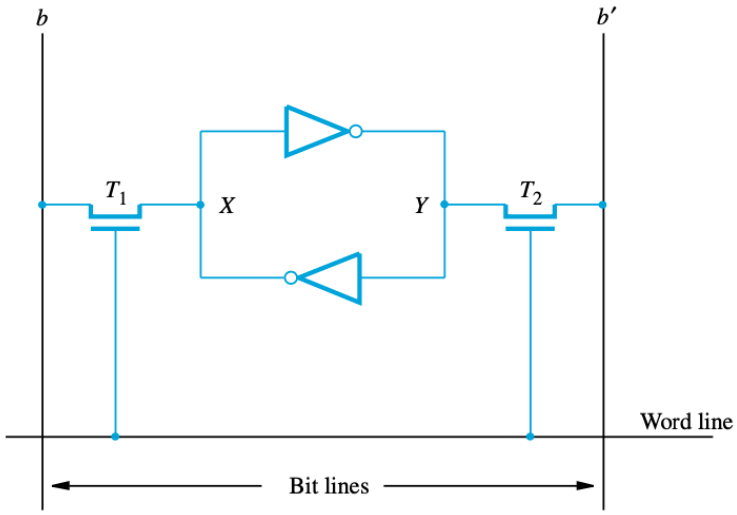


Figure 8.4 A static RAM cell.

CMOS Cell

Transistor pairs ($T3$, $T5$) and ($T4$, $T6$) form the inverters in the latch

in state 1, the voltage at point X is maintained high by having transistors $T3$ and $T6$ on, while $T4$ and $T5$ are off.

If $T1$ and $T2$ are turned on, bit lines b and b' will have high and low signals, respectively.

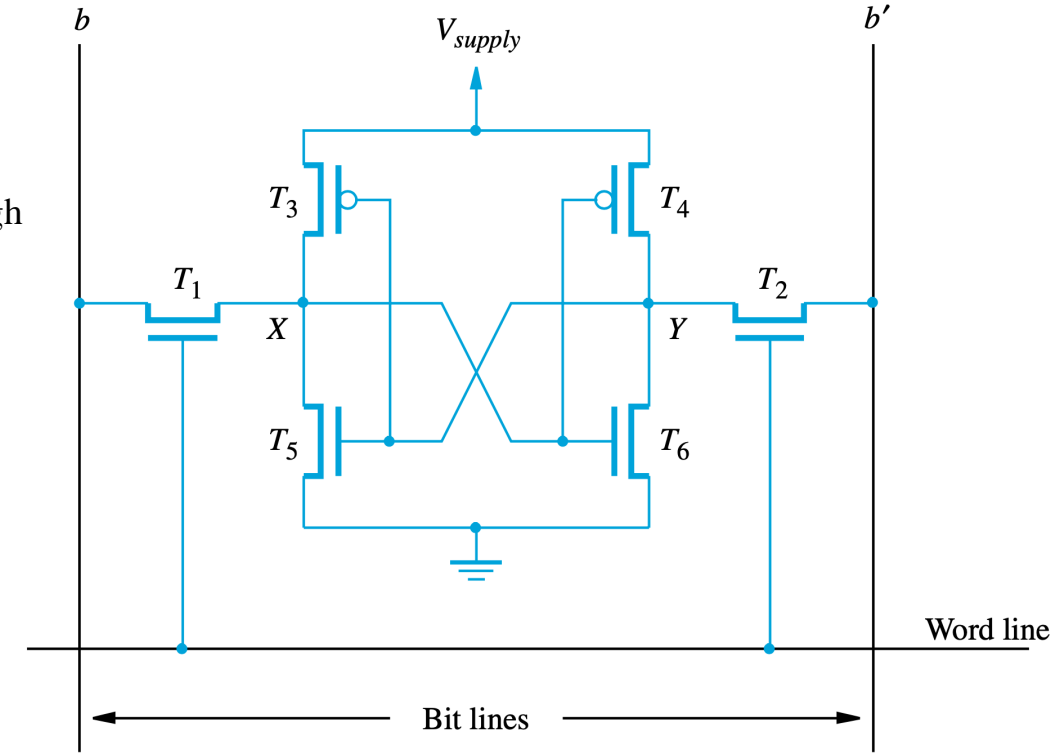


Figure 8.5 An example of a CMOS memory cell.

Functionality:

- Continuous power is needed for the cell to retain its state.
- If power is interrupted, the cell's contents are lost.
- When power is restored, the latch settles into a stable state,
- Not necessarily the same state the cell was in before the interruption.
- Hence, SRAMs are said to be *volatile* memories because their contents are lost when power is interrupted.

A major advantage of CMOS SRAMs is their 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 inverter are turned off, ensuring that there is no continuous electrical path between *Vsupply* and ground. Static RAMs can be accessed very quickly. Access times on the order of a few nanoseconds are found in commercially available chips.

Dynamic RAMs

- Static RAMs are fast, but their cells require several transistors.
- Less expensive and higher density RAMs can be implemented with simpler cells.
- But, these simpler cells do not retain their state for a long period, unless they are accessed frequently for Read or Write operations.
- Memories that use such cells are called *dynamic RAMs* (DRAMs).

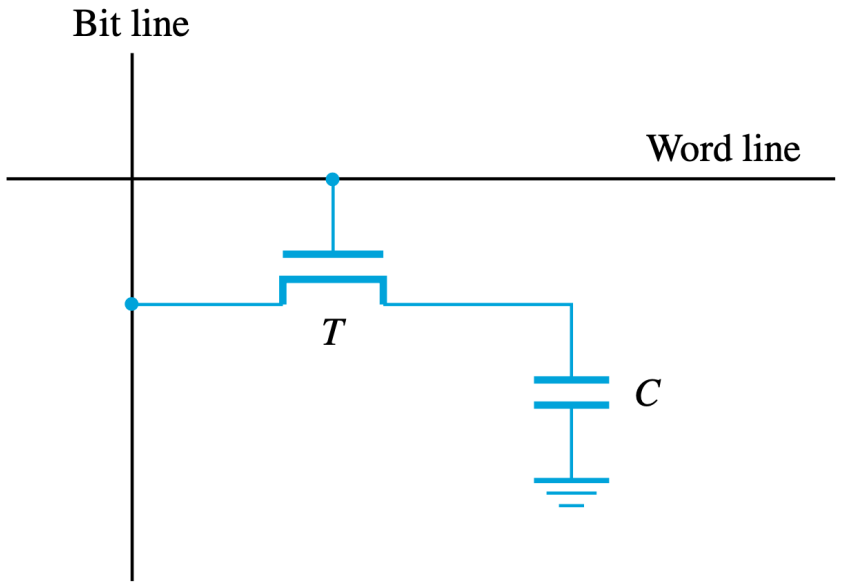


Figure 8.6 A single-transistor dynamic memory cell.

Internal organization of a 32M × 8 dynamic memory chip.

- The cells are organized in the form of a 16K × 16K array.
- The 16,384 cells in each row are divided into 2,048 groups of 8, forming 2,048 bytes of data.
- 14 address bits are needed to select a row
- Therefore,, and another 11 bits are needed to specify a group of 8 bits in the selected row.
- In total, a 25-bit address is needed to access a byte in this memory. The high-order 14 bits and the low-order 11 bits of the address constitute the row and column addresses of a byte, respectively.
- To reduce the number of pins needed for external connections, the row and column addresses are multiplexed on 14 pins.

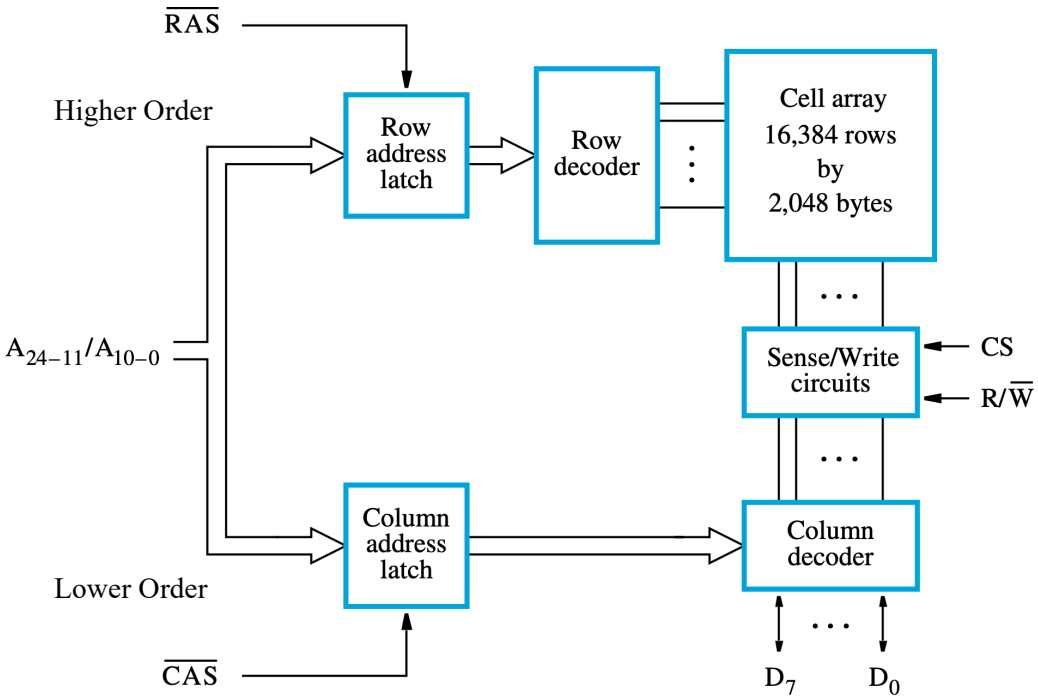


Figure 8.7 Internal organization of a 32M × 8 dynamic memory chip.

Read or a Write operation

- The row address is applied first. It is loaded into the row address latch in response to a signal pulse on an input control line called the Row Address Strobe (RAS). This causes a Read operation to be initiated, in which all cells in the selected row are read and refreshed.

- Shortly after the row address is loaded, the column address is applied to the address pins and loaded into the column address latch under control of a second control line called the Column Address Strobe (CAS).

- The information in this latch is decoded and the appropriate group of 8 Sense/Write circuits is selected.

- If the R/W control signal indicates a Read operation, the output values of the selected circuits are transferred to the data lines, D7–0.

- For a Write operation, the information on the D7–0 lines is transferred to the selected circuits, then used to overwrite the contents of the selected cells in the corresponding 8 columns.

Fast Page Mode

the contents of all 16,384 cells in the selected row are sensed, but only 8 bits are placed on the data lines, D7–0.

This byte is selected by the column address, bits A10–0. A simple addition to the circuit makes it possible to access the other bytes in the same row without having to reselect the row.

Each sense amplifier also acts as a latch. When a row address is applied, the contents of all cells in the selected row are loaded into the corresponding latches.

Then, it is only necessary to apply different column addresses to place the different bytes on the data lines.

All bytes in the selected row can be transferred in sequential order by applying a consecutive sequence of column addresses under the control of successive CAS signals. Thus, a block of data can be transferred at a much faster rate than can be achieved for transfers involving random addresses. The block transfer capability is referred to as the *fast page mode* feature.

A large block of data is often called a page

Latency

The memory access time defined earlier is not sufficient for describing the memory's performance when transferring blocks of data. During block transfers, *memory latency* is the amount of time it takes to transfer the first word of a block.

The time required to transfer a complete block depends also on the rate at which successive words can be transferred and on the size of the block. The time between successive words of a block is much shorter than the time needed to transfer the first word.

Bandwidth

The number of bits or bytes that can be transferred in one second. This measure is often referred to as the memory *bandwidth*. It depends on the speed of access to the stored data and on the number of bits that can be accessed in parallel.

Read-only Memories

Both static and dynamic RAM chips are volatile, which means that they retain information only while power is turned on. There are many applications requiring memory devices that retain the stored information when power is turned off.

Different types of non-volatile memories have been developed. Generally, their contents can be read in the same way as for their volatile counterparts discussed above. But, a special writing process is needed to place the information into a non-volatile memory. Since its normal operation involves only reading the stored data, a memory of this type is called a *read-only memory* (ROM).

A memory is called a read-only memory, or ROM, when information can be written into it only once at the time of manufacture.

A logic value 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.

To read the state of the cell, the word line is activated to close the transistor switch.

As a result, the voltage on the bit line drops 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 level, indicating a 1.

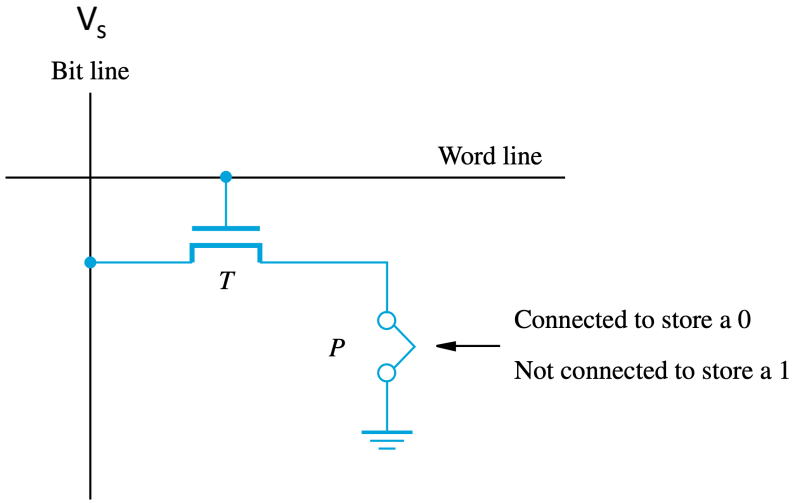


Figure 8.11 A ROM cell.

PROM

Some ROM designs allow the data to be loaded by the user, thus providing a *programmable ROM* (PROM).

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.

This process is irreversible.

PROMs provide flexibility and convenience not available with ROMs.

The cost of preparing the masks needed for storing a particular information pattern makes ROMs cost- effective only in large volumes.

The alternative technology of PROMs provides a more convenient and considerably less expensive approach, because memory chips can be programmed directly by the user.

EPROM

- It allows the stored data to be erased and new data to be written into it. Such an *erasable*, reprogrammable ROM is usually called an *EPROM*.

- It provides considerable flexibility during the development phase of digital systems.

- Since EPROMs are capable of retaining stored information for a long time, they can be used in place of ROMs or PROMs while software is being developed.

- The transistor is normally turned off, creating an open switch.
- It can be turned on by injecting charge into it that becomes trapped inside. Thus, an EPROM cell can be used to construct a memory in the same way as the previously discussed ROM cell.
- Erasure requires dissipating the charge trapped in the transistors that form the memory cells.
- This can be done by exposing the chip to ultraviolet light, which erases the entire contents of the chip. To make this possible, EPROM chips are mounted in packages that have transparent windows.

EEPROM

An EPROM must be physically removed from the circuit for reprogramming. Also, the stored information cannot be erased selectively. The entire contents of the chip are erased when exposed to ultraviolet light.

- Another type of erasable PROM can be programmed, erased, and reprogrammed electrically.
- Such a chip is called an *electrically erasable* PROM, or EEPROM.
- It does not have to be removed for erasure.
- It is possible to erase the cell contents selectively.

Disadvantage: different voltages are needed for erasing, writing, and reading the stored data, which increases circuit complexity.

Memory Hierarchy

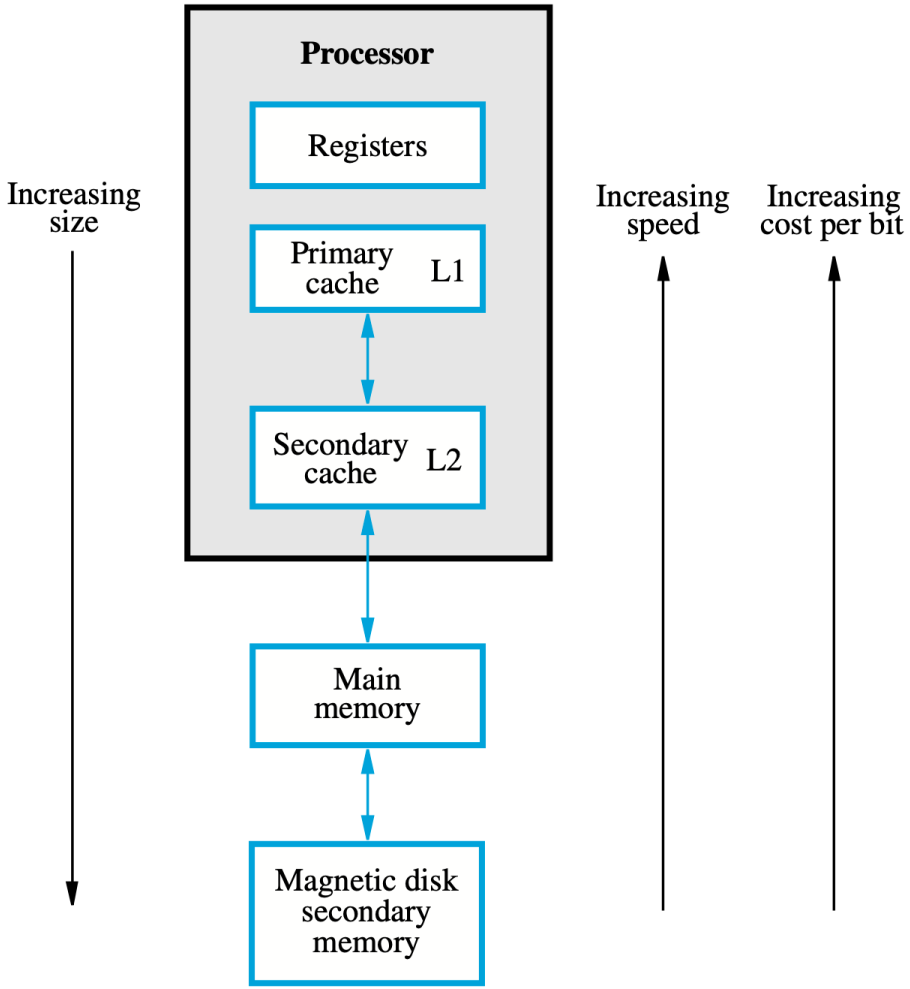


Figure 8.14 Memory hierarchy.

Cache Memories

- The cache is a **small and very fast memory**, interposed between the processor and the main memory.
- Its purpose is to make the **main memory appear much faster** to the processor.
- The effectiveness of this approach is based on a property of computer programs called ***locality of reference***.

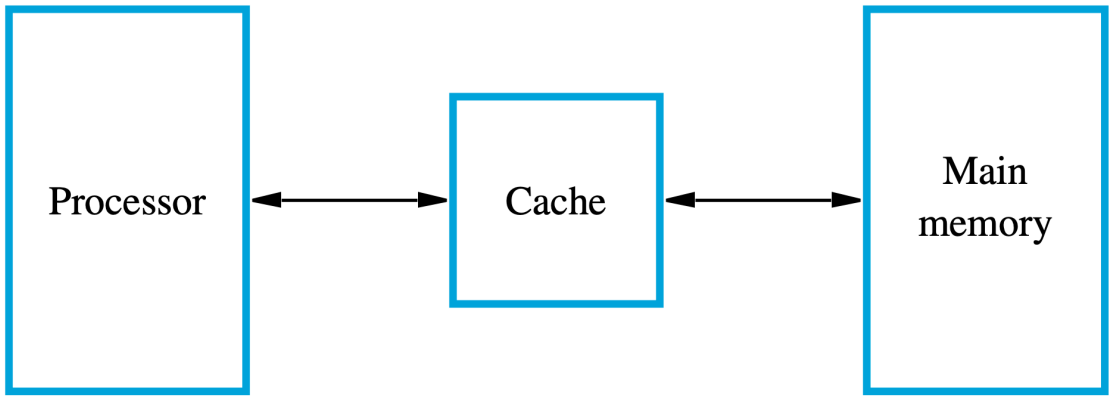


Figure 8.15 Use of a cache memory.

Cache Block and Cache Line

- The memory control circuitry is designed to take advantage of locality of reference.
- Whenever an information item, instruction or data, is first needed, this item should be brought into the cache, because it is likely to be needed again soon.
- Spatial locality suggests that instead of fetching just one item from the main memory to the cache, it is useful to fetch several items that are located at adjacent addresses as well.
- The term *cache block* refers to a set of contiguous address locations of some size.
- Another term that is often used to refer to a cache block is a *cache line*.

Mapping Functions

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. The correspondence between the main memory blocks and those in the cache is specified by a *mapping function*.

Replacement Algorithm

When the cache is full and a memory word (instruction or 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 cache's *replacement algorithm*.

Cache Hits

The cache control circuitry determines whether the requested word currently exists in the cache. If it does, the Read or Write operation is performed on the appropriate cache location. In this case, a *read* or *write hit* is said to have occurred.

Cache Miss

A Read operation for a word that is not in the cache constitutes a *Read miss*. It causes the block of words containing the requested word to be copied from the main memory into the cache. After the entire block is loaded into the cache, the particular word requested is forwarded to the processor.

Cache Mapping Functions

Consider a cache consisting of

The main memory has 64K words, which we will view as 4096 blocks of 16 words each.
This will be divided into

- 128 section blocks of 16 words each for a total of 2048 (2K) words
- Assume that the main memory is addressable by a 16-bit address.

Direct Mapping Functions

- In this technique, block *j* of the main memory maps onto block *j* modulo 128 of the cache.
- Main memory blocks **0, 128, 256, . . .** is loaded into the cache, it is stored in **cache block 0**.
- Blocks **1, 129, 257, . . .** are stored in **cache block 1**, and so on.
- Since more than one memory block is mapped onto a given cache block position, contention may arise for that position even when the cache is not full.
- Contention is resolved by allowing the new block to overwrite the currently resident block.
- Placement of a block in the cache is determined by its memory address. The memory address can be divided into three fields .

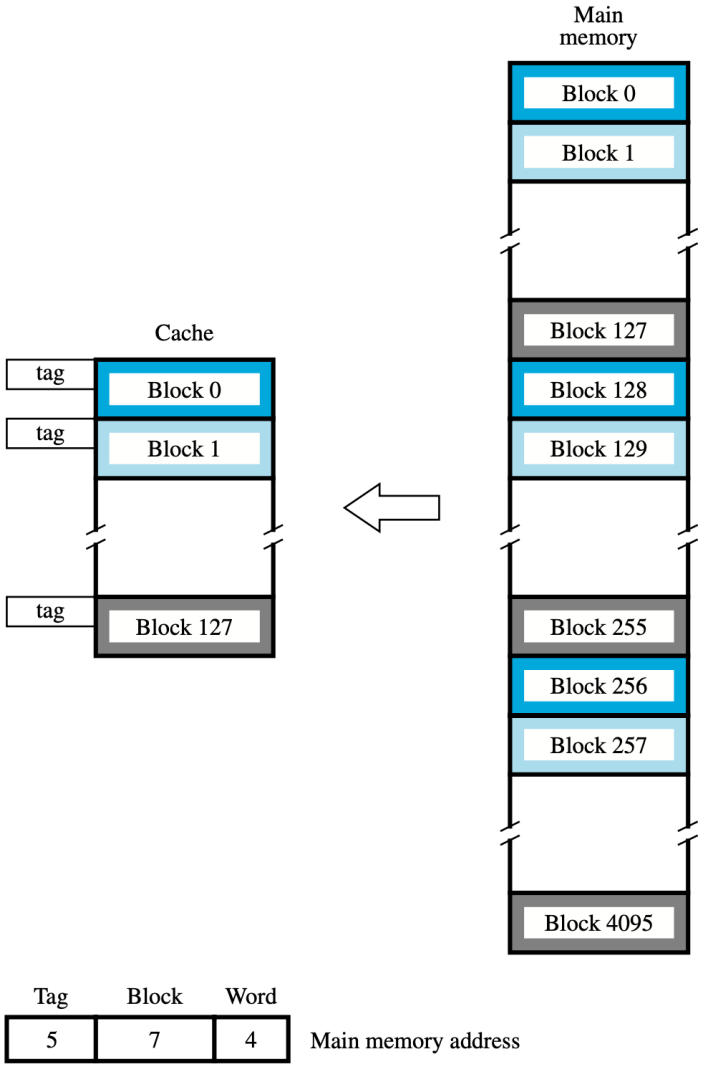
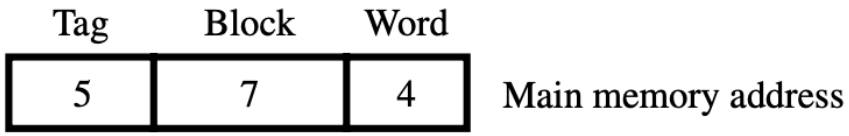


Figure 8.16 Direct-mapped cache.

Direct Mapping Functions



Tag: Used to define the 32 memory segments.
For 32 sections 5 bits are required for addressing.

Block: Used to define the data from block 0 to block 127.
For 0 to 127 total 128 blocks 7 bits are required for addressing.

Word: Define specific word within a block (here, each block has 16 words)
For each block of 16 words 4 bits are required to address specific word.

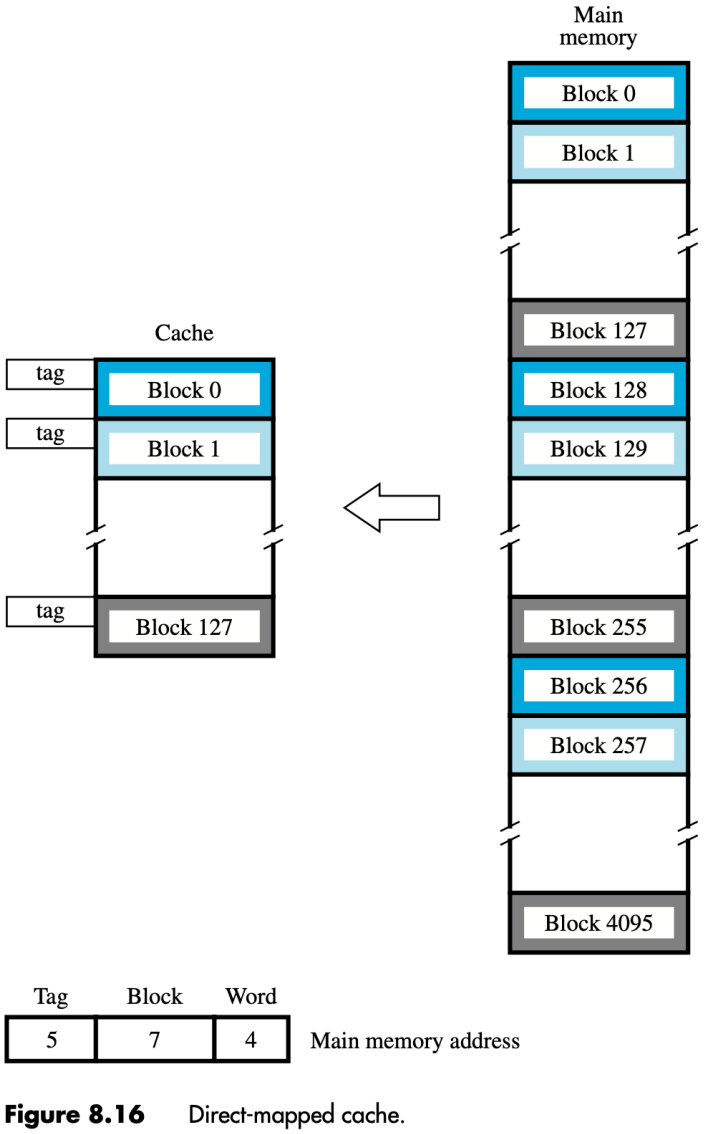


Figure 8.16 Direct-mapped cache.

Direct Mapping Functions

As execution proceeds, the 7-bit cache block field of each address generated by the processor points to a particular block location in the cache.

he high-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 be read from the main memory and loaded into the cache.

The direct-mapping technique is easy to implement, but it is not very flexible.

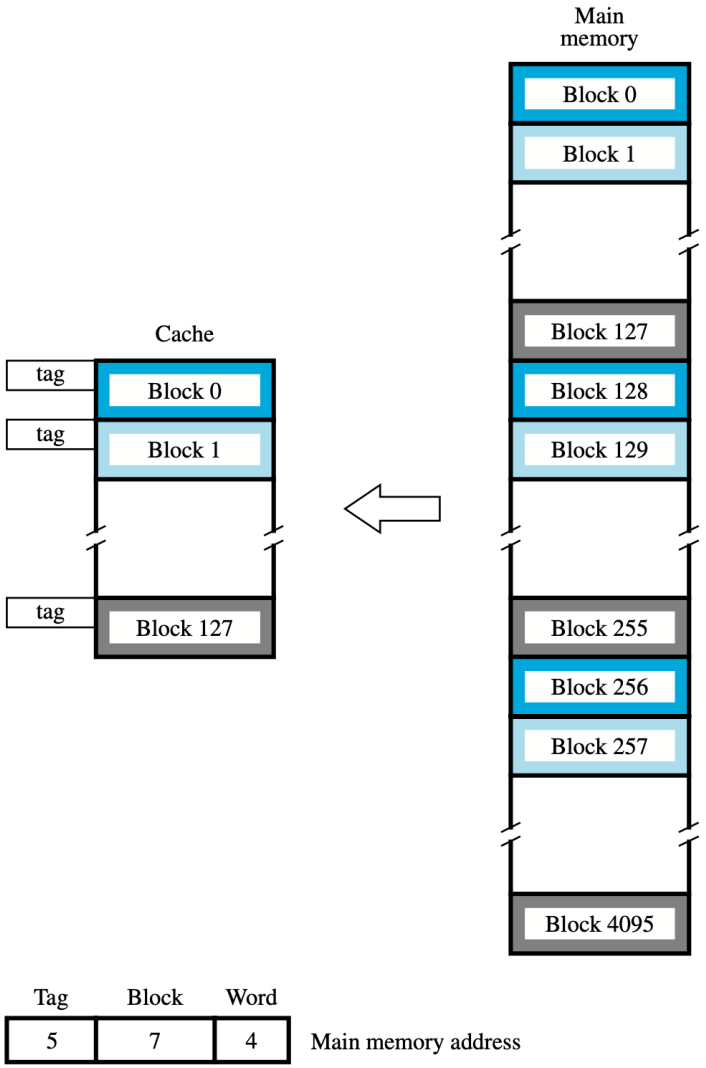


Figure 8.16 Direct-mapped cache.

Associative Mapping Functions

The most flexible mapping method, in which a main memory block can be placed into any cache block position.

12 tag bits are required to identify a memory block when it is resident in the cache.

The tag bits of an address received from the processor are compared to the tag bits of each block of the cache to see if the desired block is present.

It gives complete freedom in choosing the cache location in which to place the memory block, resulting in a more efficient use of the space in the cache.

When a new block is brought into the cache, it replaces (ejects) an existing block only if the cache is full.

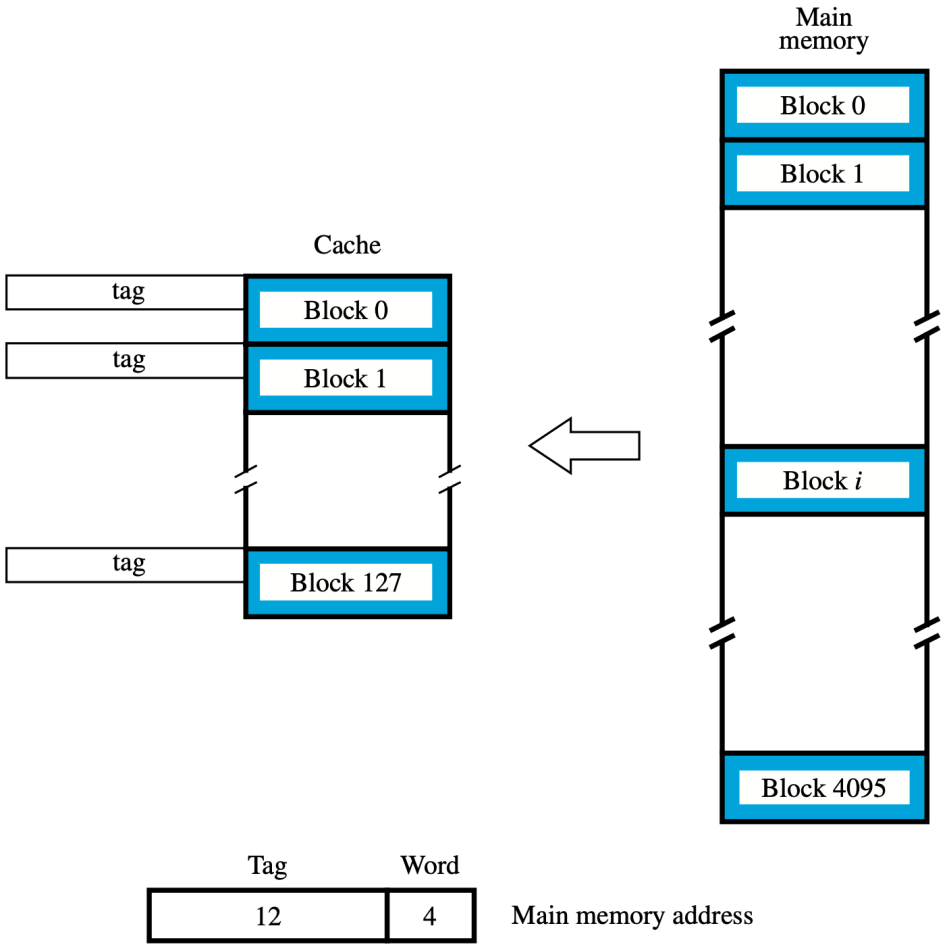


Figure 8.17 Associative-mapped cache.

Set Associative Mapping Functions

combination of the direct- and associative-mapping techniques.

The blocks of the cache are grouped into sets, and the mapping allows a block of the main memory to reside in any block of a specific set.

Hence, the contention problem of the direct method is eased by having a few choices for block placement.

In this case, memory blocks 0, 64, 128, . . . , 4032 map into cache set 0, and they can occupy either of the two block positions within this set.

Having 64 sets means that the 6-bit set field of the address determines which set of the cache might contain the desired block.

The tag field of the address must then be associatively compared to the tags of the two blocks of the set to check if the desired block is present. This two-way associative search is simple to implement.

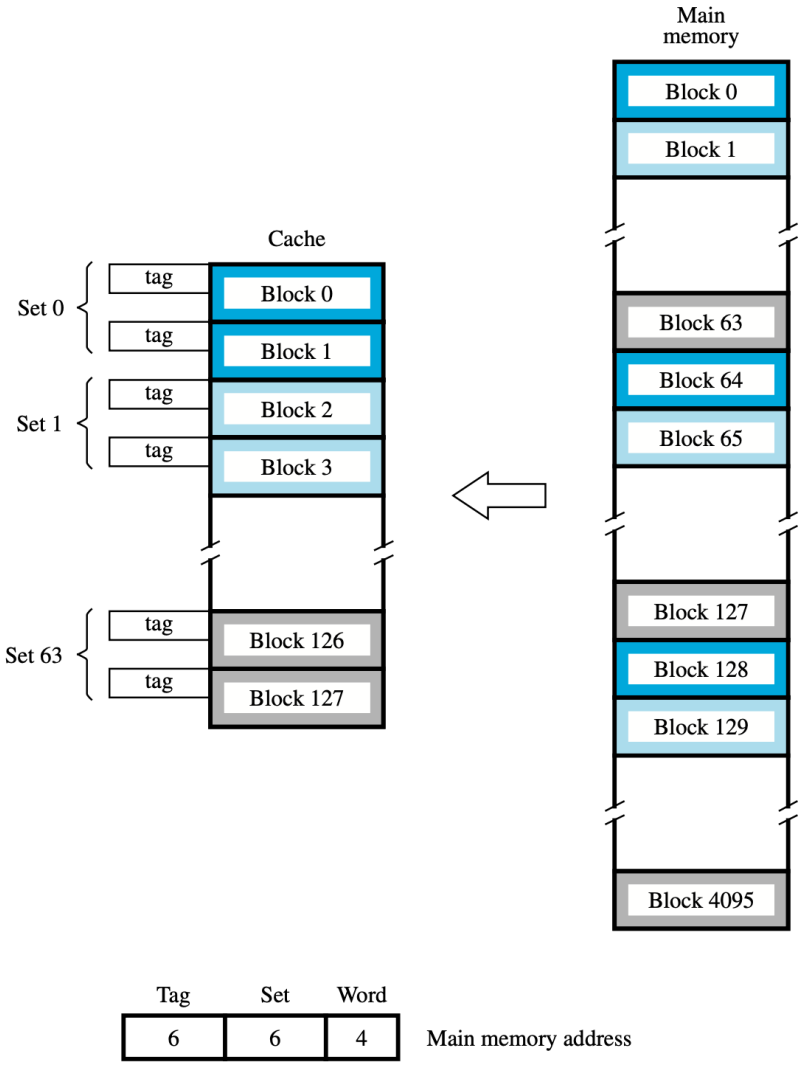


Figure 8.18 Set-associative-mapped cache with two blocks per set.

Hit Rate and Miss Penalty

An excellent indicator of the effectiveness of a particular implementation of the memory hierarchy is the **success rate in accessing information** at various levels of the hierarchy. Recall that a **successful access to data in a cache is called a hit**.

- The number of hits stated as a fraction of all attempted accesses is called the *hit rate*
- the *miss rate* is the number of misses stated as a fraction of attempted accesses.

High hit rates well over 0.9 are essential for high-performance computers.

Performance is adversely affected by the actions that need to be taken when a miss occurs. A performance penalty is incurred because of the extra time needed to bring a block of data from a slower unit in the memory hierarchy to a faster unit. During that period, the processor is stalled waiting for instructions or data. The waiting time depends on the details of the operation of the cache. For example, it depends on whether or not the load-through approach is used. We refer to the total access time seen by the processor when a miss occurs as the *miss penalty*.

H - hit rate; M - miss penalty; and C - time to access information in the cache

Thus, the average access time

$$t_{avg} = hC + (1 - h)M$$

Virtual Memory

The physical main memory is not as large as the address space of the processor.

If a program does not completely fit into the main memory, the parts of it not currently being executed are stored on a secondary storage device, typically a magnetic disk.

As these parts are needed for execution, they must first be brought into the main memory, possibly replacing other parts that are already in the memory. These actions are performed automatically by the operating system, using a scheme known as *virtual memory*.

- The binary addresses that the processor issues for either instructions or data are called *virtual addresses*.
- A special hardware unit, called the *Memory Management Unit* (MMU), keeps track of which parts of the virtual address space are in the physical memory.

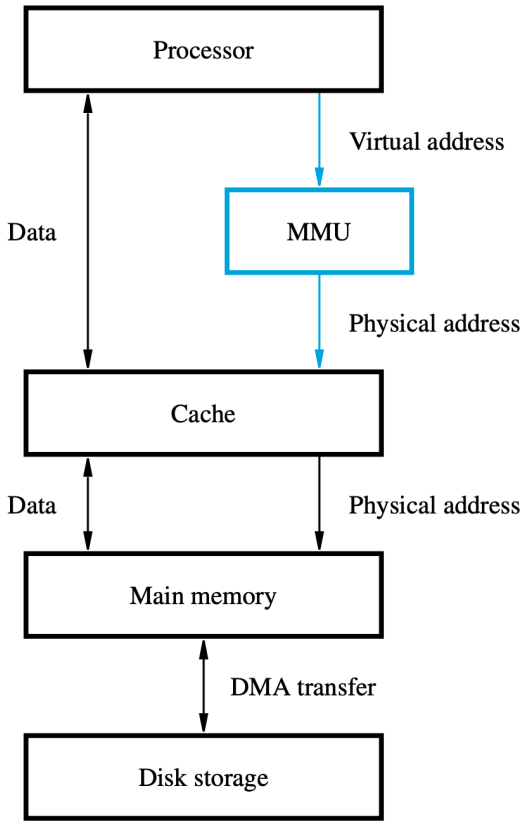


Figure 8.24 Virtual memory organization.

- A simple method for translating virtual addresses into physical addresses is to assume that all programs and data are composed of fixed-length units called *pages*,
- Each of which consists of a block of words that occupy contiguous locations in the main memory.
- Pages commonly range from 2K to 16K bytes in length.
- They constitute the basic unit of information that is transferred between the main memory and the disk whenever the MMU determines that a transfer is required.
- Pages should not be too small, because the access time of a magnetic disk is much longer (several milliseconds) than the access time of the main memory.
- On the other hand, if pages are too large, it is possible that a substantial portion of a page may not be used, yet this unnecessary data will occupy valuable space in the main memory.

Each virtual address generated by the processor, whether it is for an instruction fetch or an operand load/store operation, is interpreted as a *virtual page number* (high-order bits) followed by an *offset* (low-order bits) that specifies the location of a particular byte (or word) within a page.

Page Table: Information about the main memory location of each page is kept in a *page table*. This information includes the main memory address where the page is stored and the current status of the page.

Page Frame: An area in the main memory that can hold one page is called a *page frame*.

Page Table Base Register: The starting address of the page table is kept in a *page table base register*. By adding the virtual page number to the contents of this register, the address of the corresponding entry in the page table is obtained.

Control Bits: Each entry in the page table also includes some control bits that describe the status of the page while it is in the main memory.

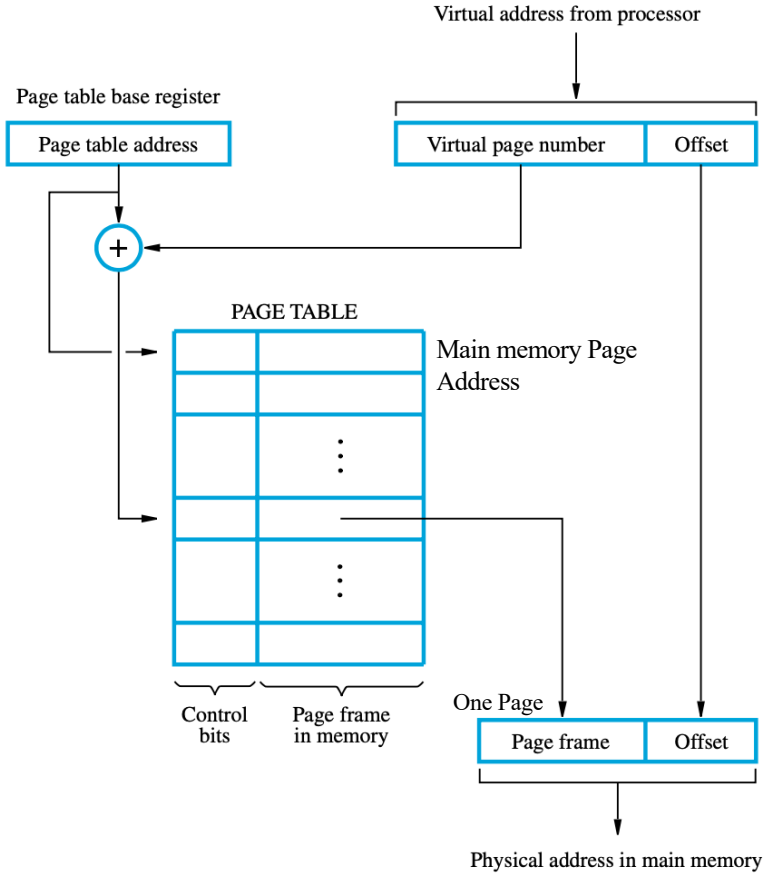


Figure 8.25 Virtual-memory address translation.